Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú
UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROYECTO FIN DE CARRERA

TITULO: Creación y desarrollo de un MMORPG basado en Web.

AUTOR: Raúl Navarro Ruiz

TITULACIÓN: Ingeniería Técnica en Informática de gestión

DIRECTOR: Jordi Daude Ventura

DEPARTAMENTO: Lenguajes y Sistemas informáticos

FECHA: 27 de Enero de 2010

TITULO: Creación y desarrollo de un MMORPG basado en Web.

APELLIDOS: Navarro Ruiz **NOMBRE:** Raúl

TITULACIÓN: Ingeniería Técnica

ESPECIALIDAD: Informática de Gestión **PLAN:** 92

DIRECTOR: Jordi Daude Ventura

DEPARTAMENTO: Lenguajes y Sistemas informáticos

CALIFICACIÓN DEL PFC

TRIBUNAL

PRESIDENTE	SECRETARIO	VOCAL
MARIO MARTÍN MUÑOZ	JULIO FERNANDEZ GONZALEZ	SERGIO SANCHEZ LOPEZ

FECHA DE LECTURA:

Este proyecto tiene en cuenta aspectos medioambientales: X Sí □ No

PROJECTO FIN DE CARRERA

RESUMEN (máximo 50 líneas)

El objetivo de este proyecto es crear un juego multijugador para ser jugado por interfaz web sin necesidad de instalar ningún software adicional.

La idea del juego es que un jugador asume el rol de un personaje que está preso en una cárcel y su objetivo es ir subiendo de nivel, explorando la prisión e ir luchando contra bots, todo esto mientras interacciona con otros jugadores intentado llegar a ser el jugador con más puntos de experiencia.

El juego usa el servidor web Apache y el MySQL como sistema gestor de Bases de Datos.

El desarrollo ha sido con el lenguaje de programación PHP, como lenguaje del lado del servidor, JavaScript como lenguaje del lado del cliente y XHTML junto con CSS para el diseño.

Palabras clave (máximo 10):

PHP	Apache	MySQL	XAMPP
Rol	Bots	Habilidades	Multijugador
MMORPG	CSS		

Índice

Parte I: Proyecto 11
1. Introducción 12
2. Contexto del Proyecto 13
2.1. Objetivos 13
2.2. Justificación 14
3. Evaluación Tecnológica 15
3.1. Lenguaje de Programación 15
3.2. Sistema Gestor de la Base de Datos 16
3.3. Servidor Web 18
3.4. Servidor de correo 18
3.5. Uso del XAMPP como fusión de las necesidades tecnológicas 19
4. Planificación Inicial 20
Parte II: Análisis y Especificación 22
5. Especificación de Requerimientos 23
5.1. Requerimientos funcionales 23
5.2. Requerimientos no funcionales 23
5.2.1. Instalación de la aplicación 23
5.2.2. Uso de la aplicación 24
5.2.3. Otros requerimientos no funcionales 24
5.2.4. Requerimientos del sistema 24
6. Modelo Conceptual 25
6.1. Diagrama de una aplicación Web 26
7. Casos de Uso 27
7.1. Iniciar sesión 28
7.2. Creación de cuenta 29
7.3. Gestión de Usuarios 30
7.4. Creación de un mensaje global 31
7.5. Explorar el mapa 31
7.6. Creación de combates 32
7.7. Configuración del juego 33
7.8. Comprar armas, armaduras 34
7.9. Salir de la aplicación 35
8. Diagramas de secuencia 36
8.1. Creación de cuenta 36
8.2. Creación de combates 37
Parte III: Diseño 40
9. Arquitectura del diseño de la aplicación 41
10. Diagramas de colaboración 43
10.1. Diagrama de colaboración de la creación de combate 43
11. Diseño de la aplicación 44
11.1. Diseño de la interfaz de usuario 44
11.1.1. Pantallas generales 44
11.1.2. Pantalla de ayuda en línea 47
11.1.3. Pantalla del menú de administrador 47
11.2. Diseño de los delitos 48
11.2.1. Habilidades 48
11.2.2. Golpes Especiales 49
11.2.3. Progresión de Niveles 50
11.2.4. Progresión de los puntos de acción 51
11.3. Caracterización de los personajes 52
11.4. Diseño gráfico de las armas 54
11.4.1. Gráfica del daño medio de las armas 55
11.5. Regla de los 9's 56
Parte IV: Implementación y pruebas 58
12. Tecnologías utilizadas 59
12.1. Navegador Web 59

13. Implementación 60
13.1. Estrategia para el desarrollo de la aplicación 60
13.2. Archivos creados y desarrollo de la aplicación 60
13.2.1. Uso de los principales archivos .php 62
13.3. Uso de plantillas 63
13.4. Seguridad de la aplicación 64
13.5. Tablas de la Base de Datos 65
13.5.1. Estructura de la tabla armaduras 66
13.5.2. Estructura de tabla armas 67
13.5.3. Estructura de las tablas de atributos 68
13.5.4. Estructura de tabla de bots 69
13.5.5. Estructura de tabla de las zonas 70
13.5.6. Estructura de tabla de los anuncios 71
13.5.7. Estructura de tabla del chat 71
13.5.8. Estructura de tabla de control 72
13.5.9. Estructura de tabla del foro 74
13.5.10. Estructura de tabla de los jugadores 75
14. Testeo de la aplicación 79
14.1. Test realizados sobre la aplicación 79
14.2. Test sobre los tiempos de respuesta 80
14.2.1. Prueba I 81
14.2.2. Prueba II 84
14.2.3. Conclusiones de las pruebas 86

Parte V: Mantenimiento 88
15. Manual de Usuario 89
15.1. Pantalla inicial de la aplicación 89
15.2. Pantalla de Bienvenida 89
15.3. Cajas de Información y expansión de habilidades 91
15.4. Acceso a las diferentes zonas 93
15.5. Ranking de jugadores 95
15.6. Exploración 95
15.7. Combate 97
15.7.1. Barras de Salud, Experiencia y p.a.'s 100
15.8. Foro 101
16. Manual de Administrador 102
16.1. Requisitos mínimos 102
16.2. Instalación de la aplicación 102
16.2.1. Comprobación de la correcta instalación 104
16.3. Instalación del servidor de correo 104
16.4. Menú de Administrador 105
16.4.1. Configuración de la aplicación 105
16.4.2. Insertar mensajes 106
16.4.3. Editar Jugadores 106

Parte VI: Balances y Conclusiones 109
17. Planificación Final y Análisis económica 110
17.1. Análisis de Costes 110
17.2. Comparativa entre la estimación inicial y la final 112
17.2.1. Conclusiones sobre los datos 113
17.3. Análisis económico 113
18. Análisis Medioambiental 116
19. Análisis de Accesibilidad 117
20. Conclusiones y Mejoras 119
20.1. Conclusiones Personales 119
20.2. Conclusiones Técnicas 120
20.3. Mejoras 121

Bibliografía 122

Anexo I: Consultas 123

Índice de las imágenes

Parte I: Proyecto 11
Imagen 3.1. Gráfica de rendimiento de los diferentes motores de almacenado. [1] 17
Parte II: Análisis y Especificación 22
Imagen 6.1. Modelo Conceptual del proyecto 25
Imagen 6.2. Diagrama de una aplicación Web 26
Imagen 7.1. Diagrama del modelo de casos de uso 27
Parte III: Diseño 40
Imagen 9.1. Arquitectura web en 3 capas 41
Imagen 10.1. Diagrama de colaboración para crear un combate 43
Imagen 10.2. Diagrama de colaboración para crear un turno 43
Imagen 10.3. Diagrama de colaboración para finalizar un combate 43
Imagen 11.1. Página inicial de la aplicación, con los diferentes campos de información 44
Imagen 11.2. Captura de un instante del juego para mostrar cómo se divide la información 46
Imagen 11.3. Pantalla de ayuda en línea del juego 47
Imagen 11.4. Vista de la pantalla inicial de administrador 47
Imagen 11.5. Progresión de los puntos de experiencia necesarios para subir de nivel 50
Imagen 11.6. Progresión de los puntos de acción en función del nivel y de los delitos 51
Imagen 11.7. Boceto original de los tres delitos 52
Imagen 11.8. Pintado de los delitos usando Photoshop CS2 53
Imagen 11.9. Esta es la imagen utilizada como fondo para los personajes 53
Imagen 11.10. Diseño final de los delitos 54
Imagen 11.11. Gráfico de dispersión de los valores de daño y puntos de acción en las armas 55
Parte IV: Implementación y pruebas 58
Imagen 12.1. Uso de los diferentes navegadores web en septiembre del 2009[4] 59
Imagen 13.1. Ejemplo de cómo se muestra la información después de remplazar los campos 63
Imagen 13.2. Código del archivo de plantillas objeto.php 64
Imagen 14.1. Imagen que muestra el tiempo que ha tardado en cargar la página 80
Imagen 14.2. Medición de la velocidad de subida y de bajada del PC donde se realizaron las pruebas 81
Imagen 14.3 Tiempo real y tiempo medio de respuesta para cada usuario 82
Imagen 14.4. Tiempo acumulado de retardo global para 25 usuarios 82
Imagen 14.5. Gráfica que muestra los resultados de la prueba y sus variaciones 83
Imagen 14.6. Tiempo real y tiempo medio de respuesta para cada usuario 83
Imagen 14.7. Tiempo acumulado de retardo global para 25 usuarios 84
Imagen 14.8. Medición de la velocidad de subida y de bajada del PC donde se realizaron las pruebas 84
Imagen 14.9. Tiempo real y tiempo medio de respuesta para cada usuario 85
Imagen 14.10. Tiempo acumulado de retardo global 85
Parte V: Mantenimiento 88
Imagen 15.1. Página inicial de la aplicación, con los diferentes campos de información 89
Imagen 15.2. Pantalla de bienvenida una vez entra el usuario al juego por primera vez 90
Imagen 15.3. Imagen que muestra como las habilidades del jugador se expanden y contraen 91
Imagen 15.4. Pantalla de información sobre las habilidades 91
Imagen 15.5. Captura de la ventana de información emergente del juego 92
Imagen 15.6. Pantalla del juego en la zona de celdas 93
Imagen 15.7. Captura de la lista de armas en la zona de celdas 94
Imagen 15.8. Imagen de la compra de medicinas 94
Imagen 15.9. Captura de la pantalla de ranking de jugadores 95
Imagen 15.10. Botones de movimiento 95
Imagen 15.11. Captura del mapa de la prisión con 3 jugadores. 96
Imagen 15.12. Captura de la pantalla de combate. 97
Imagen 15.13. Captura que muestra al jugador los atributos del arma seleccionada 98
Imagen 15.14. Captura que muestra información del turno 1 del jugador. 99
Imagen 15.15. Captura donde el jugador gana el combate al bot. 99
Imagen 15.16. Captura de la pantalla de victoria. 100
Imagen 15.17. Captura de la pantalla de objeto 100

Imagen 15.18. Imagen de los tres colores de las barras de salud y p.a. 100
Imagen 15.19. Captura de la pantalla del foro 101
Imagen 15.20. Aquí podríamos responder un mensaje de un tema 101
Imagen 16.1. Imagen del link para acceder al phpMyAdmin. 102
Imagen 16.2. Captura dentro de la aplicación phpMyAdmin 103
Imagen 16.3. Imagen de la configuración para el correcto funcionamiento del juego 103
Imagen 16.4. Captura de pantalla del panel de Control de la aplicación XAMPP 104
Imagen 16.5. Vista de la pantalla del menú de administrador 105
Imagen 16.6. Vista de la pantalla de la configuración dentro del menú de administrador 105
Imagen 16.7. Vista de la pantalla de inserción de mensajes dentro del menú de administrador 106
Imagen 16.8. Vista de la pantalla de edición de jugadores dentro del menú de administrador 106
Imagen 16.9. Vista de la pantalla de edición de los diferentes jugadores. 107

Parte VI: Balances y Conclusiones 109

Imagen 17.1. Gráfico que indica el porcentaje de tiempo que ha consumido cada recurso. 111
Imagen 17.2. Gráfico que muestra la variación temporal entre el tiempo estimado y el real. 112
Imagen 17.3. Gráfico que muestra el coste por cada tipo de tarea. 115

Nota de Agradecimientos

Han sido muchas semanas de trabajo intenso, pero en la vida hay que ser agradecido con la gente que te ayuda, esto es un pequeño homenaje a esas personas que han estado allí apoyándome continuamente durante este periodo.

- Primero de todo a mis padres, por su infinita paciencia que han tenido conmigo durante toda la carrera.
- A mi novia, Mariona Mondelo, sin su apoyo no sé cuánto tiempo más hubiera necesitado para llegar hoy donde estoy.
- A Emilio López, sobre todo gracias por todas las veces que me has ayudado con todo lo que te he pedido y por estar ahí.
- A Alberto Otero, gracias por esos magníficos dibujos y caracterización de los delitos.
- A Roger Mateu y a Jordi Gálvez, por todos sus consejos y ayudas que me han ofrecido durante la carrera y con este proyecto.
- A Jordi Daude por ser mi tutor y querer dedicar parte de su tiempo a este trabajo.
- A la familia y los amigos que me han dado ánimos.

Parte I: Proyecto

1. Introducción

Este proyecto ha supuesto una de mis mayores satisfacciones personales a la hora de culminarlo y ha sido el de realizar un juego de ordenador. Sin embargo con la culminación del proyecto no significa que deje apartado el mantenimiento y posterior mejora del juego.

Anteriormente había desarrollado un juego basado en Excel sobre la gestión de un restaurante pero no tiene nada que ver ni en tiempo ni en nivel con el que presento a continuación.

Cualquier proyecto de final de carrera es una realidad a la que se tiene que enfrentar cualquier alumno de una ingeniería y es donde, mediante esfuerzo y trabajo ha de demostrar los conocimientos que ha adquirido durante el periodo que ha durado la carrera.

Pues bien, con este proyecto creo que he podido cumplir ambas cosas, el demostrar una serie de conocimientos y el de lograr uno de mis objetivos personales.

2. Contexto del Proyecto

Este proyecto está orientado a cualquier usuario de internet que quiera pasar un buen rato jugando a un juego que ofrezca unas mínimas condiciones de calidad y diversión y que pueda continuar cualquier otro día sabiendo que la progresión que consiguió con su personaje se quedó guardada.

Cualquier usuario que le gusten los juegos de rol y que haya jugado al juego de *La Prisión* creado por Dinamic Multimedia, se sentirá identificado rápidamente.

2.1. Objetivos

Como he dicho anteriormente el principal objetivo del proyecto ha sido el poder cumplir un objetivo personal que era el desarrollo de un juego de ordenador, cuando inicié el proyecto tenía dudas de si sería capaz de desarrollarlo para que tuviera unas condiciones mínimas de jugabilidad, seguridad y diversión.

Es cierto que no fue inmediato que me decantara por este proyecto, barajé otras opciones como fueron el crear una página de comercio electrónico, o por ejemplo la gestión de una biblioteca, por ejemplo, pero finalmente me decidí por este.

Otro objetivo fue el profundizar en los conocimientos previos que tenía con los lenguajes lenguajes PHP[1], el JavaScript[2], el XHTML[3] o bien el SQL[4], éste fue un punto que me trajo ciertos problemas puesto que mis conocimientos no eran precisamente muy extensos ya que se había trabajado poco con estos lenguajes durante la carrera.

Por último decir en este aspecto, que un objetivo implícito al realizar una página web con el contenido que sea, es aprender a montar un servidor. Por lo tanto debía aprender otra cosa nueva que no me habían enseñado en la carrera.

[1] PHP es un acrónimo recursivo que significa ***P**HP **H**ypertext **P**re-processor*

[2] Lenguaje interpretado por páginas web con una sintaxis parecida a C o al Java. Su interpretación se realiza en el lado del cliente.

[3] XHTML, siglas de e**X**tensible **H**ypertext **M**arkup **L**anguage (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. En su versión 1.0, XHTML es la versión XML de HTML y cumple las especificaciones, más estrictas, de XML.

[4] SQL (Structured Query Language) o lenguaje de consulta estructurado. Es un lenguaje estándar normalizado de comunicación con las bases de datos relacionales.

2.2. Justificación

Desde que tenía 8 años hemos tenido un ordenador en casa, eran a finales de los 80 y mi padre compro un Amstrad 8086, recuerdo que desde entonces ya jugaba con el ordenador una vez acababa las tareas del colegio.

A finales de los 90 salieron los primeros MMORPG[5], esto significaba que mediante tu conexión modem podías jugar en tiempo real con cientos de personas de la otra parte del mundo. El primero de estos juegos como tal fue *Ultima Online.*

En el año 2000 salió al mercado un juego español llamado *La Prisión*, que fue el primero de su categoría hecho íntegramente en España, fue un boom de ventas, por entonces aún no había ni Messenger ni ninguna red social como Facebook o Twentti, por lo que a pesar de pagar una cuota mensual muchos usuarios jugaban a diario yo incluido.

Luego vinieron muchos más como *Dark Age of Camelot*, *World of Warcraft* o *Warhammer Online,* todos mantienen la misma idea de los primeros juegos de este tipo: libertad para el jugador, diversión asegurada y la posibilidad de conocer gente de otras partes del mundo y culturas.

Mi proyecto intenta emular el juego de *La Prisión*, es evidente que no puedo competir con él puesto que lleva casi 9 años en el mercado y ha tenido un grupo muy numeroso de personas para su desarrollo además de un elevado presupuesto y tiempo.

Sin embargo, intento mantener la idea original del juego, que fue la de integrar una especie de chat para que los diferentes jugadores pudieran interactuar entre sí, no sin olvidar sobretodo la jugabilidad y la seguridad de la aplicación.

[5] Los videojuegos de rol multijugador masivos en línea o MMORPG (siglas del inglés de *massively multiplayer online role-playing games*), son videojuegos de rol que permiten a cientos o miles de jugadores introducirse en un mundo virtual de forma simultánea a través de Internet, e interactuar entre ellos.

3. Evaluación Tecnológica

Al principio decidir qué debía utilizar fue complicado, sabía lo que quería conseguir pero no como lograrlo. Baraje muchas opciones sobre lo que debía usar, me planteé la posibilidad de realizar el juego usando un ERP[6], el problema es que no podía orientarlo a multijugador y mi aplicación perdía una de sus bazas más importantes. También me planteé realizarlo en Excel y Visual Basic pero esa idea fue rápidamente descartada.

Necesitaba un lenguaje de programación, que no fuera muy complicado de aprender y que me ofreciera potencia a la hora de programar.

Cuando decidí que lo quería orientar a multijugador y que lo necesitaba hacer en una ventana de explorador me di cuenta que el uso de HTML era fundamental.

Pero al realizar la aplicación en XHTML, le di un valor añadido al proyecto, puesto que mi proyecto cumple con los estándares de la W3C[7].

Dicho esto necesitaba un gestor de base de datos para guardar la información MySQL ofrecía una solución a mis problemas, también quise utilizar PostgreSQL[8] pero me decanté por la primera opción.

Finalmente necesitaba crear páginas dinámicamente, esto lo conseguí con el lenguaje de programación PHP.

3.1. Lenguaje de Programación

Como he comentado en la evaluación tecnológica, el lenguaje utilizado ha sido el PHP.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado al usar especialmente para desarrollo web y puede ser incrustado dentro de código XHTML.

Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Además es de programario libre y cuenta con un gran soporte de la comunidad, haciendo que haya multitud de archivos, códigos, tutoriales para buscar ayuda.

Este lenguaje tiene unas características añadidas que voy a describir a continuación:

[6] Los sistemas de planificación de recursos empresariales, o ERP (por sus siglas en inglés, Enterprise resource planning) son sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía comprometida en la producción de bienes o servicios.

[7] El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

[8] PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Ventajas:

- Podemos poner código PHP en etiquetas XHTML.
- Sencillo de aprender, parecido a C
- Todo lo que se pueda transmitir por vía HTTP se puede programar.
- Las variables que se pasan en el espacio de la dirección las interpreta el PHP de forma automática.
- Es de programario libre, no se ha de pagar para su utilización.
- Existen varias versiones del intérprete para Windows o Linux, esto significa que se pude migrar las aplicaciones de una plataforma a otra sin tener que modificar el código.
- Muy buen soporte de acceso a Bases de Datos.
- Seguridad al comprobar si los parámetros de entrada son validos, puesto que los hace el servidor y no el cliente.

Inconvenientes: (se han intentado minimizarlos)

- Al mezclar sentencias XHTML y PHP el código resulta complicado de entender.
- Todo el trabajo lo realiza el servidor.

Teniendo en cuenta que la aplicación es una página web, hace falta usar una maquetación para darle un estilo visual, en este sentido se ha usado el estándar XHTML 1.0 transicional, CSS[9] 2.1 y con la ayuda del Javascript.

3.2. Sistema Gestor de la Base de Datos

El sistema gestor que se ha decidido utilizar ha sido el MySQL.

El MySQL es de programario libre, posiblemente sea el más utilizado debido a su rapidez y facilidad de uso. Además existen muchas librerías y herramientas que permiten su uso a través de una gran cantidad de lenguajes de programación.

Además de ser una base de datos muy rápida, fácil de usar y segura, como gestor de base de datos es una aplicación que se encarga de manejar los datos almacenados en la base de datos de una forma fácil y cómoda.

Permite el uso del lenguaje de programación SQL.

El motor de guardado utilizado ha sido el MyISAM.

La principal característica de este tipo de almacenamiento es la gran velocidad que se obtiene en las consultas (lectura y escritura rápidas), ya que no tiene que hacer comprobaciones de la integridad referencial, ni bloquear las tablas para realizar las operaciones por la ausencia de características de atomicidad.

Este tipo de tablas está especialmente indicado para sistemas que no tienen un número elevado de inserciones como pueden ser las páginas web.

[9] Las hojas de estilo en cascada (en inglés Cascading Style Sheets), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

En la siguiente gráfica se pueden observar los diferentes motores de almacenamiento que puede usar el MySQL.

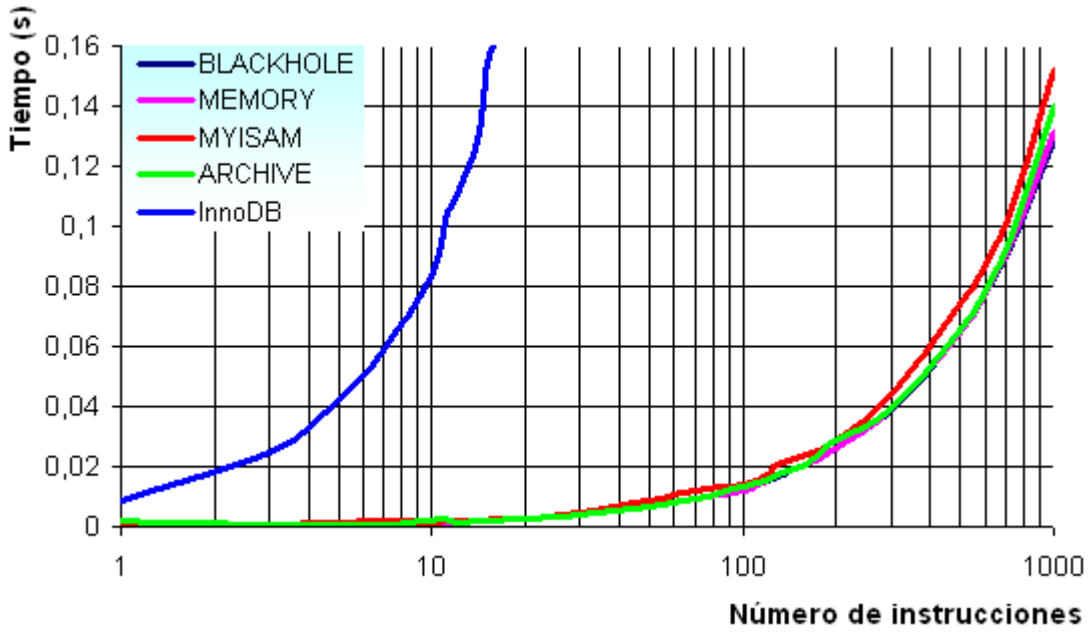


Imagen 3.1. Gráfica de rendimiento de los diferentes motores de almacenado. [1]

El porqué se uso un motor de almacenamiento y no cualquiera de los demás, viene porque los otros motores tenían las características que describo a continuación, y para mi proyecto MyISAM cumplía mejor los requerimientos que ninguno de estos cuatro.

- **Memory:** Mantiene los datos en memoria, lo que permite obtener una velocidad muy alta. Por contra, los datos se pierden al apagar el servidor.

- **Merge:** Posibilita acceder a varias tablas con la misma estructura como si se tratase de una misma tabla.

- **InnoDB:** soporta transacciones a cambio de un sacrificio de la velocidad de lectura y escritura en un factor 1:10.

- **BlackHole:** Procesa todas las consultas pero no almacena los datos en ningún sitio. Es como un agujero negro.

3.3. Servidor Web

Todo el grueso del proyecto se ejecuta en un servidor web.

Un servidor web es un programa que se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet.

El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

Hay dos situaciones posibles en la que la aplicación (es decir, el proyecto) se pueda encontrar ubicado:

- En un ordenador personal
- En un proveedor de servicios de páginas web con bases de datos.

En este caso los servidores Apache son la mejor solución como servidor web, puesto que además de ser de programario libre y de ser la que aplican la mayoría de proveedores de páginas web, se integra perfectamente con el PHP y tiene muchos módulos para extender su funcionalidad.

Apache es un servidor de red para el protocolo HTTP y es multiplataforma[10].

