



Directora  
**Myriam Guerra Balic**

Doctorando  
**Xavier de Blas Foix**

## **Tesis doctoral**

---

Proyecto Chronojump-Boscosystem.  
Herramienta informática libre  
para el estudio cinemático  
del salto vertical.





**Universitat Ramon Llull**

CLIF G. 59069740 Universitat Ramon Llull Fundació Privada. Regre. Fund. Generalitat de Catalunya núm. 472 (28-02-90)

## **TESIS DOCTORAL**

**Título Proyecto Chronojump-Boscosystem. Herramienta libre para el estudio cinemático del salto vertical.**

**Realizada por Xavier de Blas Foix**

**en el Centro Facultat de Psicologia i Ciències de l'Educació i de l'Esport Blanquerna**

**y en el Departamento Ciències de l'Activitat Física i de l'Esport**

**Dirigida por Dra. Miriam Guerra Balic**





Dedicat a l'Alberto García-Fojeda.

Ho veus, Alberto?

Ja funciona!

La gent ho fa servir!

---

*La Facultat de Psicologia, Ciències de l'Educació i de l'Esport Blanquerna ha atorgat un  
quadrimestre sabàtic per a la finalització d'aquesta tesi.*

---

## Cómo se ha escrito este documento

Esta tesis se ha redactado usando LyX, una interfaz gráfica o frontend de  $\text{\LaTeX}$ . Las referencias bibliográficas se han mantenido desde JabRef y se han insertado en el documento  $\text{\LaTeX}$  gracias a Bibtex8 y Biblatex. Además se han usado los paquetes Appendix, Babel, Chngcntr, Epigraph, Geometry, Gmetric, Multicol, Nameref, Fancybox, T1 y Tocloft.

La portada y la contraportada han sido compuestas con Inkscape usando la fotografía “Les deu i deu minuts” de Roger Ferrer Ibáñez (*rofi* a Flickr) licenciada como Creative Commons BY-NC-SA 2.0. <http://www.flickr.com/photos/rofi/6836879696/>. Dichas imágenes se han integrado con PDF Editor.

Las ilustraciones de tipo diagrama (exceptuando la figura 1.1) han sido diseñadas con Dia, Inkscape, Kig, Kicad y R. Las capturas de pantalla han sido realizadas a partir de programas diseñados en la tesis y retocadas con Gimp. Las gráficas estadísticas provienen de R.

El diagrama de aportaciones (figura 1.1) se ha generado con Gnuclad, después de realizar algunas adaptaciones en su código.

Las nubes de etiquetas al inicio del marco teórico (figura 4.1) se han realizado con un script generado ad-hoc denominado latextotagcloud.pl y documentado aquí: <http://tex.stackexchange.com/questions/917/wordle-like-word-clouds/21713#21713>

Todos los programas citados tienen licencia libre y se han usado bajo el sistema operativo libre: Linux.

# Índice

<b>1. Agradecimientos</b>	<b>1</b>
<b>2. Introducción</b>	<b>9</b>
2.1. Justificación del trabajo . . . . .	9
2.2. Tema concreto . . . . .	10
2.3. Objetivos . . . . .	11
 <b>Parte I   Fundamentación teórica</b>	 <b>13</b>
<b>3. Estado actual de la cuestión</b>	<b>15</b>
3.1. Medición a partir de la distancia . . . . .	15
3.2. Medición a partir del tiempo . . . . .	16
3.3. Medición a partir de la aceleración . . . . .	16
3.4. Medición a partir de filmación . . . . .	20
3.5. Coste de las herramientas . . . . .	21
3.6. Licencia de las herramientas . . . . .	21
3.7. Compartición de datos . . . . .	21
 <b>4. Marco teórico</b>	 <b>23</b>
 <b>Parte II   Estudios de medición del tiempo</b>	 <b>69</b>
<b>5. Desarrollo y validación de un microcontrolador</b>	<b>71</b>
5.1. Objetivo . . . . .	71
5.2. Metodología . . . . .	71
5.3. Resultados y discusión . . . . .	74
 <b>6. Desarrollo y validación de una plataforma de contactos</b>	 <b>80</b>
6.1. Objetivo . . . . .	80
6.2. Metodología . . . . .	80
6.2.1. Creación de plataformas de contactos . . . . .	81
6.2.2. Validación por determinación de la presión mínima . . . . .	82
6.2.3. Validación por comparación de los tiempos de activación/desactivación . . . . .	84
6.3. Resultados y discusión . . . . .	86
6.3.1. Creación de la plataforma de contactos de Chronojump-Boscosystem . . . . .	86
6.3.2. Validación por determinación de la presión mínima . . . . .	87
6.3.3. Validación por comparación de los tiempos de activación/desactivación . . . . .	88

<b>7. Desarrollo y validación de un software de ordenador</b>	<b>95</b>
7.1. Objetivo . . . . .	95
7.2. Metodología . . . . .	95
7.2.1. Iteraciones . . . . .	95
7.2.2. Historias de usuario . . . . .	96
7.2.3. Herramientas para la creación del software . . . . .	99
7.2.4. Validación del software . . . . .	100
7.3. Resultados y discusión . . . . .	100
<b>8. Distribución e internacionalización</b>	<b>111</b>
8.1. Objetivos . . . . .	111
8.2. Metodología . . . . .	111
8.3. Resultados y discusión . . . . .	112
8.3.1. Resultados pertinentes a las herramientas diseñadas . . . . .	112
8.3.2. Resultados correspondientes al uso de las herramientas . . . . .	114
8.3.3. Presencia en actos académicos y publicaciones. . . . .	118
 <b>Parte III Estudios de detección sin marcadores del ángulo de flexión de la articulación de la rodilla</b>	 <b>125</b>
<b>9. Desarrollo y validación de software de seguimiento</b>	<b>127</b>
9.1. Objetivo . . . . .	127
9.2. Metodología . . . . .	127
9.2.1. Especificaciones . . . . .	127
9.2.2. Obtención de imágenes . . . . .	128
9.2.3. Desarrollo del software . . . . .	129
9.2.4. Validación . . . . .	130
9.3. Resultados y discusión . . . . .	131
<b>10. Creación de un modelo de predicción</b>	<b>136</b>
10.1. Objetivo . . . . .	136
10.2. Metodología . . . . .	136
10.2.1. Diseño del experimento . . . . .	137
10.2.2. Ubicación de los marcadores . . . . .	138
10.2.3. Comparación entre saltos según indumentaria . . . . .	139
10.2.4. Mortalidad de la muestra . . . . .	140
10.2.5. Variables predictivas . . . . .	140
10.2.6. Saltos descartados . . . . .	144
10.2.7. Diseño del modelo . . . . .	146

10.3. Resultados y discusión . . . . .	146
10.3.1. Resultados . . . . .	146
10.3.2. Validación . . . . .	146
10.3.3. Implementación del modelo en software de visión por computador . . . . .	152
<b>Parte IV Estudios de elaboración de tablas de percentiles</b>	<b>153</b>
<b>11.Creación de criterios de fiabilidad de los datos</b>	<b>155</b>
11.1. Objetivos . . . . .	155
11.2. Metodología . . . . .	155
11.2.1. Formación del grupo de discusión . . . . .	155
11.2.2. Primera reunión del grupo de discusión . . . . .	156
11.2.3. Segunda reunión del grupo de discusión . . . . .	156
11.3. Resultados y discusión . . . . .	157
11.3.1. Relativos a la diferenciación entre niveles de fiabilidad de los datos según su origen . . . . .	157
11.3.2. Relativos al establecimiento de identificadores sobre datos erróneos según el propio dato . . . . .	158
<b>12.Desarrollo de software para compartir datos</b>	<b>160</b>
12.1. Objetivos . . . . .	160
12.2. Metodología . . . . .	160
12.3. Resultados y discusión . . . . .	161
<b>Parte V Conclusiones, limitaciones y líneas de futuro</b>	<b>167</b>
<b>13.Conclusiones</b>	<b>169</b>
<b>14.Limitaciones</b>	<b>172</b>
<b>15.Líneas de futuro</b>	<b>173</b>
15.1. En relación a los estudios de medición del tiempo . . . . .	173
15.2. En relación a los estudios de detección sin marcadores del ángulo de flexión de la articulación de la rodilla . . . . .	173
15.3. En relación a los estudios de elaboración de tablas de percentiles . . . . .	175
15.4. En relación a otras aplicaciones que se deriven de las tecnologías desarrolladas . . . . .	175
15.5. En relación al uso de las herramientas en la docencia . . . . .	176
<b>Parte VI Referencias bibliográficas</b>	<b>177</b>

<b>Parte VII</b>	<b>Índice alfabético</b>	<b>197</b>
<b>Parte VIII</b>	<b>Anexos</b>	<b>207</b>
<b>A.</b>	<b>Diagramas de las herramientas creadas</b>	<b>209</b>
A.1.	Planos de los microcontroladores . . . . .	209
A.2.	Diagramas de flujo del Firmware de Chronopic . . . . .	209
A.3.	Diagrama de la base de datos de Chronojump . . . . .	209
A.4.	Diagrama de las clases de Chronojump y ChronojumpMini . . . . .	209
<b>B.</b>	<b>Gráficos generados por el servidor Chronojump-Server</b>	<b>223</b>
<b>C.</b>	<b>Listas de materiales</b>	<b>228</b>
<b>D.</b>	<b>Ejemplos de aportaciones de la comunidad</b>	<b>229</b>
D.1.	Ejemplos en el Foro de Chronojump . . . . .	229
D.2.	Informe de fallos . . . . .	231
<b>E.</b>	<b>Documentos de texto vinculados al proyecto</b>	<b>233</b>
E.1.	Consentimiento informado de los experimentos (catalán) . . . . .	233
E.2.	Documentos de información para los miembros de la muestra en los experimentos (catalán) . . . . .	233
E.3.	Proyecto en el Master Internacional en Desarrollo de Aplicaciones en Software Libre UOC Universitat Oberta de Catalunya . . . . .	245

## Índice de tablas

3.1. Herramientas para la medición a partir de la distancia. . . . .	17
3.2. Herramientas para la medición a partir del tiempo. . . . .	18
3.3. Herramientas para la medición a partir de la aceleración. . . . .	19
3.4. Herramientas para la medición a partir del análisis automatizado de filmación. . . . .	20
4.1. Ángulos en las flexiones previas al salto . . . . .	31
4.2. Reproducibilidad de tests de salto . . . . .	33
4.3. Licencias más usadas en proyectos de software libre . . . . .	50
4.4. Licencias de software según los <i>Documentos Halloween</i> . . . . .	51
4.5. Lenguajes de programación populares en software libre . . . . .	55
4.6. Propósitos de la programación ágil . . . . .	58
4.7. Guión de pruebas (manual) en OpenOffice.org . . . . .	60
5.1. Frecuencias y tiempos equivalentes en la validación del microcontrolador . . . . .	73
5.2. Etapas de producción del microcontrolador . . . . .	74
6.1. Puntos evaluados en cada plataforma . . . . .	82
6.2. Muestra para la validación de las plataformas . . . . .	85
6.3. Sensibilidad de las distintas plataformas . . . . .	87
6.4. Coeficientes de correlación intraclase de las distintas situaciones . . . . .	91
7.1. Iteraciones representativas del software y sus objetivos . . . . .	96
7.2. Herramientas de software usadas en la creación del programa . . . . .	99
7.3. Lenguajes de programación usados en Chronojump . . . . .	102
7.4. Tests añadidos en el software . . . . .	106
7.5. Estadísticos de salto añadidos en el software . . . . .	108
7.6. Personas que han participado en las pruebas de Chronojump . . . . .	110
8.1. Traducciones del software . . . . .	113
8.2. Ejecuciones del software . . . . .	114
9.1. Variaciones de iluminación en las imágenes de prueba . . . . .	128
10.1. Muestra para la creación de un modelo de predicción . . . . .	137
10.2. Zonas en la predicción del ángulo . . . . .	150
11.1. Criterios para la acreditación de un evaluador como <i>fiable</i> . . . . .	157
11.2. Criterios para la acreditación de una sesión de recogida de datos como <i>fiable</i> . . . . .	158
12.1. Servicios web creados en Chronojump-Server . . . . .	162
C.1. Materiales de Chronopic 3 . . . . .	228
D.1. Informe de fallos en la web a partir de la discusión con el usuario del documento de la figura D.1 . . . . .	232

# Índice de figuras

1.1. Diagrama de aportaciones . . . . .	7
4.1. Autores y conceptos citados en el marco teórico . . . . .	23
4.2. Desequilibrio en la recepción de un salto Abalakov . . . . .	32
4.3. Esquema de la plataforma de contactos Ergojump Boscosystem y sus circuitos eléctricos equivalentes . . . . .	34
4.4. Diferencia en el despegue y la caída en el salto CMJ . . . . .	35
4.5. Obtención del tiempo de vuelo a partir de plataforma de fuerzas . . . . .	36
4.6. Aplicaciones de los microcontroladores . . . . .	37
4.7. Threshold alto y bajo . . . . .	41
4.8. Relación costes/ingresos en software como servicio o como producto . . . . .	44
4.9. Ejemplo de implementación de la programación orientada a objetos . . . . .	54
4.10. Evolución de los lenguajes de programación más populares a partir del índice TIOBE . . . . .	55
4.11. Ciclo de vida en la programación extrema . . . . .	61
4.12. Punto de vista de la evolución del software . . . . .	62
4.13. Skypic, una tarjeta de hardware estático con un microcontrolador PIC . . . . .	63
4.14. S2PROTO, una tarjeta que incorpora un FPGA . . . . .	63
5.1. Instrumentación para la validación de la sensibilidad del microcontrolador . . . . .	72
5.2. Chronopic 1 . . . . .	75
5.3. Chronopic 2 . . . . .	75
5.4. Chronopic 3. Fabricación en España . . . . .	76
5.5. Chronopic 3. Fabricación en Taiwán . . . . .	76
5.6. Software Chronopic-Firmware record . . . . .	77
5.7. Resultados de la validación de Chronopic 3 . . . . .	78
6.1. Instrumentación para la validación de la sensibilidad de las plataformas . . . . .	83
6.2. Disposición de las plataformas de contacto . . . . .	84
6.3. Composición de una fotografía de la plataforma rígida de tamaño A2 y dos de la plataforma flexible . . . . .	86
6.4. Sensibilidad de las plataformas diseñadas . . . . .	88
6.5. Validez y fiabilización de las plataformas propuestas . . . . .	89
6.6. Error estándar entre las plataformas . . . . .	90
6.7. Diferencia entre las plataformas en función del atleta . . . . .	90
6.8. Diferencia entre las plataformas en función de situación y atleta . . . . .	91
6.9. Gráfico Bland-Altman entre las dos plataformas rígidas A2 . . . . .	92
6.10. Gráfico Bland-Altman entre la plataforma rígida A2 y la plataforma de Bosco . . . . .	92
6.11. Gráfico Bland-Altman entre la plataforma flexible y la plataforma de Bosco . . . . .	93
6.12. Disposición de las plataformas de contacto A3np, A2, A3p . . . . .	94
6.13. Comparación entre los dos diseños de plataformas rígidas A3 . . . . .	94



7.1. Comparación del tamaño de Chronojump con otros proyectos de software libre . . . . .	101
7.2. Captura de pantalla de la versión 0.9.3 de Chronojump . . . . .	103
7.3. Diagrama de la base de datos de Chronojump en febrero de 2005 . . . . .	104
7.4. Diagrama de una selección de las tablas de la base de datos de Chronojump en marzo de 2011 . . . . .	105
7.5. Definir un salto simple en Chronojump . . . . .	105
7.6. Definir un salto reactivo en Chronojump . . . . .	106
7.7. Definir una carrera con tramos en Chronojump . . . . .	107
7.8. Feedback en Chronojump mediante campanas . . . . .	107
7.9. Llamada a R desde Chronojump . . . . .	109
8.1. Pings según las versiones instaladas del software Chronojump desde la activación del servidor . . . . .	115
8.2. Pings según las versiones de los sistemas operativos sobre los que se ejecuta Chronojump desde la activación del servidor . . . . .	116
8.3. Distribución en Chile por parte de terceros . . . . .	117
8.4. Distribución en Holanda por parte de terceros . . . . .	118
8.5. Cartel de la 1a Jornada Inter-universitaria de Chronojump en la UVic 2008 . . . . .	123
8.6. Cartel de la 2a Jornada Inter-universitaria de Chronojump en Blanquerna (Universitat Ramon Llull) 2009 . . . . .	124
8.7. Diagrama del proceso de detección del ángulo de la rodilla . . . . .	126
9.1. Reconocimiento de marcadores sin el uso de las mallas . . . . .	131
9.2. Reconocimiento de mallas negras y marcadores . . . . .	132
9.3. Reconocimiento de marcadores cuando un punto sale de las mallas y contacta con otra prenda blanca . . . . .	133
9.4. Identificación del centro del marcador cuando existe deformación . . . . .	134
10.1. Tiempo de vuelo y ángulo máximo de flexión en función de la indumentaria usada . . . . .	140
10.2. <i>Ángulo real</i> respecto a <i>tightsHP</i> en los distintos saltos . . . . .	141
10.3. <i>Ángulo real</i> respecto a <i>tightsHP</i> en la zona de 90 grados de <i>ángulo real</i> . . . . .	142
10.4. <i>Ángulo real</i> respecto a <i>tightsHP</i> y según intervalos de <i>kpfY</i> en la zona de 90 grados de <i>ángulo real</i> . . . . .	142
10.5. Relación entre <i>Ángulo real</i> , <i>tightsHP</i> y <i>kpfY</i> en los momentos de máxima extensión y de flexión de 90 grados en dos saltos de individuos distintos . . . . .	143
10.6. Diagrama ilustrativo sobre la relación trigonométrica entre <i>Ángulo real</i> , <i>tightsHP</i> y <i>kpfY</i> en los momentos de máxima extensión y de flexión de 90 grados . . . . .	143
10.7. Diagramas de caja de <i>kpfY</i> en la zona de 80 a 100 grados de <i>ángulo real</i> . . . . .	144
10.8. Regresiones entre las rectas <i>tightsHP</i> y <i>ángulo real</i> en las flexiones que pasan por el intervalo de 80-100 grados . . . . .	145

10.9. Regresiones entre las rectas <i>tightsHP</i> y <i>ángulo real</i> en las flexiones que pasan por el intervalo de 80-100 grados habiendo borrado las ejecuciones de curvatura muy diferenciada . . . . .	145
10.10Residuos de la aplicación del modelo . . . . .	147
10.11Residuos de la aplicación del modelo por individuo . . . . .	147
10.12Residuos de la aplicación del modelo por salto . . . . .	148
10.13Error absoluto en la predicción del ángulo entre 80-100 grados de flexión . . . . .	148
10.14Error absoluto en la predicción del ángulo en todo el rango de flexión . . . . .	149
10.15Error absoluto en la predicción del ángulo en todo el rango de flexión en las flexiones descartadas para el entrenamiento del modelo por no alcanzar los 90 grados	150
10.16Error en la predicción en rangos de ángulos . . . . .	151
10.17Predicción del ángulo en el programa de visión por computador . . . . .	152
12.1. Diagrama de la integración de Chronojump-Server en un servidor con máquina virtual Xen . . . . .	162
12.2. Gráficas del servidor. Contribuciones de los distintos evaluadores . . . . .	163
12.3. Gráficas del servidor. Personas que han realizado tests . . . . .	164
12.4. Captura de pantalla de Chronojump realizando consulta Query al servidor . . . . .	165
12.5. Gráfica correspondiente a una tabla de percentiles del salto CMJ en mujeres . . . . .	166
13.1. Interacción entre herramientas y personas en Chronojump-Boscosystem . . . . .	170
13.2. Logotipo de Chronojump-Boscosystem . . . . .	171
15.1. Cálculo del ángulo de flexión de la rodilla en tres dimensiones . . . . .	174
A.1. Chronopic 2. Esquema eléctrico . . . . .	209
A.2. Chronopic 3. Esquemático . . . . .	210
A.3. Chronopic 3 PCB. Cara de los componentes . . . . .	211
A.4. Chronopic 3 PCB. Cara inferior . . . . .	212
A.5. Chronopic 3. Serigrafías . . . . .	213
A.6. Diagrama de flujo del firmware de Chronopic . . . . .	214
A.7. Diagrama de flujo de las interrupciones del firmware de Chronopic . . . . .	215
A.8. Diagrama de la base de datos de Chronojump . . . . .	216
A.9. Diagrama de clases de Chronojump relacionadas con Chronopic . . . . .	217
A.10.Diagrama de clases de Chronojump relacionadas con el manejo de tests . . . . .	217
A.11.Diagrama de clases de Chronojump relacionadas con la ejecución de tests . . . . .	218
A.12.Diagrama de clases de Chronojump relacionadas con el participante . . . . .	218
A.13.Diagrama de clases de Chronojump relacionadas con la sesión y el servidor . . . . .	219
A.14.Diagrama de clases de Chronojump relacionadas con la base de datos . . . . .	220
A.15.Diagrama de clases de Chronojump relacionadas con las estadísticas . . . . .	221
A.16.Diagrama de clases de Chronojump relacionadas con las vistas de árbol . . . . .	222
A.17.Diagrama de clases de Chronojump relacionadas con otras funcionalidades . . . . .	222
B.1. Gráficas del servidor. Tiempo de vuelo en función del salto realizado . . . . .	223

B.2. Gráficas del servidor. Relación entre tiempo de vuelo, tipo de salto, género y nivel de práctica . . . . .	224
B.3. Gráficas del servidor. Índice de Elasticidad e Índice de Utilización de Brazos . . .	225
B.4. Gráficas del servidor. Tiempo de vuelo en función del peso extra . . . . .	226
B.5. Gráficas del servidor. Tiempo de vuelo en función de la edad . . . . .	227
D.1. Ejemplo de informe de fallos enviado por un usuario . . . . .	231
E.1. Validación de las plataformas de contacto. Consentimiento informado . . . . .	234
E.2. Creación de un modelo de predicción del ángulo de la rodilla. Consentimiento informado . . . . .	235
E.3. Disposición de las plataformas de contacto . . . . .	236



---

# 1. Agradecimientos

En este apartado personal de la tesis se desea agradecer a cada persona que ha colaborado en su idioma.

Catalán

Vull agrair profundament als meus pares per tot el que han fet pels meus estudis. Sense la seva perseverança mai hagués recorregut aquest camí, no hauria escrit aquest text, no hauria conegut a tota aquesta gent bona que m'ha recolzat i a qui ara recordaré. També vull agrair especialment al meu germà i a l'Eva pel seu acompanyament en els moments inicials i finals. Vull destacar també a la Myriam, que ha sabut donar forma a l'allau d'idees disperses que em volten pel cap.

Castellano-catalán

En el área de la informática y el software libre, no puedo olvidar la Red Hat Linux que me pasó Jordi Cebolla y todo lo que vendría después. Sí, ja sé que només em vas donar un CD, però un estímul nou, acompanyat de l'interès que manifestaves davant de qualsevol cosa que incia, és el que li cal a un alumne per despertar. Después de tantos años viajando de acto en acto, no puedo contar la gente que he conocido en el mundo del software libre. Si bien son pocos los que ahora citaré, me invade la nostalgia cuando veo las fotos con tantas caras cuyo nombre he olvidado sin remedio.

Javier Candeira, tú que fuiste el primero en ver mi conexión entre tecnología y malabarismos, "Cada uno tiene sus calles y lo más interesante son las esquinas". Agradezco a gente como Fernando Acero, Jordi Mas, Jordi Murgó, Juantomás, Juan Jesús Ojeda (juanje), Roberto Santos y Roberto Majadas por ser tan buena en su labor como divertida en la medianoche. Al último Roberto, le agradezco también el haberme presentado a José Carlos Marcos (Deckard): "Éste es tu hombre"; desgraciadamente, pasados unos años la luz de Deckard se perdió y no volvió a brillar. Nunca me olvido de Ismael Olea, por estar antes, durante y después; eso sí, no creo que el PC que tienes "a todo trapo" haciendo el updatedb cada noche sea lo mejor para levantarse productivo ;). Gracias a Amaya Rodrigo y Zazu Vega por saltar el precipicio para demostrar que no existe. Gracias a Jesús M. González-Barahona, Álvaro del Castillo y Yonderboy por darme mi primera plataforma. Gracias a los que nunca descansan: Gregorio Robles, Pedro Reina, Rodrigo Moya y toda la gente del GPUL. Gracias a todos los que no se quejan por no estar en la lista entendiendo que es un olvido pasajero.

Castellano-catalán

Agradezco a parte a los que habéis aportado código: César Tapia, Isaac Clerencia -escribe más rápido en C que en español- y a Ricardo Cárdenes. También a Zazu Vega por regalar arte. Agraeixo a en Carles Pina, la Elisabet Bermejo, en Ricardo Galli i en Llorenç Valverde pels seus (sopars-> pel·lícules-> conferències)^n. També al Carles per la donació del servidor Xen. Los cuatro fantásticos para el final: otra vez Deckard, Andoni Morales y las bestias del hardware: Juan González -te dedico: "Hazlo o no lo hagas. No existe el 'Probar'" - y Ricardo Gómez, por desafiar a su propio nombre: "Eagleman".

Gracias a la comunidad de R Hispana, muy especialmente a Patricia García, por prestar su ayuda y energía sin apenas conocerme, y sobre todo a Carlos J. Gil por entregar tantos y tantos "ciclos" al servicio del proyecto.

Catalán

A l'àmbit de les ciències de l'esport, i començant per l'INEFC de Lleida, li agraeixo a la Mercè Mateu la confiança dipositada en mi des de que em va veure caminar pel tatami. Sempre positiva i sumant alumnes. De la mateixa manera, agraeixo a Asumpta Enseñat per la seva capacitat en motivar per l'aprenentatge. A en Juan Carlos Valios i en Manuel Lapuente per donar exemple amb la seva dedicació i eficàcia. A Òscar Ortiga pel seu suport diari en el laboratori de biomecànica. A Cristina López i Juan Valdés per la seva frescura, per dir el que pensen i per facilitar sempre les coses. Als meus companys d'aventures a l'escenari i lluny d'ell: Joel Serrà-sempre estàs al meu costat quan arreglo coses amb brides-, en Xavi Llagostera i en Xavi Molina (Chiqui); agraeixo també a l'Alfonso Beamonte la paciència que tenia sempre des de l'altre costat. Gràcies a la Neus Roca per fer tot més fàcil. Finalment dono gràcies a en Fidel Molina, de la Facultat de Ciències de l'Educació, per tota la seva empenta en aque-ll projecte de tesi, que tot i que era una camí molt maco, no va arribar a veure el seu final.

D'un salt a l'INEFC de Barcelona, agraeixo a en Josep Maria Padullés per estar al costat bo de la força, com explorador, enginyer i amic. A més, li agraeixo el vincle amb tots els atletes que he conegut i que han col·laborat en el projecte, com Adrià Garcia, Aldo Tuya, Betlem Gomila, Carles Minguez, Cosme Vicens, Cristina Barcena, Danielle Pedrotti, Ginés Romero, Joan Segur, Jordi Alsina, Jordi Navarro, Jordi Sanjuan, Jorge García, Helena Olsson, Marta Miró, Nora Slizeviciute, Oriol Martínez, Poli Capo, Sergi Miró, Sergi Nuell, Sergio Fernández, Sergi Royo i Silvia Riba. Gràcies també a Jordi Porta per donar un cop de mà tan aviat com el vaig necessitar i de forma desinteressada. Finalment, gràcies a l'Anna Padullés per la seva dedicació i cura en el detall.

De la FPCEE-Blanquerna començo agraïnt a Josep Cabedo, Jordi Segura, Jaume Bantulà i Jordi Riera per acollir-me en els estudis que s'iniciaven. Gràcies a Bernat Buscà per la seva empenta en tot, i en especial per les diferents aportacions al projecte. Gràcies a Mònica Solana per l'ajuda i per contagiar la passió en les coses que afronta. Estic també en deute amb l'Azahara Fort per ensenyar-me a ser millor científic i per la important tasca de difusió que realitza. Gràcies a Maria Giné i Carme Martin per la seva empatia i recolzament. També a la Carme per tornar-me al tennis de taula. Agraïxo a en Josep Oriol Martínez i en Jesús Fortuño pel seu exemple de dedicació en vers l'alumne. Gràcies especialment a la Marta Moragas, Sacra Morejón i Susana Pérez: a vosaltres us dec tant que no us ho podria tornar. Gràcies un altre cop a la Susana, i també a en Josep Cabedo i en Josep Solà per la seva col·laboració en la tesi. Agraïxo el tacte que caracteritza a l'Anna de Montserrat i el Sixte Abadia. Vull agrair també a la resta de professors de la facultat, perquè sempre hi són, sempre ajuden, sempre s'ofereixen. Gràcies a tots ells tot és més fàcil i agradable. Finalment, gràcies als alumnes per recordar-nos que fa poc érem com ells, i per "donar" les seves cames quan cal.

Gràcies a en Xavier Aguado per mostrar-me com és el seu grup de recerca a la UCLM. El seu grup és per a mi un exemple del treball que desitjo poder transmetre als qui són al meu costat. Gràcies també pel suport documental i afectiu: sembla que al final sí que aconsegueixo tancar la tesi.

A la UVic, Xavi Peña, Carles Romagosa i José Luis López han estat de gran ajuda. Xavi es va involucrar des dels temps de Salta, i, al costat d'en Carles, van aconseguir portar a terme les primeres Jornades Universitàries de Chronojump. Carles, a més, va ser de gran ajuda en la producció de les 200 primeres unitats de Chronopic 3. En quant a José Luis, sempre està disposat a col·laborar i fer equip, el seu treball és de gran envergadura i la seva meta és la més noble. Agraïxo també l'empenta d'en Carles Cladera i en Josep Bauzà. En darrer lloc, vull recordar la il·lusió de Mireia Tapiador per l'estudi i el treball, així com lamentar, en aquest document, la seva pèrdua prematura.

Gràcies als amics que han estat pacients en aquest període. En primer lloc els de La Mora, i per no deixar-me a ningú, dono gràcies a l'Isaac i a tots i totes els que l'envolten. Gràcies també als companys del tennis de taula que han coincidit amb l'època de tesi, em refereixo als qui m'han acollit: a l'Ariel, el Prat, els Lluïssos d'Horta i al Long.

En una ronda internacional y por orden alfabético de país:

Castellano	<p>En Argentina, resulta un placer incluir a Juan Fernando Pardo (juanfer) por todas sus aportaciones: Chronopic 2, pruebas, ideas, ... y a Lisandro Mettler, por estar siempre conectado con el proyecto e iniciar una “franquicia” en su país. En los inicios del proyecto fueron importantes David Marchese, y José Pichardo. No he conocido personalmente a ninguno de los cuatro.</p> <p>Claudia García, desde Guatemala, está promoviendo la instrumentación de bajo coste y en especial Chronojump, así que le dedico esta línea y espero poder continuar ayudando a que sus compatriotas puedan investigar.</p> <p>En México, aunque algunos viviendo en EEUU, comienzo agradeciendo a Miguel de Icaza por dedicar su tiempo a cualquiera y transmitir ilusión y código en partes iguales; pero sobre todo gracias por Rupert! Gracias a Sancha, por su humor y por tu libro: “El arte editorial...”. Gracias también a Federico Mena y Francisco Urquijo por tener paciencia conmigo, no siempre es fácil.</p>
Portugués	<p>Após retorno ao seu paraíso: Brasil, estão Marco Bortoleto e Maria Luisa Belloto, com quem compartilhei tantas aventuras e desaventuras e aprendi muito. Agradeço muito a ajuda no projeto, mas agradeço em especial a paciência que tiveram comigo. Saibam que mesmo vivendo longe de vocês e do circo, continuarão para sempre os meus amigos.</p>
Francés	<p>En France, je remercie les personnes du CETRIL pour le concours annuel du logiciel libre. Le prix reçu fut très important pour le développement de Chronopic 3 et, par conséquent, pour l’expansion du projet. Merci également à Deborah Garzón pour ses traductions.</p>
Filipino	<p>Sa Pillipinas, maraming salamat sa lahat ng bumubuo ng PHASES lalo na kay Jeffrey Pagaduan, ang taong punong puno ng enerhiya at pamantayan na bumubuo ng kaalaman sa pampalakasan gamit ang libreng teknolohiya.</p>
Hindi	<p>भारत से शरद शंकर एवं ओंकार नाथ मिश्र (लोजिक ब्रिक) का आभार व्यक्त करना चाहूँगा, कंप्यूटर भिजन में अपने अनुभव मेरे साथ बांटने के लिए. सारी कोशिशें, ज्ञान एवं प्रोग्राम बाँटने के लिए धन्यवाद. आप दोनों के साथ काम करने का अनुभव भी बहुत सुखद रहा तथा भविष्य में पुनः आप लोगों के साथ कार्य करने की आशा रखता हूँ.</p>



Mandarín

我想要感謝台灣許多人的支持，首先是人超好且樂於助人的蕭志棧先生，有你我才會來到台灣並在這裡生產我們的硬體。再來，我想要感謝李恆儒老師邀請我到他實驗室並讓我待了四個月，他不但提供我一個學習窗口也讓我可以全心發展我的計畫。感謝台灣的國家高速網路與計算中心邀請我到他們的研討會演講，特別要感謝鄧偉華先生的協助，  
 韌體改寫成 C 語言並開發了線性編碼器。我也要感謝在台灣期間幾乎朝夕相處的台灣師範大學生物力學實驗室的老師與研究生們：黃長福老師、石又、李芸慧、張耀庭、梁耿豪、黃鉢登、詹明昇、李尹鑫，你們讓我感受到台灣人的友善與熱情。最後我想要謝謝乒乓媽媽俱樂部的所有人，他們像個溫馨的大家庭一樣給我許多指導與微笑。我想要感謝你們最好的方法就是再次回到台灣了吧！

Inglés

On the United States, I want to highlight TopEndSports for being the first web platform interested on the project.

I want to thank to GNOME-translators teams for their continuous dedication on the translation of software. Ordering them by languages and time of contributions I would like to thank: Arab (ar) : Djihed Afifi; Catalan (ca): Jordi Mas i Hernàndez; Chinese (China) (zh\_CN): Funda Wang, Jessica Ban, Yinghua Wang; Czech (cs): Marek Černocký, Petr Kovar; Danish (da): Joe Hansen; Dzongkha (dz): Yumkee lhamo; Finnish (fi): Ilkka Tuohela; French (fr): Jonathan Ernst, Étienne Malandain, Claude Paroz, Stéphane Raimbault, Laurent Coudeur, Bruno Brouard; Gaelic (gl): Darío Villar Veres; German (de): Mario Blättermann, Christian Kirbach; English (British) (en\_GB): David Lodge, Jen Ockwell, Philip Withnall; Norwegian bokmål (nb): Kjartan Maraas; Occitan (oc): Yannig Marchegay; Portuguese (pt): Artur Martins, Duarte Loreto; Portuguese (Brazil) (pt\_BR): Raphael Higino, Mateus Zenaide, Djavan Fagundes; Slovak (sk): Matej Urbančíč, Andrej Žnidaršič, Matej Dremelj, Martin Srebotnjak; Spanish (es): Francisco Javier F. Serrador, Eduardo Retamales, Jorge González, Daniel Mustieles; Swedish (sv): Daniel Nylander; Vietnamese (vi): Clytie Siddall. Thanks Also to Julian Thomas for his help in other translations.

I am very grateful as well to the free software projects we used to develop all our tools: Autotools, C#, Gettext, Glade-3, Grep, Gstreamer, GTK#, Innosetup, Linux, Mono, R, OpenCV, Sqlite, Vim and VirtualBox OSE; and our documentation: Bibtex, Biblatex, Dia, Gimp, Inkscape, JabRef, LyX and L<sup>A</sup>T<sub>E</sub>X.

Italiano

Infine vorrei ringraziare a Carmelo Bosco tutto quello che ha fatto per la scienza dello sport. Spero che altri appassionati saranno impegnati a sviluppare ogni dei progetti che lui ha iniziato.



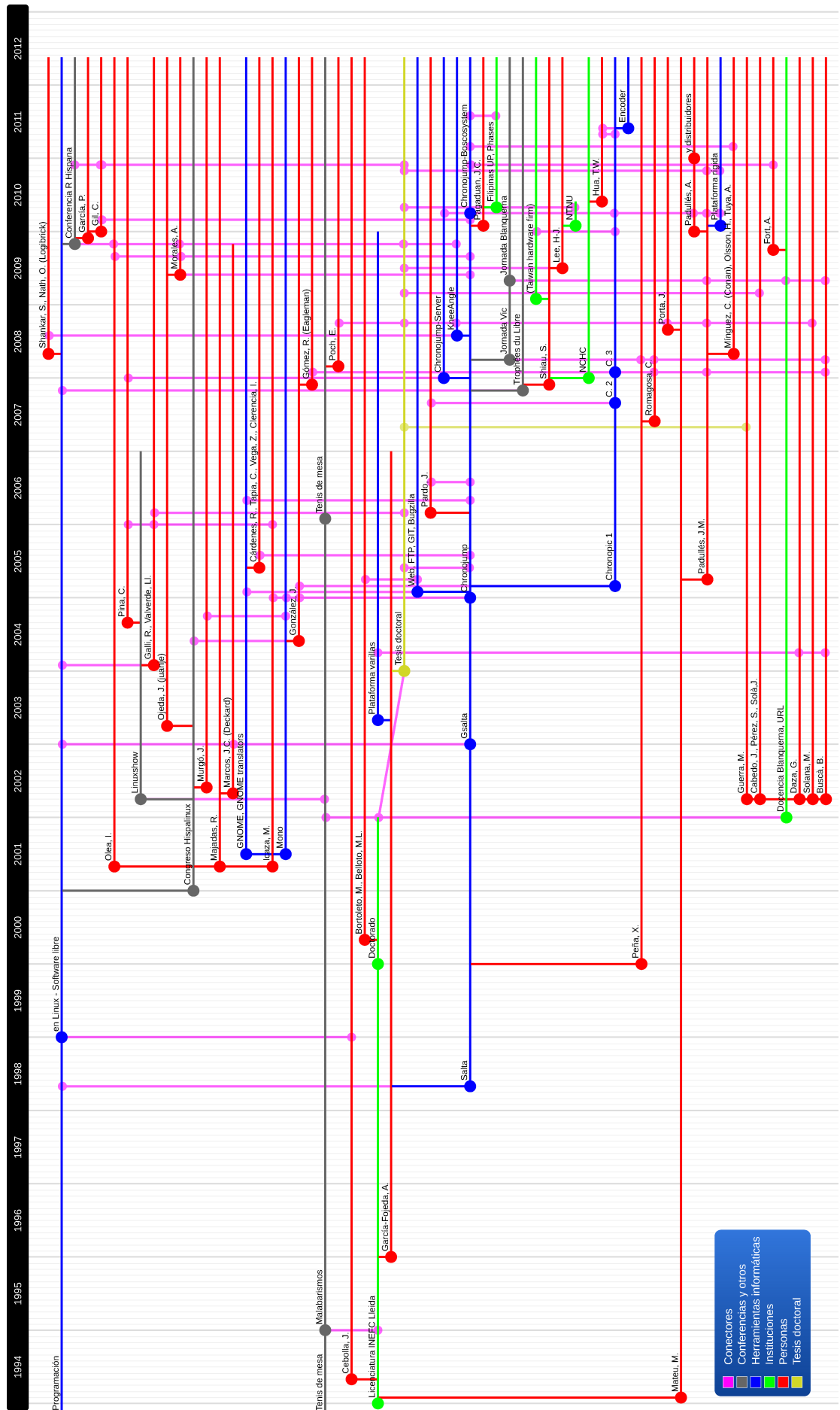


Figura 1.1: Diagrama de aportaciones.



---

## 2. Introducción

### 2.1. Justificación del trabajo

El tiempo es una de las tres magnitudes fundamentales de la mecánica, y es también la magnitud más utilizada para valorar los resultados deportivos. La fiabilidad y la precisión de su medida resultan imprescindibles para el establecimiento de resultados, así como para la valoración de la ejecución de numerosas técnicas deportivas. También en el entrenamiento es necesario medir los tiempos de las distintas fases en que se descompone un gesto deportivo, y disponer de herramientas para la estimación de las capacidades de un deportista.

Todas las herramientas informáticas de medición de tiempos de saltos que se han encontrado son *cajas negras*: se pueden usar pero no se tiene acceso a lo que hay dentro. No es posible que terceras personas verifiquen completamente su fiabilidad ni tampoco que lo puedan adaptar a sus necesidades.

El proyecto que se presenta pretende aportar una herramienta informática para la medición de los tiempos de saltos verticales mediante plataformas de contactos, añadiendo información sobre el ángulo de flexión previa a partir de sincronización con fuente de vídeo y exportando los datos a un servidor público. Para todo ello se utilizará software libre y hardware abierto, permitiendo, por tanto, que cualquiera pueda:

- añadir modificaciones o mejoras, y
- comprobar su precisión y fiabilidad, tanto de los algoritmos de medición como de los métodos de gestión.

Sólo se ha encontrado una herramienta libre que pueda medir los tiempos en el salto (ver Estado actual de la cuestión). Tampoco se han encontrado programas informáticos que permitan exportar los datos de los saltos a un servidor público para la realimentación de otros usuarios, así como la publicación de los mismos al resto de la comunidad científica.

## 2.2. Tema concreto

La presente investigación se fundamenta en diversas áreas de conocimiento, las cuales son: la Biomecánica Deportiva, la Estadística Descriptiva e Inferencial, la Informática, y, por último, la Teoría del Entrenamiento.

Dentro de estas áreas de conocimiento, el proyecto se enfoca en diferentes ramas de cada área. Así, éstas son:

- Biomecánica deportiva
  - Cinemática
  - Cineantropometría
- Estadística
  - Descriptiva
  - Inferencia
- Informática
  - Desarrollo de microprocesadores
  - Visión por computador
  - Programación de software
  - Software libre
- Teoría del entrenamiento

## 2.3. **Objetivos**

Los objetivos del proyecto derivan del propio título de la investigación:

1. Crear, validar y distribuir una herramienta de licencia libre que mida los tiempos de contacto y vuelo en el salto vertical, usando una plataforma de contactos.
2. Desarrollar y validar una herramienta libre para medir el ángulo de flexión de la articulación de la rodilla, previo a un salto CMJ, a partir de una filmación bidimensional y sin el uso de marcadores.
3. Desarrollar una herramienta libre que se integre con las anteriores y permita compartir datos entre evaluadores a fin de construir tablas de percentiles.

Cada uno de estos objetivos se tratará en distintas partes aplicadas de este documento:

Parte II: Estudios de medición del tiempo.

Parte III: Estudios de detección sin marcadores del ángulo de flexión de la articulación de la rodilla.

Parte IV: Estudios de elaboración de tablas de percentiles.





## Parte I

# Fundamentación teórica

La fundamentación teórica la componen dos capítulos:

Capítulo 3: Estado actual de la cuestión.

Capítulo 4: Marco teórico.



---

### 3. Estado actual de la cuestión

La programación hoy en día es una carrera entre los ingenieros de software intentando construir mejores y más eficientes programas a prueba de idiotas y el Universo intentando producir mejores y más grandes idiotas. De momento, el Universo está ganando.

---

Rich Cook

Existen diferentes tecnologías para la obtención de la altura en el salto vertical, se han clasificado de este modo:

- Medición a partir de la distancia.
- Medición a partir del tiempo.
- Medición a partir de la aceleración.
- Medición a partir de filmación.

#### 3.1. Medición a partir de la distancia

La forma más obvia de medir la altura de vuelo de un salto vertical es a partir de la medición de una distancia. Para este propósito, se distinguen cuatro métodos:

1. Sin marca: El saltador trata de alcanzar el punto más alto en referencia a una medición (no se usa ninguna marca). Un observador determina la altura a la que se ha llegado.
2. Contacto: El saltador realiza un contacto en el punto más alto de la trayectoria. Método que, a su vez, se subdivide en:
  - a) Con tiza (Test de Sargent).
  - b) Desplazando un listón.
  - c) Desplazando una ficha magnética.
  - d) Contacto con la mano.

3. Ultrasonido: Desde una altura inalcanzable, se envía una onda de ultrasonido verticalmente hasta la cabeza. Se medirá la distancia mínima entre el emisor y la cabeza a partir del efecto Doppler.
4. Cinta de no retorno (Test de Abalakov): El saltador despliega una cinta de no retorno (Test de Abalakov). Durante el salto se despliega una cinta métrica que el saltador tiene atada a su cintura. Posteriormente se calcula la variación de su longitud.
5. Encoder: Similar a la cinta de no retorno pero usando un encoder lineal.

La tabla 3.1 muestra los sistemas encontrados que permiten la medición del salto a partir de la distancia. Se han incluido dos productos descatalogados por estar presentes en la bibliografía científica o aparecer en publicidad en el ámbito universitario.

### 3.2. Medición a partir del tiempo

Suponiendo que la posición de despegue de un salto sea muy similar a la de recepción, se puede determinar la altura del salto a partir del tiempo de vuelo. Se diferencian los siguientes procedimientos para determinar el tiempo de vuelo:

1. Cronometrando la ausencia de contacto en plataformas de contactos, de barreras de infrarrojos o láser.
2. Cronometrando la ausencia de fuerza contra una plataforma de fuerzas.

La tabla 3.2 muestra los sistemas encontrados que permiten la medición del salto a partir del tiempo.

### 3.3. Medición a partir de la aceleración

Es posible obtener la distancia recorrida a partir de la integración doble de la aceleración. Se diferencian dos tipos de dispositivos:

1. Dispositivos que usan una instrumentación propia.
2. Software que se sirve de un teléfono móvil con acelerómetro.

La tabla 3.3 muestra los sistemas encontrados que permiten la medición del salto a partir de la aceleración.

Tabla 3.1: Herramientas para la medición a partir de la distancia.

Tipo	Producto	Fabricante	País	PVP $\approx$	Distribuidor
Sin marca	The SkyMeter - Vertical Jump Measuring Device	JumpUsa.com	USA	30\$	JumpUsa.com
Contacto-tiza	Test de Sargent (no se encuentra comercializado)				
Contacto-listón	Saltómetro digital de superficie vertical	?	?	Descat.	Psymtec
	Tandem Sport Vertical Challenger Jump Tester	Tandem	USA	273\$	Amazon
	Tandem Sport Wall Mounted Vertical Challenger Jump Tester	Tandem	USA	303\$	Amazon
	Vertec (o Vertec Wall-Mounted Vertec)	Vertec	USA	580\$	Amazon
	Yardstick	Swift	Australia	692€	Innov. Train. Syst.
Contacto-magnético	Magnetic Jump and Reach Board	S&S Worldwide	USA	55\$	S&S Worldwide
	Vertec Wall-Mounted Vertical Jump Tester	Vertec	USA	126\$	Power Systems
Contacto-mano	Electronic Vertical Jump Tester	Gill Athletics	USA	1565\$	PowerMax Equip.
Ultrasonido	Saltómetro Vertisonic	Lafayette Instr.	USA	Descatalogado	
Cinta de no retorno	T.K.K. 5406 Jump MD	Takei Scientific	Japón	41¥	Rakuten
	Vertical Jump Test Mat	Gill Athletics	USA	252\$	PowerMax Equip.
Encoder	Dynabiopsy	Betti, G. y Piga, R.	Italia	Descatalogado	

Tabla 3.2: Herramientas para la medición a partir del tiempo.

Tipo	Producto Fabricante	País	PVP $\approx$	Distribuidor
Plataforma de contactos	Axon Jump			
	Axon	Argentina	440-505\$	Fuerzaypotencia.com
	CVP123			
	Lafayette Instr.	USA	Descatalogado	
	Ergojump Plus (sin plataforma de contactos)			
	Byomedicsystem	España	560€	Byomedicsystem
	JustJump			
	Probiotics	USA	580\$	PowerSystems
	KMS Kinematics Measuring System			
	Fitness Tech.	Australia	6190\$AUD <sup>a</sup>	Fitness Tech.
	NewTest Powertimer + Jump mat			
	NewTest	Finlandia	3500€	Ele-Products Oy
	SaltoBras			
	LABIN. Univ. Sta. Cat.	Brasil	No se comercializa	
	SpeedMat + SpeedLight			
	Swift	Australia	7000€	Innov. Train. Syst.
	TESys Total Evaluation System 400 (módulo contactos)			
	Globus	Italia	1170\$	Globus
	T.K.K. 5414 Jump-DF			
	Takei Scientific	Japón	367¥	Rakuten
Infrarrojos o láser	WinLaborat			
	WinLaborat	Argentina	390\$	WinLaborat
	Ergojump Plus + infrarrojos			
	Byomedicsystem	España	900€	Byomedicsystem
	Laser Optoelectronic Jump Device EMZ-10 (plataforma láser)			
	?	Turquía	?	?
	OptoJump Next (versión 3 cm entre leds/versión 1cm entre leds)			
Plataforma de fuerzas	Microgate	Italia	2450/1840€	Byomedicsystem
	SportJump System Pro (plataforma láser)			
	Desarr. soft. dep.	España	2900€	dsd.es
	Athletjump IBV			
	IBV	España	33000€	IBV
	MuscleLab Ms4010 + plataforma de fuerza (60X80cm / 60X40cm)			
	Ergotest Innov.	Noruega	15800/14300€	Byomedicsystem
	Quattro Jump (9290AD)			
	Kistler	Suiza	14000€	Aries Ingeniería
	TESys Total Evaluation System 1000 (módulo completo)			
	Globus	Italia	3250\$	Globus

<sup>a</sup>El sistema de captación de KMS cuesta 200\$AUD (australianos) y la plataforma de contacto 760\$AUD, pero el mínimo paquete que se comercializa incluye 4 fotocélulas y porta-trolley al precio de 6190\$AUD.

Tabla 3.3: Herramientas para la medición a partir de la aceleración.

Tipo	Producto Fabricante	País	PVP $\approx$	Distribuidor
Usan instrumentación propia	Achillex Jump'n'Run Jump			
	Humotion	Alemania	1844€	Its-sport.de
	Axlo			
	Valmed SA	Italia	Descatalogado	
	FreePower (modalidad que sólo proporciona tiempos)			
	Sensorize	Italia	1000€	Sensorize
	Keimove			
	Vincid	España	Pendiente <sup>a</sup>	Vincid.com
Usa acelerómetro de teléfono	Myotest Sport/Pro			
	Myotest	Suiza	899/1299€	Its-sport.de
	Vertimetric			
	Lafayette Instr.	USA	950\$	Lafayette instr.
	Crazy jump			
	Poppy	Japón	0	App Store
	Liza-Liza Plus			
	Exelio	Italia	20-50\$	App Store

<sup>a</sup>Keimove se comenzará a vender al público a mitades de 2012.

### 3.4. Medición a partir de filmación

Es posible filmar un salto y determinar la altura de vuelo a partir del seguimiento automatizado (tracking) de uno o más puntos contrastado con un sistema de referencia. Se encuentran dos tipos de softwares:

1. Softwares que permiten el análisis tridimensional del movimiento a partir de la transformación lineal directa (DLT).
2. Softwares que no implementan DLT.

La tabla 3.4 muestra algunos de los productos encontrados que permiten la medición del salto a partir del análisis automatizado de una filmación. El listado que aparece en la tabla no es exhaustivo debido a que el número de herramientas aumenta muy deprisa y la filmación es un método poco usado en la determinación de la altura del salto. Algunos de los sistemas que no han sido incluidos son: BTS, Codamotion, eBiom, Inition, Midas, motionanalysis, Ndi, Optitrack, peak performance, Qualisys, SIMI, Sportscad, Vicon.

Tabla 3.4: Herramientas para la medición a partir del análisis automatizado de filmación.

Tipo	Software Fabricante	País	PVP $\approx$
Permiten análisis tridimensional (DLT)	APAS Ariel Performance Analysis System Ariel	USA	?
	Kinescan IBV + 6 cámaras infrarrojas IBV	España	55000€
	Kwon3D Visol	Korea	?
	SkillSpector Video4coach	Dinamarca	Gratuito
No implementan DLT	DartFish DartFish	Suiza	?
	KAVideo Sant Francisco State Univ.	USA	Gratuito (uso educacional)
	Kinovea Joan Charmant	Francia	Gratuito
	Quintic Biomechanics Quintic Consultancy	Reino Unido	2500£



### 3.5. Coste de las herramientas

El precio de venta al público de cada una de las herramientas debe entenderse como informativo. Realizar comparaciones entre las herramientas basándose en su precio puede llevar a conclusiones sesgadas debido a que todas las herramientas expuestas sirven para medir el salto, pero muchas de ellas tienen, además, otras funcionalidades. El precio expuesto corresponde a febrero de 2012.

Algunos de los precios no se encuentran disponibles para la consulta, o se ha recibido información sobre los mismos pero no se permite su divulgación. Ciertos productos se venden a precios distintos en función del país, supuestamente para favorecer la penetración en nuevos mercados. Del mismo modo, se ha observado cómo algunos precios cambiaban a lo largo de los años en función de las ventas, o de la competencia. Además, la modalidad del software que les acompaña también repercute en el precio, hecho que dificulta la exposición de los datos. Como ejemplo se destaca a Keimove (Vincid, n.d.) que se comercializará en 6 versiones de software distintas. Los precios no contemplan tasas ni gastos de envío y no se ha realizado conversión de moneda pues no se garantiza que el coste final en euros corresponda a una mera conversión de moneda.

En general, se estima que el precio de los productos es adecuado para grandes equipos deportivos o laboratorios universitarios, pero difícilmente podrá ser asumible por el entrenador individual, la escuela o el estudiante.

### 3.6. Licencia de las herramientas

La única herramienta libre que se ha encontrado es el software de análisis de vídeo Kinovea (Charmant, n.d.). El resto de softwares y hardwares de las distintas herramientas están licenciados de forma no libre. Este hecho impide la inspección completa de las mismas, así como la corrección o usos derivados. En el caso de Axon Jump (Axon, n.d.), además, se especifica que se pierde la garantía en caso de examinar la plataforma de contactos:

La garantía se invalida si la alfombra de contactos fue abierta o cortada intencionalmente por personal no autorizado. Lo mismo está planteado con el conector y los cables. (Axon, 2004, pág. 76)

### 3.7. Compartición de datos

AthletJump IBV (IBV, n.d.) es la única herramienta encontrada que permite compartir los datos entre el usuario del programa y otra persona a través de la red. Según C. González (comunicación personal, 31 de enero, 2008), esta herramienta favorece la recopilación de los datos por parte del Instituto de Biomecánica de Valencia con la finalidad de usarse en futuros proyectos del IBV, pero no serán expuestos para el uso de los usuarios u otros interesados.



## 4. Marco teórico

Si no estás confundido, no entiendes nada.

Llorenç Valverde

Debido a que el presente documento aborda distintas áreas de conocimiento, se incluye a continuación dos nubes de conceptos para ubicar al lector en los autores referenciados y los conceptos tratados. Estas ilustraciones se refieren únicamente al marco teórico. Para una revisión más completa de los conceptos desarrollados a lo largo de la tesis, se recomienda consultar el Índice alfabético (después de la bibliografía).



Figura 4.1: Autores y conceptos citados en el marco teórico.

Los autores y conceptos más tratados aparecen en mayor tamaño de letra y color rojo. Debido a que el conteo no es exacto, estos gráficos deben evaluarse a título informativo.

La mecánica es la parte de la física que estudia el movimiento de los cuerpos en sí mismo, describiéndolo y refiriéndolo a sus causas (las fuerzas). Así, en ella conviven la *cinemática*, que estudia los movimientos de los cuerpos sin tener en cuenta sus causas; y la *cinética*, que estudia las causas que provocan el movimiento o la ausencia del mismo. Respectivamente *dinámica* y *estática*.

La biomecánica es una disciplina que aplica los principios y métodos de la mecánica para el estudio de los seres vivos. Dentro de la misma, se encuentra la biomecánica del movimiento humano, que comprende el estudio del movimiento del individuo. Según Izquierdo (2008) puede dividirse en los siguientes ámbitos:

- Biomecánica médica, que se ejemplifica en:
  - Biomecánica aplicada a la Traumatología.
  - Biomecánica aplicada a la Rehabilitación.
  - Biomecánica aplicada a la Fisiología.
  - Biomecánica ortopédica.
- Biomecánica ocupacional, también conocida como *ergonomía*.
- Biomecánica de la actividad física y deportiva.

El término *cinesiología* o *kinesiología* se encuentra muy relacionado con la biomecánica. Muchos investigadores los consideran sinónimos aunque se acepta que el primer término es más próximo a la anatomía funcional y la fisiología, mientras que el segundo se fundamenta mayoritariamente en la física. Algunos autores como Ahonen y col. (2001), apostaron por la convivencia de ambos términos; en cambio, otros como Hay (1978), considera que *cinesiología* ha podido perder su utilidad. Según Gutiérrez (1999), la biomecánica deportiva o aplicada a la actividad física ha tenido un desarrollo espectacular en los últimos años, mientras que la cinesiología de la actividad física o el deporte, prácticamente no aparece en la literatura contemporánea. Este aumento de investigación en el ámbito de la biomecánica deportiva provocó que muchos cursos de cinesiología se denominasen cursos de biomecánica (McGinnis, 2005).

Continuando con McGinnis (2005), éste definió la biomecánica del ejercicio y del deporte como el estudio de las fuerzas y sus efectos en los humanos que realizan ejercicio y deporte. El mismo autor distinguió los siguientes objetivos de la biomecánica del ejercicio y del deporte:

1. Mejora de la *Performance* (nivel de ejecución o rendimiento).
  - Mejora de la técnica.
  - Mejora del equipamiento.

- 
- Mejora del entrenamiento.

## 2. Prevención de lesiones y rehabilitación.

- Técnicas para reducir las lesiones.
- Diseño de equipamientos para reducir las lesiones.

El autor manifestó que el segundo objetivo es continuación y parte del primero, pues un atleta no-lesionado ejecutará mejor que otro que padezca lesión.

Los profesores de Educación Física ven la biomecánica como una ciencia importante pero poco útil por la cantidad de información numérica que aporta y la dificultad de interpretación de los mismos. Por otro lado, estos profesores desarrollan un *ojo clínico* a partir de la continua evaluación de la técnica de sus alumnos, una labor que, aunque no use métodos de laboratorio complejos, tiene mucho que ver con la biomecánica (Aguado, 1993).

Siguiendo en el ámbito docente, Kreighbaum y Barthels (1981), señalaron que el educador físico no debe dar clases de física y matemáticas a sus alumnos para enseñar un gesto, pero sí debe aportar un conocimiento sobre los resultados que permita conseguir el objetivo planteado.

Para MacDougall, Wenger y Green (1995) la evaluación del deportista se convierte en un proceso educativo, que aporta unos resultados cuya interpretación es de gran utilidad para el deportista y para el entrenador, al incrementar el nivel de información sobre los factores que intervienen en el deporte y el deportista. La comparativa de los datos que aporta la evaluación permite:

- Verificar el estado de salud y la capacidad funcional general del deportista
- Evaluar las capacidades funcionales específicas del deportista.
- Seleccionar y clasificar al deportista. Seleccionar talentos.
- Orientar al deportista en función de sus puntos fuertes o débiles, hacia las actividades que le pueden ser más favorables.
- Verificar el estado del entrenamiento y verificar los progresos.
- Proponer modificaciones que permitan mejorar el proceso de entrenamiento.
- Fijar objetivos accesibles en las siguientes evaluaciones.
- Predecir resultados deportivos.
- Motivar.

Sale (1995) identificó los objetivos específicos en la evaluación de la fuerza, la potencia y la velocidad:

- Definir la importancia de la fuerza, la velocidad y la potencia en el rendimiento de un determinado deporte o actividad física.
- Valorar las capacidades neuromusculares del deportista.
- Desarrollar el perfil del deportista.
- Controlar el proceso de entrenamiento.
- Hacer un seguimiento de los programas de rehabilitación.

Las características neuromusculares del individuo pueden conocerse a partir de distintas pruebas que ponen de manifiesto la fuerza, velocidad y potencia. Según Zatsiorski (1989), la neuromuscular de un deportista puede realizarse de forma cualitativa o cuantitativa. En la forma cualitativa se observan:

- Articulaciones y segmentos implicados.
- Músculos y grupos musculares que intervienen.
- Tipo de contracción.
- Forma de ejecución.

Mientras que en la valoración cuantitativa se calculan:

- Variables cinemáticas
  - Posición, distancia recorrida y rango articular.
  - Tiempo o duración de la acción.
  - Velocidad media o instantánea (lineal o angular).
  - Número de repeticiones, número de series y su frecuencia.
- Variables cinéticas
  - Tipo y magnitud de la resistencia con que se trabaja.
  - Fuerza desarrollada.
  - Potencia desarrollada.
  - Trabajo.
  - Actividad electromiográfica.

---

Aguado (1993) reforzó el papel de las variables cinemáticas debido a que la descripción del movimiento en base a la posición, el tiempo, la velocidad y la aceleración es común en la técnica y táctica deportivas; así como en la puntuación otorgada por el reglamento y la evaluación del profesor de educación física.

Los tests permiten saber el estado de forma de un individuo en un momento dado. Estas pruebas pueden ocurrir en el campo o en el laboratorio. En el laboratorio se consigue controlar el ambiente con la contrapartida de que el individuo no se encuentra familiarizado con el entorno ni ejecuta las pruebas en forma real a las circunstancias de la actividad. Por ejemplo, el análisis de un swing de golf en un laboratorio permite el uso de un instrumental que difícilmente se podrá llevar al campo; pero dicho material, así como el laboratorio en sí, pueden intervenir en la ejecución del gesto aunque se haya producido familiarización con el laboratorio y su instrumental. Por otro lado, la medición en el campo es ideal por su similitud con el espacio en que se produce la competición, pero determinado instrumental es muy difícil de trasladar al lugar de acción.

McGinnis (2005) dividió la tecnología requerida para el análisis de variables cinemáticas en cuatro tipos:

- Dispositivos para la medición del tiempo.
- Sistemas para la medida de la velocidad.
- Sistemas de filmación.
- Acelerómetros.

Estos dispositivos generalmente pueden llevarse al campo y muchos de ellos no son invasivos, de manera que incluso pueden usarse en competición bajo la obtención de permiso del comité deportivo.

Padullés (2011) distinguió los siguientes tipos de test de campo para valorar la fuerza y la potencia:

- Máxima carga levantada. El levantamiento de una masa conocida es una de las formas más extendidas de evaluar la fuerza. Se puede conocer la fuerza y la potencia a partir de:
  - Máxima carga levantada en una repetición.
  - Número máximo de repeticiones posibles con una carga sub-máxima
  - Tiempo en ejecutar uno o varios movimientos con una carga y un desplazamiento conocidos.
- Determinación de la potencia mecánica mediante pesas y cronómetro. En estos tests se busca movilizar una masa lo más rápido posible.
  - Test de w5'' – Tous y Moras (Tous y Moras, 1999; Tous y col., 2000).

- Test de 1/2 squat de Piasenta.
- Saltos horizontales. Entre los que destacan:
  - Salto de longitud a pies juntos desde una posición estática.
  - Salto horizontal sin utilización de los brazos.
  - Salto horizontal con una sola pierna.
  - Multisaltos.
- Saltos verticales. Usados tradicionalmente en la evaluación de la educación física.
  - Test de Sargent (Sargent, 1921).
  - Test de Abalakov (1938).
  - Saltos sobre obstáculos.
  - Cálculo de la potencia mecánica a partir de un test de salto vertical.
  - Test de Bosco (Bosco, Luhtanen y Komi, 1983).
- Tests de lanzamientos.
  - Lanzamiento de balón medicinal por encima de la cabeza.
  - Lanzamientos con dos manos hacia atrás.
  - Lanzamientos con una mano.
  - Diseño de curvas de fuerza.
- Test de Potencia Anaeróbica
  - Test de Margaria (Margaria, Aghemo y Rovelli, 1966).
  - Test de Margaria – Kalamen (Kalamen, 1968).

El Test de Bosco está constituido por un conjunto de tests en los que se registra el tiempo de vuelo y, en ocasiones, el tiempo de contacto. Esta batería consta de diversas pruebas estandarizadas:

- Salto Squat (SJ), se parte desde parado, con flexión de rodillas de 90 grados, el tronco recto y las manos en la cintura, y se ejecuta un salto vertical sin ayudarse ni con los brazos, ni con el tronco.
- Salto Squat con cargas variables, como el anterior pero colocando cargas en el cuerpo. También conocido como (LJ) Load Jump.



- 
- Salto en contramovimiento (CMJ), similar al SJ pero en bipedestación y realizando un contramovimiento hasta llegar a los 90 grados de flexión de las rodillas.
  - Salto en caída (DJ), similar al CMJ pero dejándose caer desde una altura determinada.
  - Salto reactivo (RJ), saltos DJ continuos, repetidos durante un tiempo determinado. Este salto se puede ejecutar con saltos CMJ continuos o con saltos DJ con una ligera flexión de la rodilla.

Según Padullés (2011) los resultados obtenidos resultan de gran ayuda en el entrenamiento, ya que a partir de ellos se puede:

- Diseñar la curva carga-altura de salto.
- Diseñar la curva fuerza-velocidad.
- Seleccionar las cargas óptimas de trabajo.
- Seleccionar la altura óptima de salto en DJ.
- Determinar el número de repeticiones ideal.
- Crear tests específicos a cada especialidad.

Diversos autores elaboraron índices a partir de los saltos del Test de Bosco:

- Bosco (1987) describió un método que permite estimar indirectamente el porcentaje de fibras lentas y rápidas en el músculo vasto externo, sin recurrir a la biopsia muscular.
- Bosco (1986) comparó la realización de un salto SJ con un peso extra igual al peso corporal, respecto a otro salto SJ sin peso extra. *Indice de Bosco* =  $SJ_{pc}/SJ_0$ .
- Bosco y col. (1983) proporcionó un índice de elasticidad (IE) a partir de la comparación entre un salto SJ y un salto CMJ. En el segundo salto se produce el Ciclo de Estiramiento Acortamiento (CEA) que provoca una mejora en el rendimiento, aunque no existe un consenso en los mecanismos involucrados en dicha mejora (Wilson y Flanagan, 2008).  $IE = 100 * (CMJ - SJ)/SJ$ .
- Bobbert (1990) relacionó los saltos con contramovimiento: DJ y CMJ proponiendo el *Indice de Reactividad* =  $100 * (DJ - CMJ)/CMJ$ . El salto DJ tiene un preestiramiento rápido y brusco, mientras que el preestiramiento del CMJ se produce a velocidad moderada. Schmidtblacher (1992) dividió el CEA producido en el preestiramiento de un salto en dos tipos: rápido (duración inferior a 0,25 segundos) y lento (duración superior a 0,25 segundos). En el salto DJ se produce un CEA rápido y efectivo, mientras que el salto CMJ es de CEA lento (Komi y Gollhofer, 1999).

- Vittori, 1988 (citado en Padullés, 2011) definió el Índice de Utilización de Brazos (IUB) como relación entre los saltos CMJ y Abalakov (CMJ con la ayuda de los brazos en el impulso).  $IUB = 100 * (ABK - CMJ)/CMJ$ .
- Portolés (1994) cuantificó el porcentaje de altura respecto del salto máximo en contramovimiento que se puede mantener durante una serie de saltos repetidos; lo denominó Índice de resistencia a la fuerza rápida:  $IRFR = 100 * RJ/CMJ$ .

Para la correcta interpretación de los índices obtenidos, la ejecución de los saltos debe responder a la técnica descrita. Algunos individuos tienen dificultades en no realizar contramovimiento en el salto SJ, de modo que si éste aparece, la comparación con el CMJ para obtener el índice de elasticidad será fraudulenta, pues ambos saltos dispondrán de componente elástico. Bosco (1994b) recomendó que el evaluador presione con una mano hacia abajo, contra el hombro del atleta y la suelte repentinamente, momento en que el deportista debe saltar rápidamente, mientras que López y col. (1999) propuso el uso de un sistema de referencia formado por dos barras verticales y una goma elástica. J.M. Padullés (comunicación personal, 27 de marzo, 2009) sugirió el uso del salto Rocket como variante del salto SJ, pues el Rocket Jump se inicia en máxima flexión impidiendo la posibilidad de contramovimiento. El mismo autor advirtió que los atletas con más potencia relativa obtienen mayores valores en el salto Rocket que en el SJ.

El ángulo de flexión en el contramovimiento es materia de discusión entre los distintos investigadores. Tihanyi, Apor y Fekete (1982); Jaric, Ristanovic y Corcos (1989) demostraron que la producción de la fuerza varía con el ángulo de la articulación de la rodilla. Bosco advirtió que:

Conviene prestar especial atención al control de los resultados del CMJ ya que su fiabilidad se puede poner en entredicho si no se respeta el ángulo de flexión de la rodilla. Podemos observar que los sujetos no muy fuertes pero veloces prefieren no doblar demasiado las rodillas durante el preestiramiento (Bosco, 1978), esta flexión debería llegar a los 90 grados y difícilmente se obtiene con sujetos que no se hayan ejercitado en numerosas pruebas preliminares de aprendizaje. (Bosco, 1994b)

Por otro lado, la tabla 4.1 muestra la comparación de los distintos ángulos de flexión que se encuentran en la bibliografía.

Ingen Schenau, Bobbert y Rozendal (1987); Ingen Schenau y col. (1992); Bobbert y col. (1996) consideraron que los ángulos de flexión de la rodilla para la ejecución del salto son dependientes del patrón de movimiento individual y, por tanto, su restricción afecta a la coordinación y la efectividad del saltador.

La mayoría de entrenadores que usan el Test de Bosco para evaluar deportistas, *miden* el ángulo máximo de flexión en el contramovimiento a simple vista, siendo ésta una forma de medición muy poco fiable.

Tabla 4.1: Ángulos en las flexiones previas al salto.

Autores	SJ	CMJ	Abalakov	CDJ	QDJ
Bobbert y col., 1987				¿?	¿?
Eloranta, 1997	90			≈39	≈33
Komi y Bosco, 1978	90	90			41,1
Newton y col., 1999	110	110		45	
Vittori, 1990	libre	libre	libre		
Young y col., 1999	90	libre	libre	41,5	35,7

Los ángulos se expresan en grados. CDJ corresponde a Salto en caída con contramovimiento (Countermovement Drop Jump), variación del Salto en caída en la que se permite una considerable amplitud de movimiento en la fase de flexo-extensión. QDJ corresponde a Salto en caída rápido (Quick Drop Jump), variación del Salto en caída en la que se busca el mínimo tiempo de contacto posible. (Adaptado de Marina, 2003).

Como consideraciones finales en la ejecución, se discuten: el calentamiento, la técnica y la motivación. Las condiciones de calentamiento han de ser siempre las mismas, ya que influyen notablemente en el resultado del salto (Esper, 2000). En la pasada década, se ha discutido sobre el papel de los estiramientos previos a los ejercicios de fuerza y de potencia. Knudson (1999) puso en duda la eficacia de los estiramientos pasivos previos al ejercicio, y Knudson y col. (2001) observaron una pequeña disminución en el rendimiento del salto vertical cuando previamente se habían realizado dichos estiramientos. Yamaguchi e Ishii (2005) descubrieron que los individuos con mayor potencia en los extensores de piernas eran los que sufrían mayor disminución del rendimiento del salto ejecutado después de realizar estiramientos estáticos de 30 segundos. Los mismos autores observaron que los estiramientos dinámicos previos del salto sí que proporcionaban un mayor rendimiento. Por otro lado, Church y col. (2001) encontraron que los estiramientos tipo PNF (Facilitación Neuromuscular Propioceptiva) previos al salto, disminuían el rendimiento, mientras que Woolstenhulme y col. (2006) no observaron variaciones entre los grupos que no realizaban estiramientos y los que los realizaban, ya fueran de tipo dinámico o estático. McMillian y col. (2006) concluyeron que, en caso de que se requiriera realizar estiramientos estáticos en tareas de un alto grado de flexibilidad, potencia y agilidad, éstos deberían ir seguidos de movimientos progresivos que fueran parecidos a la actividad a desarrollar cuidando que no apareciera la fatiga.

En cuanto a la técnica de la ejecución y su entrenamiento previo, Aguado, Izquierdo y González (1999) consideran que los tests se deben entrenar previamente para minimizar los factores ligados a la técnica de ejecución. Así, las variaciones en los resultados se atribuyen a modificaciones en la fuerza. Según J.M. Padullés (comunicación personal, 8 de octubre, 2010), algunos atletas -aún conociendo la técnica de saltos- experimentan desequilibrios al realizar un salto máximo,

provocando que la caída no se produzca en el mismo lugar del despegue o que ambos pies no alcancen el suelo al mismo tiempo. Estos atletas pueden corregir el desequilibrio en un salto submáximo, aunque en este caso el valor registrado no aporta información útil. En la figura 4.2 se muestra un ejemplo de estos desequilibrios. Finalmente, García-Fogeda (2001) demostró que el uso de referencias externas como elemento de motivación provoca una mejora en el salto vertical.



Figura 4.2: Desequilibrio en la recepción de un salto Abalakov. Ejecución por parte de una medallista española en 100 metros, acostumbrada a realizar tests de saltos.

El resultado del salto se obtiene en forma de tiempo de vuelo (TV) y, en función del tipo de salto, de tiempo de contacto (TC). Para que la medición del tiempo sea objetiva es necesario que sea tomada por un instrumento electrónico y que el resultado no dependa de la habilidad o la experiencia del evaluador. Esto es aún más determinante cuando los tiempos a medir son muy cortos, como en un salto donde aparecen tiempos de vuelo de medio segundo y tiempos de contacto de tres décimas de segundo.

El error accidental, a diferencia del sistemático, es variable y depende de la persona (Aguado, 1993). Para minimizarlo es conveniente que el instrumento electrónico participe en la fase de medición, de registro y de vinculación de los datos a su contexto, ya que una incorrecta gestión de la información puede provocar retrasos excesivos, pérdida de datos e incluso la imposibilidad de recuperarlos (Storey, 2000).

En cuanto a la fiabilidad, Izquierdo (2008, pág. 62) afirmó lo siguiente:

La fiabilidad de un test refleja la confianza que se puede conceder a los datos en términos absolutos o relativos y se expresa como la consistencia, estabilidad y precisión con la que una medida valora o evalúa una característica dada.

De este modo, cuando un test administrado a un individuo ofrece el mismo resultado varias veces en las mismas condiciones, se entiende que este test es fiable. En el caso de los tests de saltos, se debe considerar la dificultad de reproducir el mismo resultado por el ejecutante, así como el error de medida del instrumento utilizado.

La tabla 4.2 recoge datos de correlaciones para diversos saltos en situaciones de test-retest y de comparación entre dos días distintos. Los datos ponen de manifiesto la dificultad que supone la repetición de una misma ejecución en un test de salto. El aumento de correlación encontrada en el test-retest del segundo día es notable, y la correlación entre ambos días sin dejar de ser satisfactoria, decae considerablemente. Según sus autores, el aumento de correlación en el segundo día es atribuible al aprendizaje aunque se haya realizado un entrenamiento previo de los tests para evitar este hecho. Respecto a la correlación entre los dos días, afirmaron que algunos factores como el estado anímico del sujeto y las horas desde la última comida no se pueden controlar plenamente.

Tabla 4.2: Reproducibilidad de tests de salto.

Correlación	SJ	CMJ	SLJ
Test-Retest día 1**	0,89	0,86	0,97
Test-Retest día 1**	0,98	0,99	0,99
Día 1 - día 2*	0,76	0,68	0,75

SLJ corresponde al Salto horizontal a pies juntos. \* $p < 0,05$  \*\* $p < 0,001$ . (Adaptado de López y col., 1999).

Martín y col. (2001) aún encontrar un empeoramiento significativo de los resultados de los saltos en contramovimiento y Abalakov en una segunda sesión respecto a la primera. Atribuyen el empeoramiento a la falta de motivación o pérdida de concentración y corroboran el papel del aprendizaje constatando que se encuentra mayor consistencia entre los saltos en la segunda sesión.

Curiosamente, los saltos y Abalakov experimentaron un empeoramiento significativo en la segunda sesión. Los autores lo atribuyen a una falta de motivación o pérdida de concentración.

El último aspecto a comentar relativo a la calidad de los tests es la validez. Ésta refleja el grado en que un test mide con precisión lo que debe medir. Según Izquierdo (2008), la validez exige que lo que se mide sea relevante para el objetivo que se persigue y que sea medido con fiabilidad, de modo que un test válido ha de ser necesariamente fiable. Por este motivo se usará durante la tesis el término *validar* como un superconjunto de *validar* y *fiabilizar*.

El instrumento electrónico que fue concebido para la medición del Test de Bosco se denomina Ergojump Boscosystem. Este sistema está compuesto por una plataforma de contactos conectada

a un microcontrolador. La plataforma de contactos (ver figura 4.3) está constituida por varios pares de varillas metálicas, paralelos y equidistantes. Cada par de varillas funciona como un contacto, uniéndose cuando el individuo está sobre ellas y volviendo a su posición inicial en caso contrario. El circuito eléctrico equivalente muestra los pulsadores conectados en paralelo, que a su vez son equivalentes a un único pulsador, que cierra el circuito cuando el sujeto está encima de la plataforma. Buscà, De Blas y Daza (2004) publicaron las instrucciones para la construcción de una plataforma del mismo tipo.

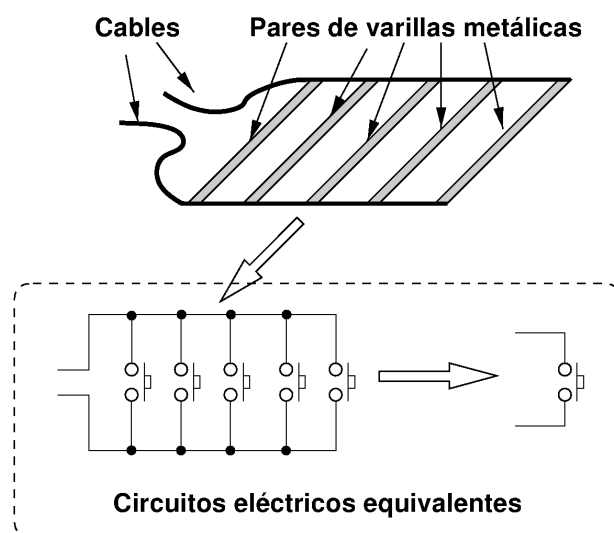


Figura 4.3: Esquema de la plataforma de contactos Ergojump Boscosystem y sus circuitos eléctricos equivalentes. (De Blas y González-Gómez, 2005).

En la comercialización de estas plataformas la distancia entre los pares de varillas ha variado, llegándose a distribuir plataformas con distancias entre varillas de hasta 10 cm. Esta separación es demasiado grande y provoca errores de fiabilidad cuando la punta del pie se encuentra en el espacio entre varillas. En el despegue, el error se manifiesta en una detección prematura del inicio del salto, mientras que en la caída conlleva una detección tardía. Por este motivo no se recomienda el uso de plataformas de varillas con una separación entre las mismas superior a 5 cm.

Además de las plataformas de contactos, existen algunos modelos de plataformas de rayos infrarrojos que tienen como ventajas: la posibilidad de realizar tests en la propia pista de atletismo incluso con zapatillas de clavos, la disminución del temor a resbalar en la caída, y el aumento de la durabilidad por ausencia de deterioro a causa de la humedad (López, Padullés y Tous, 1999).

