ESCUELA DE INGENIERÍA DE FUENLABRADA

INGENIERÍA EN SISTEMAS AUDIOVISUALES Y MULTIMEDIA

TRABAJO FIN DE GRADO

DESARROLLO DE UN JUEGO WEB DE SIMULACIÓN DE TECHNICAL DEBT

Autor : Javier Gabari Úriz

Tutor : Gregorio Robles Martínez

Curso académico 2023/2024

Trabajo Fin de Grado

Desarrollo de un Juego Web de Simulación de Technical Debt

Autor : Javier Gabari Úriz

Tutor : Gregorio Robles Martínez

La defensa del presente Proyecto Fin de Carrera se realizó el día de de 2023, siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Fuenlabrada, a de de 2023

Dedicado a

mi familia / mi abuelo / mi abuela

II

Agradecimientos

Aquí vienen los agradecimientos... Aunque está bien acordarse de la pareja, no hay que olvidarse de dar las gracias a tu madre, que aunque a veces no lo parezca disfrutará tanto de tus logros como tú... Además, la pareja quizás no sea para siempre, pero tu madre sí.

Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.

Summary

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.

Índice general

Índice de figuras

Capítulo 1

Introducción

En este capítulo se introduce el proyecto. Debería tener información general sobre el mismo, dando la información sobre el contexto en el que se ha desarrollado.

No te olvides de echarle un ojo a la página con los cinco errores de escritura más frecuentes[1].

Aconsejo a todo el mundo que mire y se inspire en memorias pasadas. Las memorias de los proyectos que he llevado yo están (casi) todas almacenadas en mi web del GSyC[2].

En mayo de 2023 me apunté a un curso de innovación docente donde nos pidieron hacer un podcast con temática docente. Aproveché entonces para hacer un podcast de unos 30 minutos donde en los primeros quince minutos introducía LaTeX y la memoria, y en los segundos hacía hincapién en aquellas cosas que más os cuestan utilizar en la memoria: las figuras, las tablas y las citas. Podéis escuchar el podcast en Internet[3].

1.1. Sección

Esto es una sección, que es una estructura menor que un capítulo.

Por cierto, a veces me comentáis que no os compila por las tildes. Eso es un problema de codificación. Al guardar el archivo, guardad la codificación de "ISO-Latin-1" a "UTF-8" (o viceversa) y funcionará.

[1] `http://www.tallerdeescritores.com/errores-de-escritura-frecuentes`

[2] `https://gsyc.urjc.es/~grex/pfcs/`

[3] `https://podcasters.spotify.com/pod/show/gregorio-robles9/episodes/Tu-memoria-de-Trabajo-Fin-de-Grado-o-de-Mster-en-LaTeX-e23hucr/a-a58kp2`

1.1.1. Estilo

Recomiendo leer los consejos prácticos sobre escribir documentos científicos en LaTeX de Diomidis Spinellis[4].

Lee sobre el uso de las comas[5]. Las comas en español no se ponen al tuntún. Y nunca, nunca entre el sujeto y el predicado (p.ej. en "Yo, hago el TFG" sobre la coma). La coma no debe separar el sujeto del predicado en una oración, pues se cortaría la secuencia natural del discurso. No se considera apropiado el uso de la llamada coma respiratoria o *coma criminal*. Solamente se suele escribir una coma para marcar el lugar que queda cuando omitimos el verbo de una oración, pero es un caso que se da de manera muy infrecuente al escribir un texto científico (p.ej. "El Real Madrid, campeón de Europa").

A continuación, viene una figura, la Figura 1.1. Observarás que el texto dentro de la referencia es el identificador de la figura (que se corresponden con el "label" dentro de la misma). También habrás tomado nota de cómo se ponen las "comillas dobles" para que se muestren correctamente. Nota que hay unas comillas de inicio (") y otras de cierre ("), y que son diferentes. Volviendo a las referencias, nota que al compilar, la primera vez se crea un diccionario con las referencias, y en la segunda compilación se "rellenan" estas referencias. Por eso hay que compilar dos veces tu memoria. Si no, no se crearán las referencias.