3.4. Servidor de correo

El juego envía correos de verificación al crear una nueva cuenta de usuario siempre y cuando este activa ésta opción.

Nos encontramos con la necesidad de buscar una aplicación que envíe estos correos de manera automática, la solución es la de usar una servidor de correo.

El servidor de correo utilizado fue el MiniRelay, que es de programario libre, y básicamente manda mensajes sin utilizar el SMTP[11] de nuestro proveedor de Internet. Simplemente hay que configurar nuestro programa de correo para que utilice la dirección IP 127.0.0.1 como servidor de SMTP.

[10] El hardware o software que es multiplataforma tiene la característica de funcionar de forma similar en distintas plataformas (distintos sistemas operativos por ejemplo).

[11] Simple Mail Transfer Protocol (SMTP) Protocolo Simple de Transferencia de Correo. Es un protocolo basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.

3.5. Uso del XAMPP como fusión de las necesidades tecnológicas

Para poder instalar todos los programas (PHP, MySQL, Apache) que he necesitado para realizar el proyecto, exceptuando el servidor de correo, he utilizado la aplicación XAMPP. [2]

XAMPP es un servidor que consiste en la base de datos MySQL, el servidor web Apache, y los intérpretes para lenguajes de script: PHP y Perl.

Su nombre proviene del acrónimo **X** (para cualquier sistema operativo) **A**pache, **M**ySQL, **P**HP, **P**erl.

Es un programario libre y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicamente. Incluye aplicaciones de gran utilidad y librerías para el manejo de una página web.

Una de estas herramientas es el phpMyAdmin[12] que ha sido imprescindible en el desarrollo del proyecto.

[12] phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.

4. Planificación Inicial

La planificación inicial del trabajo a realizar nos da unas pautas y unas directrices del tiempo que se debe destinar a cada tarea, este tiempo es muy objetivo[13] puesto que lo que una persona calcula que lo realizara en 5 días otra lo puede realizar en 2.

Las tareas a realizar de un modo global son las siguientes:

- Búsqueda y aprendizaje de las tecnologías a utilizar.
- Diseño y desarrollo del juego
    - *Cálculo de probabilidades, experiencia, combates, etc.*
- Programación y documentación del proyecto.

Profundizando un poco más podemos detallar estos apartados en:

- Búsqueda e información de las tecnologías a usar …….. 9 días
- Diseño …….. 22 días
- Implementación …….. 44 días
- Implantación …….. 1-2días
- Testeo …….. 6 días
- Documentación …….. 13 días

- **TOTAL Estimado …….. 95 días**

Se considera que:

- Los días tienen 8 horas de trabajo
- Las semanas tienen 5 días de trabajo
- No se contabilizan fiestas ni puentes.
- El aprendizaje de las tecnologías a utilizar se contabiliza en el primer punto y se tiene en cuenta el posterior perfeccionamiento y mejora en el punto de la implementación.

Toda esta información se puede ampliar en los puntos 17.1 donde se explica el análisis de costes, y en el punto 17.2 del documento donde se da una comparativa entre la estimación inicial y la final.

[13] Esto se podrá observar al final del documento donde se hace una comparativa entre el tiempo estimado y el tiempo real, en especial en el apartado de la implementación, donde el tiempo estimado inicial fue muy superior al real.

Parte II: Análisis y Especificación

5. Especificación de Requerimientos

En este apartado se quiere dar a conocer las funcionalidades básicas de la aplicación *(requerimientos funcionales)* así como otras características que no dependen del proyecto *(requerimientos no funcionales).*

5.1. Requerimientos funcionales

Este apartado trata de mostrar y documentar lo que se necesita que la aplicación realice para poder transmitirlo de una forma sencilla al posible cliente.

Requerimientos Principales:

- Creación de cuenta de usuario.
- Creación de mensajes en el chat.
- Creación de un mensaje global.
- Creación de mensajes/temas en el foro.
- Comprar armas, armaduras,…
- Vender armas, armaduras,…
- Explorar la prisión.
- Creación de combates.
- Configuración del juego.

Requerimientos Secundarios:

- Gestión de usuarios.
- Salir de la aplicación.

5.2. Requerimientos no funcionales

Los requerimientos no funcionales que se requieren estarán divididos en tres apartados para poder diferenciar a la hora de querer instalar la aplicación, usar la aplicación o ver otros requerimientos que engloban la aplicación.

5.2.1. Instalación de la aplicación

Se necesitan los siguientes requerimientos cuando deseamos instalar la aplicación y poder usarla en un ordenador personal o bien en un proveedor de servicios web en internet:

- Un ordenador con una antigüedad no inferior al año 1999[14].
- Una conexión a internet, preferiblemente de banda ancha.
- Un navegador Web, ya sea el Explorer 6 o superior, el Mozilla Firefox, Opera o Netscape (se recomienda el uso del Mozilla Firefox 3 o superior para evitar posibles incompatibilidades).

[14] Con ordenadores inferiores podría funcionar la aplicación puesto que el uso de CPU y memoria que usa es mínimo (simplemente abrir un navegador web) pero como baremo se recomienda que no sea inferior a esa fecha.

- Poder instalar o tener instalado el paquete XAMPP 1.7 o superior.
- Poder instalar o tener instalado el MiniRelay 0.9 o superior.

En el caso de no poder instalar o no tener instalado el paquete XAMPP 1.7:

- Poder instalar o tener instalado del SGBD MySQL.
- Poder instalar o disponer de un servidor web con el lenguaje PHP incorporado y las librerías para poder trabajar con la base de datos MySQL.

5.2.2. Uso de la aplicación

Para poder hacer uso de la aplicación, cualquier usuario necesita los siguientes requerimientos:

- Un ordenador con una antigüedad no inferior al año 1999.
- Una conexión a internet, preferiblemente de banda ancha.
- Un navegador Web, ya sea el Explorer 6 o superior, el Mozilla Firefox, Opera o Netscape (en estos navegadores la aplicación funciona perfectamente, aunque se recomienda el uso del Mozilla Firefox 3 o superior para evitar posibles incompatibilidades).
- Una cuenta de correo electrónico para poder verificar la cuenta[15] (preferiblemente gmail, puesto que las cuentas de Hotmail y Yahoo producen errores).

5.2.3. Otros requerimientos no funcionales

Estos requerimientos están englobados en la aplicación, dan un valor añadido a este proyecto puesto que aportan calidad.

- Tiempo de respuesta lo más bajo posible.
- Cuadros de ayuda en diversas acciones o imágenes.
- Desarrollo del juego intuitivo. Es importante que el jugador en poco tiempo se familiarice con la aplicación.
- Rendimiento de la Base de Datos elevado al usar el motor de guardado MyISAM[16].
- Rendimiento del servidor web elevado al usar compresión de páginas.

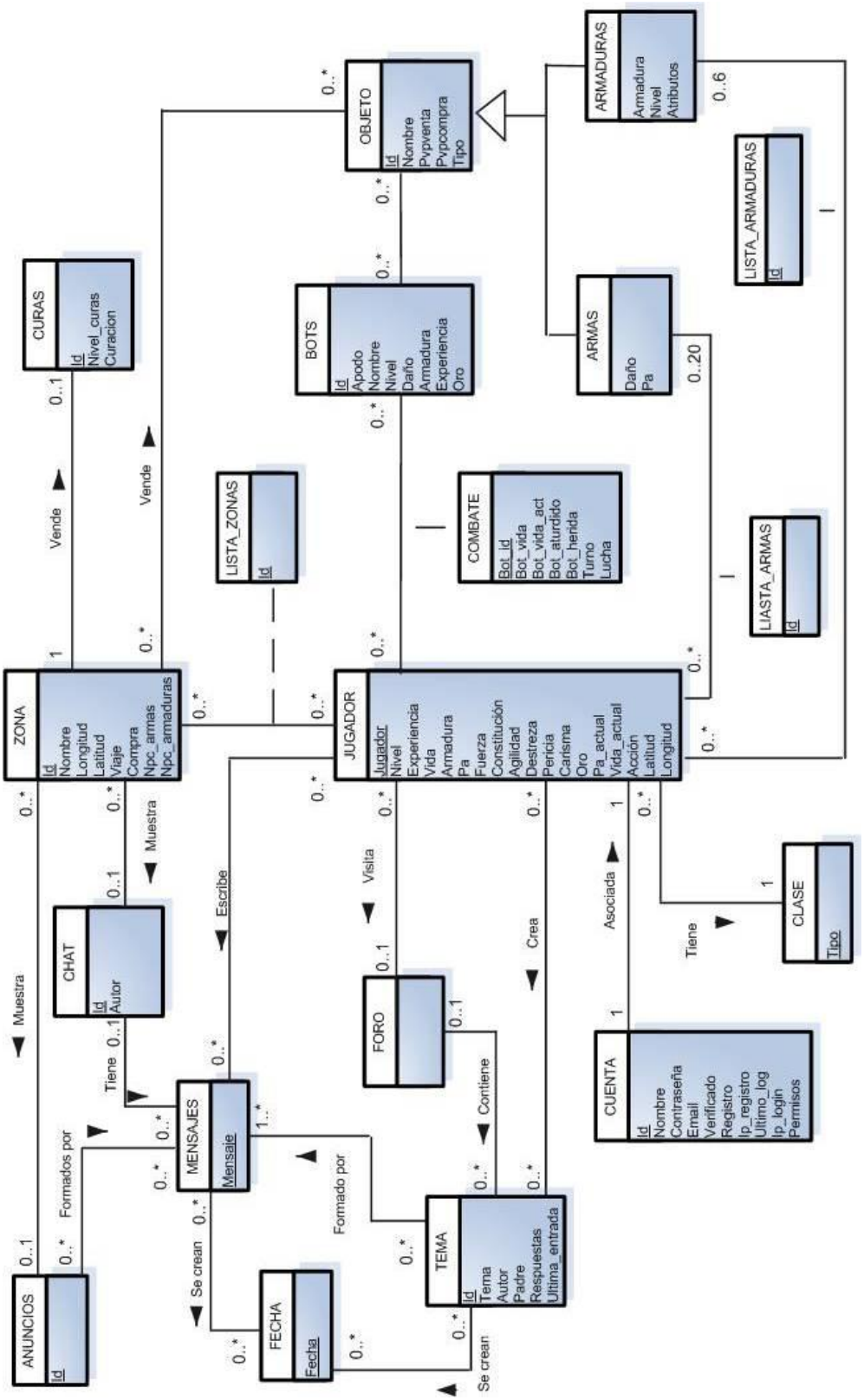
5.2.4. Requerimientos del sistema

Básicamente lo que es necesario en los requerimientos del sistema es que se pueda instalar el paquete XAMPP, o en su defecto el SGBD MySQL y un servidor web con el lenguaje PHP incorporado y las librerías para poder trabajar con la base de datos MySQL, además del el MiniRelay como gestor de correos.

[15] Esto solo sería necesario en el caso de que el administrador hubiera activado la opción de envío de código de autorización.
[16] Ver el punto 3.2 del documento para ver los diferentes motores de guardado disponibles en MySQL.

6. Modelo Conceptual

En este apartado podremos observar el modelo conceptual creado para el proyecto. En el modelo conceptual se pueden ver los conceptos y las relaciones existentes entre estos.



6.1. Diagrama de una aplicación Web

En este diagrama podemos observar cómo se comporta una aplicación web. Un navegador (Explorer, Mozilla, etc.) solicita una página web que contiene un programa PHP al servidor.

Este programa se ejecuta en el servidor y se envía al cliente, es decir al navegador web del usuario normalmente en forma de página HTML o XHTML.

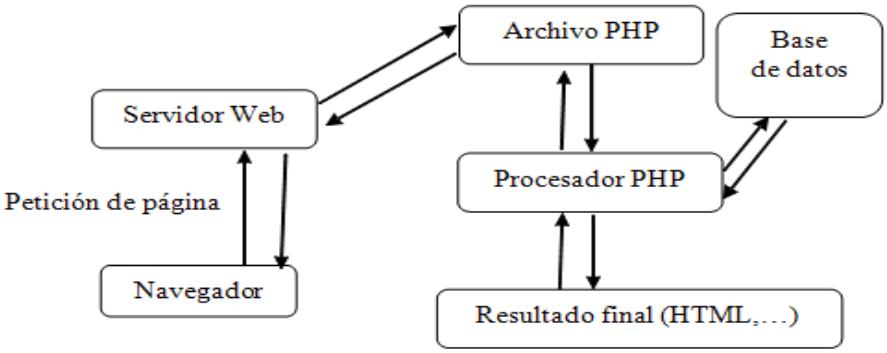


Imagen 6.2. Diagrama de una aplicación Web

7. Casos de Uso

En este apartado mostraré los casos de uso que intervienen en el sistema.

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

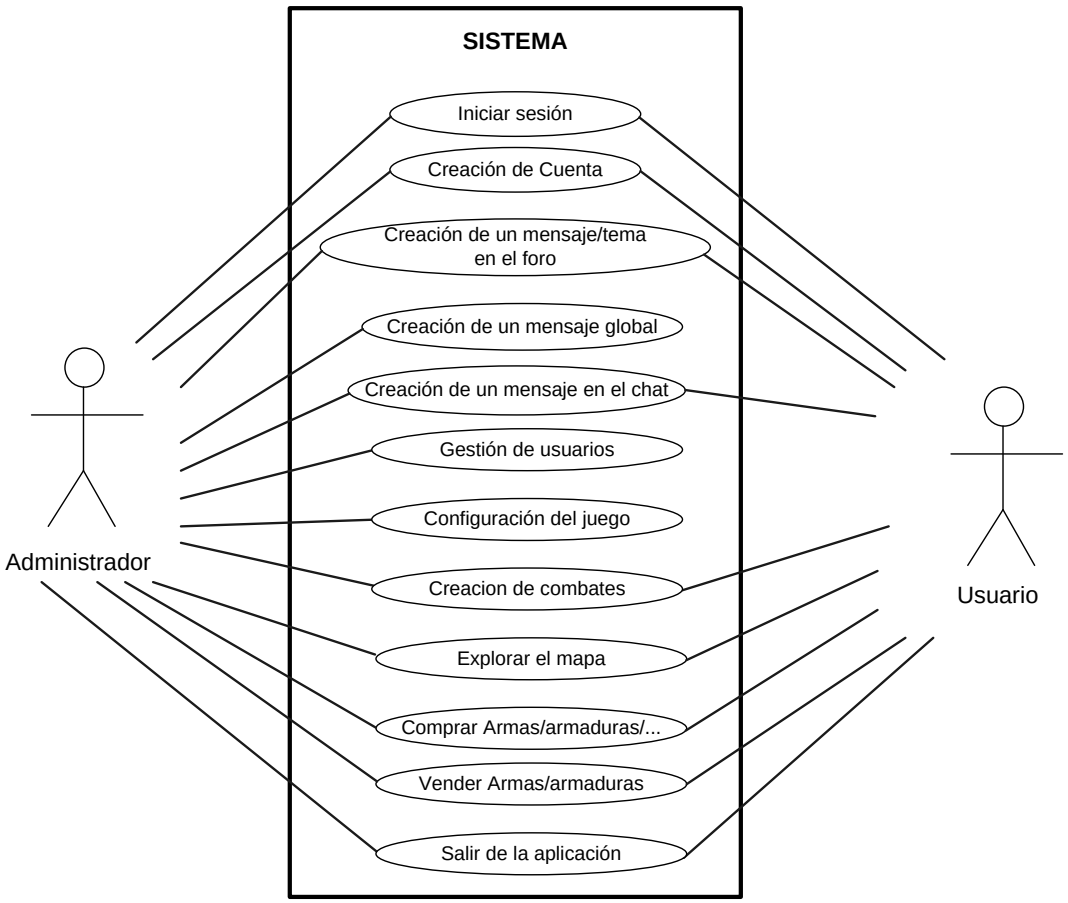


Imagen 7.1. Diagrama del modelo de casos de uso

En los errores de entrada de datos, el sistema muestra el error específico (correo electrónico que no se corresponde con el formato de correo, escritura de números en un campo que solo acepta letras, etc.).

Estos errores se engloban en el apartado de error en la entrada de datos, en los casos de uso que así lo dispongan. Esto se hace para evitar hacer más acontecimientos alternativos explicando el porqué del error.

Se describirán unos cuantos casos de uso en profundidad, no se describirán todos debido a que algunos son muy parecidos con otros ya explicados.

7.1. Iniciar sesión

El inicio de sesión es un caso de uso que contiene el curso normal y una serie de cursos alternativos que se describen a continuación.

Caso de uso: Iniciar Sesión.
Actores: usuarios, administrador.
Tipo: Primario.
Descripción: Controla el acceso a la aplicación para los usuarios, y dependiendo del tipo de usuario se habilitarán algunas opciones o se bloquearan otras.
Requerimientos: Ninguno.

Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El usuario entra en la página de la aplicación web, entra los datos de cuenta y contraseña e ingresa en la aplicación.	
	2. El sistema comprueba que el usuario existe y le otorga los derechos de acceso.
	3. Finalizada la entrada de datos del usuario y la comprobación del sistema, muestra la página de bienvenida al juego.

Curso alternativo de los acontecimientos A. error en la entrada de los datos:
2. El sistema muestra un error conforme el nombre de usuario o contraseña son erróneos y lo devuelve a la página principal de la aplicación.

Curso alternativo de los acontecimientos B. usuario baneado por el administrador:
2. El sistema muestra un error conforme el usuario esta baneado y no tiene los privilegios para entrar en el juego, y lo devuelve a la página principal de la aplicación.

Curso alternativo de los acontecimientos C. El jugador no ha verificado su cuenta:
2. El sistema envía al usuario a la página de verificación de cuenta, y no le permite la entrada al juego.

7.2. Creación de cuenta

La creación de cuenta es un caso de uso que contiene el curso normal y una serie de cursos alternativos que se describen a continuación.

Caso de uso: Creación de cuenta.
Actores: Usuarios.
Tipo: Primario.
Descripción: Se encarga de la creación de una nueva cuenta de juego.
Requerimientos: Ninguno.

Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El usuario entra en la página de la aplicación web, clica en el link de Crear Cuenta, entra los datos requeridos y acepta.	
	2. El sistema comprueba que no existe ninguna cuenta con ese nombre ni ningún jugador con ese nombre ni ningún correo con ese nombre. El sistema crea la nueva cuenta con los datos introducidos.
	3. El sistema envía un correo electrónico al usuario informando de que su cuenta ha sido creada correctamente y necesita verificarla.

Curso alternativo de los acontecimientos A. error en la entrada de los datos:
2. El sistema muestra un error conforme los datos introducidos son erróneos y lo devuelve a la página de creación de cuenta.

Curso alternativo de los acontecimientos B. no es necesario el envío del código de activación:
B3. El sistema muestra un mensaje conforme la cuenta ha sido creada correctamente e invita al usuario a entrar al juego.

Curso alternativo de los acontecimientos C. error en el envío del código de activación:
C3. El sistema muestra un mensaje conforme la cuenta ha sido creada correctamente pero ha habido un error en el envío del código de activación.

7.3. Gestión de Usuarios

La gestión de usuarios es un caso de uso que contiene el curso normal y una serie de cursos alternativos que se describen a continuación.

Caso de uso: Gestión de usuarios.
Actores: Administrador.
Tipo: Secundario.
Descripción: Se encarga de visualizar o modificar los datos de las cuentas de los usuarios de la aplicación.
Requerimientos: Haber iniciado la sesión como administrador.

Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El administrador clica en el link de edición de jugadores dentro del panel de administrador.	
	2. El sistema muestra la lista de todos los usuarios que poseen una cuenta en el juego.
3. El administrador clica encima del nombre de la cuenta de un usuario.	
	4. El sistema muestra toda la información asociada a esa cuenta.
5. El administrador modifica los campos oportunos y clica en botón de aceptar.	
	6. El sistema actualiza la información en la cuenta del usuario.

Curso alternativo de los acontecimientos A. error en la entrada de los datos:
5. El sistema muestra un error conforme los datos introducidos son erróneos y lo devuelve a la página de edición de jugadores.

Curso alternativo de los acontecimientos B. tiempo de inactividad excedido:
3. El sistema devuelve al administrador a la página principal de la aplicación si lleva 10 minutos o más sin realizar ninguna acción al sistema, situación que provoca el finalizar la sesión.

7.4. Creación de un mensaje global

La creación de un mensaje global es un caso de uso que contiene el curso normal y una serie de cursos alternativos que se describen a continuación.

Caso de uso: Creación de un mensaje global.
Actores: Administrador.
Tipo: Primario.
Descripción: Se encarga de crear un mensaje que será visto por todos los jugadores.
Requerimientos: Haber iniciado la sesión como administrador.

Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El administrador clica en el link de crear mensaje dentro del panel de administrador y escribe el mensaje que desee, acto seguido acepta.	
	2. El sistema actualiza esa información que será vista por todos los usuarios.

Curso alternativo de los acontecimientos A. error en la entrada de los datos:
1. El sistema muestra un error conforme los datos introducidos son erróneos y lo devuelve a la página de inserción de mensaje.

Curso alternativo de los acontecimientos B. tiempo de inactividad excedido:
1. El sistema devuelve al administrador a la página principal de la aplicación si lleva 10 minutos o más sin realizar ninguna acción al sistema, situación que provoca el finalizar la sesión.

7.5. Explorar el mapa

El explorar el mapa es un caso de uso que contiene el curso normal y una serie de cursos alternativos que se describen a continuación.

Caso de uso: Explorar el mapa.
Actores: Administrador, usuarios.
Tipo: Primario.
Descripción: Se encarga de modificar la posición actual del jugador en la prisión.
Requerimientos: Haber iniciado la sesión como administrador o usuario.

Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El administrador o usuario clica en una de las direcciones de movimiento.	
	2. El sistema actualiza la nueva posición del jugador.

Curso alternativo de los acontecimientos A. encontrar un bot:
2. El sistema actualiza la posición del jugador y genera un combate entre el jugador y el bot.

Curso alternativo de los acontecimientos B. tiempo de inactividad excedido:
1. El sistema devuelve al administrador o jugador a la página principal de la aplicación si lleva 10 minutos o más sin realizar ninguna acción al sistema, situación que provoca el finalizar la sesión.

7.6. Creación de combates

La creación de combates es un caso de uso que contiene el curso normal y una serie de cursos alternativos que se describen a continuación.

Caso de uso: Creación de combates.
Actores: Administrador, usuarios.
Tipo: Primario.
Descripción: Se encarga de crear un combate entre un jugador y un bot.
Requerimientos: Haber iniciado sesión como admin o usuario, haber encontrado un bot

Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El administrador o usuario obtiene el primer turno de ataque si ha ganado la iniciativa.	
	2. El sistema obtiene el turno de ataque cuando el jugador haya finalizado el suyo o bien obtiene el primer turno de ataque si gano la iniciativa.
3. El administrador o jugador obtiene su turno de ataque cuando el sistema finaliza el suyo.	
	4. El sistema posibilita el obtener una recompensa en el caso de que el jugador haya ganado el combate.
	5. El sistema actualiza la información en la cuenta del usuario.

BUCLE

Curso alternativo de los acontecimientos A. El bot gana la pelea:
2. El sistema devuelve al administrador o jugador a la celda, situación que provoca el fin del combate.

Curso alternativo de los acontecimientos B. tiempo de inactividad excedido:
3. El sistema devuelve al administrador o jugador a la página principal de la aplicación si lleva 10 minutos o más sin realizar ninguna acción al sistema, situación que provoca el finalizar la sesión.

7.7. Configuración del juego

La configuración del juego es un caso de uso que contiene el curso normal y una serie de cursos alternativos que se describen a continuación.

Caso de uso: Configuración del juego.
Actores: Administrador.
Tipo: Primario.
Descripción: Se encarga de modificar los datos de las características básicas del juego.
Requerimientos: Haber iniciado la sesión como administrador.

Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El administrador clica en el link de configuración dentro del panel de administrado, rellena los campos y acepta.	
	2. El sistema comprueba que los campos son correctos y actualiza la información.

Curso alternativo de los acontecimientos A. error en la entrada de los datos:
2. El sistema muestra un error conforme los datos introducidos son erróneos y lo devuelve a la página de configuración.

Curso alternativo de los acontecimientos B. tiempo de inactividad excedido:
1. El sistema devuelve al administrador a la página principal de la aplicación si lleva 10 minutos o más sin realizar ninguna acción al sistema, situación que provoca el finalizar la sesión.

7.8. Comprar armas, armaduras

El comprar armas, armaduras, etc. es un caso de uso que contiene el curso normal y una serie de cursos alternativos que se describen a continuación.

Caso de uso: Comprar armas, armaduras
Actores: Administrador, usuarios.
Tipo: Primario.
Descripción: Se encarga añadir una nueva arma, armadura al inventario del jugador.
Requerimientos: Haber iniciado la sesión como administrador o usuario.

Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El administrador o usuario clica en el link de compra de armas/armaduras dentro de la zona	
	2. El sistema muestra todas las armas/armaduras disponibles en esa zona.
3. El administrador o usuario clica en el arma/armadura que desea comprar.	
	4. El sistema añade la nueva arma/armadura al inventario del jugador.

Curso alternativo de los acontecimientos A. error límite de armas excedido:
4. El sistema muestra un error conforme el jugador ya tiene el máximo número de armas equipadas y no puede equiparse más.

Curso alternativo de los acontecimientos B. tiempo de inactividad excedido:
3. El sistema devuelve al administrador a la página principal de la aplicación si lleva 10 minutos o más sin realizar ninguna acción al sistema, situación que provoca el finalizar la sesión.