En las plataformas de contactos e infrarrojos se calcula la altura del salto asumiendo que el centro de gravedad del saltador se encuentra en el punto más alto justo en la mitad de la fase de vuelo. Esto es correcto sólo cuando el centro de gravedad en el momento de despegue está a la misma altura que en el momento del impacto con el suelo. Kibele (1998) analizó las diferencias entre la medida de la altura del salto a partir del tiempo de vuelo y la misma medida a partir de la integración numérica de la fuerza registrada en una plataforma de fuerzas. El autor observó que la diferencia de altura del centro de gravedad en el despegue y la recepción es significativa y

que conlleva un aumento de la altura del salto en la medición realizada a partir del tiempo de vuelo; pero que este aumento no es estadísticamente significativo. Lara (2007) encontró que el centro de gravedad de los sujetos en la recepción se encuentra  $4,28 \pm 2,70$  cm más abajo de la posición de despegue (ver parte izquierda de figura 4.4) lo que conlleva una sobreestimación de la altura lograda en 2,1 cm. Por otro lado, J.M. Padullés (comunicación personal, 8 de octubre, 2010), propuso la realización de un QDJ o salto en caída rápida justo después de la caída. Dicho salto no se registra y su finalidad es la de promover una posición de llegada en el primer salto que facilite el segundo. Según el autor, la intención de rebotar provoca una preactivación que conlleva una extensión de las piernas y una flexión plantar del pie, de modo que la altura del centro de gravedad en la caída del primer salto resulta muy similar a la del despegue anterior. Un estudio piloto paralelo al desarrollo de esta tesis determinó, de forma cualitativa, que la variación en el protocolo propuesta por J.M. Padullés, rebaja el error en la medida de la altura del salto a partir del tiempo de vuelo debido a que se consigue una posición de llegada muy similar a la de despegue (ver parte derecha de figura 4.4).

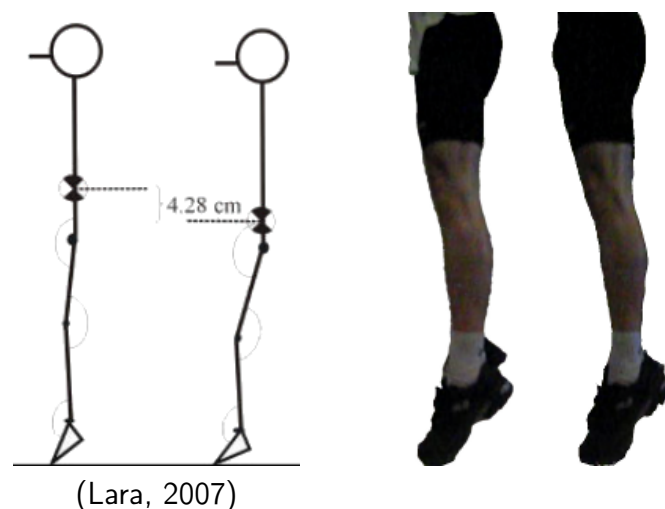


Figura 4.4: Diferencia en el despegue y la caída en el salto CMJ.

La imagen de la izquierda muestra la diferencia media a partir de un conjunto de ejecuciones. La imagen de la derecha corresponde a una prueba piloto que se llevó a cabo una única ejecución realizando rebote después de la caída.

Una alternativa a las plataformas de contactos son las plataformas de fuerzas. Éstas facilitan el valor de esta magnitud física durante todo el movimiento y sobre el eje vertical (uniaxial) o en los tres ejes (triaxial). Las plataformas de fuerzas diseñadas para el salto vertical muestran la información de tiempo de vuelo y/o contacto en el momento de terminar el salto, mientras que en las plataformas de fuerzas de uso genérico es necesario extraer la información a partir de una gráfica como la que muestra la figura 4.5. Aunque el estudio de las gráficas de fuerza y la obtención del tiempo de vuelo son contenidos muy interesantes para la docencia (Linthorne, 2001), la demora requerida en la obtención de los tiempos en una plataforma de fuerzas, así como el elevado precio, su difícil portabilidad y la necesidad de calibración, son las principales causas de la popularidad de las plataformas de contactos e infrarrojos.

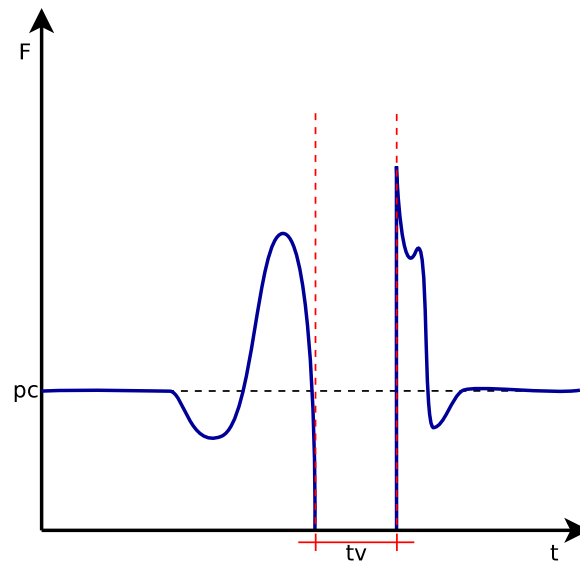


Figura 4.5: Obtención del tiempo de vuelo a partir de plataforma de fuerzas.  $F$ : Fuerza;  $pc$ : peso corporal;  $t$ : tiempo;  $tv$ : tiempo de vuelo.

Los acelerómetros son otra alternativa a las plataformas de contactos. Thompson y Bemben (1999) validó el uso de acelerómetros para la medición de la potencia en el tren superior. En cuanto al tren inferior, en los últimos años han aparecido multitud de sistemas que miden la aceleración en el salto ayudándose, en ocasiones, de giróscopos. Los acelerómetros calculan el espacio recorrido (altura de vuelo) a partir de una integración doble, operación que conlleva un error considerable y debe ser corregido a partir de algoritmos de filtrado. Aunque la validez de estos sistemas ha sido demostrada (Casartelli, Müller y Maffiuletti, 2010; Picerno, Camomilla y Capranica, 2011), el uso de los acelerómetros aún es incómodo debido a que obliga al ejecutante a estar muy quieto antes del movimiento, para que se pueda identificar correctamente el inicio del mismo.

El complemento de la plataforma de contactos es el microcontrolador, que se encarga de alimentarla y captar sus cambios de estado. Concretamente, un microcontrolador (también llamado *empotrado*) es un chip o circuito integrado que contiene las tres unidades funcionales de un ordenador: Unidad Central de Proceso (CPU), memoria y periféricos. La CPU se encarga de procesar las instrucciones que conforman su programa, mientras que la memoria almacena temporalmente los datos que requiere la CPU y los periféricos permitirán realizar operaciones de entrada y salida de información. El microcontrolador se conecta a una tarjeta de circuito impreso que le permite comunicarse con otros circuitos integrados conectados a (generalmente *montados en*) la misma placa. El programa que desempeña el microcontrolador se denomina Firmware, considerado software y hardware a la vez, pues tiene una parte de código y está integrado en un dispositivo. El cambio de estado de la plataforma es detectado por el microcontrolador gracias al cambio de estado del circuito eléctrico (abierto  $\leftrightarrow$  cerrado). Debido a las múltiples conexiones de la plataforma, este cambio no es limpio y el firmware deberá encargarse de eliminar los pulsos es-



púeos o rebotes de la electrónica que puedan llegar pocos milisegundos después del cambio de estado. La figura 4.6 muestra algunas aplicaciones de los microcontroladores.

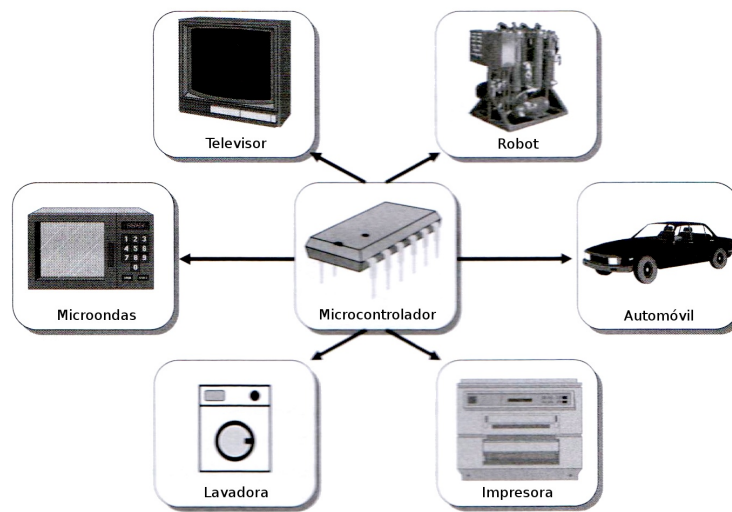


Figura 4.6: Aplicaciones de los microcontroladores. (Pérez, 2007).

De la extensa gama de microcontroladores, es necesario destacar los que comercializa la empresa Microchip -denominados PIC- por ser muy usados. Según Pérez (2007), estos microcontroladores deben su popularidad a:

- Su buena relación precio/prestaciones.
- La facilidad de desarrollo de aplicaciones basadas en ellos.
- Su facilidad de reprogramación.
- La existencia de extensa documentación y diversas herramientas de programación de libre distribución.
- Su disponibilidad como circuitos normalizados y la existencia de numerosos desarrolladores de sistemas que los utilizan.

A diferencia de los microcontroladores, los ordenadores personales modernos son manejados por sistemas operativos multitarea, lo que significa que diversos procesos se ejecutan al mismo tiempo. En realidad se asignan muy pequeñas cantidades de tiempo a cada tarea, y se van alternando las tareas de manera que parece que se están haciendo todas a la vez. El cambio de tareas provoca que una tarea no se inicie al instante, sino que necesite esperar un cierto tiempo, denominado latencia, la cual no es constante.

Existen tres tipos de implementaciones de multitarea: concurrente, preferente y real. La multitarea concurrente, es la tarea que se está ejecutando, la que decide cuándo debe dejar paso a otras, y entonces será el sistema operativo el que asigne la nueva tarea. Este tipo de multitarea está en

desuso debido a que cuando una tarea se queda bloqueada puede no dar paso a la siguiente dejando todo el sistema sin respuesta. En cambio, en la multitarea preferente es el sistema operativo el que marca los cambios de tarea y puede resolver el problema de que una tarea se quede *colgada*. Por último, la multitarea real depende de que el ordenador tenga varios microprocesadores y que cada uno se dedique específicamente a una tarea.

Recientemente han aparecido sistemas operativos en tiempo real que posibilitan el cronometrado fiable por parte de un ordenador personal. Estos sistemas operativos tienen una latencia que tiende a cero en la captación de eventos distintos a los propios del sistema operativo. El sistema operativo en tiempo real más conocido es Real Time Linux (RTLinux) (Yodaiken y col., 1999) aunque también se puede reconfigurar (*parchear* en la jerga informática) un sistema operativo convencional basado en Linux para conseguir la mínima latencia en los procesos que se requiera. Con estas configuraciones es posible cronometrar a una frecuencia de microsegundo ( $10^{-6}$  segundos) (Beal y col., 2003) y controlar multitud de dispositivos distintos (Proctor, 2002). Algunas de las aplicaciones en uso de estos sistemas en el campo de la adquisición de datos son: radioaltímetro de escáner (Wright y Walsh, 1999), receptor de GPS (Ledvina, Mota y Kintner, 2000), radiómetro solar de flujo de espectro (Clanton, 2002) y cámara telescópica (Beard y col., 2002).

La última pieza de hardware a la que se hace referencia son las cámaras de vídeo. Padullés (2011) distinguió los siguientes tipos actuales de cámaras de vídeo:

- Cámaras de alta resolución. Día a día aparece una nueva cámara que aporta más detalle a la imagen, llegando en la actualidad a una resolución de 22 megapíxeles. Este valor dista ampliamente de las resoluciones de pantalla más populares en los ordenadores domésticos, que apenas alcanzan los 1,3 megapíxeles (W3Schools, 2011).
- Cámaras de visión artificial. Permiten, entre otras características, disparar en el momento en que detecta como óptimo.
- Cámaras inteligentes. Incorporan múltiples procesadores, controles de imagen, un software configurable y puertos de configuración con el exterior.
- Cámaras de alta velocidad. Su característica distintiva es el número de imágenes que registran por segundo.
- Cámaras 3D. Un láser y un software de triangulación les permite reconstruir una escena tridimensional realizando todos los cálculos en la propia cámara enviándolos al ordenador.
- Cámaras térmicas. Muestran la temperatura de alguna parte del cuerpo a partir del registro de la radiación infrarroja emitida.

En biomecánica deportiva, muy a menudo, se usa la imagen para obtener variables cinemáticas. Cuando se trabaja en una única imagen se denomina fotogrametría, mientras que el análisis a

---

partir de vídeo recibe los nombres de cinegrametría y videogrametría. La metodología usada suele implicar el recuento de fotogramas para determinar el tiempo, mientras que el espacio se mide observando la separación entre el objeto que se analiza y otro que se usa como referencia. La precisión en estas medidas depende de la velocidad y la resolución de la cámara, así como de la cantidad de objetos de referencia y de sus posiciones. Hoy en día, gracias a las cámaras de alta velocidad que ha popularizado Casio con su gama Exilim, resulta asequible económicamente la filmación a frecuencias de 150-1200 Hz, muy por encima de los 25 Hz que permitían las cámaras que usaban los entrenadores hace pocos años.

La filmación a alta velocidad tiene como inconveniente el volumen de imágenes a procesar. Cuando las acciones y variables mecánicas a evaluar son pocas, el procesamiento manual es viable. Todo lo contrario, cuando el producto de variables por acciones e imágenes es elevado, entonces es imprescindible un proceso automático o semiautomático. El procesamiento automático de las imágenes es propio del campo de la visión por computador.

Para Bradski y Kaehler (2008, pág. 2), “La visión por computador es la transformación de datos que provienen de una fotografía o un vídeo, en una decisión o una nueva representación”. Por *decisión*, se entiende el diagnóstico que se produce ante una situación esperada. Así por ejemplo, si se buscan personas en una imagen y se detectan tres objetos que se corresponden al modelo de persona, la decisión es que existen tres personas. Por otro lado, convertir una imagen de color a escala de grises o corregir la perspectiva, son ejemplos de nuevas *representaciones*.

Para captar los segmentos y las articulaciones del practicante se usan marcadores de tipo activo o de tipo pasivo. El primero emite una luz a una frecuencia determinada, de modo que el software de procesamiento pueda distinguir entre marcadores. En cambio, los marcadores pasivos reflejan la luz y, debido a que todos brillan del mismo modo, la distinción entre ellos es más compleja. Los marcadores posibilitan la captación de puntos corporales de forma automatizada, pero tienen como inconvenientes la posible interferencia en el gesto a analizar, la problemática de determinar el centro articular a partir de varios marcadores en determinados sujetos, la necesidad de colocación de los mismos por parte de un experto y la posibilidad de perder algún marcador a mitad de la ejecución.

La detección de objetos sin marcadores es mucho más compleja que con ellos. A continuación se citan tres técnicas usadas para este propósito:

**Reconocimiento de gestos:** Consiste en identificar las partes que se mueven de la imagen y en qué dirección lo hacen. Se basa en regiones de interés (partes móviles de la imagen) de un tamaño determinado usando plantillas (Bradski y Davis, 2002; Novák, Ondřej y Ondřej, 2006). Esta técnica es poco adecuada para la biomecánica debido a que se detecta el movimiento de objetos, pero existe poca precisión sobre el mismo.

**Haar Cascades:** Trata de encontrar objetos a partir de multitud de fotografías de negativos (en que los objetos no aparecen) y de positivos (sí aparecen). La cantidad de fotografías

necesarias son entre 5000 y 10000 para los negativos y aproximadamente una cuarta parte para los positivos (Adolf, 2003). Esta técnica es útil para encontrar objetos determinados en fotografías o vídeos en que uno no sabe si habrá sólo alguno de ellos, muchos o ninguno, como sucede en los estudios citológicos (Jiang y col., 2007). Las cascadas de Haar no inciden en la precisión de la detección de los límites de los objetos, aspecto imprescindible en la biomecánica deportiva para poder obtener su centro, y por tanto calcular variables cinemáticas, como la velocidad del objeto entre fotogramas. Por este motivo y porque en biomecánica deportiva se sabe cuantos objetos se desea encontrar, esta técnica no es, tal vez, la más indicada para esta área del conocimiento.

**Segmentación de objetos:** Se pretende separar el fondo respecto a los objetos de primer plano para tratar de reconocer a éstos. Se entiende que el fondo es inmóvil y sólo experimenta variaciones por motivos atmosféricos. La detección de objetos de primer plano acostumbra a ir acompañado de la detección de la piel humana, tarea que se lleva a cabo a partir de la identificación automatizada de su color (Porikli, 2003). Una de las más populares aplicaciones de esta técnica es la vigilancia de un edificio (Chen y col., 2005). La segmentación de objetos se puede dividir en las siguientes técnicas de visión por computador:

**Eliminación del fondo:** Separa los objetos principales del resto.

**Eliminación de sombra:** La selección de un umbral o threshold permite identificar la sombra como parte del fondo, aunque las sombras se muevan es fácil identificarlas por sus características de color (Porikli, 2003).

**Seguimiento de objetos:** Puede realizarse hacia atrás (Back tracking) o hacia adelante (Forward tracking). Hacia atrás se buscan objetos en la imagen actual y se comprueba si corresponden a los de la imagen anterior. Hacia delante se buscan objetos en la presente imagen a partir de los que se encontraron en imágenes anteriores.

El reconocimiento de objetos en cualquier técnica de visión por computador requiere de una iluminación suficiente (para que pueda apreciarse el detalle) y constante (para que se pueda evaluar el movimiento). Tal y como expresaron Trucco y Verri (1998): “Asumiendo que las condiciones de iluminación no cambian, los cambios en la imagen son producidos por el movimiento relativo entre la cámara y la escena”. Conseguir esta iluminación no siempre es sencillo pues las cámaras de alta velocidad tienen el obturador cerrado mucho más tiempo que una cámara convencional, precisando de luz natural adecuada o luz artificial potente.

La detección de objetos en pantalla requiere de un proceso de aceptación y descarte de puntos a partir de un umbral determinado. El correcto ajuste de dicho umbral es determinante para

conseguir el resultado esperado. A lo largo de la tesis se utiliza la denominación en inglés *threshold* como umbral por ser la forma más aceptada y conocida. La figura 4.7 muestra la detección de personas en una cámara de vigilancia con un threshold alto y uno bajo, en las imágenes izquierda y derecha, respectivamente. En el threshold alto se captan pocos puntos pues el umbral de aceptación está alto. En concreto, se captan puntos correspondientes a las dos personas pero no se consigue obtener enteramente a ninguna de ellas. En la imagen de la derecha se observa el resultado del threshold bajo, al poner un umbral más bajo se captan más puntos de modo que se visualiza enteramente a las dos personas, pero se captan también sus sombras y otros puntos. Algunos autores, como Porikli (2003) ajustaron el threshold de forma empírica en función de la iluminación y *ruido* de la escena. Por ruido se entienden las variaciones aleatorias del brillo o color que se observan en la imagen y son atribuibles a la captación de la misma -no existen en la visualización real-.

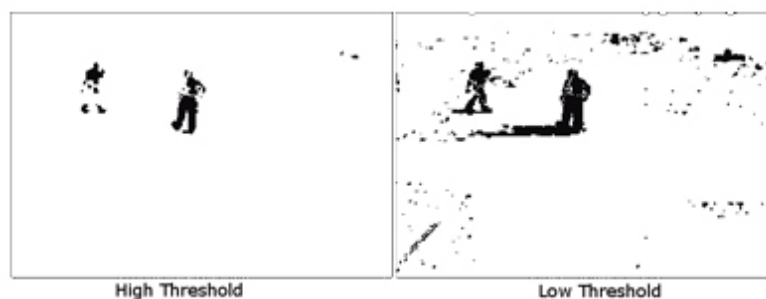


Figura 4.7: Threshold alto y bajo. (Adaptado de Nikhil, 2008).

Las técnicas de visión por computador deben implementarse en un programa informático. Debido a que la programación de las mismas puede ser un proceso costoso, existen conjuntos de funciones ya realizadas, preparadas para poder ser usadas en otros programas informáticos. Estos conjuntos de funciones se denominan *bibliotecas*. Las bibliotecas son la clave para el desarrollo rápido de aplicaciones modernas y de este modo no se pierde el tiempo en reescribir lo que otros han hecho, sino que se reutiliza.

En el ámbito de la visión por computador destaca la biblioteca Open Source Computer Vision (OpenCV). Esta biblioteca se inició en los laboratorios de Intel Research en Estados Unidos y Rusia, pero hoy en día es un proyecto global que se enriquece de colaboraciones de participantes de todo el mundo. La biblioteca contiene más de 500 funciones que abarcan muchas áreas de la visión, entre las que se citan: inspección de productos industriales, imagen médica, seguridad, interfaz de usuario, calibración de cámara, visión estereoscópica y robótica (Bradski y Kaehler, 2008). El grupo de usuarios de OpenCV lo conforman más de 40 mil personas (Bradski, 2011). Por último, debido a que OpenCV funciona en los sistemas operativos Linux, MacOSX y Windows, se puede usar para programar aplicaciones informáticas para estas plataformas. El término *plataforma* se entiende en el contexto informático como conjunto formado por tipo de ordenador y sistema operativo, y no debe confundirse con la plataforma de contactos.

En la creación de un programa informático es imprescindible concretar la licencia del código que se escribe. Un programa informático lo conforman miles de líneas de texto denominadas *código fuente* que se ejecutará en uno o varios sistemas, condicionado bajo la licencia que se haya elegido. La comprensión de las licencias de software requiere de un breve recorrido histórico. En los inicios del software, tal y como relató Fogel (2005), éste no tenía valor por sí mismo, sino que otorgaba un valor al hardware sobre el que funcionaba. Los vendedores de hardware conseguían vender más equipos si el software instalado era más atractivo y los usuarios realizaban copias del software libremente. La falta de estandarización de la industria de hardware llevó a problemas de compatibilidad que hacían que un software sólo funcionase en determinado hardware, hecho que llevó a la aparición de nuevos lenguajes de programación que permitían que un programa escrito para un sistema concreto pudiese ser traducido a otros convirtiendo al software en multiplataforma. La creciente universalidad del software y la poca diferencia que existía entre las máquinas que se comercializaban, llevó a popularizar la venta del software como estrategia de mercado. El valor en auge del software preocupaba a las empresas que lo distribuían de modo que iniciaron restricciones a base de licencias que impedían el acceso, modificación y distribución del código para que la competencia no pudiera beneficiarse de las mejoras realizadas por los usuarios. Más adelante, Stallman y col. (2002) denominarían *privativo* al software que restringe las libertades de los usuarios, y definirían el software libre como el que cumple cuatro libertades:

Libertad 0 para ejecutar el programa con cualquier propósito.

Libertad 1 para estudiarlo y adaptarlo a nuestras necesidades. Esto exige el acceso al código fuente.

Libertad 2 de redistribución, de modo que se nos permita colaborar con vecinos y amigos.

Libertad 3 para mejorar el programa y publicar las mejoras. También exige el código fuente.

El proyecto Debian formalizó una definición de software libre Debian (n.d.) más precisa que -entre otras características- separa el programa original de las adaptaciones que se hayan realizado del mismo. A partir de ésta, la Open Software Initiative enunciaría su licencia Open Source Initiative (n.d.), popularizando el término *Open Source*, que enfatiza la facilidad de acceso tratando de acercarse al ámbito empresarial y ocultando la filosofía de libertad que se encuentra implícita (Stallman, 2007); por tanto, sustituyendo la connotación ética por la pragmática. El término *Open Source* tuvo también sus inicios para evitar la confusión que existía en lengua anglosajona con el término *Free*, que se refiere tanto a libertad como a gratuidad. Para Wayner (2001, pág. 119) “el término *libre* tiene también la problemática de ser uno de los dispositivos de marketing más potentes de todos los tiempos”, hecho que acarrea connotaciones peyorativas. Como ejemplo de la polisemia de *Free* en el ámbito deportivo, según M. Zok (comunicación personal, 16 de febrero, 2012), los productos FreePower y FreeSense deben dicho vocablo a la libertad de movimientos que va asociado a sus productos, pues el practicante no requiere estar encima de una plataforma o en

---

un laboratorio. Volviendo al software libre, en lengua inglesa se optó por heredar una construcción del castellano, *Libre Software*, término que convive con las construcciones *FLOSS* (Free, Libre, y Open Source), y *FOSS* (Free y Open Source). El proyecto de tesis que se presenta continuará con la denominación *Software Libre*, pues en castellano el término se encuentra aceptado y dicha confusión es mínima.

Al contrario, el software privativo se distribuye bajo venta de licencias y el usuario debe pagar por cada copia, o debe acogerse a una serie de cláusulas para impedir la libre distribución del software. Además, el software privativo no permite la consulta ni modificación del código fuente. En ocasiones, algunos programas informáticos se autodenominan “casi-libres” porque el código está disponible pero no es modificable o no es posible redistribuirlo; debido a que estos programas privan de dichas libertades, han sido considerados también como software privativo. En ocasiones, el software privativo es llamado “comercial”, pero esta denominación puede llevar a confusión, ya que el software libre también se puede comercializar.

González-Barahona (2003) destacó el papel que tiene la venta de servicios en el software en general y en el software libre en particular. El autor propuso como modelo de negocio la venta del software como parte de un conjunto que ofrezca lo que el usuario solicite (garantía de mantenimiento o certificado para ciertas tareas), incluir formación sobre su uso o adaptación del programa a las necesidades del cliente que esté dispuesto a pagar por ello. Galli (2004) comentó que los negocios de venta de servicios escalan [crecen] linealmente con la cantidad de profesionales, mientras que la venta de productos puede hacerlo exponencialmente llegando a ganar cantidades insanas de dinero. Galli afirmó que con la venta de servicios de software es posible ganarse la vida razonablemente bien, al igual que lo hace un abogado, arquitecto o peluquero, mientras que pocos programadores están capacitados para desarrollar un producto que pueda sostener un negocio libre de servicios. En la figura 4.8 se observa la evolución temporal de los gastos e ingresos en un negocio de servicios o de venta de productos de software; en el segundo caso, se advierte el problema del tiempo necesario para el retorno de la inversión.

El desarrollo de software libre está muy ligado a la cultura hacker. El término *hacker* apareció en los años 60 para designar a los grupos de expertos entusiastas de la tecnología y “capaces de encontrar soluciones ingeniosas a problemas complejos” (Mas, 2005, pág. 18). Para estos expertos es tan importante el reto en sí, como el compartir los resultados con el resto de la comunidad. La aparición de los primeros virus informáticos a mitad de los años 80 (Solomon, 1990; Wells, 1996) llevó a los medios a llamar hackers a los criminales informáticos, y el término hacker se dividió en hacker y *cracker*, siendo este último el referido a quien rompe la seguridad de un sistema informático (Raymond, 2000).

Himanen (2001) contrapuso la ética protestante en que el trabajo y el dinero son considerados fines últimos, con la ética hacker que promueve el reconocimiento ante iguales, la diversión y la realización personal. Mientras las empresas regidas por la ética protestante consiguen dinero al

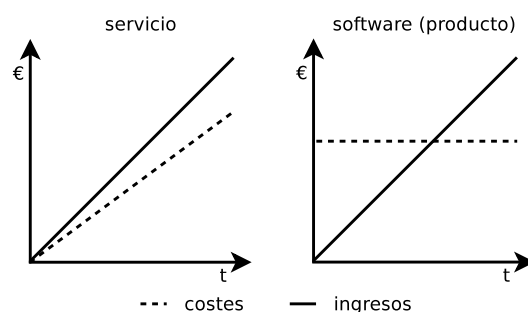


Figura 4.8: Relación costes/ingresos en software como servicio o como producto. (Adaptado de Laprêvotte, 2009).

apropiarse de la información a través de patentes y marcas registradas, los hackers consideran que “la información es un extraordinario bien, y que además es un deber de naturaleza ética compartir su competencia y pericia elaborando software gratuito” (Raymond, 2000).

En el año 1984, Richard Stallman comenzó el proyecto GNU's Not Unix (GNU) con la voluntad de crear un sistema operativo similar al popular sistema UNIX, pero totalmente libre. Seis años más tarde se habían desarrollado múltiples programas libres, pero faltaba el núcleo del sistema operativo (también denominado *kernel*). En el año 1991, un estudiante finlandés de 21 años, llamado Linus Torvalds, anunciaba que iniciaba el desarrollo de un kernel que se uniría a las herramientas GNU para constituirse como sistema operativo completo. Hasta la fecha, se usan dos denominaciones para referirse al sistema operativo resultante: Linux y GNU/Linux. En este documento -sin desmerecer la labor del proyecto GNU- se usará la denominación Linux por ser la más aceptada. Linus Torvalds, creador de Linux, escribió en Himanen (2001):

Para ellos [los hackers], la supervivencia no es lo principal. Pueden subsistir bastante bien a base de donuts y pepsi-colas. Hablando en serio, desde el momento en que se puede tener un ordenador en el escritorio, no es probable que la primera preocupación que nos mueva sea cómo conseguir comer caliente o conservar el techo que nos cubija. [...] Así llega a crearse algo como el sistema Linux. No se trata de hacer mucho dinero. La razón por la que los hackers de Linux hacen algo es porque lo encuentran muy interesante y les gusta compartir eso tan interesante con los demás. De repente, se obtiene entretenimiento del hecho de estar haciendo algo interesante, a la vez que se alcanza una repercusión social.

Hoy en día la mayoría de las aportaciones en Linux provienen de empresas, tal y como publicaron Corbet (2007) y Kidman (2010). El primer autor estimó que el 65 % de las colaboraciones en el código del kernel 2.6.20 -publicado en febrero de 2007- fue contribuido por “personas pagadas



---

por compañías de tecnología”, mientras que el segundo autor -a partir de la monitorización del código entre diciembre de 2008 y enero de 2010- afirmó que el 75 % del código de las versiones 2.6.28 - 2.6.32 fue aportado por “gente pagada para realizarlo”.

El porcentaje de uso de Linux difiere en función del tipo de instalación: ordenador de escritorio, dispositivo portable, servidor y supercomputador. En el escritorio se han creado entornos para que el usuario pueda realizar sus tareas de forma sencilla y rápida. Los entornos de escritorio GNOME (The GNOME Project, n.d.) y KDE (KDE Project, n.d.) destacan por su trayectoria, sus aportaciones y por traducirse constantemente a decenas de idiomas (GNOME Translators, 2012; KDE Translators, 2012). Aún los esfuerzos realizados para llevar Linux al escritorio, se repite en los estudios de mercado que la cuota de mercado de Linux se mantiene en el 1 %, mientras que hay evidencias que muestran que, en realidad, el uso de Linux es mucho mayor (Holwerda, 2009; Martin, 2010). Por otro lado, en los dispositivos portables (o embebidos) destaca el papel de la versión de Linux denominada Android. El 52,5 % de las ventas de smartphones responden a teléfonos inteligentes en las que está instalado este sistema operativo (Gartner, 2011). En cuanto a los servidores, según datos de diciembre de 2011, Linux es el sistema operativo de 28 de las 42 empresas de hosting [hospedaje] más fiables (Netcraft, 2011). Por último, según datos de noviembre de 2011, el 91,4 % de las 500 computadoras más potentes disponibles comercialmente funcionan bajo Linux (Top500.org, 2011).

Los beneficios del software libre son identificados por distintos actores. La compañía de consultoría en tecnologías de información Gbdirect destacó: fiabilidad, estabilidad, auditabilidad, coste, flexibilidad, libertad, soporte y responsabilidad (Gbdirect, n.d.). Wheeler (2007) realizó un profundo análisis sobre los beneficios en términos relativos a la calidad del software (fiabilidad, rendimiento, escalabilidad y seguridad), también en términos relativos al mercado (cuota de mercado, coste total de propiedad, innovación y soporte) y finalmente en otros términos como la libertad, protección contra litigio y flexibilidad. El Departamento de Defensa de Estados Unidos señaló como beneficios el aumento de fiabilidad y seguridad, flexibilidad, libertad y coste (Departamento de Defensa de los Estados Unidos, 2009). En el Open Source Summit de la NASA de 2011, Wheeler (2011) afirmó: “En muchos casos, el software libre tiene el potencial de aumentar la funcionalidad, calidad y flexibilidad, al mismo tiempo que se reduce el coste y el tiempo de desarrollo”. En representación de Google, DiBona (2011) destacó la independencia respecto a compañías de software externas, la capacidad de reparar y mejorar sus servicios, y la proximidad a la ética de Google, como motivos por los que su compañía apuesta por el software libre.

Parker (2000) describió las implicaciones que tiene el software libre en nuestra sociedad:

- Gobierno: La vinculación entre software libre y estándares abiertos permite mejorar la accesibilidad en la interacción de los ciudadanos con el gobierno, ya que este último tiene el deber de ofrecer un acceso al ciudadano independiente del software concreto de procesador de textos, navegador o sistema operativo que tenga instalado.

- Educación: Promoviendo soluciones libres para el aprendizaje de los contenidos en lugar de las que comercializan las empresas de software privativo y que son ofrecidas a los alumnos con licencias de bajo coste para el ámbito universitario.
- Ciencia: Entendiendo el método científico como un proceso de descubrimiento y un proceso de justificación, los resultados científicos deben ser replicables y para ello las herramientas informáticas usadas en los estudios deben ser accesibles en un futuro. El software privativo no suele facilitar el acceso a versiones antiguas, mientras que en el software libre todas las versiones publicadas acostumbran a estar disponibles en los servidores oficiales. Además, el software libre enfatiza la revisión de pares, pues el código fuente de los programas licenciados de este modo es revisado y corregido por la comunidad de usuarios de forma gratuita.
- Negocios: Los negocios agresivos buscan básicamente dos cosas: primero, volverse competitivos por la innovación en un producto, oferta de mejores servicios o rebaja de costes, entre otros y segundo, mantener esa posición de forma duradera para poder generar beneficios. Se entiende que las propiedades del software libre caracterizadas por reducido coste, aumento del control, aumento de la transparencia, estabilidad y flexibilidad, pueden ayudar a una empresa a mantener su ventaja competitiva. Por otro lado, el software libre ha demostrado ser más seguro que el privativo, de modo que es más difícil que se pueda obtener la información sensible de las empresas por parte de su competencia.

Profundizando en la educación y la ciencia, se debe tener en cuenta que el software libre apuesta por la libertad de la información y la cooperación en el desarrollo (Raymond, 1999). Según Himanen (2001), estas características se corresponden con el modelo abierto de la ciencia, (modelo que ha demostrado ser la mejor manera de creación de información), y las obligaciones de su ética, como son:

1. Las fuentes deben ser citadas.
2. La solución no se puede mantener en secreto.

Joyner y Stein (2007) escribió en Open Source Mathematical:

La mejor esperanza que podemos tener para un prueba rigurosa es la inspección abierta y la habilidad de usar todo el código del cuál dependen los resultados. Si el programa es privativo, esto no es posible. Tenemos todo el derecho a ser desconfiados, no sólo por una vaga desconfianza a los ordenadores sino porque incluso los mejores programadores cometen errores regularmente.

El segundo autor de dicho artículo, William Stein, es también autor de SAGE (Stein y col., 2009), un conjunto de programas matemáticos de licencia libre. Joyner y Stein (2007), con la finalidad

---

de reivindicar que no es suficiente con las descripciones o especificaciones del comportamiento de un programa para saber cómo actuará en determinadas situaciones, citaron textualmente el apartado: *¿Por qué normalmente no debes preocuparte de la implementación interna?*, del Tutorial del programa Mathematica:

[...] incluso aunque se conozca la descripción básica del algoritmo usado para un propósito particular, es extremadamente difícil alcanzar una conclusión fiable acerca de cómo la implementación detallada de un algoritmo se comportará en circunstancias particulares. (Wolfram Mathematica, n.d.)

Para que el código fuente pueda convertirse en un programa que pueda ejecutarse, habitualmente deberá usarse un compilador. Éste se encargará de convertir el texto escrito por una persona en algún lenguaje de programación, a un lenguaje que pueda ser ejecutado por la máquina. Thompson (1984) defendió que aunque se pueda ver el código fuente de una aplicación, si el compilador y el entorno de ejecución no son libres, no se puede confiar en un programa. De este modo, el autor manifestó que además del programa requeriremos que el sistema operativo también sea libre. Veinte años más tarde, Spinellis (2003) demostró una proposición paralela: no se puede confiar en un entorno de ejecución si no se puede ver el código fuente de los programas.

A continuación se aborda la situación de la propiedad intelectual en España con el fin de subrayar los aspectos relativos al licenciamiento del software en el contexto de este país. La propiedad intelectual se conforma a partir de una serie de privilegios que se otorgan sobre bienes intangibles con valor económico. Entre los que González-Barahona, Seoane y Robles (2003) destacaron:

**Marcas** que protegen los símbolos. En el caso del software libre son raramente usadas, sólo los grandes proyectos como Debian, GNOME, GNU, o Linux, tienen registrado su nombre.

**Indicaciones geográficas** que protegen las denominaciones de origen.

**Secreto industrial** que permite ocultar información, y en el caso del software privativo le permite ocultar su código fuente.

**Copyright** o derechos de autor, que protegen a los trabajos de la copia no autorizada, impidiendo reproducirlas sin permiso, total o parcialmente, con o sin modificaciones. Estos derechos se dividen en derechos morales y patrimoniales. Los primeros son vitalicios y otorgan al autor el control sobre la divulgación de la obra, el reconocimiento de autoría, el respeto a la integridad de la obra y el derecho a la modificación y retirada. Los segundos dan derecho a la explotación económica y pueden ser cedidos por un contrato denominado licencia. Cabe notar que en España la copia privada es legal en

la música, pintura, cine, pero no en el caso particular del software. La violación del copyright ha dado lugar al término piratería, que según Stallman (2003) es exagerado, pues el término original se refiere a robo con violencia en el mar.

**Patentes** que otorgan monopolio temporal sobre un invento a cambio de desvelarlo. La patente pretende promover la investigación privada sin coste para el contribuyente y sin que el resultado se pierda.

La acepción de invento patentable ha ido ampliándose a lo largo de los años y hoy en día admite, entre otros, algoritmos, programas informáticos, sustancias naturales y genes. Las patentes dificultan enormemente la programación de software libre porque muchos algoritmos están patentados, y, a la vez es muy difícil saber que lo están, sobre todo porque el software libre pone su código fuente a disposición, facilitando que pueda ser demandado por la infracción de las mismas. Mas (2005) desaprobó las patentes de software aduciendo al principio de *inventividad* debido a que la innovación informática es acumulativa y basada en experiencias e información previas. Mas señaló a las patentes de software como un impuesto encubierto que impide a las pequeñas empresas o individuos trabajar en las nuevas tecnologías. Lessig (2003) afirmó que las patentes no son malas por sí mismas, sino que son malas si no hacen un bien social. El autor matizó que no se pueda llamar bien social al beneficio de ciertas compañías a costa de dificultar la innovación, y ésta es la finalidad de muchas patentes hoy en día. Como ejemplo de inventividad nula se cita a Caruso (1999) quien, entre otras patentes, se refirió a la patente sobre el "uso de un libro ilustrado para enseñar a los conserjes como limpiar un edificio".

Para Lessig, el principal daño de las patentes son las amenazas de orden judicial para detener el uso de una tecnología patentada, valga como ejemplo julio de 2011, cuando hubo la disputa por dos patentes entre Apple y HTC que podría provocar la retirada de todos los móviles HTC en Estados Unidos, pues así lo pedía la demanda de la compañía de la manzana. Fogel (2005) describió la necesidad de adquisición de multitud de patentes de software en las empresas de Estados Unidos argumentando que en el software privativo, una empresa demandada por violar patentes, suele intentar acusar a la empresa denunciante de la violación de un número similar de patentes de su propiedad, con la finalidad de que ninguna de las dos empresas tenga que pagar nada exceptuando a sus abogados de propiedad intelectual. Fogel señaló además que la mayoría de los programadores de software -ya sean del ámbito del software libre o no- creen que las patentes de software deben ser abolidas.

Warshofsky (1994) reprodujo un memorándum de Bill Gates en que definió la estrategia de Microsoft acerca de las patentes de software:

La solución... es el intercambio de patentes... y patentar cuanto más podamos... Una empresa que en el futuro se inicie sin patentes propias será forzada a pagar cualquier precio que las grandes compañías elijan imponer. Este precio podría ser alto, ya que las empresas consolidadas tienen un interés en excluir a los competidores futuros.

---

Jefferson (1813) se refirió a la patentabilidad de las ideas con estas palabras:

Si la naturaleza ha hecho una cosa menos susceptible de propiedad exclusiva que todas las demás, ésta es la obra de la facultad de pensar llamada idea, que el individuo puede poseer exclusivamente mientras la guarde para sí; pero en el momento que se divulga, pasa, por fuerza, a ser posesión de todos, y quien la recibe no puede desprenderse de ella. Su carácter particular consiste también en que nadie la posee menos porque todos los demás la posean en su totalidad. Aquél que recibe conocimientos de mí, recibe conocimientos él sin disminuir los míos; así como quien enciende su vela en la mía recibe luz sin dejarme a oscuras. Que las ideas se difundan de uno a otro por todo el globo, para instrucción recíproca y moral del hombre, y para mejorar sus condiciones, parece haber sido el designio peculiar y benévolo de la naturaleza, cuando las creó, como el fuego, expansibles por todo el espacio, sin que disminuya su intensidad en ningún momento.... Los inventos, entonces, no pueden, por naturaleza, ser objeto de propiedad.

Se propone diferenciar las licencias de software libre según la aceptación o no del *copyleft*. Copyleft es una característica de algunas licencias que indica expresamente que las obras derivadas que puedan surgir de una obra licenciada como copyleft, deberán ser también licenciadas del mismo modo, impidiendo, así, que puedan convertirse en privativas (Stallman, 2002). Las licencias más representativas, sin y con copyleft, son las siguientes:

[No copyleft] Licencias tipo Berkeley Software Distribution (BSD). Estas licencias obligan únicamente a dar crédito a los autores, autorizando otros usos como la creación de un programa derivado privativo, que es percibido como una crítica por parte de la comunidad de software libre, pues no garantiza la libertad de versiones futuras de los programas. La licencia BSD se redactó para licenciar el software generado en los proyectos de investigación universitarios financiados por el gobierno de Estados Unidos, debido a que las universidades que desarrollaban dichos proyectos rehusaban comercializar el software que habían elaborado, pues entendían que ya había sido pagado por el gobierno, y que era lícito que cualquiera pudiera usarlo y adaptarlo como creyese conveniente. La licencia BSD original tenía una cláusula -ya retirada- que obligaba a mostrar la autoría del laboratorio de Berkeley en cualquier obra derivada. En el presente documento, al referirse a licencias *tipo BSD*, se entiende que son las licencias sin la cláusula mencionada.

[Copyleft] Licencia Pública General de GNU (GNU GPL). Es la licencia más usada en los proyectos de software libre (ver tabla 4.3). Esta licencia, además de reconocer la autoría, garantiza que ninguna obra derivada será privativa y prohíbe el uso de algoritmos patentados, obligando a que se conceda licencia de uso de la patente libre de tasas o a que se abandone el uso de la GPL.

Tabla 4.3: Licencias más usadas en proyectos de software libre.

Licencia	Copyleft	Uso ( %)
GNU General Public License (GPL) 2.0	Sí	42,34
MIT License	No	11,49
Artistic License (Perl)	Sí (débil)	7,94
GNU Lesser General Public License (LGPL) 2.1	Sí (débil)	7,08
BSD License 2.0	No	6,81
GNU General Public License (GPL) 3.0	Sí	6,42
Apache License 2.0	No	5,48
Code Project Open 1.02 License	No	2,11
Microsoft Public License (Ms-PL)	Sí (débil)	1,88
Mozilla Public License (MPL) 1.1	Sí (débil)	1,02

(Adaptado de BlackDuck Software Knowledge Base, 2012b). Datos correspondientes al 8 de marzo de 2012.

Otra lectura interesante sobre las licencias de software la ofrecen los Documentos Halloween. Estos son documentos confidenciales sobre las estrategias de la empresa Microsoft ante el software libre. La tabla 4.4 muestra una taxonomía de las licencias de software en el primero de los documentos. Esta clasificación se cree buena a excepción de las *aportaciones públicas a la parte principal del software*, que en el caso de software tipo BSD también deben ser consideradas como importantes.

Las licencias de software libre disponen de cláusulas de negación de garantía para proteger al autor de demandas por código que no funciona en una redistribución de su software. Este aspecto ha sido criticado desde el software privativo, pero cabe recordar que este último normalmente sólo garantiza que el soporte [el CD o DVD] es correcto y el programa en cuestión se ejecuta (González-Barahona, Seoane y Robles, 2003).

La mayoría del software necesita almacenar la información con la que se está trabajando en uno o varios archivos. Para que el contenido de un archivo sea interpretado correctamente por un software, el archivo debe ser almacenado en algún formato concreto. Del mismo modo que existe software privativo, también existen formatos cerrados (también denominados propietarios). Los formatos cerrados responden al interés de las empresas que los desarrollan, para que los datos generados por sus aplicaciones sean codificados, y así obligar al usuario a depender de sus productos, pues otros softwares equivalentes no podrán leer correctamente los datos de dichos archivos (Openformats.org, 2011). En contraposición, The Linux Information Project (2007) denomina *formato abierto* al que se encuentra publicado para que pueda ser estudiado por cualquiera, y

Tabla 4.4: Licencias de software según los *Documentos Halloween*.

	Disponible gratis	Redistribuable	Uso ilimitado	Código fuente disponible	Código fuente modificable	Aportaciones núcleo	Derivados deben ser libres
Comercial .....							
De prueba .....	Sí <sup>a</sup>	Sí					
Uso no comercial .....	Sí <sup>b</sup>	Sí					
Shareware .....	Sí <sup>c</sup>	Sí					
Programas sin <i>royalties</i> ( <i>freeware</i> ) .....	Sí	Sí	Sí				
Bibliotecas sin <i>royalties</i> .....	Sí	Sí	Sí	Sí			
Open source (tipo BSD) .....	Sí	Sí	Sí	Sí	Sí		
Open source (tipo Apache) .....	Sí	Sí	Sí	Sí	Sí	Sí	
Open source (tipo GNU/Linux) .....	Sí	Sí	Sí	Sí	Sí	Sí	Sí

Aportaciones núcleo: se refiere a si se producen aportaciones públicas a la parte principal del software por parte de personas ajenas al equipo principal de desarrolladores.

<sup>a</sup> Con limitaciones. <sup>b</sup> Depende del uso. <sup>c</sup> Licencias no activadas. (Adaptado de Raymond, 2004).

*formato libre* al formato abierto que no se encuentra gravado por derechos de autor, patentes, marcas registradas u otras restricciones para que cualquier persona pueda usarlo sin ningún coste monetario para cualquier propósito deseado. Openformats.org (2011) indican cuatro razones por las que no es conveniente usar formatos propietarios:

1. Riesgo de que el destinatario no pueda ser capaz de acceder a los datos.
2. Riesgo de transmisión de información confidencial.
3. Posible contribución a la propagación de virus y exposición propia al propio virus.
4. Apoyo a los monopolios existentes en el dominio de la comunicación electrónica.

El software libre acostumbra a usar formatos libres, pero algunos programas, bajo petición del usuario, también ofrecen la posibilidad de trabajar con formatos abiertos o incluso cerrados.

Expuesto el licenciamiento del software, se aborda ahora su desarrollo que incluye el paradigma de desarrollo, el lenguaje usado y la ingeniería del software.

Existen tres paradigmas fundamentales en la programación de software: programación imperativa, declarativa y orientada a objetos.

La programación imperativa consiste en la escritura a partir de un lenguaje de programación de un conjunto de algoritmos que el ordenador ejecuta. Entendiendo algoritmo como un “conjunto ordenado y finito de operaciones que permite hallar la solución de un problema” (RAE). El ordenador ejecutará los pasos descritos para conseguir completar la tarea. En este paradigma se especifica completamente cómo debe realizarse una tarea, y la descripción va muy ligada a las características de la máquina en que se ejecuta.

En la programación declarativa, a diferencia de la anterior, se especifica *qué* debe conseguirse en lugar de *cómo* se debe actuar (Julián, 2007), dando un papel de mucho más relieve a la lógica del programa que a su control. Las instrucciones a realizar no se introducen en el ordenador, serán inferidas por el sistema a partir de la descripción que el programador realice.

Finalmente, la programación orientada a objetos se caracteriza por usar objetos con una identidad, comportamiento y estado, que realizan las tareas asignadas a partir de su propia definición y la interacción con otros. Un objeto tiene definidas las propiedades y comportamientos en su *clase*, de modo que un objeto es una *instancia* -entendido como una unidad en particular- de la clase. La *abstracción* de los objetos es la información esencial que permite conocer sus características fundamentales sin preocuparse de detalles más internos. Dicha información viene *encapsulada* para evitar su modificación accidental por parte de otros objetos. Se destacan también las propiedades de *herencia* y *polimorfismo*; la primera permite que unos objetos puedan servirse de definiciones de



---

otros objetos, mientras que el polimorfismo posibilita que un objeto realice una acción concreta de una forma diferente a otros objetos de clase distinta. A continuación se ejemplifican las propiedades de la programación orientada a objetos con un supuesto programa informático que simule las condiciones de vida de un coral en un arrecife:

Sin aspirar a realizar una clasificación científica completa, el coral es un ser vivo y en concreto se considera un animal. Por tanto, el coral también lleva a cabo las funciones esenciales de los animales: alimentación, circulación, excreción, movimiento, reproducción, respiración y respuesta. De la multitud de familias existentes, se comenta la familia de las Euphyllias, y dentro de ellas la *Catalaphyllia Jardinei* (de ahora en adelante CJ). La simulación de la vida de una CJ usando la programación orientada a objetos incluiría:

Clase: Un tipo determinado de coral, en este caso la CJ.

Abstracción: La información esencial de datos y métodos de la clase CJ como son: la edad máxima, tamaño máximo, condiciones ideales de luz, corriente y temperatura, así como sus funciones.

Encapsulación: Otros objetos pueden saber cuántos pólipos tiene y el tamaño de los mismos, pero los mecanismos concretos de cómo se produce la alimentación no son relevantes para ellos.

Herencia: Los datos y los métodos de la clase CJ se heredan de clases superiores en la jerarquía. La reproducción es una característica heredada de los animales y la respuesta a otros corales, tratando de infectarlos alargando los pólipos, es un comportamiento heredado de las Euphyllias.

Polimorfismo: La reproducción es muy diferente a la de otros animales, pues se produce de forma asexual generando colonias de individuos que emergen de su esqueleto cálcico (Durso, 2005).

Objeto: Un ejemplar concreto de CJ. Este ejemplar tiene unos valores concretos de edad, tamaño, color y número de pólipos.

Como ejemplo de implementación, el software podría crear una matriz tridimensional de  $10^3$  objetos de la clase coral. En la dimensión vertical se produciría variación de iluminación, siendo los superiores los más iluminados. Los dos ejes restantes corresponderían a la alimentación y a la circulación de agua. Pasado un tiempo, se podría observar a los corales que se encuentran más sanos atendiendo a color, número de pólipos y tamaño. La figura 4.9 muestra código para la implementación en lenguaje C#.

Programar usando objetos establece unos niveles de segmentación que pretende solucionar los problemas que se experimentan en los grandes programas que usan la programación imperativa,

```

public class CJ : Euphyllia { //Clase CJ hereda de Euphyllia
    public CJ (int alimentacion, int iluminacion, int corriente) {
        //Se construye el objeto
    }
    public override alimentarse { //Método heredado pero con variaciones
        //Se usarán las variables: alimentacion, iluminacion y corriente entre otras
    }
    //Otros métodos heredados: crecer, cambiar-color, aumentar pólipo, atacar, ...
}
public static class Experimento {
    //Construcción de los objetos
    ArrayList corales = new ArrayList(1000); //Array que contendrá los 1000 corales
    for(int alimentacion = 1; alimentacion <= 10; alimentacion ++)
        for(int iluminacion = 1; iluminacion <= 10; iluminacion ++)
            for(int corriente = 1; corriente <= 10; corriente ++)
                corales.Add(new CJ (alimentacion, iluminacion, corriente));
    while(true) { //Bucle
        //Cada hora se alimenta a los corales llamando a: Alimentar. Cada 12 horas tomar datos
    }
}
public class Alimentar (ArrayList corales) { cada coral se alimentará
    foreach(CJ cj in corales)
        cj.alimentarse();
}

```

Figura 4.9: Ejemplo de implementación de la programación orientada a objetos.

como son: la aparición de sorpresas en el uso compartido de los datos, el control de las implicaciones que suponen los cambios en el código y la dificultad de reutilizar código (López y Xirgo, 2004).

La figura 4.10 muestra cómo han evolucionado en la última década los lenguajes de programación más populares. La programación orientada a objetos está fuertemente representada en este listado de lenguajes, debido a que la mayoría de ellos están concebidos para trabajar con este paradigma. El lenguaje C es un ejemplo de programación imperativa que continúa siendo muy popular pero cabe decir que acostumbra a vincularse a bibliotecas gráficas orientadas a objetos y esto lleva a que se acabe usando este paradigma aunque el lenguaje no haya sido concebido para ello.

La tabla 4.5 muestra el uso de lenguajes de programación en software libre. Tal y como se aprecia, los resultados son claramente diferentes a los de la gráfica anterior. La primera diferencia entre ambos conjuntos de datos estriba en el método de recogida de la información: en el primer caso se recopilan consultas sobre términos relacionados en la red, mientras que en la segunda -debido a que aborda el software libre- se identifica el código fuente de los proyectos. La segunda diferencia radica en el ámbito de desarrollo de los proyectos: muchos proyectos empresariales se desarrollan en Java o Visual Basic debido a que ha sido un requisito habitual en la contratación de programadores informáticos, mientras que en el software libre el programador se adhiere al proyecto que le interesa o crea su propio proyecto y, en ambos casos, él mismo decide el lenguaje. Por último, el éxito del lenguaje C en la tabla se debe en gran parte al proyecto GNOME -uno de los proyectos de escritorios más notorios en el software libre. El proyecto GNOME tiene una base de código mayoritaria en C, pero se sirve de bibliotecas de objetos, como se ha comentado anteriormente.

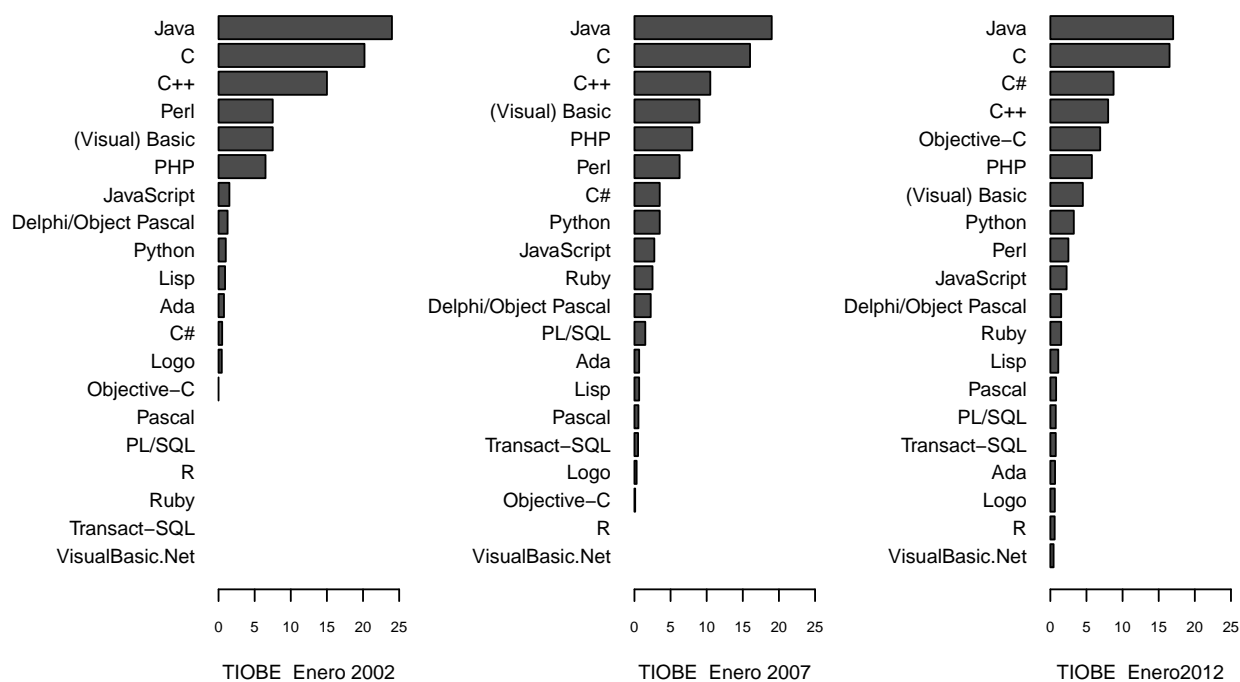


Figura 4.10: Evolución de los lenguajes de programación más populares en la actualidad a partir del índice TIOBE. (Elaborado a partir de TIOBE Software BV, 2012).

Tabla 4.5: Lenguajes de programación populares en software libre.

Posición	Lenguaje	%	Posición	Lenguaje	%
1	C	44,13	9	Ensamblador	1,60
2	C++	13,24	10	Autoconf	1,44
3	Java	8,93	11	SQL	1,24
4	Javascript	6,20	12	Ruby	1,21
5	Shell	5,51	13	C#	1,10
6	PHP	4,57	14	XML Schema	0,67
7	Python	2,51	15	Eiffel	0,64
8	Perl	2,24			

(Adaptado de BlackDuck Software Knowledge Base, 2012a). Datos correspondientes al 8 de marzo de 2012.

Habiendo mostrado la creciente presencia de la orientación a objetos, se aborda a continuación la gestión de proyectos de software.

Mas y col. (2007) calificaron la buena gestión de un proyecto de software como la que es capaz de reducir al mínimo las posibilidades de fallo de éste durante todo el transcurso del mismo. Los autores apuntaron como condiciones de fallo a la no consecución de los requisitos iniciales del proyecto, la inviabilidad económica del mismo y a un resultado que impida mantenerlo o le impida evolucionar. Por otro lado, los mismos autores también propusieron una acepción de la buena gestión de un proyecto como la que optimiza los recursos en cada fase del mismo.

Un proyecto de software acostumbra a gestionarse por una única persona que se encarga de realizar un seguimiento del desarrollo, así como la coordinación de los distintos equipos, tarea para la cual se suelen usar gestores de tareas, hojas de cálculo o diagramas de *Gantt*.

Raymond (1999) diferenció entre dos métodos de desarrollo del software a partir de la analogía con la construcción de una catedral o un bazar. En el software de tipo catedral, un equipo de expertos selecciona a un personal concreto y reducido para realizar una gran labor que no será finalizada, y por tanto usable, hasta pasado largo tiempo. Por otro lado, en el modelo de bazar existe una idea pero que puede cambiar considerablemente por la acción de multitud de personas, cuantas más mejor, tratando desde el primer momento que sea usable. Raymond argumentó que el descubrimiento del modelo de bazar le había “trastocado”, porque contraponía las enseñanzas tradicionales que abogaban por la planificación y centralización de los proyectos de una determinada “complejidad crítica”. Muchos proyectos de software libre son estandartes del modelo de bazar y lo demuestran publicando continuamente versiones en las que han colaborado gran cantidad de autores siguiendo el estilo “Libera pronto, libera a menudo”, entendiendo *libera* como el proceso de publicar nuevas versiones. Se destaca como ejemplo de bazar al desarrollo de Linux. El creador de Linux, Linus Torvalds, refiriéndose a la publicación continua del código, manifestó: “Dado un número suficientemente elevado de ojos, todos los errores se convierten en obvios”, frase que Raymond (1999) denominó la “Ley de Linus” y demostró su veracidad con el siguiente supuesto:

Si la Ley de Linus fuera falsa, entonces cualquier sistema suficientemente complejo como el kernel de Linux, que está siendo manipulado por tantos, debería haberse colapsado en un punto bajo el peso de ciertas interacciones imprevistas y errores "muy profundos" inadvertidos. Pero si es cierta, bastaría para explicar la relativa ausencia de errores en el código de Linux (pág. 9).

Thompson (2008) relató los problemas que producían las máquinas de votación Diebold AccuVote-TSX aparentemente cada cien votos. Por muchos sistemas de auditoría que tuvieran, en el momento de las votaciones aparecían errores que obligaban a reiniciar el sistema cuando votaban personas mayores. Se decía que la máquina se colgaba cada 100 votaciones, pero estudios posteriores demostraron que cuando un ciudadano en lugar de hacer un clic rápido para votar, éste

---

mantenía el botón apretado, se producía un error de arrastrar y soltar no implementado en el sistema, y las personas mayores que no tenían experiencia en el manejo de máquinas eran más propensas a *encontrar* ese error. Años más tarde, John Hopkins obtuvo el código fuente de la aplicación y rápidamente descubrió muchos más errores que no habían sido advertidos por los mecanismos de auditoría (Kohn y col., 2004). Schindler (2007) comparó el tiempo de descubrimiento y resolución de fallos graves en software libre y software privativo, saliendo el primero claramente favorecido.

El Departamento de Defensa de los Estados Unidos (2009) se refirió de este modo a la fiabilidad del código compartido:

La revisión de pares continua y amplia que permite el código fuente a disposición del público apoya la fiabilidad del software y los esfuerzos de seguridad a través de la identificación y eliminación de los defectos que de otro modo podrían pasar desapercibidos por un equipo de desarrollo más limitado.

Independientemente de si un proyecto sigue el desarrollo de una catedral o de un bazar, Asundi (2001) encontró tres lecciones vinculadas a la gestión de proyectos en que el software privativo puede aprender del de licencia libre:

Organización del personal: En lugar de asignarse roles a cada miembro, en el software libre cada uno asume voluntariamente el suyo.

Comunicación informal: La comunicación informal es vital para coordinar el desarrollo de un proyecto de software (Herbsleb y Grinter, 1999). En el software libre, es frecuente la comunicación informal entre desarrolladores, así como entre éstos y los usuarios.

Mejora del soporte al usuario: Muchos programas comerciales tienen un mal soporte al usuario mientras que Von Hippel y Lakhani (2000) demostraron que los proyectos de software libre, en general, tienen un soporte que resulta de ayuda, gracias a la gran cantidad de personas que se prestan a ofrecer información.

Estas tres lecciones son relativas a la interacción entre los desarrolladores y de éstos con los usuarios, característica de la metodología de la programación *ágil*, siendo ésta habitual en los proyectos de software libre sin ser exclusiva de éstos. Los propósitos de la programación *ágil* se muestran en la tabla 4.6.

La programación *ágil* enunció un listado de doce principios que completaron sus propósitos:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Tabla 4.6: Propósitos de la programación ágil.

Las elementos de la izquierda se valoran más que los de la derecha.		
Individuos e interacciones	⇔	Procesos y herramientas
Software funcionando	⇔	Documentación extensiva
Colaboración con el cliente	⇔	Negociación contractual
Respuesta ante el cambio	⇔	Seguir un plan

(Adaptado de Fowler y Highsmith, 2001).

- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

(Fowler y Highsmith, 2001, pág. 7)

Tal y como se deduce de los principios citados, las metodologías ágiles priman la publicación de nuevas funcionalidades y versiones por encima de los calendarios rígidos de versiones característicos de las metodologías tradicionales.

---

De entre las distintas metodologías de programación ágil, destaca la programación extrema (XP por eXtreme Programming) por su popularidad. Su descubridor, Beck (2000), describió los cambios que debe afrontar la programación con estas palabras:

En el software todo cambia. Los requisitos cambian. Los diseños cambian. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio, porque el cambio va a ocurrir; el problema, más bien, es nuestra incapacidad para hacer frente al cambio. (pág. 30)

La programación extrema se basa en la simplicidad, la comunicación permanente con el cliente para comprobar los requisitos constantemente y la programación por parejas, en que un programador escribe código y otro lo prueba para después cambiarse los roles (Calleja y Riesco, 2007). En la programación extrema, antes de programar alguna funcionalidad, se programa una prueba sobre la misma, de modo que cuando la funcionalidad esté escrita, deberá superar la prueba definida. Si se precisa, por ejemplo, programar una funcionalidad de un software para que se conviertan las unidades del sistema internacional en unidades métricas, primero se realizará un pequeño programa denominado *prueba* que comprobará si la conversión se ha realizado correctamente. Mas y col. (2007) añadieron que los modelos tradicionales de creación del software incluían la fase de pruebas como un proceso que había que llevar a cabo una vez finalizado el desarrollo, siendo esto altamente contraproducente, tanto por el coste de arreglar fallos o deficiencias una vez terminado el desarrollo, como por la naturaleza evolutiva de muchos proyectos (especialmente los de software libre), en los que la fase de desarrollo no termina nunca, estrictamente hablando. Aunque a primera vista puede parecer el doble de trabajo, a lo largo se es capaz de advertir errores antes de publicar una versión. Beck (2000) identificó a la programación extrema como una disciplina de desarrollo de software, y defendió el término *disciplina* de este modo:

Es una disciplina porque hay algunas cosas que debes hacer si quieres usar la programación extrema. No se trata de elegir entre escribir los tests o no. Si no haces los tests, no estás en la programación extrema: fin de la discusión. (pág. 9)

El autor afirmó que esta modalidad de programación debe ser llevada a cabo por equipos de dos a diez personas que dedican una fracción de cada día a la programación de los tests por parejas. Algunos proyectos no pueden seguir las directrices de la programación extrema, pero desean igualmente definir un conjunto de pruebas que se llevan a cabo manualmente. Mas y col. (2007) clasificaron las técnicas manuales de comprobación del software en técnicas de tipo informal, en las que cada uno de los miembros de un grupo de testadores, instala la aplicación, la prueba sin ningún plan determinado e informa de los errores que van encontrando para que sean solucionados; y técnicas formales, en que se elaboran guiones de pruebas (como el que se muestra en la tabla 4.7) con el objetivo de “asegurar que la funcionalidad más común del programa no se

Tabla 4.7: Guión de pruebas (manual) en OpenOffice.org.

Copiar/pegar celdas combinadas en otras celdas combinadas	
Descripción	Estado: Terminada
	Documentos necesarios: -
	1. Abrir nuevo documento de hoja de cálculo.
	2. Seleccionar área de celdas A1:D5.
	3. Combinar área: Formato - Combinar celdas.
	4. Seleccionar área de celdas F7:G7 y combinarla: Formato - Combinar celdas.
	5. Copiar área de celdas F7:G7 al portapapeles: CTRL-C.
	6. Pegar portapapeles en A1.
	7. Verificar que el área de celdas A1:D5 no está combinada excepto A1:B1.
	8. Cerrar documento.

Adaptado de [http://quaste.services.openoffice.org/index.php?option=com\\_tcs&task=tcs\\_show&tcsid=3039&language=1](http://quaste.services.openoffice.org/index.php?option=com_tcs&task=tcs_show&tcsid=3039&language=1)

ha visto afectada por las mejoras introducidas desde la última versión, y que un usuario medio no encontrará ningún error grave”.

El estándar de errores en la industria del software es de 1 a 25 errores por cada 1000 líneas de código terminado (McConnell, 2004). En el software privativo se acostumbra a dedicar equipos de programadores a la detección de fallos, mientras que en el software libre esta labor no es usual por lo poco *glamouroso* de la tarea y porque se confía en la comunidad de usuarios (Fogel, 2005). En efecto, en el software libre los usuarios juegan un importante papel en las pruebas, llegando a tratar como co-desarrolladores, pues esta actitud es la forma menos problemática de programar deprisa y resolver los fallos de forma efectiva (Raymond, 1999). Robles (2006) afirmó que los que crean proyectos abiertos que posibilitan el envío de feedback y otras contribuciones tienen una ventaja clara sobre los programadores de proyectos privativos. Por otro lado, McConnell (1999) argumentó que la corrección de errores en el software libre se produce en la etapa de programación y no en la de diseño, que se correspondería con la mejor práctica.

Otra característica de las metodologías ágiles son las *historias de usuario*. En éstas, el cliente o usuario escribe con sus propias palabras lo que el software debería hacer. Estas historias no tienen un lenguaje técnico, excepto cuando el cliente es un programador que también se servirá del programa para desarrollar una extensión. Las historias de usuario servirán para realizar la planificación de las versiones a publicar, así como para que el cliente de un software pueda determinar si se ha conseguido lo que se pretendía con el desarrollo.

Según Cohn (2004), las historias de usuario se componen de tres aspectos:



1. Una descripción escrita, que se usará para la planificación y como recordatorio.
2. Conversaciones sobre la historia, que servirán para extraer los detalles sobre la misma.
3. Tests y documentos, que serán útiles para determinar si la historia se ha terminado de implementar.

Debido a que normalmente las historias de usuario comienzan con la escritura en una tarjeta o bloc de notas, Jeffries (2001) denominó *tarjeta*, *conversación* y *confirmación* a los tres aspectos citados. Hoy en día muchas historias de usuario se escriben directamente en interfaces web y se encuentran disponibles para que los desarrolladores y los usuarios puedan consultarlas.

Se llama *ciclo de vida del software* a las fases por las que pasa un software desde que se concibe como proyecto, hasta que está listo para usarse. Las metodologías ágiles suelen usar modelos iterativos de ciclo de vida del software, que permiten que las fases de análisis, diseño, desarrollo y pruebas se retroalimenten, y empiecen cuanto antes sea posible. De este modo, se denomina *iteración* al proceso que acaba llevando a una versión concreta. En la figura 4.11 se propone un modelo de ciclo de vida del software en programación extrema.

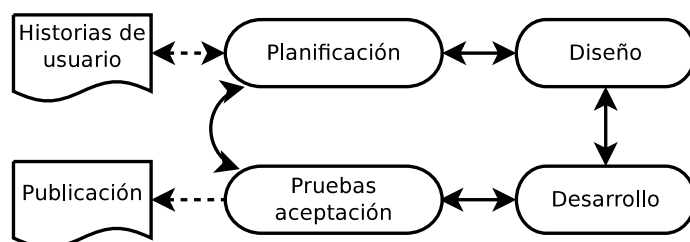


Figura 4.11: Ciclo de vida en la programación extrema. (Adaptado de Mas y col., 2007).

A medida que se van produciendo las iteraciones, el código fuente va cambiando y creciendo tal y como ilustró Robles (2006) y se muestra en la figura 4.12. El cálculo del coste de un programa informático de un tamaño determinado y su tiempo de desarrollo son objetivos de la ingeniería del software. Los modelos COCOMO (Boehm, 1984) y COCOMO II (Boehm y col., 2000) permiten realizar estos cálculos a partir de unos coeficientes que se introducen para cada caso. Existen distintas modalidades de COCOMO: básico, intermedio y detallado. El modelo básico destaca por su sencillez debido a que realiza los cálculos inspeccionando únicamente la cantidad de código escrito en cada lenguaje. Los pequeños proyectos que se realizan con metodologías ágiles son denominados *orgánicos* en COCOMO, pues, según su clasificación, orgánicos se corresponde con proyectos formados por un número reducido de personas con buena experiencia en trabajar siguiendo requisitos flexibles. González-Barahona y col. (2001) explicaron que los modelos COCOMO son buenos para determinar el coste de desarrollo en la industria del software propietario según modelos tradicionales, pero su estimación es muy exagerada en la valoración de pequeños proyectos de software libre por la publicación continua de versiones, los mecanismos de control

y la solución de errores por parte de la comunidad, la reutilización e intercambio de código entre proyectos y el modelo de desarrollo distribuido en que los desarrolladores se encuentran en diferentes países.

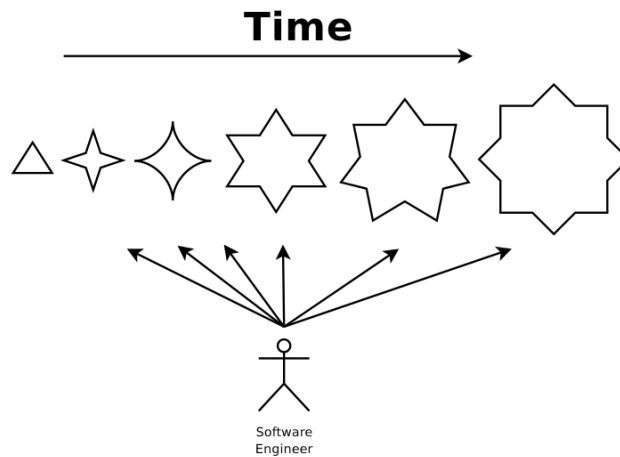


Figura 4.12: Punto de vista de la evolución del software (Robles, 2006).

A medida que el software va creciendo, los programadores dividen el desarrollo en distintas versiones. Es común que exista una versión estable, que se caracteriza por un número bajo de fallos, y una o varias versiones inestables en que se prueban funcionalidades determinadas. Los programas informáticos de control de versiones se dedican a esta labor, facilitando además que distintos programadores puedan estar trabajando en el mismo código desde diferentes partes del mundo. Se destacan tres programas de control de versiones libres por ser muy acogidos por los proyectos de software libre: Bazaar (Canonical Ltd, n.d.), GIT (Hamano y Torvalds, n.d.), y Mercurial (Mackall, n.d.). Por otro lado, día a día, los usuarios y los mismos programadores piden nuevas funcionalidades o advierten de errores. Es común que ambos tipos de solicitudes (nueva funcionalidad y error) sean administradas con una herramienta de seguimiento de fallos. Bugzilla (Mozilla Foundation, n.d.), Trac (Edgewall Software, n.d.).

El proceso de diseño del hardware es muy distinto en función de si se trata de hardware estático o de hardware reconfigurable. El primero tiene una existencia física, ya que está constituido por un conjunto de sistemas electrónicos, mientras que el segundo está constituido por un conjunto de código de programación que permite especificar con todo detalle su estructura y funcionalidad. A partir de este código se generan unos ficheros de configuración para que se reconfiguren los dispositivos del tipo Field Programmable Gate Array (FPGA). La figura 4.13 es un ejemplo de hardware estático basado en un PIC y la figura 4.14 es un ejemplo de FPGA.

La modificación del firmware puede ser suficiente para corregir o añadir una funcionalidad en el hardware, pero muy comúnmente la modificación requerirá de un cambio en el circuito mismo. Cualquier modificación en el hardware es muy costosa si éste es estático, mientras que es asumible si es reconfigurable. En el primer caso se requiere financiar tanto la fabricación de prototipos como



Figura 4.13: Skypic, una tarjeta de hardware estático con un microcontrolador PIC. (González-Gómez y Prieto-Moreno, 2005).

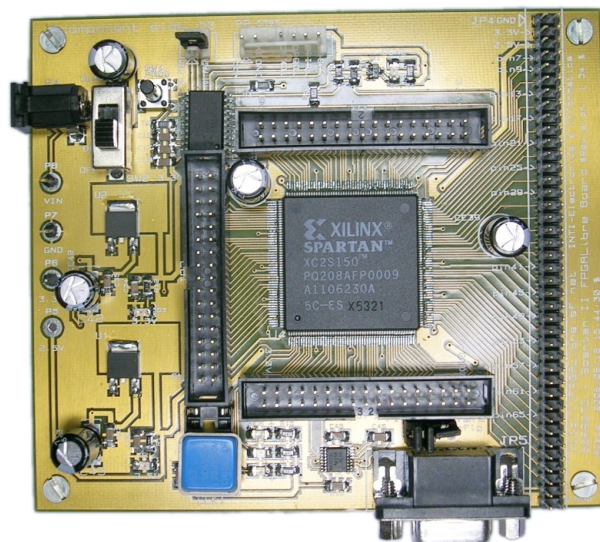


Figura 4.14: S2PROTO, una tarjeta que incorpora un FPGA. (FPGA Libre, n.d.)

la producción en sí, de modo que aunque no se produzca ningún error en el diseño, el proceso conlleva la inversión de varios miles de euros. Por otro lado, en el hardware reconfigurable el proceso de modificación es mucho más parecido al proceso de generación del software y se pueden simular situaciones para identificar si la programación es correcta. En definitiva, el hardware reconfigurable tiene un ciclo de vida similar al del software, añadiendo y corrigiendo a lo largo del tiempo; mientras que el hardware estático requiere de una planificación y revisión muy importante antes de la producción.

En cuanto al hardware, desgraciadamente no existe una definición unificada que pueda servir para catalogar un hardware como libre o no, aunque el creador del proyecto GNU, Richard Stallman, ha postulado que las ideas del software libre se pueden aplicar a los archivos necesarios para su diseño y especificación, pero no al circuito físico en sí (Stallman, 1999). Se cita a continuación la definición de *diseño de hardware abierto* de González, González y Gómez-Arribas (2003):

Un diseño se considera hardware abierto si ofrece las cuatro libertades del software libre en el esquemático [archivo que indica los componentes lógicos y las señales que se conectan entre ellos], PCB [archivo que especifica el lugar físico en el que situar los componentes, sus dimensiones, encapsulados y qué caminos siguen las pistas para unir sus pines] y el fichero GERBER que contiene la información para la fabricación (este último puede no estar disponible).

González, González y Gómez-Arribas (2003) propusieron clasificar el diseño del hardware en función de las restricciones impuestas por el programa de diseño. Los autores identificaron las restricciones en cada uno de los componentes comentados: esquemático, PCB y fichero de fabricación. El último componente puede estar disponible en formato GERBER o no, recibiendo la valoración L o X, respectivamente. En cambio, el esquemático y el PCB pueden tener cualquiera de los siguientes valores:

- |   |   |
|---|---|
| L | El programa con el que se ha diseñado el plano es libre.  |
| P | El programa es privativo y sólo se puede ejecutar en sistemas operativos privativos.  |
| M | El programa es propietario, pero es multiplataforma, siendo alguna de las plataformas un sistema operativo libre. Siendo éste el caso de la conocida herramienta EAGLE (CadSoft, n.d.). |

De este modo, un hardware con diseño LLL no tiene restricciones ya que cualquier persona lo puede ver y modificar usando el programa libre con el que se ha diseñado, sin estar ligado a ningún fabricante ni sistema operativo, y además el archivo GERBER está disponible. Los autores expresan que no conocen ningún diseño de este tipo en el momento de la publicación del artículo.

Las organizaciones prestigiosas que impulsan el hardware abierto acostumbran a redactar sus propias licencias, valgan como ejemplos el MIT con la MIT Open Source License (Massachusetts

---

Institute of Technology, 2007), la NASA y su Open Source Software Agreement (National Aeronautics and Space Administration, 2009), y el CERN con la CERN Open Hardware Licence (Bij, 2011). Algunos proyectos de organizaciones pequeñas también redactaron sus propias licencias, como por ejemplo Balloonboard (Balloonboard.org, 2003) o LART Linux Advanced Radio Terminal (Bakker, 2000), pero la mayoría de proyectos pequeños usaron licencias redactadas por otros. Entre estos últimos, se destacan la tarjeta Skypic (González-Gómez y Prieto-Moreno, 2005), el microcontrolador Arduino (Arduino, n.d.), y el proyecto Global Village Construction Set, que pretende documentar la construcción de unas 50 máquinas industriales que permitan construir una civilización con las comodidades de hoy en día (Open Source Ecology, n.d. Winters, 2011). Skypic y Arduino están licenciados bajo Creative Commons Attribution-Share Alike, y Global Village Construction Set se encuentra licenciado dualmente bajo Creative Commons by-sa3.0 y GNU Free Documentation License 1.2.

Las licencias Creative Commons nacieron gracias a la voluntad de distribuir documentación y contenido multimedia en la red. Estas licencias establecieron cuatro parámetros que el autor puede elegir para completar su licencia:

- Reconocimiento (by): Se deberá reconocer la autoría original.
- No comercial (nc): No es posible usar la obra con finalidades comerciales.
- Sin obras derivadas (nd): No se permite distribuir modificaciones de la obra.
- Compartir igual (sa): Si existiesen obras derivadas deberían licenciarse del mismo modo que la obra original.