A continuación un bloque "verbatim", que se utiliza para mostrar texto tal cual. Se puede utilizar para ofrecer el contenido de correos electrónicos, código, entre otras cosas.

```
From gaurav at gold-solutions.co.uk  Fri Jan 14 14:51:11 2005
From: gaurav at gold-solutions.co.uk (gaurav_gold)
Date: Fri Jan 14 19:25:51 2005
Subject: [Mailman-Users] mailman issues
Message-ID: <003c01c4fa40$1d99b4c0$94592252@gaurav7klgnyif>

Dear Sir/Madam,
How can people reply to the mailing list?  How do i turn off
this feature? How can i also enable a feature where if someone
replies the newsletter the email gets deleted?
Thanks
```

[4] https://github.com/dspinellis/latex-advice

[5] http://narrativabreve.com/2015/02/opiniones-de-un-corrector-de-estilo-11-recetas-par
html

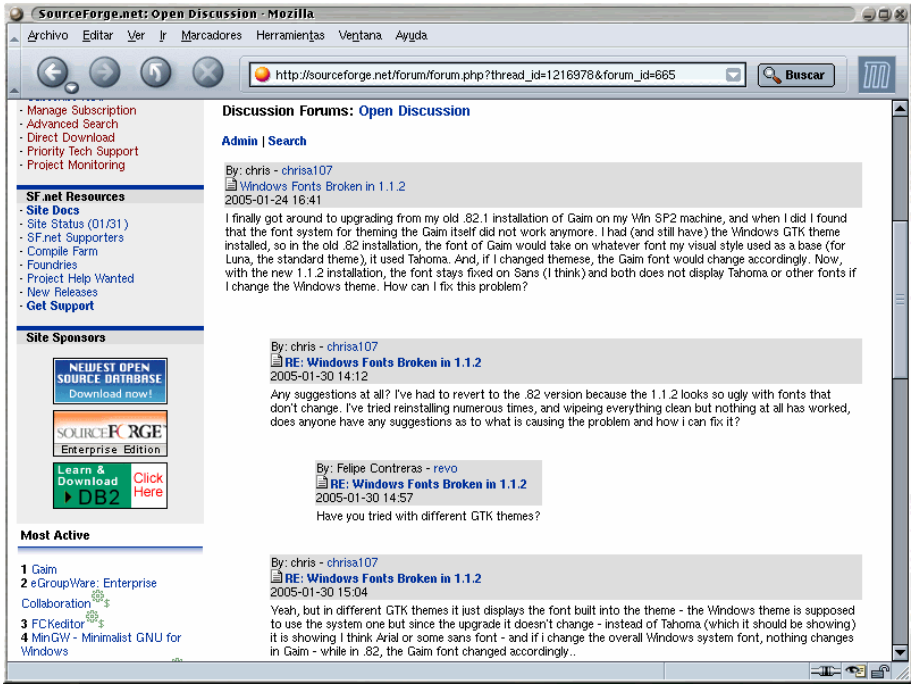

Figura 1.1: Página con enlaces a hilos

```
From msapiro at value.net  Fri Jan 14 19:48:51 2005
From: msapiro at value.net (Mark Sapiro)
Date: Fri Jan 14 19:49:04 2005
Subject: [Mailman-Users] mailman issues
In-Reply-To: <003c01c4fa40$1d99b4c0$94592252@gaurav7klgnyif>
Message-ID: <PC17302005011410485105780lb04d55@msapiro>

gaurav_gold wrote:
>How can people reply to the mailing list?  How do i turn off
this feature? How can i also enable a feature where if someone
replies the newsletter the email gets deleted?

See the FAQ
>Mailman FAQ: http://www.python.org/cgi-bin/faqw-mm.py
article 3.11
```