7.9. Salir de la aplicación

El salir de la aplicación es un caso de uso que contiene el curso normal que se describen a continuación.

Caso de uso: Salir de la aplicación.
Actores: Administrador, usuarios.
Tipo: Secundario.
Descripción: Cierra la sesión actual y vuelve a la página principal de la aplicación
Requerimientos: Haber iniciado la sesión como usuario o administrador.

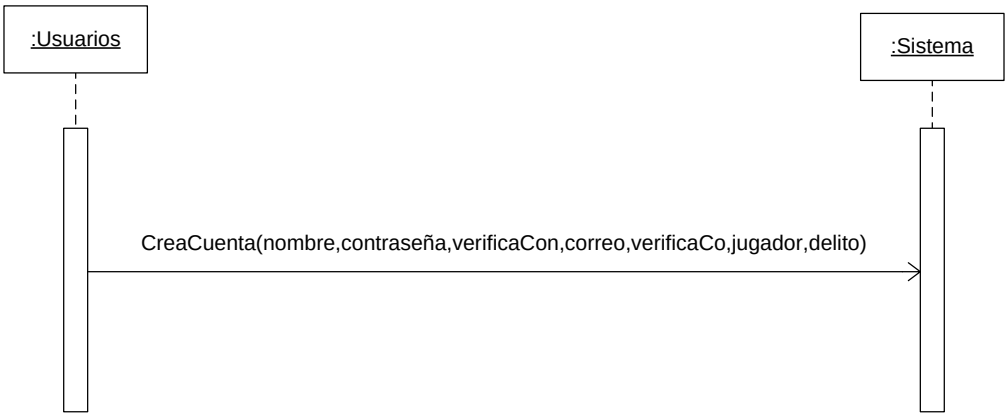
Curso normal de los acontecimientos:

Acciones de los actores	Respuesta del sistema
1. El usuario o administrador clica en el link de cerrar la sesión.	
	2. El sistema cierra la sesión y devuelve a la página principal de la aplicación.

8. Diagramas de secuencia

Los diagramas de secuencia muestran las acciones que tienen los casos de uso. Mostraré algunos ejemplos de diagramas de secuencia de los casos de uso que he desarrollado anteriormente, no haré todos los diagramas de secuencia por el mismo motivo que no desarrollé todos los caso de uso.

8.1. Creación de cuenta



Contracto	**CreaCuenta (nombre, contraseña, verificaCon, correo, verificaCo, jugador, delito)**

PRE:

1. La c:cuenta con c.nombre=nombre no existe en el sistema.
2. El c.jugador=jugador no existe en el sistema.
3. El c.correo=correo no existe en el sistema.

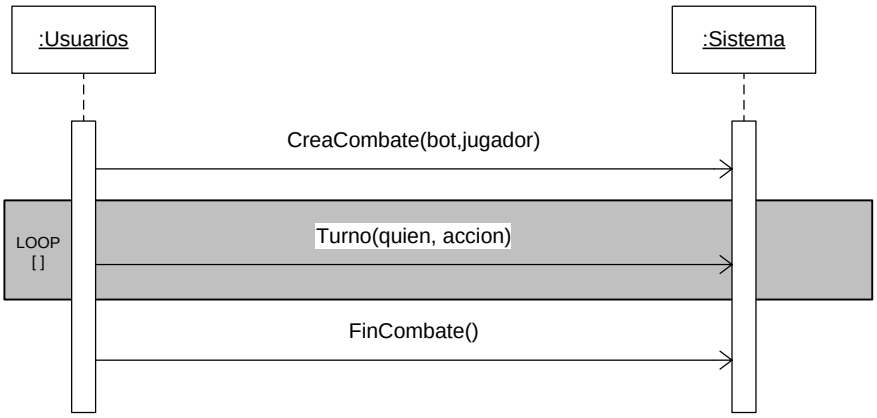
POST:

1. El sistema guarda la información de la c:cuenta en el sistema.

Responsabilidades

CreaCuenta (nombre, contraseña, verificaCon, correo, verificaCo, jugador, delito)
Es responsable de crear una nueva cuenta con los campos requeridos

8.2. Creación de combates



Contracto	**CreaCombate(bot, jugador)**

PRE:

1. Existe un usuario j:jugador que es administrador o usuario identificado por j.id y está dentro del juego.
2. Existe un b:bot identificado por b.id y es un b válido.
3. No existe ningún combate activo.

POST:

1. Se crea un nuevo combate asociado a la relación j:jugador y b:bot.
2. c pasa a ser el combate activo.

Contracto	**Turno(quien, acción)**

PRE:

1. Existe un c:combate activo.
2. La acción a realizar es válida.
3. Quien, es el j:jugador o el b:bot en el primer turno y viene dado por la lista de iniciativas.

POST:

1. Actualizamos la información del combate.
2. Quien, pasa a ser el j:jugador en caso de que tuviera el turno el bot o pasa a ser el b:bot en caso de que tuviera el turno el jugador.
3. La PRE se mantiene.

Contracte	**FinCombate()**

PRE:

1. Existe un c:combate activo
2. Si el j:jugador ganó el c:combate se genera una recompensa.

POST:

1. Se actualiza la información del j:jugador y se actualizan los datos del usuario, si acepto la recompensa.
2. No existe ningún c activo.

Responsabilidades

CreaCombate (bot, jugador)

Es responsable de obtener los datos del jugador y del bot que se usaran en el combate.

Turno (quien, acción, iniciativa)

Es responsable de otorgar el turno de combate al jugador o al bot.

FinalizaCombate()

Es responsable de finalizar el combate.

Parte III: Diseño

9. Arquitectura del diseño de la aplicación

El diseño del proyecto sigue el modelo de la programación por capas, en concreto se ha usado la de 3 capas.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

- **Capa de presentación:** es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.

- **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

- **Capa de datos**: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

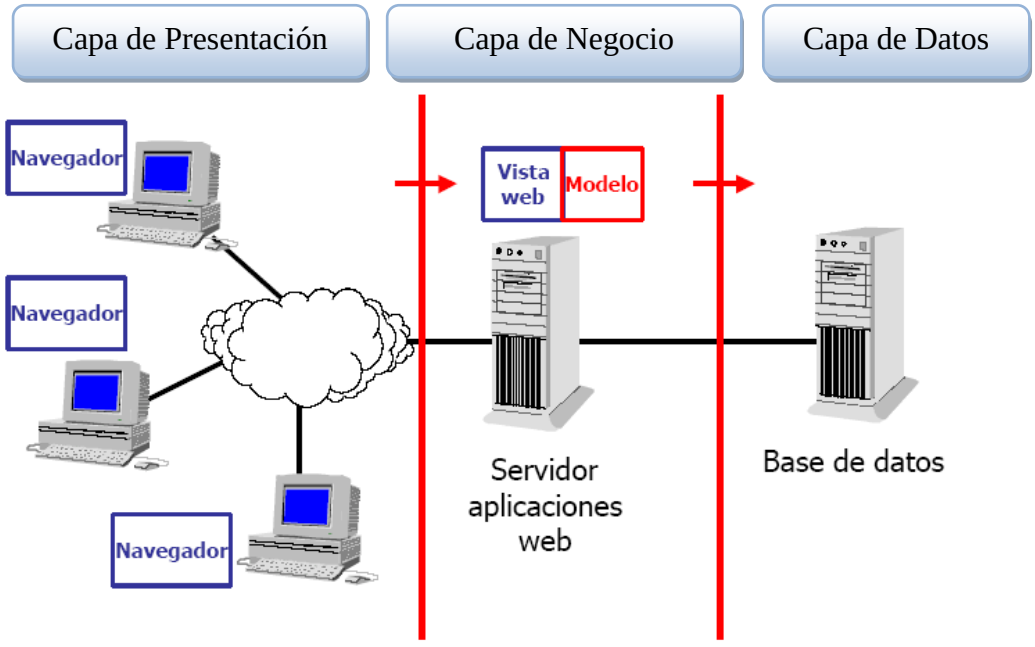


Imagen 9.1. Arquitectura web en 3 capas

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor).

Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores.

Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

10. Diagramas de colaboración

Los diagramas de colaboración indican gráficamente las relaciones entre los diferentes objetos para conseguir los objetivos de cada caso de uso. Cada diagrama de secuencia de cada caso de uso tiene su correspondiente diagrama de colaboración.

Solo mostraré el diagrama de colaboración del caso de uso de creación de combate, los otros diagramas de secuencia son triviales.

10.1. Diagrama de colaboración de la creación de combate

El controlador del sistema es un controlador de caso de uso artificial, es decir que no se encuentra en modelo conceptual del problema.

- **CrearCombate(bot,jugador)** Crea un nuevo combate, el controlador busca los datos del bot y del jugador en el repositorio de bots y usuarios. Acto seguido el controlador crea el nuevo combate con los datos obtenidos.

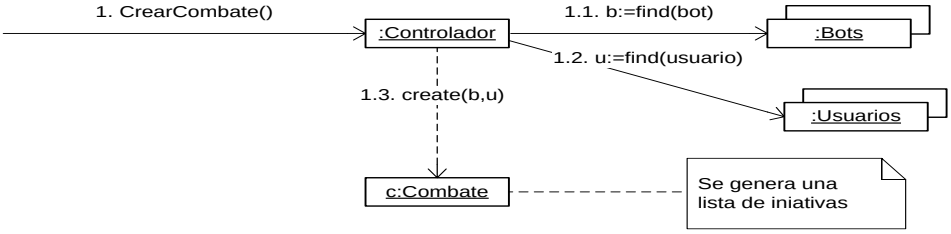


Imagen 10.1. Diagrama de colaboración para crear un combate

- **Turno (quien, acción)** Crea un nuevo turno, el controlador recibe los datos y crea el nuevo turno.

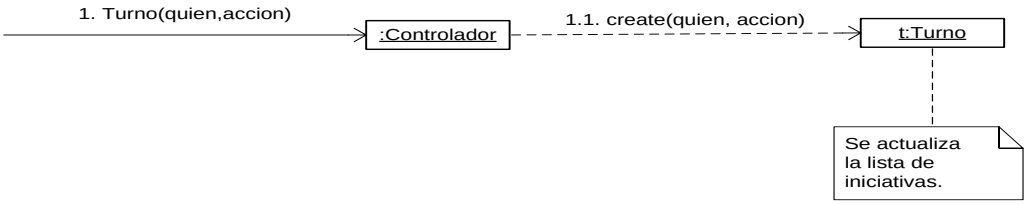


Imagen 10.2. Diagrama de colaboración para crear un turno

- **FinCombate()** Finaliza el combate, el controlador recibe la petición y actualiza la información del usuario con los resultados del combate.

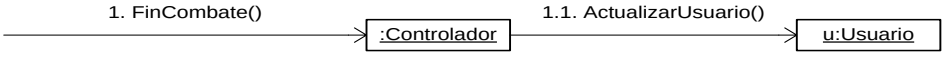


Imagen 10.3. Diagrama de colaboración para finalizar un combate

11. Diseño de la aplicación

En el apartado de diseño mostraré como se ha creado la aplicación, así como los delitos, la experiencia de los diferentes niveles y en general lo relacionado con el diseño del juego.

Este punto se podría extender explicando cada detalle de cómo se creó el juego, pero considero que no es el objetivo de esta memoria, en el caso de querer ver el motor interno del juego el código está comentado de tal manera que se entiende perfectamente que se está realizando en cada caso.

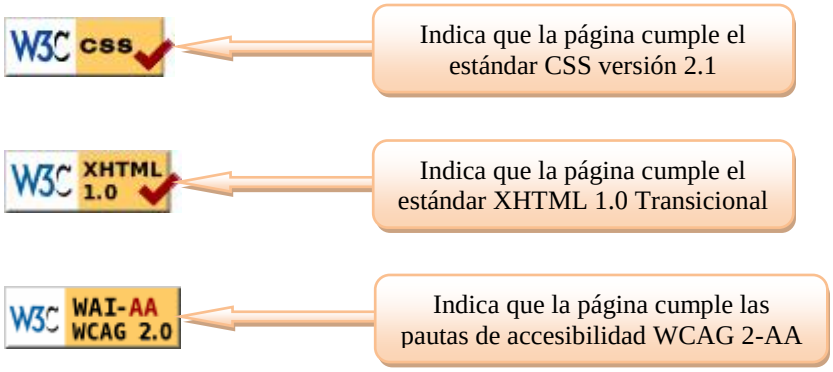
11.1. Diseño de la interfaz de usuario

Para el diseño de la interfaz se ha tenido en cuenta que la distribución de la información sea clara, entendedora y sobretodo que sea intuitivo el poder acceder a los diferentes apartados de la aplicación.

11.1.1. Pantallas generales

Todas las pantallas siguen el mismo patrón exceptuando la página de ayuda y el menú de administrador que luego hablaré de ellas.

Imagen 11.1. Página inicial de la aplicación, con los diferentes campos de información



Para todas las demás pantallas se sigue la misma división que la que muestro en la imagen de la pantalla inicial, es decir:

1- Es la cabecera de la página, aquí siempre irá el nombre del juego.

2- La columna izquierda tendrá diferentes usos, cuando el jugador no esté conectado le aparecerán los campos que se ven en la imagen, el de ingreso y el de cuentas, cuando esté conectado aparecerá información sobre personaje del jugador.

3- Este punto es el contenido principal de información de cada pantalla, en esta columna se informará de todo lo relativo al juego, desde los combates hasta información de la zona actual del jugador, si está explorando, si está consultando el ranking, etc.

4- La columna derecha en general mostrará unos links fijos e información sobre la posición del jugador y estadísticas del combate, es donde se encontrarán la mayoría de links para realizar las diferentes acciones del juego.

5- El pie de página siempre será el mismo, indica el tiempo que se ha tardado en generar la página y el número de consultas que se lanzan a la base de datos.

Para hacernos una idea de cómo se divide la información en un instante del juego muestro la siguiente imagen:

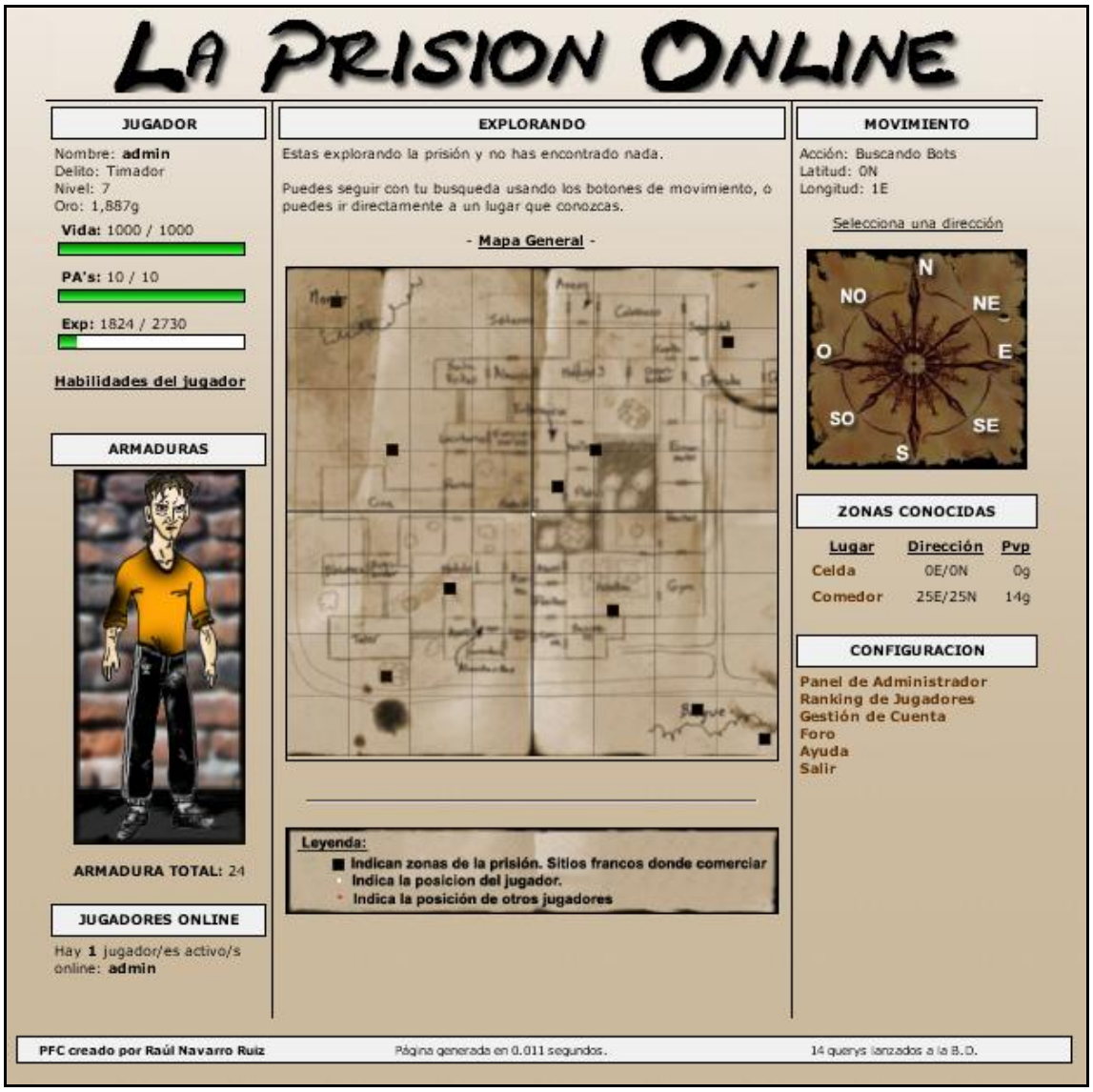


Imagen 11.2. Captura de un instante del juego para mostrar cómo se divide la información

Como he comentado anteriormente, podemos observar que la cabecera siempre se mantiene igual.

En el caso de la columna izquierda muestro información sobre las características, habilidades y armadura del jugador así como los jugadores conectados.

La columna central muestra información sobre la acción actual del jugador, en este caso el jugador está explorando la prisión.

La columna derecha muestra el menú de navegación, para desplazarse por el mapa y luego son links a diferentes acciones, como por ejemplo, el ir a la zona de celdas o del comedor, así como acceder al menú de administrador, el foro, etc.

El pie de página siempre es igual, pero variando el tiempo y el número de querys.

11.1.2. Pantalla de ayuda en línea

La pantalla de ayuda básicamente solo contiene texto plano con algunas imágenes, pero no tiene nada que ver con la estructura usada en las pantallas del juego.

La ayuda tiene links para ir directamente a la información solicitada, pero el código de estas funcionalidades las mostraré en el apartado de implementación.

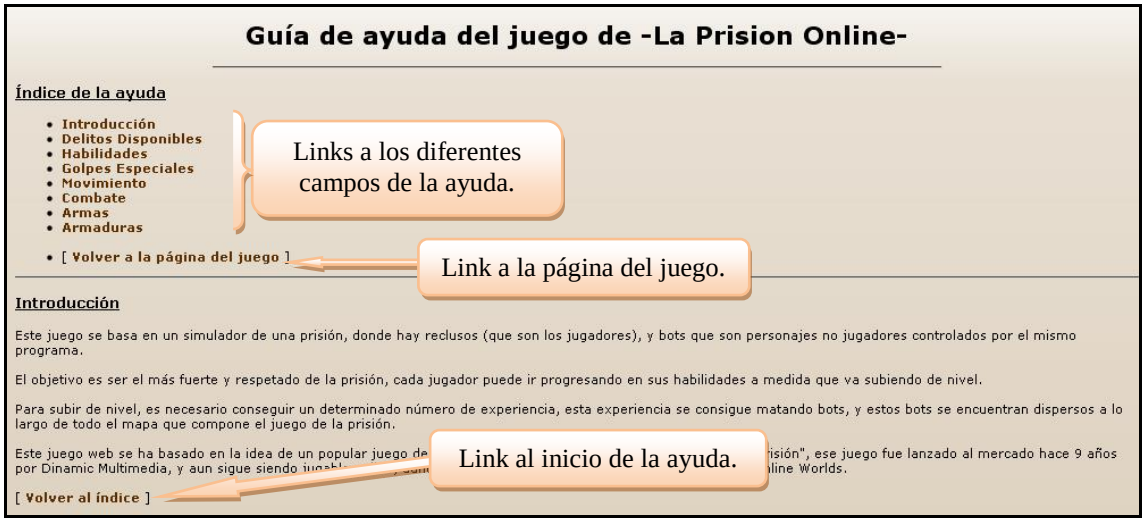


Imagen 11.3. Pantalla de ayuda en línea del juego

Esta página es más extensa, pero mostrando una pequeña parte se entiende la idea de cómo se ha diseñado, en caso de querer verla entera se sugiere que se mire dentro de la aplicación.

11.1.3. Pantalla del menú de administrador

La pantalla de administrador se asemeja en la estructura de las páginas generales del juego, pero en vez de tener cinco campos solo la he diseñado para tres.

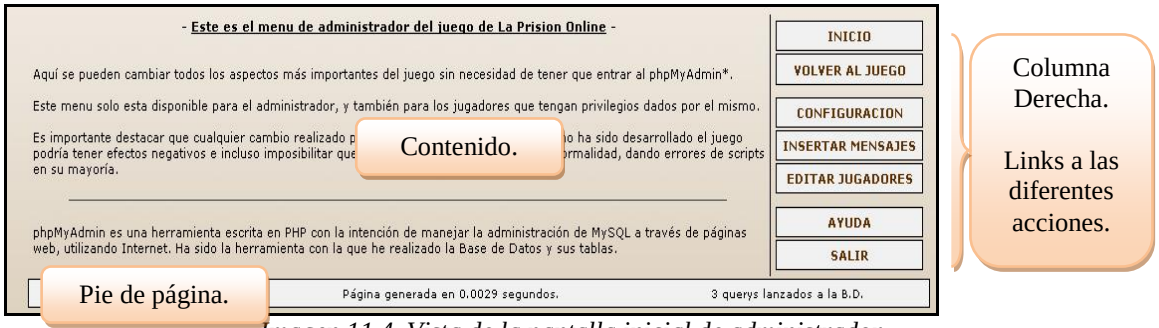


Imagen 11.4. Vista de la pantalla inicial de administrador

Como se puede ver en la imagen se han suprimido el campo de la columna izquierda y el campo del nombre del juego, este último se podría haber mantenido pero se decidió omitirlo puesto que en un futuro si se ponían banners de publicidad no entorpeciera la navegación del administrador.

11.2. Diseño de los delitos

En este punto se quiere dar una pincelada superficial sobre los diferentes delitos, sus habilidades y los golpes especiales.

Hay tres clases de delitos en el juego, cada una tiene sus características propias, haciendo por ejemplo, que una clase sea más efectiva que otra con las armas a distancia o tenga un penalizador en el uso de armas contundentes, etc.

Mercenario
- Especialista en Armas a distancia.
- Bonus en la herida sangrante.
- Torpe con las armas contundentes.
- Más capacidad del uso de las armas *(puntos de acción elevados)*
- Requiere más experiencia para subir cada nivel.

Narcotraficante
- Especialista en Armas Blancas.
- Bonus en el corte profundo.
- Torpe con las armas a distancia.
- Agilidad elevada.
- Experiencia moderada para subir cada nivel.

Timador
- Especialista en Armas Contundentes.
- Bonus en el aturdimiento
- Torpe con las armas a blancas.
- Carisma elevado.
- Requiere menos experiencia para subir de nivel.

11.2.1. Habilidades

Las habilidades de los jugadores influyen en las probabilidades de realizar impactos, evadir golpes, hacer golpes críticos, etc.

- **Fuerza:** Indica lo fuerte en general que es el jugador, este atributo se tiene en cuenta para los cálculos relacionados con el daño de las armas contundentes y con las probabilidades de realizar un aturdimiento.

- **Constitución**: Indica la resistencia de un jugador, a más constitución menos daño recibirá de los bots, se ha de tener en cuenta que también influye la protección en este cálculo. Además la constitución se tiene en cuenta para los cálculos relacionados con el aturdimiento.

- **Agilidad**: Indica lo rápido y ágil que es un jugador. Influye en determinar quien empieza los combates, en las probabilidades de realizar golpes críticos, en las posibilidades de evadir golpes, en las probabilidades de huir de una pelea y en las probabilidades de ataques especiales.

- **Destreza**: Indica lo diestro que es el jugador en realizar golpes. Este valor influye en el daño con armas blancas y en las probabilidades de realizar un corte profundo.

- **Pericia**: Indica la puntería que tiene el jugador. Este valor se tiene en cuenta para determinar el daño con armas a distancia así como para realizar con éxito una herida sangrante.

- **Carisma**: Indica la labia del jugador para obtener mejores precios. En general a mayor carisma, menor precio se necesitará para comprar armas, armaduras y curaciones. También obtendrá un mejor precio al vender las armas.

11.2.2. Golpes Especiales

Los golpes especiales, son golpes que por su naturaleza otorgan un gran bonus al jugador que los realiza con éxito.

- **Herida Sangrante:** Exclusivo de armas a distancia. Determina lo certero de un disparo con armas a distancia y provocará una herida que sangrará hasta que se cierre por completo.

- **Corte Profundo:** Exclusivo de armas blancas. Determina nuestra destreza para efectuar mucho más daño con armas blancas.

- **Aturdimiento:** Exclusivo de armas contundentes. Si un personaje es aturdido, perderá completamente el turno de combate hasta que recupere la consciencia.

11.2.3. Progresión de Niveles

Al jugar a un juego de rol normalmente todos los personajes tienen una evolución diferente de la experiencia a conseguir para subir de nivel. De igual forma los puntos de experiencia para subir de nivel, en niveles bajos, son muy diferentes que en los niveles altos siendo estos últimos mucho más difíciles de conseguir.