En la versión 3.0 de estas licencias se aceptaron las siguientes combinaciones:

- Combinaciones que licencian a productos más libres (fully open):
  - Reconocimiento (by).
  - Reconocimiento - CompartirIgual (by-sa).
- Combinaciones que licencian a productos más restrictivos:
  - Reconocimiento - NoComercial (by-nc).
  - Reconocimiento - NoComercial - CompartirIgual (by-nc-sa).
  - Reconocimiento - NoComercial - SinObraDerivada (by-nc-nd).
  - Reconocimiento - SinObraDerivada (by-nd).

Por último, Creative Commons ha incorporado recientemente mecanismos para marcar obras como Domino publico (“sin derechos” o “copyright desconocido”). Las licencias Creative Commons están fuertemente inspiradas en las licencias de software libre, tal y como se constata en el parecido del parámetro *compartir igual* con el copyleft.

La aparición de estas licencias tomó mayor relieve desde la aparición de la Web 2.0. La nueva web hace realidad por fin el paradigma de comunicación *de muchos a muchos* gracias a sitios como Flickr, YouTube, Digg, Twitter o la Wikipedia. Estos sitios se encuentran repletos de contenido generado, organizado y filtrado por el usuario, que se sirve de mecanismos de moderación digitales. Graham (2010) se refiere a la Web 2.0 como:

- La ‘web de lectura y escritura’.
- No son páginas estáticas, sino contenido dinámico.
- La construcción de redes a partir de la creación de vínculos entre distintos ‘nodos’.
- Importancia de la participación y del contenido generado por el usuario.
- Se caracteriza por ser colaborativa y abierta.

Las licencias Creative Commons han tenido también un importante impacto en el ámbito académico, tal y como se observa en los artículos publicados en las revistas de la Public Library of Science (PLOS) (Public Library of Science, n.d.), y los materiales de las universidades adscritas al OpenCourseWare (n.d.), de entre las que se destacan The Open University (The Open University, n.d.) y la Universitat Oberta de Catalunya (Universitat Oberta de Catalunya, 2008). Fuera del ámbito formal, June Cohen se manifestó así al respecto de las licencias Creative Commons en el sitio de conferencias TED:

No se trata sólo de crecimiento, se trata de cómo la audiencia global se ha convertido en un equipo global, acogiendo nuestra marca y promoviendo más innovación. Una licencia Creative Commons comunica claramente que se tiene un postulado realmente serio acerca de la difusión de las ideas. (Creative Commons, 2011)

El contenido compartido colabora con la creación de una memoria colectiva creciente e imparable y así lo apunta Candeira (2001) en la etapa pre-2.0 mientras promovió la popularización de los Weblogs o diarios colectivos sobre diferentes temáticas, como Slashdot y Barrapunto. La proliferación de los servicios web, entendidos como “sistemas de software diseñados para permitir la interoperabilidad entre máquinas a través de una red” (adaptado de W3C, 2004), ha permitido que los sitios web sean más funcionales. En el sitio web de Flickr, los usuarios de cámaras fotográficas suben sus fotografías con la finalidad de compartirlas con cualquier visitante de la red. Entre noviembre de 2006 y febrero de 2007, el número de nuevos usuarios en Flickr fue

---

de 950.000 (Mislove y col., 2008); además, en los últimos tres años, se subieron tres millones de fotografías al año (Flickr Blog, 2009). Buscadores como Yahoo y Flickr mostraban, a final de 2010, 400 millones de trabajos licenciados bajo Creative Commons, el 40 % de los cuales habían sido marcados con las opciones más abiertas: Reconocimiento (by), Reconocimiento - Compartirlgual (by-sa), o dominio público (Creative Commons, 2011).

Los cambios en la forma en que los usuarios producen y comparten la información y conocimiento aumentan su autonomía, participación y diversidad (Vickery, 2007). Por otro lado, el crecimiento aumenta la preocupación por la fiabilidad de los datos. La Wikipedia en lengua inglesa (Wikipedia, 2012) es el sitio web que recopila y publica mayor información sobre estudios comparativos sobre su propia fiabilidad, llevando a numerosos autores a estudiar el fenómeno desde distintas ópticas (Korfiatis, Poulos y Bokos, 2006; Alder y col., 2008; Okoli, 2009).

Racionero (1983) afirmó que las tres revoluciones tecnológicas del mundo -agrícola, urbana e industrial- vinieron sucedidas por revoluciones ideológicas -copernicana, darwiniana y freudiana. En el año 1983, este autor esperaba que la revolución informática fuera acompañada por una revolución cultural con valores que defendieran la calidad de vida, el estado estacionario, la descentralización espacial y tecnológica, y el ocio creativo; una revolución que resolviera la contradicción entre una tecnología avanzada y los valores anticuados con que ésta se aplica. Goldsmith (1999, pág. 134) también se refirió al acceso a la cultura global como pieza importante de la comunidad global del futuro:

El acceso a la cultura irá mucho más allá de la capacidad para comprender el arte o la música. Será necesario comprender mejor a las personas. No es casual que los regímenes represivos, que fomentaban el odio hacia otros grupos, traten de impedir el flujo de las comunicaciones, por ejemplo, mediante la quema de libros.

El presente marco teórico ha recorrido diversas áreas de conocimiento tratando de enmarcar tests de salto en la biomecánica deportiva con la finalidad de evaluar la fuerza y potencia del deportista. La exposición sobre el desarrollo de software y hardware, así como de las licencias de éstos y de los contenidos en red, pretende actuar de glosario narrado para la comprensión de las partes aplicadas de la tesis.





## Parte II

# Estudios de medición del tiempo

El objetivo principal de esta primera parte aplicada es:

Crear, validar y distribuir una herramienta de licencia libre que mida los tiempos de contacto y vuelo en el salto vertical, usando una plataforma de contactos.

Dicho objetivo principal se trata a lo largo de los siguientes cuatro capítulos, cada uno con sus objetivos específicos, metodología, resultados y discusión.

Capítulo 5: Desarrollo y validación de un microcontrolador.

Capítulo 6: Desarrollo y validación de una plataforma de contactos.

Capítulo 7: Desarrollo y validación de un software de ordenador.

Capítulo 8: Distribución e internacionalización.

El desarrollo de instrumentos que se describe en los capítulos 5, 6 y 7 se ha realizado en paralelo, mientras que las validaciones se han hecho en el mismo orden que los capítulos.



---

## 5. Desarrollo y validación de un microcontrolador

En dos ocasiones me han preguntado: 'si pone datos incorrectos en la máquina, ¿saldrán las respuestas correctas?'. Soy absolutamente incapaz de hacerme una idea del tipo de confusión de ideas que pueden provocar que alguien haga una pregunta así.

---

Charles Babbage

Un software de ordenador que se encargue de cronometrar un dispositivo externo, como una plataforma de contactos, tiene una fiabilidad muy baja, pues si la multitarea del sistema operativo en que se ejecuta es de tipo concurrente o preferente, el programa tendrá una variabilidad de latenciaa que afectará al cronometrado. Por otro lado, es difícil garantizar que el equipo del usuario tendrá multitarea real y que la tarea de cronometrado dispondrá de un microprocesador exclusivo, o que el usuario podrá configurar un sistema operativo de tiempo real. Por tanto, se requiere que el cronometrado de un dispositivo externo sea realizado por un microcontrolador -y no por un ordenador personal- pues el microcontrolador al ser monotarea puede realizar lo que se precisa sin riesgo a *distraerse* en otras cosas.

El presente capítulo detalla el proceso de construcción y validación de un microcontrolador que funcione a modo de cronometrador digital y envíe los registros de tiempo a un software de ordenador.

### 5.1. Objetivo

El objetivo específico de este primer apartado es el siguiente:

Crear y validar un microcontrolador (PIC) de hardware abierto que permita cronometrar los estados ofrecidos por una plataforma de contactos o célula fotoeléctrica.

El microcontrolador se denominará Chronopic, del cual hay tres versiones desarrolladas.

### 5.2. Metodología

Debido a que la tarea del microcontrolador era muy específica y estable, se optó por un hardware estático.

El software que se usó para el diseño del microcontrolador fue la herramienta libre KiCad (KiCad Developers, n.d.). Se prefirió ésta antes que la conocida herramienta privativa EAGLE porque la primera garantiza que la placa será Hardware abierto tipo LLL en la clasificación de open hardware (González, González y Gómez-Arribas, 2003).

El firmware se encarga de grabar los tiempos en un *buffer* para ser enviados al ordenador cuando éste pueda recibirlos. Así, la multitarea del ordenador puede enlentecerse en el proceso de gestionar y mostrar los datos al usuario, pero no provoca retrasos en el cronometrado. El mínimo tiempo registrado en la medición de saltos es de 50 ms<sup>1</sup>, los pulsos inferiores son considerados espúreos por “rebotes” de electricidad. La conexión con el ordenador es por puerto USB de manera que no sólo se puede transmitir información sino que además la placa se alimenta del primero.

El firmware se programó en lenguaje ensamblador usando las herramientas GNU, constituidas por las *gputils* y el simulador *gpsim*.

Se grabó un *bootloader* con la finalidad de que el firmware pueda ser actualizado fácilmente desde cualquier ordenador.

Se creó un programa informático en Python para facilitar la grabación del firmware en el sistema operativo Windows y en Linux.

La fabricación del microcontrolador se produjo en diversas etapas con la finalidad de hacerlo evolucionar a partir de una fabricación manual nacional hasta una producción industrial internacional.

Las herramientas usadas para la validación del microcontrolador fueron los generadores de ondas cuadradas Hameg HM 8035 y Hameg HM 8030-4, y el osciloscopio Agilent 54621A (ver figura 5.1).



Figura 5.1: Instrumentación para la validación de la sensibilidad del microcontrolador. A la izquierda se muestra el Hameg HM 8035 y a la derecha el Agilent 54621A.

Debido a que los tiempos más cortos de un tiempo de contacto en el salto no son inferiores a 150 ms y los más largos no son superiores a 700 ms, se usó este rango de valores. Para generarlos se usaron ondas cuadradas en frecuencias determinadas, tal y como explica la tabla 5.1.

<sup>1</sup>El microcontrolador diseñado también se usa para el cronometrado de carreras a partir de células fotoeléctricas así como para la decodificación de registros de un encoder lineal -usos no descritos en esta tesis-. En el primer caso se acepta un mínimo de 10 ms, mientras que en el segundo no existe tal mínimo.

Además, se experimentó con frecuencias distintas al rango de valores de saltos para ver cómo el microcontrolador respondía a frecuencias que podrían corresponder a otros tipos de test.

Tabla 5.1: Frecuencias y tiempos equivalentes en la validación del microcontrolador.

Frecuencia (Hz)	Tiempo total (ms)	TC o TV (ms)	Comentario
10,0000	100	50	Tiempo mínimo que se registra.
3,3333	300	150	Estimación de TC mínimo de un salto.
0,7143	1400	700	Estimación de TV máximo de un salto.
0,2500	4000	2000	Tiempo máximo que se desea explorar.

El protocolo fue el siguiente:

1. Se generaron ondas cuadradas de 50 % de carga con los generadores de ondas:
  - Hameg HM 8035 para el rango comprendido entre los 9 Hz y los 1,5 Hz a intervalos de 0,5 Hz.
  - Hameg HM 8030-4 en las frecuencias: 1,5 Hz; 1 Hz; 0,5 Hz y 0,25 Hz.
2. Se detectaron las ondas usando el osciloscopio y el microcontrolador creado.
3. Se compararon los resultados.

Para evitar que un hardware interfiriese sobre otro, nunca se usaron los dos generadores a la vez, ni se captó la onda generada al mismo tiempo con el osciloscopio y el microcontrolador. Así, el protocolo detallado para la captación de una frecuencia de ondas fue el siguiente:

1. Se creó la onda con el generador de ondas a la frecuencia deseada (según el regulador del mismo generador)
2. Se conectó el osciloscopio y se tomó el valor que muestra éste como valor real de la onda.
3. Se desconectó el osciloscopio y se conectó el microcontrolador diseñado.
4. Se registraron los impulsos eléctricos con el microcontrolador.
5. Se compararon los valores del microcontrolador con los del osciloscopio.

Para la comparación de los resultados se calculó el error relativo de cada tiempo de contacto y cada tiempo de vuelo detectado por el microcontrolador diseñado respecto a las señales detectadas por el osciloscopio, según la fórmula:

$$Error\ relativo\ (\%) = 100 * |T_{exp} - T_{gold}| / T_{gold}$$

siendo  $T_{exp}$  el tiempo encontrado en la plataforma de contactos diseñada o experimental, y  $T_{gold}$  el tiempo de vuelo correspondiente al Gold Standard: el osciloscopio. Posteriormente se obtuvo la media aritmética del error relativo en los tiempos de contacto y el error relativo en los tiempos de vuelo.

### 5.3. Resultados y discusión

Hasta la fecha se han desarrollado tres versiones del microcontrolador. La tabla 5.2 muestra las etapas seguidas en la producción del mismo.

Tabla 5.2: Etapas de producción del microcontrolador.

Nombre	Fecha de lanz.	País de desarr.	Producción	Unidades
Figura	Autor y comentarios			
Chronopic 1 (figura 5.2)	2005	España	Manual	150
	Juan González-Gómez. Skypic y tarjeta de expansión para saltos.			
Chronopic 2 (figura 5.3)	2007	Argentina	Manual	?
	Juan Fernando Pardo. Unidad mínima con componentes presentes en Argentina.			
Chronopic 3 (figura 5.4)	2008	España	Industrial	200
	Ricardo Gómez González.			
Chronopic 3 (figura 5.5)	2010	Taiwán	Industrial	500
	Ricardo Gómez González. Producción de bajo coste.			

En los apéndices de esta tesis se muestran: los planos de los microcontroladores (apéndice A.1) y la lista de materiales (apéndice C). Los archivos fuente en KiCad así como los archivos para la fabricación GERBER y taladros se encuentran en la Página técnica de Chronopic 3 (Gómez y González, 2008). Se licenció Chronopic 3 como Reconocimiento-CompartirIgual 2.5 España (CC BY-SA 2.5), y el código fuente del Firmware y el Bootloader se colgaron en un subdirectorío del GIT de Chronojump (Chronojump, n.d.).

El programa informático grabador del firmware se denominó Chronopic-Firmwarecord (Morales y De Blas, 2010) (ver figura 5.6). Se crearon versiones para el sistema operativo Windows y

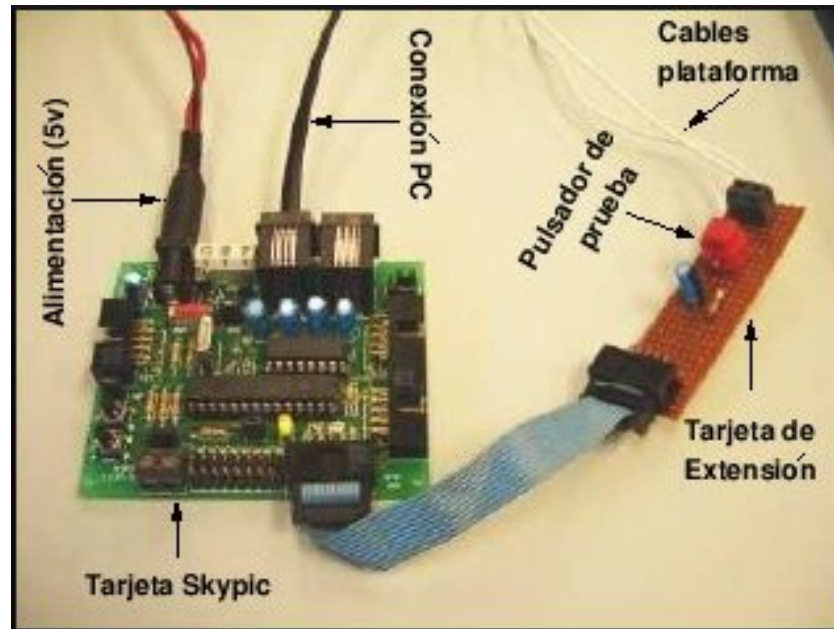


Figura 5.2: Chronopic 1.

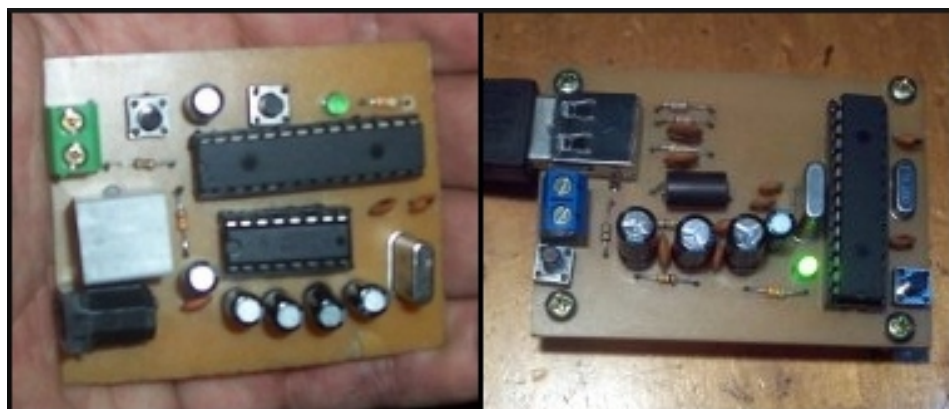


Figura 5.3: Chronopic 2. A la izquierda se muestra la versión serie y a la derecha la versión USB.

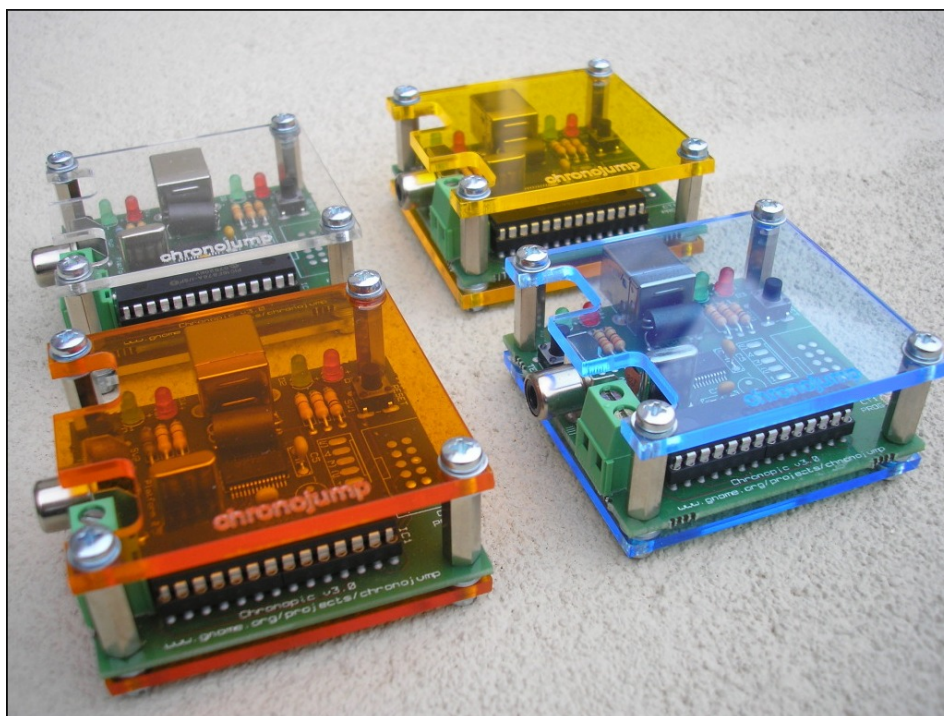


Figura 5.4: Chronopic 3. Fabricación en España.

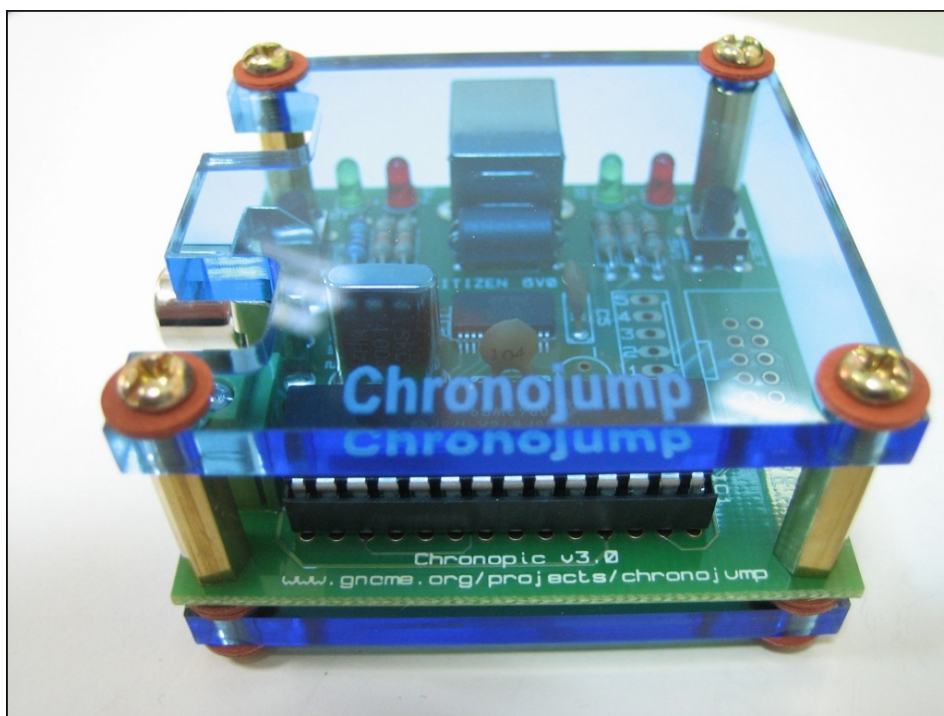


Figura 5.5: Chronopic 3. Fabricación en Taiwán.



para Linux, y su código fuente se publicó también en un subdirectorio del GIT de Chronojump (Chronojump, n.d.).

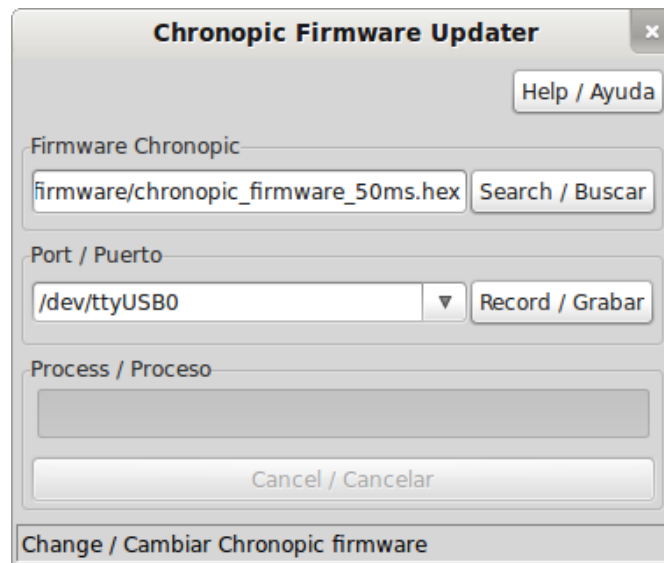


Figura 5.6: Software Chronopic-Firmwarecord.

La figura 5.7 ilustra los resultados de la validación de Chronopic 3. La media de errores absolutos en todas las frecuencias fue 0,13 % en tiempo de contacto y 0,14 % en tiempo de vuelo, mientras que calculando únicamente las frecuencias en el intervalo de saltos (150-700 ms), la media de errores absolutos fue de 0,1 % y 0,18 % para tiempo de contacto y tiempo de vuelo, respectivamente.

En cuanto a la fiabilidad, los valores obtenidos para una misma frecuencia fueron siempre los mismos, exceptuando en las dos zonas de inestabilidades, que se encontraron en los siguientes rangos:

- Zona cercana a 50 ms: 53,248 ms - 50,251 ms.
- Zona cercana a 25 ms: 25 ms - 24,752 ms.

En estas zonas se perdía la fiabilidad debido a que se producían resultados distintos para una misma frecuencia de onda de entrada. Los resultados obtenidos eran múltiplos de 50 en la zona cercana a 50 ms y múltiplos de 25 en la otra zona.

Una posible interpretación de esta variabilidad se vincula a los pulsos espúreos. Cuando un impulso es inferior a 50 ms, éste es despreciado por ser espúreo, y al contrario, es aceptado cuando es igual o superior a 50 ms. Se estima que un único estímulo enviado al Chronopic es aceptado o rechazado sin problemas, pero la onda enviada por el generador era un continuo de impulsos que al parecer interferían entre ellos cuando su frecuencia era cercana a la zona en que se juzga si

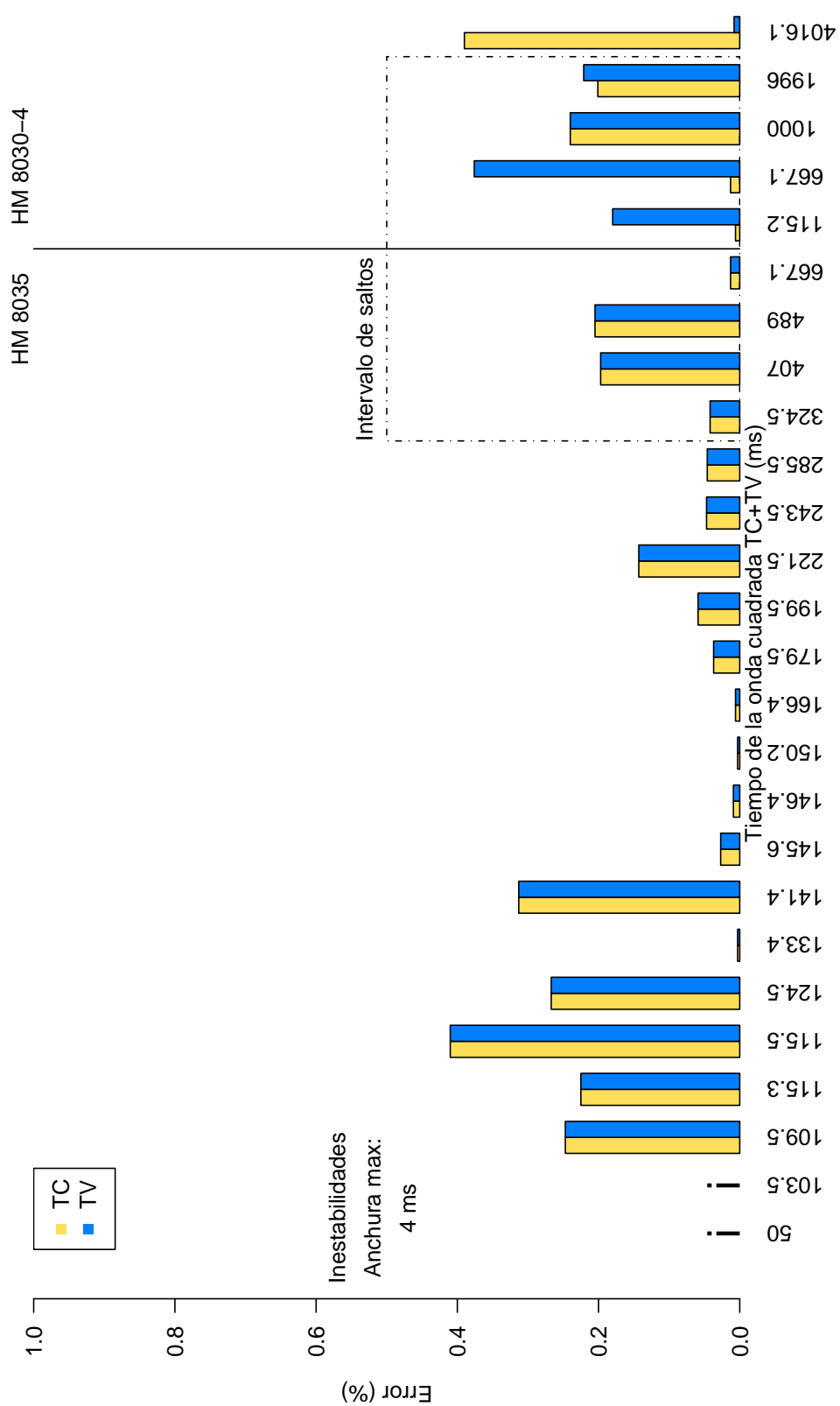


Figura 5.7: Resultados de la validación de Chronopic 3.

son aceptados, o un submúltiplo de ella (zona cercana a 25 ms). En este caso, cabría revisar las interrupciones del firmware para corregir este error.

Aceptando la argumentación previa, este error no es significativo para la situación de saltos, pues el error sólo aparecería cuando se envía una onda continua (con generador de ondas).

En caso que no se acepte la argumentación previa, y por tanto se crea que el error podría presentarse en una sesión de saltos con personas, cabe tener en cuenta que la posibilidad de que aparezca es muy baja por los siguientes motivos:

- Ocurre en un rango muy inferior a los tiempos de salto (tres veces inferior a los mínimos tiempos de contacto).
- La amplitud del rango es muy pequeña: 3 ms.
- El Firmware no se ha modificado desde el año 2005 y ningún usuario se ha quejado de valores erróneos por parte del microcontrolador.

Por otro lado, debido a que un mal contacto podría tener un tiempo comprendido en alguna de las dos zonas encontradas, se recomienda, como precaución, no usar plataformas de varillas en las que éstas se encuentren excesivamente separadas (no usar separación de más de 5 cm) pues podrían ofrecer tiempos cortos erróneos en caso de mala colocación del pie. Otros factores que podrían favorecer los contactos erróneos de poca duración son: saltar descalzos en plataformas de varillas, y ejecución del salto con una técnica incorrecta en la que aparezca un presalto y no sea identificado por el evaluador.

Para la validación del microcontrolador, es ventajoso usar generadores de ondas en lugar de saltadores reales en una plataforma de contactos, puesto que estos últimos no son capaces de realizar  $n$  saltos iguales, y se prescinde además de la variabilidad que podría añadir la plataforma de contactos. Por otro lado, el uso de generadores invita a continuar estudiando si son los causantes de la inestabilidad citada.

Los resultados obtenidos se consideran satisfactorios, y se complementan con los del siguiente apartado, pues en él se validan y fiabilizan plataformas de contactos a partir de estímulos detectados por dos microcontroladores, Chronopic 3 y un sistema Ergojump Boscosystem.

## 6. Desarrollo y validación de una plataforma de contactos

Si tienes un sólo clavo, tiendes a ver a los martillos como un problema.

---

Larry Wall

La plataforma de contactos es el dispositivo de detección de los estados de un salto (individuo en el suelo, individuo en el aire). Cada vez que el participante despegue o aterrice se producirá un cambio de estado en su circuito eléctrico. El microcontrolador validado en el capítulo anterior se encargará de detectar el cambio eléctrico en la plataforma, medir los periodos de duración de los estados, filtrar las señales eléctricas espúreas y enviar la duración de los estados a un ordenador.

### 6.1. Objetivo

El objetivo del presente apartado es el siguiente:

Crear y validar una plataforma de contactos que permita conocer con precisión cuándo un sujeto se encuentra en ella o fuera de ella.

Se entiende que la validación incluye el proceso de fiabilización.

### 6.2. Metodología

La metodología se dividió en dos apartados:<sup>2</sup>

- Se crearon distintas plataformas de contactos.
- Se validaron las plataformas, que a su vez contemplaba diversos aspectos:
  - Se comprobó la presión mínima que se precisaba para realizar contacto en diferentes puntos de la plataforma. Se entiende que una plataforma es más sensible cuando la fuerza necesaria para su activación es menor.
  - Se compararon los tiempos de activación/desactivación de las plataformas propuestas respecto a la plataforma del sistema Ergojump Boscosystem, y entre ellas.

---

<sup>2</sup>Las plataformas han sido diseñadas y validadas con la colaboración de Josep Maria Padullés.

### 6.2.1. Creación de plataformas de contactos

Para la creación de las plataformas de contactos se usaron pares de materiales conductivos separados en algunas zonas por materiales no conductivos. Cuando el peso del sujeto vence la presión de los materiales no conductivos, se produce contacto entre los circuitos debido a la alimentación de un microcontrolador, del mismo modo que sucede en la plataforma de contactos diseñada por Carmelo Bosco (Bosco, Luhtanen y Komi, 1983) descrita por Buscà, De Blas y Daza (2004).

Con el fin de satisfacer las necesidades de distintos entrenadores, se diseñaron plataformas de dos tipos: rígidas y flexibles, y se experimentó con diversos tamaños de plataformas.

En cuanto a las rígidas, se crearon dos modelos, uno de tamaño A2 y otro A3. Se construyeron usando dos placas de fibra de vidrio -una superior y otra inferior- separadas entre sí por cinta de doble cara. Dichas placas tenían una lámina de cobre en cada una de sus caras para garantizar la conductividad. La cinta de doble cara se ubicó a lo largo de los cuatro lados de la plataforma con la finalidad de mantener las dos placas cercanas pero sin que hubiera contacto hasta que no se venciera la resistencia de la misma por el peso del sujeto. Se colocó la cinta de forma intermitente para que la plataforma “respirase” de manera que no se produjera un “vacío” al despegar el individuo; que provocaría que las placas estuvieran en contacto demasiado tiempo y acortaría erróneamente la medición del salto. En el caso de la plataforma A2, se colocó además una pequeña porción de cinta en el centro para evitar que se deforme debido a su mayor tamaño. El cable de conexión con el microcontrolador se soldó a la placa inferior y la conexión se protegió para que no hubiera contacto con la superior. Para prevenir heridas, mejorar la adherencia y embellecer el conjunto, se cubrieron las placas con vinilo, pero sin olvidar que la plataforma debía respirar por los laterales.

La plataforma flexible se realizó usando una espuma de poliéster de 1 cm de grosor. La espuma se perforó con un taladro de 3 cm de diámetro. Se realizaron tantos agujeros como fue posible, sin que los agujeros se tocasen entre ellos. Se recubrió cada una de las caras de la espuma de poliéster por otras dos espumas de 0,4 cm de grosor (una arriba y otra abajo). Estas nuevas espumas estaban plastificadas y tenían una fina capa de aluminio en la cara que entra en contacto con la espuma de poliéster. Las tres capas se mantenían unidas gracias a cinta de pintor. Finalmente se recubrió todo el conjunto con nylon y PVC.

### 6.2.2. Validación por determinación de la presión mínima

Para la primera de las validaciones se usó una galga extensiométrica del sistema MuscleLab (Noruega), el microcontrolador Chronopic 3 y un ordenador personal. El procedimiento se ilustra en la figura 6.1.

1. En primer lugar se realizaron marcas en distintos puntos de las plataformas siguiendo la disposición señalada en la tabla 6.1.
2. Se calibró la galga del *MuscleLab* mediante un peso conocido.
3. A continuación, se conectó la plataforma al microcontrolador Chronopic 3, que estaba conectado a un ordenador para recibir alimentación.
4. Se colocaron pesos de 2 Kg y 1 Kg hasta determinar el mínimo de fuerza (pesos añadidos + peso de la galga) para cerrar el circuito de la plataforma de contactos. Este paso se realizó en cada uno de los puntos marcados en cada plataforma.
5. Finalmente, se realizó un análisis descriptivo consistente en la obtención de la media aritmética y la desviación estándar de la fuerza necesaria para la activación en los distintos puntos de cada plataforma, y se compararon los resultados entre plataformas.

Tabla 6.1: Puntos evaluados en cada plataforma.

Rígida Din-A2					
(0, 0)					(60, 0)
(4, 5)	(17, 5)	(30, 5)	(43, 5)	(56, 5)	
(4, 21)	(17, 21)	(30, 21)	(43, 21)	(56, 21)	
(4, 37)	(17, 37)	(30, 37)	(43, 37)	(56, 37)	
(0, 42)					(60, 42)

Rígida Din-A3 y flexible Din-A3			
(0, 0)			(42, 0)
(5, 5)	(21, 5)	(37, 5)	
(5, 15)	(21, 15)	(37, 15)	
(5, 25)	(21, 25)	(37, 25)	
(0, 30)			(42, 30)

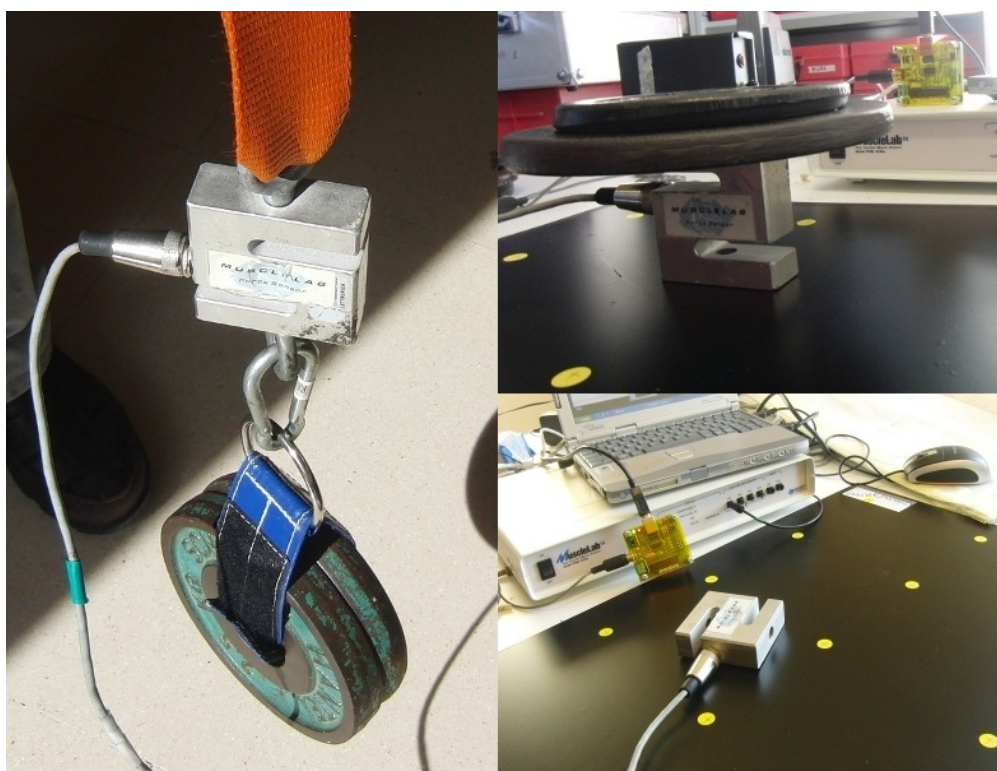


Figura 6.1: Instrumentación para la validación de la sensibilidad de las plataformas. A la izquierda: calibración de la galga extensiométrica mediante un peso conocido. Arriba a la derecha: ubicación de pesos sobre la galga para determinar la presión mínima en cada punto de la plataforma. Abajo a la derecha: instrumentación usada: plataforma de contactos, galga extensiométrica, MuscleLab conectado a ordenador personal con software de MuscleLab.

### 6.2.3. Validación por comparación de los tiempos de activación/desactivación

Este experimento precisó de los siguientes instrumentos:

- Dos ordenadores personales.
- Dos microcontroladores *Chronopic 3*.
- Una plataforma de contactos *Ergojump Boscosystem* doblada para que la distancia entre las varillas fuera de 5 cm.
- Tres de las plataformas de contactos creadas: dos rígidas en tamaño Din-A2 y una flexible en tamaño Din-A3.

La plataforma de contactos *Ergojump Boscosystem* se usó a modo de Gold Standard.

La disposición de las plataformas se muestra en la figura 6.2. Con la finalidad de que los atletas tengan ambos pies a la misma altura, se colocó un plástico rígido grueso debajo de la plataforma de Bosco. Se estima que este plástico no interfirió en el resultado de los saltos porque era rígido.

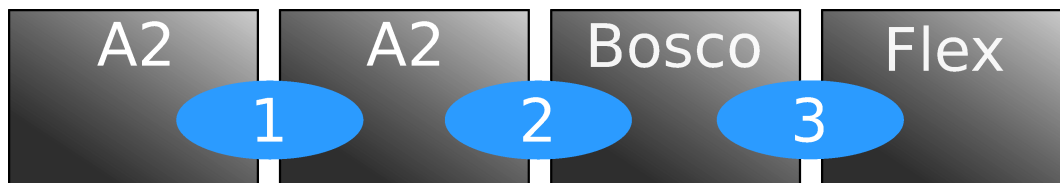


Figura 6.2: Disposición de las plataformas de contacto. Cada uno de los números se corresponde con una situación. Los sujetos pusieron un pie en cada plataforma.

La muestra estaba formada por ocho atletas adultos varones de disciplinas del atletismo relacionadas con la velocidad y el salto. Todos los atletas se encontraban familiarizados con el salto Squat y la valoración del mismo con instrumental específico, pues lo realizaban a menudo. Los atletas entrenaban juntos regularmente. Antes de su participación se les facilitó un documento explicativo sobre el experimento (Apéndice E.2) y se les pidió su consentimiento informado (Apéndice E.1) ya que su participación fue voluntaria. De cada sujeto se obtuvieron las variables: edad, altura y peso. La tabla 6.2 contiene los estadísticos sobre la muestra.

El experimento fue el siguiente:

- Los atletas realizaron un calentamiento libre.
- Se dividieron en dos grupos. Uno de los grupos realizó los saltos pasando por las situaciones 1 a 3 y el otro lo hizo en sentido inverso (3 a 1).
- Se realizaron cuatro saltos en cada situación, dos con el cuerpo orientado hacia una pared y otros dos orientado en sentido contrario.



Tabla 6.2: Muestra para la validación de las plataformas.

	Media	Mediana	Desviación estándar	Recorrido intercuartílico
Edad	24,1	23,2	3,4	2,0
Altura	177,0	177,0	4,6	7,0
Peso	71,1	70,2	5,9	4,5

- El salto realizado fue el Squat submáximo con la finalidad de que no existieran asimetrías en el despegue ni en la recepción (ver comunicación personal con J.M. Padullés en la página 31).
- La consigna previa a cada salto fue: “Cuando quieras, ¡salta!” o, en catalán: “Quan vulguis, salta!” tratando de no alterar el énfasis entre ejecuciones.

El registro se hizo con un microcontrolador Chronopic 3 conectado a cada plataforma y a cada uno de los ordenadores con el software Chronojump.

A continuación, ya sin la presencia de los practicantes, se compararon los resultados de los diferentes saltos para determinar la fiabilidad de las diferentes plataformas de contactos gracias a la herramienta estadística R(R Development Core Team, 2008).

Las pruebas estadísticas que se llevaron a cabo fueron:

1. A partir de una inspección gráfica con diagramas de caja, se estudió la validez concurrente de las plataformas por comparación con el Gold Standard (la plataforma de Bosco). Además, se fiabilizaron las plataformas rígidas tamaño Din-A2 a partir de una comparación entre ellas ante un mismo estímulo (una plataforma en cada pie y comparación de saltos submáximos equilibrados).
2. Se determinó si existían desviaciones laterales a partir de la comparación de los resultados entre las distintas situaciones, con la finalidad de apreciar si unas plataformas tenían un error sistemático que provocase una cierta tendencia a producir tiempos mayores que otras.
3. Se estudió la variabilidad de los distintos atletas en el conjunto de situaciones, y también en cada situación específica.
4. Se prosiguió con gráficos de Bland-Altman para observar las diferencias en cada situación.
5. Finalmente se calcularon los índices de correlación intraclase para cada situación.

### 6.3. Resultados y discusión

A continuación se exponen los resultados obtenidos para la creación y validación de plataformas de contactos.

#### 6.3.1. Creación de la plataforma de contactos de Chronojump-Boscosystem

Se pueden observar las plataformas finalizadas en la figura 6.3.



Figura 6.3: Composición de una fotografía de la plataforma rígida de tamaño A2 y dos de la plataforma flexible.

Las plataformas rígidas diseñadas, aún siendo más pequeñas que otras plataformas de contactos presentes en el mercado, como la plataforma del sistema Ergojump Boscosystem, producen

contacto en toda su superficie. De este modo, un ejecutante que caiga con una mayor parte del pie fuera de la misma, produciría igualmente contacto. En cambio, la plataforma de Ergojump Boscosystem tiene conexiones eléctricas en los laterales impidiendo que se produzca contacto en esa zona.

Por otro lado, la plataforma flexible se ajusta a las necesidades de quienes desean una plataforma muy grande y que se pueda transportar y plegar con facilidad.

La ausencia de varillas elimina los errores de detección prematura de inicio de salto o detección tardía del momento de la caída cuando la punta del pie se encuentra entre varillas.

### 6.3.2. Validación por determinación de la presión mínima

Es necesario conocer la presión mínima necesaria para realizar el contacto en diferentes puntos de la plataforma. La tabla 6.3 muestra las distintas sensibilidades encontradas en las plataformas diseñadas. Por la diferencia en los materiales empleados, la sensibilidad promedio de las plataformas rígidas fue aproximadamente cuatro veces peor que la sensibilidad de la plataforma flexible. La plataforma con peor sensibilidad fue la rígida A3, con un promedio de 83,776 N de fuerza necesaria para su activación.

Tabla 6.3: Sensibilidad de las distintas plataformas.

	Máximo (N)	Promedio (N)	Desviación estándar
Rígida A2	120,4	65,0	31,2
Rígida A3	133,8	83,7	37,3
Flexible	17,8	17,8	0,0

Los valores en Newtons corresponden a la fuerza mínima necesaria para su activación.

Respecto a la sensibilidad en cada uno de los puntos de las plataformas (ver figura 6.4), la plataforma flexible se comportó de un modo muy distinto a las rígidas. Debido a que la plataforma flexible era homogénea a lo largo y ancho de su recorrido, la sensibilidad encontrada fue la misma en todos los puntos analizados. Además, la plataforma flexible es más sensible que el resto. Por otro lado, las plataformas rígidas estaban constituidas por dos placas de fibra de vidrio separadas por cinta de doble cara, siendo esta última la causante de un aumento de la sensibilidad en las esquinas, los laterales, y en el caso de la plataforma rígida A2 en el centro. Las breves diferencias entre las plataformas rígidas pueden ser explicadas por la disposición de los puntos de evaluación, ya que tal y como se ha visto en la tabla 6.1, en la plataforma de tamaño DIN-A2 se llegó a evaluar 1 cm más cerca de los laterales que en la plataforma de tamaño DIN-A3.

El valor más elevado de fuerza requerida se encontró en una de las esquinas de la plataforma rígida A3 (133,8 N). Este valor de fuerza corresponde a la que ejerce un niño de 13,6 Kg de

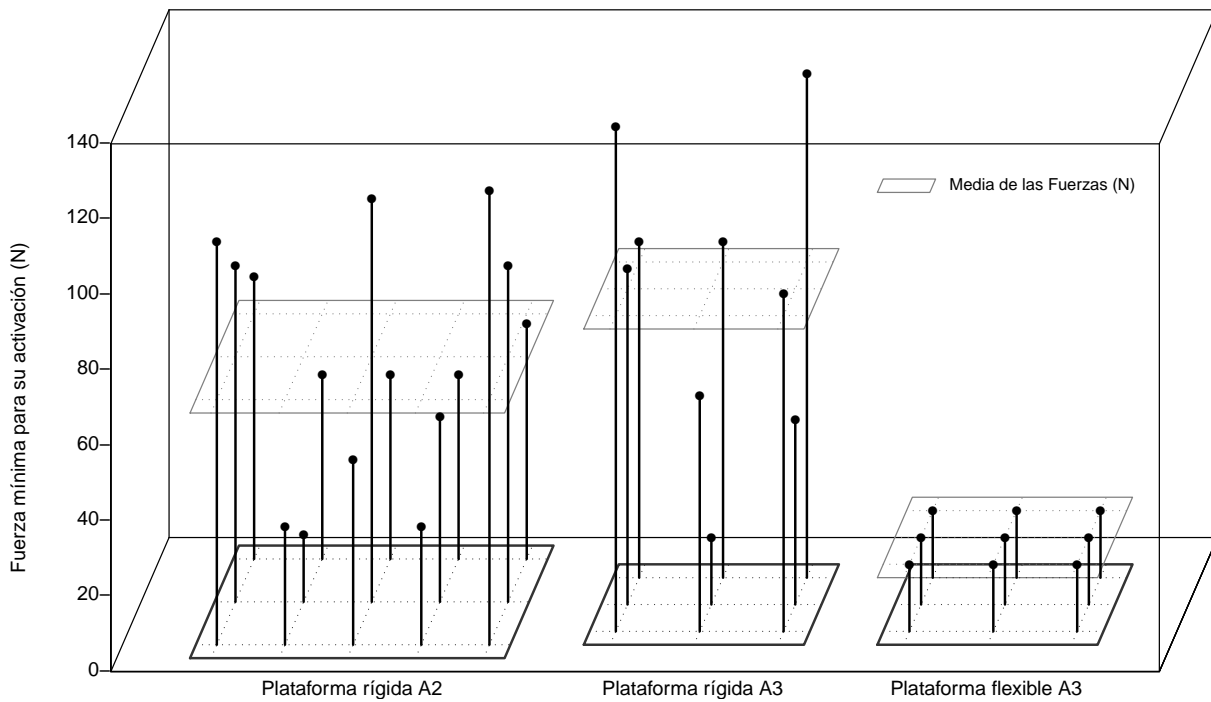


Figura 6.4: Sensibilidad de las plataformas diseñadas.

pie en aquel punto de la plataforma. Aunque el pie de este niño es mayor que la superficie de la galga usada, la zona metatarsial no lo es, de modo que sin realizar ningún movimiento este niño ya activaría la plataforma. Por tanto se concluye que la sensibilidad de las plataformas diseñadas es suficiente para detectar el despegue y la caída de un individuo, y más teniendo en cuenta que en el momento del despegue, así como en la caída, se producen fuerzas muy superiores a la de la gravedad en bipedestación.

### 6.3.3. Validación por comparación de los tiempos de activación/desactivación

En este experimento las diferencias encontradas entre las plataformas correspondían a diferencias en las mediciones por parte de los microcontroladores y de las plataformas, así como asimetrías en los saltos por parte de sus saltadores. Aunque los resultados obtenidos eran consecuencia de los tres agentes citados, el análisis estadístico incide en las diferencias entre plataformas.

El diagrama de cajas de la figura 6.5 explica la fiabilización así como la validez concurrente de las plataformas. Las distintas cajas muestran el porcentaje de diferencia entre el valor de una y otra plataforma en cada salto. Dicho porcentaje se calculó según la fórmula:

$$Error\ relativo\ (\%) = 100 * |TV_{exp} - TV_{gold}| / TV_{gold}$$

siendo  $TV_{exp}$  el tiempo de vuelo encontrado en la plataforma de contactos diseñada o experimental, y  $TV_{gold}$  el tiempo de vuelo correspondiente al Gold Standard.

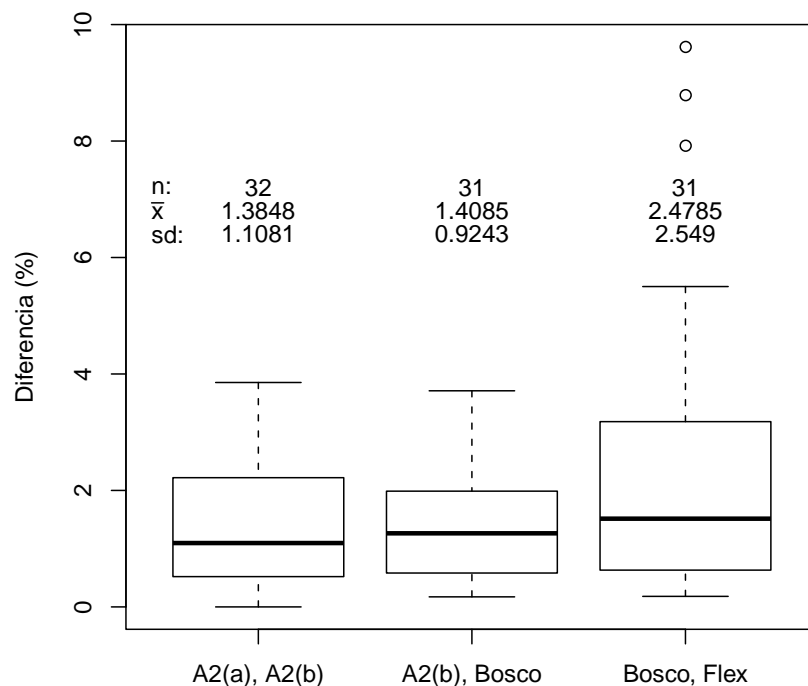


Figura 6.5: Validez y fiabilización de las plataformas propuestas. El sujeto salta con un pie en cada plataforma.

La primera caja se refiere a los saltos realizados en dos plataformas rígidas A2, denominadas A2(a) y A2(b). Como se aprecia, el promedio de las diferencias fue de 1,38 % y la desviación de 1,11. Estos valores se consideraron óptimos y proporcionaron una buena medida de fiabilización.

La segunda y tercera caja corresponden a las diferencias entre las plataformas propuestas y el Gold Standard, esta medida representa la validez concurrente. Tal y como se aprecia, la comparación entre plataforma rígida y el Gold Standard ofreció valores óptimos y muy similares a los de la comparación consigo misma. En cambio, la plataforma flexible no mostraba resultados tan satisfactorios y se identificaron tres errores notables en la medición con valores entre 8 y 10 % de error.

Se procedió a determinar la ausencia de error estándar. La figura 6.6 muestra que no existía una tendencia “lateral” de error excepto en los tres valores alejados de la situación en que se comparó la plataforma flexible con el Gold Standard. En los tres casos el valor de la plataforma flexible fue superior al de la plataforma de referencia. Por tanto, los errores se debían a una anticipación en la detección del despegue y/o, a un retardo en la detección de la caída.

En la figura 6.7 se muestra la incidencia del individuo en los distintos saltos. Como se aprecia, la variabilidad entre sujetos era incluso mayor que la variabilidad observada entre plataformas en la figura 6.5. Se observa, además, que los tres errores de detección en la situación en que intervenía la plataforma flexible, correspondían a tres sujetos distintos. La figura 6.8 explica que los atletas se comportaron de igual manera en las distintas combinaciones de plataformas.

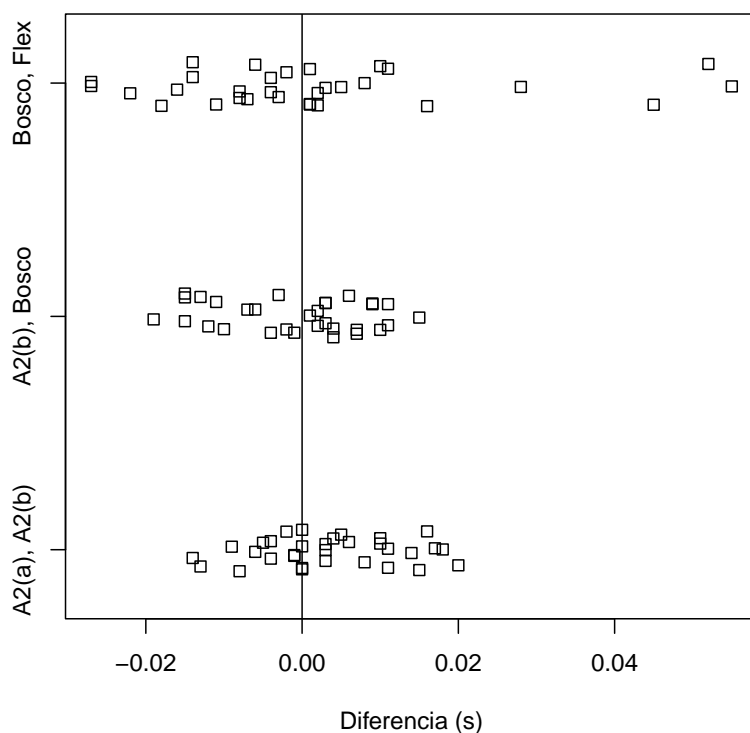


Figura 6.6: Error estándar entre las plataformas. Se altera ligeramente el eje vertical para una mejor visualización.

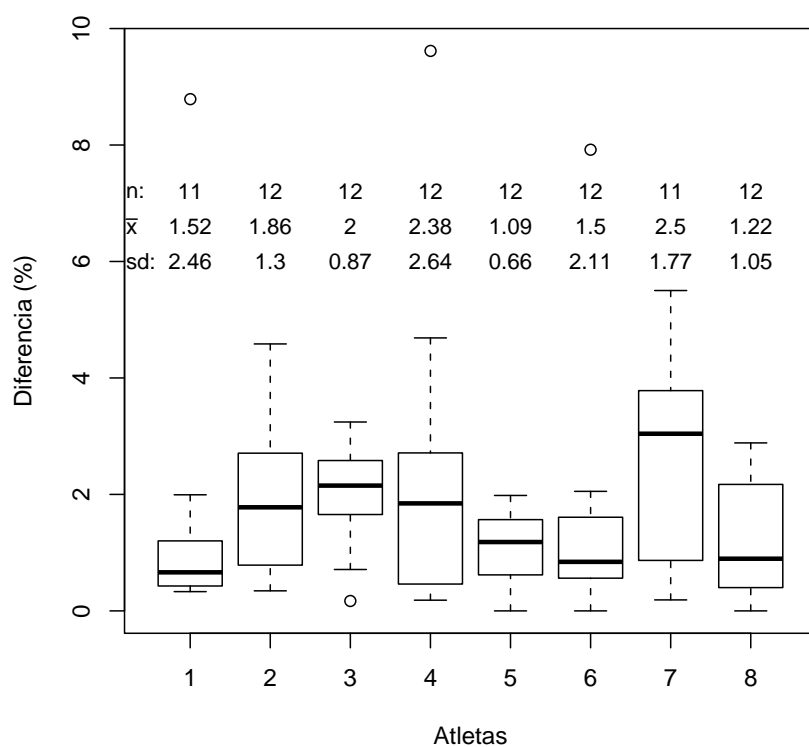


Figura 6.7: Diferencia entre las plataformas en función del atleta.

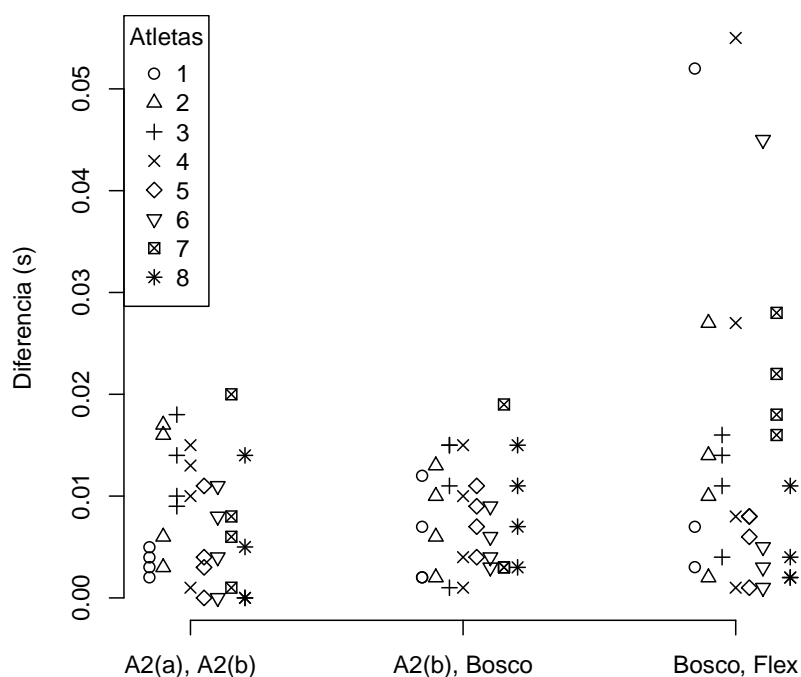


Figura 6.8: Diferencia entre las plataformas en función de situación y atleta.

Continuando con el análisis de las diferencias entre plataformas, se propone el uso de gráficos Bland-Altman (figuras 6.9, 6.10 y 6.11). Se comprueba nuevamente que la situación en que se evaluaba la plataforma flexible fue la única en que aparecieron valores fuera del intervalo de confianza.

En último lugar, la tabla 6.4 muestra los coeficientes de correlación intraclase en las tres situaciones. Todos los resultados se encuentran por encima de 0,81, y, por tanto, reflejan una relación de “agreement” casi perfecta (Landis y Koch, 1977).

Tabla 6.4: Coeficientes de correlación intraclase de las distintas situaciones.

	CCI	CCI 95 % confianza
Rígida A2(a) - Rígida A2(b)	0,949	$0.899 < \text{CCI} < 0.975$
Rígida A2 - Bosco	0,95	$0.898 < \text{CCI} < 0.975$
Flexible - Bosco	0,821	$0.664 < \text{CCI} < 0.909$

El modelo usado es el “oneway”.

A partir de los resultados obtenidos, se concluye que la plataforma rígida A2 queda validada y fiabilizada. La plataforma flexible, aún sin ofrecer malos resultados, deberá perfeccionarse para asegurar resultados deseables en el ámbito científico.

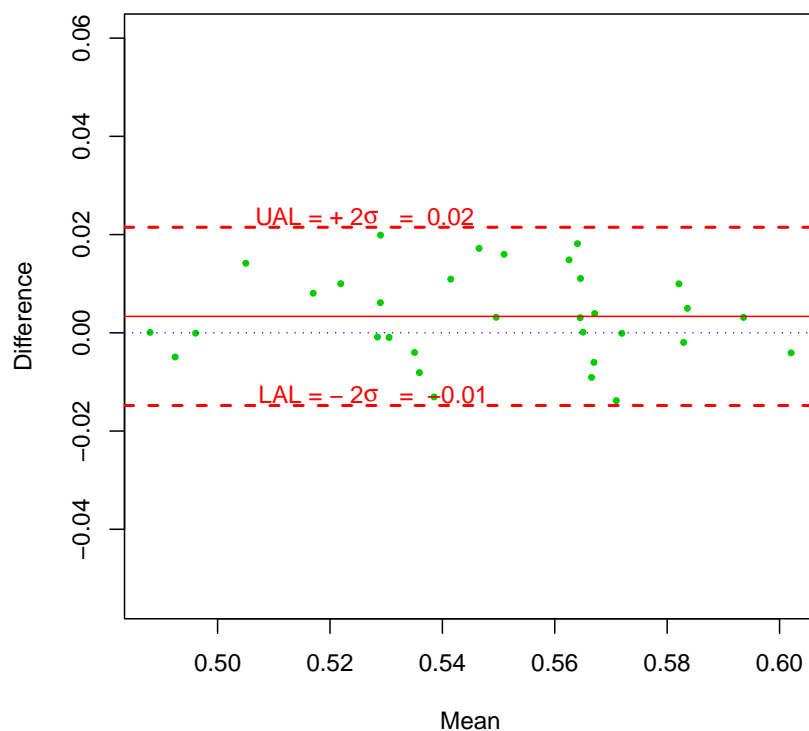


Figura 6.9: Gráfico Bland-Altman entre las dos plataformas rígidas A2. El módulo BlandAltman del paquete ResearchMethods versión 1.2 para R no permite traducir las etiquetas de texto.

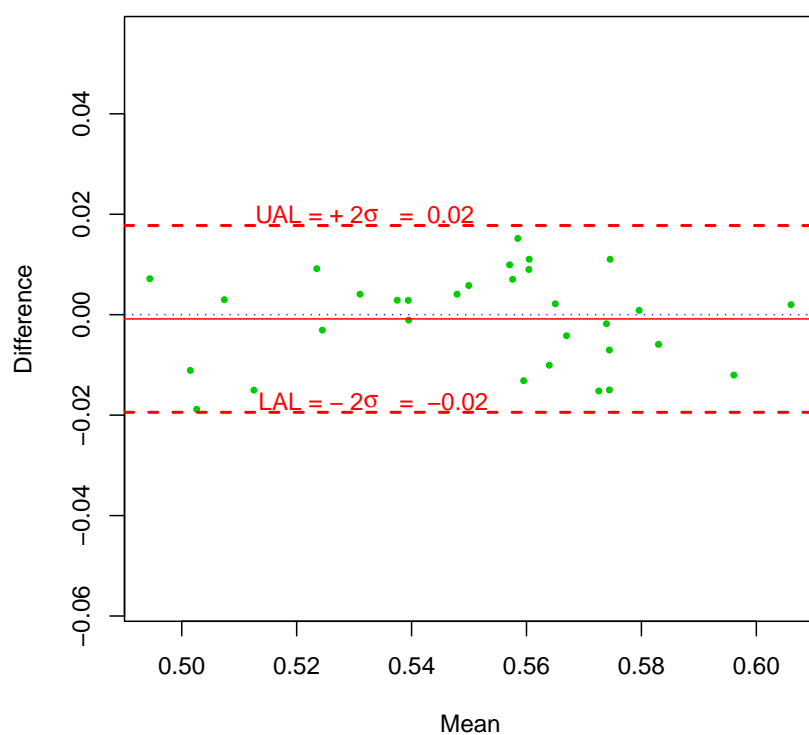


Figura 6.10: Gráfico Bland-Altman entre la plataforma rígida A2 y la plataforma de Bosco.



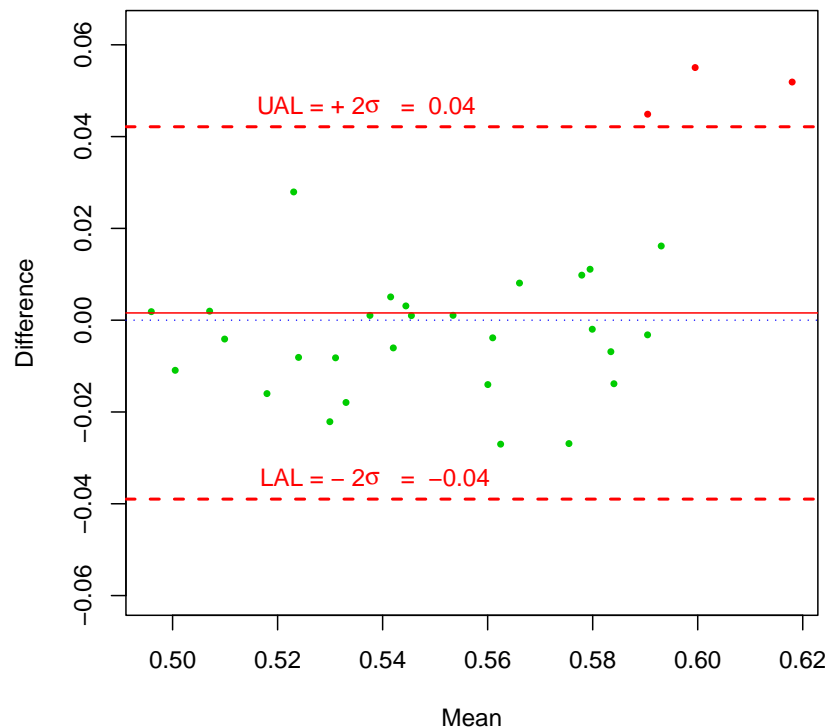


Figura 6.11: Gráfico Bland-Altman entre la plataforma flexible y la plataforma de Bosco.

En este experimento se dejó de lado la plataforma rígida A3, pues se estima que sus valores habían de ser muy similares a la plataforma rígida A2, tal y como se deduce de la prueba de sensibilidad. No obstante, surgió la duda de si era conveniente situar un pequeño trozo de cinta aislante en el centro de la misma tal y como se realizaba en la plataforma rígida A2.

En un principio se creyó que la cinta central no era necesaria en la plataforma A3, pero viendo que la resistencia a vencer por el individuo era igualmente muy leve (en la plataforma rígida A2 eran 120 N), tal vez cabía replantearse su uso pues no representaba ningún inconveniente y podía mejorar la durabilidad de la plataforma.

A partir de los resultados, y no entrando en los objetivos iniciales, se desarrolló un nuevo experimento como continuación del que se ha discutido con la finalidad de determinar si era conveniente ubicar un trozo de cinta aislante en el centro de la plataforma A3.

La muestra la constituyeron seis personas del mismo grupo de entrenamiento que en la anterior muestra de atletas. Cinco de los mismos eran varones. Se siguió el mismo protocolo que el anterior experimento, pero la comparación se estableció entre dos plataformas rígidas A3 y una plataforma rígida A2.

Se prefirió el uso de la plataforma A2 como Gold Standard en lugar de la plataforma de Bosco, porque la primera había producido buenos resultados en el experimento anterior, y porque de este modo la sensación que tendrían los atletas sería de total homogeneidad, ya que percibirían que la superficie de apoyo de cada pie es la misma. En la figura 6.12 se muestra el esquema de la disposición de las plataformas.

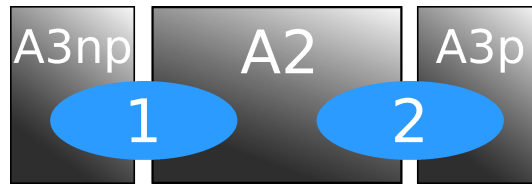


Figura 6.12: Disposición de las plataformas de contacto A3np, A2, A3p. A3p es la plataforma A3 con un punto central de cinta aislante en el centro, mientras que A3np se corresponde con la que no tiene dicho punto.

La figura 6.13 muestra los resultados de este nuevo experimento. Aunque se observa que la plataforma rígida A3 sin cinta aislante en el punto central ofreció unos valores ligeramente mejores que la plataforma con cinta aislante en el punto central, la mejoría era poco significativa y por tanto se recomienda el uso de la cinta en el centro porque podría mejorar la durabilidad de la plataforma.

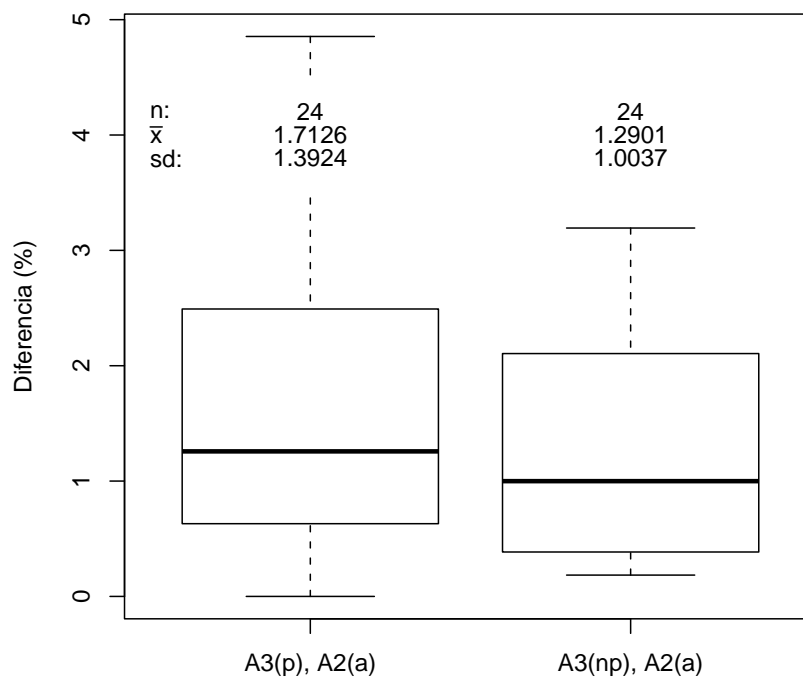


Figura 6.13: Comparación entre los dos diseños de plataformas rígidas A3.

---

## 7. Desarrollo y validación de un software de ordenador

Los hombres productivos usan un IDE profesional. Los hombres de verdad usamos Vim.

---

Miguel de Icaza

El volumen de datos generado en una sesión de saltos por parte de diversos participantes debe ser procesado de algún modo. El presente capítulo describe la creación y validación de un software de gestión que permita comunicarse con el microcontrolador y el usuario.

### 7.1. Objetivo

El objetivo planteado fue el siguiente:

Desarrollar y validar un software libre disponible en los sistemas operativos mayoritarios capaz de:

- Comunicarse con el microcontrolador.
- Gestionar los tests, individuos y sesiones usando una base de datos.
- Generar estadísticos y gráficos de los tests e índices relacionados.
- Crear informes y exportar datos a formatos libres.

El software resultado se denominó Chronojump.

### 7.2. Metodología

La metodología usada siguió los principios de las metodologías ágiles. Debido a que el desarrollo se llevó a cabo en su mayor parte por una sola persona, no se creyó conveniente invertir esfuerzos en preparar tests, y por tanto no se puede hablar de que se usó la programación extrema.

A continuación se describen las iteraciones, historias de usuario, herramientas de desarrollo y los mecanismos de validación.

#### 7.2.1. Iteraciones

La tabla 7.1 recoge las iteraciones más representativas que se han llevado a cabo en el desarrollo del software y los principales objetivos de cada una de ellas.

Tabla 7.1: Iteraciones representativas del software y sus objetivos.

Iteración	Objetivos principales
Salta <sup>a</sup>	Corregir algunos errores del programa Salta creado por Jordi Cebolla.
Gsalta <sup>a</sup>	Añadir una interfaz de usuario moderna.
Consola	Reescribir en C# bajo la plataforma Mono. Comunicar con Chronopic.
Base de datos	Añadir base de datos.
GUI	Añadir interfaz gráfica de usuario.
Estadísticas	Incorporar estadísticas y gráficas.
Tests	Añadir carreras, tiempos de reacción y ritmos.
Usable	Facilitar la instalación y el uso.
Tests - multi	Añadir tests con más de un microcontrolador.
1.0	Completar características, mejorar estabilidad y usabilidad.

<sup>a</sup> Estas iteraciones podrían clasificarse como antecedentes del proyecto, pero se incluyen al ser también realizadas por el doctorando.

### 7.2.2. Historias de usuario

A continuación se incluye una muestra de las historias de usuario de cada una de las iteraciones. Se han redactado las historias con un lenguaje más formal y se han omitido algunos adjetivos calificativos y expresiones relativas a quien las enunció.

#### 1. Iteración “Salta”

Alberto García-Fojeda “Corregir error de gestión del programa Salta.”

Alberto García-Fojeda “Facilitar la inclusión de tests por parte del usuario.”

Jordi Cebolla “Mejorar interfaz de usuario del programa Salta.”

#### 2. Iteración “Gsalta”

Alberto García-Fojeda “Añadir interfaz de usuario moderna.”

Roberto Majadas “Software libre bajo la licencia GPL.”

Deckard: “Hay que sacar una versión Linux del software para que se unan desarrolladores del mundo del software libre.”

#### 3. Iteración “Consola”

Alberto García-Fojeda "El software debería funcionar en Windows y en Linux: multiplataforma."

Alberto García-Fojeda "El programa debería permitir la ejecución del Test de Bosco."

Alberto García-Fojeda "El software debe permitir realizar los saltos presentes en la bibliografía: SJ, CMJ, ABK, DJ con o sin carga extra, RJ limitado por tiempo y por saltos."

Alberto García-Fojeda "Se debe poder crear nuevos saltos de forma gráfica y sencilla."

Alberto García-Fojeda "Programar tests saltos repetitivos de duración indeterminada, y permitir finalizarlos en cualquier momento."

Juan González Gómez "El software debe ser capaz de recoger los datos del microcontrolador."

Juan González Gómez "A parte de programa principal, es necesario crear un ejecutable mínimo que pueda comunicarse con el microcontrolador."

Manuel Lapuente "El software debe estar orientado a analizar conjuntos de sujetos para su comparación."

Jordi Murgó "Recomiendo el uso de Mono porque podrás programar deprisa, tiene multitud de bibliotecas y te soluciona el problema de los usuarios con Windows."

#### 4. Iteración "Base de datos"

Xavi Peña "Que el software permita introducir datos de los miembros de los equipos de forma fácil y rápida."

Miguel de Icaza "El software debería gestionar con una base de datos las tres entidades: salto, sujeto, sesión."

Alberto García-Fojeda "Que permita exportar datos a hojas de cálculo o SPSS."

#### 5. Iteración "GUI"

Alberto García-Fojeda "El software debe tener una interfaz de usuario parecida al ATD con menú y barra de herramientas, siguiendo la tendencia actual de los programas de Windows como el Word."

#### 6. Iteración "Estadísticas"

Alberto García-Fojeda "Se deben calcular índices como el DJ o el de Bosco."

Juan Fernando Pardo: "En latinoamérica se usa el Índice Q para comparar tiempo de contacto y tiempo de vuelo."

Alberto García-Fojeda “Mostrar estadísticas y gráficas de los resultados de los saltos y de los índices: Índice DJ, IE, IUB, Promedio de RJ, ...”

Aldo Tuya “Se podría calcular la Potencia de Bosco en los DJs, pues no se calcula la potencia en ningún salto del programa.”

#### 7. Iteración “Usable”

Bernat Buscà “Posibilitar la visualización de la altura del salto en todas las partes del programa.”

Raúl Bescós “Que la creación de sujetos sea sencilla y rápida.”

Ismael Olea “El software debe estar empaquetado para las distribuciones de Linux más importantes.”

Andoni Morales “Sugiero usar Mkbundle2 para evitar la instalación de Mono, haciendo así la instalación más fácil y rápida y generando un instalador de mucho menor tamaño.”

#### 8. Iteración “Tests”

Alberto García-Fojeda “Añadir tests de carreras simples y con varios tramos.”

Pablo “¿Cómo puedo configurar el Chronojump con las células fotoeléctricas para medir tiempos de carreras de ida y vuelta de 5-5metros, y configurarlo para medir tiempos de carreras en un rombo de 5 metros de lado?”

Michel Marina “En gimnasia saltamos usando los brazos, debería poderse realizar un salto como el DJ con ayuda de los brazos.”

Mercè Mateu “Me gustaría poder comparar la capacidad de seguir un ritmo según la lateralidad o entre extremidad inferior y superior.”

(entrenador-karate) “Quisiera poder medir el tiempo que transcurre desde que aparece un estímulo visual hasta que se termina una patada.”

Josep Maria Padullés “Sería muy útil que se pudiera evaluar el test de Margaria-Kalamen.”

Xavier de Blas “En los saltos reactivos, añadir feedback visual y auditivo para que el ejecutante sepa si los tiempos de cada salto individuo son correctos o no.”

#### 9. Iteración “Tests - multi”

Josep Maria Padullés “Si se pudiera manejar más de un Chronopic a la vez, se podría implementar el análisis de carrera que desarrollo en mi tesis.”

#### 10. Iteración “1.0”

Albert Pujadas, bertu “¿Cuando tendréis versión para Mac?”

Aldo Tuya “Me gustaría poder introducir decimales en los datos de un salto con peso extra o caída desde cierta altura.”

Jeffrey Pagaduan “En Filipinas necesitamos poder usar las unidades métricas.”

Lisandro Mettler “Vuelvo a comentarte mi idea de que la pantalla “Ejecutar salto” podría quedar visible si el evaluador lo quisiera, para mi es muy práctica verla. Me pasa que si al querer ejecutar el mismo salto con otro o el mismo evaluador, esa ventana se minimiza siempre. Visualmente me gusta verla y me da más seguridad de que estoy evaluando al saltador y el salto correspondiente.”

Copev “En muchas partes de Sudamérica usamos la diferencia obtenida entre las pruebas de: Squat Jump y Abalakov; Squat Jump y Rocket Jump; Squat Jump y Countermovement; Squat Jump y Drop Jump; Abalakov y Rocket Jump; Abalakov y Countermovement; Abalakov y Drop Jump; Rocket Jump y Countermovement; Rocket Jump y Drop Jump; Countermovement y Drop Jump. Buscamos con esto obtener el Índice de Reactividad, por eso necesitamos la diferencia en número y en porcentaje.”

### 7.2.3. Herramientas para la creación del software

A lo largo del desarrollo se usaron distintas herramientas de software. Debido a que muchas de ellas fueron reemplazadas por otras más convenientes, se ha incluido únicamente las que se usaron en la última iteración del software. El listado de herramientas se muestra en la tabla 7.2.

Tabla 7.2: Herramientas de software usadas en la creación del programa.