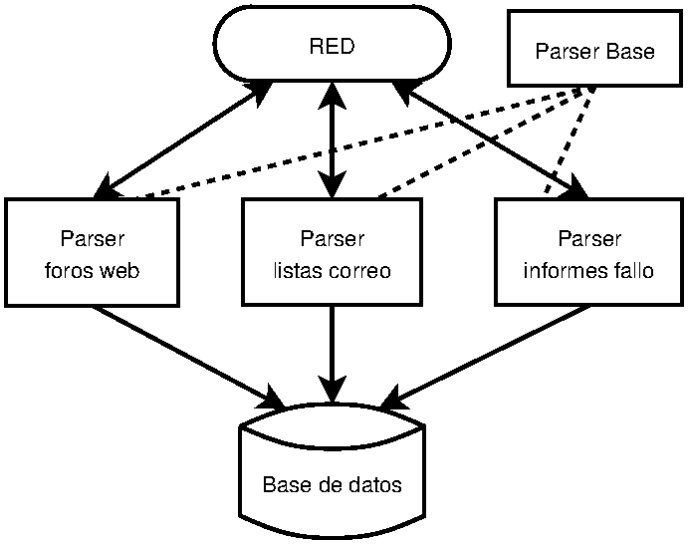

Figura 1.2: Estructura del parser básico

1.2. Estructura de la memoria

En esta sección se debería introducir la estructura de la memoria.

Así:

- En el primer capítulo se hace una intro al proyecto.
- En el capítulo 2 (ojo, otra referencia automática) se muestran los objetivos del proyecto.
- A continuación se presenta el estado del arte en el capítulo 3.
- ...

Capítulo 2

Objetivos

2.1. Objetivo general

Mi trabajo de fin de grado consiste en crear un juego Web que represente el concepto de *Technical Debt* (Deuda Técnica).

2.2. Objetivos específicos

Crear un juego Web de tipo gestor de recursos y relacionarlo con el concepto de *Technical Debt*. Programar el juego en HTML, CSS y JavaScript y añadirle funcionalidades de servidor para que los usuarios puedan registrarse, guardar su progreso, etc.

2.3. Planificación temporal

Inicié el presente trabajo en febrero de 2023 dedicándole pocas horas de febrero a julio principalmente los fines de semana y de manera más intensiva en agosto, unas 6 horas casi todos los días.

Capítulo 3

Estado del arte

Descripción de las tecnologías que utilizas en tu trabajo (HTML, JavaScript, CSS, peticiones HTTP, cookies, JSON, VSCode, GitHub) Con dos o tres párrafos por cada tecnología, vale. Se supone que aquí viene todo lo que no has hecho tú.

- JavaScript
- HTML
- CSS
- peticiones HTTP
- cookies
- JSON
- VSCode
- GitHub

Puedes citar libros, como el de Bonabeau et al., sobre procesos estigmérgicos [1]. Me encantan los procesos estigmérgicos. Deberías leer más sobre ellos. Pero quizás no ahora, que tenemos que terminar la memoria para sacarnos por fin el título. Nota que el ~ añade un espacio en blanco, pero no deja que exista un salto de línea. Imprescindible ponerlo para las citas.

Citar es importantísimo en textos científico-técnicos. Porque no partimos de cero. Es más, partir de cero es de tontos; lo suyo es aprovecharse de lo ya existente para construir encima y hacer cosas más sofisticadas. ¿Dónde puedo encontrar textos científicos que referenciar? Un buen sitio es Google Scholar[1]. Por ejemplo, si buscas por "stigmergy libre software" para encontrar trabajo sobre software libre y el concepto de *estigmergia* (¿te he comentado que me gusta el concepto de estigmergia ya?), encontrarás un artículo que escribí hace tiempo cuyo título es "Self-organized development in libre software: a model based on the stigmergy concept". Si pulsas sobre las comillas dobles (entre la estrella y el "citado por ...", justo debajo del extracto del resumen del artículo, te saldrá una ventana emergente con cómo citar. Abajo a la derecha, aparece un enlace BibTeX. Púlsalo y encontrarás la referencia en formato BibTeX, tal que así:

```
@inproceedings{robles2005self,
  title={Self-organized development in libre software:
         a model based on the stigmergy concept},
  author={Robles, Gregorio and Merelo, Juan Juli\'an
          and Gonz\'alez-Barahona, Jes\'us M.},
  booktitle={ProSim'05},
  year={2005}
}
```

Copia el texto en BibTeX y pégalo en el fichero `memoria.bib`, que es donde están las referencias bibliográficas. Para incluir la referencia en el texto de la memoria, deberás citarlo, como hemos hecho antes con [1], lo que pasa es que en vez de el identificador de la cita anterior (bonabeau:_swarm), tendrás que poner el nuevo (robles2005self). Compila el fichero `memoria.tex` (`pdflatex memoria.tex`), añade la bibliografía (`bibtex memoria.aux`) y vuelve a compilar `memoria.tex` (`pdflatex memoria.tex`)...y *voilà* ¡tenemos una nueva cita [2]!