Con estas premisas se quiso diseñar una evolución de la experiencia para los diferentes delitos que fuera equilibrada a su dificultad. Me explico, el mercenario es una clase fácil de jugar puesto que tiene muchos puntos de acción y mucha vida, no sería justo que tuviera la misma dificultad a la hora de subir de nivel que un timador, que tiene menos puntos de acción y una vida inferior.

Después de muchas pruebas se llegó a la conclusión que una progresión exponencial era lo más idóneo para el correcto funcionamiento del juego.

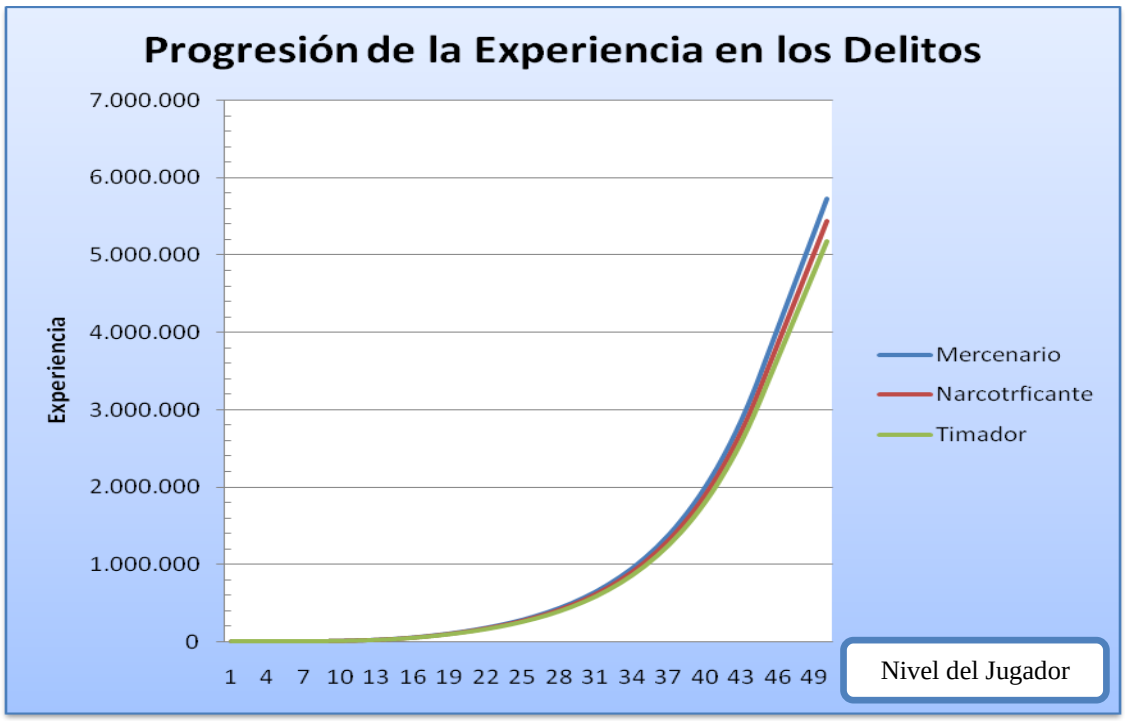


Imagen 11.5. Progresión de los puntos de experiencia necesarios para subir de nivel

En la imagen anterior se puede ver como es asequible llegar aún nivel 20, y se puede ver claramente como a partir del nivel 40 la dificultad de subir de nivel aumenta de forma drástica.

11.2.4. Progresión de los puntos de acción

En el juego original cada delito tiene unos puntos de acción diferentes, en el juego también se quiso emular este aspecto.

El diseño de los puntos de acción para cada delito va en función de los puntos de acción de las armas y el daño que realizan, con esto quiero decir que los p.a.'s (puntos de acción) no fueron asignados arbitrariamente al nivel y al delito, sino que siguen la progresión de daño de las armas.

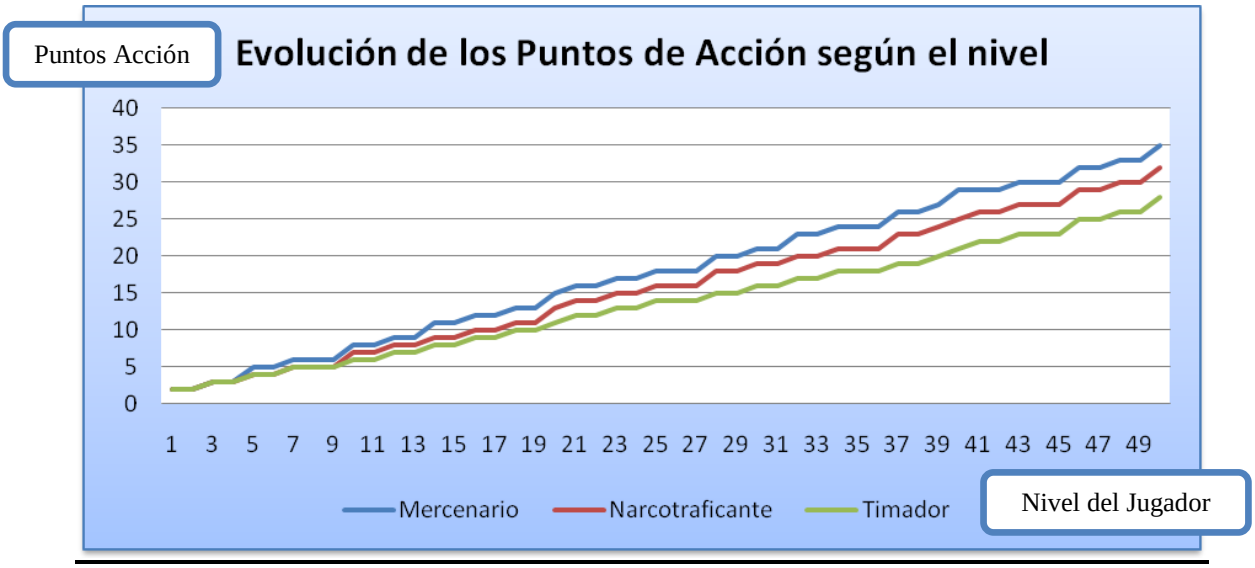


Imagen 11.6. Progresión de los puntos de acción en función del nivel y de los delitos

11.3. Caracterización de los personajes

Este es un punto importante en el proyecto puesto que un objetivo es que el jugador se identifique con su personaje, y este tiene que tener un toque divertido además de que este caracterizado según el delito que represente.

Básicamente en el juego original de Dinamic Multimedia hay unos 12 delitos, de estos delitos se eligieron tres: Mercenario, narcotraficante y timador.

Para el diseño de cada uno de ellos se dibujo sobre un papel y posteriormente se escaneo la imagen. Este fue el resultado final:

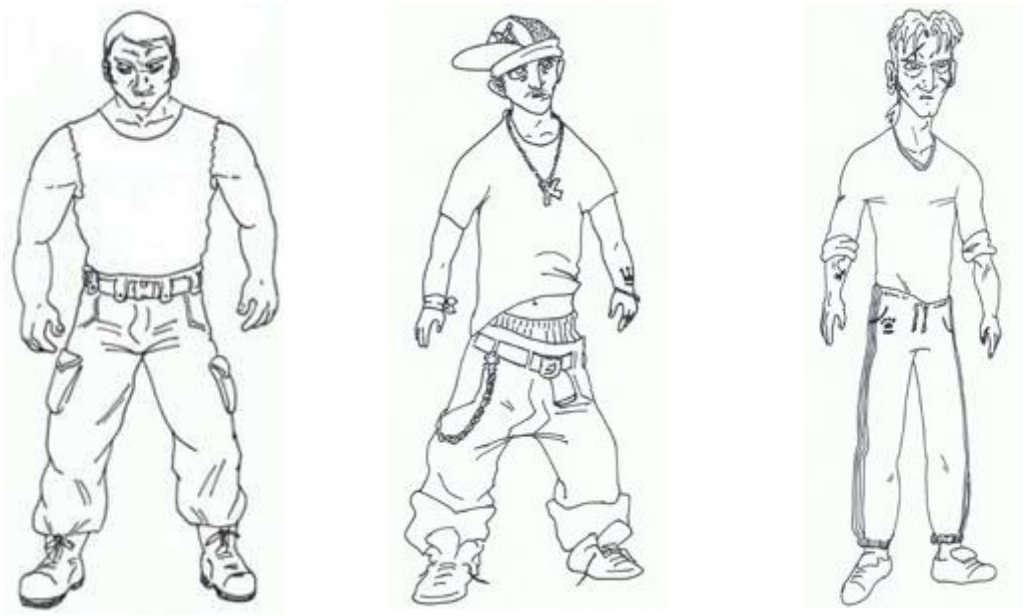


Imagen 11.7. Boceto original de los tres delitos

Se utilizo el Photoshop CS2 como herramienta para pintar los personajes. Además ofrecía técnicas de tratamiento del color, sombreados y otros efectos que se pudieron aplicar al resultado final:

Imagen 11.8. Pintado de los delitos usando Photoshop CS2

Una vez se tenían los dibujos pitados fue el momento de introducirlos en la aplicación, pero aún faltaba añadir un fondo que para obtener un mejor resultado final.

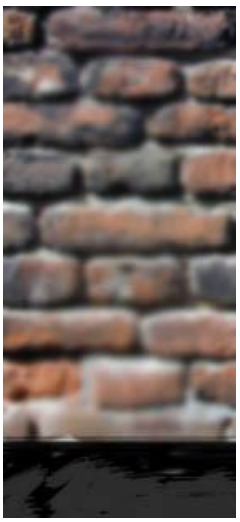


Imagen 11.9. Esta es la imagen utilizada como fondo para los personajes

Se buscaron imágenes por internet que simularan esta pared y finalmente se encontró una que fue la elegida.

Se tuvo que desenfocar el fondo para que no comiera protagonismo a los personajes y se tuvo que añadir una especie de suelo para que los personajes no estuvieran 'volando' en la foto.

También se necesitó crear un borde de la imagen para hacerla un poco más elegante, esto se desarrollo después de juntar todas las diferentes capas que formaron la imagen final.

Finalmente teniendo los personajes pintados, el fondo de la imagen solo hubo que montarlo todo y usar algunas herramientas de equilibrio de color para que los personajes se integraran en el fondo. Después de todo este trabajo, el resultado final de los personajes fue el siguiente:

Imagen 11.10. Diseño final de los delitos

En un futuro no se descarta la incorporación de nuevos delitos, pero como base tres delitos es suficiente para tener diversidad.

11.4. Diseño gráfico de las armas

Este punto fue un añadido a la idea original del proyecto, en un principio no se quería mostrar ninguna imagen del arma que se usaba o de la medicina que se compraba, pero para añadir un poco más de jugabilidad y variedad se decidió finalmente añadir estas imágenes.

Las imágenes se obtuvieron de la página de www.prisionserver.com. El problema fue que las imágenes tenían un fondo que no era apropiado para usar con la aplicación, esto lo veremos más claro en el ejemplo que mostraré a continuación.

Imagen original

Este fue el resultado después de aplicar un nuevo fondo en color negro y varios tratamientos de corrección del color.

Imagen final

Este nuevo fondo y corrección de color tuvo que hacerse a las 211 armas diferentes que componen el juego y a las 10 medicinas diferentes, haciendo un total de 221 modificaciones de imagen, fue un trabajo arduo pero que al final añade calidad a la aplicación final.

11.4.1. Gráfica del daño medio de las armas

Las armas dentro del juego no son iguales, cada una tiene unas características que hacen que no existan dos armas idénticas.

Para el diseño del daño que debían hacer las armas se tuvieron en cuenta diversos factores como por ejemplo a qué nivel serían usadas, contra los niveles de que bots serían efectivas, etc.

La siguiente imagen muestra el gráfico de dispersión del par, daño y puntos de acción que necesita cada arma para ser usada.

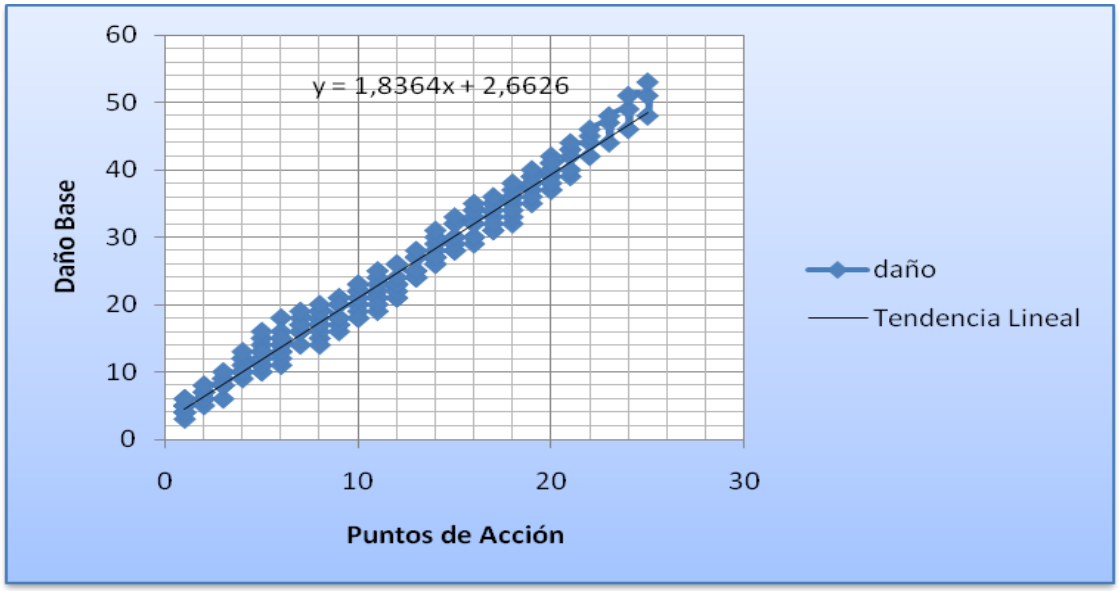


Imagen 11.11. Gráfico de dispersión de los valores de daño y puntos de acción en las armas

La y corresponde al daño base y la x a los puntos de acción.

Esta gráfica se interpreta de la siguiente manera: existe una tendencia lineal en el conjunto de todos los valores cotejados. Así como se puede ver un ejemplo de que hay armas de usan 10 puntos de acción, pero su daño es diferente (va del rango del 19 al 25 de daño base aproximadamente).

11.5. Regla de los 9's

Dentro de un combate el motor interno del juego divide en 6 zonas diferentes las posibles zonas de impacto de un golpe que se realizo con éxito.

Para calcular a que parte iba dirigido el golpe se uso la 'regla de los nueves'. Esta regla se utiliza en medicina para calcular la superficie corporal quemada

Según este método, la cabeza corresponde a un 9% de la superficie corporal total, cada extremidad superior, otro 9%, el tronco un 18%, y el dorso otro 18%, cada extremidad inferior, un 18%, y los genitales externos, el 1% restante.

Por lo tanto las probabilidades de impactar en la cabeza son un 9%, en los brazos un 18%, en las piernas un 36% y en el torso un 36%, estos porcentajes se usan en cada combate.

Consideré curiosa esta explicación del motor interno del juego, puesto que se basa en una aplicación real que se hace sobre pacientes con quemaduras.

Parte IV: Implementación y pruebas

12. Tecnologías utilizadas

Tal y como se explica en el punto 3, las tecnologías utilizadas fueron el lenguaje de programación PHP, XHTML, JavaScript, un sistema gestor de bases de datos MySQL, usando como servidor el Apache y como servidor de correo el MiniRelay.

El desarrollo del proyecto se ha hecho utilizando el Windows XP, el 93% [3] de los usuarios que utilizan sistemas operativos usan una versión de Windows, así que se me decante por hacerlo en Windows por esta razón, aunque no influya puesto que usuarios de Linux pueden ver el funcionamiento de mi aplicación web.

Se instaló la versión 1.7.0 del XAMPP para desarrollar la aplicación, que entre otros programas incluye el servidor web Apache 2.2.11, el SGBD MySQL 5.1.30 el lenguaje de programación PHP 5.2, además también incluye la aplicación web phpMyAdmin 3.1.1 que se usó para gestionar la base de datos creada para el proyecto.

Se utilizó el Macromedia Dreamweaver 8 para codificar la aplicación con PHP y XHTML, así como el Adobe Photoshop CS2 para la creación y modificación de las diferentes imágenes que aparecen en el juego.

El XAMPP incluye el Mercury Mail Transport System v4.52 que es un servidor de correo pero utilicé el MiniRealy 0.9.77 que hace prácticamente lo mismo pero que es más sencillo a la hora de configurarlo.

12.1. Navegador Web

El juego necesita de un navegador web para ser visualizado correctamente, un navegador web es un programa que permite visualizar la información que contiene una página web (ya esté está alojada en un servidor dentro de la World Wide Web o en uno local).

El navegador interpreta el código, XHTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

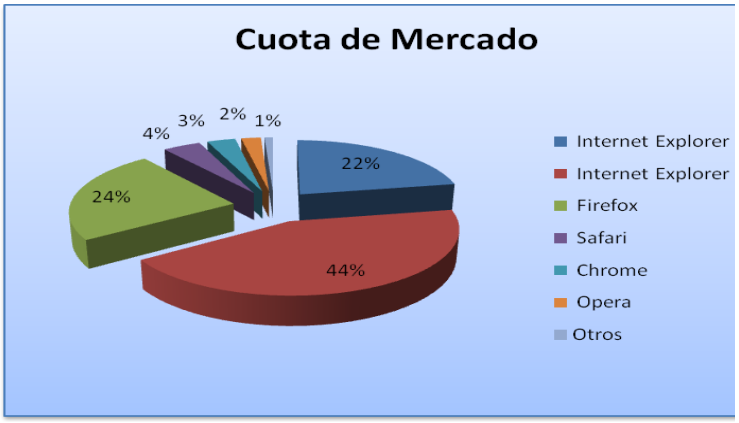


Imagen 12.1. Uso de los diferentes navegadores web en septiembre del 2009 [4]

Mi proyecto puede ser interpretado por cualquiera de los navegadores que existen en el mercado, no obstante es preferible el uso del Mozilla Firefox como predeterminado o del Internet Explorer. Esto se debe para evitar problemas de incompatibilidad en las vistas de las páginas.

13. Implementación

13.1. Estrategia para el desarrollo de la aplicación

El proyecto se ha desarrollado bajo la filosofía de la programación extrema[17], es el más destacado de los procesos ágiles de desarrollo de software.

Se ha intentado hacer un Desarrollo iterativo e incremental, es decir añadiendo pequeñas mejoras, unas tras otras, se han realizado pruebas continuamente para intentar corregir todos los errores antes de añadir nuevas funcionalidades, sobretodo el código ha intentado ser simple en la medida de lo posible.

13.2. Archivos creados y desarrollo de la aplicación

Cada archivo con extensión php se encarga de una parte de la aplicación, habrá un parte que se encargará de la gestión de usuarios, otra de todo lo relacionado con el combate, etc…

Todos estos archivos se encuentran en la raíz de la carpeta de la aplicación, luego tenemos otras carpetas que se encargan de otros aspectos: la carpeta imágenes tiene todas las imágenes que se usan en el juego, así mismo tiene tres subcarpetas que se encargan de organizar mejor las imágenes, luego encontramos la carpeta de scripts que contiene los archivos JavaScript que usa la aplicación y por último tenemos la carpeta de plantillas, aquí es donde se mostrará la página de cada parte de la aplicación después de llenarla con los datos necesarios.

Podemos ver la división más fácilmente de esta forma:

- **Carpeta raíz:**
  - administrador.php
  - ayuda.php
  - busqueda.php
  - combate.php
  - configuración.php
  - cookies.php
  - cuentas.php
  - foro.php
  - index.php
  - ingresar.php
  - miscelanea.php
  - zonas.php
  - **Carpeta Imágenes**
    - 18 elementos imagen .jpg

[17] La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Es decir, que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

        - **Carpeta armas**
            - 211 elementos imagen .gif
        - **Carpeta medicinas**
            - 10 elementos imagen .gif
        - **Carpeta rosa**
            - 9 elementos imagen .jpg
    - **Carpeta Scripts**
        - expande_armaduras.js
        - expande_habilidades.js
        - muestra_informacion.js
    - **Carpeta plantillas**
        - chat.php
        - columna_derecha.php
        - columna_derecha_combate.php
        - columna_derecha_links.php
        - columna_izquierda_mercenario.php
        - columna_izquierda_narco.php
        - columna_izquierda_timador.php
        - combate.php
        - derrota.php
        - edita_cuenta.php
        - error.php
        - ingresar.php
        - inicio.php
        - mapa.php
        - objeto.php
        - panel_administrador.php
        - principal.php
        - recuperar_contraseña.php
        - registro.php
        - verificar_cuenta.php
        - victoria.php
        - zonas.php

Básicamente primero se pide al usuario que ingrese al juego, todas las acciones pasan por el archivo index.php en la carpeta raíz, para entenderlo es como el organizador que luego deriva los procesos a otros archivos, cada vez que se accede al index.php se comprueba que el usuario esté conectado así como que el servidor este activo, estas son una de las muchas comprobaciones que se realizan.

El juego utiliza cookies[18], estas son importantes puesto que mediante ellas tenemos información del tiempo que lleva jugador conectado a la aplicación, en el caso de que el jugador este 10 minutos sin realizar ninguna acción es desconectado automáticamente para evitar que consuma unos recursos que no usa, dando así preferencia sobre los jugadores que los necesitan.

[18] Es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas. En ocasiones también se le llama "huella".

Se ha realizado una programación modular, haciendo que se puedan añadir funcionalidades en un futuro sin tener que modificar todo el código. Simplemente se crearía un nuevo archivo PHP con el código y se añadiría una línea al archivo index.php

Por ejemplo algo similar a esto:

```
//Muestra el ranking de los mejores 25 jugadores
elseif ($ir[0] == "ranking") { ranking(); }
```

El código de los archivos .js que son el JavaScript que utiliza la aplicación se ha obtenido de la página http://www.dynamicdrive.com/ al ser funcionalidades muy concretas y que ya estaban programadas se consideró que el tiempo que se debía dedicar a crear algo que ya estaba creado no estaba justificado.

Se usan varios estilos visuales: para el chat, para las páginas generales, para el mapa, para el menú de administrador y para la ayuda. Todo estos estilos están en la carpeta plantillas, dentro de la plantilla que se utiliza para visualizar la información, a excepción de las páginas generales que están en el archivo principal.php. Por ejemplo el estilo visual del chat estará en la carpeta plantillas en el archivo chat.php

Una de las características del juego es que se guardan siempre todos los aspectos, es decir, un jugador que esté en un combate puede decidir cerrar la pestaña del navegador si viera que está perdiendo el combate, pero esto no serviría de nada puesto que la información se guarda a medida que el jugador realiza acciones, entonces cuando el jugador se volviera a conectar volvería al punto exacto de la pelea donde la dejó.

13.2.1. Uso de los principales archivos .php

En este punto quiero explicar brevemente para que se usa cada archivo .php, las plantillas se comentarán en el siguiente apartado.

- **administrador.php:** Se encarga de mostrar todo lo relacionado con el panel de administrador.
- **ayuda.php:** Este archivo muestra todo lo relacionado con la página de ayuda, y sus estilos.
- **busqueda.php:** Todo lo relacionado con el movimiento del jugador, así como saber en qué lugar se encuentra en cada momento, y los eventos que ocurren en esos lugares.
- **combate.php:** Aquí se calculan todas las probabilidades y acciones de la batalla, así como las probabilidades de obtener equipo, los puntos que se reciben al pasar de nivel, etc. En general todo lo relacionado con el combate.
- **configuración.php:** Todo lo relacionado con la configuración para acceder a la base de datos
- **cookies.php:** Para la creación de cookies y los errores que se puedan producir en estas.

- **cuentas.php:** Todo lo relacionado con la gestión de las cuentas de usuarios, desde crear una nueva, hasta cambiar contraseñas, etc.

- **foro.php:** Se encarga de lo relacionado con el foro, insertar nuevos mensajes, responder los existentes, etc.

- **index.php:** Este es el programa principal, todas las acciones pasan por este archivo, desde la comprobación de que el usuario esté conectado hasta el chat, el mapa de la prisión, etc.

- **ingresar.php:** Lo relacionado con que el jugador se conecte al juego y salga sin problemas, maneja también las cookies.

- **miscelanea.php:** Este archivo contiene muchas funciones que se usan a menudo durante el programa.

- **zonas.php:** Aquí se calcula todo lo relacionado con las acciones disponible en las diferentes zonas. Como la compra/venta de armas y armaduras, compra de medicinas, de salvoconductos, etc.

13.3. Uso de plantillas

Las plantillas son una forma fácil y eficaz de mostrar la información, En mi proyecto el uso de plantillas está justificado puesto que uso la sintaxis Heredoc[19]. La diferencia entre Heredoc y otras formas más tradicionales de mandar cadenas de caracteres a la salida estándar es que con Heredoc se pueden transmitir grandes cadenas de texto con poco esfuerzo.

Las plantillas contienen campos con la siguiente estructura {{*nombre*}}; estos campos son la información que mostraré. Esta información se mostrará después de aplicar una función que se encuentra en miscelánea.php que lo que hace es remplazar esas estructuras que se encuentran en las plantillas por la información que previamente he guardado.