- |   |  |
|---|--|
| ■ Autotools - Scripts de compilación                | ■ Linux - Sistema operativo                      |
| ■ C# - Lenguaje de programación orientado a objetos | ■ Mono - Framework de desarrollo multiplataforma |
| ■ Gettext - Soporte de traducción                   | ■ R - Lenguaje estadístico                       |
| ■ Glade-3 - Generador de interfaces de usuario      | ■ OpenCV - biblioteca de computación visual      |
| ■ Grep - Parseador de cadenas de texto              | ■ Sqlite - Base de datos                         |
| ■ Gstreamer - Framework multimedia                  | ■ Vim - Editor de texto                          |
| ■ GTK# - Herramienta de biblioteca gráfica          | ■ VirtualBox OSE - Software de virtualización    |
| ■ Innosetup - Instalador                            |  |

El seguimiento de fallos se realizó con el Bugzilla de GNOME, y el control de versiones se llevó a cabo inicialmente usando CVS, posteriormente se migró a Subversion (SVN) y finalmente se volvió a migrar a GIT.

La elección de C# y Mono fue debida a los siguientes motivos:

- Interés por aprender a programar en un lenguaje orientado a objetos.
- Sinergia de estos proyectos, ya que el inicio del proyecto coincidió con la popularización de éstos, y programadores informáticos de referencia como Miguel de Icaza o Jordi Murgó los apoyaron (Wikipedia, 2011b; Molist, 2005).
- Disponibilidad de multitud de bibliotecas y facilidad de usar funciones de C.
- Gracias al proyecto Mono se posibilitó la programación multiplataforma en C# convirtiendo uno de los grandes handicaps de este lenguaje (McConnell, 2004) en un signo de distinción.
- Mono proporciona además un entorno de ejecución controlado, facilitando la escritura rápida de software, la ejecución con velocidad y el trabajo con herramientas habituales, como las Autotools (Dumbill y Bornstein, 2004).

### 7.2.4. Validación del software

La principal funcionalidad del software es la comunicación con Chronopic. Esta funcionalidad fue validada y separada del resto del código para que no sufriera modificaciones. Cualquier funcionalidad nueva en el software se probó manualmente antes de subirla al GIT. Además, un grupo de entre tres y cuatro personas se encargó de realizar pruebas antes de la publicación de cada nueva versión. Dichas pruebas iban orientadas a:

- las nuevas características, y
- las funcionalidades básicas.

Los usuarios fueron tratados como testeadores con la finalidad de obtener informes de error útiles.

## 7.3. Resultados y discusión

El software obtenido se denominó Chronojump. Este programa informático tiene, según los servicios web de Ohloh (Black Duck Software, Inc. 2012) a día 8 de marzo de 2012, un tamaño de 109.180 líneas (un 30 % de las cuales son líneas de marcado). Continuando con los servicios de Ohloh, el modelo de cálculo de costes de proyectos COCOMO es de 27 personas-año y el precio total del desarrollo se estima en 1.470.367 dólares. La valoración de COCOMO es tradicional



en los entornos de software empresarial, pero se considera muy sobrevalorada en proyectos de software libre en que las metodologías de desarrollo son más flexibles, tal y como se afirma en la propia página de Ohloh.

La figura 7.1 compara el tamaño de Chronojump con otros proyectos de software libre. Como se puede apreciar, Chronojump es mayor que la mayoría de los proyectos deportivos, pero mucho menor que proyectos de ámbito general, que reciben aportaciones de desarrolladores de ámbitos diversos y que, en general, existen desde hace más tiempo.

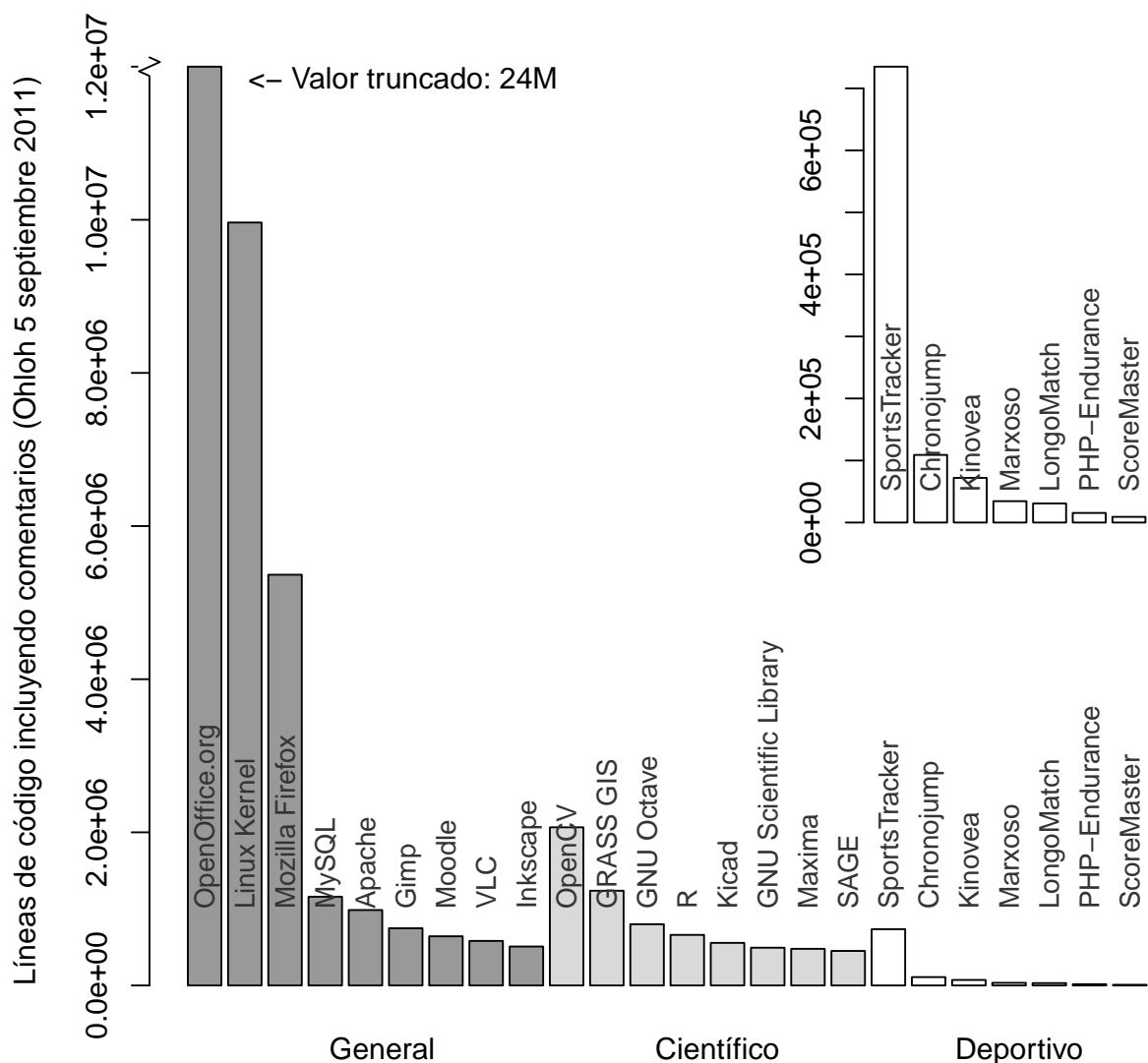


Figura 7.1: Comparación del tamaño de Chronojump con otros proyectos de software libre.

El uso de los distintos lenguajes de programación se observa en la tabla 7.3.

Tabla 7.3: Lenguajes de programación usados en Chronojump.

Lenguaje	Porcentaje de uso
C#	44
XML	25
C	10
R	7
Otros	14

Fuente: <http://www.ohloh.net/p/1005/analyses/latest>

Al ser un programa orientado a objetos, el código se organizó en clases. En el apéndice A.4 se han incluido nueve diagramas de las clases usadas que reflejan la herencia de las mismas.

En la figura 7.2 se ofrece una captura de pantalla de la última versión publicada. La numeración de la versión es la 0.93 y su fecha de publicación fue el 20 de abril de 2011.

En cuanto a la base de datos, los campos de las distintas tablas fueron creciendo tal y como se muestra en la comparativa entre algunas de las tablas más relevantes en febrero de 2005 (figura 7.3) y en marzo de 2011 (figura 7.4). El diagrama de la base de datos completo y actualizado se encuentra en el apéndice A.3.

El software funciona en los sistemas operativos Linux (empaquetado para Fedora y Debian/Ubuntu) y Windows. A corto plazo se espera publicar una versión para MacOSX, pues ésta es una petición de diversos usuarios en los últimos años y la plataforma Mono ha evolucionado lo suficiente como para que la portabilidad a MacOSX sea sencilla. Respecto a las versiones para Windows y Linux, cabe comentar que ambas han sido imprescindibles, la primera porque es la que usan más usuarios, y la segunda porque es la que usan quienes han colaborado más en el desarrollo del software.

Los tests añadidos al software se encuentran recogidos en la tabla 7.4. Además se añadió la funcionalidad para que cada usuario pueda definir sus propios tests de salto o carrera usando las ventanas que se observan en las figuras 7.5, 7.6 y 7.7.

En los saltos repetitivos, se añadieron señales auditivas y visuales en forma de campana verde y roja para producir un feedback de cada salto. De este modo, el entrenador puede indicar que el ejecutante se detenga cuando haya oído la campana roja tres veces. Así, el programa no sólo permite ejecutar un test, sino que también puede promover la alteración de la duración del mismo. En la figura 7.8 se reproduce la ventana de configuración de las señales.

La tabla 7.5 ilustra los estadísticos añadidos referentes al salto. El software muestra promedio y desviación estándar de los distintos resultados, permite la selección de conjuntos de sujetos

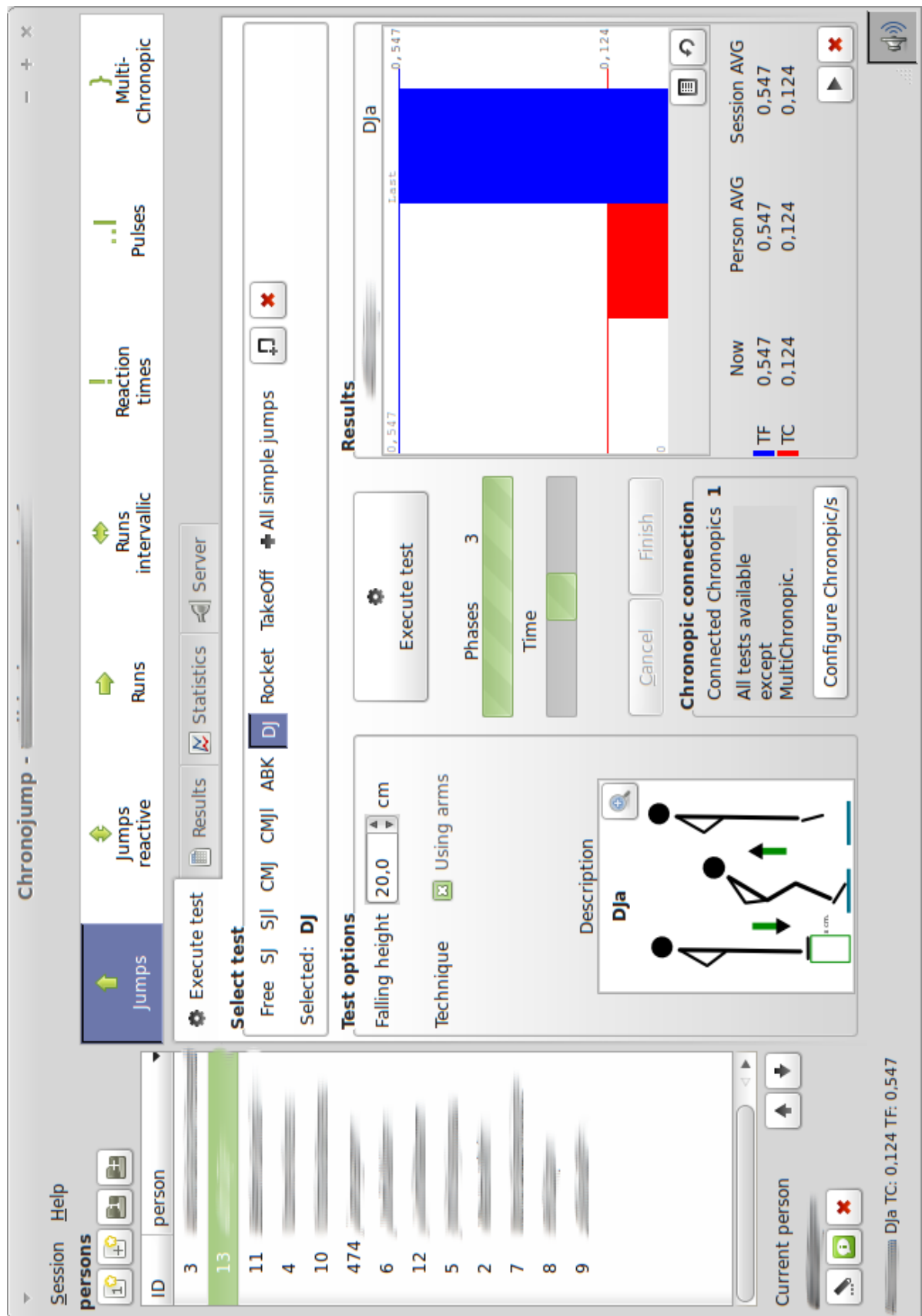


Figura 7.2: Captura de pantalla de la versión 0.9.3 de Chronojump con la interface en lengua inglesa. Se han ocultado los nombres de los atletas.

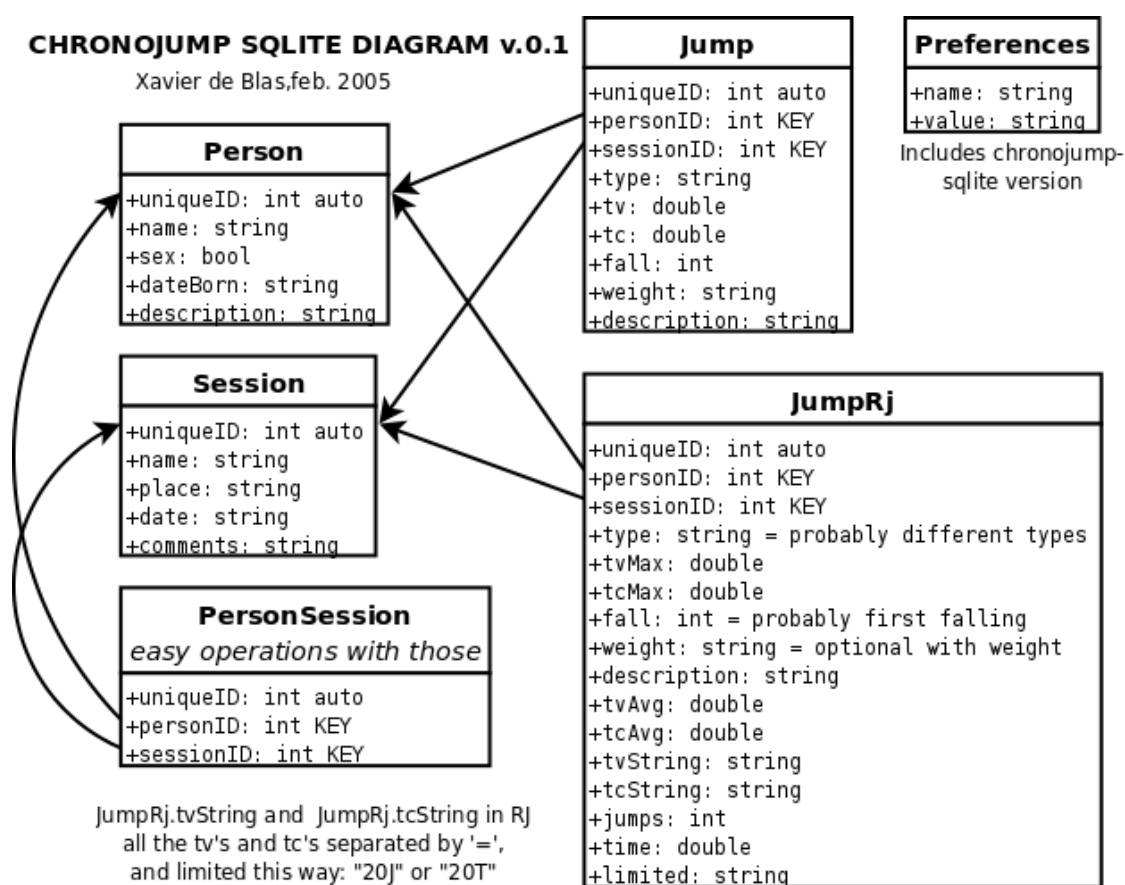


Figura 7.3: Diagrama de la base de datos de Chronojump en febrero de 2005.

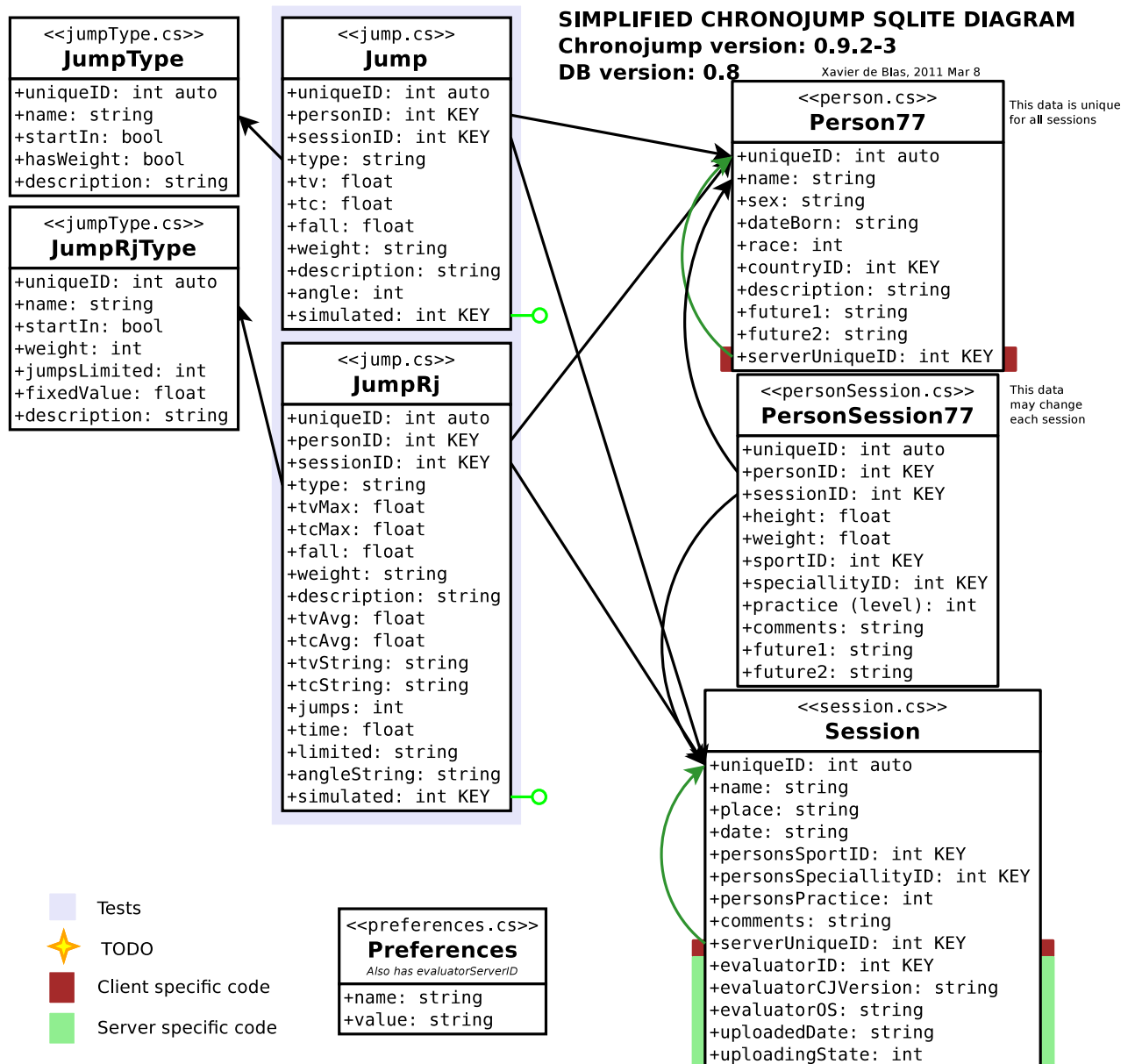


Figura 7.4: Diagrama de una selección de las tablas de la base de datos de Chronojump en marzo de 2011.

**Crear un tipo de salto nuevo**

Crear un tipo de salto nuevo en la base de datos

**Nombre**

Comienza dentro ☒ sí ☐ no

Peso extra ☐ sí ☒ no

Descripción

Figura 7.5: Definir un salto simple en Chronojump.

Tabla 7.4: Tests añadidos en el software.

Saltos		Carreras		Otros
simples	reactivos	simples	con tramos	
Libre	RJ(j)	Personalizada	Por vueltas	Tiempo de reacción
SJ	RJ(t)	20m	Por tiempo	Pulso libre
SJ-I	Ilimitado	100m	Ilimitado	Pulso personalizado
CMJ	Hexágono	200m	20m10veces	MultiChronopic
CMJ-I	Triple salto	400m	7m30segundos	Análisis de carrera
ABK		1000m	20m resistencia	
DJa		2000m	MTGUG	
DJna		Barra de Gesell		
Rocket		20 yardas		
Batida		505		
Batida-I		Illinois		
		Margaria-K		
		Shuttle		
		ZigZag		

**Crear un tipo de salto nuevo**

Crear un tipo de salto nuevo en la base de datos

**Nombre** Triple salto lastrado

Limitado por valor

Opciones principales

☒ saltos

☐ tiempo ☒ fijo 3

☐ ilimitado

Comienza dentro ☐ sí ☒ no

Peso extra ☒ sí ☐ no

Descripción Triple salto en que se define un peso extra en la ejecución

Cancelar Aceptar

Figura 7.6: Definir un salto reactivo en Chronojump.

**Crea un tipo de carrera nueva**

Crea un tipo de carrera nueva en la base de datos

**Nombre** 15s 5-3-3

Limitado por valor

Opciones principales

- ☐ tramos (m)
- ☒ tiempo (s) ☒ fijo 15
- ☐ ilimitado

Distancia

- ☐ Variable (el evaluador seleccionará la distancia en cada prueba)
- ☐ Fija (la distancia siempre será la misma)
- ☒ Diferente (cada pista tiene diferente distancia ideal para pruebas de agilidad)

Número de pistas 3

Distancia de cada pista (m)

5 3 3

Descripción

Carrera de agilidad de 15" de duración en que se propone un circuito de tres pistas (5m, 3m, 3m) hasta finalizar el tiempo.

Cancelar Aceptar

Figura 7.7: Definir una carrera con tramos en Chronojump.

**Condiciones en una prueba**

Seleccionar condiciones para las campanas

**Valores mejores y peores**

- ☒ ☒ y mostrar el mejor TV / TC
- ☒ ☒ y mostrar el peor TV / TC

**Otras condiciones**

Condición	valor	acción
TV >=	0,60	<input type="checkbox"/> <input checked="" type="checkbox"/>
TV <=	0,30	<input type="checkbox"/> <input checked="" type="checkbox"/>
TC >=	0,30	<input type="checkbox"/> <input checked="" type="checkbox"/>
TC <=	0,20	<input type="checkbox"/> <input checked="" type="checkbox"/>
TV / TC >=	2,75	<input type="checkbox"/> <input checked="" type="checkbox"/>

**Probar campanas**

- ☒ probar campana "buena" ☒
- ☐ probar campana "mala" ☒

i prueba!

Cerrar

Figura 7.8: Feedback en Chronojump mediante campanas.

y saltos específicos bajo distintos criterios, y realiza comparaciones intra o inter sesión. Además, incorpora estadísticos referentes a la carrera.

Tabla 7.5: Estadísticos de salto añadidos en el software.

Saltos		
simples sin TC	simples con TC	reactivos
Sin índices	Índice Dj	Índice medio
Fuerza-velocidad	Índice Q	Potencia (Bosco)
Índice de elasticidad		Evolución
Índice de utilización de brazos		RjIndex AVG y SD
Resta entre tests		QIndex AVG y SD
Picos de potencia de Lewis, Harman, Sayers <sup>a</sup> , Shetty, Cannavan, Lara <sup>b</sup>		

<sup>a</sup> Se incluyen dos índices de potencia de Sayers. <sup>b</sup> Se incluyen cinco índices de potencia de Lara.

La creación de gráficos la desempeña GDK en la ejecución de tests, y R en las estadísticas. Así, GDK se usa para gráficos en tiempo real donde se requieren tiempos de respuesta rápidos y reúnen poca información, mientras que R es usado en gráficos que se muestran a petición del usuario y suelen requerir de más complejidad. La figura 7.9 ilustra la conexión con R.

Por último, se programó la generación de informes con los datos, estadísticos y gráficos, que los usuarios seleccionen, así como la exportación de los datos al formato: valores separados por comas (CSV) para su operación desde cualquier hoja de cálculo (Wikipedia, 2011a).

Respecto a la validación, se procedió según lo descrito en la metodología:

La funcionalidad principal del software: “comunicarse con el microcontrolador Chronopic” se codificó en la clase *Chronopic*. Los métodos de dicha clase han sido validados en el experimento de validación de Chronopic descrito en el capítulo 5, y su código se encuentra separado gracias a la encapsulación, y no se ha modificado.

El principal desarrollador realizó pruebas manuales antes de subir nuevas funcionalidades al GIT, y un conjunto de personas participaron en las pruebas de funcionalidades básicas y nuevas características antes de la publicación de cada nueva versión. La tabla 7.6 muestra las personas que han participado en las pruebas en al menos una de las versiones, en cada versión se conseguía que participasen al menos tres de los citados.



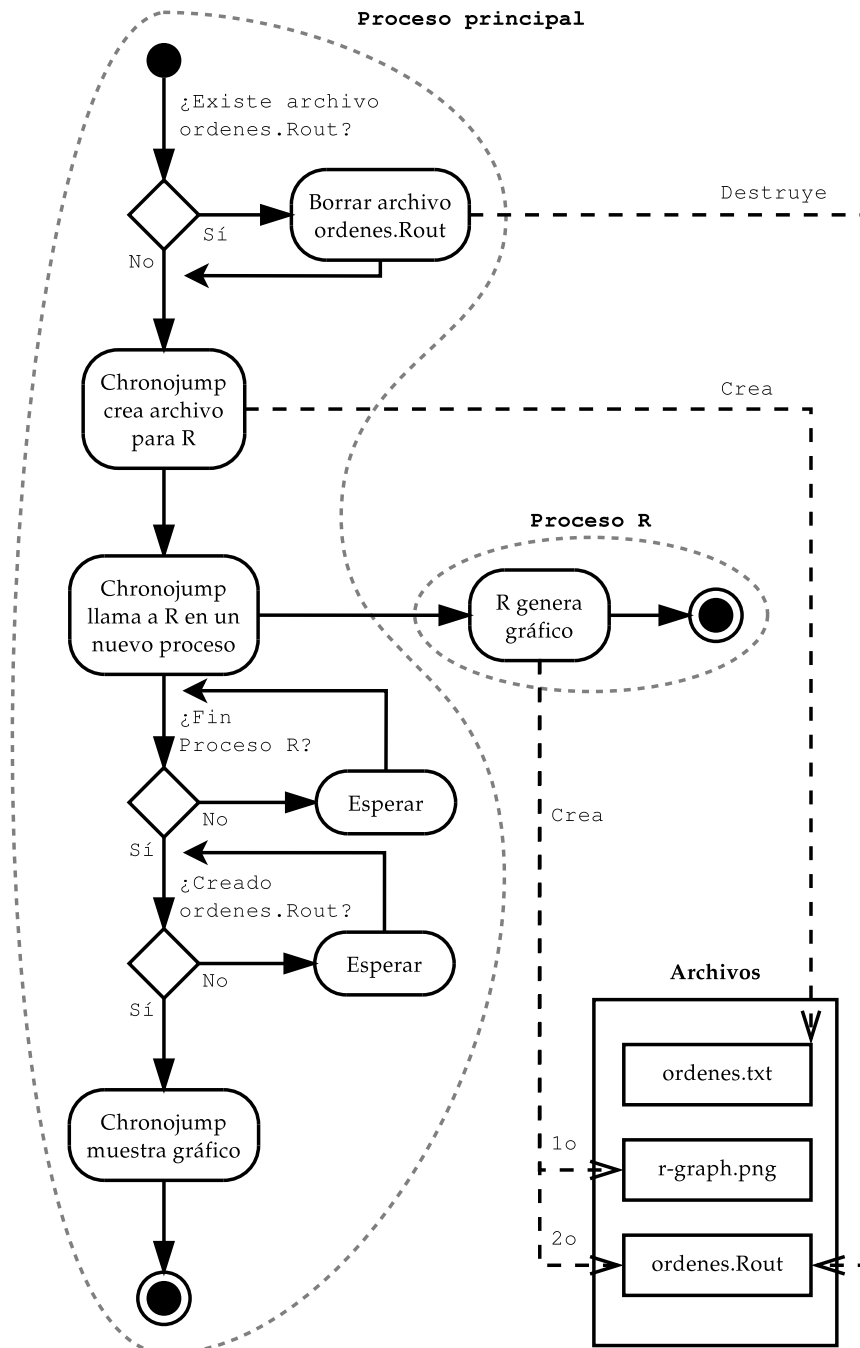


Figura 7.9: Llamada a R desde Chronojump. De Blas, Morales y Olea (2009)

Tabla 7.6: Personas que han participado en las pruebas de Chronojump.

- |                               |                                     |
|-------------------------------|-------------------------------------|
| ■ Albert Pujadas              | ■ Josep Maria Padullés <sup>a</sup> |
| ■ Aldo Tuya                   | ■ Juan Fernando Pardo <sup>a</sup>  |
| ■ Andoni Morales <sup>a</sup> | ■ Lisandro Mettler                  |
| ■ Helena Olsson               | ■ Xavi Peña                         |
| ■ Ismael Olea <sup>a</sup>    |                                     |

<sup>a</sup> Estas personas han participado a lo largo de varios años.

Además, otros usuarios han participado como testeadores proporcionando informes de error útiles tal ya sea por correo privado así como usando los foros dispuestos a tal efecto.

---

## 8. Distribución e internacionalización

Los hombres de verdad no hacen copias de seguridad. Publican sus cosas en servidores FTP públicos, y dejan que el resto del mundo las copie

---

Linus Torvalds

Cualquier creación humana no puede olvidar quién va a servirse de la misma y dónde se encuentra. Por este motivo se debe tener en cuenta las características idiomáticas del usuario y prestar atención a la distribución de programas y aparatos electrónicos.

### 8.1. Objetivos

El objetivo específico de este capítulo es el que sigue:

Facilitar la distribución e internacionalización de las herramientas creadas.

Dicho objetivo se puede alcanzar a través de los siguientes objetivos secundarios:

- Traducir el software al mayor número de idiomas posible.
- Distribuir mundialmente el software y el hardware.
- Mantener una comunidad alrededor de la tecnología descrita.
- Documentar la construcción, el uso y la validación de las herramientas.
- Difundir las herramientas en actos académicos.

### 8.2. Metodología

En primer lugar se incluyó el proyecto de software en el repositorio de proyectos de GNOME para que los GNOME translators pudieran traducir libremente las cadenas de texto del software. El uso de Gettext permitió que los traductores pudieran realizar su función de forma cómoda y el GIT facilitó el control de las versiones.

En segundo lugar, se usó el FTP de GNOME para colgar las versiones del software y el GIT de Chronojump para los planos del hardware. Dicho FTP está replicado en multitud de servidores facilitando las copias de la información y la velocidad de descarga. En cuanto al hardware en sí, se contactó con un distribuidor para que enviara la maquinaria a quienes no estén dispuestos a fabricarse sus propias herramientas.

Se creó un foro en inglés y otro en español para poder responder a las preguntas que tengan los usuarios, así como permitir que se contacten entre ellos; se creó también un Twitter para cada uno de los dos idiomas con la finalidad de anunciar novedades.

Se generaron documentos sobre la construcción, el uso y la validación. Para el primer caso, debido a que las herramientas son libres y están creadas a su vez con herramientas libres, sólo se requirió publicar los archivos GERBER de Chronopic, el código fuente de Chronojump y otros archivos de soporte para que cualquiera pudiera compilar y fabricar estas herramientas en cualquier parte del mundo. Respecto al uso, se colgó un manual de usuario en inglés y en español y se actualizó periódicamente en la medida de lo posible.

Finalmente, se participó en eventos académicos en distintas partes del mundo para dar a conocer las herramientas diseñadas, su validación y sus aplicaciones.

### 8.3. Resultados y discusión

Este apartado se divide en tres subapartados<sup>3</sup>:

1. Resultados pertinentes a las herramientas diseñadas.
2. Resultados correspondientes al uso de las herramientas.
3. Presencia en actos académicos y publicaciones.

#### 8.3.1. Resultados pertinentes a las herramientas diseñadas

A día 19 de setiembre de 2011, las cadenas de texto del software son 1447. La tabla 8.1 muestra el estado de la traducción de dichas cadenas a partir de la información publicada en GNOME translators<sup>4</sup>. La organización de voluntarios de traducción de software al catalán, Softcatalà y la suite ofimática OpenOffice.org, asumen que una traducción está madura cuando ésta supera el 80 % de las cadenas y la instalación se encuentra también disponible en dicho idioma. Siguiendo este criterio, y teniendo en cuenta que el inglés es el idioma original, el software se encuentra en siete idiomas.

El código fuente de los programas y los planos de los instrumentos se encuentran públicamente disponibles en el GIT de Chronojump. El manual de usuario se ha publicado en castellano (De Blas, 2004-2011) e inglés (De Blas, 2011). Ambas versiones del manual están escritas en L<sup>A</sup>T<sub>E</sub>X y su código se encuentra en el GIT, mientras que su resultado (PDF) está incluido en el software

---

<sup>3</sup>Esta estructura es distinta a la que invitan los objetivos de este apartado y la metodología. El motivo de esta diferenciación es que los *Resultados correspondientes al uso de las herramientas* reciben información por parte del servidor Chronojump-server -descrito más adelante-, y la presencia en actos académicos y publicaciones debe su existencia a la colaboración no-prevista por parte de otras personas.

<sup>4</sup><http://l10n.gnome.org/module/chronojump/>

Tabla 8.1: Traducciones del software.

Idioma (código)	Estado de la traducción (%)	Idioma (código)	Estado de la traducción (%)
Eslovaco (sl)	99	Danés (da)	40
Alemán (de)	98	Chino de China (zh_CN)	39
Español (es)	99	Dzongkha (dz)	25
Francés (fr)	98	Portugués (pt)	23
Checo (cs)	97	Finlandés (fi)	22
Catalán (ca)	91	Vietnamita (vi)	20
Inglés británico (en_GB)	73	Sueco (sv)	10
Portugués de Brasil (pt_BR)	60	Occitano (oc)	9
Gallego (gl)	57	Noruego bokmål (nb)	3
		Árabe (ar)	2

El inglés no está incluido pues es el idioma original y, por lo tanto, está siempre traducido al 100 %.

y se puede descargar desde la página Web. En cuanto a las validaciones, se publicó *Validity of new Chronopic V.3 Open Hardware to Measure Time on Jump related Tests* (De Blas, González-Gómez y Gómez, 2009), y se espera publicar esta tesis doctoral así como algunos artículos en lengua inglesa derivados de la misma tesis.

El FTP de GNOME y el GIT de Chronojump han facilitado la publicación de versiones del software para Linux y para Windows, así como para colgar documentos específicos del hardware. En cuanto a la distribución del hardware, se ha configurado un entorno Google Apps y se ha asignado a una sola persona las tareas de construcción, contacto con clientes y distribución.

El software MyBB, gestiona los foros convirtiéndolos en una herramienta más para la organización de la información. Además, se complementan con un par de cuentas de Twitter para la divulgación rápida de noticias. En cuanto a su actividad, el foro en castellano goza de una comunidad más provechosa que el anglosajón, pero ninguno de los dos se nutre de comentarios diarios que aporten riqueza por parte de la comunidad. Los motivos de esta poca actividad parecen ser una combinación de los siguientes:

- Los usuarios están acostumbrados a preguntar por teléfono o por correo, y resulta difícil convencerlos para que usen otro canal.
- Los usuarios tienen dificultades para usar los foros.

- Los usuarios no quieren compartir sus informaciones o dudas de forma pública.
- El volumen de usuarios no es el suficiente como para que se genere el deseable volumen de comentarios en el foro.

No es objetivo de esta investigación el determinar cuál o cuáles de los motivos anteriores son más relevantes. Simplemente se dejará pasar el tiempo tratando de mejorar la accesibilidad del foro, al tiempo que aumenta la cantidad de usuarios acostumbrados a compartir la información en la red.

El software ganó el concurso internacional Trophées du Libre en el año 2007 en la categoría de Educación. El premio consistió en 3000 euros y un ordenador portátil de gama media. El dinero sirvió para financiar el diseño del microcontrolador Chronopic 3 y el código base del software KneeAngle que se explica en el capítulo 9.

### 8.3.2. Resultados correspondientes al uso de las herramientas

En cuanto al uso del software, el servicio web UploadPing de Chronojump-Server (ver página 165) permite obtener unos datos sobre el uso mínimo del software. Se ha usado el vocablo *mínimo* debido a que muchas conexiones no se detectan tal y como se explica en la citada página. Desde 1 de marzo de 2010 a 6 de marzo de 2012 se han identificado 1085 nuevas instalaciones del software, que corresponde a 1,3 instalaciones diarias. La tabla 8.2 muestra una distribución de frecuencias de las ejecuciones del software en el mismo periodo.

Tabla 8.2: Ejecuciones del software.

Ejecuciones	Frecuencia	Ejecuciones	Frecuencia
1-10	914	61-70	3
11-20	102	71-80	2
21-30	29	81-90	1
31-40	17	91-100	1
41-50	7	101-110	1
51-60	6	111-120	2

No se registran las ejecuciones del software por parte del autor del programa.

Las figuras 8.1 y 8.2 muestran la versiones usadas y el sistema operativo del usuario a lo largo de todo el periodo en que el servidor ha estado disponible.

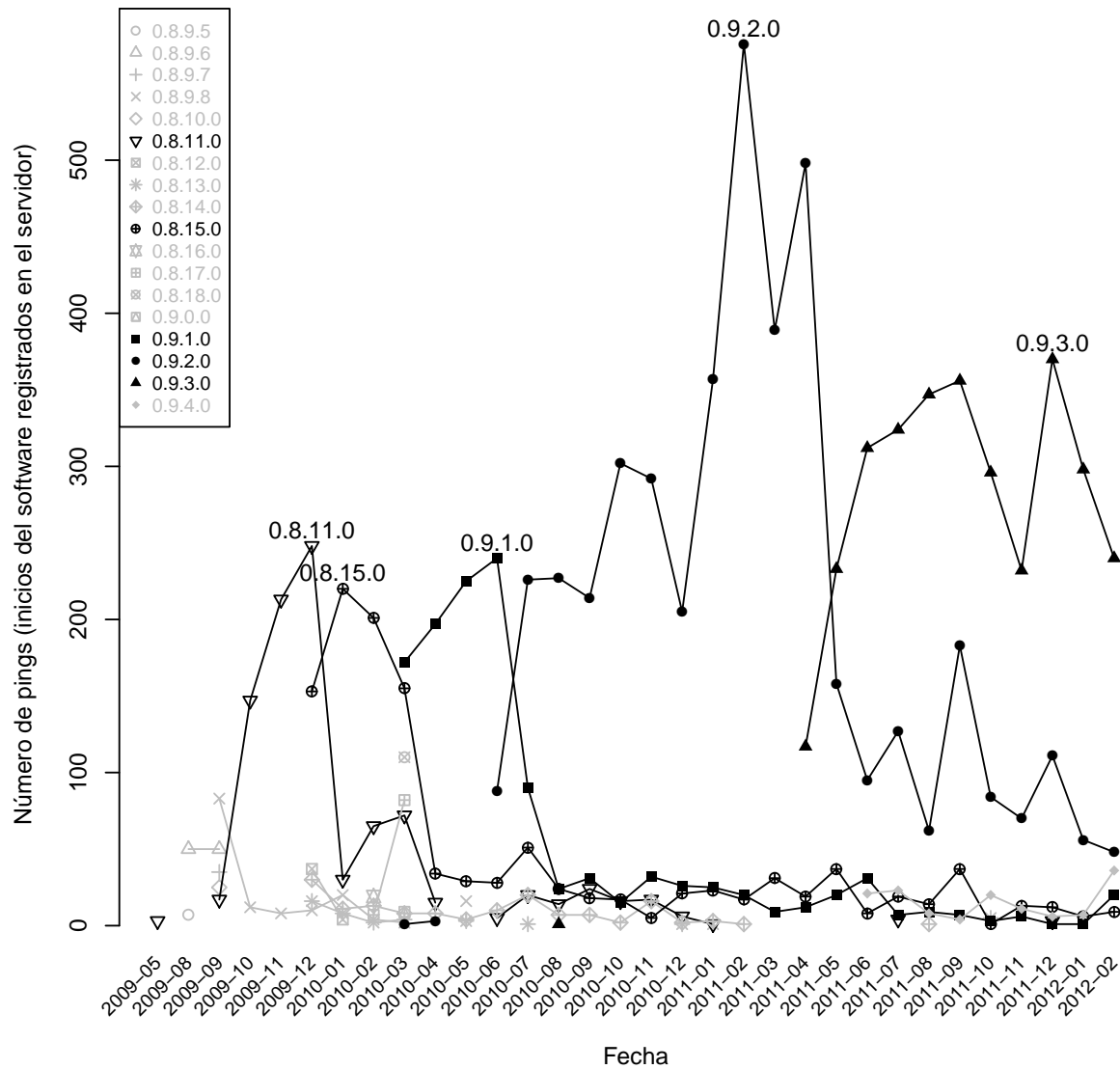


Figura 8.1: Pings según las versiones instaladas del software Chronojump desde la activación del servidor. Se muestran en negro las versiones que en algún mes han alcanzado los 200 pings.

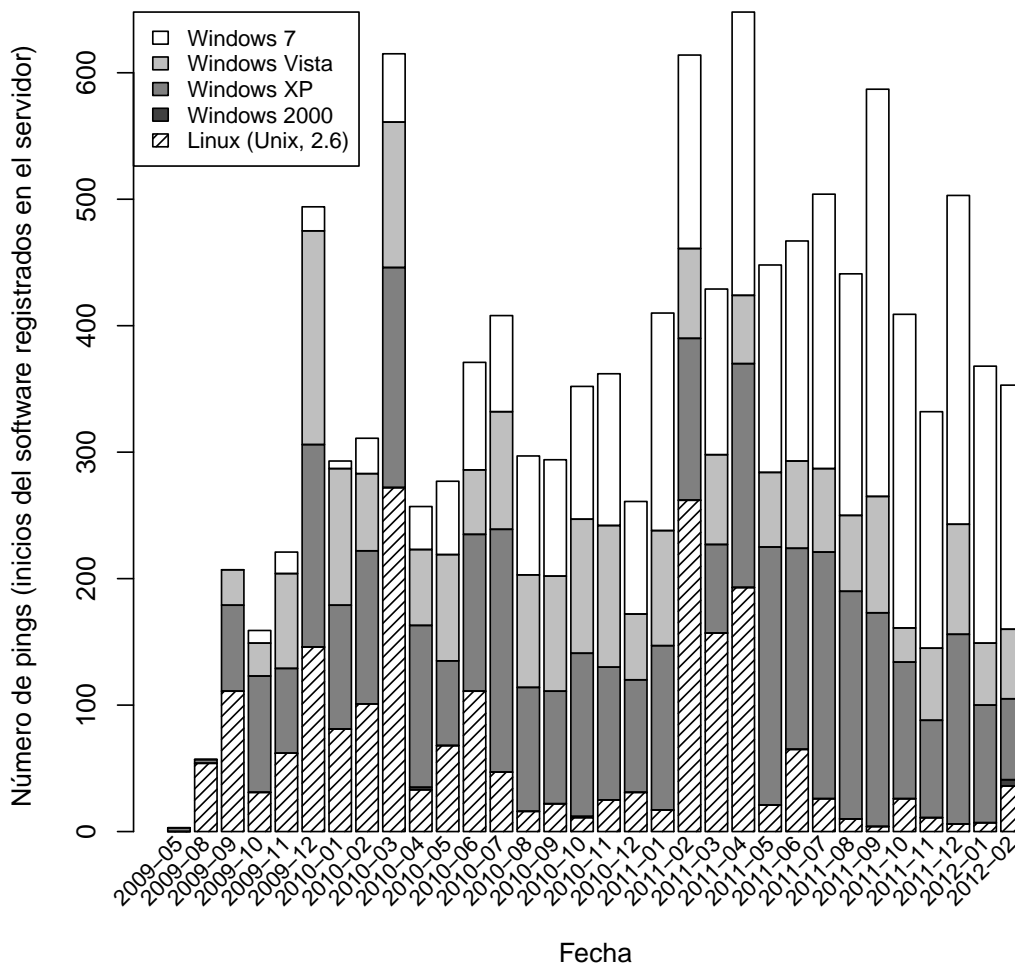


Figura 8.2: Pings según las versiones de los sistemas operativos sobre los que se ejecuta Chrono-jump desde la activación del servidor.



En cuanto al hardware Chronopic 3, necesario para detectar tests con Chronojump, se han vendido 241 unidades en el periodo febrero 2010 - febrero 2012, lo que corresponde a un microcontrolador cada 3 días. El precio de venta al público es de 43€. Cabe comentar que este registro no contempla a los usuarios que se fabrican su propio Chronopic o las posibles ventas por parte de algún fabricante desconocido.

En el mismo periodo: febrero 2010 - febrero 2012, se han vendido 144 plataformas rígidas. El precio de venta al público es de 75€ en tamaño DIN A2, 55€ en tamaño DIN\_A3, y 45€ en tamaño DIN A4.

Por países, los pedidos realizados a la tienda virtual corresponden en un 80.3 % a clientes españoles, mientras que el resto se reparte en 23 países. Este dato constata la importancia del contacto real entre las personas que promueven la tecnología y sus usuarios como motor de adopción de las herramientas por parte de estos últimos.

Además de las ventas propias, dos distribuidores han iniciado la venta de los productos. Las figuras 8.3 y 8.4 corresponden a la distribución desde Chile y Holanda, aunque en ambos casos se distribuye a cualquier país. Los distribuidores fabrican sus propias plataformas de contacto y las envían junto con Chronopic y un CD de Chronojump. El distribuidor holandés añade un curso de formación y denomina al conjunto: Power Tester.



http://www.healthperformance.cl/productos/evaluacion.html

Destacado 2011

Plataforma de saltos con Chip Chronojump

PLATAFORMA DE 3 PLACAS 80 X 100 CM	PLATAFORMA DE 2 PLACAS 80 X 60 CM	EXTENSIONES DE PLACAS INDEPENDIENTES 30 X 40CM
Incluye:	Incluye:	Incluye:
- Plataforma con sistema de contacto de 3 placas, c/ superficie lavable	- Plataforma con sistema de contacto de 3 placas, c/ superficie lavable	- sistema de una placa, c/ superficie lavable
- ChronoChip	- ChronoChip	- Extension RCA de 1 mt
- Extension RCA de 1 mt	- Extension RCA de 1 mt	- Software de calculo de saltos
- Software de calculo de saltos Chronojump	- Software de calculo de saltos Chronojump	- Bolso de transporte
- Bolso de transporte	- Bolso de transporte	- Cd con documentos ( manuales de uso )
- Cd con documentos ( manuales de uso )	- Cd con documentos ( manuales de uso )	

Figura 8.3: Distribución en Chile por parte de terceros.

## Power Tester software en cursus

<http://www.galileo2000.nl/>

De Power Tester wordt gekoppeld aan uw computer. De software wordt meegeleverd.

- U kunt kiezen voor 1 of meerdere personen in een sessie.
- U kunt kiezen uit 44 tests.
- Klik op de toets en de meting start.
- De metingen zijn nauwkeurig op 1/1000 seconde.
- Op de volgende pagina ...

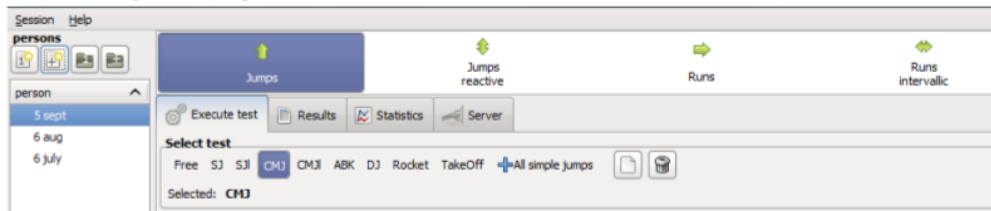


Figura 8.4: Distribución en Holanda por parte de terceros.

### 8.3.3. Presencia en actos académicos y publicaciones.

A continuación se listan publicaciones y presentaciones en torno a las herramientas creadas. En primer lugar se incluyen las desarrolladas por parte del doctorando, seguidamente por parte de otros autores, y finalmente se hace referencia a jornadas y cursos relacionados. Se espera que en los foros de Chronojump aparezcan más referencias en un futuro.

#### 1. Publicaciones y presentaciones por parte del doctorando<sup>5</sup>:

- Buscà, De Blas y Daza (2004). *Instrucciones para la construcción de una plataforma de contactos para la medición de la capacidad de salto/s*. Recuperado de [http://chronojump.org/construction\\_contact\\_platform\\_es.html](http://chronojump.org/construction_contact_platform_es.html).
- De Blas (2005) *Primeres aproximacions a la creació d'un mètode de mesura, gestió i estadística de la capacitat de salt a partir de l'avaluació de temps de vol i temps de contacte a una plataforma de contactes amb programari i maquinari lliure. Projecte ChronoJump*. Revista Aloma. Revista de Psicologia i Ciències de l'Educació.
- De Blas y González-Gómez (2005) Proyecto Chronojump: Sistema de Medida de la Capacidad de Salto usando Software y Hardware Libres. En: *Actas I Congreso de Tecnologías del Software Libre*. A Coruña.
- De Blas (2006) Creation of a free computer science instrument for the measurement of temporary events from contact platforms named Chronojump, and its application in sport sciences. *Session de pôsters del 11th European Congress of Sports Science*. Lausanne (Suiza).

<sup>5</sup>Algunas de estas referencias no se incluyen en la bibliografía debido a que no se consideran relevantes para este documento por su contenido, sino por la marca de actividad que han supuesto a lo largo del proyecto. Las referencias que sí se incluyen son las citadas en alguna otra parte del documento.

- De Blas y García-Fogeda (2006) Aplicações do software e hardware livre Chronojump para a cronometragem de eventos esportivos temporais mediante a plataforma de contato. *I Congresso Internacional de Biodinâmica da UNESP. Rio Claro, (Brasil) marzo 2006.*
- De Blas (2006) *Chronojump: ferramenta de desenvolvimento.* Universidade de Évora (Portugal).
- De Blas (2007) *Chronojump Project.* Les Trophées du libre. Soissons (Francia). Primer premio a "Mejor aplicación libre en el ámbito de la educación" concedido por CETRIL.
- De Blas (2008) Projecte Chronojump. *1a Jornada Inter-Universitària Chronojump.* Facultat d'Educació, UVic. Vic.
- Buscà y De Blas (2008) A new low cost gadget design to measure the weight bar velocity in a bench press using Chronojump. En *13th European Congress of Sports Science.* Storil (Portugal).
- De Blas, González-Gómez y Gómez (2009) Validity of new Chronopic V.3 Open Hardware to Measure Time on Jump related Tests. At *09 Annual meeting of American College of Sports Medicine.* Seattle (USA).
- De Blas (2009) Integración de R en software libre de análisis del rendimiento deportivo multiplataforma y en entorno cliente-servidor. En: *I Conferencia Hispana R-Project.* Murcia.
- De Blas (2009) Compartir dades en la recerca. Servidor Chronojump. *2a Jornada Inter-Universitària Chronojump.* Facultat Blanquerna, URL. Barcelona.
- Buscà y De Blas (2010). Validez de un nuevo dispositivo para medir la velocidad de desplazamiento de un press de banca utilizando Chronojump. En: *Lecturas, Educación Física y Deportes, Revista Digital* 141.
- De Blas (2010) Proposal of a free technological tool to study kinematics of jump: Time measurement, markerless knee angle detection and creation of ready reckoners. Presentación para el Laboratorio de Biomecánica de la NTNU. Taipei (Taiwán).
- De Blas (2010) Presentación técnica sobre desarrollo de aplicaciones libres y Chronojump. En National Applied Research Laboratories, National Center for High-Performance Computing. Hsinchu Headquarters (Taiwán).
- De Blas (2010) Ciclo de conferencias sobre Chronojump en Filipinas en mayo de 2010 incluyendo:
  - Exposición técnica en el laboratorio de ciencias del ejercicio de la Universidad de Filipinas en Diliman.
  - Participación en Seminario de Mejora del Entrenamiento en la Southville International School and Colleges en Paranaque.

- Participación en Seminario de Mejora del Entrenamiento en el National College of Public Administration and Governance de la Univerdad de Filipinas en Diliman.
- Exposición técnica en el National Institute of Physics, de la Universidad de Filipinas en Diliman.
- Presentación para estudiantes de primer curso de grado de ciencias del deporte en el Medical Arts Building de la Universidad de Santo Tomás.
- Entrevista en radio en Strata 2000, Ortigas Center Pasig City.
- De Blas y Gil (2010) Un modelo lineal mixto para la predicción del ángulo de la rodilla en la flexión previa a un salto sin el uso de marcadores. En: *II Conferencia Hispana R-Project*. Oviedo.
- De Blas y Padullés (2010) Sharing a Million Data of Jump and Run Tests. En: *World Congress on Science in Athletics*. Barcelona.
- De Blas (2004-2011) Manual de Chronojump en castellano. Recuperado de [http://chronojump.org/docs/chronojump\\_manual\\_es.pdf](http://chronojump.org/docs/chronojump_manual_es.pdf)
- De Blas (2011) Chronojump Manual (english). Recuperado de [http://chronojump.org/docs/chronojump\\_manual\\_en.pdf](http://chronojump.org/docs/chronojump_manual_en.pdf)
- De Blas (2011) Tecnologies de programari lliure en l'esport. EUSES, Universitat de Girona.
- De Blas (2011) Aplicació de Chronojump a l'atletisme. Universitat de Vic.

Además, el doctorando publicó diversos artículos sobre software libre y presentó la ponencia-espectáculo "Linuxshow" 111 veces en el periodo 2002-2007. Estos actos contribuyeron al recibimiento de apoyo inicial por parte de multitud de especialistas en esta área.

## 2. Publicaciones por parte de otros autores:

- Topend Sports Network (n.d.) publica "Chronojump" como herramienta para conocer el nivel de fitness. El mismo autor se refiere además a Chronojump como herramienta para el cronometrado de carreras.
- Martínez Ramírez y Marín Villada (2008) publican "Perfiles de potencia en miembros inferiores en la modalidad de drop jump en los deportistas preseleccionados por Risaralda a juegos deportivos nacionales 2008". Estudio en que se usa Chronojump para la captación de los datos.
- Wikipedia (2010) dedica el artículo "Chronojump Boscosystem" al proyecto.
- Cardinale (2010) dedica "Low cost vertical jump tests solution" íntegramente a Chronojump.

- Romero y Tous (2010) incluyen una reseña en el libro “Prevención de Lesiones en el Deporte. Claves para un rendimiento deportivo”. Los autores se refieren a Chronojump como material de evaluación de los factores neuromusculares.
- Madrigal y col. (2010) publican “Exercise Science Academic Programs and Research in the Philippines”. En este documento se refieren a Chronojump como herramienta para realizar estudios de fiabilidad.
- Ruddock (2010) publica “Affordable technology: jump to it!” donde comenta con cierto nivel de detalle seis herramientas informáticas de bajo coste vinculadas a la actividad física y el deporte.
- Torres (2011) en su tesis doctoral: “Perfil físico-técnico de jugadores de golf amateurs de élite y su relación con el rendimiento” utilizó Chronojump para valorar la fuerza explosiva de los golfistas a partir del CMJ. Torres-Ronda, Sánchez-Medina y González-Badillo (2011) publicaron un artículo que continuaba con el trabajo de la tesis anteriormente citada.
- Padullés (2011) en su tesis doctoral: “Valoración de los parámetros mecánicos de la carrera. Desarrollo de un nuevo instrumento de medición” cita al proyecto en numerosas ocasiones, realizando distintas validaciones y proponiendo como aportación de la tesis un método de análisis de la carrera con la herramienta Chronojump.
- López, Padullés y Olsson (2011) usan Chronojump para el estudio: “Biomechanical analysis and functional assessment of D. Robles, World Record holder and Olympic Champion in 110m Hurdles”.
- Buscà y Font (2011) en el artículo: “A low-cost contact system to assess load displacement velocity in a resistance training machine ”; validan un diseño de barra de contactos conectada a Chronopic y Chronojump para la obtención de las velocidades de desplazamiento en ejercicios de entrenamiento con pesas. Esta solución resulta mucho más asequible económicamente que un encoder lineal.
- Pagaduan (2011) presentó “Test-Retest Reliability of Countermovement Jump on Free Technology”. Este estudio ha usado Chronojump para la obtención de los datos.
- Álvarez-Herms y col. (2011) publicaron: “Changes in heart rate recovery index after a programme of strength/endurance training in hypoxia” en que usa Chronojump para medir la capacidad anaeróbica láctica a partir del test CMJ 60s.
- Cua y Lim-Martinez (2011) publicaron: “Comparison between Leg Power Output of High School and Collegiate Female Cheerleaders”. Se usó Chronojump para medir la potencia de los sujetos.
- Joson y col. (2011) publicaron: “Relationship of upper and lower body power to swimming kinematics of male collegiate swimmers”. Se usó Chronojump para medir la potencia de los sujetos.

- Tan y col. (2011) publicaron: "Effect of acute blood occlusion training on heart rate variability occlusion and postactivation potentiation". Se usó Chronojump para medir la potencia de los sujetos.
- Gomila-Serra y col. (2011) publicaron "Evaluación de los sistemas de medición de la fuerza muscular en el ámbito de la práctica musical. Aproximación a un prototipo apto para la práctica instrumental". En este estudio se probó la idoneidad de Chronojump para registrar la frecuencia de impactos producidos por un pianista y un guitarrista, y el tiempo transcurrido entre los mismos.
- Pagaduan (2012) presentará en breve "Reliability of Four Series 15-Second Vertical Jump Test Using an Open Source Technology". Este estudio ha usado Chronojump para la obtención de los datos.

### 3. Jornadas y cursos

Se tiene constancia de docencia usando Chronojump en las facultades de ciencias de la actividad física del deporte de Blanquerna, UB (INEFC), Vic, Girona. También se ha difundido información sobre cursos del Comité Olímpico Guatemalteco, así como de la Universidad de Filipinas. Finalmente se adjuntan (figuras 8.5 y 8.6) los carteles de las dos jornadas interuniversitarias específicas sobre Chronojump que se realizaron en Cataluña.

# CHRONO JUMP

**LLOC:** Universitat de VIC    Aula Magna Torre dels Frares    **PREU:** 30€ dinar inclòs

**Amb**    **Dissabte, 19 d'abril**

Carles Romagosa    😎  
Xavi Peña    😊

Bernat Buscà    😊  
Xavier de Blas    😊

Josep Ma Padullés    😎  
José Miguel Vélez    😊  
Raúl Bescós    😊

**UVic**    **Blanquerna**    **INEFC**

9 Acreditacions  
9.30 Presentació  
9.45 Projecte Chronojump  
10.45 Fonaments teòrics en l'avaluació del rendiment esportiu amb plataformes de contactes  
11.45 Aplicació de Chronojump a l'entrenament  
13 Homenatge a **Carmelo Bosco**  
13.30 Dinar  
15 Construcció de dispositius de mesura  
17 Instal·lació, construcció i adquisició de Chronojump i Chronopic  
17.30 Ús de Chronojump  
18 Entrega de certificacions

**INSCRIPCIONS:**    màxim 200 persones per ordre d'inscripció  
[www.gnome.org/projects/chronojump](http://www.gnome.org/projects/chronojump)

**Organitza:** UVIC FACULTAT D'EDUCACIÓ    **Col·laboren:** CHRONO JUMP    Facultat de Psicologia, Ciències de l'Educació i de l'Esport Blanquerna Universitat Ramon Llull    INEFC Institut Nacional d'Educació Física de Catalunya    **Disseny:** Inkscape Gimp

Figura 8.5: Cartel de la 1a Jornada Inter-universitaria de Chronojump en la UVic 2008.

# V Jornada Esport Blanquerna

## II Jornada Interuniversitària CHRONOJUMP

- 9 h Acreditacions.
- 9.30 h Presentació a càrrec del degà, Climent Giné.
- 9.45 h Construcció de dispositius de mesura II.
- 10.15 h Nous tests: anàlisi de la cursa, MTGUG, temps d'execució sota estímul discriminatiu.
- 10.45 h Descans.
- 11 h Compartir dades en la recerca. Servidor Chronojump.
- 11.30 h Avaluació del rendiment esportiu en l'atletisme.
- 12.30 h Repeated-Sprint Ability Team-Sport.
- 13 h Dinar. Premi a la millor funda de Chronopic.
- 14.45 h Presentació de Chronojump 0,9.
- 15.15 h Biomecànica esportiva. Del rendiment a l'escola.
- 16.45 h Experiències del Test de Bosco en l'atletisme.
- 17.15 h Taula rodona moderada per Bernat Buscà.
- 18 h Lliurament de certificacions.

..... Josep Ma Padullés  
 ..... Josep Ma Padullés i  
 Maria Giné

..... Xavier de Blas  
 ..... Miguel Vélez  
 ..... Javier Jorge

..... Xavier de Blas  
 ..... Xavier Aguado  
 ..... J. Luis López del Amo  
 ..... Tots els conferenciantes

**DATA i LLOC:** **9** de maig, FPCEE Blanquerna C/Císter, 34. Barcelona.

**INSCRIPCIONS:** <http://www.chronojump.org> **DATA LÍMIT:** 7 de maig.

**PREU:** Inscripció gratuïta. Dinar 11€.

Cal inscriure's prèviament tant per a la jornada com per al dinar.

**Organitza**  
  
 Facultat de Psicologia, Ciències  
 de l'Educació i de l'Esport Blanquerna  
**Universitat Ramon Llull**

**Col·laboren**  
 **INEFC**  
 Institut Nacional  
 d'Educació Física  
 de Catalunya  
**UVIC**  
 FACULTAT D'EDUCACIÓ

  
**UNIVERSIDAD  
 DE  
 CASTILLA-LA MANCHA**



Disseny: Xavier de Blas  
 GEM/Lluis, inalecsp, dmp

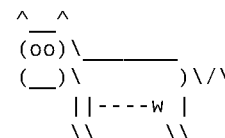


Figura 8.6: Cartel de la 2a Jornada Inter-universitaria de Chronojump en Blanquerna (Universitat Ramon Llull) 2009.



## Parte III

# Estudios de detección sin marcadores del ángulo de flexión de la articulación de la rodilla

El objetivo principal de la presente parte es:

Desarrollar y validar una herramienta libre para medir el ángulo de flexión de la articulación de la rodilla, previo a un salto CMJ, a partir de una filmación bidimensional y sin el uso de marcadores.

La utilidad de esta herramienta es determinar si el protocolo de ejecución de un salto CMJ ha sido el correcto, pues, como se ha dicho en el marco teórico, el ángulo máximo de flexión en el contramovimiento es un componente de la técnica muy difícil de ser evaluado a simple vista. La flexión previa definida en el protocolo del Test de Bosco debe ser de 90 grados y, por tanto, la instrumentación debe ser muy precisa en el rango de movimiento cercano al ángulo recto.

El software de seguimiento obtiene la forma de las piernas recubiertas por mallas negras y, el ángulo de las rodillas a partir de marcadores blancos. Estos datos se usan para el entrenamiento del modelo de predicción, que una vez implementado en el software, determina el ángulo de la articulación de las rodillas en nuevos saltadores únicamente con el uso de mallas negras (sin necesidad de situar marcadores).

Cabe señalar que, aunque se podrían considerar las mallas negras como un marcador de grandes dimensiones, éstas no son llamadas marcadores. Las mallas son muy diferentes en forma y tamaño respecto a los marcadores tradicionales y no requieren de una colocación en puntos anatómicos concretos. Además, las mallas no serán necesarias en un futuro, pues se mejorará el algoritmo para que sea capaz de detectar pantalones cortos y piel. En definitiva, no se ha optado por tratar a las mallas como marcadores, ya que la aportación que se presenta concluirá sin el uso de las mismas, lo que comporta que sea sin marcadores.

Se realizaron también saltos con pantalones cortos sólo para determinar si las mallas afectaban de algún modo la ejecución del salto, por tanto el software debía ser capaz de trabajar sin las mallas. En este caso se usaron marcadores blancos recubiertos por un reborde negro.

El diagrama de flujo de la figura 8.7 explica el proceso seguido:

Capítulo 9: Desarrollo y validación de software de seguimiento.

Capítulo 10: Creación de un modelo de predicción.

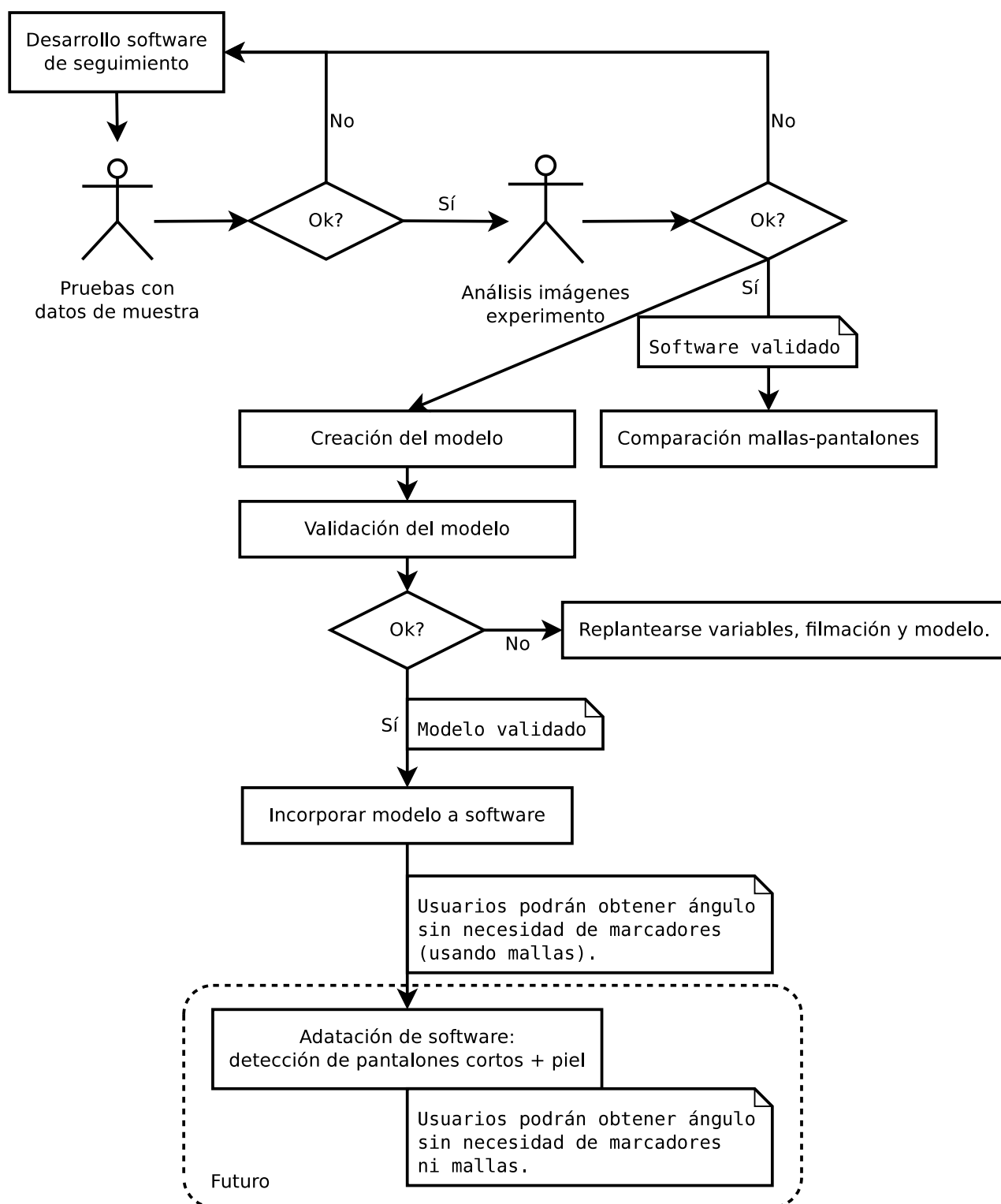


Figura 8.7: Diagrama del proceso de detección del ángulo de la rodilla.

---

## 9. Desarrollo y validación de software de seguimiento

Si miras fijamente la realidad, podrás ver los píxeles.

---

Desconocido

Es imprescindible contar con un software que pueda explorar automáticamente cada una de las imágenes de vídeo de un salto para poder extraer datos numéricos de las mismas. El presente capítulo expone el diseño y validación de dicho programa.

### 9.1. Objetivo

A continuación se formaliza el objetivo:

Desarrollar y validar un software libre capaz de realizar un seguimiento de la silueta de las piernas recubiertas por mallas negras, y realizar un seguimiento de marcadores.

El software resultado se denominó KneeAngle.

### 9.2. Metodología

En el presente desarrollo de software, a diferencia del que se expone en el capítulo 7, no se usaron las Historias de Usuario para determinar los requisitos que se debían cumplir. En el software de seguimiento fue el propio desarrollador quien estableció las especificaciones del programa. Otra diferencia importante fue la ausencia de iteraciones, pues en este caso se perseguía un objetivo final concreto y no era necesario dividir el proceso en etapas que fueran presentadas al usuario.

#### 9.2.1. Especificaciones

A continuación se define el conjunto de requisitos funcionales que debía cumplir el software por orden de ejecución. El programa debía ser capaz de:

- Leer archivos de vídeo de formatos populares.
- Determinar el umbral de luminosidad (threshold) de forma automática.
- Detectar la forma de las piernas usando mallas negras.
- Detectar marcadores blancos sobre fondo negro (mallas o pequeño borde negro del marcador).

- Pintar automáticamente líneas y escribir textos en las imágenes.
- Calcular nuevas variables a partir de operaciones con los datos encontrados.
- Ofrecer control al usuario para poder modificar parámetros y resultados.
- Exportar los datos a valores separados por comas (CSV).

Requisitos no funcionales:

- El programa debía funcionar en los mismos sistemas operativos que Chronojump.
- El programa debía ser rápido para que en pocos años pueda *correr* en tiempo real.
- Todas sus bibliotecas debían ser software libre.

### 9.2.2. Obtención de imágenes

Previo al desarrollo, fue necesario registrar vídeos de saltos en distintas condiciones de luminosidad para poder ajustar el programa. Un software concebido a partir de pruebas con imágenes en condiciones de luminosidad similar es poco capaz de trabajar en condiciones distintas, por este motivo se buscó que la iluminación fuera heterogénea, tal y como describe la tabla 9.1.

Tabla 9.1: Variaciones de iluminación en las imágenes de prueba.

Tipo de luz	Ligera	↔	Intensa
Natural	Al aire libre en día nublado u oscureciendo. En condiciones de sombra.		Al aire libre en día soleado
Artificial	4 Fluorescentes 100 W zenital a 3 m aprox.	Lámpara de leds de 100 W frontal a 3 m aprox.	Focos 500 W frontal a 3 m aprox.
Ambas	Sol entra lateral por una ventana y existe iluminación artificial.		

El movimiento registrado fue el salto CMJ, y las condiciones de filmación fueron:

- La cámara se colocó a una altura similar a la de la rodilla en flexión de 90 grados con una orientación hacia la rodilla sin variación de altura. Los problemas de perspectiva se minimizaron debido a que el objeto de máximo interés estaba en el centro de la imagen, la cadera y el tobillo se encontraban alejados de las esquinas y la cámara distaba más de 3 metros. El plano de filmación fue lateral.
- Se usó la cámara Casio F1 HS de alta velocidad. El registro se produjo a 300 imágenes por segundo.
- En cuanto a la vestimenta, se filmaron dos tipos de secuencias:
  - Sujeto con mallas negras. La vestimenta del sujeto fue: malla negra sin decoraciones blancas y el resto de prendas blancas. El sujeto saltó usando zapatillas deportivas. Se rechazaron zapatillas cuyas decoraciones fueran oscuras en la zona próxima a las mallas. Además, el sujeto llevaba tres marcadores blancos por encima de las mallas en las zonas de cadera, rodilla y en algún lugar sobre las mallas contenido en el segmento rodilla-tobillo y próximo a este último.
  - Sujeto con pantalones cortos. Esta vestimenta difería de la anterior, ya que se usó pantalón corto sin decoraciones blancas y se ubicó un marcador blanco con un reborde negro en cada uno de los tres puntos articulares.
- Los marcadores estaban creados a partir de etiquetas autoadhesivas APLI. El tamaño de los marcadores únicamente blancos fue 4x3,6 cm. El tamaño de los marcadores blancos con reborde negro fue 4x2,5 cm y tenía un reborde interior de 2,5 mm de grosor pintado con rotulador negro permanente Edding 500.
- El fondo del plano de filmación era fijo y claro para que no se confundiera con las mallas.
- La iluminación a lo largo de una ejecución era invariable pues el movimiento detectado duraba menos de 2 segundos.
- Se inició y terminó cada grabación manualmente garantizando que el sujeto permaneciera de pie al inicio durante al menos dos imágenes.

### 9.2.3. Desarrollo del software

El software se desarrolló en C++ usando la biblioteca OpenCV, que cumple los requisitos no funcionales, pues es multiplataforma, está licenciada como software libre y la ejecución bajo C++ es veloz.

OpenCV permitía, además, resolver los requisitos funcionales enunciados tal y como se detalla a continuación:

- OpenCV es capaz de capturar desde un archivo de vídeo contenido en AVI.
- El umbral de luminosidad o threshold se calculó a partir de la identificación del brillo de la imagen y su correspondencia con el threshold en la experimentación con filmaciones de distinta luminosidad.
- La forma de las piernas revestidas con mallas negras se determinó por segmentación de Objetos y particularmente buscando el contorno más grande.
- La detección de los marcadores difería en función de momento:
  1. En la primera imagen, debido a que el sujeto se encontraba en bipedestación, los marcadores se detectaron buscando tres áreas blancas de tamaño similar, con una componente horizontal similar y situadas en la parte superior, central e inferior de la imagen. En las filmaciones con mallas negras era más sencillo, ya que los marcadores estaban contenidos en el contorno más grande.
  2. En las siguientes imágenes se usaba Back tracking y opcionalmente Forward tracking usando una predicción de R. Debido a que la filmación se realizó a 300 imágenes por segundo y a que el movimiento no es muy rápido, fue sencillo encontrar un marcador a partir de su posición en otros fotogramas, pues seguramente compartían una parte de su área.
  3. Si en algún momento no se encontraban los marcadores esperados, se pedía la intervención del usuario. Este podía efectuar las siguientes acciones: reasignar marcadores, modificar el threshold general o el cercano a cualquiera de los marcadores, modificar el radio de threshold de cualquier marcador, y, finalmente, avanzar o retroceder fotogramas.

El marcador fue detectado como una nube de puntos blancos conectados cuyo centro se calculaba usando el centro del rango de puntos en cada uno de sus ejes.

- El resto de requisitos funcionales descritos fue trivial de implementar con las funciones de OpenCV y el acceso a disco de C++.

Debido a que no se disponía de experiencia previa en OpenCV, se contactó con programadores para que escribieran una unidad mínima funcional, y después se siguió a partir de la base creada.

#### 9.2.4. Validación

La validación se realizó de forma cualitativa a partir de la observación de cómo se comportaba el programa ante los vídeos que fueron analizados en el capítulo 10. Se observaron los siguientes indicadores:

- El programa no se cuelga.
- Se detecta correctamente lo que se persigue en cada imagen (marcadores y/o mallas).
- Los elementos añadidos como flechas o textos se visualizan correctamente.
- Se vuelcan correctamente los datos a archivos de texto.

### 9.3. Resultados y discusión

El programa obtenido, KneeAngle, está creado y está disponible en la red. Según el análisis COCOMO proporcionado por Sloccount, dispone de 2719 líneas de código en C++, y reescribir el programa de nuevo necesitaría 0,57 personas/año y un presupuesto de 77,227 dólares.

KneeAngle fue desarrollado inicialmente en agosto de 2008 por Sharad Shankar y Onkar Nath Mishra de la empresa Logibrick (India). Esta primera versión tenía 281 líneas de código C++. Posteriormente se refinaron los métodos, se añadieron funcionalidades y se escribió una interface de usuario, lo que supuso que el código creciese en una orden de magnitud.

El programa debía funcionar en Linux, Windows y MacOSX, pero hasta el momento sólo se ha creado la instalación para Linux y no ha podido ser probado en otros sistemas.

La figura 9.1 muestra cómo el programa reconoce los marcadores blancos con reborde negro sin el uso de las mallas negras. La figura 9.2 muestra cómo son reconocidas las mallas negras (el contorno convertido a blanco más grande) y los marcadores en ellas. Por último, la figura 9.3 muestra cómo el programa se ajusta semi-automáticamente para reconocer los marcadores, aunque éstos se vean fuera de las mallas en contacto con una prenda de igual color al marcador. La efectividad de estos ajustes depende de lo cerca que estén los puntos en conflicto y del ajuste del threshold general y particular de cada marcador.

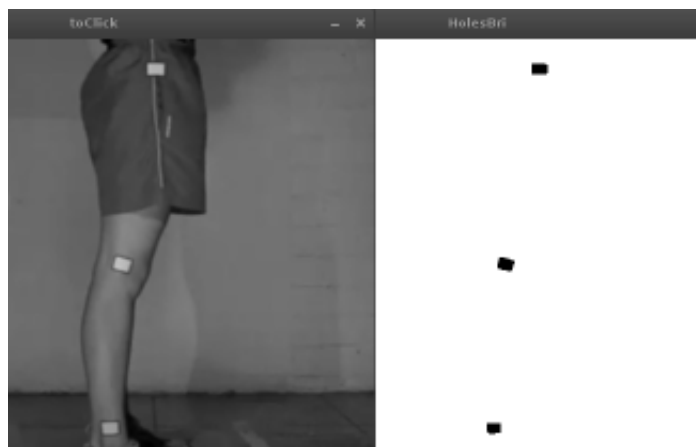


Figura 9.1: Reconocimiento de marcadores sin el uso de las mallas.