También existe la posibilidad de poner notas al pie de página, por ejemplo, una para indicarte que visite la página del GSyC[2].

Hemos hablado de cómo incluir figuras. Pero no hemos dicho nada de tablas. A mí me gustan las tablas. Mucho. Aquí un ejemplo de tabla, la Tabla 3.1 (siento ser pesado, pero nota cómo he puesto la referencia).

[1] `http://scholar.google.com`

[2] `http://gsyc.es`

Uno	2	3
Cuatro	5	6
Siete	8	9

Cuadro 3.1: Ejemplo de tabla. Aquí viene una pequeña descripción (el *caption*, el pie de tabla/figura) del contenido de la tabla. Si la tabla no es autoexplicativa, siempre viene bien aclararla aquí.

Hay un sitio en Internet donde puedes diseñar las tablas fácilmente y luego hacer un corta y pega del resultado en tu editor. Puedes probarlo en `https://www.tablesgenerator.com/`.

3.1. HTML

Del inglés, *HyperText Markup Language*, HTML es el lenguaje de marcado por excelencia para la creación de páginas Web. Mediante etiquetas, define la estructura del contenido que se debe mostrar. Para añadir elementos externos basta con referenciarlos para que el navegador los busque y los una para la visualización, mientras que la propia página Web solo contiene texto. Actualmente es el estándar que siguen todos los navegadores.

3.2. JavaScript

JavaScript (habitualmente abreviado como JS) es el lenguaje de programación más usado en Web. Se complementa con HTML gestionando cómo se comporta una página web cuando ocurre un evento. Generalmente se ejecuta en el navegador en el lado cliente, aunque también se puede ejecutar en otros entornos (como Node.js que introduciré en la sección 3.4) para actuar del lado del servidor.

3.3. CSS

Cascading Style Sheets, o en español Hojas de Estilo en Cascada, es el lenguaje de estilos que se encarga de definir estéticamente la página web indicando cómo se debe renderizar cada

elemento estructurado.

3.4. Node.js

Node.js es un entorno de ejecución de JavaScript pensado para cubrir la necesidad de crear servidores altamente escalables. El entorno está orientado a atender eventos asíncronos. Cabe destacar, también, que se trata de un entorno de código abierto desarrollado por OpenJS Foundation.

3.5. JSON

JSON (*JavaScript Object Notation*) es un formato para intercambiar datos basado en la sintaxis de objetos de JavaScript. Representa las estructuras de datos de una manera simple tanto para el humano como para la máquina.

3.6. Protocolo HTTP

Es el encargado de realizar la comunicación entre cliente y servidor mediante mensajes de solicitud y de respuesta. En los mensajes gestionados por HTTP solo viaja texto, lo que hace que sean considerablemente ligeros. Los mensajes de solicitud (del cliente al servidor) utilizan 3 métodos dependiendo de su objetivo:

- GET para solicitar recursos al servidor,
- POST para enviar datos al servidor
- y HEAD para solicitar información que pueda ir directamente en las cabeceras de la respuesta.

En los mensajes de respuesta (del servidor al cliente) se utilizan códigos numéricos para indicar qué ha sucedido con la solicitud y el contenido solicitado va codificado en el cuerpo del mensaje.

3.7. Cookies

Las Cookies HTTP son pequeños fragmentos de información creados por el servidor para ser almacenados en el lado del cliente de manera se le pueda recordar"dicha información al servidor la próxima vez que se conecte el mismo usuario. Se guardan en forma de pares nombre-valor y se envían en las cabeceras de cada petición que hace el cliente al servidor correspondiente.

3.8. Visual Studio Code

Para escribir toda la parte de código de este trabajo he utilizado Visual Studio Code. Es un editor de código creado por Microsoft y de código abierto que es ligero y altamente personalizable. Soporta una alta cantidad de lenguajes de forma nativa y muchos otros mediante extensiones. También cuenta con control integrado de Git, usado en este trabajo para guardar los progresos.

Capítulo 4

Diseño e implementación

En este capítulo se describe la arquitectura del juego y su implementación.