Para entenderlo de una forma más sencilla, la información la guardo en unas 'variables' que luego las muestro por pantalla.

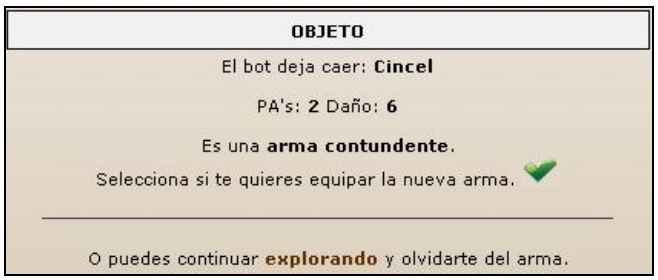


Imagen 13.1. Ejemplo de cómo se muestra la información después de remplazar los campos

[19] Heredoc (conjunción apocópica de Here-document o Documento-aquí en español) es una forma de representar cadenas en los lenguajes de programación

En el caso de que utilizar la plantilla objeto.php, podríamos observar cómo se muestra la información por pantalla, hay campos que no se han usado como {{*lugar*}} puesto que objeto.php es compartida para mostrar armas y armaduras.

Este trozo de código corresponde a la plantilla objeto.php, podemos observar claramente lo que he explicado en los párrafos anteriores.

```
<?php
$plantilla = <<<FIN
    <table width="100%">
    <tr><td class="titulo"><center>OBJETO</center></td></tr>
    <tr><td>
    <center>
        {{objeto}}
        {{lugar}}
        {{atributos}}<br />
        {{seleccion}}
        {{equipo}}<br />
        {{olvidarlo}}
    </center>
    </td></tr>
    </table>
FIN;
?>
```

Imagen 13.2. Código del archivo de plantillas objeto.php

13.4. Seguridad de la aplicación

La seguridad ha sido un elemento clave en el desarrollo del proyecto. Hay múltiples mecanismos de seguridad, si más no hay unos más importantes que otros, algunos serían:

- Cada usuario tiene un nombre de jugador y un nombre de cuenta, así pues los jugadores dentro del juego no conocen los nombres de cuenta de los otros usuarios, con esto se evita que un jugador malintencionado pueda robar la cuenta de otro jugador.
- Las contraseñas son encriptadas utilizando el algoritmo de codificación md5[20]
- Los campos de contraseña se muestran con ***** en vez de con letras.
- Se puede bloquear/banear a los jugadores que muestren un comportamiento inapropiado.
- Existe un filtro en el chat para evitar insultos o palabras malsonantes, así como el uso de código HTML.
- Al crear una cuenta se pide un correo electrónico que sea real para enviar un código de verificación, con esto se evita el uso de bots para crear cuentas.
- Se almacena información del último acceso del usuario así como desde que IP se conectó. Con esto se evita el uso de multicuentas y se tiene mejor control del jugador.

[20] MD5 (abreviatura de *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.

13.5. Tablas de la Base de Datos

En este apartado explicaré como están desarrolladas las diferentes tablas de la Base de Datos y que significan los campos de cada una de ellas.

Hay un total de 12 tablas, la mayoría de ellas son solo de consulta, es decir que las inserciones, actualizaciones y los borrados que se hagan serán muy de vez en cuando, puesto que es la estructura de donde se obtienen los datos y han sido testeados para que el juego este equilibrado en estos aspectos.

El prefijo pfc_ se ha usado en todas las tablas, tal vez no sería necesario pero de este modo tenemos una característica común a todas ellas (además de estar en la misma base de datos evidentemente).

- Las tablas principalmente de consulta son:
  - **pfc_armaduras**
  - **pfc_armas**
  - **pfc_atributosmerce**
  - **pfc_atributosnarco**
  - **pfc_atributostimador**
  - **pfc_bots**
  - **pfc_zonas**

- Existe una tabla que será de consulta y donde se pueden realizar actualizaciones esporádicas, pero que por su naturaleza no habrá ni inserciones ni borrados, tiene una mención aparte, esta tabla es la: **pfc_control**

- Existe otra categoría, por llamarlo así, que son tablas que si están enfocadas a recibir inserciones (mayoritariamente), no recibirán actualizaciones ni tampoco borrados a no ser que el administrador lo vea necesario, estas tablas son:

  - **pfc_anuncios**
  - **pfc_chat**
  - **pfc_foro**

- Por último tenemos la tabla de **pfc_jugadores** que será la que lleve la información actualizada de todos los detalles que pasan en el juego para ese jugador, esta tabla sí que recibirá inserciones (al crear un nuevo jugador), actualizaciones (cualquier acción del jugador que realice dentro del juego), borrados (al eliminar un jugador)

La estructura de las tablas siguen el patrón de primero indica el nombre el campo, luego el tipo (entero, char, etc.) y finalmente el valor predeterminado.

13.5.1. Estructura de la tabla armaduras

Esta tabla se encarga de almacenar todo lo referente a las armaduras, por ejemplo cuando un jugador quiera comprar una armadura se utilizará esta tabla para extraer la información.

Contiene 222 entradas y son las diferentes armas que nos podemos encontrar dentro del juego.

La estructura de la tabla es la siguiente:

id	smallint(3)	null
nombre	varchar(50)	null
armadura	smallint(4)	0
nivel	smallint(3)	0
tipo	smallint(3)	0
pvpventa	smallint(5)	0
pvpcompra	smallint(5)	0
atributos	varchar(25)	null

Descripción de los campos:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *nombre* es el nombre que tiene la armadura.
- El campo *armadura* es un valor numérico que indica los puntos de defensa que te otorga la armadura.
- El campo *nivel* es un valor que indica el nivel mínimo para poder usar esa armadura.
- El campo *tipo* es un valor que indica de que parte del cuerpo es la armadura, hay 6 posibles opciones: cabeza, torso, brazos, manos, piernas, pies.
- El campo *pvpventa* es un valor que nos indica el precio base que cuesta comprar la armadura, si está disponible, en una zona.
- El campo *pvpventa* es un valor que nos indica el precio base que cuesta vender la armadura, si está disponible, en una zona.
- El campo *atributos* es un de array de 6 valores separados por comas, que nos indica si la armadura en cuestión tiene algún bonificador a alguna habilidad del jugador.
  Ejemplo: 0,0,0,0,0,1 Significa que la armadura tiene un bonus de +1 en carisma.

13.5.2. Estructura de tabla armas

Esta tabla es similar a la tabla de armaduras pero se encarga de almacenar todo lo referente a las armas. Contiene 211 entradas que son las armas disponibles en el juego.

La estructura de la tabla es la siguiente:

id	smallint(3)	null
nombre	varchar(30)	null
daño	smallint(4)	0
pa	smallint(3)	0
tipo	smallint(2)	0
pvpventa	smallint(5)	0
pvpcompra	smallint(5)	0

Descripción de los campos:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *nombre* es el nombre que tiene el arma.
- El campo *daño* es un valor numérico que indica el valor base a tomar para calcular los puntos de golpe que recibe el bot.
- El campo *pa* es un valor numérico que indica los puntos de acción requeridos para usar esa arma.
- El campo *tipo* es un valor que indica de que naturaleza es el arma, hay tres tipos: Blancas, Contundentes y a distancia
- El campo *pvpventa* es un valor que nos indica el precio base que cuesta comprar el arma, si está disponible, en una zona.
- El campo *pvpventa* es un valor que nos indica el precio base que cuesta vender el arma, si está disponible, en una zona.

13.5.3. Estructura de las tablas de atributos

Estas tres tablas (atributosmerce, atributosnarco, atributostimador) tienen la misma estructura, se podrían haber fusionado en una sola, pero esto se descartó por varios motivos:

La tabla resultante era más complicada de entender a simple vista que haciendo las tres por separado, esto era debido, a que pasábamos de una tabla de 10 columnas a una de 30. Al haber usado tres tablas, las consultas son más rápidas, obteniendo así una mayor eficiencia.

Los nombres de los campos son los mismos para cada una de estas tres tablas.

La estructura de la tabla es la siguiente:

nivel	smallint(2)	null
experiencia	int(8)	0
vida	smallint(4)	0
fuerza	smallint(4)	0
constitucion	smallint(4)	0
agilidad	smallint(4)	0
destreza	smallint(4)	0
pericia	smallint(4)	0
carisma	smallint(4)	0
pa	smallint(5)	0

Estas tablas se utilizan como referencia cuando un jugador sube de nivel, indicando los valores de vida, fuerza, etc… que aumenta el jugador con el cambio de nivel.

Cada tabla contiene 50 entradas, correspondientes a los 50 niveles por los que un jugador puede pasar.

Descripción de los campos:

- El campo *nivel* es la clave primaria de esta tabla.
- El campo *experiencia* es el valor que necesita alcanzar el jugador para subir de nivel, estos valores no son lineales y son diferentes según el delito del jugador.
- El campo *vida* es un valor numérico que indica los puntos de vida que le aumentan al jugador en ese nivel.
- Los campos *fuerza*, *constitución*, *destreza*, *agilidad*, *pericia*, *carisma* y *pa* siguen la misma explicación que el campo vida, pero aumentando la habilidad en cuestión, si es que se aumenta.

13.5.4. Estructura de tabla de bots

Esta tabla se encarga de almacenar la información de los diferentes bots con los que el jugador puede encontrarse en el juego.

Contiene 160 entradas, se pueden añadir tantas entradas como diversidad de bots queramos que tenga el juego.

La estructura de la tabla es la siguiente:

id	smallint(3)	null
apodo	varchar(20)	null
nombre	varchar(20)	null
nivel	smallint(3)	0
vida	smallint(4)	0
daño	smallint(4)	0
armadura	smallint(4)	0
experiencia	smallint(4)	0
oro	smallint(4)	0

Descripción de los campos:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *apodo* es el nombre del clan al que pertenece el bot, la prisión está dividida en 10 clanes.
- El campo *nombre* es el rango que ocupa el bot en el clan, puede ser un 'novato' hasta un 'líder'.
- El campo *nivel* es un valor numérico que indica el nivel de ese bot.
- El campo *vida* es un valor numérico que indica el valor base de la vida que tendrá el bot. En el juego nos podemos encontrar con bots del mismo clan y del mismo rango con diferente vida.
- El campo *daño* es un valor numérico que nos indica el daño base que hace ese bot.
- El campo *armadura* es un valor numérico que nos indica la armadura que tiene ese bot.
- El campo *experiencia* es un valor numérico que nos indica la experiencia que nos otorga ese bot en el caso de que logremos acabar con él.
- El campo *oro* es un valor numérico que nos indica las monedas de oro que obtendríamos en el caso de acabar con él.

13.5.5. Estructura de tabla de las zonas

Esta tabla se encarga de almacenar la información correspondiente a las diferentes zonas que existen en la prisión. Contiene 10 entradas por defecto.

La estructura de la tabla es la siguiente:

id	smallint(3)	null
nombre	varchar(25)	null
longitud	smallint(4)	null
latitud	smallint(4)	null
viaje	smallint(6)	null
compra	smallint(6)	null
curacion	smallint(5)	0
nivel_curas	mediumint(3)	null
npcarmas	tinyint(3)	0
npcarmaduras	tinyint(3)	0
armas	varchar(75)	null
armaduras	varchar(75)	null

Descripción de los campos:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *nombre* es el nombre de la zona actual.
- El campo *longitud* indica la posición del eje vertical donde se encuentra la zona.
- El campo *latitud* indica la posición del eje horizontal donde se encuentra la zona.
- El campo *viaje* indica el coste base que se ha de pagar para poder viajar a esa zona 'al instante'.
- El campo *compra* indica el precio base que cuesta comprar el salvoconducto a esa zona de la prisión.
- El campo *curación* indica el precio base que cuesta recuperar los puntos de salud en esa zona.
- El campo *nivel_curas* indica el nivel a partir del cual no se le ofrecerá la posibilidad de recuperar puntos de salud en esa zona al jugador.
- El campo *npcarmas* indica si en esa zona hay o no hay un comerciante que vende armas.
- El campo *npcarmaduras* indica si en esa zona hay o no hay un comerciante que vende armaduras.
- El campo *armas* es un array de valores numéricos separados por comas, que nos indica las armas disponibles en esa zona.
- El campo *armaduras* es un array de valores numéricos separados por comas, que nos indica las *armaduras* disponibles en esa zona.

13.5.6. Estructura de tabla de los anuncios

Esta tabla se encarga de guardar los mensajes que el administrador escribe y que una vez dentro de las *zonas*, los jugadores pueden ver, es una especie de tablón de anuncios.

La estructura de la tabla es la siguiente:

Id	mediumint(8)	null
Mensaje	text	null
fecha	datetime	0000-00-00 00:00:00

Descripción de los campos:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *mensaje* contiene el mensaje en cuestión que se ha insertado.
- El campo *fecha* indica el día-mes-año junto con la hora-minuto-segundo que se insertó el mensaje, es útil puesto que solo se muestra el mensaje más reciente por pantalla.

13.5.7. Estructura de tabla del chat

Esta tabla se encarga de almacenar la información referente al chat que hay integrado dentro del juego. Por defecto no contendrá ninguna entrada.

La estructura de la tabla es la siguiente:

id	int(10)	null
mensaje	varchar(150)	null
autor	varchar(20)	null
fecha	time	00:00:00

Descripción de los campos:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *mensaje* es el contenido del mensaje que ha escrito un jugador. Este mensaje pasa por un filtro, para evitar palabras mal sonantes, escritura XHTML, etc.
- El campo *autor* es el nombre del jugador que escribe el mensaje.
- El campo *fecha* indica la hora-minuto-segundo que se escribió el mensaje.

13.5.8. Estructura de tabla de control

Esta tabla contiene información esencial para el correcto desarrollo del juego. De una forma práctica se pueden habilitar o deshabilitar contenidos dentro del juego que de otra manera serían más costosos de realizar.

Esta tabla solo contiene y ha de contener una única entrada.

La estructura de la tabla es la siguiente:

id	tinyint(3)	null
juego	varchar(50)	null
mantenimiento	tinyint(3)	0
verifica_correo	tinyint(3)	0
delito1	varchar(20)	null
delito2	varchar(20)	null
delito3	varchar(20)	null
correo_administrador	varchar(100)	null
pagina_juego	varchar(180)	null
foro	tinyint(3)	0
foro_direccion	varchar(150)	null
noticias	tinyint(3)	0
chat	tinyint(3)	0
jugadores_online	tinyint(3)	0
comprimir_juego	tinyint(3)	0
tamaño_mapa	smallint(5)	0

Descripción de los campos:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *juego* es el nombre del juego en sí, esto es útil para mostrar mensajes de bienvenida, en las pantallas de registro, etc.
- El campo *mantenimiento* cierra el juego haciendo que los jugadores vean un mensaje de error debido a que el juego no está disponible temporalmente. El valor 1 indica abierto, el valor 0 indica cerrado.
- El campo *verifica_correo* Si esta activo, cada nueva cuenta que se cree se enviara un correo de verificación al dueño de la misma. Sin introducir ese código correctamente no se podrá activar la cuenta y no se podrá jugar.
- El campo *delito1, delito2* y *delito3* muestran los nombres de los delitos dentro del juego, en caso de querer poner otros nombres seria en este sitio donde se tendrían que cambiar.
- El campo *correo_administrador e*s necesario para mostrar en la pantalla de registro en caso de que haya algún problema y cuando se envía el correo a los jugadores.
- El campo *pagina_juego* muestra mostrar la dirección completa en donde está instalado el juego. Es necesario que sea correcta y no se deje en blanco, puesto que el correo que se envía al registrarse en el juego necesitara esta dirección.

- El campo *foro* sirve para mostrar o no el link del foro. Si es un 0 quita el link, si es un 1 se usa el foro que hay dentro del juego, si es un 2 el link usará la dirección del campo foro_direccion.
- El campo foro_direccion solo es necesario si i el campo *foro* es un 2 (foro externo), en ese caso se necesita una dirección correcta para mostrarla en el link del foro.
- El campo *noticias* si está activo, muestra la última noticia escrita por el administrador. El valor 1 indica activo, el valor 0 indica desactivado.
- El campo *chat* si está activo muestra el chat que está integrado en el juego. El valor 1 indica activo, el valor 0 indica desactivado.
- El campo *jugadores_online* si está activo, muestra los jugadores que están activos en el juego. Se consideran inactivos cuando están 1 minuto sin realizar ninguna acción. El valor 1 indica activo, el valor 0 indica desactivado.
- El campo *comprimir_juego* si está activo y el servidor lo soporta, comprime las páginas haciendo que el ancho de banda se reduzca considerablemente. El valor 1 indica activo, el valor 0 indica desactivado.
- El campo *tamaño_mapa* El mapa es cuadrado y actualmente sin paredes, el valor que se indica en el recuadro es la mitad del tamaño total de un lado. Si se quiere incrementar este valor se han de añadir más bots a la Base de Datos, si se quiere reducir este valor no se usarán todos los bots que hay. Hay que tener en cuenta que el mapa que se muestra al explorar la prisión no mostraría la posición correcta en caso de que se cambie este valor, con lo que se tendría que corregir. Ha de ser múltiplo de 5.

13.5.9. Estructura de tabla del foro

Esta tabla guarda información a cerca de todo lo relacionado con el foro del juego. Por defecto no hay ningún mensaje.

La estructura de la tabla es la siguiente:

id	mediumint(6)	null
tema	varchar(50)	null
mensaje	varchar(150)	null
autor	varchar(20)	null
padre	mediumint(6)	0
respuestas	mediumint(6)	0
fecha	datetime	0000-00-00 00:00:00
ultima_entrada	datetime	0000-00-00 00:00:00

Descripción de los campos:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *tema* es el título del los diferentes hilos que tendrá el foro.
- El campo *mensaje* es el contenido del mensaje que ha escrito un jugador.
- El campo *autor* es el nombre del jugador que escribe el mensaje.
- El campo *padre* es el *id* del mensaje a que hace referencia el mensaje actual, es decir, si el padre es 0 ese hilo no tiene respuestas, si es un valor diferente a 0, ese mensaje pende el hilo con ese *id*.
- El campo *respuestas* indica el número de respuestas, si tiene, un mensaje.
- El campo *fecha* indica el año-mes-día y la hora-minuto-segundo que se escribió el mensaje.
- El campo *ultima_entrada* indica el año-mes-día y la hora-minuto-segundo que tiene el último mensaje que se escribió en ese hilo.

13.5.10. Estructura de tabla de los jugadores

Esta tabla es la tabla principal del juego por importancia y por relevancia, no está normalizada[21] y esto es debido a una mejor eficiencia a la hora de hacer las querys. Esta tabla podría estar perfectamente dividida en 6 o más tablas (tabla de combate, de lista de armas, de lista de zonas, de lista de armaduras, de cuenta, de jugadores).

Se decidió hacerlo de esta manera por temas de eficiencia, no es lo mismo lanzar una query a una tabla (con bastantes columnas, eso sí) que a 6 tablas, puesto que cada consulta que realizo a esta tabla el tiempo es 1/6 parte de lo que tardaría en hacerlo a esas 6 tablas.

Dicho esto, esta tabla se encarga de gestionar a los jugadores, podemos sub-dividirla en:

- Gestión de la cuenta del jugador.
- Gestión los atributos referentes al jugador.
- Gestión las habilidades del jugador.
- Posición actual del jugador.
- Gestión las armas.
- Gestión de las diferentes zonas.
- Gestión de las diferentes armaduras.
- Gestión referente a los combates

La estructura de la tabla es la siguiente:

id	smallint(4)	null
nombre	varchar(20)	null
contrasena	varchar(32)	null
email	varchar(30)	null
verificado	varchar(10)	0
registro	datetime	0000-00-00 00:00:00
ip_registro	varchar(15)	000.000.000.000
ultimo_log	datetime	0000-00-00 00:00:00
ip_login	varchar(15)	000.000.000.000
permisos	smallint(3)	0
jugador	varchar(20)	null
nivel	smallint(3)	1
clase	smallint(3)	0
experiencia	mediumint(10)	0
vida	smallint(5)	null
armadura	smallint(4)	5
pa	smallint(3)	null
fuerza	smallint(4)	null
constitucion	smallint(4)	null

[21] El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional. Las bases de datos relacionales se normalizan para: evitar la redundancia de los datos, evitar problemas de actualización de los datos en las tablas, proteger la integridad de los datos

agilidad	smallint(4)	null
destreza	smallint(4)	null
pericia	smallint(4)	null
carisma	smallint(4)	null
oro	mediumint(10)	350
armas	varchar(85)	null
lista_lugares	varchar(30)	1
vida_actual	smallint(5)	null
pa_actual	smallint(3)	null
accion	varchar(25)	Bienvenida
latitud	smallint(4)	0
longitud	smallint(4)	0
turno	smallint(3)	1
lucha	smallint(5)	0
bot_id	smallint(5)	0
bot_vida	smallint(6)	0
bot_vida_act	smallint(6)	0
bot_aturdido	tinyint(4)	0
bot_herida	mediumint(4)	0
id_cabeza	smallint(4)	1
id_brazos	smallint(4)	38
id_manos	smallint(4)	75
id_torso	smallint(4)	112
id_piernas	smallint(4)	149
id_pies	smallint(4)	186
objeto	smallint(4)	0

Sub-división perteneciente a la gestión de la cuenta:

- El campo *Id* es la clave primaria de esta tabla.
- El campo *nombre* indica el nombre de la cuenta del jugador, no hay que confundirlo con el nombre que tiene el jugador dentro del juego, puesto que son cosas diferentes. Este campo se creó para aumentar la seguridad global del juego.
- El campo *contraseña* este campo contiene la contraseña encriptada del jugador, uso la función md5 del PHP.
- El campo *email* indica el correo electrónico del jugador.
- El campo *verificado* indica si la cuenta asociada al jugador esta verificada (1), o no esta verificada, en este caso se mostrará el código de verificación
- El campo *registro* indica el año-mes-día y la hora-minuto-segundo que se creó la cuenta.
- El campo *ip_registro* indica desde que dirección IP se creó la cuenta.
- El campo *ultimo_log* indica el año-mes-día y la hora-minuto-segundo que se realizó la última acción dentro del juego.
- El campo *ip_login* indica desde que dirección IP se accedió a la cuenta por última vez. Este campo aporta seguridad extra al juego, puesto que es fácilmente detectable si un jugador tiene más de una cuenta.

- El campo *permisos* Indica el tipo de jugador que es: (0) si es un usuario normal. (1) Administrador si tiene permisos de administrador. (2) Baneado para que el jugador no pueda entrar al juego.
- El campo *clase* indica el delito del jugador, 1 si es mercenario, 2 si es narcotraficante, 3 si es timador.

Sub-división perteneciente a los atributos del jugador:

- El campo *jugador* este campo indica el nombre del jugador dentro del juego.
- El campo *nivel* indica el nivel actual del jugador.
- El campo *experiencia* indica la experiencia actual del jugador.
- El campo *vida* indica la vida máxima del jugador.
- El campo *vida_actual* indica la vida que tiene actualmente el jugador.
- El campo *pa* indica los puntos de acción máximos que el jugador puede tener en este nivel.
- El campo *pa_actual* indica los puntos de acción que el jugador tiene actualmente.

Sub-división perteneciente a las habilidades del jugador:

- El campo *fuerza* indica la fuerza actual del jugador.
- El campo *constitución* indica la constitución actual del jugador.
- El campo *agilidad* indica la agilidad actual del jugador.
- El campo *destreza* indica la destreza actual del jugador.
- El campo *pericia* indica la pericia actual del jugador.
- El campo *carisma* indica la carisma actual del jugador.
- El campo *oro* indica las monedas que tiene el jugador en su poder, es el dinero propiamente dicho del juego.

Sub-división perteneciente a la posición del jugador:

- El campo *acción* muestra información de lo que está haciendo el jugador en ese momento.
- El campo *latitud* indica la posición del eje horizontal donde se encuentra el jugador.
- El campo *longitud* indica la posición del eje vertical donde se encuentra el jugador.

Sub-división perteneciente a la gestión de las armas:

- El campo *armas* indica las armas disponibles del jugador, estos valores hacen referencia a la tabla de armas.

Sub-división perteneciente a la gestión de las zonas explorables:

- El campo *lista_lugares* indica los lugares o zonas conocidos por el jugador, estos valores hacen referencia a la tabla de zonas.

Sub-división perteneciente a la gestión de las armaduras:

- El campo *armadura* indica la armadura actual del jugador, este campo va en función de las armaduras que tenga equipadas el jugador.

- El campo *id_cabeza* indica el id de la protección de cabeza que tiene el jugador, este valor hace referencia a la tabla armaduras.
- El campo *id_brazos* indica el id de la protección de brazos que tiene el jugador, este valor hace referencia a la tabla armaduras.
- El campo *id_manos* indica el id de la protección de manos que tiene el jugador, este valor hace referencia a la tabla armaduras.
- El campo *id_torso* indica el id de la protección de torso que tiene el jugador, este valor hace referencia a la tabla armaduras.
- El campo *id_piernas* indica el id de la protección de piernas que tiene el jugador, este valor hace referencia a la tabla armaduras.
- El campo *id_pies* indica el id de la protección de pies que tiene el jugador, este valor hace referencia a la tabla armaduras.