El cálculo del centro articular de cada marcador requiere discusión, pues el marcador se ve como una nube de puntos blancos conectados cuyo centro se calculó usando el centro del rango de

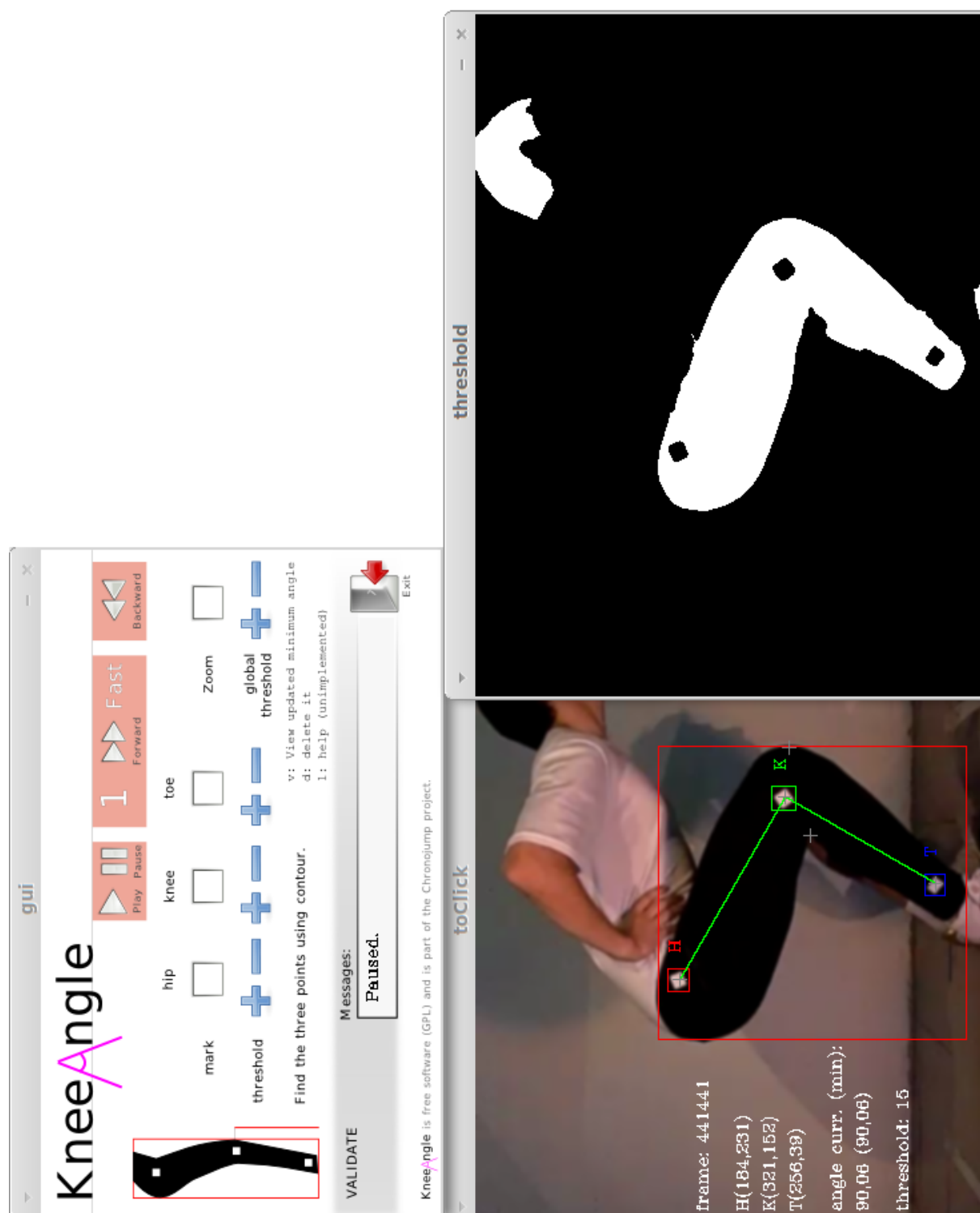


Figura 9.2: Reconocimiento de mallas negras y marcadores.



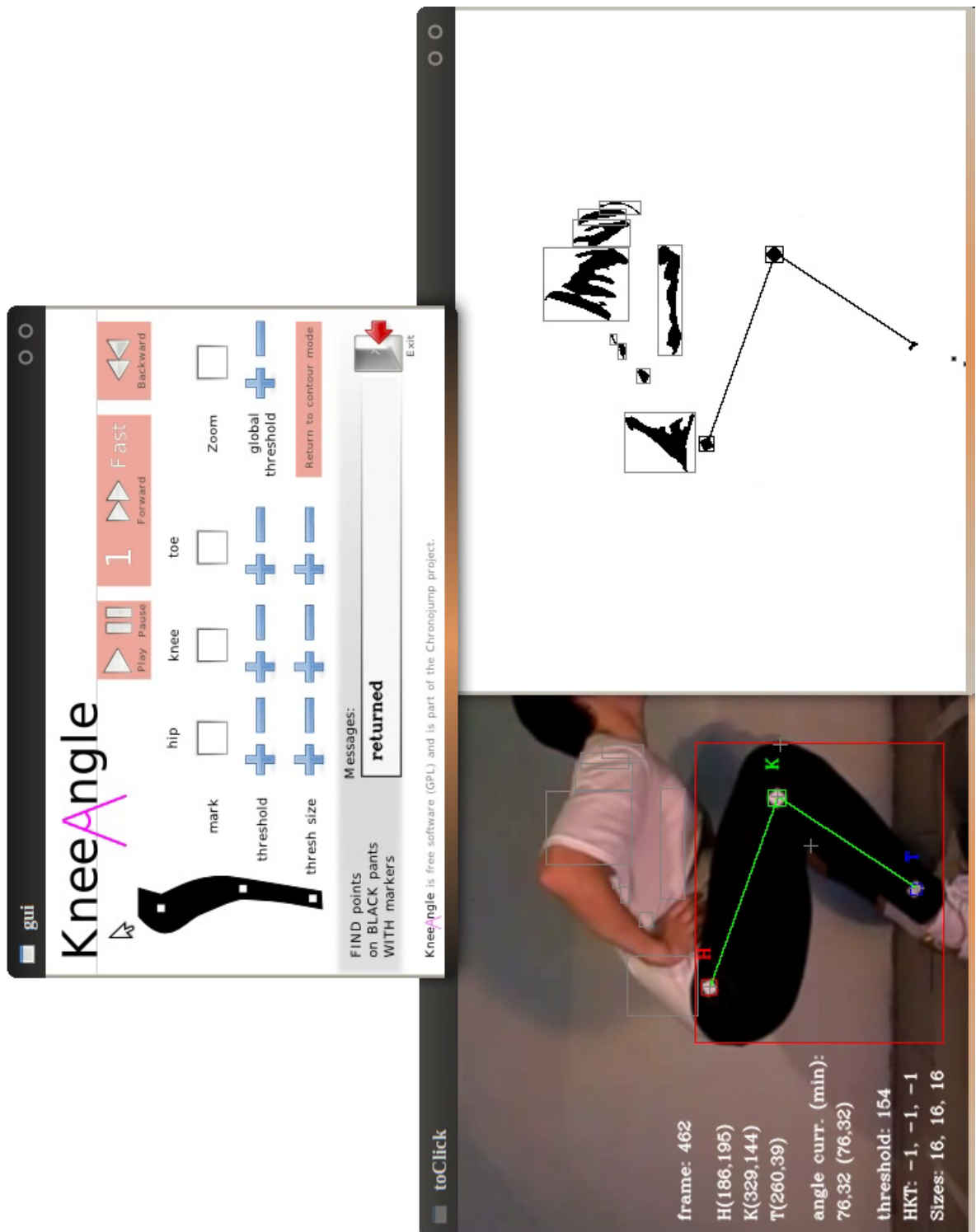


Figura 9.3: Reconocimiento de marcadores cuando un punto sale de las mallas y contacta con otra prenda blanca.

puntos en cada uno de sus ejes. El resultado que proporciona este método de cálculo no difiere prácticamente de otros métodos cuando el marcador no ha sufrido deformaciones, pero sí que lo hace cuando el plano del marcador no tiene la forma esperada. Esta deformación puede ser debida a que el plano de filmación no es paralelo al plano del marcador, a que los pliegues anatómicos o de la piel le confieran relieve o a la aparición de sombras u otros problemas en la detección. La figura 9.4 muestra las diferencias en la identificación del centro del marcador según tres métodos distintos.

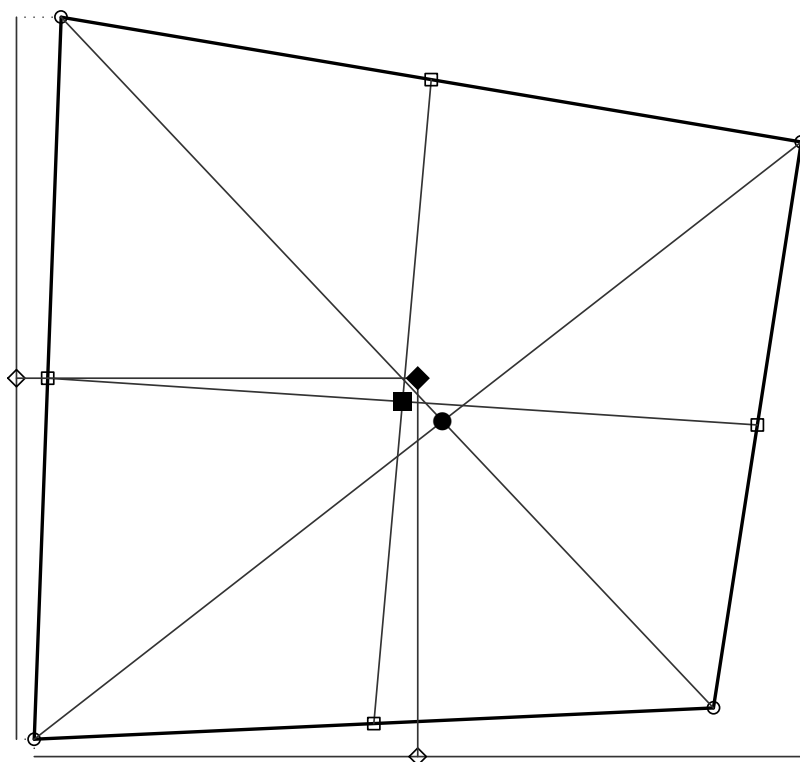


Figura 9.4: Identificación del centro del marcador cuando existe deformación. Se representan tres métodos: cálculo por bisectrices (círculos), cálculo por mediatrices (cuadrados), cálculo a partir del centro de los rangos X e Y (rombos). Este último método fue el usado en el estudio.

El método usado fue el cálculo a partir del centro de los rangos X e Y. Este método es el más rápido pero puede ser más errático que otros. Se espera que este error se *amortigüe* con el modelo de predicción que se trata en el capítulo siguiente.

En cuanto a la validación, después de diversas revisiones del programa, se observa que en el análisis de los vídeos que se presentan en el capítulo 10, los indicadores descritos ofrecen un resultado positivo:

- El programa no se bloquea. El único problema que se observa es una ralentización excesiva si el usuario elige incorrectamente un threshold demasiado alto, pues se detectan

demasiadas formas. Debido a que el programa estará documentado, este error se atribuye a un mal uso.

- Respecto a la correcta detección de lo que se persigue en cada imagen (marcadores y/o mallas), se observa que en la mayor parte de los casos no es necesaria ninguna operación manual. La excepción aparece cuando la proximidad de marcadores con otras zonas blancas obliga a usar ajustes de threshold manuales. Por otro lado, estos ajustes son necesarios únicamente para la modalidad de validación del instrumento, pues en el uso real del mismo no se usan marcadores, de modo que una iluminación moderada será suficiente para detectar las mallas de forma automatizada.
- Los elementos añadidos como flechas o textos se visualizan correctamente.
- Se vuelcan correctamente los datos a archivos de texto.

En cuanto a la fiabilidad, el instrumento siempre detecta los mismos puntos ante dos vídeos iguales. Nuevamente, en la modalidad de validación, puede requerirse intervención del usuario si la imagen tiene calidad deficiente o los marcadores entran en contacto con zonas blancas. En este caso, ante dos reproducciones del mismo vídeo, el usuario puede seleccionar parámetros de threshold distintos que producen una variación muy ligera en fotogramas localizados.

## 10. Creación de un modelo de predicción

Una de las cosas más fascinantes de los programadores es que no puedes saber si están trabajando o no, sólo con mirarlos. A menudo están sentados aparentemente tomando café, chismorreando o mirando a las nubes. Sin embargo, es posible que estén poniendo en orden todas las ideas individuales y sin relación que pululan por su mente.

---

Charles M. Strauss

El presente capítulo explora la creación de un modelo estadístico que permite predecir el ángulo de flexión de las rodillas a partir de la forma de las piernas de un sujeto en un instante dado. Para el *entrenamiento* del modelo es necesario proporcionar datos de imágenes en que se relacione el ángulo de las rodillas con variables relativas a la forma de las piernas, y con variables relativas a la posición de los marcadores.

Yeadon y Challis (1994) distinguen entre tres tipos de estudios posibles en la disciplina de la Biomecánica: estudios descriptivos, estudios experimentales y estudios teóricos o de modelización. El presente capítulo se refiere a un estudio del tercer tipo, pues se pretende obtener un conjunto de información que permita elaborar un modelo predictivo.

### 10.1. Objetivo

El objetivo del presente capítulo es el siguiente:

Desarrollar y validar un modelo que permita predecir el ángulo de flexión de la articulación de las rodillas en la fase de contramovimiento de un salto CMJ usando la forma de las piernas recubiertas por mallas de color negro sin necesidad de usar marcadores. Dicho modelo debe ser más consistente a los 90 grados.

### 10.2. Metodología<sup>6</sup>

Se realizó un experimento para captar los datos que permitían crear el modelo y *entrenarlo*.

---

<sup>6</sup>Carlos J. Gil Bellosta se prestó a dar soporte estadístico al siguiente experimento.

### 10.2.1. Diseño del experimento

Los datos se obtuvieron a partir de una muestra de 59 estudiantes universitarios de primer curso de ciencias del deporte. Otros criterios de inclusión de la muestra fueron:

- Sexo: ambos (47 hombres y 12 mujeres)
- Edad: mínimo 18 años.
- Ser físicamente activos: realizar actividad física un mínimo de dos días a la semana.

Se excluyeron:

- Los atletas de élite.
- Los sujetos que hubieran tenido un antecedente de lesión traumática en los miembros inferiores.

De cada sujeto se obtuvieron las variables: sexo, edad, altura, peso y deporte practicado. La tabla 10.1 contiene los estadísticos sobre la muestra.

Tabla 10.1: Muestra para la creación de un modelo de predicción.

	Media	Mediana	Desviación estándar	Recorrido intercuartílico
Edad	20,1	18,7	3,8	1,8
Altura	173,3	173,0	9,4	13,0
Peso	68,8	70,0	10,5	14,0

El deporte más practicado fue el fútbol seguido del básquet.

Debido a que muchos entrenadores realizan variaciones del salto CMJ y a que la finalidad del experimento era desarrollar un software que indicase si una flexión ha sido de 90 grados o no, se dejó libre el ángulo de flexión previa al salto para que los datos obtenidos tuvieran la variabilidad que acostumbran a encontrar los entrenadores. Otros condicionantes de la técnica fueron: manos en la cintura todo el tiempo, tronco recto y caída con las puntas de los pies.

El experimento se realizó en dos sesiones. En la primera se aprendió la ejecución del salto CMJ a la vez que los sujetos se familiarizaban con la instrumentación. En la segunda se realizó el experimento propiamente dicho, cuyo protocolo fue el siguiente:

1. Calentamiento libre.

2. Creación de dos grupos, el primero realizó tres saltos con mallas negras y después tres saltos con pantalones cortos, mientras que el segundo grupo lo hizo en orden contrario.
3. En el caso de los sujetos que saltaban con mallas, anteriormente se ubicaron los marcadores.
4. Se repitió el salto si la técnica no había sido adecuada, si los marcadores se habían movido, o si el sujeto consideró que el salto no había sido "máximo". Para ello, se le preguntó al finalizar cada uno de los saltos si el salto había sido "máximo". Se prestó atención a la fatiga en los sujetos que debían repetir saltos dejando un par de minutos de recuperación a los sujetos que manifestaron que se encontraban cansados.
5. El descanso máximo entre cada grupo de tres saltos fue de 5 minutos.
6. Los saltos fueron registrados a 300 imágenes por segundo.

El material usado fue el siguiente:

- Mallas negras sin decoraciones blancas.
- Pantalones cortos sin decoraciones blancas.
- Marcadores creados a partir de etiquetas autoadhesivas APLI. El tamaño de los marcadores únicamente blancos fue: 4x3,6 cm. Por otro lado, el tamaño de los marcadores blancos con reborde negro fue: 4x2,5 cm y tenía un reborde interior de 2,5 mm de grosor pintado con rotulador negro permanente Edding 500.
- Cámara de vídeo de alta velocidad Casio F1 HS.
- Software Chronojump.
- Microcontrolador Chronopic 3.
- Plataforma de contactos Ergojump Bosco Sytem.

#### 10.2.2. Ubicación de los marcadores

Los marcadores se ubicaron en puntos anatómicos que permitían trazar los segmentos que componían el ángulo de flexión de la articulación de la rodilla.

De la articulación coxofemoral se usó, como punto de referencia, el trocánter mayor, o apófisis proximal del fémur. Dicho trocánter tiene una meseta que es fácil de palpar y cuyo centro coincide con el centro articular desde el plano sagital y en bipedestación. Cuando aparece la flexión, gracias a la anteversión de la pelvis, la meseta trocantérea se atrasa y cae respecto al centro articular. En ese momento la meseta trocantérea no coincide con la articulación coxofemoral, pero representa mejor el segmento muslo por seguir la diáfisis o línea longitudinal del fémur. Se creyó más

conveniente usar el centro de la meseta que encontrar el centro articular de la coxofemoral, pues en flexión la línea coxofemoral-rodilla tiene mucha más pendiente que el fémur.

El cóndilo femoral externo es el punto anatómico óptimo de la articulación de la rodilla que permitió trazar las líneas muslo y pierna, debido a que anatómicamente responde al centro de la articulación, es bastante superficial y tiene una pequeña cresta que es fácil de palpar. Dicha cresta se encuentra justo por encima de la interlínea articular.

En cuanto al tobillo, el maléolo peroneal se corresponde con dicha articulación y es muy sencillo de identificar.

Así, los marcadores se ubicaron en estos puntos anatómicos:

- Punto central de meseta trocantérea.
- Cresta del cóndilo femoral externo.
- Maléolo peroneal.

### 10.2.3. Comparación entre saltos según indumentaria

La herramienta informática para la evaluación del salto precisó de la ejecución del mismo con mallas. Sin embargo, debido a que los deportistas están más acostumbrados a realizar actividad física en pantalón corto, se comparó primero si existían diferencias significativas en los tiempos de vuelo del salto (que se corresponden con la altura máxima alcanzada) y en la flexión máxima dependiendo de la indumentaria.

La figura 10.1 no muestra diferencias aparentes entre en tiempo de vuelo y ángulo de flexión máxima en función de la indumentaria.

Por otro lado, se realizó un análisis de la varianza para controlar el efecto del sujeto. Se examinó la hipótesis que afirmaba que no existían variaciones en las ejecuciones de los saltos entre los saltos que realizaron los distintos voluntarios con indumentarias distintas. Aunque el tiempo de salto no parece significativamente distinto ( $p$  valor 0,75), sí que se aprecian diferencias significativas en el ángulo de flexión ( $p$  valor prácticamente 0). Estas diferencias indican que los sujetos llegaron a ángulos de flexión más pequeños, dicho de otro modo, hacían flexiones más profundas. Se advierte, por tanto, que la validez externa del experimento se ve ligeramente comprometida. Este fenómeno puede ser atribuible a factores afectivos y/o incomodidad física en el uso de las indumentarias en individuos poco acostumbrados a ello.

Hecha esta comparación, el resto del estudio tuvo en cuenta únicamente los saltos realizados con mallas.

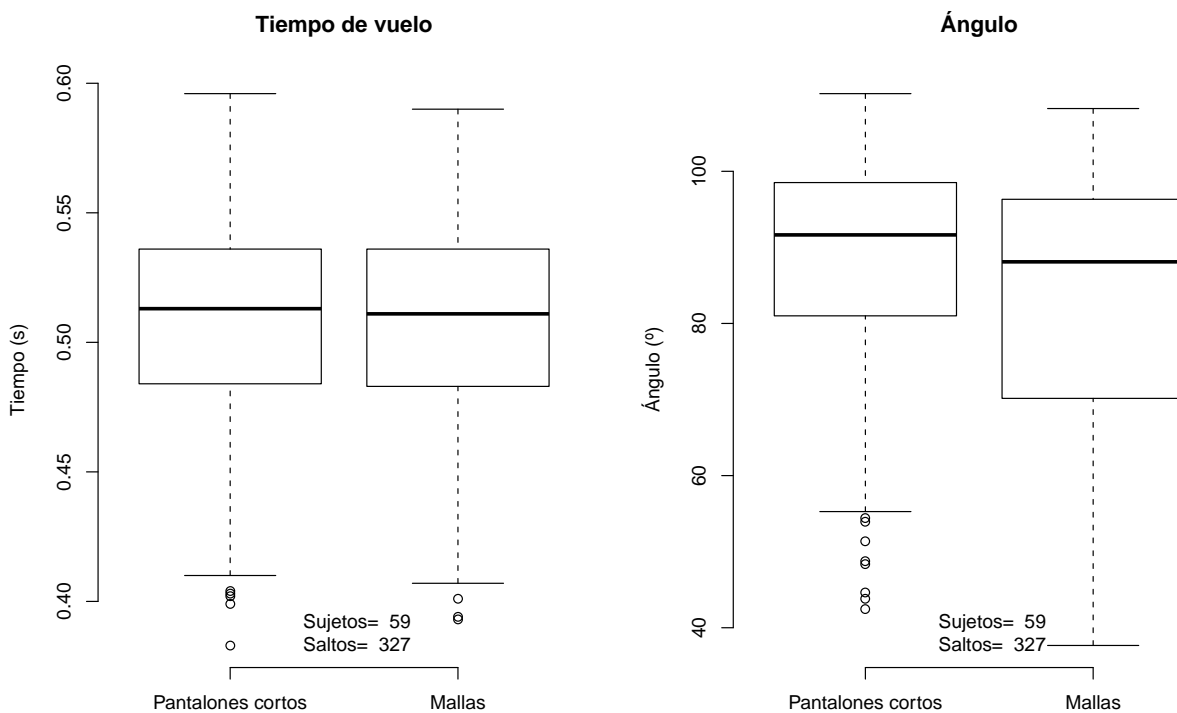


Figura 10.1: Tiempo de vuelo y ángulo máximo de flexión en función de la indumentaria usada.

#### 10.2.4. Mortalidad de la muestra

Debido a problemas de encuadre de la imagen y de iluminación, se han tenido que descartar saltos. De los 59 sujetos iniciales (47 hombres y 12 mujeres), se conservan 13 (11 hombres y 2 mujeres). El número de saltos desciende de 327 a 35.

#### 10.2.5. Variables predictivas

Para cada salto de cada sujeto se generó una secuencia de fotogramas. En cada uno de ellos, mediante segmentación de objetos, se calcularon estas tres variables:

1. *TightsHP*: Se refiere al tamaño vertical de las mallas en un fotograma concreto. La flexión de rodillas provoca que la altura de las mallas disminuya. Esta dimensión se registra con un porcentaje que expresa la altura de la malla respecto a la altura máxima de las mallas obtenida en la situación inicial de bipedestación. El nombre de la variable responde a la altura porcentual de las mallas: *tights Height Percent*.
2. *KpfY*: Altura de la rodilla respecto a *tightsHP*. El nombre de la variable responde a la componente vertical de *knee point front*. Esta variable no se refiere al centro articular de la rodilla. Se refiere a la altura del punto geométrico más avanzado de la rótula en el plano sagital. En función del tipo de patela, alta o baja, y del comportamiento de la misma, se



producen variaciones entre sujetos en la altura de la rótula respecto al centro articular de la rodilla en el movimiento de flexo-extensión. Se usa *KpfY* y no un punto anatómico debido a que *KpfY* es fácilmente observable por un software de visión por computador.

### 3. *Ángulo real*: Ángulo proporcionado por los marcadores.

En la secuencia desde el inicio de la flexión hasta la flexión máxima, cada salto presentó una media de 262 fotogramas. Hay que hacer constar que en algunos fotogramas aislados, el software que calculó las variables anteriores introdujo imprecisiones (outliers) que fueron detectados a lo largo del análisis y que fueron eliminados del conjunto de datos para la construcción del modelo final.

La figura 10.2 revela una clara relación funcional entre *tightsHP* y el *ángulo real*. La gráfica muestra cómo cada uno de los saltos analizados se comportó de forma similar en la evolución de *tightsHP* respecto al *ángulo real*.

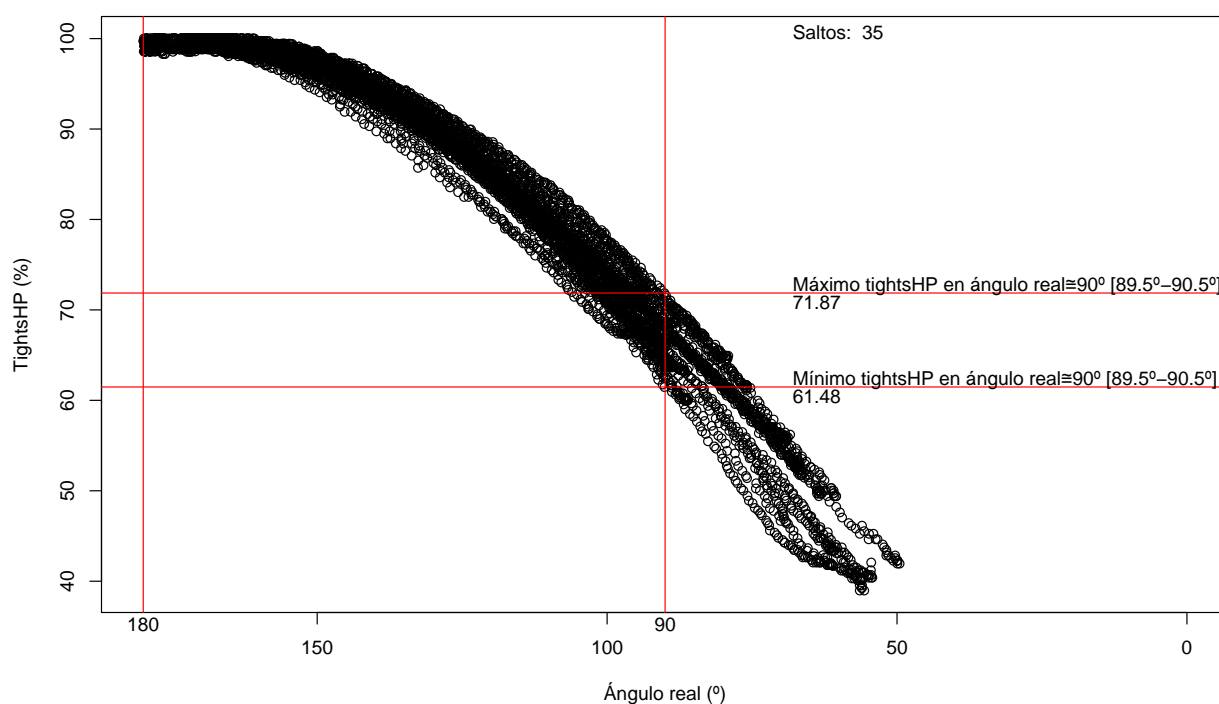


Figura 10.2: *Ángulo real* respecto a *tightsHP* en los distintos saltos.

En la zona de interés alrededor de los ángulos de flexión próximos a los 90 grados (figura 10.3), se observa cómo la relación entre ambas variables fue prácticamente lineal.

Además, la flexión de 90 grados ocurrió en un intervalo de *tightsHP* entre 61,5 y 71,5. Si se agrupan las distintas observaciones según intervalos de valores *kpfY*, se observa cómo esta variable permite afinar la predicción del *ángulo real* (figura 10.4).

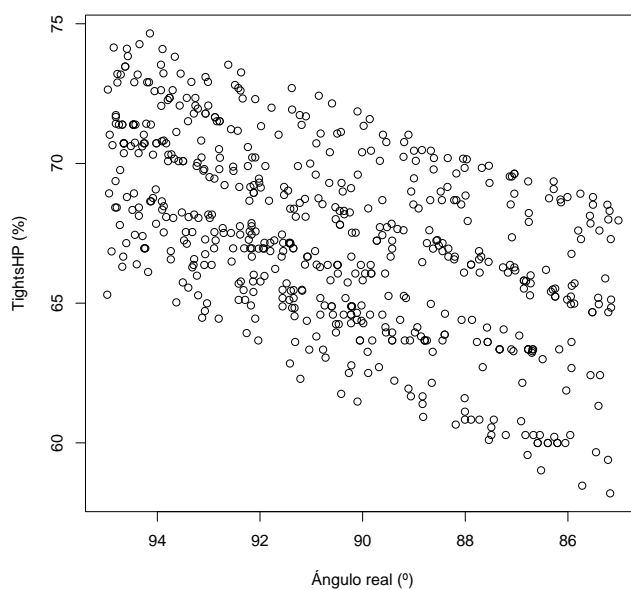


Figura 10.3: *Ángulo real* respecto a *tightsHP* en la zona de 90 grados de *ángulo real*.

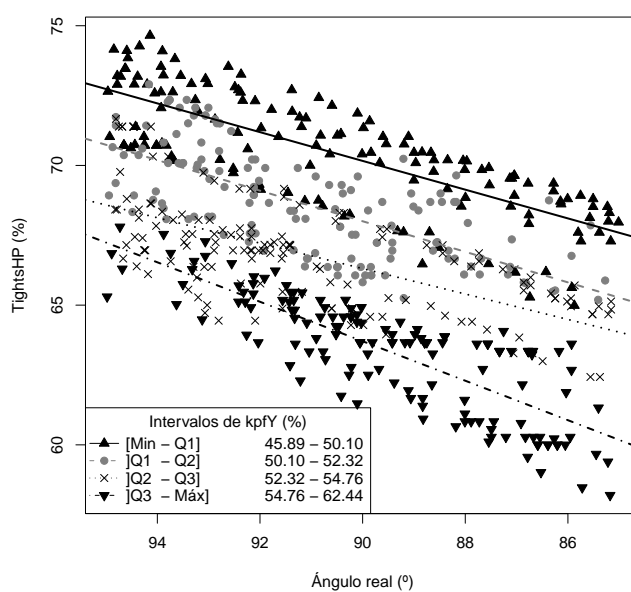


Figura 10.4: *Ángulo real* respecto a *tightsHP* y según intervalos de *kpfY* en la zona de 90 grados de *ángulo real*.

El análisis de las tres figuras anteriores motivó que se planteara un modelo lineal para la predicción del ángulo de la rodilla en la zona de 80 a 100 grados.

Analizando las figuras 10.5 y 10.6, se observa que las fotografías correspondientes al salto 24-4 de la foto composición, así como las líneas discontinuas en el diagrama, mostraban una flexión de 90 grados en que la altura de la rodilla ( $kpfY$ ) estaba en el centro vertical del recuadro que conforma la altura de las mallas. Por otro lado, las fotografías correspondientes al salto 38-5, así como las líneas continuas en el diagrama, corresponden a una flexión de 90 grados en que  $kpfY$  estaba mucho más alto debido a que el sujeto se encontraba en una posición más sentada.

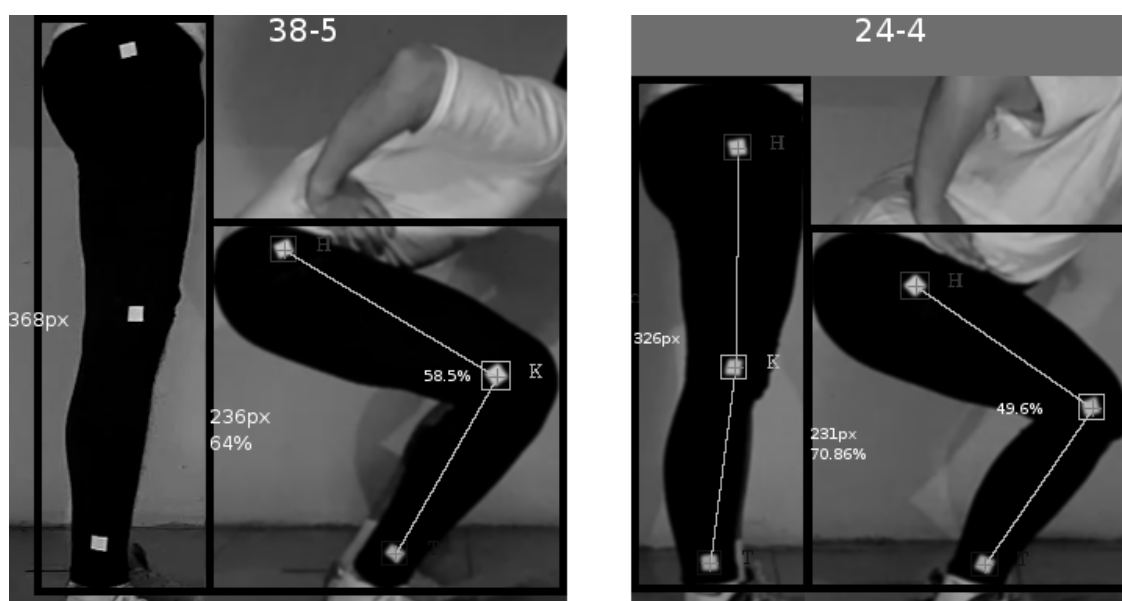


Figura 10.5: Relación entre *Ángulo real*, *tightsHP* y  $kpfY$  en los momentos de máxima extensión y de flexión de 90 grados en dos saltos de individuos distintos.

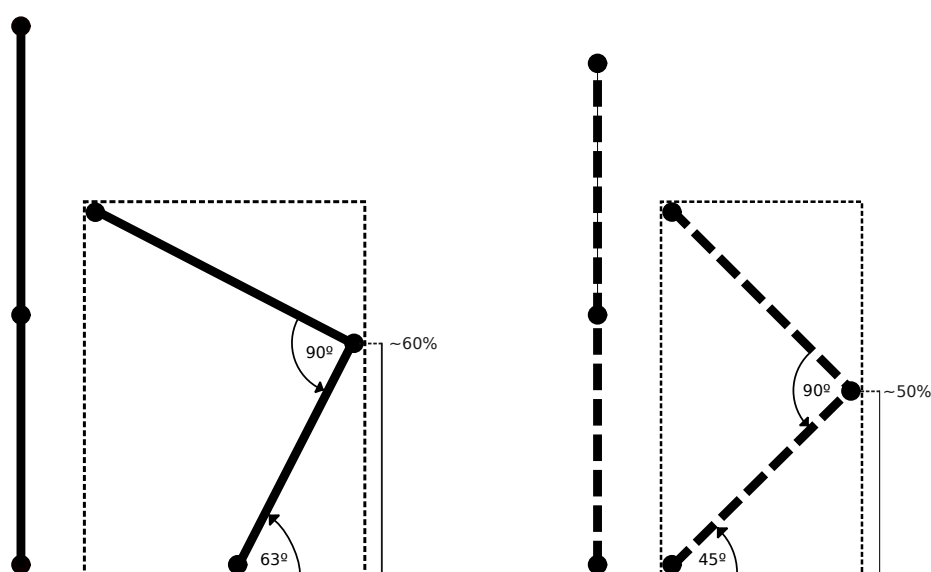


Figura 10.6: Diagrama ilustrativo sobre la relación trigonométrica entre *Ángulo real*, *tightsHP* y  $kpfY$  en los momentos de máxima extensión y de flexión de 90 grados.

En la figura 10.7 se observa cómo la variable  $kpfY$  era dependiente del sujeto en la zona de 80 a 100 grados.

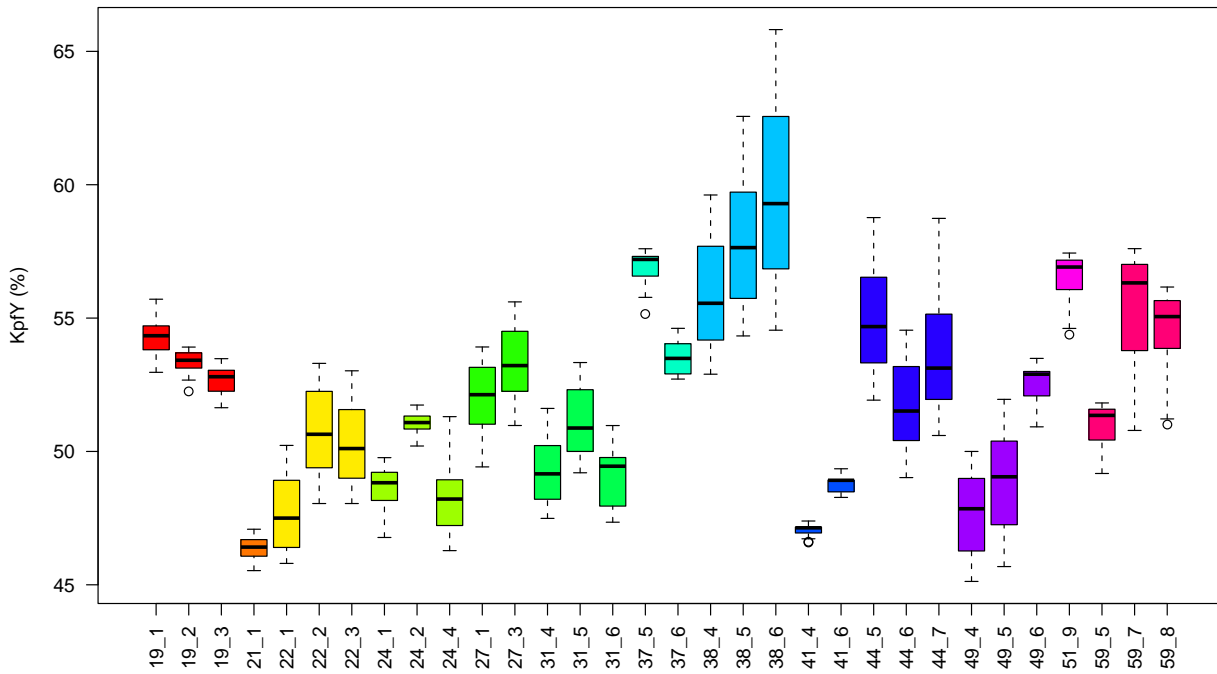


Figura 10.7: Diagramas de caja de  $kpfY$  en la zona de 80 a 100 grados de *ángulo real*. Los saltos del mismo sujeto tienen el mismo color. No aparecen 3 cajas correspondientes a 3 saltos debido a que la flexión no llegó a los 100 grados en esos casos.

La nomenclatura en el eje horizontal indica el nombre del sujeto y el número de salto. Así  $19\_1$  corresponde al primer salto del sujeto 19.

Aunque la variable  $tightsHP$  guardó una relación importante con el ángulo de flexión, ésta variaba de individuo en individuo e incluso entre los saltos del mismo individuo. Sin embargo, la variable  $kpfY$  recogía gran parte de la información relativa a estas variaciones que ayudaba a explicar el ángulo de flexión.

#### 10.2.6. Saltos descartados

Se descartaron tres saltos cuya flexión no llegó a los 100 grados. Posteriormente, se representaron las rectas de regresión entre  $tightsHP$  y el *ángulo real* salto a salto (ver figura 10.8). Los saltos que tenían una pendiente más distinta al resto correspondían a saltos en los que no se alcanzaron flexiones próximas a los 90 grados:  $21\_1$ ,  $37\_5$ ,  $37\_6$ ,  $41\_4$ ,  $41\_6$ . Estos cinco saltos también fueron eliminados en la construcción del modelo. Los saltos incluidos se muestran en la figura 10.9.

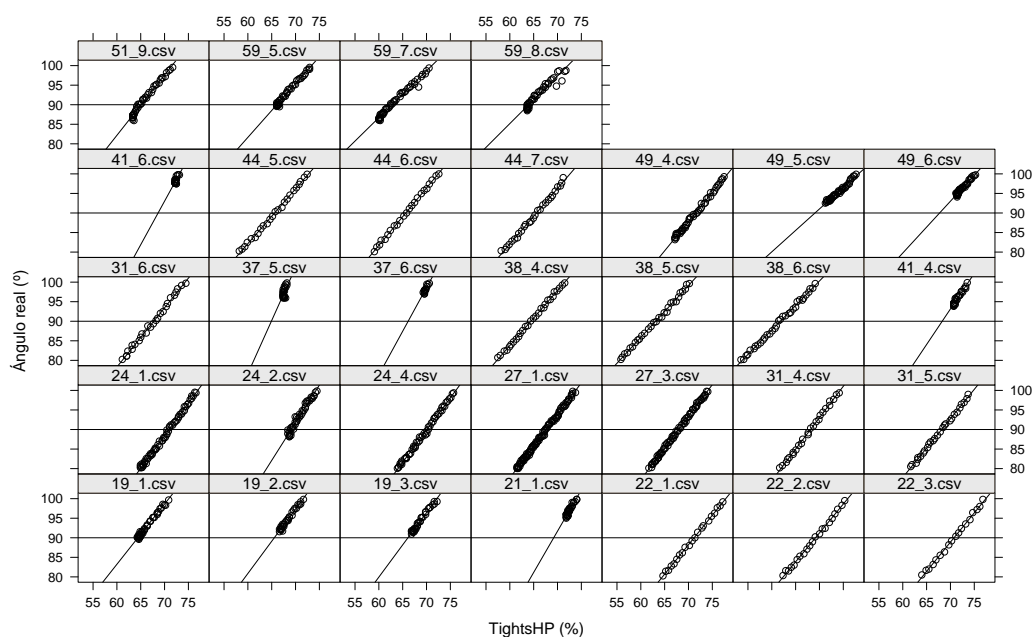


Figura 10.8: Regresiones entre las rectas *tightsHP* y *ángulo real* en las flexiones que pasan por el intervalo de 80-100 grados.

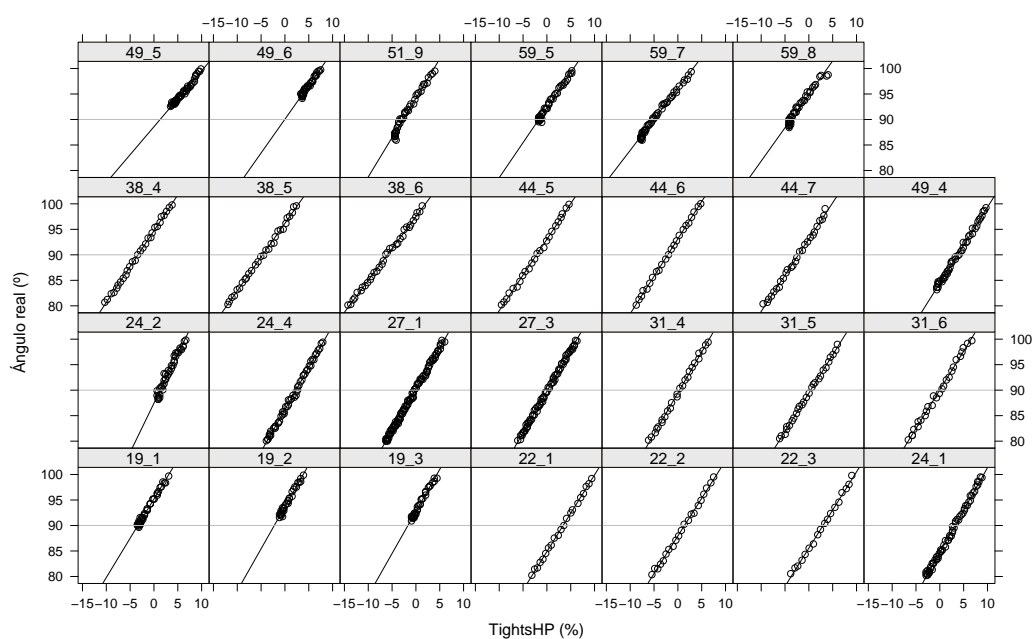


Figura 10.9: Regresiones entre las rectas *tightsHP* y *ángulo real* en las flexiones que pasan por el intervalo de 80-100 grados habiendo borrado las ejecuciones de curvatura muy diferenciada.

### 10.2.7. Diseño del modelo

El modelo planteado fue lineal mixto con dos términos fijos, *tightsHP* y *kpfY*, y otros (el individuo y el salto de cada individuo) que eran aleatorios dado que el interés del modelo consistía en extrapolar los resultados a saltos e individuos distintos.

El modelo fue:

```
angulo.predicho <- lme(angle ~ tightsHP + kpfY,  
                        random = ~ 1 | id.ind / id.jmp,  
                        data = dat.90)
```

donde *dat.90* era el conjunto de los datos representado en la figura 10.9. El modelo se describió usando la nomenclatura del paquete de R usado en el análisis, *nlme* (Pinheiro y col., 2010).

## 10.3. Resultados y discusión

### 10.3.1. Resultados

La diferencia entre el ángulo real y el *ángulo predicho* por el modelo fue satisfactoria, tal y como se aprecia en la figura 10.10. No obstante, se identificaron tres observaciones alejadas del resto. La exploración manual de los datos y las imágenes mostraron que esas tres observaciones correspondían a fotogramas en los que el software de seguimiento había detectado incorrectamente algún punto articular debido a problemas de iluminación. Estos tres fotogramas fueron eliminados.

Las figuras 10.11 y 10.12 muestran los valores residuales del modelo final por individuo y por salto respectivamente. Se observa que el error absoluto en la aplicación del modelo en los distintos sujetos es normalmente inferior a 2 grados. El sujeto 24 es el que tiene valores más alejados de 0, y esto se aprecia en sus dos saltos.

### 10.3.2. Validación

Para la validación del modelo, en primer lugar se determinó el error absoluto en la predicción del ángulo en todos los fotogramas entre 80 y 100 grados de flexión. Para ello se procedió a entrenar el modelo con el 70 % de los datos de forma aleatoria y a predecir, gracias al entrenamiento, el 30 % restante de los datos, se midió el error y se repitió el proceso de forma automatizada 100 veces. Esta manera de proceder se asemeja a la situación de uso del software por parte del usuario final, en la que el modelo creado y entrenado se aplica a saltos sobre los que no se tenía conocimiento previo.

La media de error absoluto fue 2,624 grados, la desviación estándar: 1,575 y su histograma se puede ver en la figura 10.13.

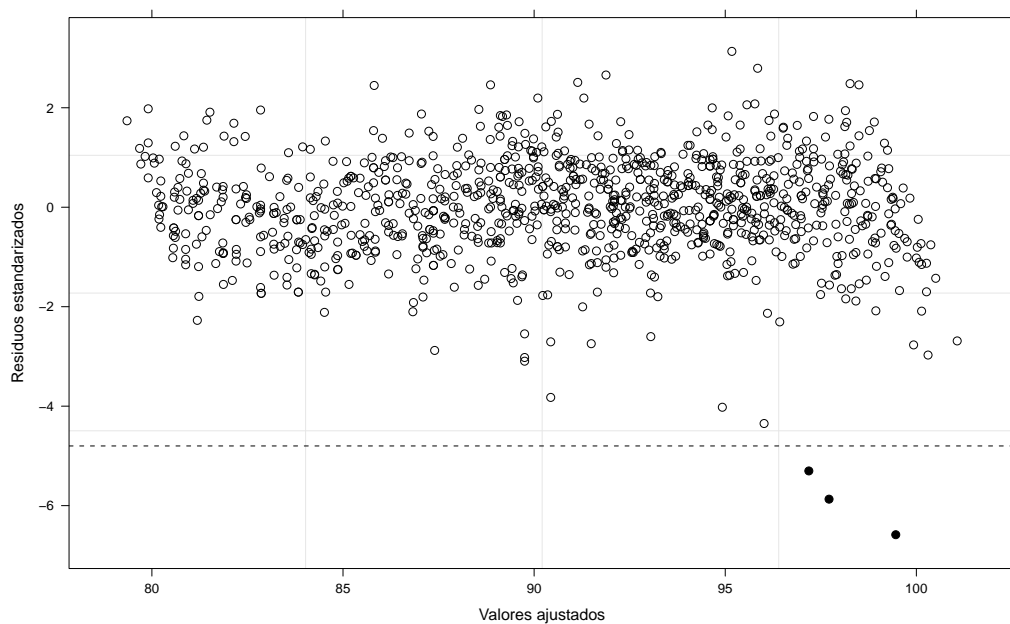


Figura 10.10: Residuos de la aplicación del modelo.

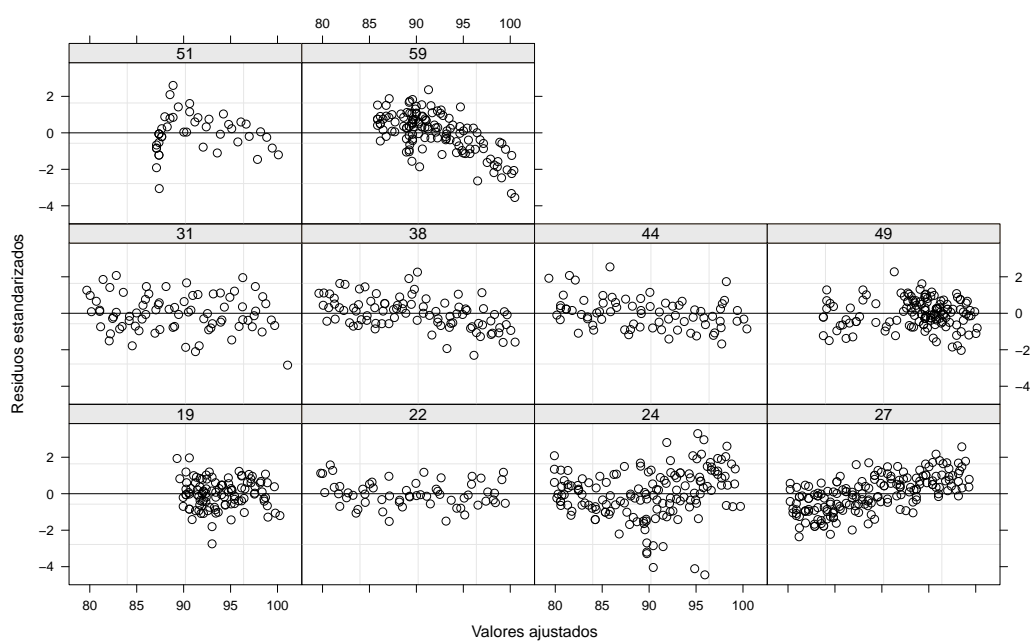


Figura 10.11: Residuos de la aplicación del modelo por individuo.

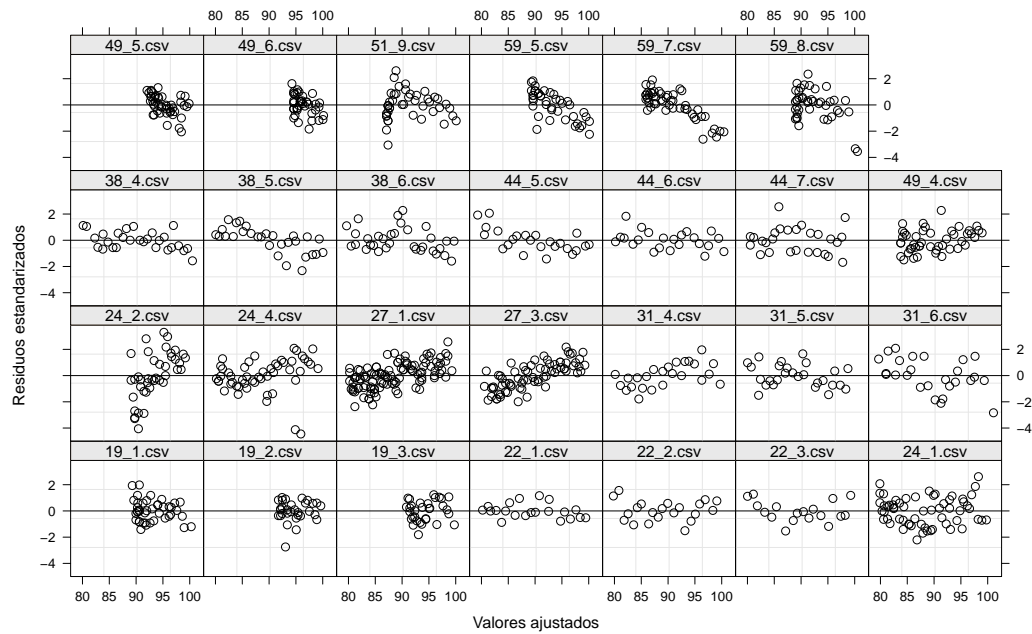


Figura 10.12: Residuos de la aplicación del modelo por salto.

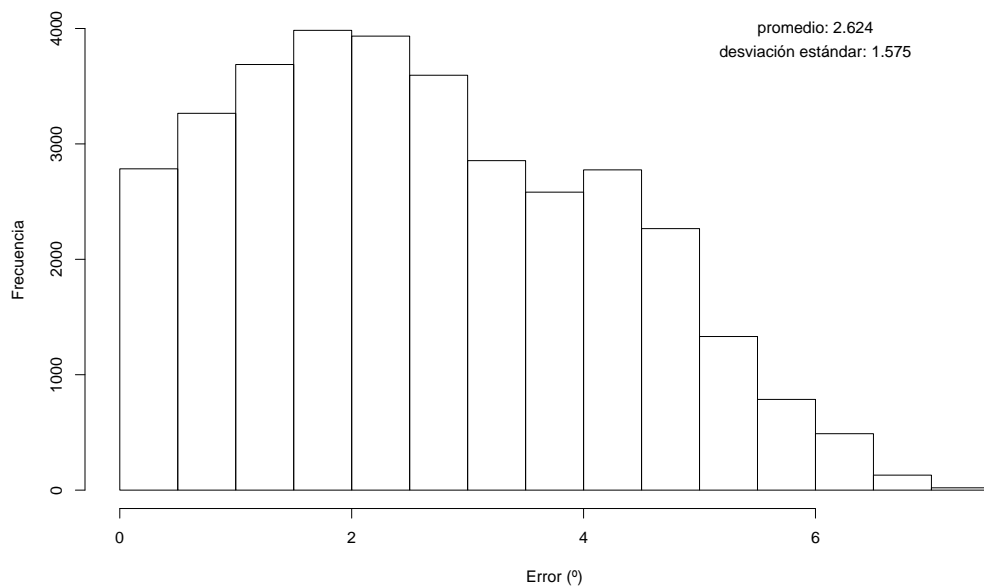


Figura 10.13: Error absoluto en la predicción del ángulo entre 80-100 grados de flexión.



Se procedió a determinar el error en la predicción de los ángulos en todo el rango de flexión (no únicamente en el rango 80-100 grados). Obviamente el modelo se comporta mejor cuando se predicen los ángulos para los que ha sido entrenado. La figura 10.14 muestra los diagramas de caja del error en distintos ángulos. Se observa que el error de predicción crece fuera del intervalo de interés, pero no lo suficiente como para que una flexión lejana a ese intervalo sea predicha erróneamente como cercana a los 90 grados. Por ejemplo, continuando con la figura 10.14, en el conjunto de datos probado, a los 138 grados de flexión puede aparecer un error máximo de 11 grados, aunque estos 11 grados podrían situar la predicción en 127 grados, este valor está muy alejado de los 90 grados que pide la ejecución del test.

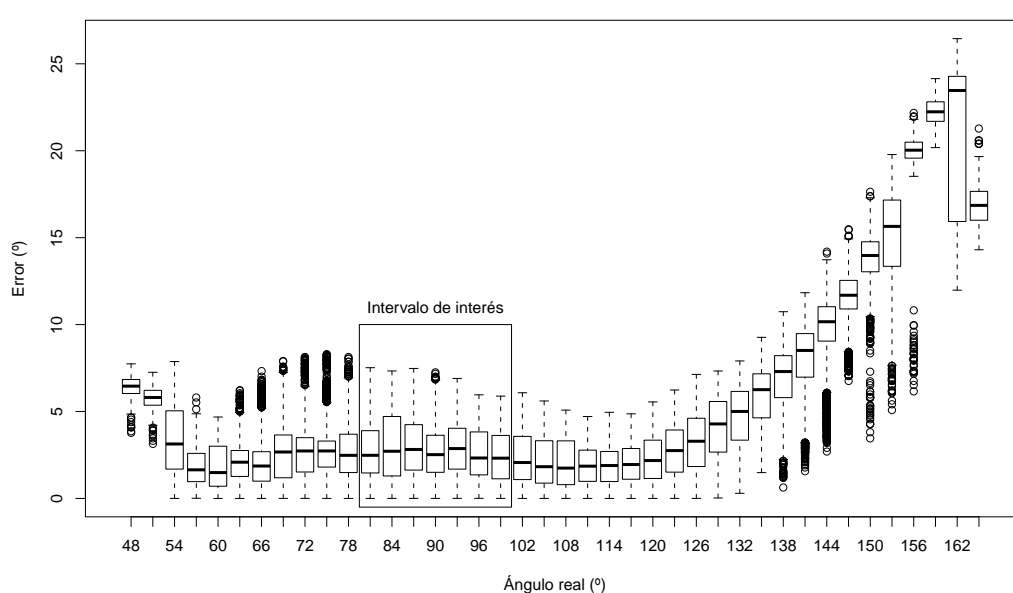


Figura 10.14: Error absoluto en la predicción del ángulo en todo el rango de flexión.

Continuando con la validación, se calculó el error en la predicción de los ángulos en todo el rango de flexión, pero sólo de los saltos que habían sido descartados para el entrenamiento del modelo debido a que la flexión no llegaba a los 90 grados. Los resultados se observan en la figura 10.15.

Con la finalidad de obtener una información cualitativa sobre el error de predicción, se usaron los mismos datos que en la gráfica anterior (figura 10.15), pero se operó únicamente en el fotograma de mayor flexión de cada uno de los saltos. De este modo se construyó un gráfico ilustrativo de acierto en el fotograma de mayor interés. En primer lugar se separó el rango de ángulos en cinco zonas (tabla 10.2).

Finalmente se preparó el gráfico que mostraba el error de predicción en cada una de las zonas (figura 10.16). Estudiando dicho gráfico, se observa nuevamente que los errores detectados eran mayores cuanto más alejado se encontraba el ángulo de los 90 grados, pues era el intervalo para el que se había diseñado el modelo. La principal aportación de esta ilustración es que cuando se detectaba un ángulo en la zona de "90", en realidad ese ángulo se encontraba en dicha zona.

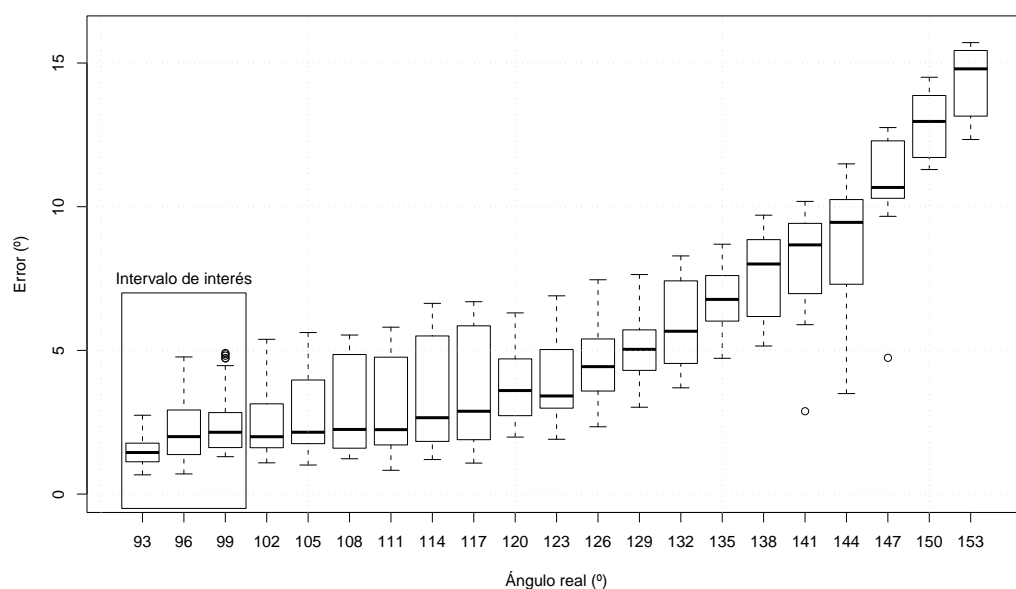


Figura 10.15: Error absoluto en la predicción del ángulo en todo el rango de flexión en las flexiones descartadas para el entrenamiento del modelo por no alcanzar los 90 grados.

Tabla 10.2: Zonas en la predicción del ángulo.

Zona	Límite mínimo	Límite máximo
“Muy baja”		$< 80^{\circ}$
“Baja”	$\geq 80^{\circ}$	$< 85^{\circ}$
“90”	$\geq 85^{\circ}$	$\leq 95^{\circ}$
“Alta”	$> 95^{\circ}$	$\leq 100^{\circ}$
“Muy alta”	$> 100^{\circ}$	

Cuando se detectaba un ángulo en las zonas dos zonas contiguas: “Baja” y “Alta”, en un 50 % de las ocasiones se correspondía con la zona del ángulo real, y en los otros casos aparecía divergencia de una zona arriba o abajo. En último lugar, cuando el ángulo era detectado en las zonas “Muy baja” o “Muy alta”, en un 95,2 % de las ocasiones el ángulo se había detectado en la zona correcta, mientras que en una ocasión había un error de una zona contigua.

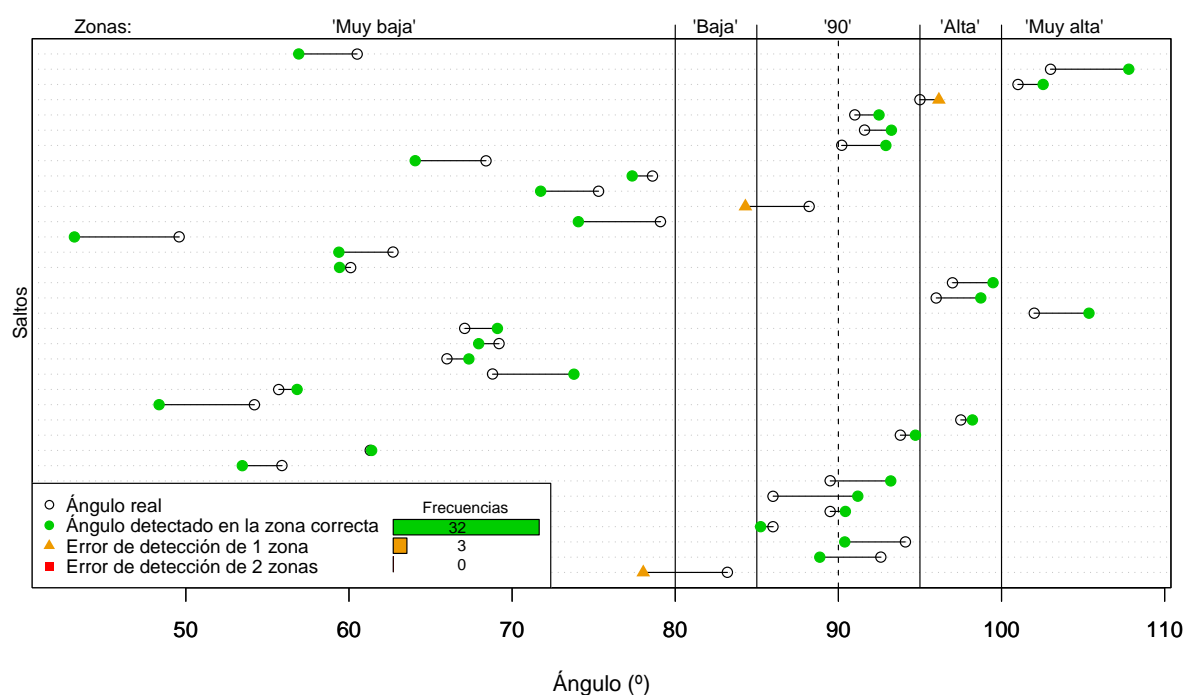


Figura 10.16: Error en la predicción en rangos de ángulos.

Así, la valoración cualitativa repercute en los mensajes que se pueden mostrar al usuario final. Se recomienda repetir el salto si se detecta un ángulo máximo de flexión en las zonas “Muy baja” o “Muy alta”. Se recomienda aceptar el salto cuando se detecta un ángulo en la zona “90”, siempre y cuando el resto de la ejecución sea correcto. Finalmente, en las zonas “Baja” y “Alta” se dejará a juicio del evaluador. En todos los casos se muestra el fotograma de máxima flexión, con el ángulo detectado y el error estimado.

El error encontrado se considera aceptable para el uso por parte de entrenadores. De este modo pueden saber el ángulo de flexión previo a un salto sin necesidad del uso de marcadores. Para entornos de investigación se recomienda el uso de sistemas de fotogrametría con marcadores, como los indicados en el estado actual de la cuestión.

### 10.3.3. Implementación del modelo en software de visión por computador

Fue necesario integrar el modelo, así como el conjunto de datos con los que había sido entrenado, en el software de visión por computador KneeAngle. Debido a que KneeAngle se había escrito en lenguaje C++, la biblioteca que se usó para incluir código R era R-Inside (Eddelbuettel, 2009).

En la figura 10.17 se observa la predicción del ángulo a partir de *tightsHP* (denominado *rectHP* en la actual versión del software) y *kpfY*. En dicha imagen se pueden apreciar los marcadores blancos en los puntos articulares, que fueron usados únicamente para comprobar el error en la predicción.



Figura 10.17: Predicción del ángulo en el programa de visión por computador.

## Parte IV

# Estudios de elaboración de tablas de percentiles

Existen herramientas informáticas que permiten compartir privadamente los datos de un test deportivo. Por ejemplo, SmartCoach posibilita la comunicación entre practicante y entrenador, y AthletJump entre usuario y el fabricante. Por otro lado, se han publicado documentos con datos de tests físico-deportivos en general (Generalitat de Catalunya, 1993; García, Sainz y Durán, 2011), y datos de saltos (Bosco, 1991; Bosco, 1994a; Bosco, 1994b; Garrido, 2004; Marina, 2003). Lamentablemente, no se han encontrado herramientas que unan la tecnología con la voluntad de publicación, de modo que los resultados de un test sean publicados para que puedan ser usados por cualquier persona interesada.

El objetivo principal de la presente parte es:

Desarrollar una herramienta libre que se integre con las anteriores y permita compartir datos entre evaluadores a fin de construir tablas de percentiles.

Cuando esté terminada la herramienta, los usuarios evaluarán tests en diferentes poblaciones. La compartición de los datos resultantes permitirá emitir juicios de valor cuando aparezcan nuevos. Así, cuando un practicante realice un salto determinado, su resultado se podrá comparar con otros del mismo género, edad, altura y peso, que además practican su mismo deporte al mismo nivel.

Los siguientes capítulos tratan el proceso a seguir:

Capítulo 11: Creación de criterios de fiabilidad de los datos.

Capítulo 12: Creación de un modelo de predicción.



---

## 11. Creación de criterios de fiabilidad de los datos

Una computadora puede ser llamada *inteligente* si logra engañar a una persona haciéndole creer que es un humano.

---

Alan Mathison Turing

A menudo nos preguntamos sobre lo fiable que es una fuente de información en la red. Este capítulo pretende definir criterios de fiabilidad de modo que se pueda etiquetar como *fiable* a un conjunto de datos de tests deportivos colgados en la red. Se pretende que cualquier usuario pueda decidir compartir datos en el gran conjunto de datos denominado *general* o en el subconjunto que se denominará: *fiable*. De este modo, se podrán utilizar estos datos, ya sea en el ámbito profesional, académico y científico.

### 11.1. Objetivos

Los objetivos que se plantean son:

- Definir criterios para diferenciar entre diferentes niveles de fiabilidad de datos según su origen.
- Establecer identificadores sobre datos erróneos según el propio dato.

### 11.2. Metodología

Para el primer objetivo se condujo un grupo de discusión para identificar los criterios de fiabilidad de datos. El segundo objetivo se persiguió a partir de revisión bibliográfica, la experiencia del doctorando y la observación de los datos que los evaluadores iban colgando.

A continuación se profundiza en la metodología relativa al primer objetivo.

#### 11.2.1. Formación del grupo de discusión

El grupo de discusión debía estar formado por profesores universitarios con la titulación de doctor y con experiencia en tests vinculados a la actividad física. También se consideró la proximidad entre los miembros como un criterio clave, pues no se deseaba añadir la problemática del desplazamiento a la ya difícil tarea de encontrar horarios comunes en la Universidad. Finalmente, se valoraron las competencias detectadas en relación a la capacidad de aportación y de diálogo en los actos académicos en que se haya podido participar con cada uno de ellos.

Después de evaluar los criterios citados, se contactó con tres personas. La invitación se realizó de forma individual con la finalidad de presentar el problema e invitarlos al grupo. En la invitación se evitó contaminar al participante con impresiones que pudieran influenciar su criterio. Los tres invitados aceptaron formar parte del grupo.

### 11.2.2. Primera reunión del grupo de discusión

El día 11 de marzo de 2009 se reunió el grupo por primera vez y se le pidió lo siguiente:

1. Encontrar unos indicadores (observables, operativos) que permitan discernir entre datos más y menos fiables.
2. Cuando estuvieran descritos los indicadores, determinar cuáles son necesarios y suficientes para considerar los datos como fiables.

La petición evolucionó en torno a los siguientes temas:

- Se generó debate sobre la fiabilidad de los datos aportados por publicidad general, industria farmacéutica y el vandalismo en páginas web como las de la Wikipedia.
- Se discutió en torno a si se debe evaluar a un investigador en concreto o al conjunto de datos que desea compartir.
- En cuanto al posible perfil de Doctor del evaluador, se realizó una crítica sobre la competencia de una persona con dicha titulación.
- Se comentaron los mínimos que debería pasar un conjunto de datos: Comité ético, participación voluntaria, consentimiento informado, utilidad de los datos (o del estudio), aportaciones novedosas.
- Se discutió alrededor de la figura del doble observador.

Finalmente se llegó a un consenso en que aparecieron dos figuras de evaluador: Evaluador-Entrenador y Evaluador-Investigador. Se apuntaron un conjunto de criterios de calidad para cada perfil. Además se enunció un listado de criterios de fiabilidad del proyecto.

Acabada la reunión, se envió el acta de la misma y se les convocó para un mes posterior.

### 11.2.3. Segunda reunión del grupo de discusión

El objetivo de la segunda reunión fue concretar los criterios acordados en la primera reunión. Se alcanzó un consenso con facilidad y se establecieron los criterios definitivos.



### 11.3. Resultados y discusión

#### 11.3.1. Relativos a la diferenciación entre niveles de fiabilidad de los datos según su origen

Se acordó que el conjunto de datos que se compartan en el servidor será etiquetado como *fiable* cuando el evaluador sea *fiable* y la sesión de evaluación sea *fiable*. Las tablas 11.1 y 11.2 recogen los criterios que se consideraron para la acreditación de un evaluador y una sesión de recogida de datos, respectivamente.

Tabla 11.1: Criterios para la acreditación de un evaluador como *fiable*.

Perfil Evaluador-Entrenador	Perfil Evaluador-Investigador
1. Titulación de Entrenador Nacional.	1. Máxima titulación conseguida.
2. Trabajo actual: club, institución, categoría.	2. ¿Ha realizado docencia en la Universidad?
3. Años en el entrenamiento de deportistas de élite.	3. Investigaciones que requieran de medición en el ámbito de las ciencias de la actividad física y el deporte.
4. Histórico de entrenador en cuanto a deportes y especialidades.	4. Otras investigaciones realizadas.
5. Experiencia en el uso de instrumental de medida.	5. Participación en grupos de investigación con vinculación universitaria.
6. Trayectoria profesional (100 palabras).	6. Becas obtenidas.
	7. Trayectoria profesional (100 palabras).

Si el evaluador tiene un perfil mixto, se le pedirá que responda a ambos perfiles.

Se decidió que un organismo debería encargarse de la evaluación de estos criterios. Se pidió al grupo de investigación SAFE: Salut, Activitat Física i Esport, perteneciente a la Facultat de Psicologia, Ciències de l'Educació i de l'Esport Blanquerna de la Universitat Ramon Llull, que se encargase de acreditar los conjuntos de datos que se compartirán en base a los criterios que se han definido. Los motivos de la petición son la coherencia con sus líneas de investigación y la participación de varios de sus miembros en el estudio. El grupo SAFE accede a desarrollar esta labor cuando el software esté preparado para tratar los formularios de criterios de fiabilidad.

Tabla 11.2: Criterios para la acreditación de una sesión de recogida de datos como *fiable*.

Criterios de fiabilidad de la sesión de recogida de datos
Consentimiento informado.
Calentamiento realizado.
Variaciones del protocolo de los tests (si las hay).
Nuevos protocolos de tests y variaciones (si las hay).
¿Los datos han sido publicados? Aportar referencia.
Justificación de la recogida de los datos.

### 11.3.2. Relativos al establecimiento de identificadores sobre datos erróneos según el propio dato

Tanto en el grupo general como en el fiable, se evaluó la calidad de los datos a partir de un conjunto de identificadores. Obviamente los datos no debían ser rechazados al momento sino que se debía considerar el grupo evaluado y se debía contactar con el evaluador para esclarecer las dudas. Los identificadores establecidos fueron:

- En cuanto a los saltos:
  - Tiempos de vuelo demasiado bajos en saltos sin carga extra:  $< 0,3$  s.
  - Tiempos de vuelo demasiado altos:  $> 0,8$  s.
  - Tiempos de contacto demasiado bajos:  $< 0,15$  s.
  - Tiempos de contacto demasiado altos en saltos sin carga extra:  $> 0,4$  s.
  - Valores de saltos squat o rocket superiores a los saltos en contramovimiento para los mismos sujetos.
  - Valores de saltos con contramovimiento superiores a los de Abalakov para los mismos sujetos. Para Lees y Barton (1996), la mejora en el rendimiento en un salto vertical cuando se usan libremente los brazos varía entre un 9,7 y un 14,8 % dependiendo del método utilizado en su estimación y cálculo.
  - Valores repetidos de saltos.
  - Resultados de las mujeres superiores a los hombres.

- En cuanto a los sujetos:
  - Los datos de los sujetos se encuentran incompletos.
  - Altura demasiado baja:  $< 0,5$  m.
  - Altura demasiado baja:  $> 2,1$  m.
  - Peso demasiado bajo:  $< 30$  Kg.
  - Peso demasiado alto:  $> 200$  Kg.
  - Dudas en cuanto al deporte de los practicantes. Por ejemplo, diversos evaluadores de Filipinas identificaron a sus saltadores como deportistas de Kiiking. El Kiiking es una modalidad deportiva popular en Estonia, pero no es relevante el número de adeptos en Asia.

Estos identificadores son dinámicos, de modo que con el tiempo se producirán cambios en los mismos y/o en sus valores límite.

## 12. Desarrollo de software para compartir datos

Todavía hay mucha gente que ofrece contenidos por el mero placer de saber que la información puede resultar útil a otras personas.

---

Vinton Cerf

Llegados a este punto, se dispone de herramientas para la captación y gestión de los datos, y además se han establecido criterios sobre su fiabilidad. El último paso es crear un software que permita compartir la información.

### 12.1. Objetivos

Los objetivos de este capítulo son los siguientes:

1. Crear un software servidor con licencia libre que se comunique bidireccionalmente con Chronojump para compartir datos.
2. Adaptar el software Chronojump para que se comunique con el software servidor.
3. Proteger la privacidad y seguridad de los datos compartidos.
4. Implementar en el software servidor estadísticas de los diferentes tests para distintas poblaciones.
5. Construir tablas de percentiles para facilitar la comparación de los resultados de un participante en una población en base a una o más variables.

### 12.2. Metodología

En referencia al primer objetivo, se procedió del siguiente modo:

1. Se reutilizaron las clases del software Chronojump excepto las que requieren de interfaz gráfica de usuario (GUI).
2. Se clasificaron las tablas y atributos de la base de datos en función de si eran propias del software cliente Chronojump, del software servidor o de ambas.
3. Se construyeron servicios web para recoger los datos y publicarlos.

4. Se inició una máquina virtual Xen dedicada para el proyecto. Dicha máquina ejerció de servidor gracias al software Apache que servía páginas web y al software servidor de Chronojump
5. El software se denominó Chronojump-Server.

Respecto al segundo objetivo, se crearon servicios web para cubrir la funcionalidad descrita, así como interfaces gráficas para acceder a dichos servicios.

En cuanto al tercer objetivo, se integró el *hashing* de contraseñas. Se decidió que los nombres o comentarios de los atletas no suben al servidor; no se muestra la fecha de nacimiento de los atletas ni el correo electrónico del evaluador. Tampoco se suben las consultas SQL para prevenir la inyección en las mismas.

El cuarto objetivo se resolvió escribiendo un script en R que se conectaba a la base de datos mediante RSQLite para cada estadístico que se deseaba generar. Los scripts sólo se ejecutan cuando hay nuevos datos y en un momento de baja carga para el servidor, y cada uno produce como resultado una gráfica en la página web del servidor.

Finalmente, la metodología a seguir referente al último objetivo fue la escritura de scripts en R para cada índice poblacional. Del mismo modo que para el objetivo anterior, los scripts generan gráficas en la página web del servidor, pero además, el software Chronojump puede pedir la creación de gráficos en que se posiciona a un participante en los datos del servidor. La generación de los gráficos para el participante es local: se envían los datos y no los gráficos.

### 12.3. Resultados y discusión

El software Chronojump-Server se ha creado y está funcionando en la máquina virtual Xen. En los apéndices A.3 y A.4 se puede consultar el estado actual de la base de datos y las clases, respectivamente. La última revisión del código específico del servidor es de sólo 564 líneas, valor muy bajo debido a que la mayoría de su funcionalidad está compartida con el software Chronojump. La figura 12.1 describe la arquitectura del servidor Xen para la funcionalidad de servidor de páginas web y de métodos de Chronojump.

En cuanto a la adaptación de Chronojump para la comunicación con Chronojump-Server, se han creado 27 servicios web (ver tabla 12.1).

Los servicios web han permitido que los usuarios subieran sus datos. Hasta el día 16 de febrero de 2012 se han subido datos de 751 personas. Se han compartido 3332 saltos simples y 130 saltos reactivos de estos practicantes. Estos datos han sido compartidos voluntariamente por 24 evaluadores.

Las figuras 12.2 y 12.3 muestran los evaluadores que han compartido datos y los sujetos que han realizado tests. La primera gráfica muestra dos vistas de los mismos datos: por un lado se distinguen las aportaciones de cada evaluador a partir de barras horizontales apiladas; por otro

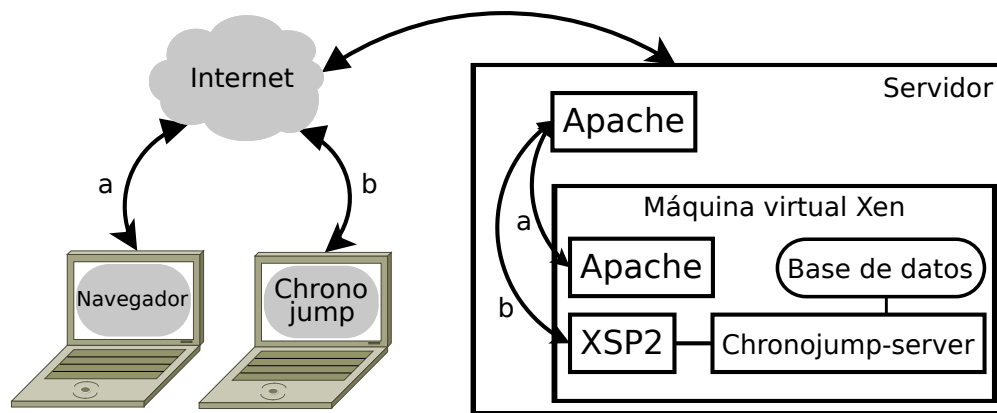


Figura 12.1: Diagrama de la integración de Chronojump-Server en un servidor con máquina virtual Xen.

Tabla 12.1: Servicios web creados en Chronojump-Server.