4.1. Arquitectura general

Aunque la parte principal de este trabajo es el juego sobre Technical Debt, la aplicación web cuenta con más partes que complementan y completan su uso. Como se puede ver en la figura 4.1, la estructura básica de la aplicación cuenta con:

- Dos pantallas estáticas:
  - Pantalla de Inicio
  - Pantalla de inicio de sesión
- Dos pantallas dinámicas:
  - Juego sin sesión iniciada
  - Juego con sesión iniciada

Los cambios entre pantallas y las acciones que se pueden realizar en cada una de ellas están gestionadas mediante una arquitectura cliente-servidor.

La pantalla principal es la primera que vería el usuario

Por ejemplo, puedes verlo en la figura 1.2. LaTeX pone las figuras donde mejor cuadran. Y eso quiere decir que quizás no lo haga donde lo hemos puesto... Eso no es malo. A veces

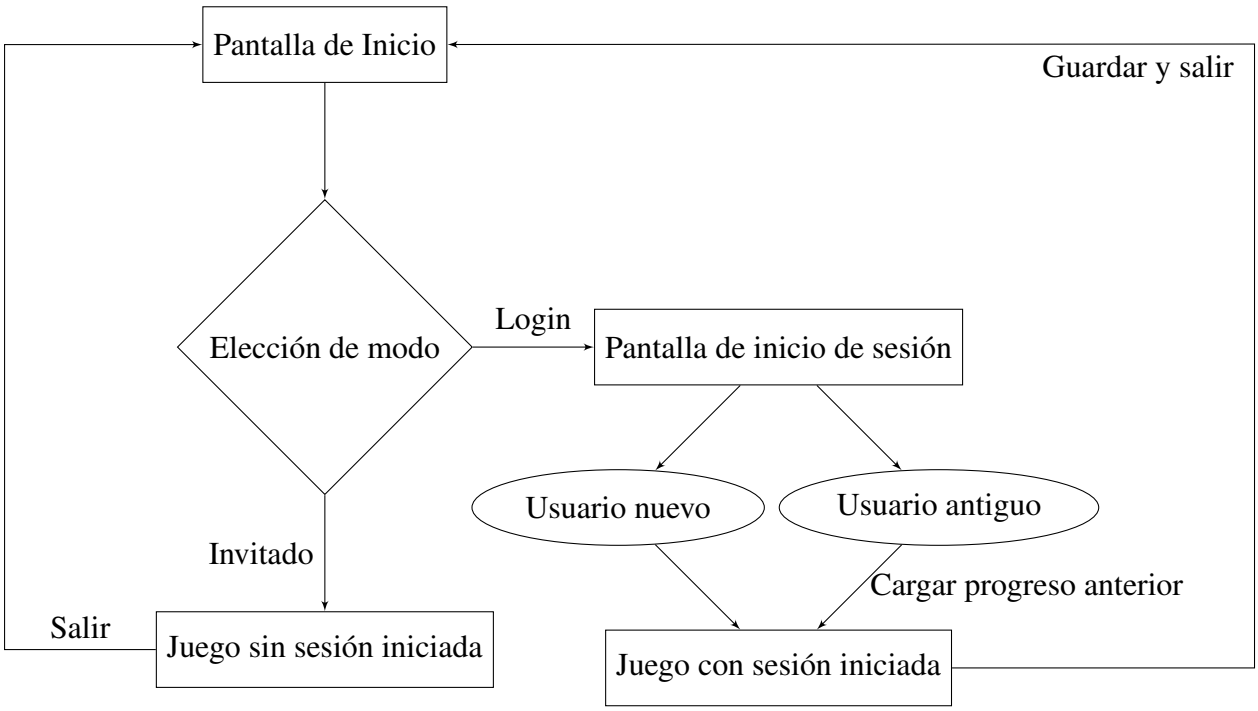

Figura 4.1: Estructura del software

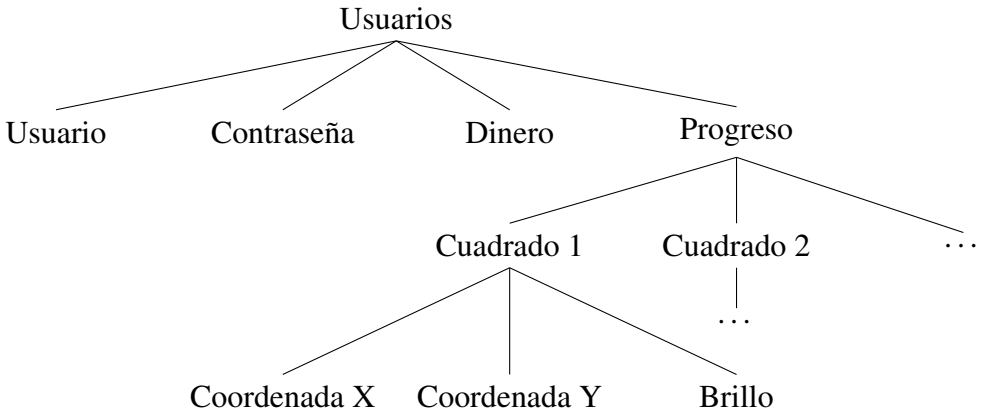

Figura 4.2: Estructura de los datos de los usuarios

queda un poco raro, pero es la filosofía de LaTeX: tú al contenido, que yo me encargo de la maquetación.

Recuerda que toda figura que añadas a tu memoria debe ser explicada. Sí, aunque te parezca evidente lo que se ve en la figura 1.2, la figura en sí solamente es un apoyo a tu texto. Así que explica lo que se ve en la figura, haciendo referencia a la misma tal y como ves aquí. Por ejemplo: En la figura 1.2 se puede ver que la estructura del *parser* básico, que consta de seis componentes diferentes: los datos se obtienen de la red, y según el tipo de dato, se pasará a un *parser* específico y bla, bla, bla...

Si utilizas una base de datos, no te olvides de incluir también un diagrama de entidad-relación.

Capítulo 5

Experimentos y validación

Este capítulo se introdujo como requisito en 2019. Describe los experimentos y casos de test que tuviste que implementar para validar tus resultados. Incluye también los resultados de validación que permiten afirmar que tus resultados son correctos.