Sub-división perteneciente a la gestión del combate:

- El campo *lucha* indica si el jugador está en combate o no. Este campo seria prescindible en cierta manera, puesto que se podría hacer la comprobación si el campo bot_id esta vacio o no, pero añade un poco más de seguridad.
- El campo *turno* indica el turno de combate del jugador, puede ser un 1 o un 2, puesto que el combate se divide en rondas y cada ronda tiene dos turnos tanto para el jugador como para el bot.
- El campo *bot_id* indica el *id* del bot que está en combate con el jugador, este campo hace referencia a la tabla de bots. Será un 0 si el jugador no está en combate.
- El campo *bot_vida* indica la vida máxima del bot que tendrá en ese combate.
- El campo *bot_vida_act* indica la vida actual del bot en ese combate.
- El campo *bot_aturdido* indica si el bot está aturdido (1), o no lo está (0).
- El campo *bot_herida* indica si el bot tiene una herida sangrante (1), o no la tiene (0).
- El campo *objeto* indica el id del objeto que ha dejado caer el bot en el combate, este valor hace referencia a la tabla de armaduras o de armas, dependiendo si el objeto es una armadura o un arma.

14. Testeo de la aplicación

Esta parte es muy importante para el correcto desarrollo del juego, muestra los diferentes test y juegos de prueba que la aplicación ha pasado.

14.1. Test realizados sobre la aplicación

Durante todo el desarrollo del proyecto se han ido haciendo sucesivos test a medida que se completaban las diferentes partes, una vez se tuvieron todas las partes los test continuaron haciéndose.

He de dar una mención especial a los amigos y compañeros que me han ayudado haciendo de beta testers[22] ya que gracias a ellos he encontrado fallos que se me habían pasado por alto.

En general esta es la lista de todos los test que se han realizado sobre el juego:

- Test sobre las creaciones de cuentas.
- Test sobre el cambio de contraseña.
- Test sobre la verificación de cuentas.
- Test sobre el correcto acceso al juego.
- Test de informe de errores.
- Test sobre el combate.
    - El correcto funcionamiento de las peleas.
    - El correcto incremento de experiencia, oro, al ganar la pelea o subir de nivel.
    - El correcto equipamiento del objeto caído en una pelea.
- Test sobre el correcto funcionamiento del juego en las zonas.
    - Comprar, vender, equipar armas.
    - Comprar armaduras, salvoconductos, medicinas.
    - Ir correctamente a las zonas.
- Test sobre el movimiento de personaje sobre el mapa.
- Test sobre el menú de administrador.
    - La correcta inserción de mensajes.
    - La correcta edición de cuentas.
    - El cambio en la configuración.
- Test sobre foro y el chat.
    - Creación de temas, inserción de mensajes, aplicado de filtros.
- Test sobre la correcta muestra de habilidades, experiencia, puntos de acción...
- Test sobre la seguridad de la aplicación.
- En general test sobre la aplicación al completo, pudiendo haber alguna pequeña omisión en los puntos de arriba.
- Test sobre el acceso al juego a terceras personas creando cuentas y probándolo íntegramente.
- Test sobre el tiempo de respuesta del juego (explicado en el siguiente punto).

[22] Los beta testers usan sus conocimientos informáticos y su tiempo para detectar errores en el software y así poder informar de éstos para que los desarrolladores los corrijan.

14.2. Test sobre los tiempos de respuesta

La capacidad de la base de datos es finita y el tiempo de respuesta de la aplicación crece en función de los usuarios que acceden a ella, en este punto se quiere poner valores numéricos a estos aspectos.

Para calcular el tiempo de respuesta de la aplicación se creó la función *tiempo()*

```
function tiempo(){
    list($picosegundos, $segundos) = explode(" ",microtime());
    return ((float)$picosegundos + (float)$segundos);
}
```

Lo que realizamos después es una simple resta, puesto que tendremos el valor del tiempo inicial, cuando cargó la página llamo a *tiempo()*, y el valor final del tiempo actual, haciendo la resta nos da un valor numérico en segundos que nosotros lo acotamos en 4 decimales:

```
"tiempo_querys"=>round(tiempo() - $tiempoinicial, 4),
```

El resultado sería algo parecido a esto en la pantalla inicial de la aplicación, este valor como veremos a continuación variará en función del número de jugadores y de la potencia del servidor.

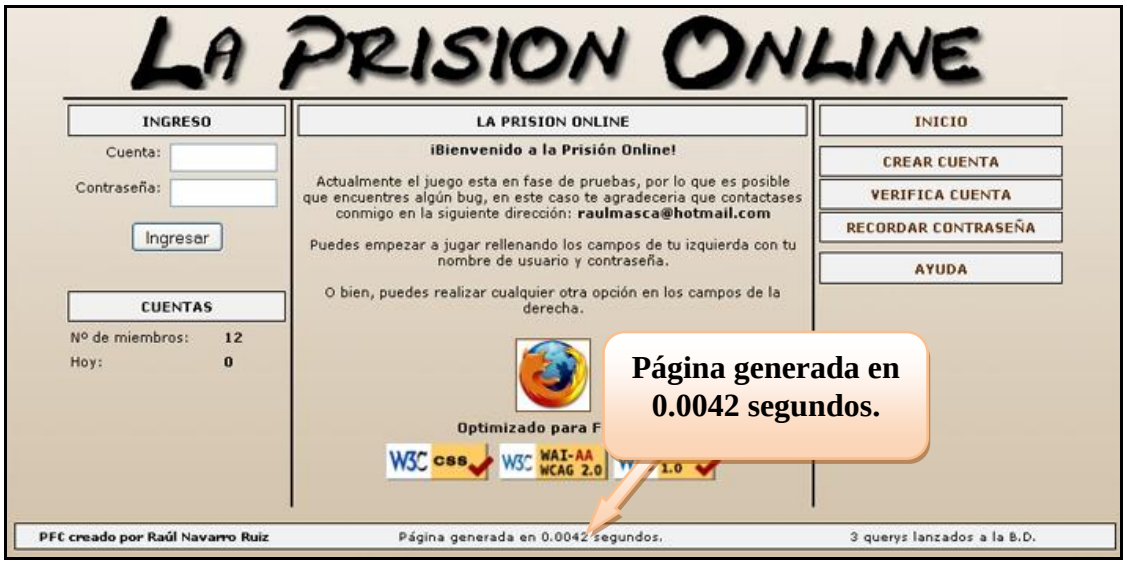


Imagen 14.1. Imagen que muestra el tiempo que ha tardado en cargar la página

En el peor de los casos un jugador lanza 21 querys a la base de datos al cargar una página, este caso es cuando un jugador está en combate con un bot, el tiempo medio de respuesta es de 0.0174 segundos (este valor se ha conseguido después de hacer 10 mediciones y coger la media aritmética).

Podríamos especular pues que si tenemos a 100 personas jugando al mismo tiempo y que todas estén en un combate el tiempo medio de respuesta seria 1.74 segundos, pero al no ser lineal sino que cada jugador aporta algo más de retraso, se podría llegar a afirmar que 100 personas jugando a la vez tendrían un retraso de 2 segundos al cargar la página.

En el mejor de los casos el tiempo de respuesta es 0 esto significa que la persona no está realizando ninguna acción ni está lanzando querys al servidor, pero yo estoy buscando el peor de los casos cuando la persona está mandando peticiones al servidor, puesto que si no realiza ninguna acción a los 10 minutos se desconectará automáticamente.

Entonces en el mejor de los casos el usuario lanza 3 querys al servidor (pantalla inicial), con un tiempo de 0.0036 segundos, en el caso de tener 100 personas este tiempo seria de 0.36 segundos, redondeamos debido al retraso acumulado y nos daría un retraso de 0.5 segundos al cargar la página.

Con estos datos podemos hacer un pronóstico del tiempo medio de respuesta para 100 usuarios y sería de 1.25 segundos aproximadamente.

Se quisieron verificar estos datos teóricos con una prueba realizada con varias personas conectada a la misma vez, para ello se realizaron dos pruebas sobre dos ordenadores diferentes.

14.2.1. Prueba I

La primera prueba se realizó sobre un ordenador Intel Quad Core a 2.40Mhz, 2Gb de RAM y conexión de hasta 20Mb con Jazztel, con una velocidad de subida de 765Kb/seg. [5]

Imagen 14.2. Medición de la velocidad de subida y de bajada del PC donde se realizaron las pruebas

La prueba se realizó el 16 de Octubre del 2009 y duró 10 minutos, se formo un grupo de voluntarios entre amigos y conocidos que llego a las 25 personas.

La prueba consistió en dos fases, la primera se pidió a los jugadores que entraran en un combate y que después de usar su primer turno mandaran los resultados del tiempo que aparecía en la página.

Los resultados de la primera parte de esta prueba fueron los siguientes:

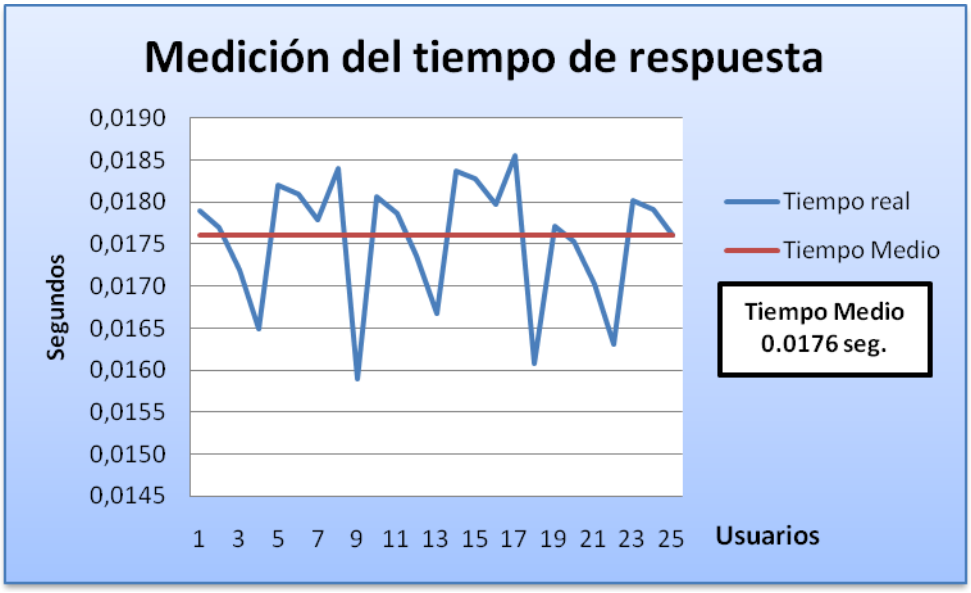


Imagen 14.3 Tiempo real y tiempo medio de respuesta para cada usuario

El tiempo acumulado muestra una gráfica del tiempo que tuvieron que esperar los usuarios para visualizar la página

Hay que tener en cuenta que no es lo mismo el tiempo que se tarda en generar la página (fue el tiempo que enviaron los jugadores), al tiempo acumulado, que sería el tiempo real que tardaron en visualizar la página.

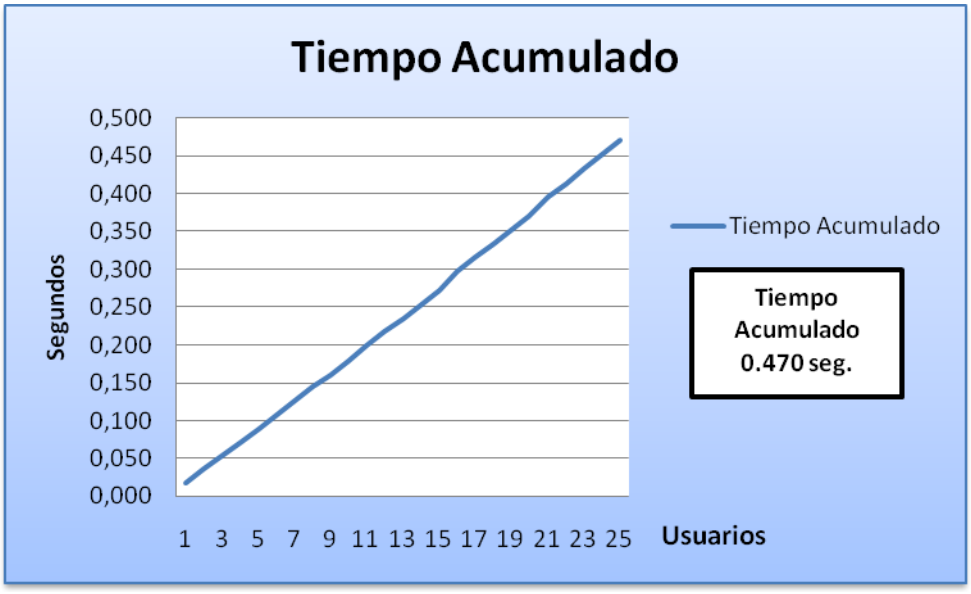


Imagen 14.4. Tiempo acumulado de retardo global para 25 usuarios

Estos tiempos son en el peor de los casos, puesto que el realizar un combate es lo que más recursos consumen al servidor en forma de querys.

Finalmente hubo una ligera variación entre el tiempo estimado y el tiempo real, en la siguiente gráfica podemos observar estas diferencias.

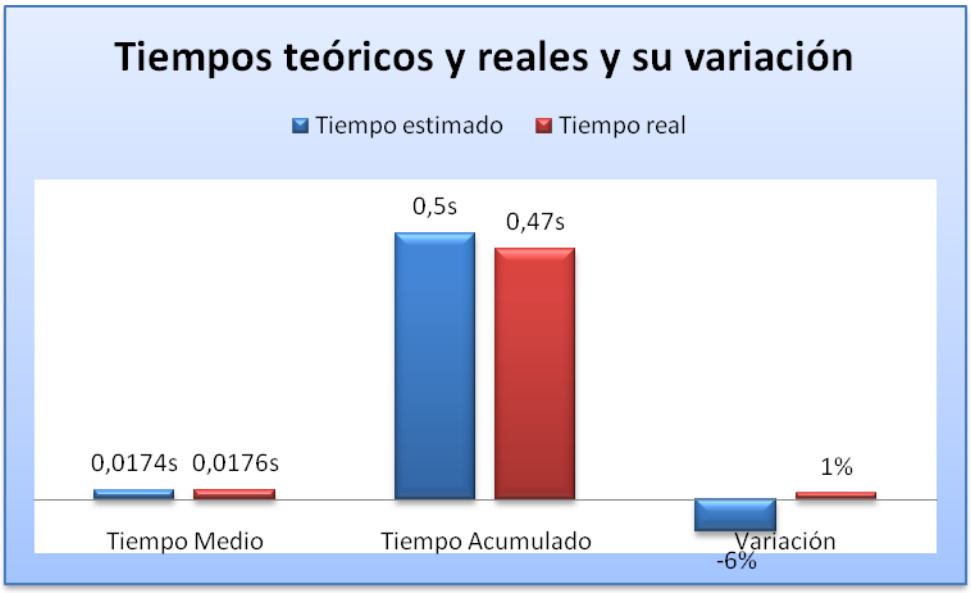


Imagen 14.5. Gráfica que muestra los resultados de la prueba y sus variaciones

La segunda parte de la prueba consistió en dar libertad a los jugadores para realizar las acciones que quisieran, en un momento dado se les pidió que cesaran esa actividad y que enviaran los tiempos que habían necesitado para generar la página, con esto se obtendría un tiempo medio general, puesto que ya no estaríamos realizando las pruebas en el peor de los casos.

El resultado fue el siguiente:

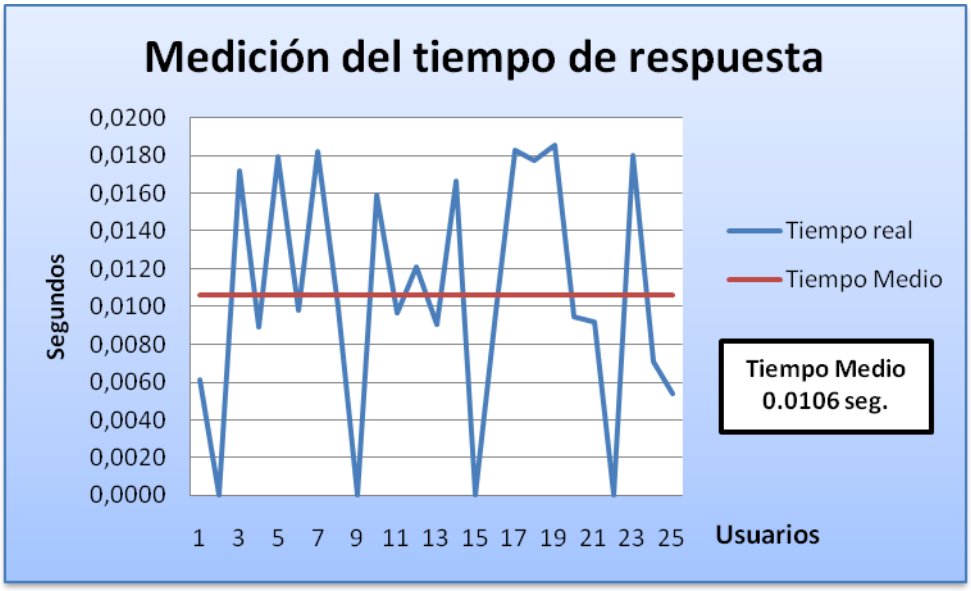


Imagen 14.6. Tiempo real y tiempo medio de respuesta para cada usuario

Los picos con un 0.0000 significan que el usuario no había lanzado ningún query al servidor, esto fue debido a por ejemplo que un jugador estuviera consultando la ayuda, o bien que estuviera más de 1 minuto sin realizar ninguna acción, o que estuviera

mirando otros aspectos que no necesitaran el envío de querys al servidor (por ejemplo el filtrado de la consulta de armaduras, la consulta de sus habilidades, etc..)

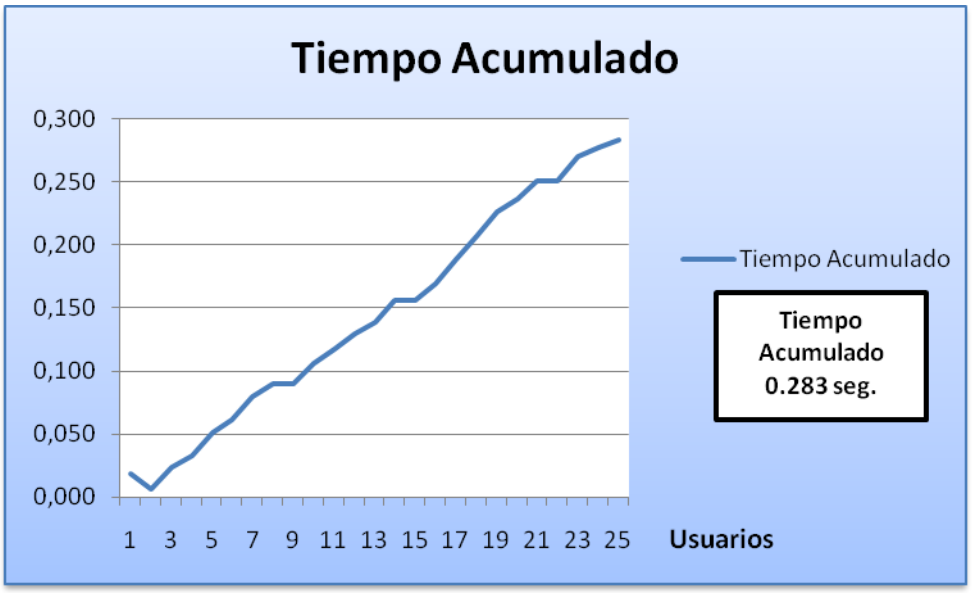


Imagen 14.7. Tiempo acumulado de retardo global para 25 usuarios

Esta gráfica muestra el retardo global medio que tuvieron que esperar los jugadores para visualizar la página.

En general los resultados fueron mejor de lo esperado puesto que aunque el tiempo medio fuera 1% superior a lo esperado en el peor de los casos, la variación del tiempo acumulado fue un 6% inferior en el peor de los casos.

14.2.2. Prueba II

La segunda prueba se realizó sobre un ordenador portátil AMD Athlon de doble núcleo a 1.7Ghz, 2Gb de RAM y conexión WiFi sobre un ordenador con conexión de hasta 20Mb de Jazztel. La velocidad de subida fueron 811Kb/s. [6]

Imagen 14.8. Medición de la velocidad de subida y de bajada del PC donde se realizaron las pruebas

La prueba se realizó el 16 de Octubre del 2009 y duró 10 minutos, el mismo grupo de voluntarios que participaron en la primera prueba participaron en la segunda.

Puesto que en esta prueba se quería comprobar cómo afectaba el tener una máquina menos potente a la hora de utilizarla como servidor solo se realizó una prueba, que fue

el dar libertad a los jugadores para realizar las acciones que quisieran, en un momento dado se les pidió que cesaran esa actividad y que enviaran los tiempos que habían necesitado para generar la página.

Los resultados de esta prueba fueron los siguientes:

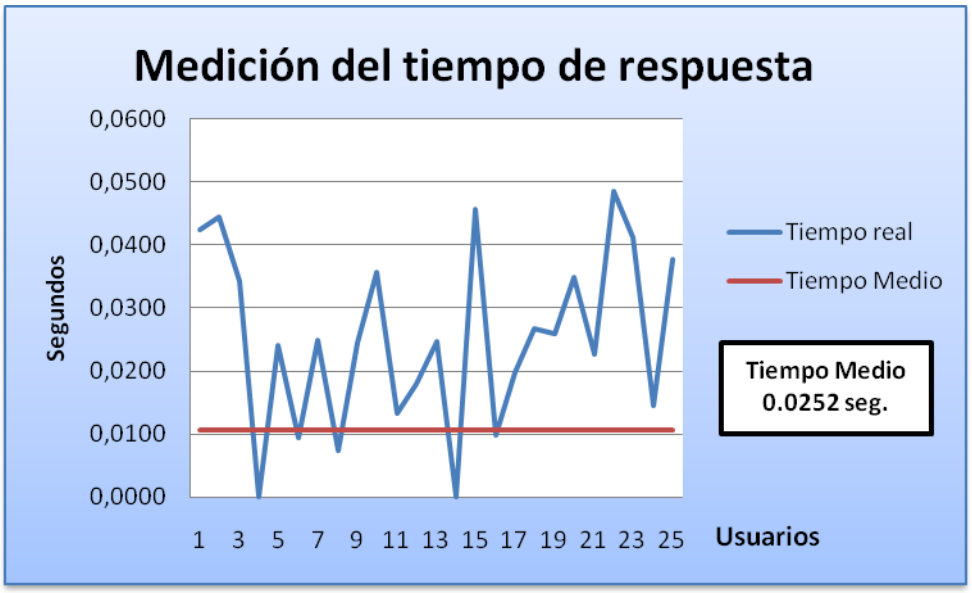


Imagen 14.9. Tiempo real y tiempo medio de respuesta para cada usuario

En esta gráfica ya podemos corroborar lo que nos pensabamos al principio, que el retardo seria muy superior al obtenido con la otra máquina.

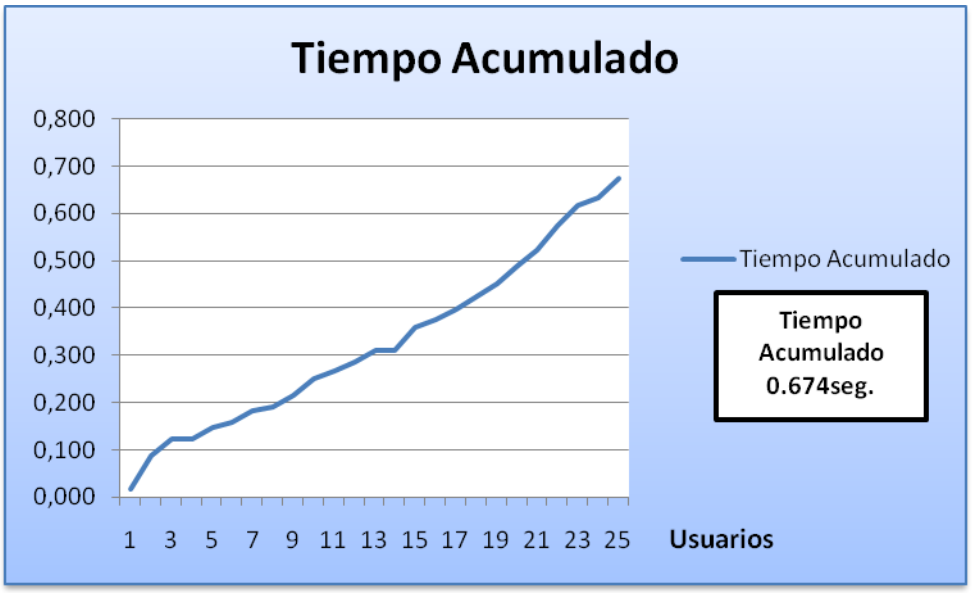


Imagen 14.10. Tiempo acumulado de retardo global

En esta gráfica podemos observar que el tiempo medio de respuesta en cargar la página fue de 0.674 segundos para 25 usuarios.

Se esperaban unos resultados más negativos que los obtenidos, es cierto que el tiempo medio de respuesta es un 237% superior al obtenido en la primera prueba y que el

tiempo acumulado fue un 238% superior, no obstante para el tipo de máquina considero que fueron unos buenos resultados.

14.2.3. Conclusiones de las pruebas

- Es tan importante un buen diseño del funcionamiento del juego como una buena máquina que actúe como servidor.

- En esta prueba el ancho de banda no afecto puesto que transmitía sobre unos 800kb/s de media, suficiente para cubrir las necesidades de las 25 personas.

- Un tiempo medio de generación de la página de 0.0106 segundos, y un retardo medio de 0.252 segundos es más que aceptable para ese grupo de personas.