Relacionados con subir tests	Relacionados con subir tipos de test	Relacionados con subir otros datos	Otros servicios
UploadJump	UploadJumpType	UpdateSession	ConnectDatabase
UploadJumpRj	UploadJumpRjType	UploadSession	DisConnectDatabase
UploadRun	UploadRunType	UploadPerson	CanI
UploadRunI	UploadRunIType	UploadPSIfNeeded	CanINew
UploadRT		UploadSport	Query
UploadPulse		UploadPing	Stats
UploadMultiChronopic		UploadEvaluator	SelectEvaluators
		EditEvaluator	ListDirectory

se diferencian los registros de cada tipo de test en una línea de tiempo usando un sub-gráfico de líneas. En la progresión temporal se observan las aportaciones en dos periodos: primer trimestre de 2010 y segundo trimestre de 2011. El vacío temporal va asociado al desarrollo tecnológico del servidor. Éste comenzó a estar activo a mitades de 2009, pero los evaluadores comenzaron a compartir datos al inicio de 2010. Al poco tiempo se detectaron problemas en la organización de los datos que obligaron a desactivar la opción de subir datos, corregir los datos subidos, corregir la estructura de los datos en el servidor, publicar una nueva versión del software Chronojump que fuera compatible con los cambios, y volver a activar el servidor sólo para versiones de Chronojump actualizadas.

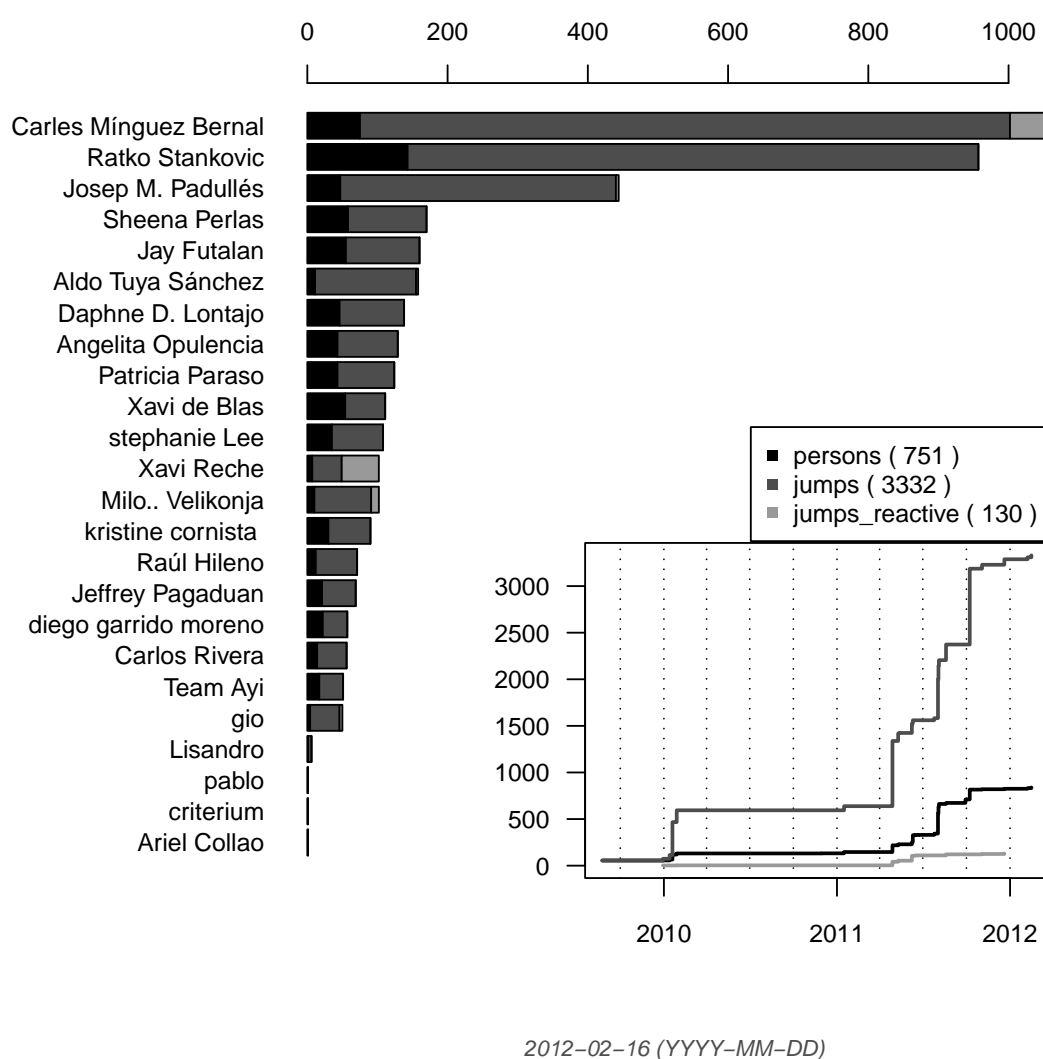
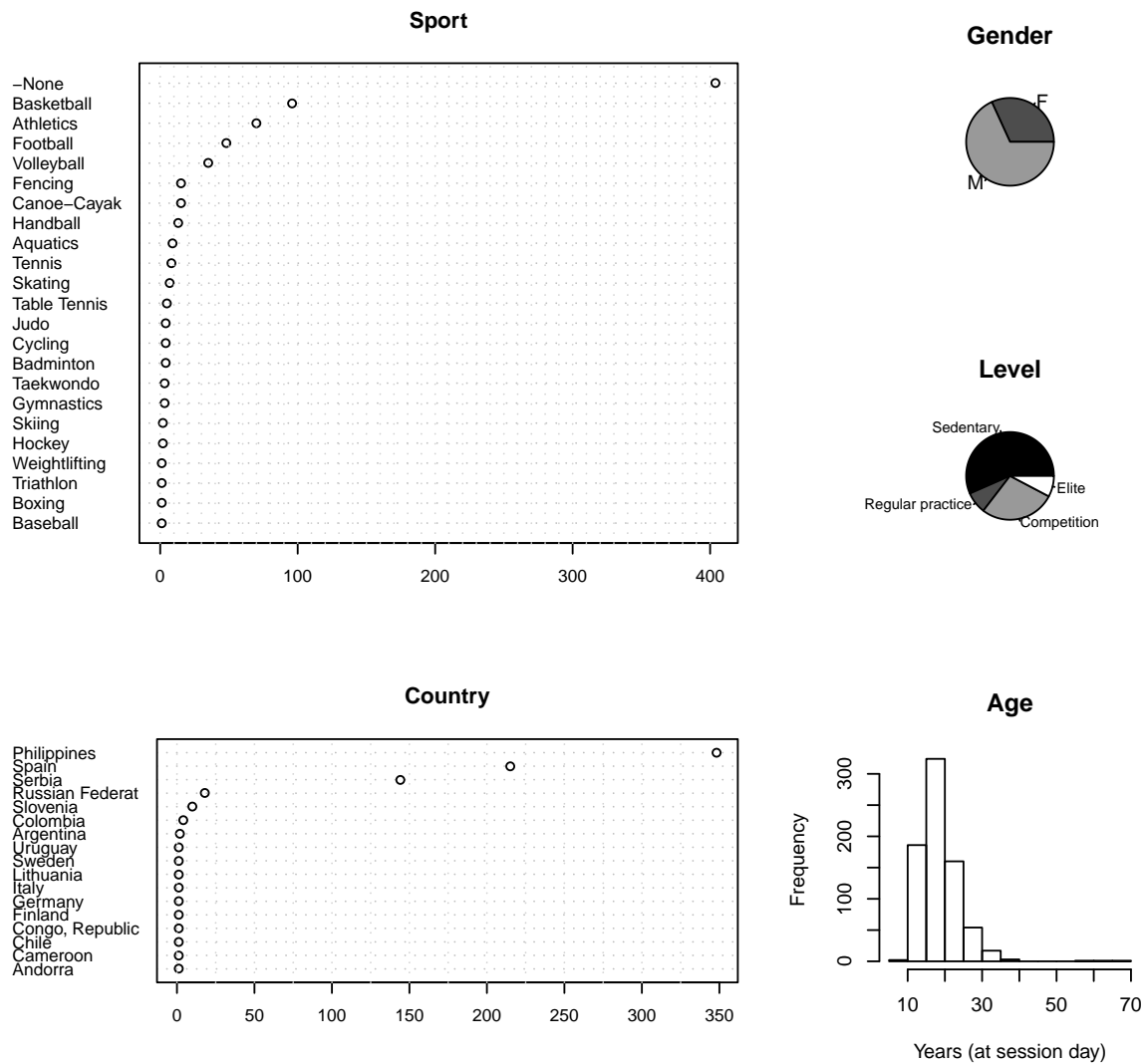


Figura 12.2: Gráficas del servidor. Contribuciones de los distintos evaluadores.

Las gráficas de las figuras citadas (12.2 y 12.3) se generan automáticamente en el servidor a partir de scripts. Además se han escrito otros scripts para la generación de gráficos de comparación de los distintos tests. En el apéndice B se incluyen dichos gráficos.



2012-02-16 (YYYY-MM-DD)

Figura 12.3: Gráficas del servidor. Personas que han realizado tests.



Como ejemplo de uso de los datos, la figura 12.4 muestra un uso del servicio web Query en que se realiza una consulta al servidor. La consulta realizada el día 19 de julio de 2011 pide los tiempos de vuelo en el salto Squat en mujeres que practican modalidades de atletismo a nivel de élite. El servicio web ofrece 53 resultados y la media de tiempo de vuelo es de 0,578 segundos.

**Consulta al servidor**  
Realizar consulta al servidor.

**Variables de la prueba**

Tipo de prueba: Jump simple

Test: SJ

Variable: TV

**Variables de personas**

Sexo: Mujeres

Edad: Cualquiera 1

Y: Any 1

Continente: Europa

País: España

Deporte: Atletico

Especialidad: Cualquiera

Nivel: 3:Élite

**Evaluador**

Evaluador: -1:Cualquiera

**Realizar consulta**

SELECT COUNT(TV), AVG(TV) FROM jump, person77, personSession77 WHERE jump.type = 'SJ' AND person77.sex

Buscar Resultados: 53 saltos  
Media 0,578 segundos

Cerrar

Figura 12.4: Captura de pantalla de Chronojump realizando consulta Query al servidor.

El servicio web UploadPing permite tener una idea del uso del software Chronojump. Cada vez que un usuario ejecuta el programa, éste trata de conectarse con el servidor y le envía una información muy breve. Esta señal de actividad se denomina ping. En este servicio concreto, se envían los siguientes datos: identificador del usuario (si éste está registrado), versión del programa, sistema operativo e identificador de instalación. El último parámetro pretende ser una forma distintiva para diferenciar entre los usuarios que no se hayan registrado. Los usuarios registrados son los que pueden compartir datos y por tanto deben elegir un nombre de usuario y correo electrónico únicos, mientras que para el resto, se usa un número generado aleatoriamente en la instalación del software que pretende ser un sistema de identificación razonablemente fiable. Inicialmente se usaba la IP del usuario, pero este valor es cambiante y a menudo compartido por diversos usuarios. A partir de marzo de 2010 se comenzó a usar un número aleatorio de 10 cifras como identificador de instalación. Desde marzo de 2010 a 6 de marzo de 2012 se han identificado 1085 códigos distintos, siendo la probabilidad de asignar un código ya existente a una nueva instalación de un 0,0001 %. No todas las ejecuciones del programa son registradas en el servidor, debido a que

puede ser que el usuario no esté conectado a Internet en el momento de arranque del programa, o esté detrás de un firewall poco permisivo o cancele la conexión con el servidor para acelerar el inicio del programa. En un futuro se almacenarán localmente esas conexiones infructuosas para enviarlas todas juntas cuando haya conexión. En el apartado 8.3.2 se muestran los resultados de la distribución e internacionalización del software Chronojump a partir de los datos obtenidos por UploadPing.

Respecto a la seguridad de las contraseñas, se ha usado BCrypt.net (Miller y Slager, 2006-2007) que implementa el código de hashing de contraseñas basado en Blowfish (Provos y Mazieres, 1999). Las ventajas del uso de este algoritmo en la protección de contraseñas son que se inicia lentamente, de modo que dificulta la fuerza bruta y que la computación requerida para probar una contraseña puede crecer de forma adaptativa tal y como aumentan las capacidades del hardware.

Por último, en cuanto a la generación de tablas de percentiles, se han dispuesto las herramientas, pero no se han publicado resultados porque se requieren muchos más datos para que la información sea relevante. La ausencia de datos se ejemplifica en la figura 12.5. En esta gráfica se relaciona el tiempo de vuelo con la altura del sujeto y se muestran percentiles del primero. La gráfica debe incluir además un icono que represente al practicante para que se ubique en la población.

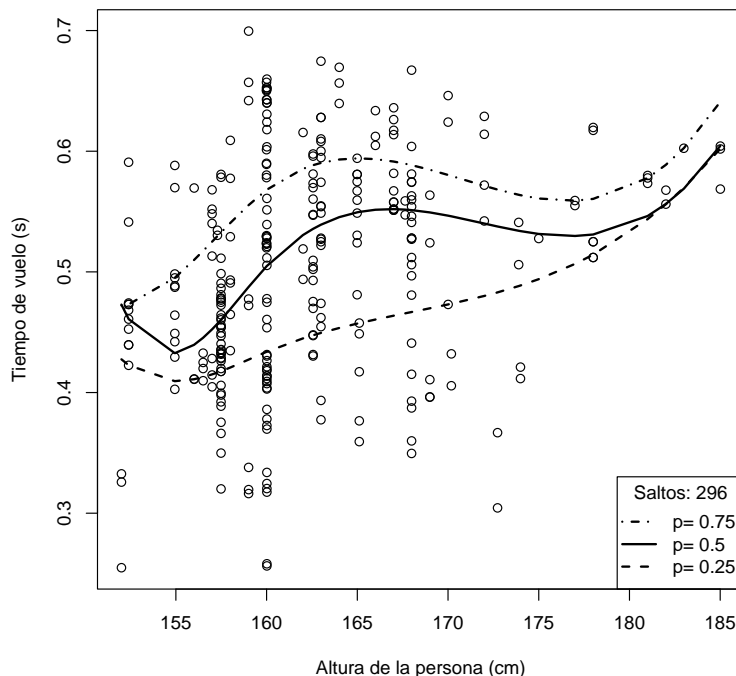


Figura 12.5: Gráfica correspondiente a una tabla de percentiles del salto CMJ en mujeres. Se examinan las variables: tiempo de vuelo y altura del sujeto.

## Parte V

# Conclusiones, limitaciones y líneas de futuro

En esta parte se especifican las conclusiones de la investigación junto a las limitaciones de la misma y se abren caminos para continuar el trabajo.



---

## 13. Conclusiones

Nadie debe empezar un proyecto grande. Empiezas con uno pequeño y trivial y nunca debes esperar que crezca; si lo haces solamente sobre-diseñarás y generalmente pensarás que es más importante de lo que lo es en esta etapa. O peor, puedes asustarte por el tamaño de lo que tu esperas que crezca. Así que empieza pequeño y piensa en los detalles. No pienses acerca de la foto grande y el diseño elegante. Si no resuelve una necesidad inmediata, seguramente está sobre-diseñado. Y no esperes que la gente salte a ayudarte, no es así como estas cosas funcionan. Primero debes tener algo medianamente usable y otros dirán *hey, esto casi funciona para mí* y se involucrarán en el proyecto.

---

Linus Torvalds

En referencia al primer objetivo:

Crear, validar y distribuir una herramienta de licencia libre que mida los tiempos de contacto y vuelo en el salto vertical, usando una plataforma de contactos.

Se ha creado un microcontrolador, plataformas de contactos de fibra de vidrio y un software de ordenador para su uso en la medición de tiempos en el salto vertical. Los tres productos se han diseñado usando software libre y el producto final es también libre y de bajo coste. Las herramientas han sido validadas con resultados satisfactorios. El software permite al usuario definir sus tests, así como usar tests predefinidos, entre los que se incluye el protocolo del Test de Bosco, otros tipos de saltos y algunos tests deportivos vinculados al tiempo, ya sean carreras, ritmos o tiempos de reacción. A parte de mostrar el resultado de un test, el software facilita un feedback configurable que permite alterar la duración de un test repetitivo. El programa se encuentra disponible en 7 idiomas y se ejecuta en dos sistemas operativos. Chronojump ganó el concurso internacional Trophées du Libre en el año 2007 en la categoría de Educación.

La información de construcción de todas las herramientas es pública, de modo que cualquiera puede producirlas. Como alternativa, se comercializan las herramientas, vendiéndose un microcontrolador cada 3 días. Hasta la fecha se han contabilizado 18 publicaciones ajenas al autor vinculadas a estas herramientas, 16 de ellas son de tipo científico.

En referencia al segundo objetivo:

Desarrollar y validar una herramienta libre para medir el ángulo de flexión de la articulación de la rodilla, previo a un salto CMJ, a partir de una filmación bidimensional y sin el uso de marcadores.

La herramienta ha sido diseñada y validada. Tal y como se esperaba, en la validación se observa que el error en la predicción es más bajo en la zona de 80-100 grados. En los ángulos alejados a dicha zona el error crece ligeramente, pero no lo suficiente como para que un salto con una flexión previa demasiado profunda (ángulo menor de 80º) o demasiado ligera (ángulo superior a 100º) sea confundida con una flexión adecuada: 85-95º. La herramienta se considera adecuada para el uso por parte de entrenadores.

Finalmente, respecto al tercer objetivo:

Desarrollar una herramienta libre que se integre con las anteriores y permita compartir datos entre evaluadores a fin de construir tablas de percentiles.

Se han creado criterios de fiabilidad de sesión (6 indicadores), de evaluador en función del perfil del mismo (6 y 7 ind.), y de los propios datos (14 ind.). El software servidor se ha desarrollado permitiendo conectar la obtención de datos usando instrumental propio, con la publicación abierta de los mismos. Dicho software se ha incluido en un servidor Xen y vinculado a Chronojump usando 27 servicios web. Hasta el 16 de febrero de 2012, se han identificado un total de 24 evaluadores que han compartido 3332 saltos simples y 130 saltos reactivos de un total de 751 personas. Se espera que los datos compartidos sigan creciendo.

A raíz de la obtención de la marca registrada Boscosystem, el proyecto pasa a denominarse Chronojump-Boscosystem. La interacción entre las herramientas diseñadas y las personas se ilustra en la figura 13.1. El logotipo de Chronojump-Boscosystem se incluye en la figura 13.2.

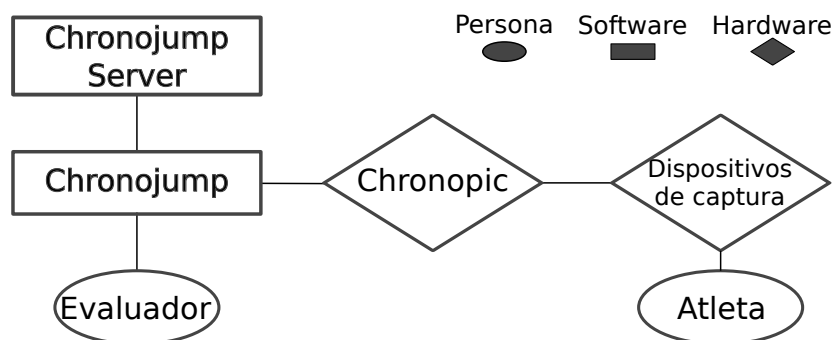


Figura 13.1: Interacción entre herramientas y personas en Chronojump-Boscosystem. En la ilustración se entiende que el software KneeAngle queda incluido dentro del Software Chronojump.



Figura 13.2: Logotipo de Chronojump-Boscosystem

## 14. Limitaciones

Los ordenadores son buenos siguiendo instrucciones, no leyendo tu mente.

---

Donald Knuth

En cuanto al software Chronojump, debido a que es un programa ligado a un hardware específico, el número de usuarios no es elevado. Este hecho dificulta la detección de errores. Se espera que la cantidad de usuarios vaya en aumento y éstos promuevan un incremento de las historias de usuario con propuestas de mejora y advertencias de errores, mejorando así el producto final.

Respecto a la creación del modelo de predicción del ángulo de flexión sin el uso de los marcadores, cabe recordar que dos terceras partes de los saltos registrados no pudieron ser analizados por problemas de encuadre de la imagen y deficiencias en la iluminación.

Continuando con el modelo de predicción, los problemas afectivos o incomodidades vinculadas al salto con las mallas o los pantalones cortos provocaron que el ángulo de flexión difiriera en función de la indumentaria. La validez externa de la detección sin marcadores se vió comprometida por estas diferencias.

Hasta la fecha no se han aplicado los criterios de fiabilidad de los datos de saltos en el servidor. Se ha preferido esperar a tener más datos antes de proceder a la petición de los criterios establecidos a cada uno de los evaluadores. Por tanto, aún no se ha etiquetado ningún dato como *fiable*.



---

## 15. Líneas de futuro

No me cuentes lo que vas a hacer. Me encantará verlo cuando esté terminado.

---

Juantomás García

### 15.1. En relación a los estudios de medición del tiempo

Publicar versión del software para la plataforma MacOSX, así como versiones reducidas para teléfonos inteligentes y tabletas bajo los sistemas operativos Android e IOS.

Aunque la plataforma de contactos está compuesta por fibra de vidrio y ésta es muy resistente, estudiar la durabilidad de la plataforma y desarrollar un método para verificar su fiabilidad.

Continuar con el desarrollo del software según las tareas que están descritas aquí en el Bugzilla<sup>7</sup> relativo al software (55 tareas el 6 de marzo de 2012).

Habiendo constado que el 80,3 % de los clientes de la tienda virtual son de España, promover la difusión de Chronojump-Boscosystem en otros países a partir de la vinculación con profesores universitarios. Esta expansión permitirá llegar a más usuarios y mejorar el soporte del programa en otros idiomas.

### 15.2. En relación a los estudios de detección sin marcadores del ángulo de flexión de la articulación de la rodilla

Para estudios futuros se recomienda ampliar la muestra y/o mejorar las condiciones de filmación para que el modelo pueda ser entrenado mejor y se obtenga un error más pequeño. Además, debido a que todos los scripts necesarios están creados, será muy rápido volver a analizar los datos en conjuntos mucho mayores.

Con la finalidad de solucionar el problema de la diferencia de flexión en función de la indumentaria, en estudios posteriores se pedirá a los sujetos que rellenen un formulario sobre el uso de las indumentarias, y se trabajará en la detección de las piernas con el uso de pantalón corto y piel.

Calcular el ángulo de flexión tridimensional a partir de una filmación lateral (bidimensional) con el esquema que se expone en la figura 15.1. Validar este método usando filmación tridimensional.

Comparar la validez concurrente de KneeAngle con la de observadores expertos. Estos observadores estimarían el ángulo de flexión previo al salto de forma directa.

Integrar KneeAngle en el software Chronojump de modo que pueda ser usado fácilmente.

---

<sup>7</sup>[https://bugzilla.gnome.org/buglist.cgi?product=chronojump&bug\\_status=UNCONFIRMED&bug\\_status=NEW&bug\\_status=ASSIGNED&bug\\_status=REOPENED&component=chronojump](https://bugzilla.gnome.org/buglist.cgi?product=chronojump&bug_status=UNCONFIRMED&bug_status=NEW&bug_status=ASSIGNED&bug_status=REOPENED&component=chronojump)

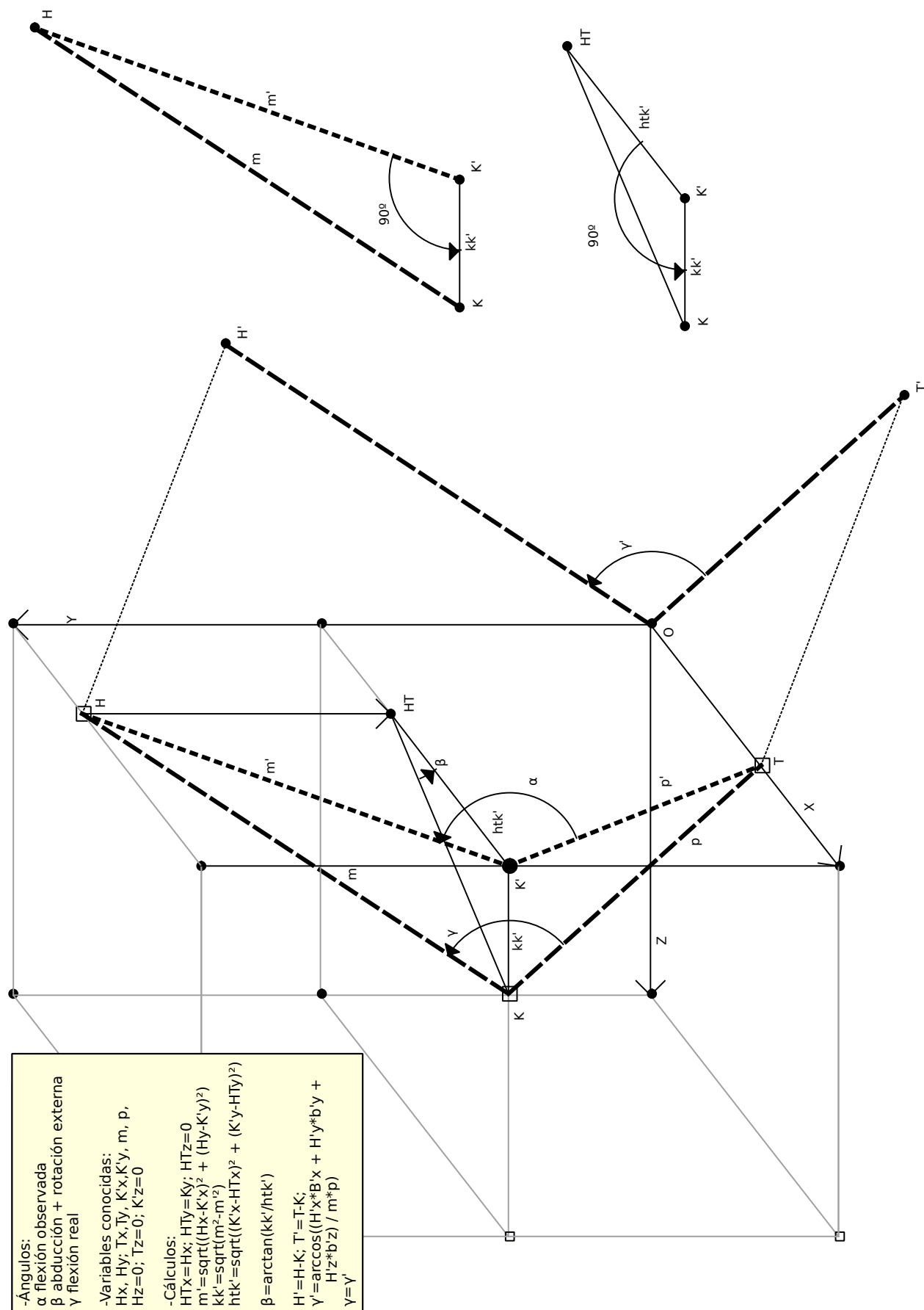


Figura 15.1: Cálculo del ángulo de flexión de la rodilla en tres dimensiones.

### **15.3. En relación a los estudios de elaboración de tablas de percentiles**

Automatizar el proceso de etiquetado de datos fiables, de modo que se faciliten los formularios a los evaluadores y se contrasten los datos.

Promover la obtención y publicación de datos de saltos en diferentes poblaciones para favorecer la creación de una base de datos universal.

### **15.4. En relación a otras aplicaciones que se deriven de las tecnologías desarrolladas**

Las tecnologías desarrolladas se usan también para la gestión de carreras, tiempos de reacción, ritmos y otros tests que puedan requerir uno o más microcontroladores (multicanal). Esta tesis sólo se ha centrado en la aplicación relativa a los saltos, pero en paralelo se han desarrollado las funcionalidades para los otros tests. A continuación, se nombran tareas a desarrollar vinculadas a dichos tests:

- Relativas a los tests existentes:
  - Mejorar la señal de Chronopic en su uso, como el tiempo de reacción añadiendo un zumbador (buzzer) y una luz de mayor tamaño.
  - Diseñar un esquema eléctrico para la sincronización de varios Chronopics.
  - Diseñar un esquema eléctrico tipo AND para el uso de varias fotocélulas como una sola, y así evitar dobles contactos en una carrera.
- Relativas a nuevos tests: Diseñar, fabricar, validar y comercializar...
  1. encoder lineal
  2. plataforma de contactos de infrarrojos
  3. plataforma de fuerzas
  4. galga extensiométrica
  5. acelerómetro

Para estas aplicaciones será necesario:

1. Adaptar el microcontrolador Chronopic 3 para que pueda funcionar a mayor velocidad. Actualmente opera a 4 MHz, siendo 20 MHz el máximo que tolera el PIC .

2. Explorar los dispositivos existentes en distintos países. En algunos casos será conveniente crear un dispositivo nuevo, mientras que en otros casos será suficiente con adaptarlos al Chronopic.
3. Programar el firmware de Chronopic para que reciba datos de los dispositivos.
4. Programar el software Chronojump para que gestione los tests y sus datos.

En mayo de 2012, Chronojump-Boscosystem se une a la Fundación CIDIDA (Centro Internacional de Documentación, Investigación y Desarrollo del Atletismo) con sede en España, y el NCHC (National Center for High-Performance Computing) de Taiwán, en un proyecto de 3 años con el objeto de crear tecnologías libres para la medición en el deporte. Este proyecto pretende ayudar a desarrollar las aplicaciones descritas anteriormente. Algunas de las tareas ya han sido desarrolladas, como la conversión de Chronopic a 20 Mhz y la traducción del firmware a lenguaje C para que pueda crecer en funcionalidades de forma más rápida. Estas dos tareas requieren de una nueva validación que se llevará a cabo más adelante.

### **15.5. En relación al uso de las herramientas en la docencia**

En la Universitat de Vic, Josep Bauzà y Carles Cladera han realizado un proyecto de fin de grado que concluye en la creación de diez prácticas docentes para el aprendizaje de conceptos vinculados a la fuerza y la potencia a partir del uso de Chronojump-Boscosystem. Las prácticas repasan los conceptos fisiológicos, asisten en el uso de la tecnología, desarrollan una solución a un problema e incluyen un cuestionario de evaluación. Además, han creado vídeos explicativos sobre la ejecución de los saltos del test de Bosco. Todo el trabajo estará disponible en inglés a finales de junio de 2012 para que pueda ser incorporado al programa informático y traducido por los GNOME translators.

Parte VI

## Referencias bibliográficas



## Referencias bibliográficas

- Adolf, F. (sep. de 2003). *OpenCV's Rapid Object Detection*. Recuperado de [http://robotik.inflomatik.info/other/opencv/OpenCV\\\_ObjectDetection\\\_HowTo.pdf](http://robotik.inflomatik.info/other/opencv/OpenCV\_ObjectDetection\_HowTo.pdf).
- Aguado, X. (1993). *Eficacia y técnica deportiva*. Barcelona: INDE.
- Aguado, X., Izquierdo, M. y González, J.L. (1999). «Biomecánica de la fuerza muscular y su valoración: Análisis cinético de la marcha, natación, gimnasia artística, bádminton y ejercicios de musculación». En: ed. por A Ferro. Vol. ICD 21. Madrid: Ministerio de Educación y Cultura, Consejo Superior de Deportes. Cap. Consideraciones sobre conceptos y clasificaciones de la fuerza muscular desde el punto de vista mecánico, págs. 7-26.
- Ahonen, J. y col. (2001). *Kinesiología y anatomía aplicada a la actividad física*. Barcelona: Paidotribo.
- Alder, B.T. y col. (2008). «Assigning trust to Wikipedia content». En: *Proceedings of the 4th International Symposium on Wikis*. ACM, pág. 26.
- Álvarez-Herms, J. y col. (2011). Changes in heart rate recovery index after a programme of strength/endurance training in hypoxia. En: *Apunts. Medicina de l'Esport*. DOI: 10.1016/j.apunts.2011.07.003. Recuperado de <http://www.apunts.org/ficheros/eop/S1886-6581%2811%2900040-5.pdf>.
- Arduino (n.d.). *Arduino. Preguntas frecuentes*. Recuperado de <http://arduino.cc/es/Main/FAQ>.
- Asundi, J. (2001). «Software engineering lessons from open source projects». En: *Making Sense of the Bazaar: Proceedings of the 1st Workshop on Open Source Software Engineering*. Feller, J., Fitzgerald, B. & van der Hoek, A.(eds).
- Axon (2004). *Axon Jump. Manual del Usuario*. Recuperado de <http://www.ideadeportiva.com/download/Manual%20del%20Usuario%20Axon%20Jump.pdf>.
- (n.d.). Axon Jump [Software]. Recuperado de <http://www.axonjump.com.ar/>.
- Bakker, J.D. (2000). *Licencia de LART Linux Advanced Radio Terminal*. Recuperado de <http://www.lartmaker.nl/LICENSE>.
- Balloonboard.org (2003). *Balloon Licence*. Recuperado de <http://balloonboard.org/licence.html>.
- Beal, D. y col. (2003). Linux as a Real Time Operating System. En: *Metrowerks Corporation—A Motorola Company, OCERA*.

- Beard, S. y col. (2002). «The Ultracam camera control and data acquisition system». En: *Proceedings of SPIE, Boston, MA, August*, págs. 27-28.
- Beck, K. (2000). *Extreme programming explained: embrace change*. 1.<sup>a</sup> ed. Addison-Wesley Professional.
- Bij, E. van der (2011). *CERN Open Hardware Licence - Introduction*. Recuperado de <http://www.ohwr.org/projects/cernohl/wiki>.
- Black Duck Software, Inc. (2012). *Ohloh: Chronojump estimated cost*. Recuperado de [http://www.ohloh.net/p/10015/estimated\\_cost](http://www.ohloh.net/p/10015/estimated_cost).
- BlackDuck Software Knowledge Base (2012a). *Language Breakdown in Open Source*. Recuperado de <http://www.blackducksoftware.com/oss/projects#language>.
- (2012b). *Top 20 Most Commonly Used Licenses in Open Source Projects*. Recuperado de <http://www.blackducksoftware.com/oss/licenses>.
- Bobbert, M. F. y col. (1996). Why is countermovement jump height greater than squat jump height? En: *Medicine and science in sports and exercise* 28, págs. 1402-1412. DOI: 10.1097/00005768-199611000-00009.
- Bobbert, M.F. (1990). Drop jumping as a training method for jumping ability. En: *Sports Med* 9.1, págs. 7-22. DOI: 10.2165/00007256-199009010-00002.
- Boehm, B.W. (1984). Software engineering economics. En: *Software Engineering, IEEE Transactions* 1, págs. 4-21. DOI: 10.1109/TSE.1984.5010193.
- Boehm, B.W. y col. (2000). *Software Cost Estimation with Cocomo II with Cdrom*. Prentice Hall PTR, Upper Saddle River, NJ.
- Bosco (1986). The effect of extra-load conditioning on muscle performance in athletes. En: *Med Sci Sports Exerc* 18.8, págs. 415-419. DOI: 10.1249/00005768-198608000-00009.
- Bosco, C. (1987). Valoraciones funcionales de la fuerza dinámica explosiva y de la potencia anaeróbica aláctica con el test de Bosco. En: *Apuntes de medicina del deporte* 24, págs. 151-158. Recuperado de [http://www.apunts.org/apunts/ctl\\_servlet?\\_f=60&ident=13104976](http://www.apunts.org/apunts/ctl_servlet?_f=60&ident=13104976).
- (1991). *Aspectos fisiológicos de la preparación física del futbolista*. Barcelona: Paidotribo.
- (1994a). *La preparazione fisica nella Pallavolo femminile*. Società Stampa Sportiva.
- (1994b). *La valoración de la fuerza con el test de Bosco*. Deporte y entrenamiento. Barcelona: Paidotribo.



- Bosco, C., Luhtanen, P. y Komi, P.V. (1983). A simple method for measurement of mechanical power in jumping. En: *European Journal Applied Physiology Occupational Physiology* 51.3, págs. 357-364. DOI: 10.1007/BF00422166.
- Bosco, C. y col. (1983). Mechanical power test and fiber composition of human leg extensor muscles. En: *Eur J Appl Physiol Occup Physiol* 51.1, págs. 129-35. DOI: 10.1007/BF00952545.
- Bradski, G. R. y Davis, J. W. (2002). Motion segmentation and pose recognition with motion history gradients. En: *Mach. Vision Appl.* 13.3, págs. 174-184. DOI: 10.1007/s001380100064.
- Bradski, G. (2011). *OpenCV Wiki*. Recuperado de <http://opencv.willowgarage.com/wiki/>.
- Bradski, G. y Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media.
- Buscà, B., De Blas, X. y Daza, E. (2004). *Instrucciones para la construcción de una plataforma de contactos para la medición de la capacidad de salto/s*. Recuperado de [http://chronojump.org/construction\\_contact\\_platform\\_es.html](http://chronojump.org/construction_contact_platform_es.html).
- Buscà, B. y Font, A. (2011). A low-cost contact system to assess load displacement velocity in a resistance training machine. En: *Journal of Sports Science and Medicine* 10, págs. 472-477. Recuperado de <http://www.jssm.org/vol10/n3/8/v10n3-8text.php>.
- CadSoft (n.d.). EAGLE [Software]. Recuperado de <http://www.cadsoftusa.com/>.
- Calleja, M.A. y Riesco, Á.M. (2007). *Análisis, diseño y mantenimiento del software*. Universidad Nacional de Educación a Distancia.
- Candeira, J. (2001). La web como memoria organizada: el hipocampo colectivo de la red. En: *Revista de Occidente* 239, págs. 87-113.
- Canonical Ltd (n.d.). Bazaar [Software]. Recuperado de <http://bazaar.canonical.com/en/>.
- Cardinale, M. (2010). *Low cost vertical jump tests solution*. Recuperado de <http://marcocardinale.blogspot.com/2010/11/low-cost-vertical-jump-tests-solution.html>.
- Caruso, D (feb. de 1999). *Patent Absurdity*. New York Times. Recuperado de <http://www.nytimes.com/library/tech/99/02/biztech/articles/01digi.html>.
- Casartelli, N., Müller, R. y Maffiuletti, N.A. (2010). Validity and reliability of the myotest accelerometric system for the assessment of vertical jump height. En: *Journal of Strength and Conditioning Research* 24.11, pág. 3186. DOI: 10.1519/JSC.0b013e3181d8595c.
- Charmant, J. (n.d.). Kinovea [Software]. Recuperado de <http://www.kinovea.org>.

- Chen, T.P. y col. (2005). Computer Vision Workload Analysis: Case Studio of Video Surveillance Systems. En: *Intel Technology Journal* 9.2, págs. 108-118.
- Chronojump (n.d.). *GIT de Chronojump*. Recuperado de <http://git.gnome.org/browse/chronojump/tree>.
- Church, J.B. y col. (2001). Effect of warm-up and flexibility treatments on vertical jump performance. En: *Journal of strength and conditioning research* 15.3, pág. 332. DOI: 10.1519/00124278-200108000-00012.
- Clanton, S. (2002). Embedded Real-Time Linux for Instrument Control and Data Logging. En: *RECON* 20020063582. Recuperado de <http://www.linuxjournal.com/article/5667>.
- Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Profesional.
- Corbet, J. (2007). Who wrote 2.6.20. En: *LWN. net*. Recuperado de <http://lwn.net/Articles/222773/>.
- Creative Commons (2011). *The Power of Open*. Creative Commons. Recuperado de <http://thepowerofopen.org/>.
- Cua, M. y Lim-Martinez, S. (2011). «Comparison between Leg Power Output of High School and Collegiate Female Cheerleaders». En: *International Journal of Exercise Science: Conference Abstract Submissions*. Vol. 5. 2, pág. 19.
- De Blas, X. (2004-2011). *Manual de Chronojump en castellano*. Recuperado de [http://chronojump.org/docs/chronojump\\_manual\\_es.pdf](http://chronojump.org/docs/chronojump_manual_es.pdf).
- (2011). *Chronojump Manual (english)*. Recuperado de [http://chronojump.org/docs/chronojump\\_manual\\_en.pdf](http://chronojump.org/docs/chronojump_manual_en.pdf).
- De Blas, X., González-Gómez, J. y Gómez, R. (2009). *Validity of new Chronopic V.3 Open Hardware to Measure Time on Jump related Tests*. Poster session presented at the 09 Annual meeting of American College of Sports Medicine, Seattle, USA. Recuperado de <http://chronojump.org/posters.html>.
- De Blas, X. y González-Gómez, J. (2005). Proyecto Chronojump: Sistema de Medida de la Capacidad de Salto usando Software y Hardware Libres. En: *Actas I Congreso de Tecnologías del Software Libre*.
- De Blas, X., Morales, A. y Olea, I. (2009). «Integración de R en software libre de análisis del rendimiento deportivo multiplataforma y en entorno cliente-servidor». Primeras Jornada de usuarios de R. Murcia.

- Debian (n.d.). *Debian free software guidelines*. Recuperado de [http://www.debian.org/social\\_contract.html#guidelines](http://www.debian.org/social_contract.html#guidelines).
- Departamento de Defensa de los Estados Unidos (2009). *Clarifying Guidance Regarding Open Source Software (OSS)*. Recuperado de <http://cio-nii.defense.gov/sites/oss/20090SS.pdf>.
- DiBona, C. (2011). *Google, NASA and Open Source*. Recuperado de <http://www.slideshare.net/ckleclerc/2011-nasa-open-source-summit-chris-dibona>.
- Dumbill, E. y Bornstein, N.M. (2004). *Mono: A Deveper's Notebook*. O'Reilly Media.
- Durso, R. (jun. de 2005). Asexual Reproduction of *Catalaphyllia jardinei* (Elegance coral). En: *Reefkeeping* 4.5. Recuperado de <http://reefkeeping.com/issues/2005-06/nftt/index.php>.
- Eddelbuettel, D. (2009). R-Inside (Version 0.2.3) [Software]. Recuperado de <http://dirk.eddelbuettel.com/code/rinside.html>.
- Edgewall Software (n.d.). Trac [Software]. Recuperado de <http://trac.edgewall.org>.
- Esper, A. (sep. de 2000). Influencia de los distintos tipos de calentamientos musculares sobre la saltabilidad. En: *Lecturas, Educación Física y Deportes, Revista Digital* 25. Recuperado de <http://www.efdeportes.com/efd25a/calen.htm>.
- Flickr Blog (oct. de 2009). *4,000,000,000*. Recuperado de <http://blog.flickr.net/en/2009/10/12/4000000000/>.
- Fogel, K. (2005). *Producing open source software: How to run a successful free software project*. O'Reilly Media, Inc. Recuperado de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.9848&rep=rep1&type=pdf>.
- Fowler, M. y Highsmith, J. (2001). The agile manifesto. En: *Software Development* 9.8, págs. 28-35.
- FPGA Libre (n.d.). *FPGA Libre Hardware*. Recuperado de <http://fpgalibre.sourceforge.net/hard.html>.
- Gómez, R. y González, J. (2008). *Página técnica de Chronopic 3*. Recuperado de [http://www.iearobotics.com/wiki/index.php?title=Chronopic\\_3.0](http://www.iearobotics.com/wiki/index.php?title=Chronopic_3.0).
- Galli, R. (2004). *¿De qué viviremos los informáticos?, y tú ¿por qué no enseñas gratis?* Recuperado de <http://bulma.net/body.phtml?nIdNoticia=1961>.
- García, J., Sainz, A. y Durán, J.P. (2011). *Análisis estadístico de jóvenes atletas de 14 a 17 años en el periodo 1997-2008*. Real Federación Española de Atletismo.

García-Fogeda, A. (2001). «Valoración de las características mecánicas del salto de detente vertical en función de referencias externas». Tesis doct. Universitat de Barcelona.

Garrido, R.P. (nov. de 2004). Test de Bosco. Evaluación de la potencia anaeróbica de 765 deportistas de alto nivel. En: *Lecturas, Educación Física y Deportes, Revista Digital* 78.

Gartner (2011). *Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent*. Recuperado de <http://www.gartner.com/it/page.jsp?id=1848514>.

Gbdirect (n.d.). *Benefits of Using Open Source Software*. Recuperado de <http://open-source.gbdirect.co.uk/migration/benefit.html>.

Generalitat de Catalunya, ed. (1993). *La Bateria Eurofit a Catalunya*. 1.<sup>a</sup> ed. Barcelona: Departament de la Presidència. Direcció General de l'Esport.

GNOME Translators (2012). *Equipos de traducción del GNOME*. Recuperado de <http://110n.gnome.org/teams/>.

Goldsmith, Marshall (1999). Comunidades globales y comunidades de elección. En: *La comunidad del futuro*. Ed. por Peter F. Drucker. Barcelona: Granica, págs. 131-146.

Gomila-Serra, B. y col. (2011). Evaluación de los sistemas de medición de la fuerza muscular en el ámbito de la práctica musical. Aproximación a un prototipo apto para la práctica instrumental. En: *Trauma* 22.4, págs. 226-234.

González-Barahona, J.M. (2003). *El software como servicio. O de cómo producir programas libres y no morir en el intento*. Recuperado de <http://sinetgy.org/~jgb/articulos/software-servicio/>.

González-Barahona, J.M., Seoane, J. y Robles, G. (2003). *Introducción al software libre*. UOC.

González-Barahona, J.M. y col. (2001). Contando Patatas: El tamaño de Debian 2.2. En:

González-Gómez, J. y Prieto-Moreno, A. (2005). *Hardware Libre: la Tarjeta Skypic, una Entrenadora para Microcontroladores PIC*. I Congreso de Tecnologías de Software Libre. Facultad de Informática, A Coruña. Recuperado de <http://www.iearobotics.com/personal/juan/publicaciones/art9/index.html>.

González, I., González, J. y Gómez-Arribas, F. (sep. de 2003). *Hardware libre: clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux*. Escuela Politécnica Superior, Universidad Autónoma de Madrid. Recuperado de <http://www.iearobotics.com/personal/juan/publicaciones/art4/index.html>.

- Graham, P. (2010). *Web 2.0 And Why?* Recuperado de <http://fossbox.org.uk/blog/wp-content/uploads/2010/03/1-what-why-how.pdf>.
- Gutiérrez, M. (1999). *Biomecánica deportiva*. Madrid: Editorial Síntesis.
- Hamano, J. y Torvalds, L. (n.d.). GIT [Software]. Recuperado de <http://git-scm.com/>.
- Hay, J.G. (1978). *The biomechanics of sport techniques*. London: Prentice-Hall, Inc.
- Herbsleb, J.D. y Grinter, R.E. (1999). «Splitting the organization and integrating the code: Conway's law revisited». En: *Proceeding of the 21st International Conference on Software Engineering*. Los Angeles: Published by the IEEE Computer Society.
- Himanen, P. (2001). *La ética del hacker y el espíritu de la era de la información*. Destino.
- Holwerda, T. (2009). *Ballmer: Linux Bigger Competitor than Apple*. Recuperado de [http://www.osnews.com/story/21035/Ballmer\\_Linux\\_Bigger\\_Competitor\\_than\\_Apple](http://www.osnews.com/story/21035/Ballmer_Linux_Bigger_Competitor_than_Apple).
- IBV (n.d.). AthletJump/IBV [Software]. Instituto Biomecánico de Valencia. Recuperado de [http://www.ibv.org/index.php/es/productos/aplicaciones-biomecanicas/show\\_product/99/223](http://www.ibv.org/index.php/es/productos/aplicaciones-biomecanicas/show_product/99/223).
- Ingen Schenau, G. J. van, Bobbert, M.F. y Rozendal, R.H. (1987). The unique action of biarticular muscles in comple movements. En: *Journal of Anatomy* 155, págs. 1-5.
- Ingen Schenau, G.J. van y col. (1992). The constrained control of force and position in multi joint mevements. En: *Neuroscience* 46, págs. 197-207.
- Izquierdo, M. (2008). *Biomecánica y Bases Neuromusculares de la Actividad Física y el Deporte*. Madrid: Editorial Médica Panamericana, S.A.
- Jaric, S., Ristanovic, D. y Corcos, D. M. (1989). The relationship between muscle kinetic parameters and kinematic variables in a complex movement. En: *Eur J Appl Physiol Occup Physiol* 59.5, págs. 370-6. DOI: 10.1007/BF02389813.
- Jefferson, T. (1813). Letter to Isaac McPherson. En: *The Writings of Thomas Jefferson*. Recuperado de <http://odur.llet.rug.nl/~usa/P/tj3/writings/brf/jef1220.htm>.
- Jeffries, R. (ago. de 2001). Essential XP: Card, Conversation, and Confirmation. En: *XP Magazine*.
- Jiang, S. y col. (2007). Detection of molecular particles in live cells via machine learning. En: *Cytometry Part A* 71.8, págs. 563-575. DOI: 10.1002/cyto.a.20404.
- Joson, R. y col. (2011). Relationship of upper and lower body power to swimming kinematics of male collegiate swimmers. En: *Journal of Sports Science and Health* 13 (supp 1). Presentado

en 3rd International Conference on Sport and Exercise Science, February 7-10, Imperial Queen's Park Hotel, Bangkok, Thailand, pág. 60.

Joyner, David y Stein, William (2007). Open Source Mathematical Software. En: *Notices of the American Mathematical Society* 54.10, pág. 1279.

Julián, P. (2007). *Programación Declarativa*. Recuperado de [http://www.inf-cr.uclm.es/www/pjulian/teaching/sl\\_apPD.pdf](http://www.inf-cr.uclm.es/www/pjulian/teaching/sl_apPD.pdf).

Kalamen, J. (1968). «Measurement of maximum muscular power in man». Tesis doct. Ohio State University Columbus.

KDE Project (n.d.). KDE [Software]. Recuperado de <http://www.kde.org>.

KDE Translators (2012). *KDE translation teams*. Recuperado de <http://l10n.kde.org/teams-list.php>.

Kibele, A. (1998). Possibilities and limitations in the biomechanical analysis of countermovement jumps: a methodological study. En: *Journal of Applied Biomechanics* 14, págs. 105-117.

KiCad Developers (n.d.). KiCad [Software]. Recuperado de [http://kicad.sourceforge.net/wiki/Main\\_Page](http://kicad.sourceforge.net/wiki/Main_Page).

Kidman, A. (2010). *75 % of Linux code now written by paid developers*. Recuperado de <http://apcmag.com/linux-now-75-corporate.htm>.

Knudson, D. (1999). Stretching during warm-up: do we have enough evidence? En: *Journal of Physical Education Recreation and Dance* 70, págs. 24-27.

Knudson, D. y col. (2001). Acute effects of stretching are not evident in the kinematics of the vertical jump. En: *Journal of Strength and Conditioning Research* 15, págs. 98-101. DOI: 10.1519/00124278-200102000-00017.

Kohno, T. y col. (2004). «Analysis of an electronic voting system». En: *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, págs. 27-40.

Komi, P.V. y Gollhofer, A. (1999). Stretch reflexes can have an important role in force enhancement during SSC exercise. En: *Journal of Applied Biomechanics* 13, págs. 451-460.

Korfiatis, N. T., Poulos, M. y Bokos, G. (2006). Evaluating authoritative sources using social networks: an insight from Wikipedia. En: *Online Information Review* 30, págs. 252-262. DOI: 10.1108/14684520610675780.

Kreighbaum, E. y Barthels, K.M. (1981). *Biomechanics: a qualitative approach for studying human movement*. Minnesota, USA: Burgess Publishing Company.

- Landis, J.R. y Koch, G.G. (1977). The measurement of observer agreement for categorical data. En: *Biometrics*, págs. 159-174. DOI: 10.2307/2529310.
- Laprévôte, A. (2009). *Free software business models*. Presentado en fOSSa Free/OSS & Academia Conference. Grenoble. Recuperado de <http://slidesare.net/fossaconference/oss-business-models>.
- Lara, A. J. (2007). «Biomecánica de la arquitectura muscular y potencia mecánica de salto en jóvenes». Tesis doct. Toledo: Universidad de Castilla-La Mancha.
- Ledvina, B., Mota, F. y Kintner, PM (2000). «A coming of age for gps: A rtlinux based gps receiver». En: *Proc. of the Workshop on Real Time Operating Systems and Applications and Second Real Time Linux Workshop (in conjunction with IEEE RTSS 2000), Orlando, Florida*.
- Lees, A. y Barton, G. (1996). The interpretation of relative momentum data to assess the contribution of the free limbs to the generation of vertical velocity in activities. En: *Journal of Sports Sciences* 14.6, págs. 503-511. DOI: 10.1080/02640419608727737.
- Lessig, L. (2003). *The future of ideas*. Random House New York.
- Linthorne, N.P. (2001). Analysis of standing vertical jumps using a force platform. En: *American Journal of Physics* 69, pág. 1198. DOI: 10.1119/1.1397460. Recuperado de [http://dspace.brunel.ac.uk/bitstream/2438/1392/3/Standing+Vertical+Jump+\(Linthorne\).pdf](http://dspace.brunel.ac.uk/bitstream/2438/1392/3/Standing+Vertical+Jump+(Linthorne).pdf).
- López, J.A.P. y Xirgo, L.R. (2004). *Introducción al desarrollo del software*. UOC.
- López, J.L., Padullés, J.M. y Olsson, H.J. (2011). Biomechanical analysis and functional assessment of D. Robles, World Record holder and Olympic Champion in 110m Hurdles. En: *Actas de la 29a Conferencia de la International Society of Biomechanics in Sports ISBS en Porto (Portugal)*, págs. 315-318.
- López, J.L., Padullés, J.M. y Tous, J. (1999). «Nuevas tendencias en fuerza y musculación». En: ed. por J. Tous. ERGO. Cap. Valoración del entrenamiento de la fuerza, págs. 182-204.
- López, J.L. y col. (1999). Anàlisi de la reproductibilitat en tres tests de salt amb plataforma de forces i contactes. En: *Revista Apunts* 58, págs. 62-66.
- MacDougall, J.D., Wenger, H.A. y Green, H.J (1995). *Evaluación fisiológica del deportista*. Barcelona: Paidotribo.
- Mackall, M. (n.d.). Mercurial [Software]. Recuperado de <http://mercurial.selenic.com/>.
- Madrigal, N. y col. (2010). Exercise Science Academic Programs and Research in the Philippines. En: *International Journal of Exercise Science* 3.4, pág. 1.

- Margaria, R., Aghemo, P. y Rovelli, E. (1966). Measurement of muscular power (anaerobic) in man. En: *Journal of Applied Physiology* 21, págs. 1662-1664.
- Marina, M. (2003). «Valoración, entrenamiento y evolución de la capacidad de salto en gimnasia artística de competición». Tesis doct. INEFC, Universitat de Barcelona.
- Martín, R. y col. (2001). Fiabilidad de las pruebas de fuerza en salto vertical y velocidad de carrera en escolares de 6 a 8 años. En: *Revista Apunts* 63, págs. 40-45.
- Martin, C. (2010). *Debunking the 1 % Myth*. Recuperado de <http://broadcast.oreilly.com/2010/09/debunking-the-1-myth.html>.
- Martínez Ramírez, L.M. y Marín Villada, A.F. (2008). «Perfiles de potencia en miembros inferiores en la modalidad de Drop Jump en los deportistas preseleccionados por Risaralda a Juegos Deportivos Nacionales 2008». Tesis de lic. Universidad Tecnológica de Pereira.
- Mas, J. (2005). *Software Libre. Técnicamente viable, económicamente sostenible, y socialmente justo*. Barcelona: Infonomía.
- Mas, J. y col. (2007). *Ingeniería del software en entornos de SL*. 2.ª ed. UOC.
- Massachusetts Institute of Technology (2007). *MIT CSAIL Open Source Hardware Designs*. Recuperado de <http://csg.csail.mit.edu/oshd/index.html>.
- McConnell, S. (1999). Open-source methodology: Ready for prime time. En: *IEEE software* 16.4, págs. 6-8.
- (2004). *Code complete*. Vol. 2. Microsoft Press.
- McGinnis, P.M. (2005). *Biomechanics of Sport and Exercise*. 2.ª ed. USA: Human Kinetics.
- McMillian, D.J. y col. (2006). Dynamic vs. static-stretching warm up: the effect on power and agility performance. En: *Journal of Strength and Conditioning Research* 20.3, pág. 492. DOI: 10.1519/18205.1.
- Miller, D. y Slager, D. (2006-2007). BCrypt.cs [Software]. Recuperado de <http://derekslager.com/blog/attachment.ashx/bcrypt-dotnet-strong-password-hashing-for-dotnet-and-mono/BCrypt.cs>.
- Mislove, A. y col. (2008). «Growth of the flickr social network». En: *WOSP '08: Proceedings of the first workshop on Online social networks*. New York, NY, USA: ACM, págs. 25-30.
- Molist, Mercè (2005). Entrevista Jordi Murgó. En: *Revista @rroba* 95. Recuperado de <http://ww2.grn.es/merce/2005/savage.html>.



- Morales, A. y De Blas, X. (2010). Chronopic-Firmwarecord [Software]. Recuperado de <http://git.gnome.org/browse/chronojump/tree/chronopic-firmware/chronopic-firmwarecord>.
- Mozilla Foundation (n.d.). Bugzilla [Software]. Recuperado de <http://www.bugzilla.org/>.
- National Aeronautics and Space Administration (2009). *NASA Open Source Software Agreement*. Recuperado de <http://ti.arc.nasa.gov/opensource/nosa/>.
- Netcraft (2011). *Most Reliable Hosting Company Sites in December 2011*. Recuperado de <http://news.netcraft.com/archives/2012/01/01/most-reliable-hosting-company-sites-in-december-2011.html>.
- Nikhil (2008). *Introduction to video analytics*. Recuperado de <http://7evin.blogspot.com/2008/10/introduction-to-video-analytics-video.html>.
- Novák, Ondřej, Ondřej, František y Ondřej, Jan (2006). *Gesture recognition*. Recuperado de [http://ibm-cvut.felk.cvut.cz/srp2/gesture\\_recognition/doc/doc.pdf](http://ibm-cvut.felk.cvut.cz/srp2/gesture_recognition/doc/doc.pdf).
- Okoli, C. (2009). «A Brief Review of Studies of Wikipedia in Peer-Reviewed Journals». En: *2009 Third International Conference on Digital Society*. IEEE, págs. 155-160.
- Open Source Ecology (n.d.). *Global Village Construction Set*. Recuperado de <http://opensourceecology.org/gvcs.php>.
- Open Source Initiative (n.d.). *The Open Source Definition*. Recuperado de <http://opensource.org/docs/osd>.
- OpenCourseWare (n.d.). *OpenCourseWare Websites*. Recuperado de <http://www.ocwconsortium.org/en/courses/ocwsites>.
- Openformats.org (2011). *Why use open formats?* Recuperado de <http://www.openformats.org/enShowAll>.
- Padullés, J.M. (2011). «Valoración de los parámetros mecánicos de la carrera. Desarrollo de un nuevo instrumento de medición». Tesis doct. Barcelona: INEFC, Universitat de Barcelona.
- Pagaduan, J.C. (nov. de 2011). «Test-Retest Reliability of Countermovement Jump on Free Technology». Presentación oral en la 5th Asia-Pacific Conference on Exercise and Sports Science en Shangai.
- (feb. de 2012). «Reliability of Four Series 15-Second Vertical Jump Test Using an Open Source Technology». Presentación oral en la 3a Conferencia Internacional del Deporte y Ciencias del Ejercicio en Tailandia.

- Parker, K.A. (2000). *Open Sources And The Open Society*. Recuperado de <http://agora.ph.i.gvsu.edu/kap/OSFS/OSFSEssay.pdf>.
- Pérez, E.M. (2007). *Microcontroladores PIC: sistema integrado para el autoaprendizaje*. Marcombo.
- Picerno, P., Camomilla, V. y Capranica, L. (2011). Countermovement jump performance assessment using a wearable 3D inertial measurement unit. En: *Journal of Sports Sciences* 29.2, págs. 139-146. DOI: 10.1080/02640414.2010.523089.
- Pinheiro, Jose y col. (2010). nlme: Linear and Nonlinear Mixed Effects Models [Software]. R package version 3.1-97.
- Porikli, F.M. (2003). Human Body Tracking by Adaptive Background Models and Mean-Shift Analysis. En: *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, págs. 1-9. Recuperado de <http://www.merl.com/reports/docs/TR2003-36.pdf>.
- Portolés, J. (1994). Entrenamiento de la fuerza en el fútbol. En: *Actualizaciones en Fisiología del Ejercicio* 2.1, págs. 24-38.
- Proctor, F. (dic. de 2002). *Introduction to Linux for Real-Time Control*. Inf. téc. National Institute of Standards y Technology, Intelligent Systems Division,
- Provos, N. y Mazieres, D. (1999). «A future-adaptable password scheme». En: *Proceedings of the Annual USENIX Technical Conference*. Citeseer.
- Public Library of Science (n.d.). *Open-Access License*. Recuperado de <http://www.plosone.org/static/license.action>.
- R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. Recuperado de <http://www.R-project.org>.
- Racionero, Luis (1983). *Del paro al ocio*. Barcelona: Anagrama.
- Raymond, E. (ed) (2000). *The Jargon File*. Recuperado de <http://www.tuxedo.org/~esr/jargon>.
- Raymond, E. (1999). The cathedral and the bazaar. En: *Knowledge, Technology & Policy* 12.3, págs. 23-49. DOI: 10.1007/s12130-999-1026-0.
- Raymond, E.S. (2004). The halloween documents. En: *Open Source Initiative*, págs. 10-23.
- Robles, G. (2006). «Software Engineering Research on Libre Software: Data Sources, Methodologies and Results». Tesis doct. Madrid: Universidad Rey Juan Carlos. Escuela superior de ciencias

experimentales y tecnología. Recuperado de <http://gsyc.escet.urjc.es/~grex/thesis-grex.pdf>.

Romero, D. y Tous, J. (2010). *Prevención de Lesiones en el Deporte. Claves para un rendimiento deportivo óptimo*. Ed. Médica Panamericana.

Ruddock, A. (2010). Affordable technology: jump to it! En: *Peak Performance* 295, págs. 10-11. Recuperado de <http://www.broadbeachcats.com.au/cmsAdmin/uploads/295.pdf>.

Sale, D. G. (1995). «Evaluación fisiológica del deportista». En: ed. por J. D. MacDougall. Barcelona: Paidotribo. Cap. Evaluación de la fuerza y de la potencia, págs. 37-132.

Sargent, D.A. (1921). The physical test of a man. En: *Am Phys Educ Rev* 26, págs. 188-194.

Schindler, E. (2007). Enterprise Developers Programming Speed? Check. Time to Fix Bugs? Not So Fast. En: *CIO* 26. Recuperado de [http://advice.cio.com/esther\\_schindler/enterprise\\_developers\\_programming\\_speed\\_check\\_time\\_to\\_fix\\_bugs\\_not\\_so\\_much?](http://advice.cio.com/esther_schindler/enterprise_developers_programming_speed_check_time_to_fix_bugs_not_so_much?)

Schmidtbleicher, D. (1992). Strength and Power in Sport. En: *The Encyclopaedia of Sports Medicine*. Ed. por P.V. Komi. Vol. 3. Oxford: Blackwell. Cap. Training for power events, págs. 169-179.

Solomon, A. (1990). *A Brief History of PC Viruses*. Recuperado de <http://www.bocklabs.wisc.edu/~janda/solomhis.html>.

Spinellis, Diomidis (2003). Reflections on trusting trust revisited. En: *Commun. ACM* 46.6, pág. 112. DOI: 10.1145/777313.777347.

Stallman, R. (1999). *Free Hardware*. Recuperado de [http://features.linuxtoday.com/news\\_story.php3?ltsn=1999-06-22-005-05-NW-LF](http://features.linuxtoday.com/news_story.php3?ltsn=1999-06-22-005-05-NW-LF).

— (2002). What is copyleft. En: *Free software foundation*.

— (2003). Some confusing or loaded words and phrases that are worth avoiding. En: *Free Software Free Society: Selected Essays of Richard M. Stallman*. Recuperado de <http://www.gnu.org/philosophy/words-to-avoid.html>.

— (2007). *Why Open Source misses the point of Free Software*. Recuperado de <http://www.gnu.org/philosophy/open-source-misses-the-point.html>.

Stallman, R. y col. (2002). The free software definition. En: *Free Software Free Society: Selected Essays of Richard M. Stallman*, págs. 41-44.

Stein, W. A. y col. (2009). *Sage Mathematics Software (Version 3.3)*. <http://www.sagemath.org>. The Sage Development Team.

Storey, D. (2000). *Informática. Tecnología de la información eficaz: Un manual para pensar y trabajar con inteligencia*. Barcelona: Blume.

Tan, I. y col. (2011). Effect of acute blood occlusion training on heart rate variability occlusion and postactivation potentiation. En: *Journal of Sports Science and Health* 13 (supp 1). Presentado en 3rd International Conference on Sport and Exercise Science, February 7-10, Imperial Queen's Park Hotel, Bangkok, Thailand, pág. 56.

The GNOME Project (n.d.). GNOME [Software]. Recuperado de <http://www.gnome.org>.

The Linux Information Project (2007). *Free File Format Definition*. Recuperado de [http://www.linfo.org/free\\_file\\_format.html](http://www.linfo.org/free_file_format.html).

The Open University (n.d.). *Copyright of the OpenLearn website*. Recuperado de <http://www8.open.ac.uk/about/main/admin-and-governance/policies-and-statements/copyright-the-openlearn-website>.

Thompson, C. (2008). Can You Count on Voting Machines? En: *New York Times Magazine* 6. Recuperado de <http://www.verifiedvotingfoundation.org/article.php?id=6542>.

Thompson, C.J. y Bemben, M.G. (1999). Reliability and comparability of the accelerometer as a measure of muscular power. En: *Medicine & Science in Sports & Exercise* 31.6, pág. 897. DOI: 10.1097/00005768-199906000-00020.

Thompson, K. (1984). Reflections on trusting trust. En: *Commun. ACM* 27.8, págs. 761-763. DOI: 10.1145/358198.358210.

TIOBE Software BV (2012). *TIOBE Programming Community Index for February 2012*. Recuperado de <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.

Tihanyi, J., Apor, P. y Fekete, G. (1982). Force-velocity-power characteristics and fiber composition in human knee extensor muscles. En: *Eur J Appl Physiol Occup Physiol* 48.3, págs. 331-43. DOI: 10.1007/BF00430223.

Top500.org (2011). *TOP500 Statistics. November 2011, Operating system Family*. Recuperado de <http://i.top500.org/stats>.

Topend Sports Network (n.d.). *Chronoj*. Recuperado de <http://www.topendsports.com/testing/chronojump>.

Torres-Ronda, L., Sánchez-Medina, L. y González-Badillo, J.J. (2011). Relationship between golf performance, anthropometry and muscle strength in young elite golfers. En: *Actas del 16th Congreso del European College of Sports Science, Liverpool, 2011*.

- Torres, L. (2011). «Perfil físico-técnico de jugadores de golf amateurs de élite y su relación con el rendimiento». Tesis doct.
- Tous, J. y Moras, G. (1999). Control del entrenamiento de la fuerza mediante el número de repeticiones realizado por bloques de tiempo. En: *Libro de Actas del IV Congreso de Ciencias de la Actividad Física y el Deporte*, págs. 449-459.
- Tous, J. y col. (2000). «W5" test: assesing average velocity and power with affordable equipment». En: *Proceedings of the 5th Annual Congress of the European College of Sport Science*. Ed. por J. Avela, Komi P.V. y J. Komulainen. Jyväskylä, Finland, pág. 748.
- Trucco, E. y Verri, A. (1998). *Introductory techniques for 3-D computer vision*. Vol. 201. Prentice Hall New Jersey.
- Universitat Oberta de Catalunya (2008). *Sobre el UOC OpenCourseWare*. Recuperado de <http://ocw.uoc.edu/bfquienes-somos>.
- Vickery G., Wunsch-Vincent S. (2007). *Participative Web and user-created content : Web 2.0, wikis and social networking*. Organisation for Economic Co-operation y Development Publishing.
- Vincid (n.d.). Keimove [Equipo y software]. Recuperado de <http://www.vincid.com>.
- Von Hippel, E. y Lakhani, K.R. (2000). «How open source software works: "free" user-to-user assistance». En: *Research Policy*.
- W3C (2004). *Web Services Architecture*. Recuperado de <http://www.w3.org/TR/ws-arch/>.
- W3Schools (2011). *Browser Display Statistics*. Recuperado de [http://www.w3schools.com/browsers/browsers\\_display.asp](http://www.w3schools.com/browsers/browsers_display.asp).
- Warshofsky, F. (1994). *The patent wars: The battle to own the world's technology*. Wiley New York.
- Wayner, P. (2001). *La ofensiva del software libre. Cómo Linux y el movimiento del software libre se impusieron frente a los titanes de la alta tecnología*. Barcelona: Ediciones Granica.
- Wells, J. (1996). *Virus Timeline*. Recuperado de <http://www.bockabs.wisc.edu/>.
- Wheeler, D.A. (2007). *Why open source software/free software (OSS/FS, FLOSS, or FOSS)? Look at the numbers*. Recuperado de [http://www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html).
- (2011). *Open Source Software (OSS or FLOSS), the U.S. Department of Defense (DoD), and NASA*. Recuperado de <http://www.slideshare.net/ckleclerc/2011-nasa-open-source-summit-david-wheeler>.

- Wikipedia (2010). Chronojump Boscosystem — Wikipedia, La enciclopedia libre. [Internet; descargado 13-enero-2012]. Recuperado de [http://es.wikipedia.org/w/index.php?title=Chronojump\\_Boscosystem&oldid=38140476](http://es.wikipedia.org/w/index.php?title=Chronojump_Boscosystem&oldid=38140476).
- (2011a). Comparison of spreadsheet software — Wikipedia, The Free Encyclopedia. [Online; accessed 13-January-2012]. Recuperado de [http://en.wikipedia.org/w/index.php?title=Comparison\\_of\\_spreadsheet\\_software&oldid=467123099](http://en.wikipedia.org/w/index.php?title=Comparison_of_spreadsheet_software&oldid=467123099).
- (2011b). Miguel de Icaza — Wikipedia, La enciclopedia libre. [Internet; descargado 13-enero-2012]. Recuperado de [http://es.wikipedia.org/w/index.php?title=Miguel\\_de\\_Icaza&oldid=52005762](http://es.wikipedia.org/w/index.php?title=Miguel_de_Icaza&oldid=52005762).
- (2012). Reliability of Wikipedia — Wikipedia, The Free Encyclopedia. [Online; accessed 13-January-2012]. Recuperado de [http://en.wikipedia.org/w/index.php?title=Reliability\\_of\\_Wikipedia&oldid=470322949](http://en.wikipedia.org/w/index.php?title=Reliability_of_Wikipedia&oldid=470322949).
- Wilson, J. M. y Flanagan, E. P. (2008). The Role of Elastic Energy in Activities with High Force and Power Requirements: A Brief Review. En: *Journal of Strength and Conditioning Research* 22.5, págs. 1705-1715. DOI: 10.1519/JSC.0b013e31817ae4a7.
- Winters, J. (2011). Steam Punks. En: *Mechanical engineering* 133.6, págs. 32-37. Recuperado de [http://memagazine.asme.org/Articles/2011/June/Steam\\_Punks.cfm](http://memagazine.asme.org/Articles/2011/June/Steam_Punks.cfm).
- Wolfram Mathematica (n.d.). Mathematica tutorial [Software manual]. Recuperado de <http://reference.wolfram.com/mathematica/tutorial/WhyYouDoNotUsuallyNeedToKnowAboutInternals.html>.
- Woolstenhulme, M.T. y col. (2006). Ballistic stretching increases flexibility and acute vertical jump height when combined with basketball activity. En: *Journal of strength and conditioning research* 20.4, pág. 799. DOI: 10.1519/00124278-200611000-00012.
- Wright, C.W. y Walsh, E.J. (1999). Hunting hurricanes. En: *Linux Journal* 58es, págs. 20-30.
- Yamaguchi, T. e Ishii, K. (2005). Effects of static stretching for 30 seconds and dynamic stretching on leg extension power. En: *Journal of Strength and Conditioning Research* 19.3, págs. 677-683. DOI: 10.1519/00124278-200508000-00032.
- Yeadon, M.R. y Challis, J.H. (1994). The future of performance-related sports biomechanics research. En: *Journal of Sports Sciences* 12, págs. 3-32. DOI: 10.1519/00124278-200508000-00032.
- Yodaiken, V. y col. (1999). «The rtlinux manifesto». En: *Proc. of the 5th Linux Expo*. Citeseer.

Zatsiorski, V.M. (1989). Valoración de la velocidad. En: *Metrología deportiva*. La Habana: Editorial Planeta, págs. 130-160.