Capítulo 6

Resultados

En este capítulo se incluyen los resultados de tu trabajo fin de grado.

Si es una herramienta de análisis lo que has realizado, aquí puedes poner ejemplos de haberla utilizado para que se vea su utilidad.

Capítulo 7

Conclusiones

7.1. Consecución de objetivos

Esta sección es la sección espejo de las dos primeras del capítulo de objetivos, donde se planteaba el objetivo general y se elaboraban los específicos.

Es aquí donde hay que debatir qué se ha conseguido y qué no. Cuando algo no se ha conseguido, se ha de justificar, en términos de qué problemas se han encontrado y qué medidas se han tomado para mitigar esos problemas.

Y si has llegado hasta aquí, siempre es bueno pasarle el corrector ortográfico, que las erratas quedan fatal en la memoria final. Para eso, en Linux tenemos aspell, que se ejecuta de la siguiente manera desde la línea de *shell*:

```
aspell --lang=es_ES -c memoria.tex
```

7.2. Aplicación de lo aprendido

Aquí viene lo que has aprendido durante el Grado/Máster y que has aplicado en el TFG/TFM. Una buena idea es poner las asignaturas más relacionadas y comentar en un párrafo los conocimientos y habilidades puestos en práctica.

1. a

2. b

7.3. Lecciones aprendidas

Aquí viene lo que has aprendido en el Trabajo Fin de Grado/Máster.

1. Aquí viene uno.
2. Aquí viene otro.

7.4. Trabajos futuros

Ningún proyecto ni software se termina, así que aquí vienen ideas y funcionalidades que estaría bien tener implementadas en el futuro.

Es un apartado que sirve para dar ideas de cara a futuros TFGs/TFMs.

Apéndice A

Manual de usuario

Esto es un apéndice. Si has creado una aplicación, siempre viene bien tener un manual de usuario. Pues ponlo aquí.

Bibliografía

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Articial Systems*. Oxford University Press, Inc., 1999.

[2] G. Robles, J. J. Merelo, and J. M. González-Barahona. Self-organized development in libre software: a model based on the stigmergy concept. In *ProSim'05*, 2005.