- Con estos datos podemos afirmar que 100 personas tendrían un retardo medio de 1.132 segundos, se estimo este retardo en 1.25 segundos por lo tanto la variación es de un 12% inferior a lo estimado.

- El juego se diseño para soportar un número finito de personas, considero que un retardo de 2 o más segundos para 100 personas hubiera sido un fracaso.

Parte V: Mantenimiento

15. Manual de Usuario

Uno de los objetivos del proyecto es que el juego fuera sencillo e intuitivo de entender para alguien que no estuviera familiarizado con esta temática.

Así pues, en el manual de usuario se explicaran las diferentes pantallas y acciones que el jugador puede desarrollar, este manual pretende ser unos primeros pasos para alguien que quiera jugar al juego por primera vez.

Se completa con la ayuda en línea existente dentro del juego y con el punto 9.2 que es el de diseño de los delitos.

15.1. Pantalla inicial de la aplicación

Esta es la pantalla inicial con la que se encuentra cualquier usuario que acceda a la aplicación. Aquí se puede acceder al juego poniendo tu nombre de cuenta y contraseña en los campos de la izquierda o realizar cualquier otra acción en el menú de la derecha.

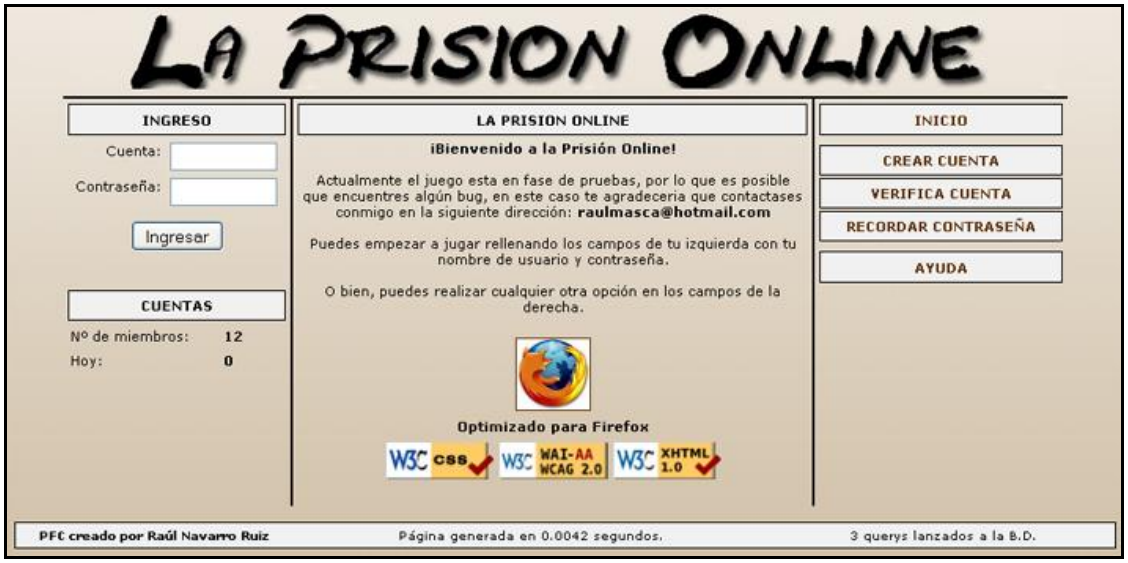


Imagen 15.1. Página inicial de la aplicación, con los diferentes campos de información

15.2. Pantalla de Bienvenida

La primera vez que el jugador entra dentro del juego verá una pantalla similar a la siguiente solo cambiara en función del delito seleccionado.

En esta pantalla se da la bienvenida al juego y se advierte al jugador que tiene que ir a comprar un arma para poder realizar los combates, puesto que por defecto no tiene ninguna.

Podemos ver los diferentes campos de información que muestra la pantalla, explicados en los rectángulos de texto.

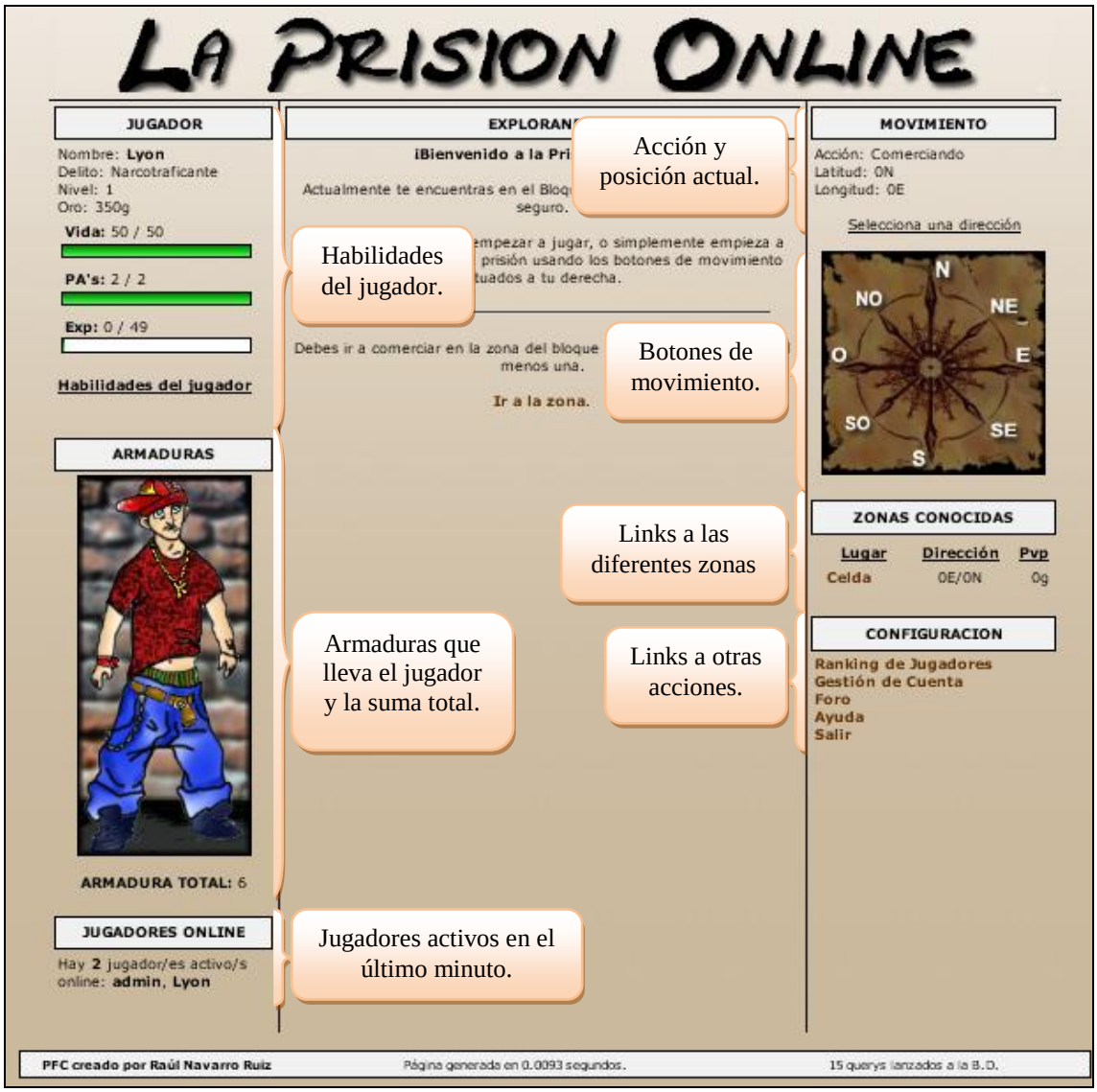


Imagen 15.2. Pantalla de bienvenida una vez entra el usuario al juego por primera vez

El juego no es lineal, esto significa que el jugador no tiene porque ir a comprar el arma ahora mismo, puede consultar la ayuda, puede hablar por el chat o puede ir al foro interno a leer los mensajes que puedan haber dejado otros jugadores.

15.3. Cajas de Información y expansión de habilidades

Hay ciertos lugares donde se pude clicar para expandir y mostrar más información de la que hay visible a simple vista, estos lugares son reconocibles puesto que cuando se pasa el ratón por encima de ellos el puntero cambia su forma a un dedo.

Podemos ver este ejemplo en las habilidades del jugador:

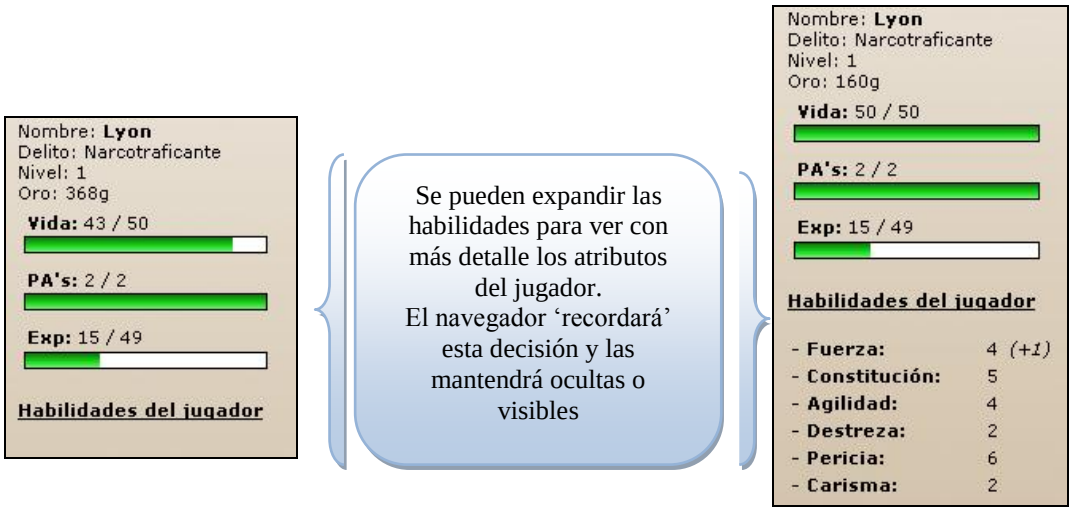


Imagen 15.3. Imagen que muestra como las habilidades del jugador se expanden y contraen

Si un atributo tiene un *(+número)* al lado esto es debido a que una parte de armadura otorga un bonus a ese atributo.

Además de estos en el juego dependiendo por donde se mueva el ratón se muestran cajas con información justo debajo del puntero del mouse.

Imagen 15.4. Pantalla de información sobre las habilidades

Por ejemplo en esta imagen se puede observar cómo se muestra información en una caja, al pasar el puntero por la barra de Exp esta muestra información de lo que es y el tanto por ciento de experiencia conseguida hasta el momento.

Imagen 15.5. Captura de la ventana de información emergente del juego

En este otro ejemplo se puede observar como dependiendo de la zona por donde se pase el ratón en la figura del jugador se mostrará información sobre la armadura que lleva actualmente, en este caso, se miro que armadura llevaba puesta el jugador en el torso.

Hay 6 zonas donde se puede mirar esta información: En la cabeza, en el torso, en los dos brazos, en las dos manos, en las dos piernas y en los dos pies.

La suma de protecciones de las diferentes partes indica la armadura total que tiene actualmente el personaje.

15.4. Acceso a las diferentes zonas

Un jugador tiene que ir a las zonas para poder comerciar y curarse las heridas, así como para poder usar el chat que hay integrado en el juego y ver los mensajes del administrador.

En esta imagen se puede ver el jugador dentro de la zona de celdas.

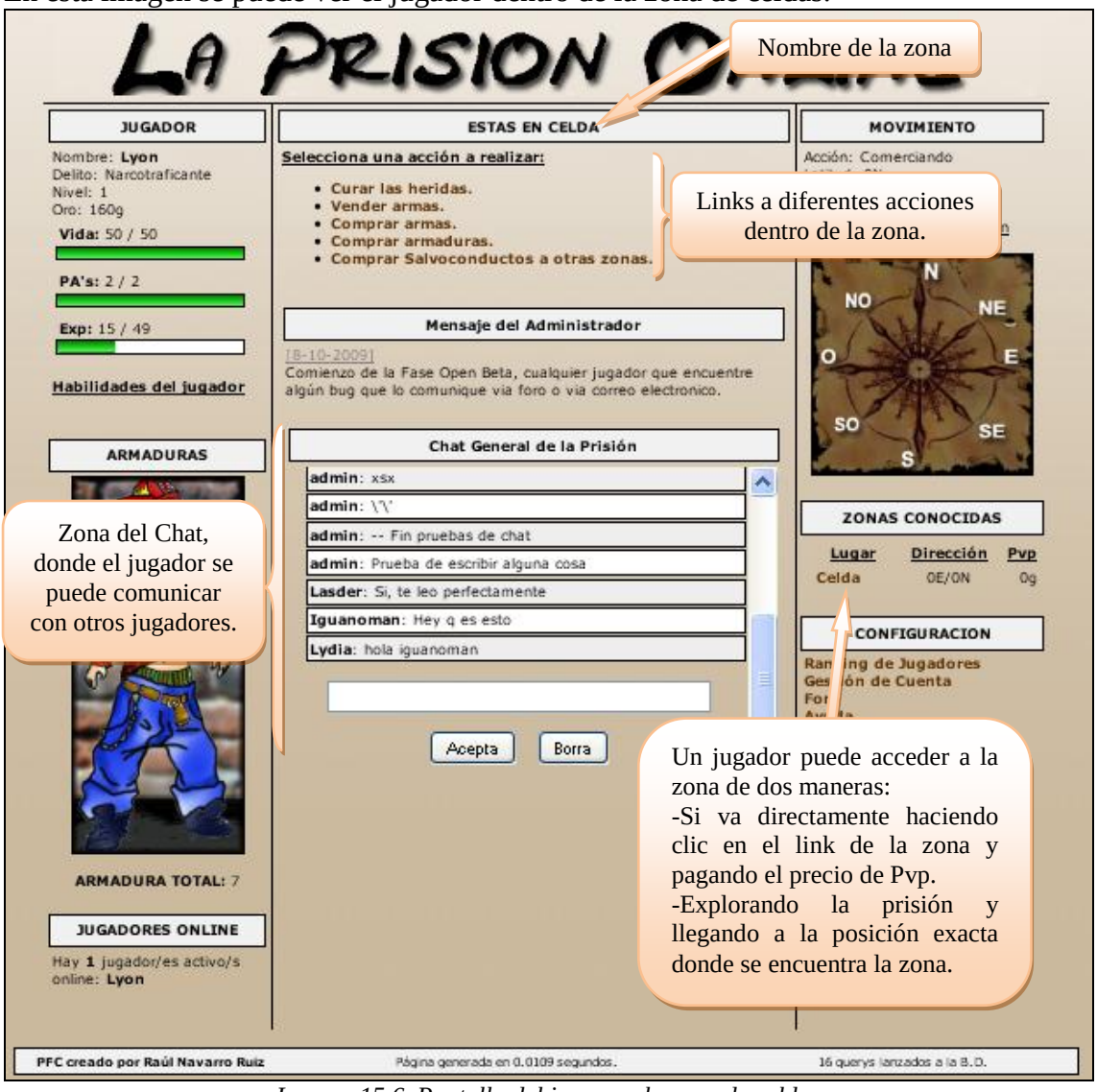


Imagen 15.6. Pantalla del juego en la zona de celdas

Cada zona es diferente entre sí, hay un nivel máximo por zona para curar las heridas, es para fomentar la exploración de la prisión entre los jugadores y que no tengan que estar siempre en la misma zona.

También hay que decir que no todas las zonas tienen porque vender armas o armaduras. Hay zonas que solo venderán armas, otras que solo venderán armaduras, otras que no venderán nada y otras que venderán ambas, la mejor forma de saber que se vende en cada zona es ir al link de compra de salvoconductos y ver la información que aparece.

Si por ejemplo el jugador quiere comprar un arma le saldrá una ventana similar a esta:

Imagen 15.7. Captura de la lista de armas en la zona de celdas

En esta imagen se puede observar las diferentes armas que están a la venta en esta zona en concreto.

Normalmente cada zona que venda armas tendrá las propias acorde con el nivel del jugador a esa zona, por ejemplo en la zona de celdas se supone que jugadores de más nivel que 7 u 8 no tendrán ningún aliciente puesto que los bots de los alrededores son muy débiles y las armas y armaduras son obsoletas para su nivel.

Las mejores armas y armaduras se encontraran luchando contra bots, hay un plus en los bots líder y jefe en las probabilidades de que dejen caer mejores objetos.

En esta imagen se aprecia que el arma 'Mini Lanza' no está disponible a la venta puesto que el jugador ya la posee en su inventario.

Un jugador no puede llevar armas repetidas.

La compra de armaduras y salvoconductos son similares y no se muestran capturas de pantalla por ese motivo.

La venta de armas también es muy similar y tampoco requiere de más explicación.

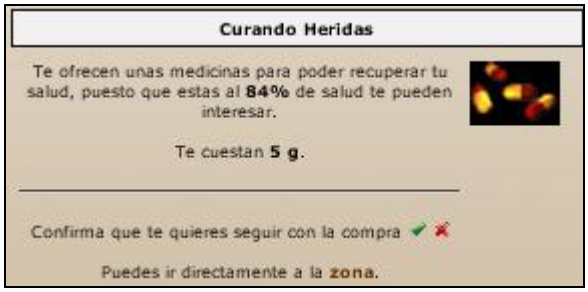
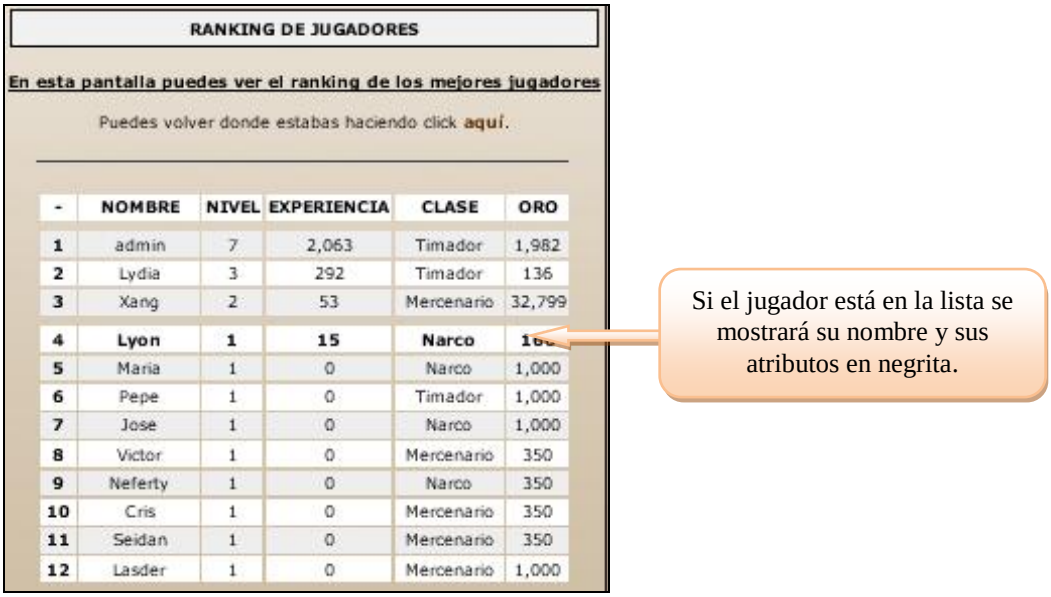


Imagen 15.8. Imagen de la compra de medicinas

En esta imagen se puede ver una captura de la pantalla de curación en el caso de que el jugador necesite recuperar salud (el jugador recupera salud de tres formas: Comprando curas, subiendo de nivel, explorando la prisión.)

15.5. Ranking de jugadores

En esta pantalla se puede ver una lista de los 25 mejores jugadores, la lista se ordena en función de la experiencia que tiene el jugador acumulada y muestra información sobre el nivel, la clase o el oro que posee el jugador.



-	NOMBRE	NIVEL	EXPERIENCIA	CLASE	ORO
1	admin	7	2,063	Timador	1,982
2	Lydia	3	292	Timador	136
3	Xang	2	53	Mercenario	32,799
4	**Lyon**	**1**	**15**	**Narco**	**16**
5	Maria	1	0	Narco	1,000
6	Pepe	1	0	Timador	1,000
7	Jose	1	0	Narco	1,000
8	Victor	1	0	Mercenario	350
9	Neferty	1	0	Narco	350
10	Cris	1	0	Mercenario	350
11	Seidan	1	0	Mercenario	350
12	Lasder	1	0	Mercenario	1,000

Imagen 15.9. Captura de la pantalla de ranking de jugadores

Esta pantalla es importante puesto que una de las mejoras podría ser el dar premios a los 3 mejores jugadores del ranking cada mes, así pues es una pantalla que fomenta la competitividad entre los jugadores para ver quién es el mejor posicionado.

15.6. Exploración

Para moverte por el mapa has de usar los botones de movimiento que encontrarás en la columna de la derecha dentro del juego.

Arriba de estos botones se te indicará la posición actual en la que te encuentras. Esta posición, en el caso de que estés buscando bots, también la verás reflejada en el mapa con un punto blanco.

Imagen 15.10. Botones de movimiento

Haciendo clic en cualquier dirección el personaje se moverá por el mapa.

Si llegas al límite del mapa, no se te permitirá moverte más allá, cada vez que te mueves existe la probabilidad de encontrarte con un bot. Pero el bot que te encuentras es aleatorio en un rango de niveles determinado.

También podrás ver las posiciones de los otros jugadores en rojo y tu posición en blanco, así como unos cuadrados negros que te indicarán que allí se encuentra una zona.

Para 'descubrir' una nueva zona puedes explorar los cuadrados negros del mapa o bien comprar el salvoconducto que te lleva.

Pasando el puntero del ratón por encima de los cuadrados negros se te indicará que zona es la que hay escondida.

Imagen 15.11. Captura del mapa de la prisión con 3 jugadores.

15.7. Combate

El combate es la esencia del juego, también el hecho de que los jugadores puedan comunicarse entre ellos y tengan alternativas en la compra de armas o armaduras, pero en definitiva el juego está orientado al PvE[23].

El jugador lucha contra los bots cuando sale a explorar el mapa de la prisión con los botones de movimiento.

En la siguiente imagen se muestra una captura de pantalla del inicio de un combate, explicando las diferentes opciones que hay y que significa cada cosa.

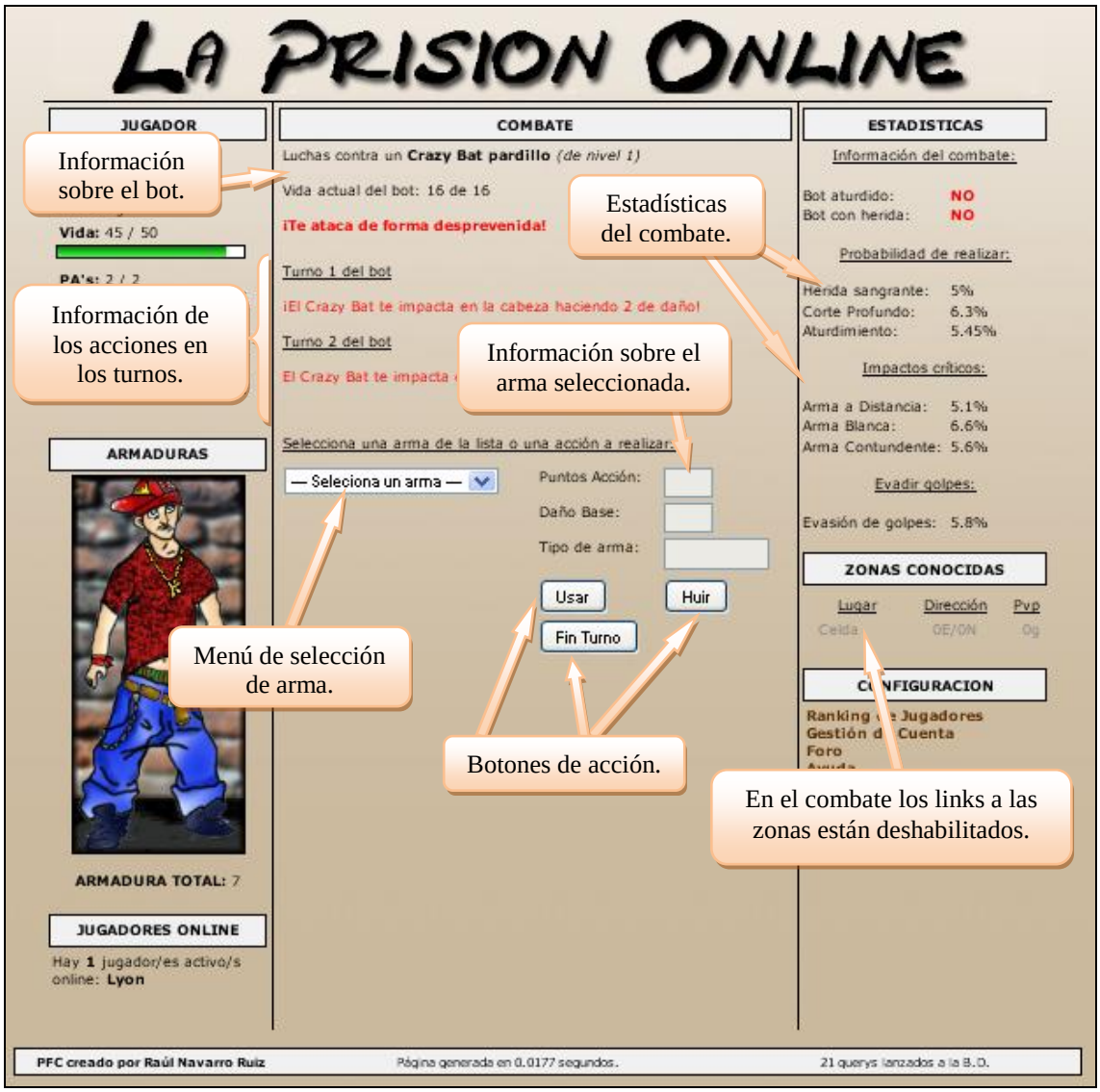


Imagen 15.12. Captura de la pantalla de combate.