Parte VII

## Índice alfabético



## Índice alfabético

- Abstracción, 52–54
- Acelerómetro, 16, 19, 27, 36, 175
- Algoritmo, 9, 36, 47–49, 52, 125, 166
- Android, 45, 173
- Apache, 161
- Autoconf, 55
- Autoría, 47, 49, 65
- Autotools, 99, 100
  
- Barreras de infrarrojos, 18, 175
- Bazaar, 62
- Biblioteca, 41, 51, 54, 97, 99, 128, 129, 152
- Biomecánica, 24, 25, 39, 119, 136
  - deportiva, 10, 24, 38, 40, 67
  - médica, 24
  - ocupacional, véase Ergonomía
- Bootloader, 72, 74
- BSD, véase Software, licencia del, Licencia tipo
  - Berkeley Software Distribution
- Buffer, 72
- Bugzilla, 62, 100, 173
  
- Caja negra, 9
- Cámara de vídeo, 38
  - de alta velocidad, 39, 40, 138
- CDJ, véase Tests, de salto, en caída con contramovimiento
- CEA, véase Ciclo de Estiramiento-Acortamiento
- Centro de gravedad, 34, 35
- Chronojump, 74, 85, 95–124, 128, 138, 160, 161, 163, 166, 170, 172, 173, 176
  - captura de pantalla de, 103, 105–107, 165
  - coste de desarrollo de, 100
  - diagrama de clases de, 217–222
  - diagrama de la base de datos de, 104, 105, 216
- GIT de, 77, 111, 112
  - manual de, 112
- Chronojump-Boscosystem, 170, 173
- Chronojump-Server, 114, 161–166, 223–227
  - gráficos, 163, 164, 223–227
  - servicios web de
    - Query, 165
    - UploadPing, 114, 165
- Chronojump-server, 112
- Chronopic-Firmwarecord, 74, 77
- Ciclo de Estiramiento-Acortamiento, 29
- Cineantropometría, 10
- Cinegrametría, 39
- Cinemática, 10, 24, 26, 27, 38, 40
- Cinesiología, 24
- Cinética, 24, 26
- Clase, 52–54, 102, 108, 160, 161, 209–222
- CMJ, véase Tests, de salto, en contramovimiento
- COCOMO, véase Software, gestión del proyecto de, coste
- COCOMO II, véase Software, gestión del proyecto de, coste
- Código fuente, 36, 42–50, 53–62, 77, 100, 102, 108, 112, 114, 131, 152, 161
- Compilador, 47
- Copia privada, 47
- Copyleft, 49–50, 66
- Copyright, 47
- CPU, véase Unidad Central de Proceso
- Cracker, 43
- Creative Commons, 65–67
- CSV, véase Valores separados por comas
- Cultura, 43, 67
- CVS, 100
  
- Datos, 9, 32, 36, 38, 39, 50, 53, 54, 72, 95, 97, 99, 108, 127, 136, 141, 146

- base de datos, 96, 97, 99, 102, 105
- compartir datos, 11, 21, 153–166, 170
- fiabilidad de los, 67, 71, 155–159, 172, 175
- Debian (proyecto), 42, 47
- Derechos de autor, véase Copyright
- Dinámica, 24
- DJ, véase Tests, de salto, en caída
- DLT, véase Transformación lineal directa
- EAGLE, 64, 72
- Empotrado, véase Microcontrolador
- Encapsulación, 52–54, 108
- Ensamblador, 55, 72
- Ergojump Boscosystem, 18, 33, 34, 79, 80, 84, 86, 138
- Ergonomía, 24
- Error de medida, 34, 35, 73, 77–79, 87, 89–91, 134, 146–151, 170, 173
  - accidental, 32
  - sistemático, 32, 85
- Estática, 24
- Evaluación, véase Valoración
- Filtro, 36, 66, 80
- Firmware, 36, 62, 72, 74, 79, 176, 209–215
- Formato, 50–52, 64, 95, 108, 127
- FOSS, véase Software, licencia del, FLOSS
- Fotogrametría, 38, 151
- FreePower, 19, 42
- FreeSense, 42
- Freeware, 51
- FTP, 111, 113
- Fuerza, 16, 24, 26–31, 34, 35, 67, 80, 82, 87, 108, 121, 122
- Fuerza bruta, 166
- Galga extensiométrica, 82, 83, 88, 175
- Gantt, diagrama de, 56
- GDK, 108
- Generador de ondas, 79
- Hameg HM 8030-4, 72, 73
- Hameg HM 8035, 72, 73
- Gettext, 99, 111
- GIT, 62, 74, 100, 108, 111, 112
- Glade-3, 99
- GNOME, 45, 47, 54, 100, 111, 113
  - translators, 111, 112, 176
- GNU, véase GNU's Not Unix
- GNU GPL, véase Software, licencia del, Licencia Pública General de GNU
- GNU's Not Unix, 44, 47, 64, 72
- GNU/Linux, véase Linux
- Gold Standard, 74, 84, 85, 88, 89, 93
- Google Apps, 113
- Gpsim, 72
- Gputils, 72
- Grep, 99
- Gstreamer, 99
- GTK#, 99
- Hacker, 43–44
- Hardware, 21, 36–38, 42, 67, 71–79, 111, 166, 172
  - archivo GERBER, 64, 74, 112
  - archivo PCB, 64, 211, 212
  - ciclo de vida, 64
  - coste del, 64
  - desarrollo del, 62–64, 71–74
  - errores del, 64, 77–79
  - estático, 62–64, 71
  - licencia del, 64–65
    - Hardware abierto, 9, 64, 65, 71, 72, 113, 118, 119
    - LLL, 64, 72
    - reconfigurable, 62–64
- Hashing, 161, 166
- Herencia, 52–54, 209–222
- Idea, 49, 66
- IE, véase Índice, de Elasticidad

- Indicación geográfica, 47
- Índice
  - de Bosco, 97
  - de Elasticidad, 29, 30, 98, 108, 225
  - de Reactividad, 29, 99
  - de Resistencia a la Fuerza Rápida, 30
  - de Utilización de Brazos, 30, 98, 108, 225
  - DJ, 97, 108
  - Q, 97, 108
- indumentaria, 139, 172, 173
- Innosetup, 99
- Innovación, 45, 46, 48, 66
- Interrupciones, 79, 215
- Invento, 48–49
- IUB, véase Índice, de Utilización de Brazos
- KDE, 45
- Kernel, 44, 56
- KiCad, 72, 74
- Kinesiología, véase Cinesiología
- Kinovea, 20, 21
- KneeAngle, 114, 127–135, 152, 170, 173
- Latencia, 37, 38, 71
- L<sup>A</sup>T<sub>E</sub>X, 112
- Lenguaje de programación, 47, 52, 54–55
  - C, 54, 55
  - C++, 55, 129–131, 152
  - C#, 54, 55, 96, 99, 100, 102
  - Java, 55
  - Perl, 55
  - Python, 55, 72
  - Shell, 55
  - Visual Basic, 55
- Linux, 38, 41, 44, 45, 47, 56, 72, 77, 96, 98, 99, 102, 113, 131
  - distribución de
    - Debian, 102
    - Fedora, 102
    - Ubuntu, 102
  - Real Time Linux, 38
- LJ, véase Tests, de salto, squat con carga
- MacOSX, 41, 99, 102, 131, 173
- Marca registrada, 44, 47, 52, 66, 170
- Marcador, 39, 120, 125–152, 172, 173
- Mecánica, 9, 24
- Mercurial, 62
- Microcontrolador, 34, 36–37, 71–81, 88, 95, 97, 169, 175
  - Arduino, 65
  - Chronopic, 71, 77, 96, 98, 100, 108, 112, 121, 175
  - Chronopic 1, 74, 75
  - Chronopic 2, 74, 75
    - planos de, 209
  - Chronopic 3, 73, 74, 82, 84, 85, 114, 117, 138, 175
    - firmware de, 214, 215
    - fotografía de, 76
    - materiales de, 228
    - planos de, 210–213
    - validación de, 78, 79, 113, 119
- S2PROTO, 63
- Skypic, 63, 65, 74
- Modelo de predicción, 120, 125, 134, 136–152, 172, 173, 235
- Mono, 96–100, 102
- Multiplataforma, véase Plataforma, referido a ordenador y sistema operativo
- Multitarea, 37, 71, 72
  - concurrente, 37
  - preferente, 37
  - real, 37
- MuscleLab, 18, 82, 83
- MyBB, 113
- Objeto (programación), 52–54
- Ohloh, 100
- Open Source Computer Vision, 41, 99, 129, 130

- OpenCV, véase Open Source Computer Vision
- OpenOffice.org, 60, 112
- Ordenador personal, 37, 45, 71, 82, 83
- Osciloscopio Agilent 54621A, 72–74
- Patente, 44, 48–49, 52
- PIC, véase Microcontrolador
- Ping, 114–116, 165
- Piratería, 48
- Plataforma
- de contactos, 9, 11, 16, 18, 21, 33–35, 41, 69, 71, 74, 79–94, 118, 119, 138, 169, 173
  - de Bosco, 84, 85, 92, 93
  - de fuerzas, 16, 18, 34–36, 175
  - de infrarrojos, 16, 18, 34, 35, 175
  - referido a ordenador y sistema operativo, 41, 64, 97, 99, 100, 119, 129
- Polimorfismo, 52–54
- Potencia, 26–28, 30, 31, 36, 67, 98, 108, 120, 121
- Power Tester, 117
- Propiedad intelectual, 47–49
- Pulso espúreo, 37, 72, 77
- QDJ, véase Tests, de salto, en caída rápido
- R, 85, 99, 108, 109, 119, 120, 130, 146, 152, 161
- SQLite, 161
- Resolución (píxeles), 38, 39
- Revisión de pares, 46, 57
- RJ, véase Tests, de salto, reactivo
- Script, 99, 161, 163, 173
- Secreto industrial, 47
- Servicios web, 66, 100, 160–162, 170
- Servidor, 9, 45, 46, 111, 114–116, 119, 157, 160–166, 170, 172, 219, 223–227
- Shareware, 51
- Sistema operativo, 37, 44, 45, 47, 64, 71, 114, 116, 165
- en tiempo real, 38, 71
- SJ, véase Tests, de salto, squat
- SLJ, véase Tests, de salto, horizontal a pies juntos
- Softcatalà, 112
- Software, 15, 16, 20, 21, 36, 41–48
- comunicación en el, 57, 59
  - desarrollo del, 52, 95–100, 129–130, 152, 160–161
  - errores del, 36, 46, 56–57, 59–60, 62, 96, 100, 110, 172
  - garantía del, 21, 43, 50
  - gestión del proyecto de, 56–62
  - ciclo de vida, 61
  - coste, 21, 61, 100, 131
  - historias de usuario, 60–61, 96–99, 127, 172
  - iteración, 61, 96–99
- licencia del, 21, 42–50
- FLOSS, 43
  - Libre Software, 43
  - Licencia Pública General de GNU, 49–51, 96
  - Licencia tipo Berkeley Software Distribution, 49–51
  - Open Source, 42–43, 45, 51, 65
  - Software comercial, 21, 43, 51, 57
  - Software libre, 9, 42–50, 54, 57, 59, 60, 66
  - Software privativo, 42, 43, 47, 48, 50, 57, 60
- metodología de programación del
- modelo de Catedral y Bazar, 56, 57
  - Programación ágil, 57–61, 95
  - Programación extrema, 59, 61, 95
  - programación por parejas, 59
- modelo de negocio del, 43, 46, 59

- paradigma de programación del, 52–56
  - Programación declarativa, 52
  - Programación imperativa, 52, 53
  - Programación orientada a objetos, 52–56, 99, 100, 102
- pruebas del, 59–61, 100, 108
- uso del, 45, 114–116, 165
- Software libre, 95, 101, 119, 120, 127, 128, 169
- SQL, 55, 99, 161
  - Inyección de, 161
- Sqlite, 99
- Subversion, 100
- SVN, véase Subversion
- Tabla de percentiles, 153, 160, 166, 175
- TC, véase Tiempo, de contacto
- Test-retest, 33, 121
- Tests, 27
  - 1/2 squat Piasenta, 28
  - de Bosco, 28–30, 33, 97, 125, 169
  - de campo, 27
  - de laboratorio, 25, 27
  - de Margaria, 28
  - de Margaria-Kalamen, 28, 98, 106
  - de salto, 28
    - Abalakov, 16, 28, 30–33, 97, 99, 106, 158
    - en caída, 29, 31, 97–99, 106
    - en caída con contramovimiento, 31
    - en caída rápido, 31, 35
    - en contramovimiento, 11, 29–31, 33, 35, 97, 99, 106, 121, 125, 128, 136, 137, 158, 166, 170
    - horizontal a pies juntos, 28, 33
    - multisaltos, 28
    - reactivo, 29, 97, 98, 106, 108, 161, 170
    - rocket, 30, 99, 106, 158
    - Sargent, 15, 17, 28
    - squat, 28–31, 33, 84, 85, 97, 99, 106, 158, 165
    - squat con carga, 28, 29
  - fiabilidad de los, 9, 30, 32–34, 38, 45, 57, 77, 85, 121, 135
  - precisión de los, 9, 32, 33, 39
  - reproductibilidad de los, 33
  - validez de los, 33, 36, 85, 88, 89, 119, 139, 173
- w5", 27
- Tiempo
  - de contacto, 11, 28, 31, 32, 35, 72, 73, 77, 97, 108
  - de vuelo, 11, 16, 28, 32, 34–36, 73, 74, 77, 88, 97, 139, 140, 165, 166, 223, 224, 226, 227
- Tiempo real, 38, 71, 108, 128
- Trabajo (mecánica), 26
- Trac, 62
- Transformación lineal directa, 20
- TV, véase Tiempo, de vuelo
- Twitter, 66, 112, 113
- Unidad Central de Proceso, 36
- UNIX, 44
- Valoración, 9, 25, 26, 84, 121
  - cualitativa, 26, 151
  - cuantitativa, 26
  - neuromuscular, 26, 121
- Valores separados por comas, 108, 128
- Velocidad, 26, 27, 29, 38–40, 108, 119, 121
- Videogrametría, 39
- Vim, 95, 99
- VirtualBox OSE, 99
- Visión por computador, 39, 125–135, 152
  - contorno, 130–132
  - Haar cascades, 39
  - iluminación, 40, 128, 129, 135, 140, 146, 172
  - Reconocimiento de gestos, 39
  - Segmentación de objetos, 40, 130, 140

Threshold, 40–41, 127, 130, 131, 134, 135

Tracking, 20

Back tracking, 40, 130

Forward tracking, 40, 130

Wikipedia, 66, 67, 120, 156

Windows, 41, 72, 74, 102, 113, 131

Xen, 161, 162, 170

XP, véase Software, metodología, Programación  
extrema







Parte VIII

Anexos



# Anexos

## A. Diagramas de las herramientas creadas

### A.1. Planos de los microcontroladores

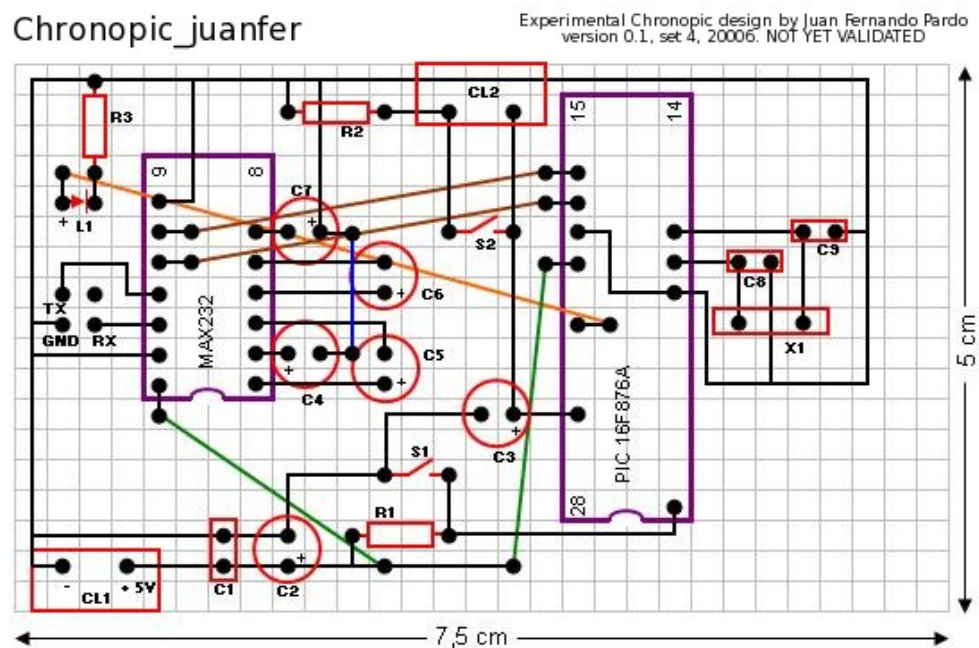


Figura A.1: Chronopic 2. Esquema eléctrico.

### A.2. Diagramas de flujo del Firmware de Chronopic

### A.3. Diagrama de la base de datos de Chronojump

### A.4. Diagrama de las clases de Chronojump y ChronojumpMini

Los diagramas que siguen a continuación no son realmente diagramas de clases pues se ha reducido considerablemente su funcionalidad. Los diagramas que se adjuntan incluyen únicamente el nombre de la clase, un símbolo de flecha indicando la herencia, y un sombreado en las cajas de las clases que implementan entorno gráfico.

Figura A.2: Chronopic 3. Esquemático.

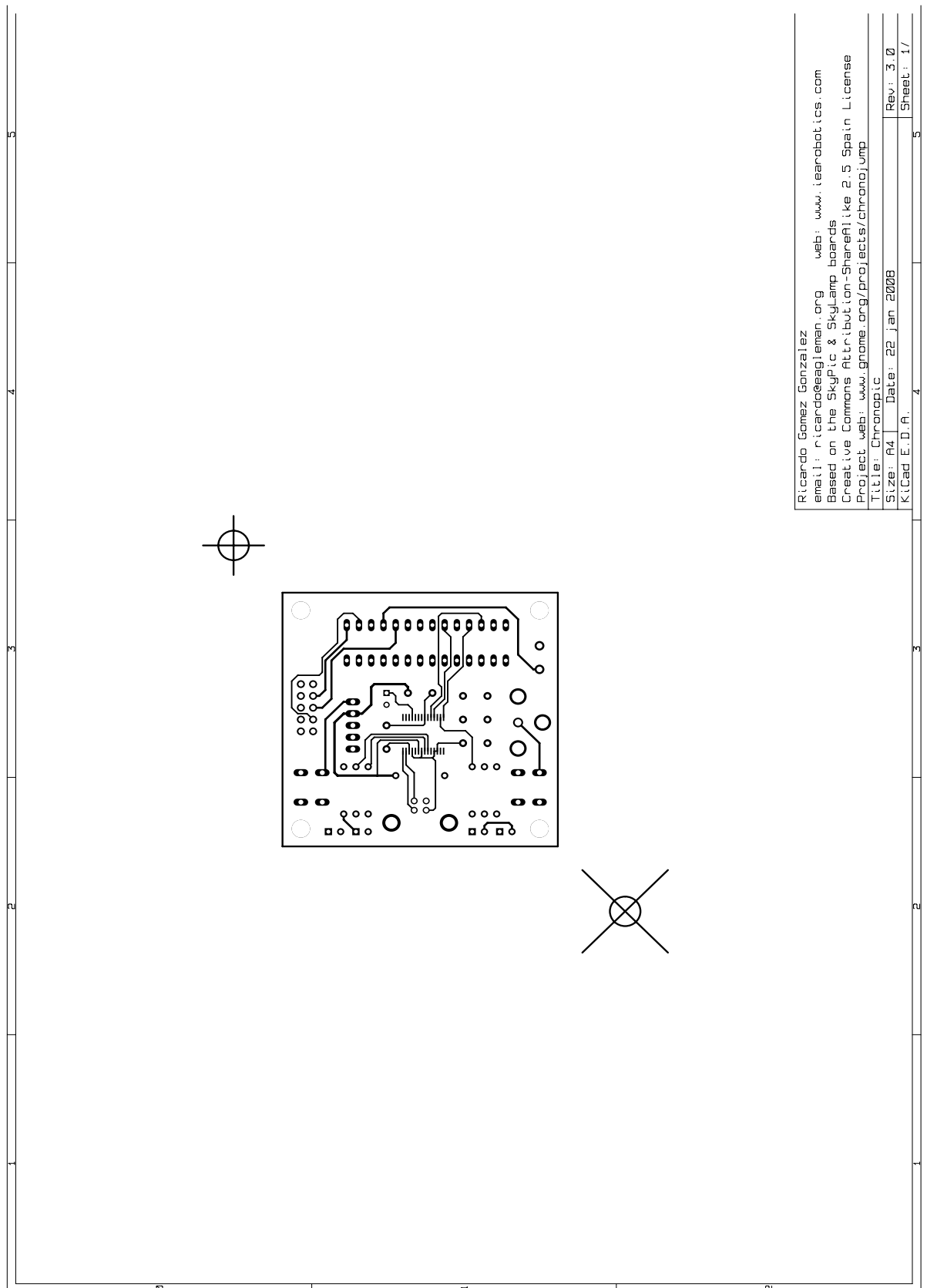


Figura A.3: Chronopic 3 PCB. Cara de los componentes.

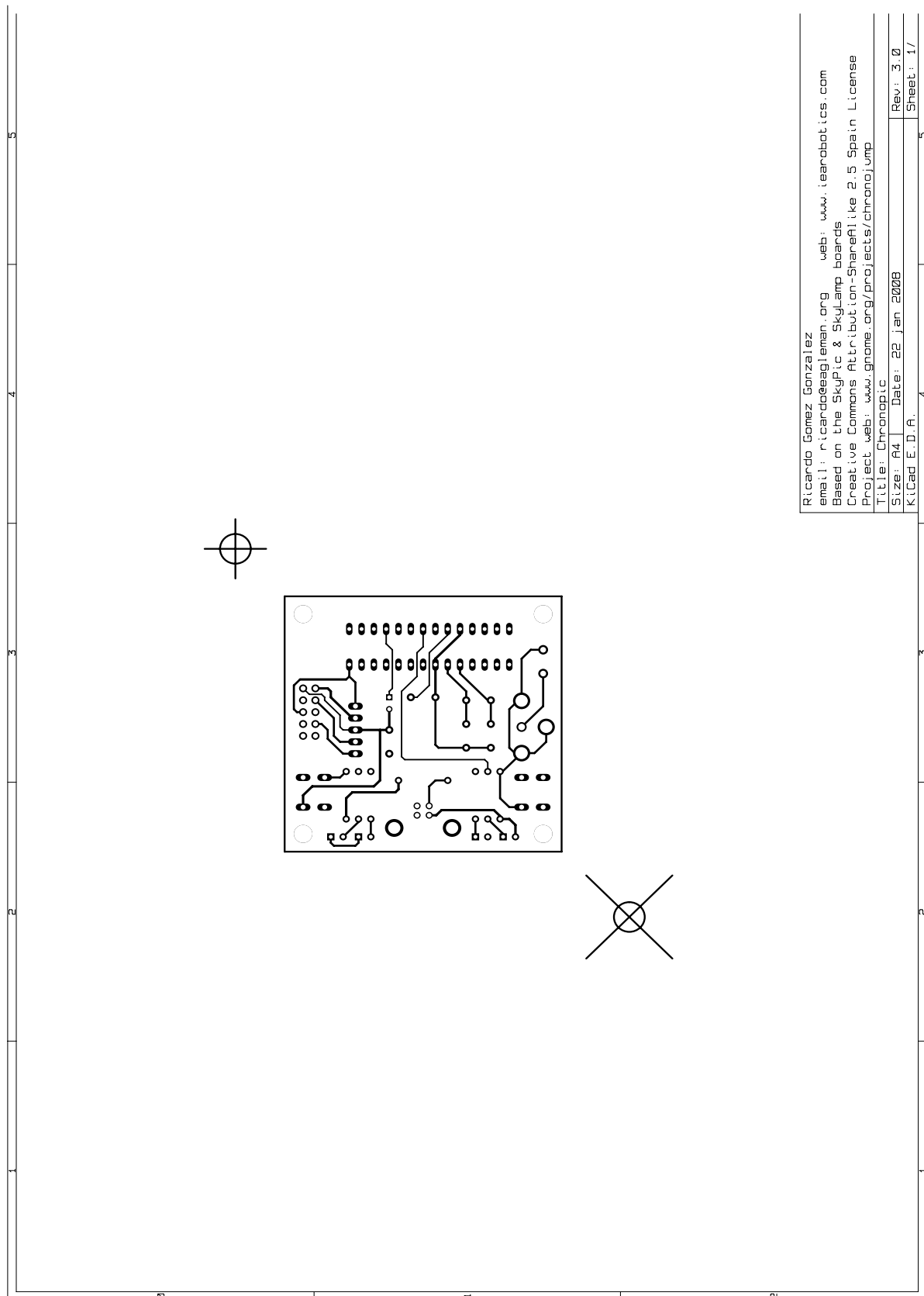


Figura A.4: Chronopic 3 PCB. Cara inferior.



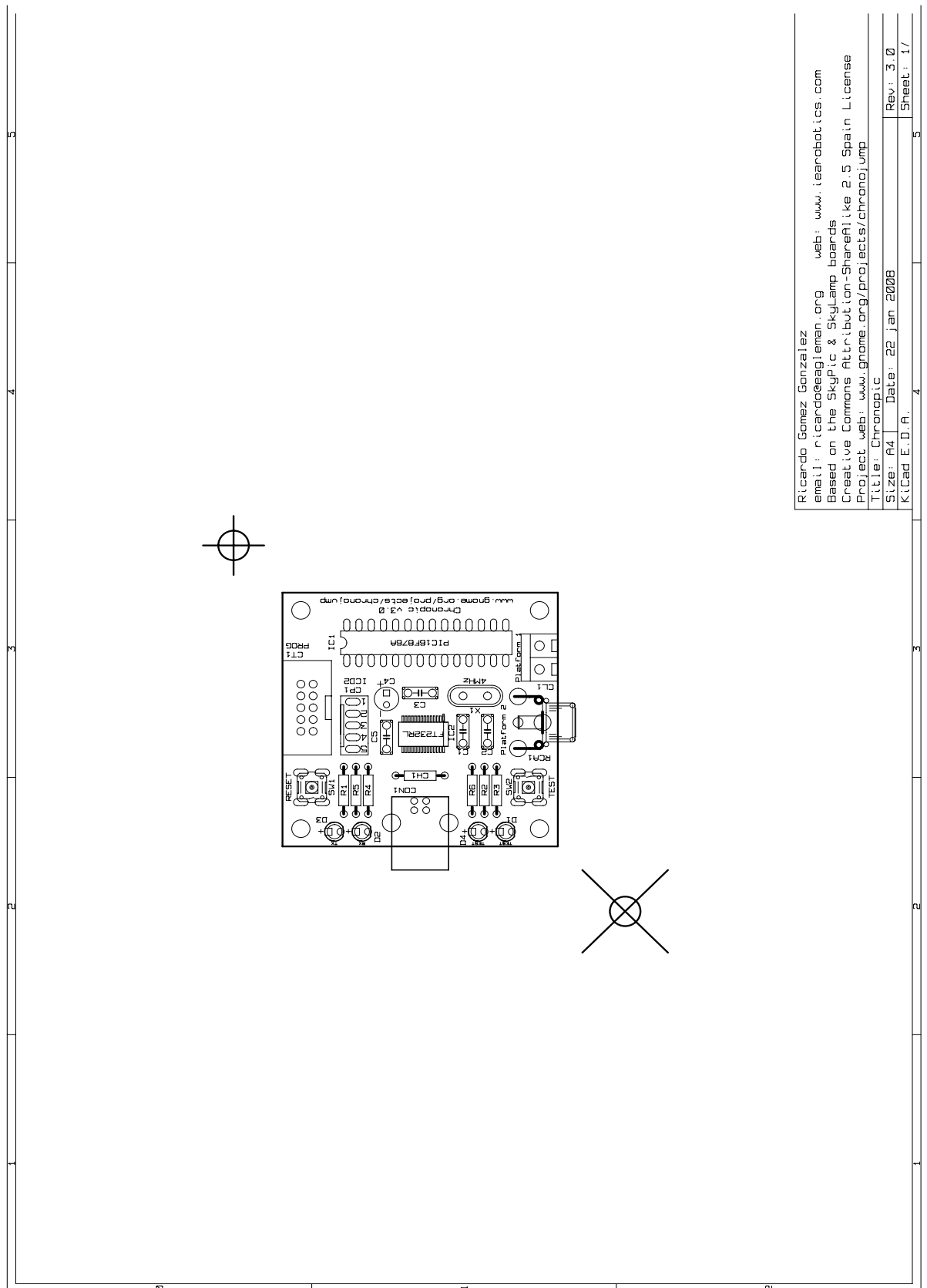


Figura A.5: Chronopic 3. Serigrafías.

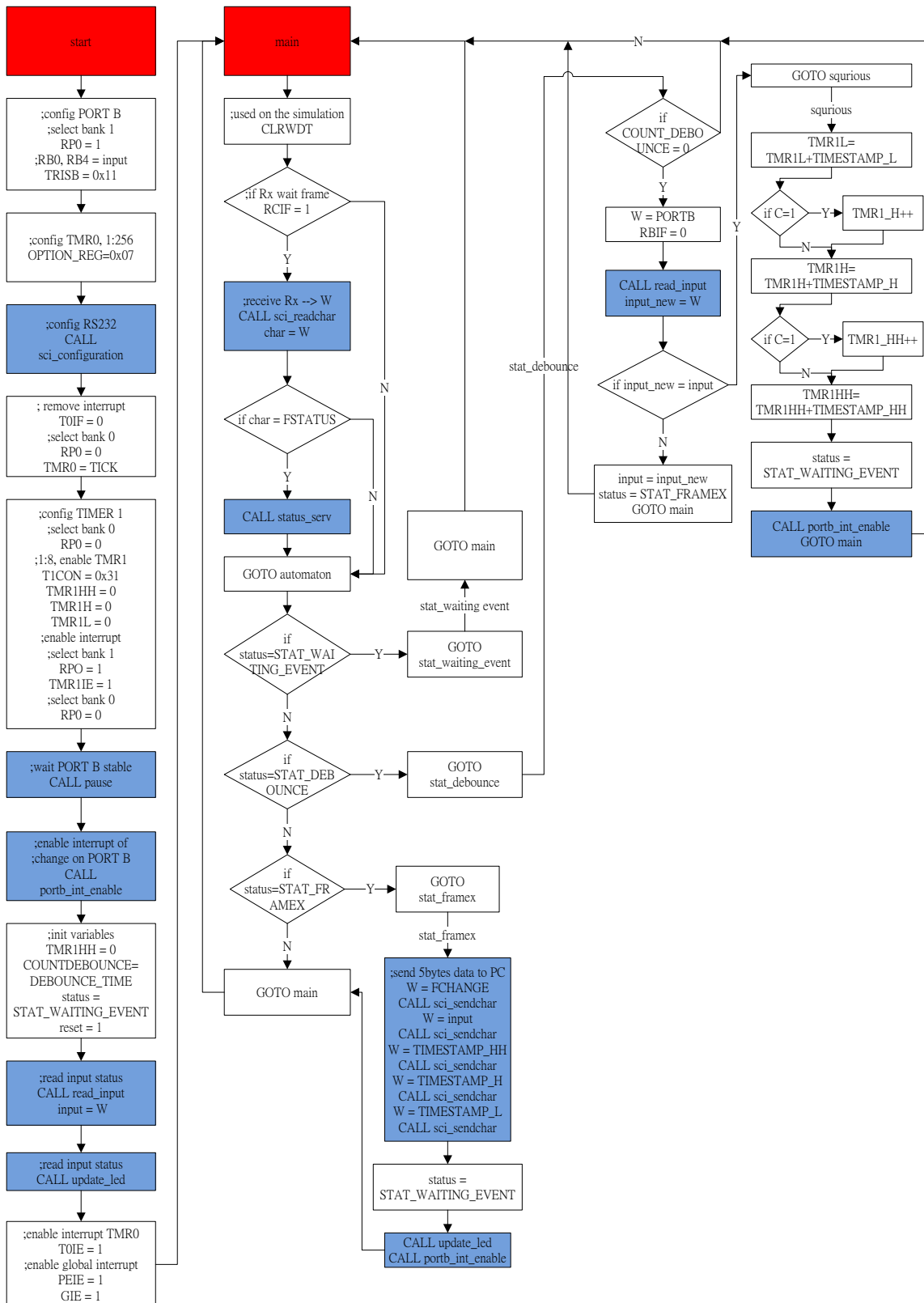


Figura A.6: Diagrama de flujo del firmware de Chronopic.

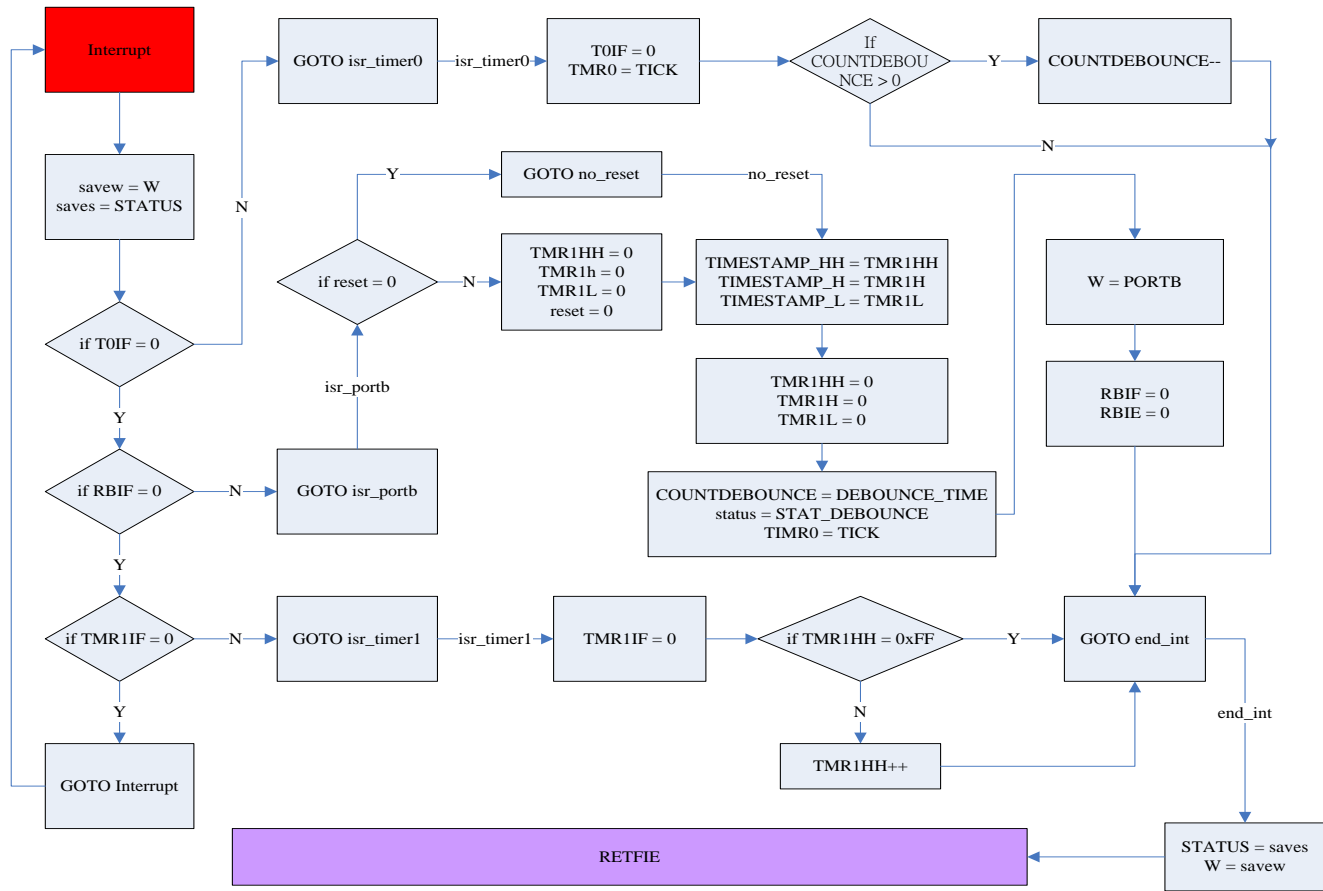


Figura A.7: Diagrama de flujo de las interrupciones del firmware de Chronopic.

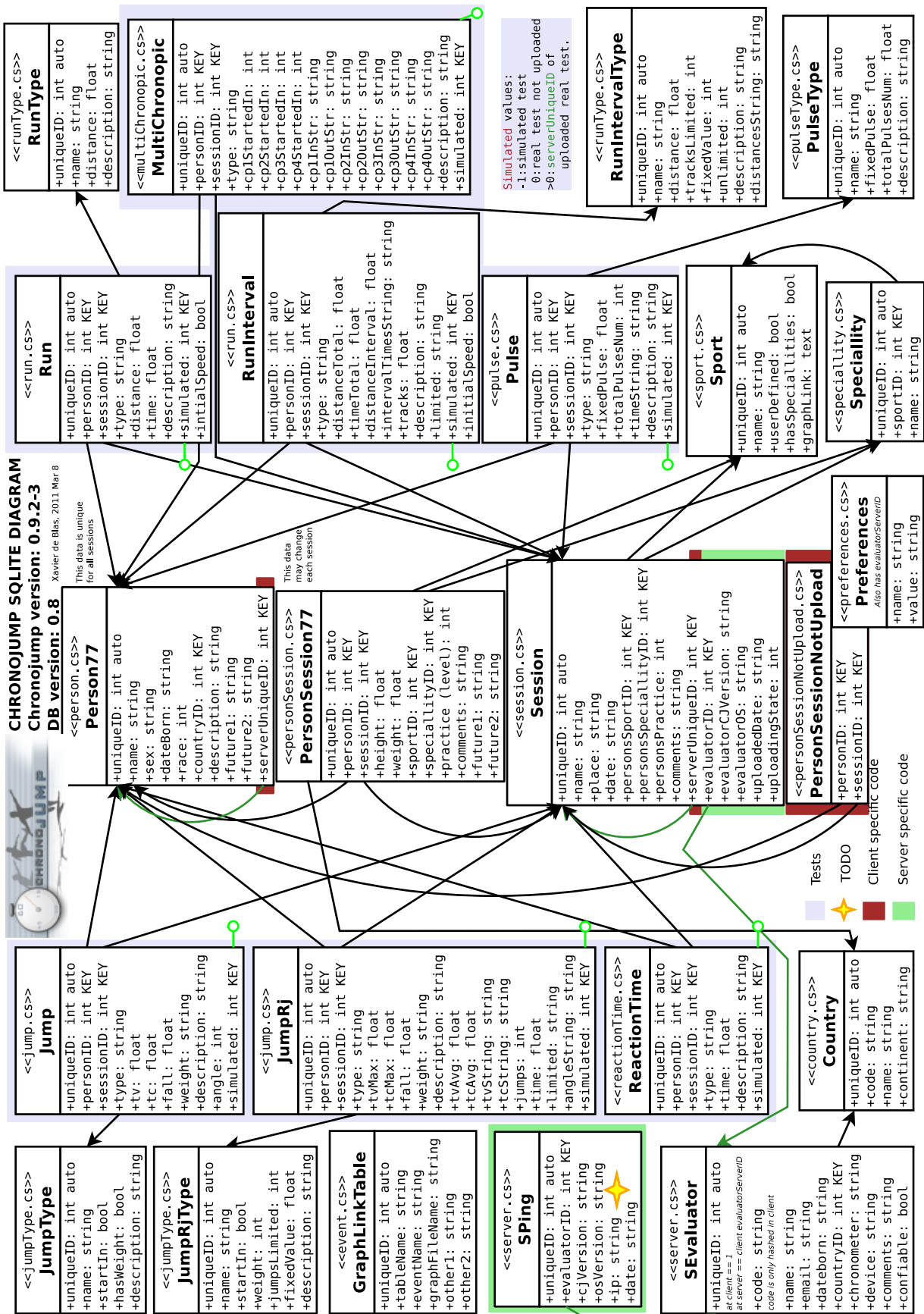


Figura A.8: Diagrama de la base de datos de Chronojump.

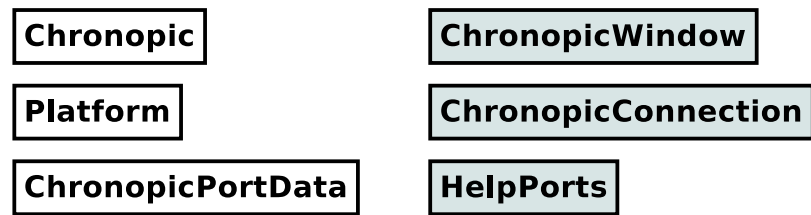


Figura A.9: Diagrama de clases de Chronojump relacionadas con Chronopic.

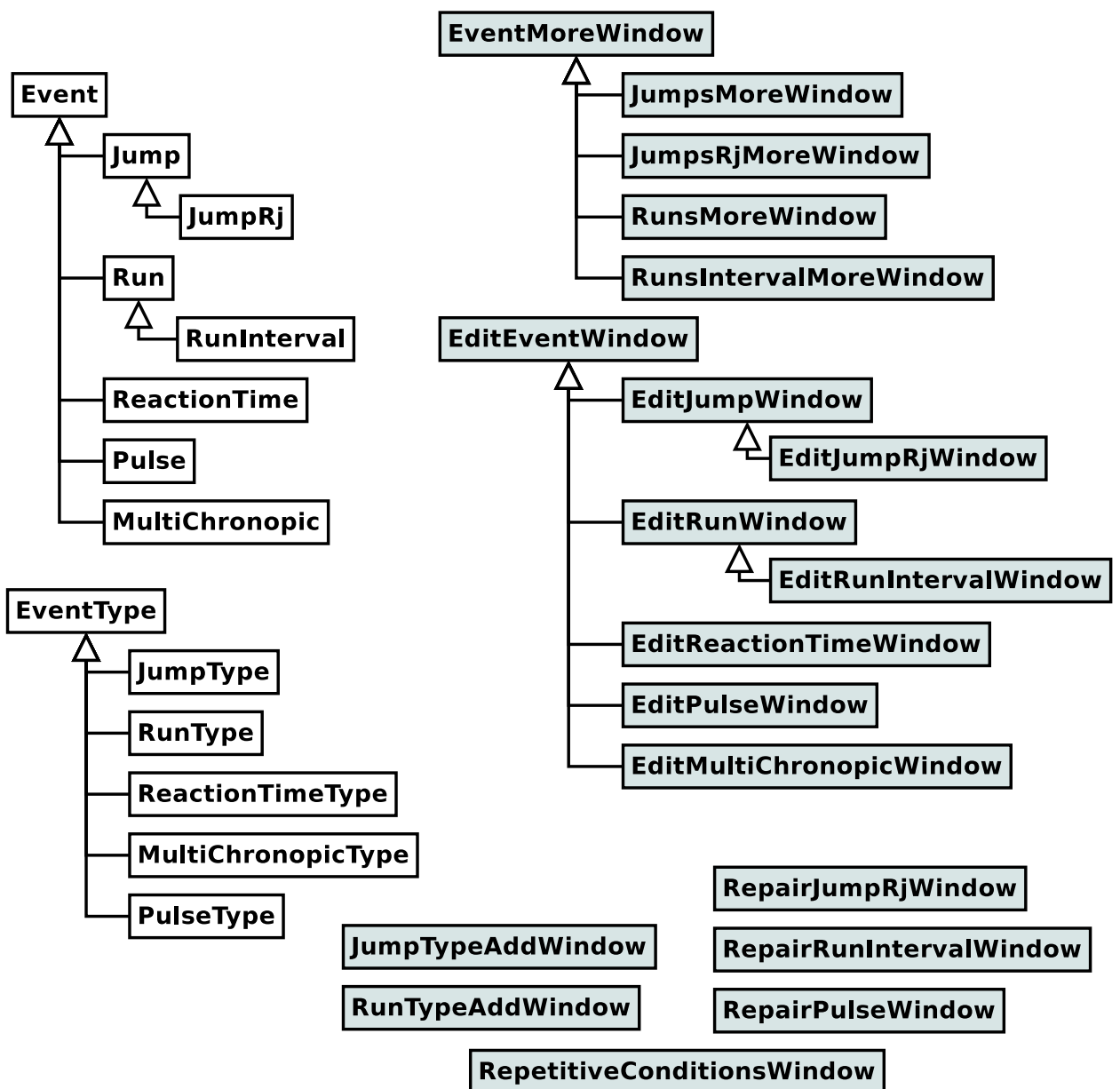


Figura A.10: Diagrama de clases de Chronojump relacionadas con el manejo de tests.

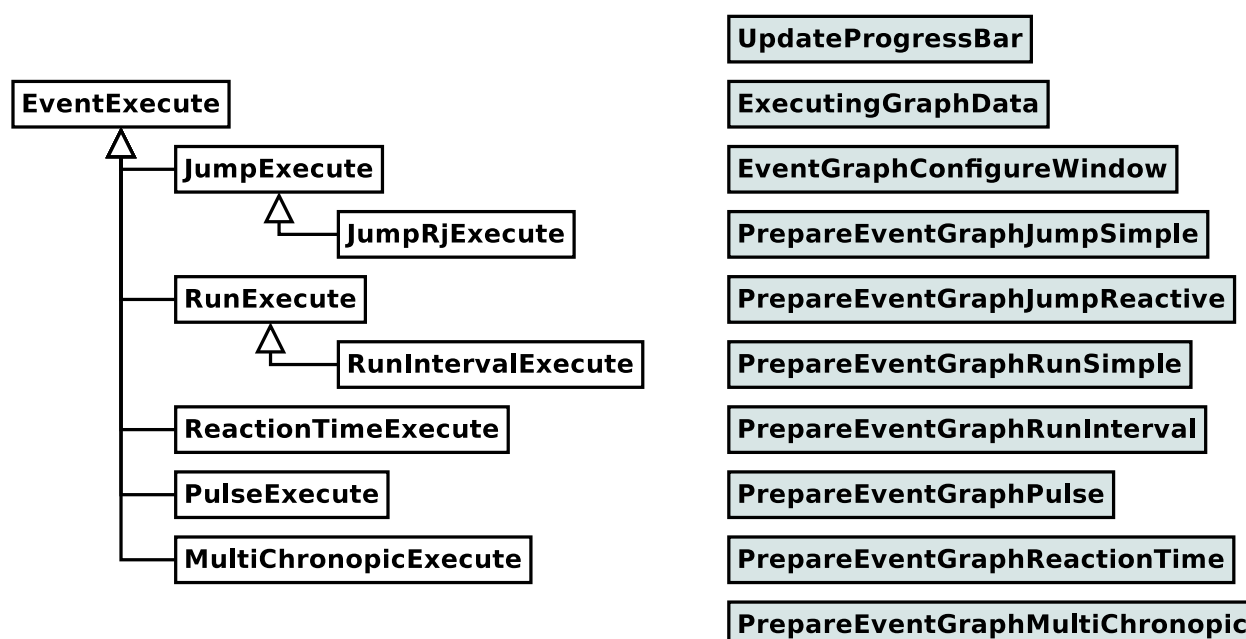


Figura A.11: Diagrama de clases de Chronojump relacionadas con la ejecución de tests.

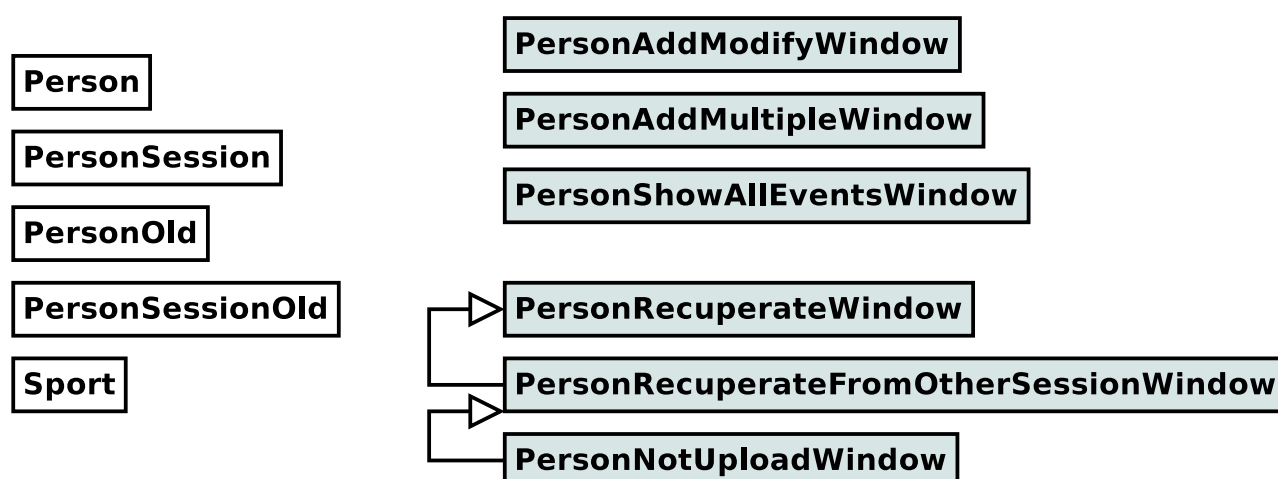


Figura A.12: Diagrama de clases de Chronojump relacionadas con el participante.

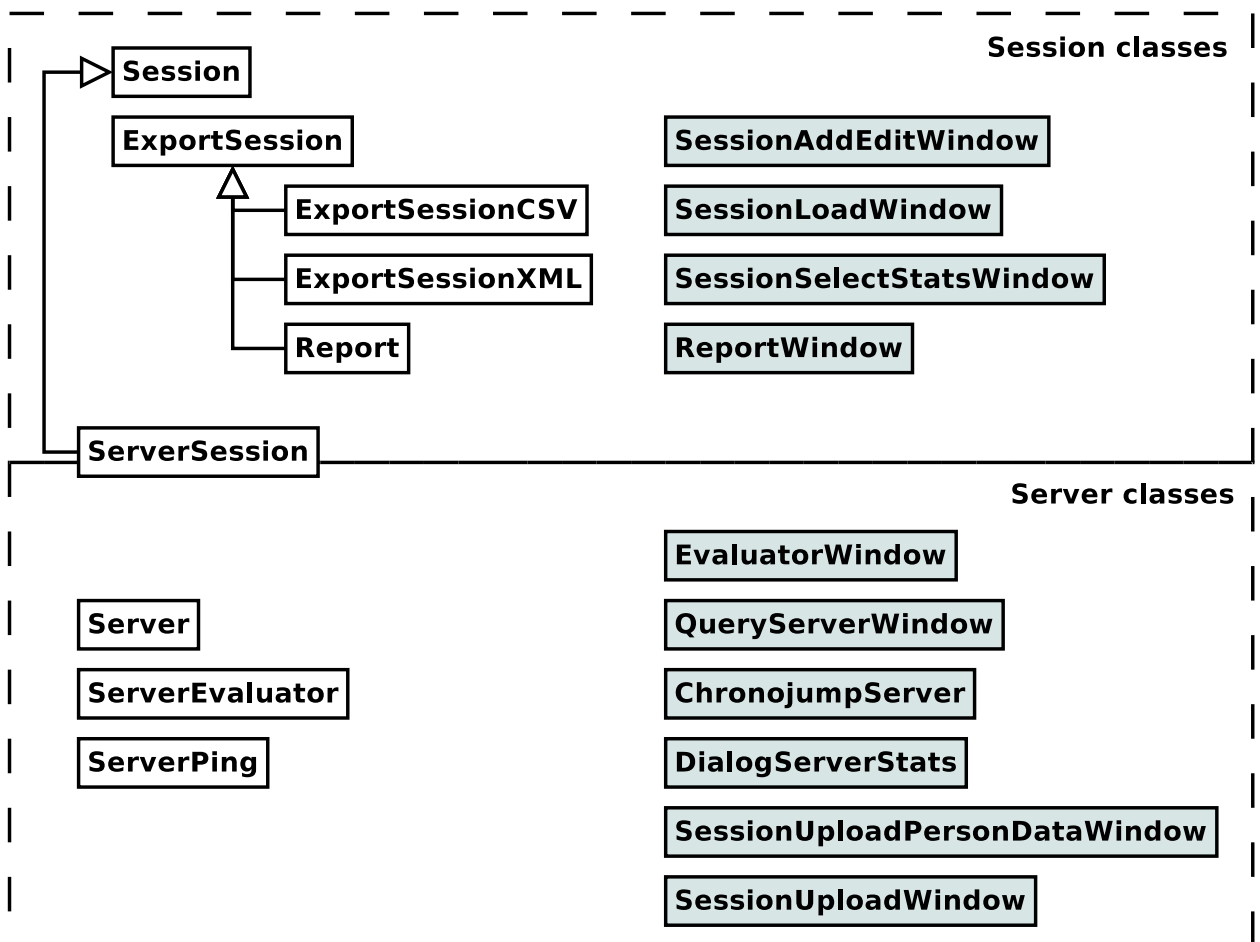


Figura A.13: Diagrama de clases de Chronojump relacionadas con la sesión y el servidor.

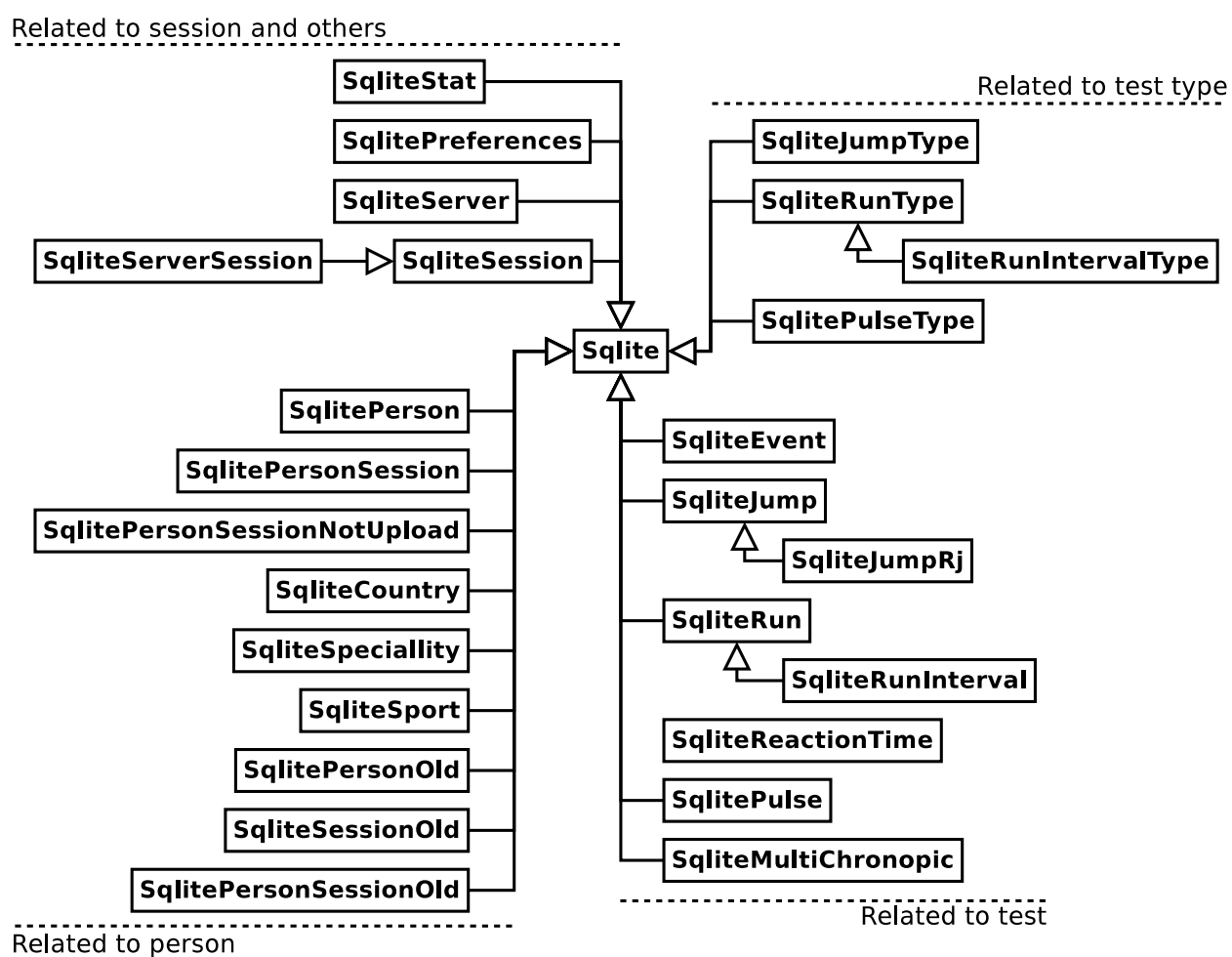


Figura A.14: Diagrama de clases de Chronojump relacionadas con la base de datos.



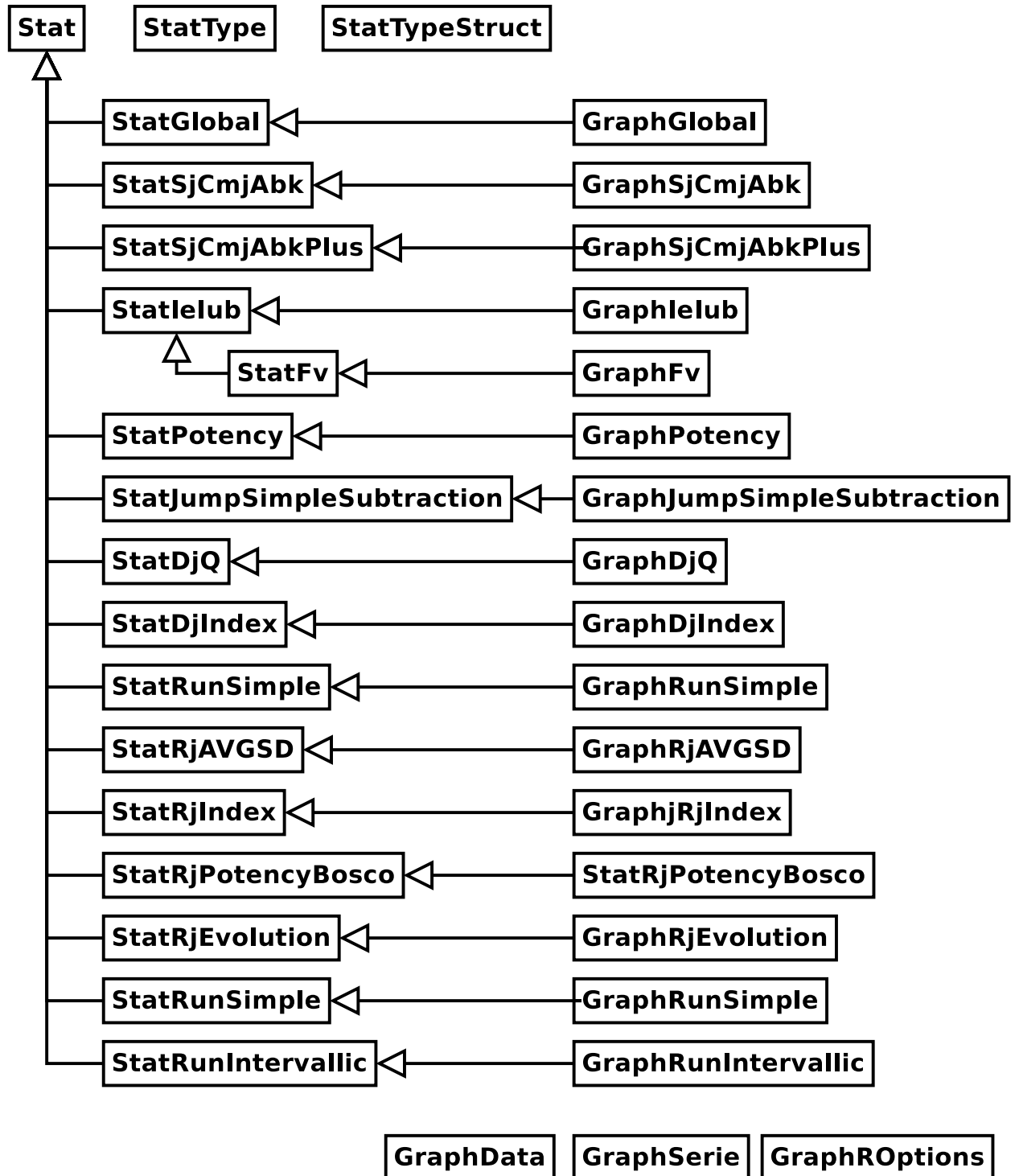


Figura A.15: Diagrama de clases de Chronojump relacionadas con las estadísticas.

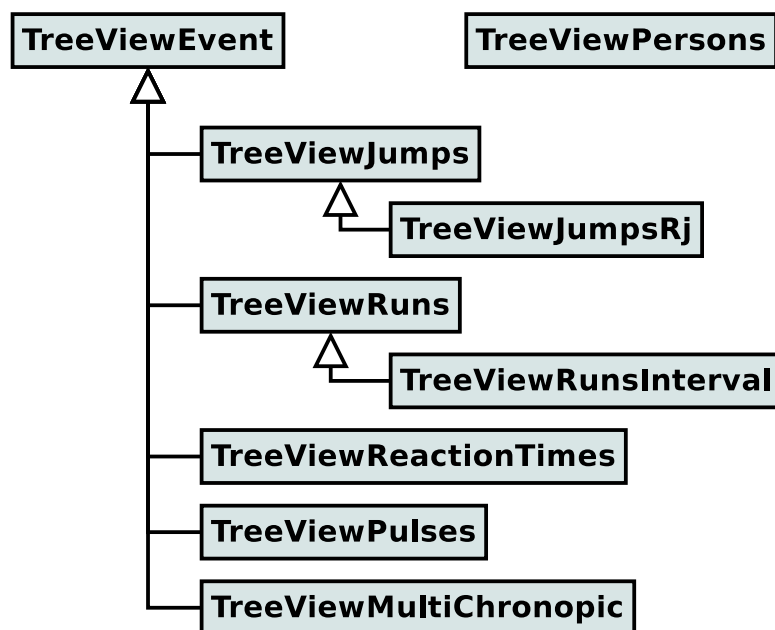


Figura A.16: .Diagrama de clases de Chronojump relacionadas con las vistas de árbol

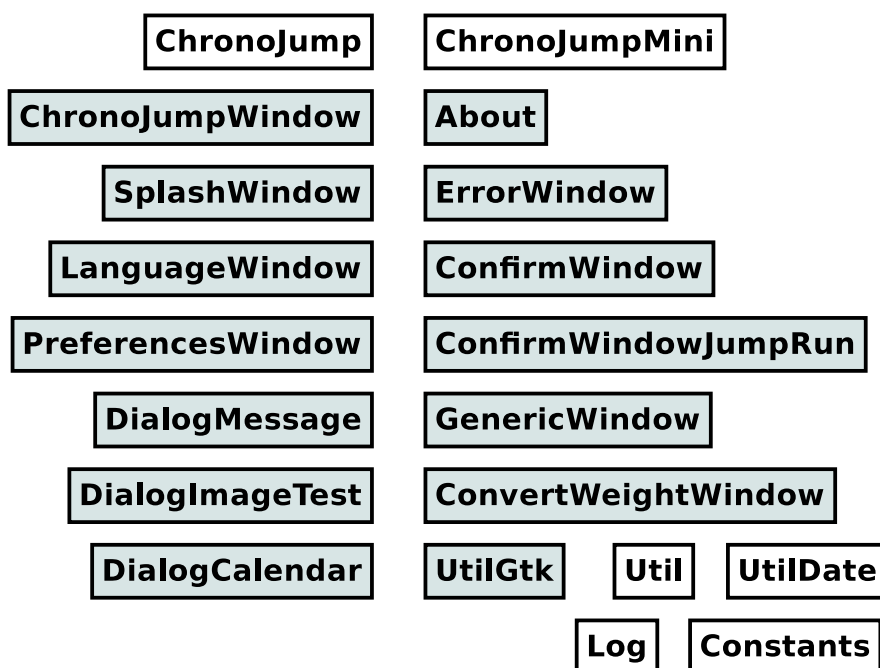


Figura A.17: .Diagrama de clases de Chronojump relacionadas con otras funcionalidades

## B. Gráficos generados por el servidor Chronojump-Server

A continuación se incuyen los gráficos generados por el servidor de Chronojump a día 7 de junio de 2011. Estos gráficos corresponden a las comparativas de tests. Los gráficos relativos a los evaluadores y los sujetos participantes se muestran en las figuras 12.2 y 12.3.

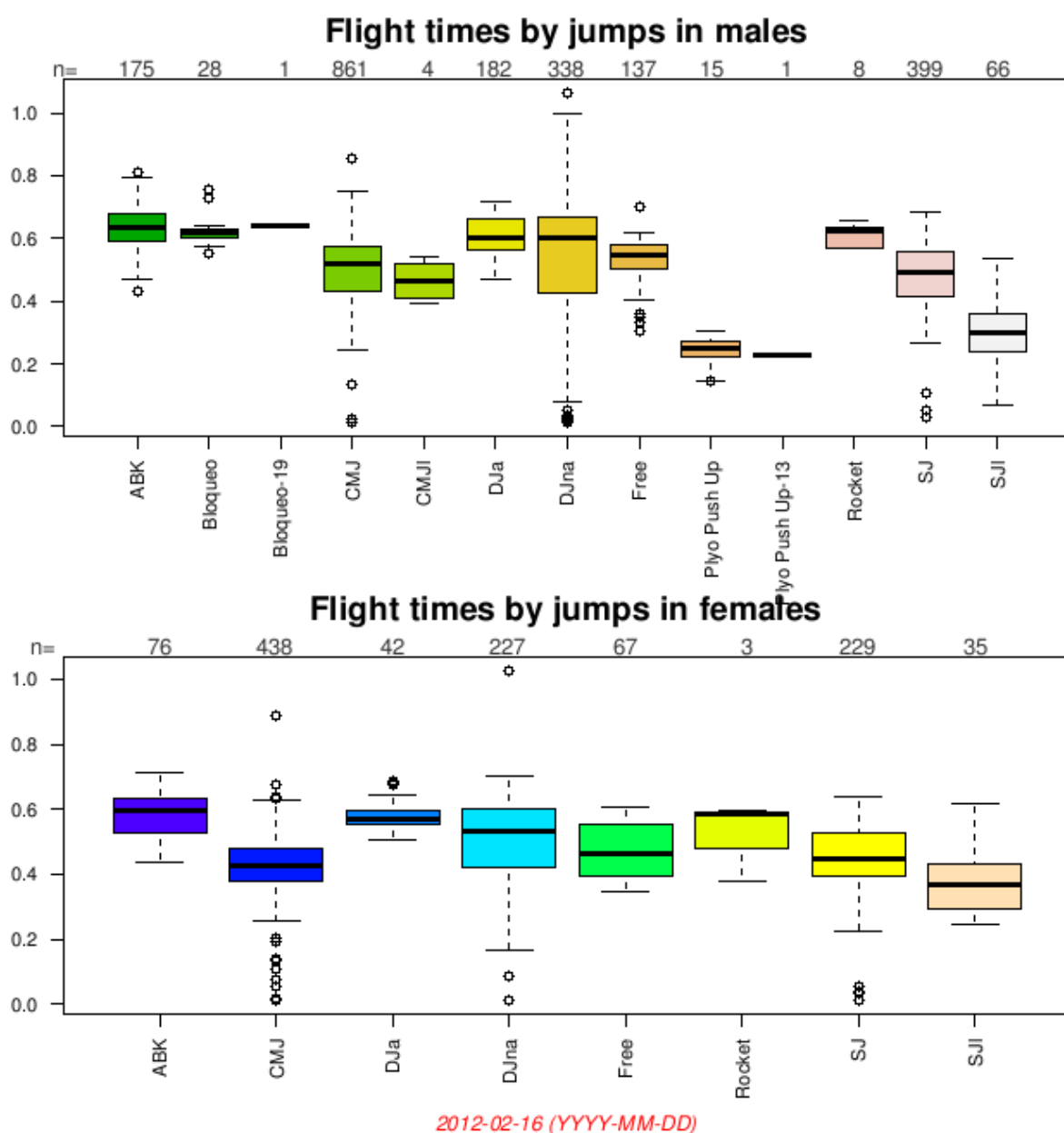


Figura B.1: Gráficas del servidor. Tiempo de vuelo en función del salto realizado.

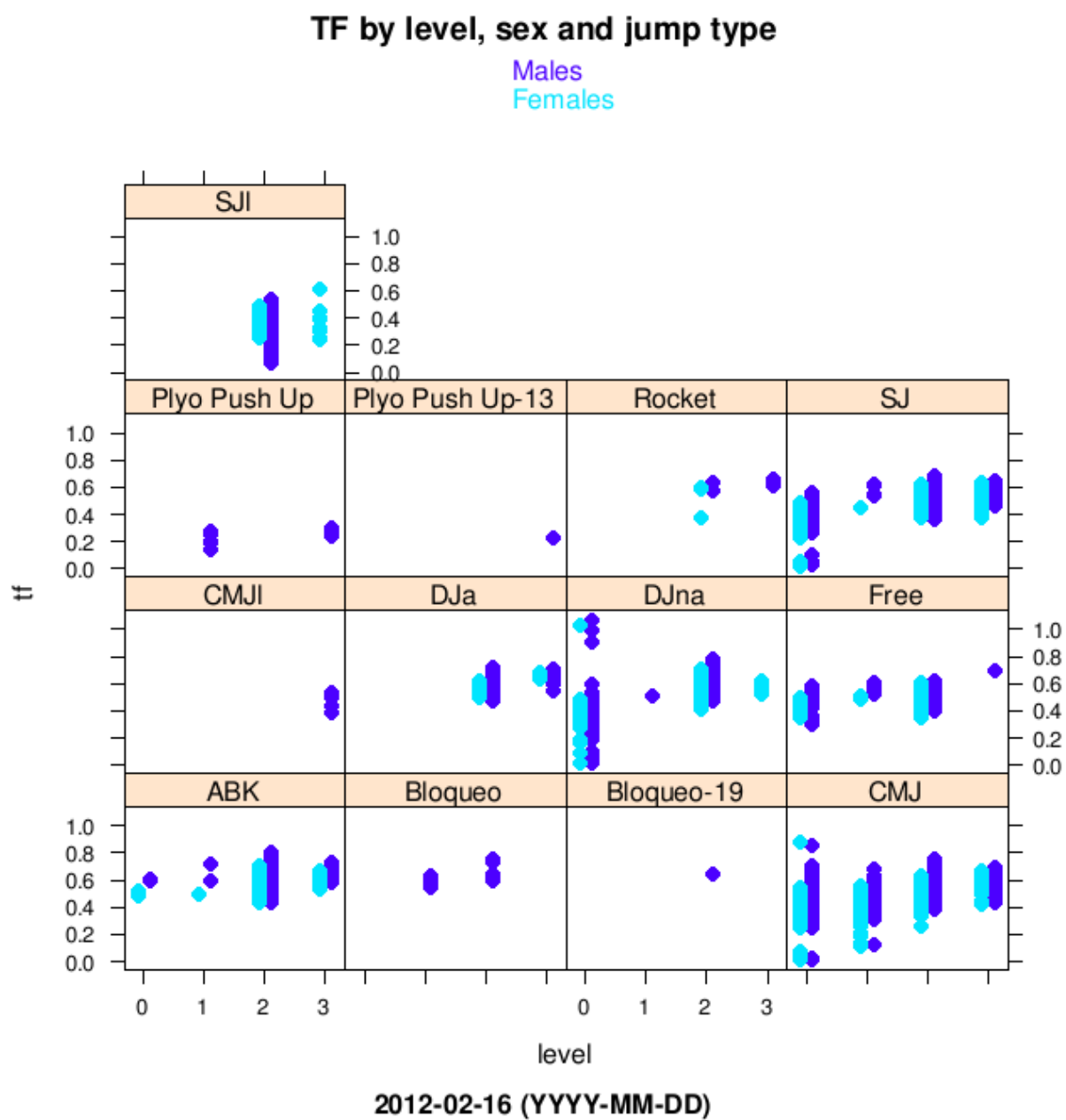
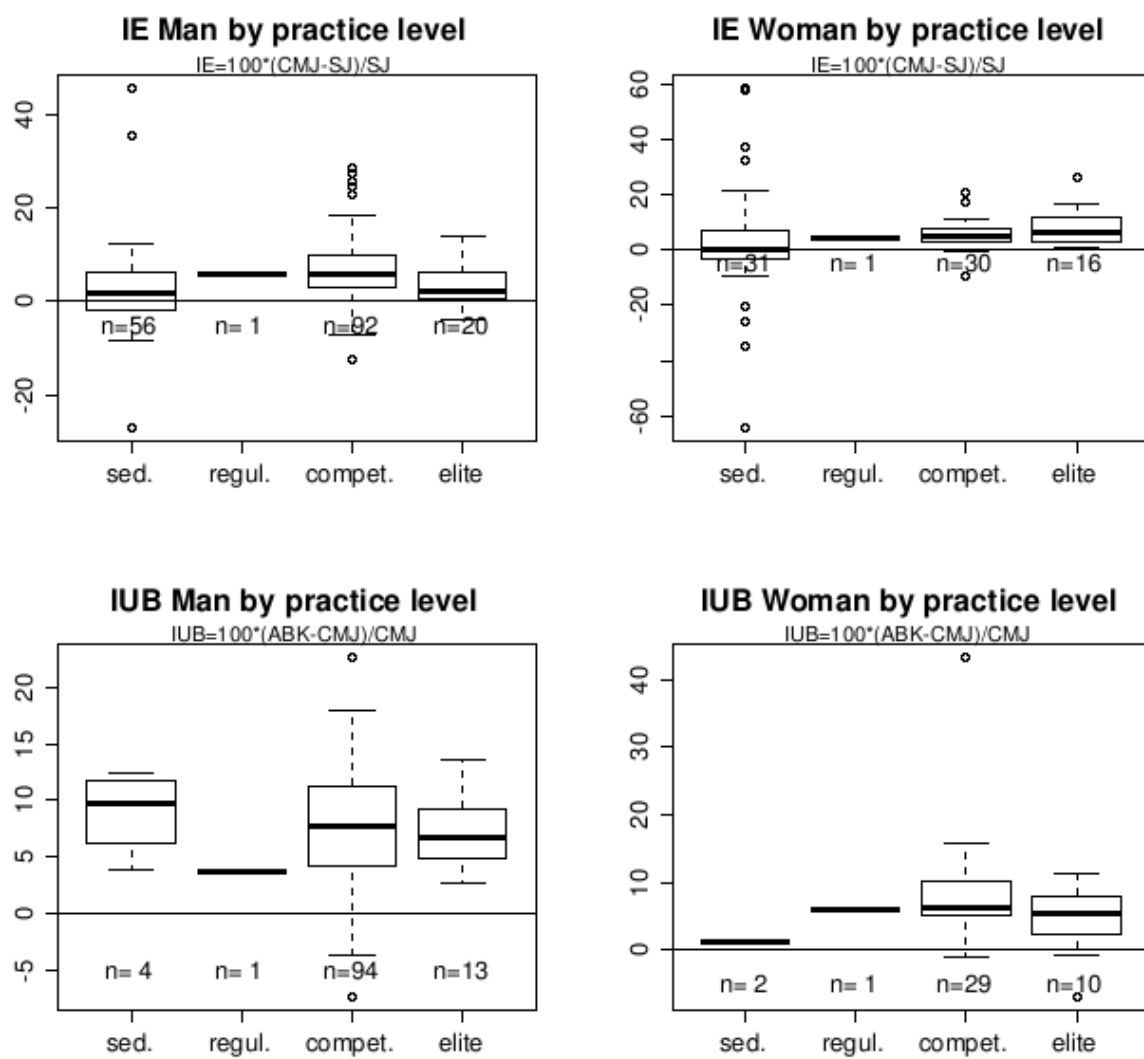


Figura B.2: Gráficas del servidor. Relación entre tiempo de vuelo, tipo de salto, género y nivel de práctica.

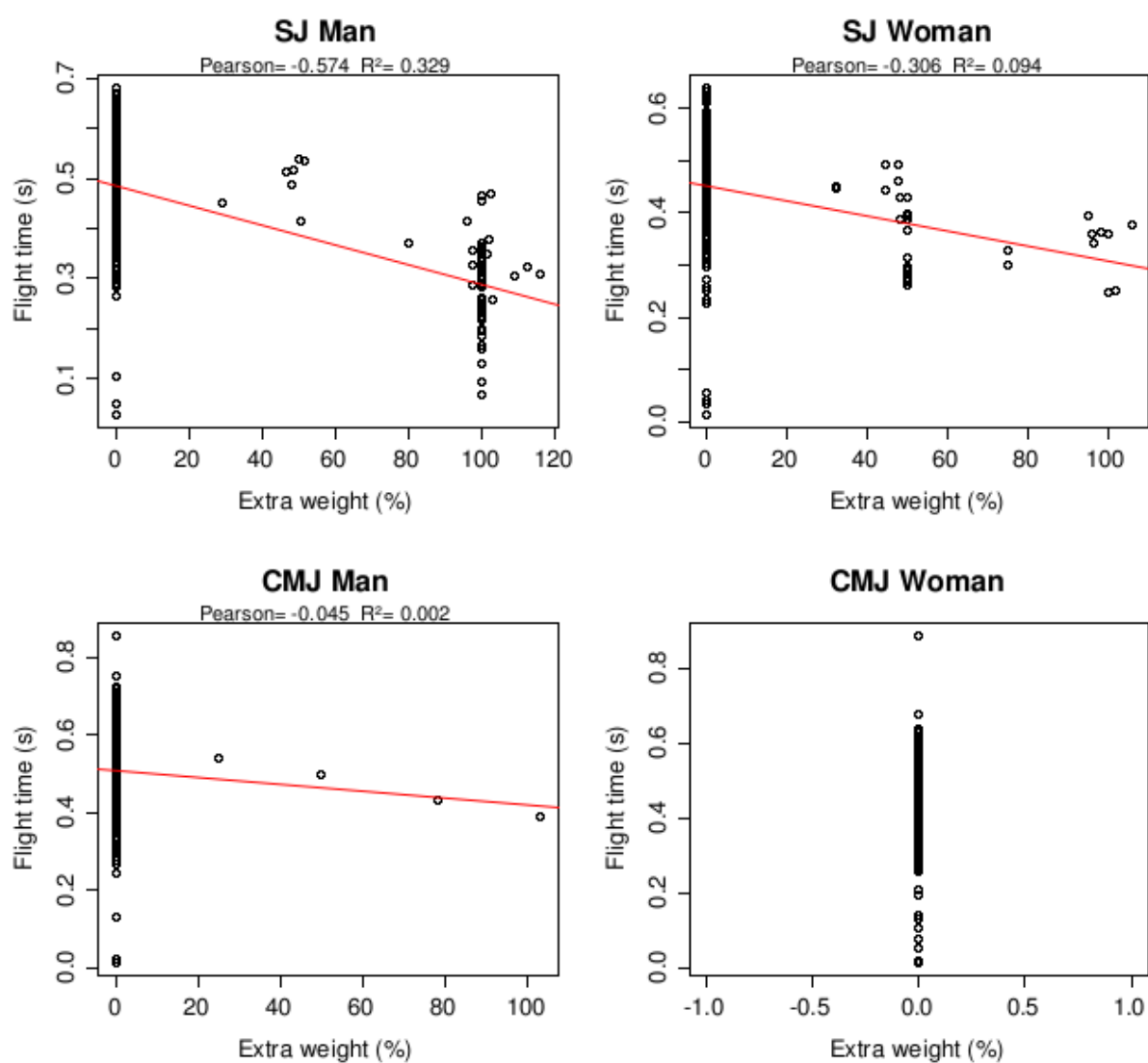
## Elasticity Index and Using Arms Index



Values are AVG of both jumps of each person on every session. Date of graph: 2012-02-16 (YYYY-MM-DD)

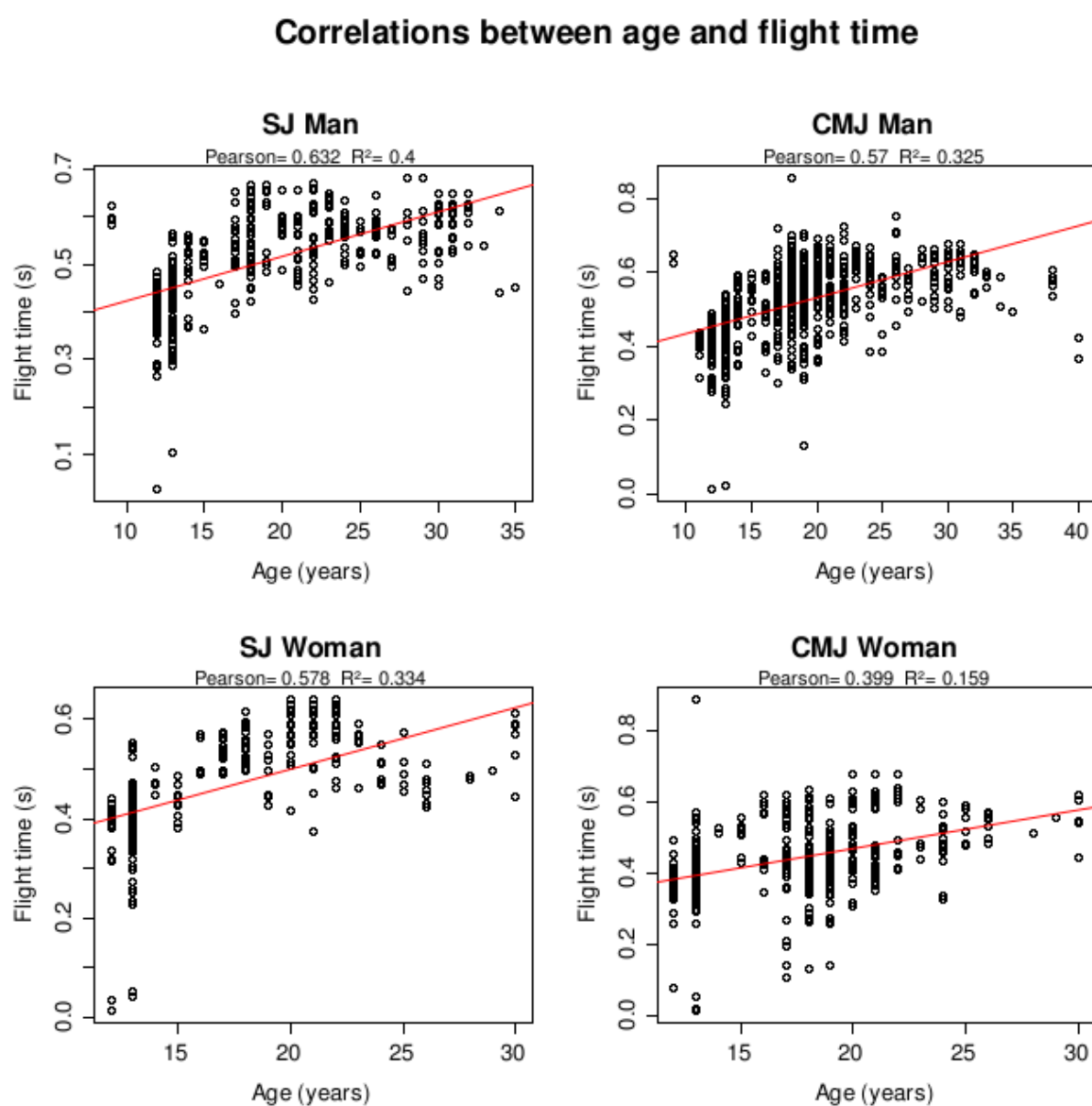
Figura B.3: Gráficas del servidor. Índice de Elasticidad e Índice de Utilización de Brazos.

### Correlations between extra weight in % and flight time



2012-02-16 (YYYY-MM-DD)

Figura B.4: Gráficas del servidor. Tiempo de vuelo en función del peso extra.



2012-02-16 (YYYY-MM-DD)

Figura B.5: Gráficas del servidor. Tiempo de vuelo en función de la edad.

## C. Listas de materiales

Tabla C.1: Materiales de Chronopic 3.

Componente	Componente en PCB	Descripción	Cantidad
Resistencia	R2, R4, R5, R6	330R	4
Resistencia	R3	174R	1
Resistencia	R1	10K	1
Choque	CH1	VK200 <sup>b</sup>	1
Condensador electrolítico	C3	1uF	1
Condensador cerámico multicapa	C1, C2	100nF	2
Condensador cerámico	C4 <sub>a</sub> , C5	22pF	2
Cristal de cuarzo de bajo perfil	X1	4MHz	1
Led verde 3mm	D1, D2		2
Led rojo 3mm	D3, D4		2
PIC16F876A	IC1		1
FT232RL	IC2		1
Pulsador pequeño	SW1, SW2		2
Harting 10 vías recto	CT1 <sub>a</sub>		1
Header vertical 2,5mm 5 ways polarizado	CP1 <sub>a</sub>		1
Clema 2 vías	CL1		1
Conector RCA hembra	CL2		1
Conector USB	X1	acodado	1

<sup>a</sup>Estos componentes fueron descartados en la fabricación taiwanesa para rebajar costes.

<sup>b</sup>El componente VK200 fue sustituido por el VK2000 en la fabricación taiwanesa por ser el primero muy inespecífico.



---

## D. Ejemplos de aportaciones de la comunidad

### D.1. Ejemplos en el Foro de Chronojump

Título: Repeated-sprint ability

Autor: xavitecnologys

Fecha: 2-6-2011

URL: <http://foro.chronojump.org/showthread.php?tid=104>

Hola,

Abro este nuevo tema para pedir ayuda y opiniones sobre esta metodología de entrenamiento.

Estoy tratando de confeccionar una mejora del chronojump, mediante la incorporación de test RSA. Primero explicare que es lo que estoy haciendo ahora y cuáles son los objetivos de esta mejora.

Ahora, estoy recopilando información sobre validaciones de distintos tests de RSA, en los que he destacado 4 principalmente: - Test de Bangsbo (3 cambios de dirección) - Test de Spencer (Lineal) - Test de Impellizzeri (1 cambio de dirección) - Test Yo-Yo, aunque este estoy pensando en eliminarlo tratándose del casi el mismo que Impellizzeri, y no está tan determinado como un RSA, sino com un test progresivo de alta intesdad.

A modo de curiosidad, ayer leí sobre el Square test (3 cambios de dirección, pero todos  $>90^\circ$ ), que es principalmente de agilidad, pero me parece interesante poderlo pasar a RSA, simplemente añadiendo repeticiones. Os dejo la referencia: Gaya A, Silva G. Projeto Esporte Brasil. Manual de Aplicação de Medidas e Testes, Normas e Critérios de Avaliação. Porto Alegre; 2007.

Los test estudiados, son cerrados, en los que marcan un  $n^\circ$  X de repeticiones con un tiempo X de descanso. A modo de funcionalidad creo que sería mejor si se pudiera dar la opción de definir el  $n^\circ$  de repeticiones que se quieren y el descanso que se quiere dar, dependiendo del momento de la temporada a lo mejor nos interesará poder modificar estas 2 variables para poder jugar con la densidad de la sesión. La variable de controlar las series no la incluyo porqué simplemente es repetir el mismo test al cabo de X minutos, que el programa no puede controlar (por ahora).

A parte de estos tests estandarizados, creo que es interesante contemplar una opción de "modo libre", donde se indique la distancia total i el  $n^\circ$  de cambios de direcciones, que en el caso de ser 1 ó 2, se podría incluso introducir la distancia entre tramos, ya que podríamos controlarlos mediante chronopics. Digo 1 ó 2 porqué, más de dos implicaría la utilización de un  $5^\circ$  chronopic, que si no me equivoco, no se puede, corrígeme si me equioco Xavi.

A parte de los distintos tests, hay otro aspecto importante, que quizá es más tema de programación, que es la posibilidad de poder decir el  $n^\circ$  de atletas que van a participar, sabiendo que dependiendo del test y del tiempo de descanso debería estar restringido a un máximo. La opción de las campanas puede

ser buena para dar las salidas, incluso para eliminar un chronopic (y en algún caso, darle otro uso en los cambios de dirección). Esta parte, es interesante, sobre todo en deportes de conjunto.

Con todo esto lo que quiero hacer, es un excel, donde poder introducir todas estas variables más las clásicas de un test RSA: todos los tiempos, mejor tiempo, peor tiempo, desviación estándar y los 4 índices de fatiga (Bangsbo, Wragg, Impellizzeri, Fitzsimons). El excel aún está bastante verde, cuando este un poco más evolucionado, colgaré una primera parte, con las variables más clásicas, cuando la termine y lo siguiente será a empezar a introducir las variables de los test 1 por 1, para que le echéis un ojo y podáis opinar y criticar (espero que no sea más tarde de la semana que viene).

Esto es de momento lo que se pretende hacer, estoy abierto a todo tipo de opiniones, observaciones, críticas y por supuesto, colaboraciones por parte vuestra. Si necesitáis alguna referencia bibliográfica sobre el tema, pedirla y si la tengo la subiré.

Saludos!

Título: RE: Petición de funcionalidades de Chronojump

Autor: pavlomg

Fecha: 11-7-2011

URL: <http://foro.chronojump.org/showthread.php?tid=5&pid=376#pid376>

Por fin he podido hacer pruebas en la pista con varios patinadores. Coloqué la célula a distintas alturas, teniendo en cuenta que el reglamento de patinaje exige que esté entre 15 y 30 cm del suelo. Me encontré con otros problemas añadidos a los que me temía: Si la pongo en una distancia cercana a los 30 cm., los patinadores mayores cortan la célula hasta 3 veces (una con la pierna derecha, otra con la punta del patín derecho y otra con la otra pierna, mientras que los mas pequeños la cortan dos veces). Si la pongo a 15 cm. hay algunos que la saltan y no detecta nada, por lo que esta altura queda completamente descartada. El mejor resultado me sale poniéndola a 20 cm. En este caso algunos la cortan una vez y otros dos. La solución que encontré momentáneamente es realizar una carrera de tramos, marcando que los deportistas harán 2 tramos. Si el patinador corta dos veces la célula, no hay ningún problema, en cuanto llegue a la otra cerrará el ciclo, suma los dos tiempos y ya tenemos el tiempo total. Si el patinador corta una vez la célula de salida, tendremos que pulsar el botón del chronopic a mano y así marcará el segundo paso y cuando llegue a meta hará el cierre y todo bien. Es una solución un poco precaria pero de momento sirve. El tema de manipular el tornillo de retardo de la célula, no sirve porque sólo se mueve entre 5 y 10 milisegundos, mientras que las múltiples detecciones de la célula de salida rondan entre los 50 y 100 milisegundos, por lo que no hacemos nada moviendo este parámetro.

Sigo pensando que la mejor solución sería un añadido en el software que permitiera un retardo, a determinar, entre detecciones. Otra solución sería usar dos chronopic, uno para cada célula y que el primer se encargara solo de echar a andar el crono y el segundo de pararlo; pero esto, creo, que también requiere ajustes en el software, por lo que es complicar la solución tontamente.

## D.2. Informe de fallos

Los usuarios del software raramente han publicado un informe de fallo en la web del proyecto. A los usuarios siempre les ha sido más cómodo enviar correos, hablar por teléfono, o esperar a tener una conversación personal en la que en ocasiones se transmite el fallo o conjunto de fallos encontrados. La figura D.1 es un ejemplo de un informe de múltiples fallos y peticiones de mejora recogido el 17 de junio de 2011.

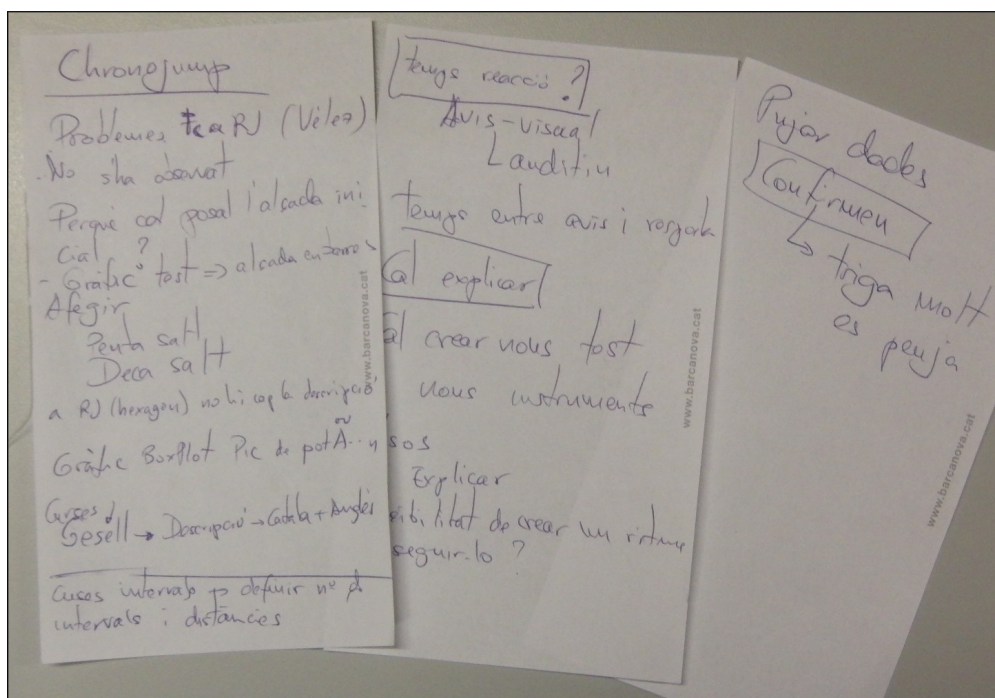


Figura D.1: Ejemplo de informe de fallos enviado por un usuario.

La tabla D.1 muestra el informe de fallos introducido en la web después de discutir los diferentes ítems.

Tabla D.1: Informe de fallos en la web a partir de la discusión con el usuario del documento de la figura D.1.

ID	Severidad	Prioridad	SO	Sumario
652846	Mejora	Normal	Todos	Improve info on dj falling height.
652847	Mejora	Baja	Todos	Show height on jumps, jumpsRj execute.
652848	Mejora	Baja	Todos	Bars on execute graphs.
652849	Trivial	Normal	Todos	Not confuse triple jump with triple jump.
652850	Mejora	Baja	Todos	Improve hexagon description.
652851	Mejora	Alta	Todos	Better gui on new jump test distance.
652852	Mejora	Baja	Todos	Add metronome.
652853	Normal	Urgente	Todos	Network progresssbar: If net problems don't loose responsitivity.
652854	Mejora	Normal	Todos	Photocells with AND.
652855	Mejora	Normal	Todos	Multichronopic sync.
652856	Mejora	Alta	Todos	Reaction time signal.

Cada uno de los *bugs* dispone de una descripción más detallada que el sumario que se muestra.

---

## E. Documentos de texto vinculados al proyecto

### E.1. Consentimiento informado de los experimentos (catalán)

### E.2. Documentos de información para los miembros de la muestra en los experimentos (catalán)



Facultat de Psicologia, Ciències  
de l'Educació i de l'Esport Blanquerna  
Universitat Ramon Llull

Document d'informació per als membres de la mostra

#### Títol:

“Fiabilització de les plataformes de contacte Chronojump Boscosystem”.

#### Identificació dels investigadors

Els investigadors: Xavier de Blas i Josep Maria Padullés, són professors de diferents facultats de ciències de l'activitat física i l'esport de Barcelona. El primer exerceix a la Facultat de Psicologia, Ciències de l'Educació i de l'Esport Blanquerna de la Universitat Ramon Llull, mentre que el segon és docent a l'Institut Nacional d'Educació Física de Catalunya adscrit a la Universitat de Barcelona.

Dins el currículum dels professionals, val a destacar la seva vinculació amb les tecnologies de mesura en l'esport, i en concret amb els microcontroladors, les plataformes de contacte i les càmeres d'alta velocitat.

#### Objectius de l'estudi

El present estudi és part de la Tesi Doctoral dirigida per Myriam Guerra i que té com a objectiu la construcció d'una eina informàtica per a la mesura dels temps de contacte i vol en el salt vertical així com la detecció de l'angle de flexió de les cames en la fase prèvia al vol.

La part de la Tesi Doctoral a la que es refereix aquest estudi comprèn la fiabilització de les plataformes de contacte que formen part de l'eina informàtica proposada. Aquestes plataformes són: Plataforma rígida Din A2, plataforma rígida Din A3 i plataforma flexible.

Els objectius de l'estudi són els següents:

1. Esbrinar la pressió mínima que cal exercir per a realitzar contacte en diferents punts de la plataforma.



Facultat de Psicologia, Ciències  
de l'Educació i de l'Esport Blanquerna  
**Universitat Ramon Llull**

## Consentiment informat

**Títol: “Fiabilització de les plataformes de contacte Chronojump Boscosystem”.**

Jo ....., amb DNI .....  
(Nom i cognoms del participant) (DNI del participant)

He llegit el full d'informació que m'ha estat lliurat.

He rebut suficient informació sobre l'estudi.

He pogut fer preguntes sobre l'estudi a Xavier de Blas i Josep Maria Padullés, investigadors d'aquest estudi.

Comprenc que la meua participació és voluntària, i que puc retirar-me de l'estudi

- quan vulgui
- sense donar explicacions
- sense que això tingui repercussió en l'àmbit acadèmic.

Si mentre es fa la prova es produeix algun incident fortuït, permeto decidir a els investigadors responsables sobre les accions a prendre.

Dono lliurement la meua conformitat per a participar en aquest estudi,  
AUTORITZANT als investigador: Xavier de Blas i Josep Maria Padullés, a difondre la informació i les imatges que se'n derivin d'aquesta prova sempre amb voluntat i interès sanitari, docent i científic, i entenent que es salvaguarda la meua identitat i intimitat en tot moment.

Els investigadors conservaran els registres que es derivin al llarg de la prova, així com la informació que se'n derivi dels mateixos, en els termes legalment previstos.

.....  
(Signatura participant) (Data)

.....  
(Signatura investigadors: Xavier de Blas, Josep Maria Padullés) (Data)

Figura E.1: Validación de las plataformas de contacto. Consentimiento informado.



Facultat de Psicologia, Ciències  
de l'Educació i de l'Esport Blanquerna  
**Universitat Ramon Llull**

## Consentiment informat

**Títol: “Detecció de l’angle de flexió de les cames en salt en contramoviment a partir de filmació lateral sense marcadors”.**

Jo ....., amb DNI .....  
(Nom i cognoms del participant) (DNI del participant)

He llegit el full d'informació que m'ha estat lliurat.

He pogut fer preguntes sobre l'estudi.

He rebut suficient informació sobre l'estudi.

He parlat amb en Xavier de Blas, investigador principal d'aquest estudi.

Comprenc que la meua participació és voluntària, i que puc retirar-me de l'estudi

- quan vulgui
- sense donar explicacions
- sense que això repercuteixi en la meua atenció com a estudiant.

Si mentre es fa la prova es produeix algun incident fortuït, el voluntari deixaria decidir a l'investigador principal o responsables presents les accions a prendre.

Dono lliurement la meua conformitat per a participar en aquest estudi,

AUTORITZANT a l'investigador principal: Xavier de Blas, a difondre la informació i les imatges que se'n derivin d'aquesta prova sempre amb voluntat i interès sanitari, docent i científic, i

EXIGINT que es salvaguardi la meua identitat i intimitat en tot moment.

L'investigador principal conservarà els registres que es derivin al llarg de la prova, així com la informació que se'n derivi dels mateixos, en els termes legalment previstos.

.....  
(Signatura participant) (Data)

.....  
(Signatura investigador: Xavier de Blas) (Data)

Figura E.2: Creación de un modelo de predicción del ángulo de la rodilla. Consentimiento informado.

2. Comparar els temps d'activació/desactivació de les plataformes proposades respecte a la plataforma del sistema Ergojump Boscosystem i entre elles.

## Metodologia i procediments a seguir

### Metodologia del primer objectiu

Per al primer objectiu s'usarà la galga extensiomètrica del sistema *MuscleLab*, el microcontrolador *Chronopic* i un ordinador personal. El procediment serà el següent:

1. En primer lloc es calibrarà la galga del *MuscleLab* amb pesos coneguts.
2. Tot seguit es connectarà la plataforma al microcontrolador *Chronopic*, que alhora estarà connectat a un ordinador per a rebre alimentació.
3. Finalment es col·locaran pesos de 2 Kg i 1 Kg fins a determinar el mínim de força per a tancar el circuit de la plataforma de contactes.
4. El tercer pas es repetirà en diferents punts de cada plataforma i en diferents plataformes.

### Metodologia del segon objectiu

Per al segon objectiu s'usaran els següents instruments:

- Dos ordinadors personals.
- Dos microcontroladors *Chronopic*.
- Una plataforma de contactes *Ergojump Boscosystem* doblegada per a que la distància entre les varetes sigui de 5 cm.
- Tres plataformes de contactes *Chronojump Boscosystem*: dues rígides Din A2 i una flexible.

Les plataformes es disposaran en tres espais tal i com s'il·lustra a continuació:

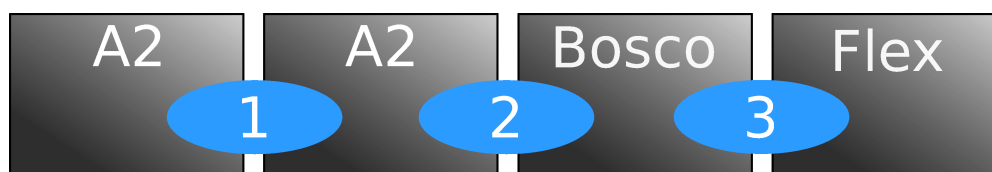


Figura E.3: Disposició de les plataformes de contacte.

L'experiment serà el següent:



- Es dividiran els atletes en dos grups. Un dels grups realitzarà els salts passant per les situacions 1->3 i l'altre ho farà en sentit invers.
- Els atletes escalfaran de forma lliure.
- Els atletes tindran un peu a cada plataforma.
- El salt realitzat serà el CMJ submàxim doncs és senzill de realitzar i així es més fàcil prevenir asimetries. A més els atletes es troben prou familiaritzats amb aquest salt.
- Es realitzaran dos salts en cada situació.

El registre es realitzarà amb un *Chronopic* connectat a cada plataforma i a cadascun dels ordinadors amb el software *Chronojump*.

A continuació, ja sense la presència dels practicants, es compararan els resultats dels diferents salts per tal de determinar la fiabilitat de les diferents plataformes de contactes gràcies a l'eina estadística *R*.

## **Beneficis esperats pel subjecte i/o la societat**

Un cop acabat l'estudi, el subjecte coneixerà les següents dades cinemàtiques pertinents a la seva execució:

- temps de vol
- alçada de vol

El practicant podrà comparar les dades descrites amb les dades poblacionals que existeixen i les que es derivaran de la base de dades poblacional que es crea en la Tesi Doctoral a la que es vincula aquest projecte.

El producte del que se'n desprèn la Tesi Doctoral que es presentarà suposa una aportació per a les branques de Biomecànica esportiva, Teoria de l'Entrenament, i Fisiologia. Aquestes disciplines es beneficiaran d'una eina que permetrà extreure dades estadístiques a partir del temps de vol i el temps de contacte detectats en les plataformes de contacte que es fiabilitzen en aquest estudi.

Degut a que les instruccions de fabricació de les plataformes de contactes rígides està disponible a la xarxa<sup>8</sup>

i que la resta de tecnologies de Chronojump Boscosystem es poden construir per un mateix o baixar de la xarxa, es transmet al conjunt de la societat una eina que poden usar en diferents contextes.

---

<sup>8</sup>Construcción de plataforma de contactos modular de circuito impreso [http://chronojump.org/construction\\_modular\\_circuit\\_platform\\_es.html](http://chronojump.org/construction_modular_circuit_platform_es.html)

## **Incomoditats i riscos derivats de l'estudi**

Es creu que no existeixen incomoditats de l'estudi degut a que els atletes es troben acostumats a realitzar aquests salts.

Degut a que hi ha diverses plataformes de contactes, s'instal·larà una fina estora per damunt de les mateixes per tal de que els esportistes es trobin en un medi més conegut i uniforme.

Quant a riscos derivats de l'estudi, aquest són els mateixos que qualsevol activitat en que hi hagi un escalfament suau lliure i salts verticals màxims en un entorn controlat.

## **Tractament confidencial de les dades del participant**

Els noms dels practicants no seran revelats, en el cas de que es vulgui descriure en l'estudi les dades obtingudes per un subjecte en particular, el nom del mateix serà alterat i la data de naixement ocultada.

## **Caràcter voluntari de la seva participació**

Els practicants participen de forma voluntària i poden abandonar quan ho desitgin tal i com es descriu en el document de consentiment informat.

## **Assistència mèdica en cas de situacions no desitjades**

En el cas de que aparegui alguna lesió o algun altre problema derivat de l'execució del test, i que aquest requereixi d'intervenció mèdica urgent, s'avisarà als serveis mèdics que es cregui convenient per tal d'intervenir amb celeritat.

## **Aprovació de l'estudi per part de la comissió d'ètica i recerca de la FPCEE**

L'estudi es durà a terme seguint els principis continguts a la declaració d'Helsinki i successives revisions.

El protocol ha estat valorat favorable per la comissió d'ètica i recerca de la FPCEE Blanquerna de la URL.

Serà un requisit indispensable per a l'estudi que tots els participants signin el consentiment informat.

En el mateix punt del consentiment es detalla que la participació haurà de ser voluntària amb la possibilitat d'abandonar l'estudi si es desitja i que la intervenció no tindrà cap efecte negatiu en la seva salut.

## **Consentiment informat**

La informació s'aportarà verbalment i per escrit. Els subjectes tindran l'oportunitat suficient de preguntar sobre els detalls de l'estudi.

L'imprès de consentiment informat seguirà les normes contingudes a la declaració d'Helsinki, així com la informació estipulada al Títol 1, article 12 del Reial Decret 561/1993 del 16 d'abril.

## Document d'informació per als membres de la mostra

### Títol:

“Detecció de l'angle de flexió de les cames en salt en contramoviment a partir de filmació lateral sense marcadors”.

### Identificació de l'investigador

L'investigador principal, Xavier de Blas, és professor de la Facultat de Psicologia, Ciències de l'Educació i de l'Esport Blanquerna de la Universitat Ramon Llull.

Les assignatures que imparteix en l'actualitat i que es troben vinculades a l'objecte d'estudi són:

- Biomecànica de l'Activitat Física i de l'Esport
- Tractament de la Informació Digital a l'Esport

Dins el currículum del professional, val a destacar la seva experiència en la programació informàtica en diferents entorns. Per al cas que ocupa, la programació en visió per computador és cabdal.

Ajudant a l'estudi, la també professora de la FPCEE Mònica Solana, aporta els seus coneixements en la localització dels tres punts articulars cercats.

### Objectius de l'estudi

El present estudi és part de la Tesi Doctoral dirigida per Myriam Guerra i que té com a objectiu la construcció d'una eina informàtica per a la mesura dels temps de contacte i vol en el salt vertical així com la detecció de l'angle de flexió de les cames en la fase prèvia al vol.

La part de la Tesi Doctoral a la que es refereix aquest estudi comprèn la detecció de l'angle de flexió de les cames sense l'ús de marcadors.

Degut a que actualment els algorismes per la detecció de la pell humana no són prou eficaços, s'usa una malla de color negre que identifica el que són les cames. La malla es diferencia del fons i terra (blancs), així com de la samarreta del practicant (de colors clars).

Així els objectius de l'estudi són:

- Conèixer l'alçada de vol dels practicants amb malles i amb pantaló curt.

- Conèixer l'angle de flexió màxima prèvia al vol amb malles i amb pantaló curt.
- Identificar possibles desplaçaments dels marcadors de validació en el salt en malles.
- Relacionar l'alçada de vol amb l'angle de flexió màxima prèvia a la fase de vol.
- Calcular la ubicació dels punts articulars a partir de punts geomètrics de referència i de l'anàlisi estadístic de les ubicacions dels marcadors.

## Metodologia i procediments a seguir

Quant a la realització del test, després d'un escalfament, els practicants realitzaran 3 salts amb malles i 3 salts amb pantaló curt. L'ordre d'execució diferirà en els diferents participants, així uns realitzaran primer els salts amb malles i després els de pantaló curt, mentre que altres participants ho faran a l'inrevés. Pel que fa a l'execució, els salts seran en contramoviment amb una flexió de cames lliure i amb una flexió d'espatlla permesa sempre que no s'impulsin clarament amb la mateixa. Els practicants ja han estat instruïts prèviament en l'execució del salt.

Tots els salts seran registrats amb dues instrumentacions diferents:

- Contactes: Els practicants saltaran damunt d'una plataforma de contactes de les comercialitzades en el sistema *Ergo Jump*. Aquesta anirà connectada al PIC cronometrador anomenat *Chronopic* v.3, que ha estat dissenyat ad-hoc per a la Tesi Doctoral que es vincula aquest treball. El *Chronopic* enviarà a l'instant les dades temporals registrades a un ordinador amb el programari *Chronojump*, que és també un altre disseny ad-hoc. Les dades resultants seran el temps de vol i l'alçada de vol.
- Videografia: Els practicants seran registrats amb una càmera de vídeo d'alta velocitat a 300 imatges per segon. La filmació serà lateral, amb un camp focal que compregui completament les cames en bipedestació, perpendicular al desplaçament i centrada en el genoll en una flexió entre 90 i 120 graus. Previ al salt s'usarà una petita pissarra a mode de claqueta per tal d'identificar els subjectes a posteriori. S'obtindran els fotogrames de les execucions en format digital.

Es preguntarà als practicants si el seu salt ha estat *màxim*. En cas contrari, es repetiran els salts que es creguin convenients sempre tractant de que no aparegui la fatiga realitzant descansos si és necessari. Tots els salts seran registrats, però només els identificats com a *màxims* seran usats en la recerca.

A continuació, ja sense la presència dels practicants, s'analitzaran les dades obtingudes de la videografia a partir de l'eina de programari de Visió per Computador escrita ad-hoc anomenada *Knee-Angle*. Creada inicialment pels enginyers Sharad Shankar i Onkar Nath Mishra (contractats

per a aquesta tasca) i actualitzada i preparada per la validació per Xavier de Blas. Com a resultat de l'anàlisi, s'obtindrà la posició bidimensional dels tres punts estudiats en les diferents imatges, així com d'altres punts geomètrics de referència com per exemple el punt més endarrerit de les cames, o el més avançat.

Finalment es realitzarà un anàlisi estadístic amb l'eina de programari *R* per tal de trobar relacions entre els punts geomètrics i els anatòmics que permetin la predicció dels punts anatòmics amb precisió.

## **Beneficis esperats pel subjecte i/o la societat**

Un cop acabat l'estudi, el subjecte coneixerà les següents dades cinemàtiques pertinents a la seva execució:

- temps de vol
- alçada de vol
- angle de flexió prèvia

El practicant podrà comprar les dades descrites amb les dades poblacionals que existeixen i les que es derivaran de la base de dades poblacional que es crea en la Tesi Doctoral a la que es vincula aquest projecte.

Sense entrar en la resta d'aportacions de la Tesi Doctoral i centrant-nos en el que respecta al present estudi, les branques de Biomecànica esportiva, Cinesiologia, i Medicina esportiva entre d'altres, es beneficiaran d'una eina que permet detectar la flexió de l'extremitat inferior sense marcadors. Aquesta aportació és novedosa doncs els llistats de parts del cos detectables per Visió per computador no inclouen encara les cames. Aquest anàlisi automatitzat en un ambient de camp permet des de la identificació de patologies en la marxa fins a la descripció de paràmetres de la tècnica. Tot i que les esmentades aplicacions no són directament abordades pel present estudi, doncs aquest es concreta en un salt específic, la filosofia que envolta a tota la tecnologia dissenyada (programari lliure i maquinari obert) permet que d'altres investigadors puguin adaptar el que s'ha fet a diferents aplicacions amb independència dels autors originals i sense estar subjectes a restriccions en el seu ús, modificació o redistribució més enllà de les que afavoreixin que l'eina i les seves obres derivades conservin la llicència original.

## **Incomoditats i riscos derivats de l'estudi**

Les incomoditats que poden aparèixer són:

- L'execució dels salts amb les malles proposades pot resultar incòmode per alguns dels practicants que no es troben acostumats a aquesta vestimenta. Aquesta incomoditat pot ser física i/o emotiva.
- La filmació dels salts pot incomodar emotivament a alguns dels esportistes.

La sala on s'executaran els salts està tancada de manera que es minimitza la incomoditat per factors emotius.

Per altra banda el saltar damunt d'una plataforma de contactes no es considera una incomoditat, doncs aquesta és molt fina, i difícilment pot resultar molesta per als practicants. Com a precaució s'instal·larà una fina estora per damunt de la mateixa per tal de que els esportistes es trobin en un medi més conegut.

Quant a riscos derivats de l'estudi, aquest són els mateixos que qualsevol activitat en que hi hagi un escalfament suau lliure i salts verticals màxims en un entorn controlat.

## **Tractament confidencial de les dades del participant**

Els noms dels practicants no seran revelats, en el cas de que es vulgui descriure en l'estudi les dades obtingudes per un subjecte en particular, el nom del mateix serà alterat.

Les imatges que siguin publicades en qualsevol mitjà no contindran el nom del subjecte ni en la imatge ni en el nom de l'arxiu. De la mateixa manera la cara dels executants serà ocultada.

## **Caràcter voluntari de la seva participació**

Els practicants participen de forma voluntària i poden abandonar quan ho desitgin tal i com es descriu en el document de consentiment informat.

## **Assistència mèdica en cas de situacions no desitjades**

En el cas de que aparegui alguna lesió o algun altre problema derivat de l'execució del test, els serveis mèdics de les instal·lacions on es durà a terme la prova: Pavelló Eucagest, seran avisats per tal d'intervenir amb celeritat.

## **Aprovació de l'estudi per part de la comissió d'ètica i recerca de la FPCEE**

L'estudi es dura a terme seguint els principis continguts a la declaració d'Helsinki i successives revisions.

El protocol ha estat valorat favorable per la comissió d'ètica i recerca de la FPCEE Blanquerna de la URL.

Serà un requisit indispensable per a l'estudi que tots els participants signin el consentiment informat.

En el mateix punt del consentiment es detalla que la participació haurà de ser voluntària amb la possibilitat d'abandonar l'estudi si es desitja i que la intervenció no tindrà cap efecte negatiu en la seva salut.

## **Consentiment informat**

La informació s'aportarà verbalment i per escrit. Els subjectes tindran l'oportunitat suficient de preguntar sobre els detalls de l'estudi.

L'imprès de consentiment informat seguirà les normes contingudes a la declaració d'Helsinki, així com la informació estipulada al Títol 1, article 12 del Reial Decret 561/1993 del 16 d'abril.



### **E.3. Proyecto en el Master Internacional en Desarrollo de Aplicaciones en Software Libre UOC Universitat Oberta de Catalunya**

Memoria de Proyecto Chronojump  
Master Internacional en Desarrollo de Aplicaciones en Software Libre  
UOC Universitat Oberta de Catalunya

Xavier de Blas

10 de septiembre de 2005

## **Introducción**

ChronoJump es una aplicación muy útil en el ámbito de la biomecánica de la actividad física y el deporte, materia de la que soy docente en una facultad de ciencias del deporte. Hace unos años, como becario programé una aplicación similar aunque de menores prestaciones llamada Salta con una licencia freeware. Salta funcionaba bajo MSDOS y es usada aún por algunos entrenadores de equipos de primera categoría pese a que no funciona en Windows XP.

Hoy en día, y gracias al sitio web [deporteyciencia.com](http://deporteyciencia.com), del cual soy editor, existe ya una comunidad de usuarios que preguntan cómo ir construyendo el hardware de Chronojump y proponen características nuevas. Desafortunadamente casi ninguno de los usuarios ha probado el software puesto que hasta el momento sólo es usable en GNU/Linux y es necesario bajarse las fuentes del CVS.

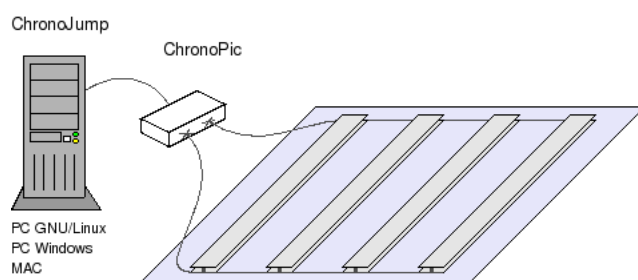
El presente proyecto representa la culminación del Master de Programación en Software Libre de la UOC. Para ello me encargo de dirigir, programar el software, y dinamizarlo (actualización de la web, listas de correo, creación de comunidad, ...)

## Herramienta Chronojump

### Descripción

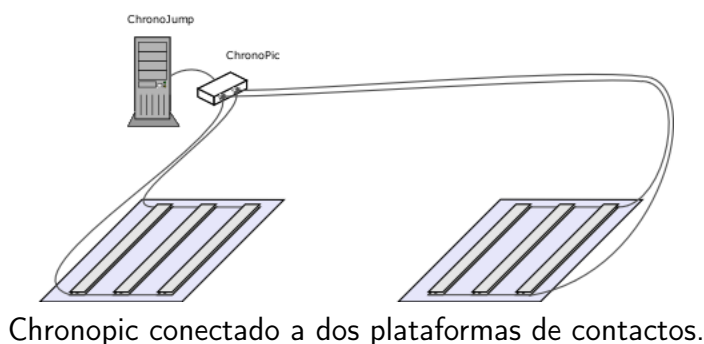
ChronoJump es un completo sistema para la medición, gestión y estadística de las fases temporales del salto. Este método se sirve de una plataforma de contactos con dos eventos (en-plataforma, fuera-de-plataforma), de un circuito impreso cronometrador diseñado ad-hoc (denominado “Chronopic”) para obtener mediciones precisas y fiables, y de un software de gestión multiplataforma.

ChronoJump permite registrar cualquier salto con las repeticiones que se desee, iniciando desde dentro o fuera de la plataforma, registrando independientemente tiempos de contacto y tiempos de vuelo, así como carga extra, altura de caída, limitación por saltos o tiempo, y descripciones de técnica. El tiempo de vuelo de un salto es indicador de la elevación del Centro de Gravedad del mismo (siempre y cuando la posición de salida y de llegada sea la misma).



Esquema de software de gestión (ChronoJump) + cronómetro (Chronopic) + plataforma de contactos.

ChronoJump debe poder usarse también para el cronometrado de carreras incluso con intervalos y para la medición del tiempo de reacción (consultar Características).



Chronopic conectado a dos plataformas de contactos.

Para el registro de datos se usa una pequeña base de datos Sqlite que le capacita para vincular saltos a sujetos y sesiones y realizar todo tipo de comparaciones entre estas entidades.

Incorpora multitud de estadísticas y gráficas de distintos saltos y tests encontrados en la bibliografía, y los grafica mediante la biblioteca gráfica NPlot.

Los datos de los saltos son exportables a hojas de cálculo gracias al formato CSV.

El cronometrador, "Chronopic" está basado en la tarjeta entrenadora Skypic<sup>9</sup> (hardware libre), y ofrece una precisión de una milésima de segundo. Otros sistemas de cronometrado del salto usan únicamente un ordenador personal conectado a la plataforma. Los ordenadores modernos son multitarea, característica que impide garantizar una medición fiable debido a que pueden estar realizando cualquier otra tarea en aquél momento.

La plataforma de contactos puede ser fabricada como se explica en el documento: "Instrucciones para la construcción de una plataforma de contactos para la medición de la capacidad de salto/s"<sup>10</sup>

Hasta el momento se han escrito 2 artículos sobre ChronoJump:

- De Blas, X (2005). Primeres aproximacions a la creació d'un mètode de mesura, gestió i estadística de la capacitat de salt a partir de l'avaluació de temps de vol i temps de contacte a una plataforma de contactes amb programari i maquinari lliure. Projecte ChronoJump. Revista Aloma. Revista de Psicologia, Ciències de l'Educació i de l'Esport, vol. 15, p.107-123. <sup>11</sup> (catalán)
- De Blas, X, González-Gómez, J. (2005) Proyecto Chronojump: Sistema de Medida de la Capacidad de Salto usando Software y Hardware Libres. Actas I Congreso de Tecnologías del Software Libre.<sup>12</sup>

Más información en la página web del proyecto Chronojump<sup>13</sup>.

---

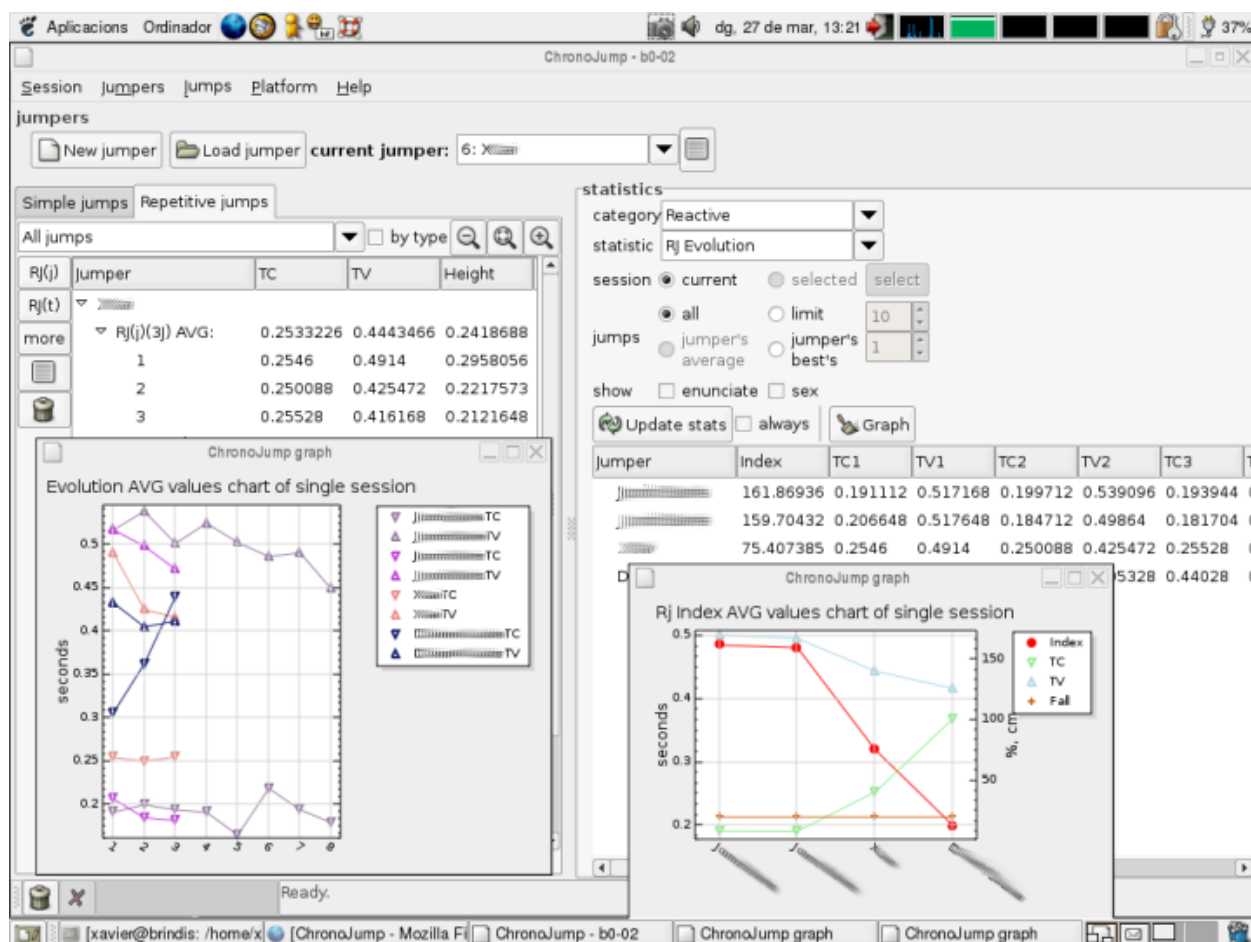
<sup>9</sup><http://www.learobotics.com/proyectos/skypic/skypic.html>

<sup>10</sup>[http://chronojump.software-libre.org/construction\\_contact\\_platform\\_es.html](http://chronojump.software-libre.org/construction_contact_platform_es.html)

<sup>11</sup>[http://chronojump.software-libre.org/articles/article\\_chronojump\\_aloma.html](http://chronojump.software-libre.org/articles/article_chronojump_aloma.html)

<sup>12</sup>\*[http://chronojump.software-libre.org/articles/chronojump\\_sistema\\_de\\_medida\\_congreso\\_gpul.pdf](http://chronojump.software-libre.org/articles/chronojump_sistema_de_medida_congreso_gpul.pdf)

<sup>13</sup><http://chronojump.software-libre.org>



Captura de pantalla v.0.1(16 junio 2005).

## **Estado de Chronojump al inicio del proyecto de master**

En fecha de inicio de 14 junio 2005, Chronojump permite...

- realizar los saltos presentes en la bibliografía: SJ, CMJ, ABK, DJ con o sin carga extra, RJ limitado por tiempo y por saltos
- crear nuevos saltos a de forma gráfica y sencilla,
- mostrar estadísticas y gráficas de los resultados de los saltos y de los índices: DJ Index, IE, IUB, RjAverage Index,
- crear y gestionar sesiones, sujetos y saltos,
- exportar a CSV,
- obtener los datos de más de una plataforma de contactos,

Existe una página web de desarrollo del proyecto<sup>14</sup> y una página de difusión<sup>15</sup> del mismo en inglés, castellano y portugués.

## **Carencias detectadas al inicio del proyecto**

En el momento de plantear el proyecto se detectó que

- Chronojump sólo funciona en sistemas GNU/Linux.
- No se ha liberado ninguna versión en .tgz ni en DEB ni RPM, únicamente se facilita acceso al CVS.
- No se encuentra disponible información de construcción del prototipo de cronómetro.
- No se usa un registro de bugs y tareas pendientes que permita coordinar las tareas, seguir su evolución y notificar resultados. Se usa un fichero de texto plano.
- El programa no está optimizado para su internacionalización.
- Se requieren más funciones para la gestión de la base de datos: buscar sujetos, juntar dos saltos en uno, juntar dos sujetos en uno, corregir un salto reactivo, ...
- La página web de difusión debería traducirse a más idiomas.
- No es posible usar células fotoeléctricas en lugar de la plataforma de contactos.

---

<sup>14</sup><http://software-libre.org/projects/chronojump/>

<sup>15</sup><http://chronojump.software-libre.org>

Exceptuando la última, que depende de colaboraciones externas que hasta el momento no han confirmado su apoyo, el resto de carencias se ha determinado que se resuelvan como tareas del proyecto.

## Proyecto Master Software Libre UOC

### Objetivos

A grandes rasgos, los objetivos del proyecto que se presenta son los siguientes:

1. Liberar una versión estable de ChronoJump en los siguientes formatos:
  - a) Empaquetada en formatos DEB y tar.gz
  - b) Compilada para plataformas Windows
  - c) Empaquetada para Mono Live CD: <http://www.mono-live.com/>
2. Crear una rama estable y una inestable en el CVS para implementar las nuevas características en la inestable.
3. Crear y mantener un sistema de petición de nuevas características y seguimiento de bugs.

Para la consecución de los mismos, se han descrito listados de tareas en el siguiente apartado.

### Planificación de Tareas

Las tareas que siguen fueron previstas al inicio del proyecto, la consecución de las mismas será discutida en el apartado E.3, mientras que la descripción y resolución de otras tareas que fueron apareciendo durante el desarrollo, se encuentra en el apartado E.3.

#### Tareas de programación

- Mejorar el soporte de Gettext para las descripciones de los estadísticos que se presentan.
- Integrar control de carreras, así el programa tendría dos partes: gestión de saltos y gestión de carreras aunque seguiría llamándose ChronoJump.
- Integrar herramientas de gestión de la base de datos (buscar saltadores, cambiar nombre de sesiones, ...).
- Preparar la release: tests, limpiar bugs.
- Crear una rama inestable e iniciar en esta la programación del zumbador, estudio de ritmos y tiempo de reacción.

### **Tareas de empaquetado**

- Estudiar como instalar el archivo de base de datos generado por Sqlite en algún directorio de usuario en los distintos sistemas operativos soportados (actualmente se usa un archivo en el directorio de ejecución del programa, o se genera de nuevo si no existe).
- Conseguir un Debian Developer que empaquete ChronoJump (software de gestión de sesiones, saltadores y saltos) y Chronopic (paquete de control del cronometrador). Una vez empaquetados para Debian, pasarían en breve al repositorio Universe en Ubuntu.
- Conseguir que Chronojump funcione en Windows.

### **Tareas de distribución**

- Conseguir traductores para el castellano o en caso contrario de momento traducirlo yo mismo (la lengua original del programa es el inglés).
- Conseguir traductores de la web al francés y alemán.
- Mantener un sistema de seguimiento de errores y petición de nuevas funcionalidades usando el sistema tareas de software-libre.org.
- Anunciar ChronoJump en las páginas web de software libre, así como las de deporte y universidad y deporte y entrenamiento.



## Desarrollo del proyecto

A continuación se discute una a una cada una de las tareas planificadas en el inicio del proyecto, así como las nuevas tareas que se han descrito mientras se realizaba el proyecto. En todos los casos, se ampliará el enunciado de la tarea, se discutirá su resolución y se marcará con una de las siguientes palabras clave: *En desarrollo*, *Completa*, o *Pospuesta*.

En el apartado de resolución de cada una de las tareas se incluye un enlace a la(s) tarea(s) correspondiente en el servidor `software-libre.org`<sup>16</sup> en caso de que esta tarea haya sido recogida en dicha plataforma de desarrollo de aplicaciones.

Cabe destacar que algunas tareas, por su previsión de brevedad en la resolución, y debido a que la mayoría debían ser completadas por la misma persona, no eran registradas en los listados de tareas. En este caso simplemente se notificaba su realización en el archivo *changelog*<sup>17</sup>

## Discusión de las tareas planificadas

### Tareas de programación

Tarea	"Mejorar el soporte de Gettext para las descripciones de los estadísticos que se presentan"
Descrip.	Gettext es la herramienta de traducción de aplicaciones GNU por excelencia. Todas las aplicaciones de software libre que desean ser traducidas a varios idiomas optan por Gettext. Más información en la página web de Gettext <a href="http://www.gnu.org/software/gettext/">http://www.gnu.org/software/gettext/</a> Hasta el momento sólo algunas cadenas de texto estaban marcadas para su traducción, pero no integraban correctamente subfrases o variables que se deben añadir a las cadenas.
Resoluc.	Marcar todas las cadenas traducibles como <i>traducibles</i> y las que no deben ser traducidas como <i>no traducibles</i> , tanto en los archivos de código C# como en Glade.  <b>244</b> Gettext y mensajes inglés (Completa) <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=244&amp;group_id=63&amp;group_project_id=75">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=244&amp;group_id=63&amp;group_project_id=75</a>  <b>286</b> Texts better translatable (Completa) <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=286&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=286&amp;group_id=63&amp;group_project_id=93</a>
Estado	Completa

<sup>16</sup><http://software-libre.org>

<sup>17</sup><http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/changelog.txt?cvsroot=chronojump>

Tarea	"Integrar control de carreras, así el programa tendría dos partes: gestión de saltos y gestión de carreras aunque seguiría llamándose ChronoJump"
Descrip.	El tiempo de vuelo de un salto puede ser usado para detectar también el tiempo que se tarda en recorrer una cierta distancia. El punto de partida es la plataforma de contactos y el de llegada es la misma plataforma en un circuito circular o de ida y vuelta. El punto de llegada también puede ser otra plataforma distinta siempre y cuando el deportista no se pueda estar en las dos a la vez.
Resoluc.	<p>El código de saltos simples es reutilizable para carreras simples y el de saltos repetitivos para carreras con intervalos. Así se ha desarrollado el soporte para la ejecución y gestión de todo tipo de carreras, incluso para la creación de nuevos tipos.</p> <p><b>263</b> Treeview run interval (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=263&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=263&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>264</b> Runs execution (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=264&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=264&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>265</b> Runs edit, delete (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=265&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=265&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>282</b> Add run type and run interval type (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=282&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=282&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>283</b> Run interval execute (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=283&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=283&amp;group_id=63&amp;group_project_id=93</a></p>
Estado	Completa

Tarea	"Integrar herramientas de gestión de la base de datos (buscar saltadores, cambiar nombre de sesiones, ...)"
Descrip.	El programa usa una base de datos en lugar de ficheros individuales para cada sesión. Esta característica facilita la gestión de la información pero no permite que los usuarios puedan copiar y pegar información de una sesión a otra. Así que se trata de proveer funciones para que el usuario pueda editar las tres entidades: <i>sesión</i> , <i>sujeto</i> y <i>salto</i> , para satisfacer las necesidades del usuario.
Resoluc.	<p>La tarea se subdivide en las siguientes:</p> <p><b>272</b> Quitar un sujeto de una sesión (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=272&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=272&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>278</b> Session edit, delete (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=278&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=278&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>288</b> Show all events of person (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=288&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=288&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>289</b> Merge two persons (En desarrollo)  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=289&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=289&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>290</b> Merge two jumpTypes and two runTypes (En desarrollo)  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=290&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=290&amp;group_id=63&amp;group_project_id=93</a></p>
Estado	En desarrollo

Tarea	"Preparar la release: tests, limpiar bugs"
Descrip.	Es necesario facilitar releases en tgz de manera que cualquier usuario pueda probar el programa.
Resoluc.	<p>En primer lugar se han corregido numerosos errores, algunos de los cuales han sido descritos en las listas de tareas:</p> <p><b>261</b> repair sorting in indexes (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=261&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=261&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>262</b> Expand jumper automatically in treeviews (1st jump) (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=262&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=262&amp;group_id=63&amp;group_project_id=93</a></p> <p>aunque la mayoría de <i>bugs</i> han sido modificados directamente y se puede seguir su evolución leyendo el archivo changelog <a href="http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/changelog.txt?cvsroot=chronojump">http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/changelog.txt?cvsroot=chronojump</a></p> <p>Se han presentado tres releases: 0.1, 0.2 y 0.3, las tres pensadas para funcionar en una instalación completa de Linux con Mono, o en el Mono Live CD, <a href="https://software-libre.org/project/showfiles.php?group_id=63">https://software-libre.org/project/showfiles.php?group_id=63</a>. Las dos últimas gozan de una buena estabilidad. Todas las releases han sido comunicadas a la comunidad, siempre acompañadas de las instrucciones de uso (hasta el momento complejas)</p> <p><a href="http://lists.software-libre.org/pipermail/chronojump-general_es/2005-September/000062.html">http://lists.software-libre.org/pipermail/chronojump-general_es/2005-September/000062.html</a></p> <p>Además, Juan González ha liberado el código del prototipo 1 de Chronopic, tanto el firmware como un programa para testarlo <a href="https://software-libre.org/project/showfiles.php?group_id=63">https://software-libre.org/project/showfiles.php?group_id=63</a>, y se ha actualizado un listado de tareas para Chronopic <a href="https://software-libre.org/pm/task.php?group_id=63&amp;group_project_id=74">https://software-libre.org/pm/task.php?group_id=63&amp;group_project_id=74</a></p>
Estado	Completa

Tarea	"Crear una rama inestable e iniciar en esta la programación del zumbador, estudio de ritmos y tiempo de reacción"
Descrip.	En el hardware, a diferencia del software, los errores salen muy caros puesto que se debe invertir dinero y más tiempo en la corrección de los mismos. Para ello mientras se trabaja en una versión de prototipo de Chronopic, se debe pensar en la versión con zumbador para relacionar los sonidos con los eventos del sujeto.
Resoluc.	<p>Se ha terminado y documentado la tarjeta Chronopic, <a href="http://www.iearobotics.com/personal/juan/proyectos/chronopic/1.0/">http://www.iearobotics.com/personal/juan/proyectos/chronopic/1.0/</a> y el primer prototipo: <a href="http://www.iearobotics.com/personal/juan/proyectos/chronopic/1.0/prototipo1.html">http://www.iearobotics.com/personal/juan/proyectos/chronopic/1.0/prototipo1.html</a>.</p> <p>También se ha previsto la separación de los firmwares <a href="http://software-libre.org/project/showfiles.php?group_id=63">http://software-libre.org/project/showfiles.php?group_id=63</a></p>
Estado	Completa

## Tareas de empaquetado

Tarea	“Estudiar como instalar el archivo de base de datos generado por Sqlite en algún directorio de usuario en los distintos sistemas operativos soportados (actualmente se usa un archivo en el directorio de ejecución del programa, o se genera de nuevo si no existe)”
Descrip.	El programa no puede estar ligado a un directorio de ejecución y debe tener un lugar único donde guardar los datos.
Resoluc.	Con la ayuda de César Tapia de Monolabs <a href="http://www.monolabs.com/">http://www.monolabs.com/</a> y reimplementando código de la aplicación GPL mCatalog <a href="http://www.mcatalog.net/">http://www.mcatalog.net/</a> , se consiguió ubicar la base de datos en el <i>home de usuario</i> y gestionar la creación del mismo si no existe.  <b>279</b> Database in /home (Completa)  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=279&amp;group_id=63&amp;group_project_id=75">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=279&amp;group_id=63&amp;group_project_id=75</a>
Estado	Completa

Tarea	“Conseguir un Debian developer que empaquete ChronoJump (software de gestión de sesiones, saltadores y saltos) y Chronopic (paquete de control del cronometrador). Una vez empaquetados para Debian, pasarían en breve al repositorio Universe en Ubuntu”
Descrip.	Es necesario facilitar releases en el formato DEB y si es posible en RPM para RedHat y Mandriva.
Resoluc.	Para el empaquetamiento se ha estudiado la utilidad Autopackage <a href="http://autopackage.org/">http://autopackage.org/</a> y se ha realizado contactos con los empaquetadores de Debian y de Mandriva: Ricardo Cárdenes y Warly respectivamente, que en breve dispondrán de una versión de Chronojump. Aunque la tarea se enunciaba como “Conseguir”, Ricardo Cárdenes se ha comprometido a crear el paquete inicial pero no a mantenerlo, y en ninguno de los dos casos se ha obtenido pruebas de su trabajo debido a que fueron contactados muy recientemente, así el estado de la tarea se etiqueta como “en proceso”.
Estado	En proceso

Tarea	"Conseguir que Chronojump funcione en Windows"
Descrip.	La mayoría de usuarios de Chronojump serán usuarios de Windows ya que éste es aún el sistema más instalado hoy en día. La plataforma Mono debería facilitar la migración.
Resoluc.	<p>Las versiones de Mono anteriores a la 1.1.8 no tenían soporte para el puerto serie, así se usaba una librería en C escrita para GNU/Linux y no existía alternativa para Windows ya que no se encontraba nadie dispuesto a reprogramar dicha librería para estos sistemas. Más información: [Chronojump-developers] Spam: Plataforma mono y puerto serie en windows <a href="http://lists.software-libre.org/pipermail/chronojump-developers/2005-August/000145.html">http://lists.software-libre.org/pipermail/chronojump-developers/2005-August/000145.html</a></p> <p>Con la versión 1.1.8 en la calle, es posible escribir una implementación única del acceso al puerto serie:</p> <ul style="list-style-type: none"> <li>■ [Chronojump-developers] Mono 1.1.8 y puerto serie <a href="http://lists.software-libre.org/pipermail/chronojump-developers/2005-September/000158.html">http://lists.software-libre.org/pipermail/chronojump-developers/2005-September/000158.html</a></li> <li>■ Mono - HowToSystemIOPorts <a href="http://www.mono-project.com/HowToSystemIOPorts">http://www.mono-project.com/HowToSystemIOPorts</a></li> </ul> <p>aunque esta no funcionará en Windows hasta la aparición de Mono 2.0 a finales de primavera de 2006.</p>
Estado	En proceso

## Tareas de distribución

Tarea	"Conseguir traductores para el castellano o en caso contrario de momento traducirlo yo mismo (la lengua original del programa es el inglés)"
Descrip.	<p>Hasta el momento, el máximo número de interesados por el proyecto usan el español como primera lengua, prueba de ello es el éxito de la lista de correo <code>chronojump_general_es</code>: <a href="http://lists.software-libre.org/pipermail/chronojump-general_es/">http://lists.software-libre.org/pipermail/chronojump-general_es/</a> respecto a su versión en inglés <a href="http://lists.software-libre.org/pipermail/chronojump-general_en/">http://lists.software-libre.org/pipermail/chronojump-general_en/</a></p> <p>Esto es debido a que el portal <a href="http://www.deporteyciencia.com">http://www.deporteyciencia.com</a> y los antiguos proyectos <i>Salta</i> y <i>Gsalta</i> han mantenido a la espera de Chronojump a un conjunto de profesionales del mundo de la motricidad y de habla española.</p> <p>Se requiere de una localización de Chronojump a este idioma.</p>
Resoluc.	<p>Finalmente la traducción la realizó Xavier de Blas</p> <p><b>287</b> Spanish translation (Completa)</p> <p><a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=287&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=287&amp;group_id=63&amp;group_project_id=93</a></p>
Estado	Completa

Tarea	"Conseguir traductores de la web al francés y alemán"
Descrip.	La web debe estar disponible en el máximo número de idiomas posible, teniendo ya traductores al español, inglés y portugués, las lenguas francesa y alemana son consideradas de alta repercusión. La traducción a numerosos idiomas se considera un valor para conseguir que más personas accedan al proyecto, así como para que éste se muestre a primera vista como más valioso.
Resoluc.	Se ha contactado con Eloy Cárdenas y Deborah Garzón (Mandriva), que se encargan de las traducciones al alemán y al francés. La tarea se considera completa debido a que ha sido enunciada como "conseguir" y los traductores ya se han comprometido en su trabajo y han dado muestra del mismo:  <b>Alemán</b> <a href="http://chronojump.software-libre.org/index_de.html">http://chronojump.software-libre.org/index_de.html</a>  <b>Francés</b> <a href="http://chronojump.software-libre.org/index_fr.html">http://chronojump.software-libre.org/index_fr.html</a>  Además Elisabet Bermejo ha realizado la traducción al catalán.  <b>Catalán</b> <a href="http://chronojump.software-libre.org/index_ca.html">http://chronojump.software-libre.org/index_ca.html</a>
Estado	Completa

Tarea	"Mantener un sistema de seguimiento de errores y petición de nuevas funcionalidades usando el sistema tareas de software-libre.org"
Descrip.	Un sistema de <i>Bugtracking</i> permite administrar las peticiones de nuevas tareas y las resoluciones de los errores advertidos. Hasta el momento se usa un simple fichero de texto TODO.txt
Resoluc.	La plataforma de Software-libre.org incluye un sistema de tareas que aunque no funciona aún a la perfección, es fácil de usar y ha sido usado extensivamente en el desarrollo del proyecto.  <b>Tareas generales</b> <a href="https://software-libre.org/pm/?group_id=63">https://software-libre.org/pm/?group_id=63</a>  <b>Subtareas bugs_and_wishlists</b> <a href="https://software-libre.org/pm/task.php?group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?group_id=63&amp;group_project_id=93</a>
Estado	Completa

Tarea	"Anunciar ChronoJump en las páginas web de software libre, así como las de deporte y universidad y deporte y entrenamiento"
Descrip.	Desde el momento en que aparece una versión usable e instalable con facilidad, cuantas más personas puedan acceder a él, mejor.
Resoluc.	<p>Debido a que la instalación en Windows todavía no ha sido resuelta, no se ha anunciado Chrono-jump en las webs específicas. Se dispone de un fichero de personas interesadas en el proyecto que ya han sido contactadas, se propone la creación de un nuevo listado de páginas web donde hacer difusión del mismo.</p> <p><b>293</b> Crear listado de webs donde anunciar chronojump (Pospuesta)</p> <p><a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=293&amp;group_id=63&amp;group_project_id=76">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=293&amp;group_id=63&amp;group_project_id=76</a></p>
Estado	Pospuesta



## Otras tareas realizadas

En el proyecto Chronojump el usuario está presente en el desarrollo, probando las diferentes muestras que se le ofrecen e interviniendo como consejero y “tester” de usabilidad y errores. Así Chronojump adopta la metodología conocida como Extreme Programming, lejos del diseño tradicional en que el usuario está tan alejado de la creación y es denominado “usuario final”. Esta característica provoca que a menudo surjan nuevos requerimientos que no se habían planificado en un principio, estos requisitos deben ser convenientemente catalogados por orden de prioridad y resueltos según la disponibilidad de los que trabajan en el proyecto.

Las tareas que se citan a continuación son las que no se habían advertido en un inicio, pero que se ha considerado necesario completar. Todas ellas se encuentran ya **completadas**.

## Validaciones

Tarea	“Realizar validaciones de Chronojump”
Descrip.	Se debe saber a ciencia cierta que Chronojump mide lo que debe medir y no cualquier otra cosa
Resoluc.	<p>Se han realizado distintas validaciones de Chronojump, todas ellas con la finalidad de identificar si el software extrae correctamente los datos de la plataforma de contactos o si no lo hace. En todos los casos los resultados han sido satisfactorios:</p> <ul style="list-style-type: none"><li>■ Validación realizada por Juan González <a href="http://lists.software-libre.org/pipermail/chronojump-developers/2005-February/000024.html">Nhttp://lists.software-libre.org/pipermail/chronojump-developers/2005-February/000024.html</a></li><li>■ Validación realizada por Josep Ma. Padullés y Xavier de Blas <a href="http://lists.software-libre.org/pipermail/chronojump-developers/2005-July/000137.html">http://lists.software-libre.org/pipermail/chronojump-developers/2005-July/000137.html</a></li></ul>

## Presentaciones

Tarea	“Presentaciones en Congresos y eventos”
Descrip.	Se debe hacer conocer Chronojump en el mayor número de eventos posible. Tanto eventos técnicos, como de divulgación, en nuestro país y fuera.
Resoluc.	<p>El 7 de Julio se presentaron Chronopic y Chronojump en el I Congreso de Tecnologías del Software Libre, A Coruña. La primera presentación fue a cargo de Juan González, mientras que la segunda fue realizada por Xavier de Blas.</p> <p><b>Artículo en el Weblog del Congreso</b> <a href="http://congreso.gpul.org/weblog/index.php?op=historia&amp;sid=33">http://congreso.gpul.org/weblog/index.php?op=historia&amp;sid=33</a></p> <p><b>Diapositivas</b> <a href="http://chronojump.software-libre.org/presentations/congreso_gpul/index.html">http://chronojump.software-libre.org/presentations/congreso_gpul/index.html</a></p> <p><b>Documento publicado</b> <a href="http://chronojump.software-libre.org/articles/chronojump_sistema_de_medida_congreso_gpul.pdf">http://chronojump.software-libre.org/articles/chronojump_sistema_de_medida_congreso_gpul.pdf</a></p> <p>Además se realizó una presentación de Chronojump en el foro de Software Libre de la Campus Party de Valencia (26 de julio) en que se usaron las diapositivas de la presentación de Galicia y se dispuso de más tiempo para realizar demostraciones. Finalmente se presentó también aunque de forma informal en la Quijote Party de Albacete (10 de julio).</p>

Tarea	"Presentación en televisión"
Descrip.	Como es sabido, la televisión es junto con la radio un medio al que -si se presenta la oportunidad- se debe acceder para presentar cualquier proyecto que se desee divulgar.
Resoluc.	<p>El programa "Contraseñas" dirigido por Adolfo Plasencia y que se emite en la UP TV (televisión universitaria no lucrativa) incluyó un reportaje de unos 20 min sobre Chronojump en un especial sobre la Campus Party.</p> 

## Programación

- Ejecución de saltos y carreras

Tarea	"Creación de un pequeño programa sin entorno gráfico para la captación de saltos"
Descrip.	Con la finalidad de realizar pruebas y también para tener un entorno con el que portar rápidamente Chronopic a cualquier plataforma informática, se necesita de un sencillo programa que simplemente muestre los cambios de evento en la plataforma.
Resoluc.	A partir del programa test-saltos-mono que permitía obtener los tiempos de vuelo, se realizaron breves adaptaciones para que mostrase también los tiempos de contacto y se denominó chronojump-mini.

Tarea	“Mejorar la interacción del usuario con los saltos que no disponen de botón”
Descrip.	Se requiere un sistema rápido para permitir que diversos sujetos distintos puedan realizar un mismo salto con el mínimo de pulsaciones por parte del que usa el programa. Esto es especialmente indicado para los tipos de saltos para los que hay que acceder a una ventana de selección.
Resoluc.	<p>Se implementó el botón “Last jump” para que no sea necesario seleccionar nuevamente el salto a ejecutar. Así cambiando el sujeto y haciendo clic en “Last jump” se consigue lo deseado.</p> <p><b>257</b> change jumper in more jumps or allow repeat last jump type</p> <p><a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=257&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=257&amp;group_id=63&amp;group_project_id=93</a></p>

Tarea	“Permitir saltos repetitivos y carreras con intervalos de duración indeterminada, y permitir finalizarlos en cualquier momento”
Descrip.	No se había previsto que un sujeto podía fatigarse o que un entrenador podía querer detener el test en un determinado momento. Era necesario facilitar la opción de <i>parar un test</i> en cualquier momento y de crear tipos de saltos repetitivos y carreras con intervalos que no tuvieran una limitación de tiempos o saltos.
Resoluc.	<p>El cambio afectó a diversas partes del programa: interfaz gráfica, hilos de ejecución de los eventos y base de datos.</p> <p><b>266</b> Stop Rj Jump</p> <p><a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=266&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=266&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>281</b> Run interval edit, delete, more, last, cancel, finish</p> <p><a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=281&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=281&amp;group_id=63&amp;group_project_id=93</a></p>

- Visualización de datos y estadísticas

Tarea	"Posibilitar la visualización de la altura del salto en todas las partes del programa"
Descrip.	Los entrenadores prefieren obtener la altura de vuelo antes que el tiempo de vuelo. El tiempo de vuelo les es significativo cuando desean compararlo con el tiempo de contacto, pero no les da por sí sólo una valoración de la ejecución. Petición realizada por Enrique Ortega en este correo: <a href="http://lists.software-libre.org/pipermail/chronojump-developers/2005-June/000129.html">http://lists.software-libre.org/pipermail/chronojump-developers/2005-June/000129.html</a>
Resoluc.	Se muestra tiempo de vuelo y altura de vuelo siempre que sea posible, además es posible seleccionar un valor u otro mediante las <i>preferencias</i> para donde sólo sea posible mostrar un valor.  <b>258</b> Graphs and stats with TV or Height <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=258&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=258&amp;group_id=63&amp;group_project_id=93</a>


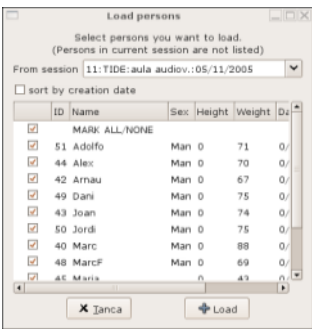
Tarea	"Añadir Índice Q"
Descrip.	Se conoce como "Índice Q" a la relación entre tiempo de vuelo y tiempo de contacto en un salto con caída previa. $IndiceQ = \frac{tv}{tc}$ Los sujetos con más potencia generarán la fuera en menor tiempo, por lo que presentarán un valor de índice Q más elevado. Este estadístico había sido solicitado por varios usuarios.
Resoluc.	El índice se resolvió reusando con los métodos que resuelven el DjIndex: $DjIndex = \frac{tv-tc}{tc} * 100$ <b>260</b> Index Q=TV/TC <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=260&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=260&amp;group_id=63&amp;group_project_id=93</a>

Tarea	"Añadir Índice F/V"
Descrip.	El Índice F/V o Índice de Bosco se caracteriza por relacionar un Squat Jump con un 100 % de carga adicional (respecto a peso del individuo), respecto a otro SJ sin carga adicional. $indiceF/V = \frac{SJ+(100\%)}{SJ} * 100$ Este estadístico había sido solicitado por varios usuarios.
Resoluc.	<b>273</b> Index F/V : h(100 %)/h(0 %) * 100 <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=273&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=273&amp;group_id=63&amp;group_project_id=93</a>

Tarea	"Estadísticos y sus gráficos son sensibles de aplicarse a distintos tipos de saltos"
Descrip.	Los estadísticos y sus gráficos deben poder aplicarse en diferentes tipos de saltos. Así si un usuario crea un nuevo tipo de salto cuyas características son similares al Drop Jump, este salto deberá poder obtener los mismos estadísticos que el DJ (hasta la fecha DjIndex e índiceQ).
Resoluc.	Resuelto con cambios en la ventana de estadísticas, en sus clases y la clase de estadísticas en la base de datos.

- Gestión de datos

Tarea	"Mejorar la exportación de datos"
Descrip.	Todos los datos de los saltos y carreras que se muestran en los <i>treeviews</i> deben poder exportarse al formato CSV.
Resoluc.	<p>Era necesario añadir la altura de vuelo en centímetros y la velocidad de despegue.</p> <p><b>259</b> CSV export with initial speed and height  <a href="https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=259&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailltask&amp;project_task_id=259&amp;group_id=63&amp;group_project_id=93</a></p>

Tarea	"Agilizar la creación y carga de múltiples sujetos de otra sesión"
Descrip.	<p>El usuario <i>Joraike</i>, al probar la release 0.1 había advertido que el proceso de crear varios usuarios o recuperar varios a partir de otra sesión, debería ser más rápido.</p> <p><a href="http://lists.software-libre.org/pipermail/chronojump-developers/2005-July/000135.html">http://lists.software-libre.org/pipermail/chronojump-developers/2005-July/000135.html</a></p>
Resoluc.	<p>Se construyó un widget específico para cada una de las tareas, en ambos casos se realizan las comprobaciones necesarias</p> <p><b>270</b> Creación múltiple de sujetos  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=270&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=270&amp;group_id=63&amp;group_project_id=93</a></p> <p><b>271</b> Recuperar múltiples sujetos de otra sesión  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=271&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=271&amp;group_id=63&amp;group_project_id=93</a></p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>Crear múltiples sujetos</p>  </div> <div style="text-align: center;"> <p>Cargar múltiples sujetos a partir de otra sesión</p>  </div> </div>

■ Otras

Tarea	"Mejorar la interfaz gráfica"
Descrip.	El usuario debe encontrar los botones en su sitio y la separación entre contenidos debe ser agradable
Resoluc.	<p>Aunque no se ha usado explícitamente el HIG de GNOME <a href="http://developer.gnome.org/projects/gup/hig/">http://developer.gnome.org/projects/gup/hig/</a>, se ha tratado de dar una apariencia a los widgets que ha repercutido sin duda en su usabilidad.</p> <p><b>277</b> Interfaces gráfica más fácil para el usuario final  <a href="https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=277&amp;group_id=63&amp;group_project_id=93">https://software-libre.org/pm/task.php?func=detailtask&amp;project_task_id=277&amp;group_id=63&amp;group_project_id=93</a></p>

## Entorno web

Tarea	"Crear o usar un gestor de contenidos web"
Descrip.	<p>Las distintas informaciones que muestra el sitio web de Chronojump, el requisito de no usar <i>frames</i> y la cantidad de idiomas en que se encuentra traducido, dificultan enormemente los cambios.</p> <p>Si aparece un nuevo colaborador habría que cambiar más de 30 páginas web, y si una página está traducida a unos idiomas sí y a otros no, habría que cambiar todas ellas para arreglar sus enlaces.</p>
Resoluc.	<p>Mientras se esperaba a que las traducciones al catalán, alemán y francés llegasen, se escribió un <i>script</i> GPL en PERL que a partir de unos directorios de información estructurada, genera todas las páginas web. Las páginas resultado se suben para ser mostradas al usuario, y las páginas de creación y el script se encuentran en el CVS de Chronojump.</p> <p><b>Programa create_web_chronojump.pl</b> <a href="http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/web/create_web_chronojump.pl?cvsroot=chronojump">http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/web/create_web_chronojump.pl?cvsroot=chronojump</a></p> <p><b>Archivos de configuración</b> <a href="http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/web/data/?cvsroot=chronojump">http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/web/data/?cvsroot=chronojump</a></p> <p><b>Ejemplo castellano</b> <a href="http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/web/data/langs/_es/?cvsroot=chronojump">http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/web/data/langs/_es/?cvsroot=chronojump</a></p>

Tarea	"Búsqueda de traductores del programa"
Descrip.	Para que el programa llegue al mayor número de usuarios posible, y dado que <i>Gettext</i> ya está perfectamente integrado en el mismo, se busca traductores o organizaciones de traductores especializados que puedan llevar a cabo la labor de traducción.
Resoluc.	<p>Se inicia contacto con los administradores de cuentas de GNOME para que faciliten una cuenta de CVS para Chronojump, así, una vez sea migrado de software-libre.org hacia gnome.org, se solicitarán los servicios del grupo internacional de traductores de GNOME.</p> <p><a href="http://developer.gnome.org/projects/gtp/">http://developer.gnome.org/projects/gtp/</a></p> <p>El problema que hay que salvar es la poca relación de Chronojump con las aplicaciones de escritorio, obstáculo que podría frenar su migración y traducción.</p>



## Conclusiones

Aunque la mayoría de las tareas que se habían descrito han sido terminadas, y muchas otras se han propuesto y también finalizado, no se han alcanzado todos los objetivos propuestos inicialmente. La principal limitación ha sido el estado actual de la plataforma Mono, que aún no ofrece soporte para puerto serie.

### En cuanto al primer objetivo:

1.- Liberar una versión estable de ChronoJump en los siguientes formatos:

- Empaquetada en formatos DEB y tar.gz
- Compilada para plataformas Windows
- Empaquetada para Mono Live CD: <http://www.mono-live.com/>

se han alcanzado tres releases o versiones en tar.gz, las tres funcionaban correctamente en el Mono Live CD, así como en cualquier instalación de Linux con Mono igual o superior a 1.1.6. Sin embargo, el contacto tardío con los desarrolladores de Debian y Mandriva y el estado actual de Mono, no han permitido la consecución de las otras dos sub tareas. Por otro lado, aunque se hubiera contactado antes con dichos desarrolladores, existía el problema de empaquetar la librería de acceso a Chronopic cuando ésta iba a migrar de inmediato a Mono, sin duda, era mejor esperar.

### En cuanto al segundo objetivo:

2.- Crear una rama estable y una inestable en el CVS para implementar las nuevas características en la inestable

cabe señalar que en cuanto al software Chronojump no era necesario esta subdivisión ya que no existía información sobre cómo fabricar el cronómetro hasta pocos días antes de la finalización de este proyecto, y por tanto el número de usuarios era demasiado bajo como para preocuparse de que hubiera dos versiones. En referencia a Chronopic, se desestima la creación de una rama inestable hasta que no se finalice el primer prototipo.

### En cuanto al tercer objetivo:

3.- Crear y mantener un sistema de petición de nuevas características y seguimiento de bugs.

el sistema fue creado y ha sido usado extensivamente.

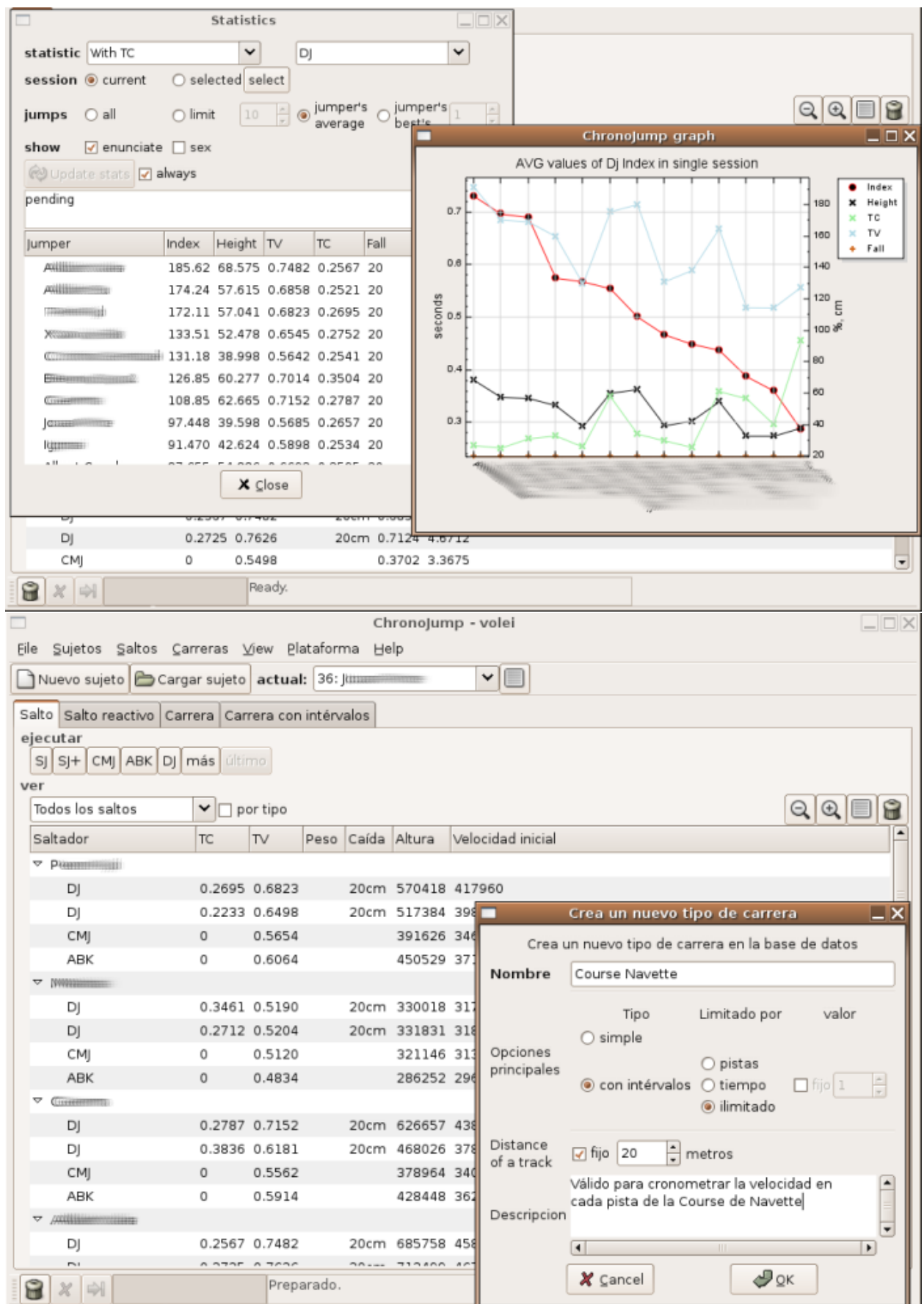
**En cuanto a las tareas,** cabe decir que éstas se generaron inicialmente como claves para conseguir los objetivos citados, pero a posteriori se ha visto como la relación no era tan estrecha como se había previsto. Aunque dos de los tres objetivos del proyecto no se han alcanzado, esto sólo denota una mala definición de estos o una falta de redefinición de los mismos a lo largo del proyecto, ya que Chronojump ha avanzado enormemente durante la duración del proyecto, y eso ha sido debido a las tareas que se habían descrito inicialmente, a las que aparecieron durante el desarrollo del proyecto, así como a otros cambios que sólo quedaron registrados en el archivo *changelog*<sup>18</sup>

En fecha de entrega del proyecto (2 setiembre 2005), Chronojump es un pequeño laboratorio de estudio cinemático del movimiento de gran potencia, validado, documentado en diversos idiomas y libre tanto a nivel de Hardware como de Software.

Por último cabe decir que la programación de Chronojump se inició hace más de 12 meses, mucho antes de que se iniciasen la asignatura de proyectos.

---

<sup>18</sup><http://cvs.software-libre.org/cgi-bin/cvsweb.cgi/chronojump/changelog.txt?cvsroot=chronojump>



Capturas de pantalla v.0.3 (1 de setiembre 2005).

## Perspectivas de futuro

Es inevitable y aconsejable pensar en el impacto que puede causar Chronojump en breve. Aunque sólo se trate de previsiones, al finalizar un proyecto es buen momento para plantear la siguiente evolución:

- A finales de octubre debería estar lista la implementación del acceso a Chronopic por el puerto serie, así el programa avanzaría hacia su uso multiplataforma. Esta implementación usará Mono 1.1.8, aunque sólo funcionará en sistemas GNU/Linux, ya que la migración del código de puerto serie de Mono hacia Windows se espera a finales de primavera de 2006, coincidiendo con Mono 2.0.
- En el 2006 el programa tendrá numerosas funcionalidades y será estable, la programación del prototipo de Chronopic con zumbador para la evaluación de ritmos se habrá iniciado, y a mitades de año se podrá adquirir con un acabado profesional por bajo precio.
- Se entrará en contacto con páginas webs y se escribirán artículos en revistas especializadas, también se presentará la evolución de este método en congresos de ciencias de la actividad física y el deporte.
- Entrenadores de diversas partes del mundo comenzarán a usar el programa a mediados del 2006, también lo harán científicos, fisioterapeutas.
- Se realizarán *talleres Chronojump*<sup>19</sup> en diversas universidades, en algunos participaré yo mismo y en otros ni siquiera tendré constancia de ello.
- En mis clases universitarias de “Biomecánica de la Actividad Física y el Deporte”, y “Tratamiento de la Información Digital en el Deporte”, los alumnos participarán directamente en el proceso de elaboración de sus instrumentaciones y dispondrán de metodologías profesionales para la evaluación del gesto deportivo sin coste alguno. Se desea que muchos otros profesores de ciencias del deporte, fisioterapia y otras titulaciones cercanas hagan lo propio y colaboren en la mejora del proyecto.

---

<sup>19</sup><http://lists.software-libre.org/pipermail/chronojump-developers/2005-May/000092.html>







**Universitat Ramon Llull**

Esta Tesis Doctoral ha sido defendida el día \_\_\_\_ de \_\_\_\_\_ de 20\_\_ en el

Centro \_\_\_\_\_

de la Universitat Ramon Llull

delante del Tribunal formado por los Doctores abajo firmantes, habiendo obtenido la calificación:

Presidente/a

\_\_\_\_\_

Vocal

\_\_\_\_\_

Vocal

\_\_\_\_\_

Vocal

\_\_\_\_\_

Secretario/aria

\_\_\_\_\_

Doctorando/a

\_\_\_\_\_



Facultat de Psicologia, Ciències  
de l'Educació i de l'Esport **Blanquerna**

**Universitat Ramon Llull**

# 2012

**Grup SAFE**

Salut, Activitat Física i Esports