[23] Jugador contra entorno o JcE, también conocido como PvE (del inglés player versus environment) es un término usado en los videojuegos de rol, sobre todo en MMORPG, para designar una modalidad en la que uno o más jugadores se enfrentan contra la máquina.

Como he dicho durante todo el desarrollo del proyecto, el juego es muy intuitivo y no se necesita de muchas horas de práctica para entender el funcionamiento, basta simplemente con hacer un par de combates para entender el funcionamiento básico.

Los botones de acción:

- **Usar:** Sirve para usar el arma seleccionada, debemos tener los Puntos de Acción requeridos para usarla.

- **Huir:** Como su nombre indica, sirve para intentar abandonar una pelea. No siempre será posible huir se tienen en cuenta factores como la agilidad del jugador, la diferencia de niveles entre bot y jugador, etc. En el caso de que el bot esté aturdido tendremos un bonus para huir.

- **Fin de Turno:** Si no queremos realizar ninguna acción hemos de apretar este botón, esto será útil cuando por ejemplo no tengamos suficientes p.a.'s para usar un arma debido a que los hemos gastado todos en el turno anterior.

Los puntos de acción se recuperan íntegramente al final de cada ronda (1 ronda equivale a los 2 turnos del jugador o del bot).

En la siguiente captura podemos observar la pantalla de combate y como se nos muestra la información del arma seleccionada.

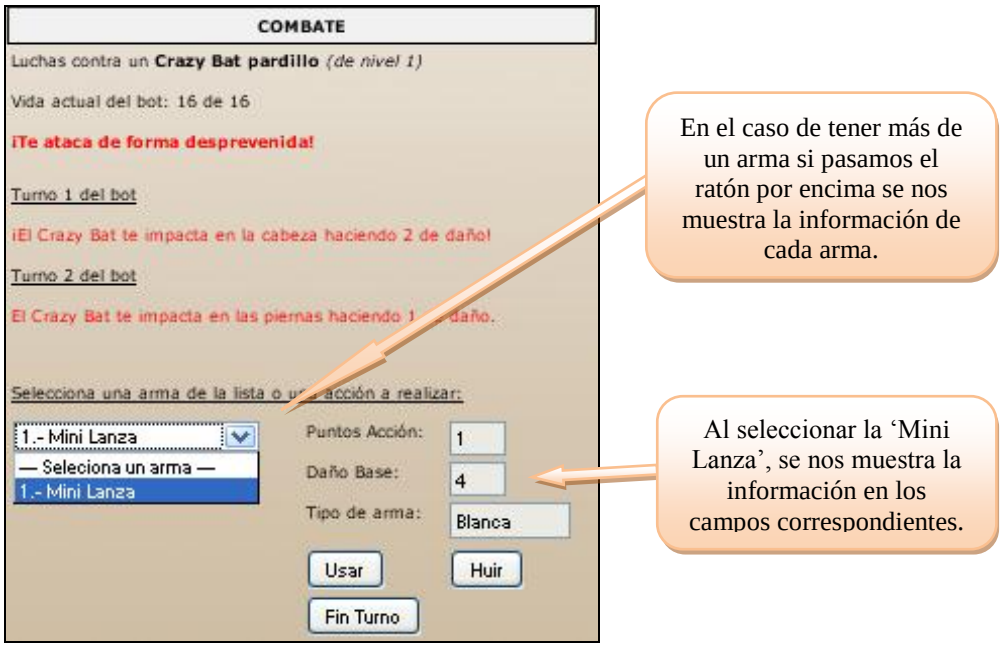


Imagen 15.13. Captura que muestra al jugador los atributos del arma seleccionada

Esta captura muestra el resultado de ejecutar la acción anterior, es decir, seleccionar la 'Mini Lanza' y usarla sobre el Crazy Bat pardillo.

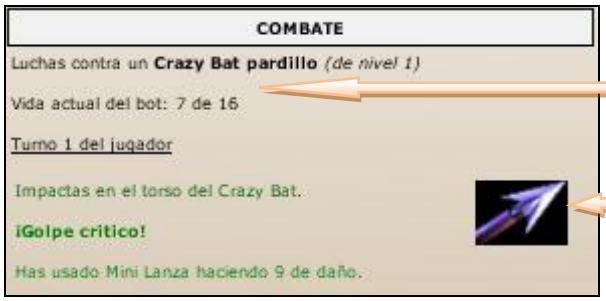


Imagen 15.14. Captura que muestra información del turno 1 del jugador.

El combate seguiría su curso normal realizando rondas del jugador y del bot… hay tres posibilidades que el jugador huya, que el jugador muera o que el jugador mate al bot.

En la siguiente captura muestro el resultado de que el jugador mata al bot.

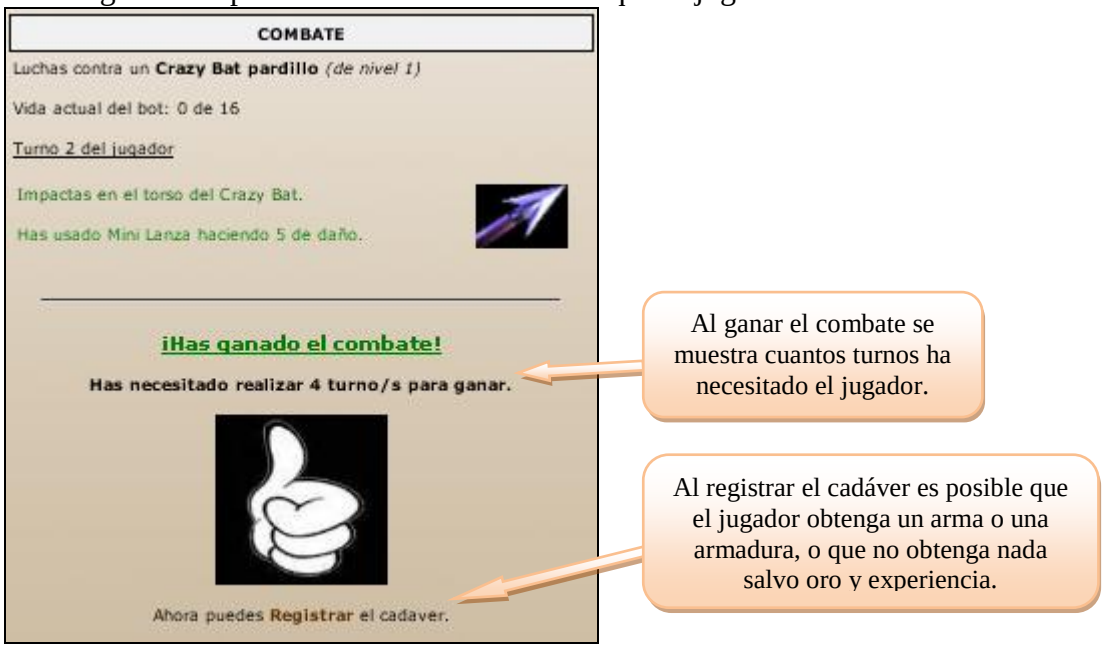


Imagen 15.15. Captura donde el jugador gana el combate al bot.

Después de darle a registrar nos aparecerá la siguiente pantalla que nos mostrará información de la experiencia y el oro que el jugador ha obtenido, así como si ha tenido suerte y ha encontrado algún objeto, en este ejemplo no se encontró ninguno.

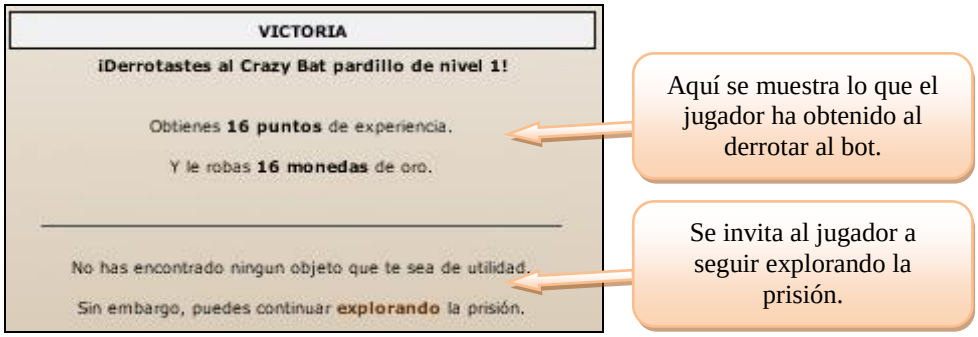


Imagen 15.16. Captura de la pantalla de victoria.

Para ver un ejemplo de que el bot dejara caer un objeto, he hecho la siguiente captura:

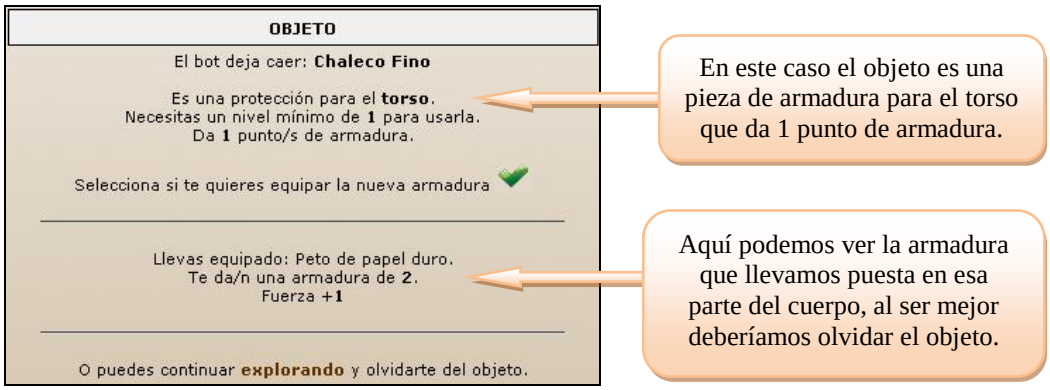


Imagen 15.17. Captura de la pantalla de objeto

15.7.1. Barras de Salud, Experiencia y p.a.'s

Las barras indican de una manera gráfica como se encuentra el jugador de vida, experiencia o puntos de acción.

Imagen 15.18. Imagen de los tres colores de las barras de salud y p.a.

Hay tres colores, el verde se mostrará si el jugador está entre el 67% y el 100%, el amarillo si está entre el 34% y el 66% y finalmente el rojo si está a menos del 33% inclusive.

La experiencia siempre estará en verde, y nunca decrecerá. Los puntos de vida y los de acción sí que sufrirán variaciones.

15.8. Foro

En la zona del foro los jugadores pueden postear mensajes, crear temas, y comunicarse entre ellos.

El foro que he integrado en el juego es muy básico pero es funcional que es lo importante, en el menú de administrador se puede elegir el usar un foro externo.

En esta captura muestro un ejemplo de cómo se vería el foro con dos mensajes:

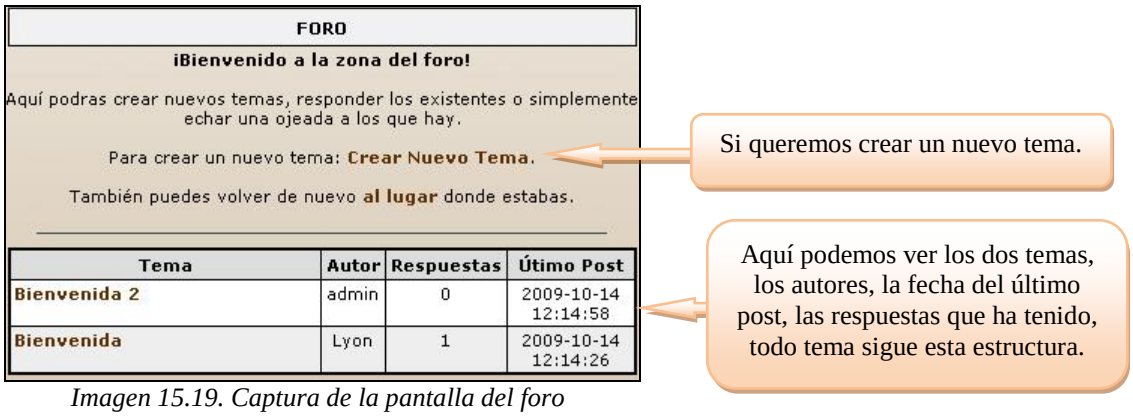


Imagen 15.19. Captura de la pantalla del foro

Si queremos responder a un tema, simplemente clicamos en el nombre del tema y se nos mostrará una pantalla similar a esta. En el caso de que el campo del mensaje esté vació no se insertaría nada.

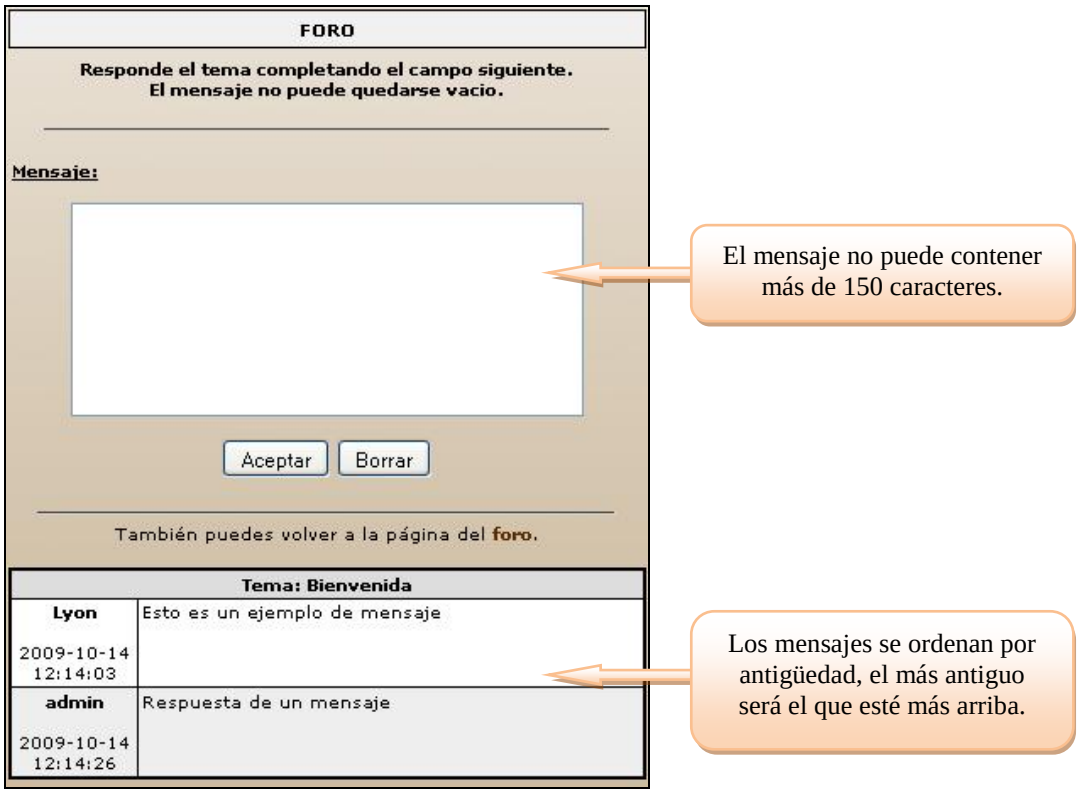


Imagen 15.20. Aquí podríamos responder un mensaje de un tema

16. Manual de Administrador

En el punto que trataré a continuación mostrare información de cómo instalar y configurar la aplicación para su correcto funcionamiento. También mostraré información del menú de administrador, de cómo funciona y sus usos.

16.1. Requisitos mínimos

Los siguientes requisitos son obligatorios para que la aplicación funcione correctamente, esta información se puede ampliar en el punto 5.2.1 Instalación de la aplicación y en punto 10 Tecnologías usadas:

- Servidor Web Apache 2.1 o superior.
- El lenguaje PHP 5.0 o superior.
- El SGBD MySQL 5.0 o superior.
- La aplicación phpMyAdmin 3.0 o superior para gestionar la base de datos.
- Un ordenador personal con conexión a internet o un servidor web en internet.
- El servidor de correo MiniRelay 0.9 o superior para el envío de correos electrónicos (opcional)
- Compresión zlib habilitada en el servidor (opcional)

Los cuatro primeros puntos se pueden sustituir por el paquete XAMPP 1.7 o superior.

16.2. Instalación de la aplicación

Para la correcta instalación de la aplicación realizar los siguientes pasos:

- Copiar todo el contenido de la carpeta de la aplicación a la carpeta raíz correspondiente a los documentos web.
    - Por ejemplo si usamos el XAMPP, tiene los archivos tendrán que ir en la carpeta c:\xampp\htdocs*nombre_carpeta*

- Crear una nueva base de datos en el phpMyAdmin con el nombre *prision*.
    - Para entrar al phpMyAdmin en el caso de haber instalado el XAMPP, abrir un nuevo navegador, escribir *localhost* y en la pantalla de bienvenida buscar el phpMyAdmin.

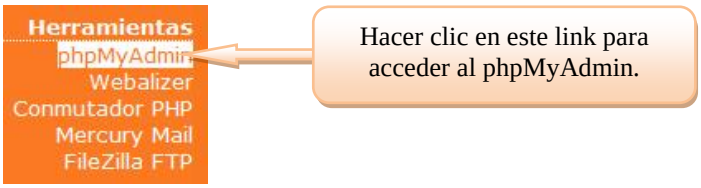


Imagen 16.1. Imagen del link para acceder al phpMyAdmin.

- Una vez creada la nueva base de datos esta estará vacía, se tienen que importar los datos para poder comenzar a usar la aplicación.
    - Hemos de buscar el archivo *prision.sql* e importarlo.

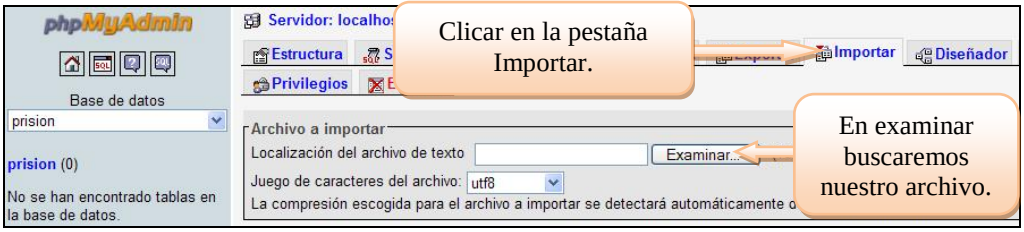


Imagen 16.2. Captura dentro de la aplicación phpMyAdmin

- Finalmente accederemos al archivo configuración.php de la carpeta raíz y cambiaremos los valores por los de nuestra configuración.
  - Por ejemplo, en este caso sería algo parecido a esto:

```
<?php // Todo lo relacionado con la configuración para acceder a la base de datos

$valoresbd = Array(
        "servidor"   => "localhost",     // Nombre del servidor de MySQL (Defecto: localhost)
        "usuario"    => "root",          // Nombre de usuario de MySQL.
        "contraseña" => "",              // Contraseña de MySQL.
        "nombre"     => "prision",       // Nombre de la base de datos de MySQL.
        "prefijo"    => "pfc",           // Prefijo para las tablas
        "pass"       => "");             // Contraseña para usar en las cookies.

?>
```

Imagen 16.3. Imagen de la configuración para el correcto funcionamiento del juego

  - Se sugiere poner una contraseña de MySQL cuando creemos nuestro usuario para mejor seguridad, no obstante en este ejemplo el usuario *root* no tenía contraseña asociada.

- Una vez hayamos hecho todo esto tendremos el usuario *admin* con la contraseña *admin* por defecto, se sugiere que se cambie la contraseña dentro de la aplicación.

16.2.1. Comprobación de la correcta instalación

Comprobar que la aplicación se ha instalado correctamente es muy sencillo. En el caso de haber instalado el XAMPP, ejecutar el archivo *xampp-control.exe* en la raíz de la carpeta donde se instaló.

Nos aparecerá el panel de control de la aplicación, simplemente tenemos que clicar Start en Apache y MySQL, teniendo el resultado que aparece en la imagen.

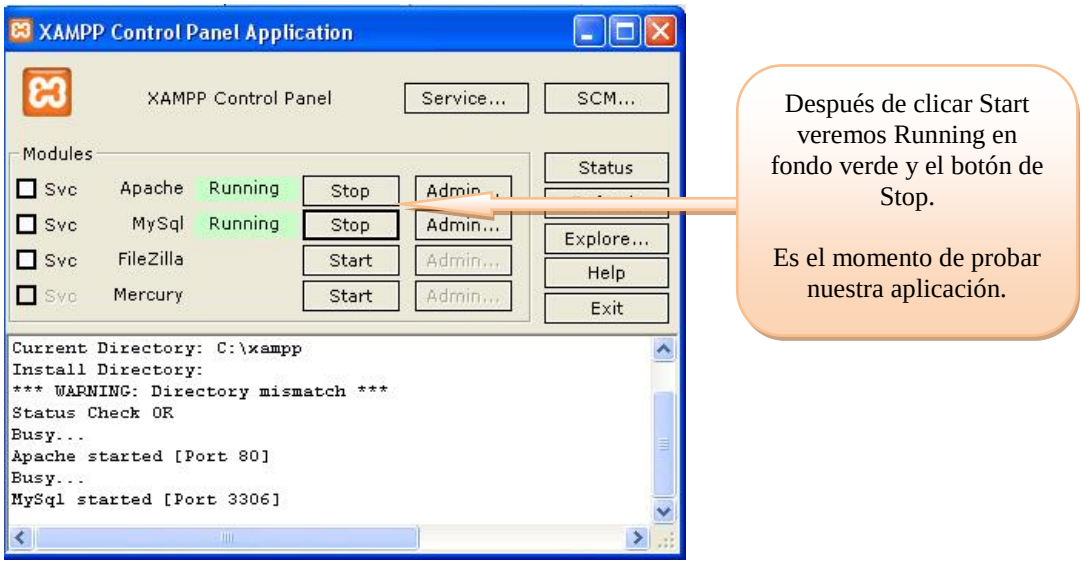


Imagen 16.4. Captura de pantalla del panel de Control de la aplicación XAMPP

Abriremos el navegador Firefox y escribiremos en la barra de dirección: *http://localhost/prision* y veremos que la barra de dirección nos redirige a la dirección: *http://localhost/prision/ingresar.php?ir=ingresar* si vemos la pantalla inicial de la aplicación es que todo ha salido correctamente.

En este caso es porque el nombre de la carpeta donde instalamos los archivos lo llamamos *prision*, si se llamara de otra manera, se tendría que cambiar el nombre.

Estamos en localhost pero los usuarios de internet se podrían conectar a nuestro servidor simplemente cambiando la palabra localhost por la IP que tenemos actualmente en la máquina.

16.3. Instalación del servidor de correo

Tenemos la aplicación funcionando pero nos falta todavía un punto que es el servidor de correo, este paso se puede omitir si accedemos al juego con la cuenta de administrador y deshabilitamos el envío de correos.

Es muy sencillo, simplemente descomprimir el *miniRelay.zip* donde queramos y ejecutar el *miniRelay.exe* para que se automatice el envío de mensajes cuando se creen nuevas cuentas

16.4. Menú de Administrador

La aplicación tiene un menú de administrador para los usuarios que tengan privilegios de administrador, en este punto se explica cómo funcionan las diferentes opciones, no obstante es muy sencillo de entender una vez se ha modificado una par de veces.

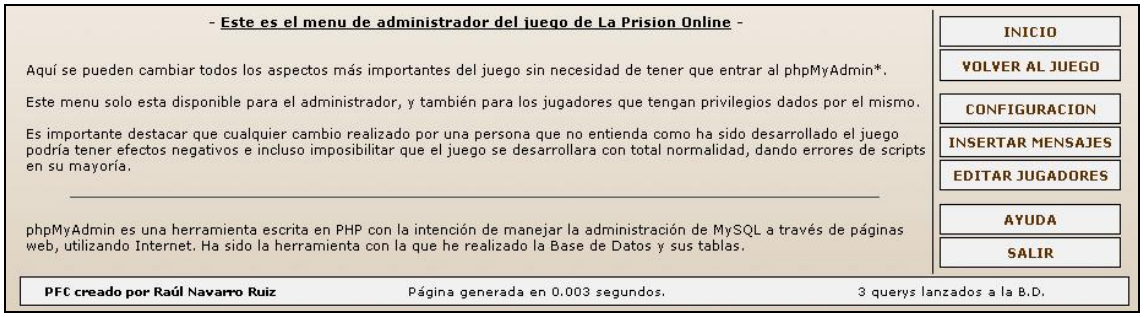


Imagen 16.5. Vista de la pantalla del menú de administrador

16.4.1. Configuración de la aplicación

En este punto se pueden configurar todos los detalles más importantes del juego, se creó para facilitar un acceso rápido a la configuración del juego.

La siguiente captura muestra esta página, los (?) muestran información para ayudar al administrador a saber qué es lo que hace ese campo.

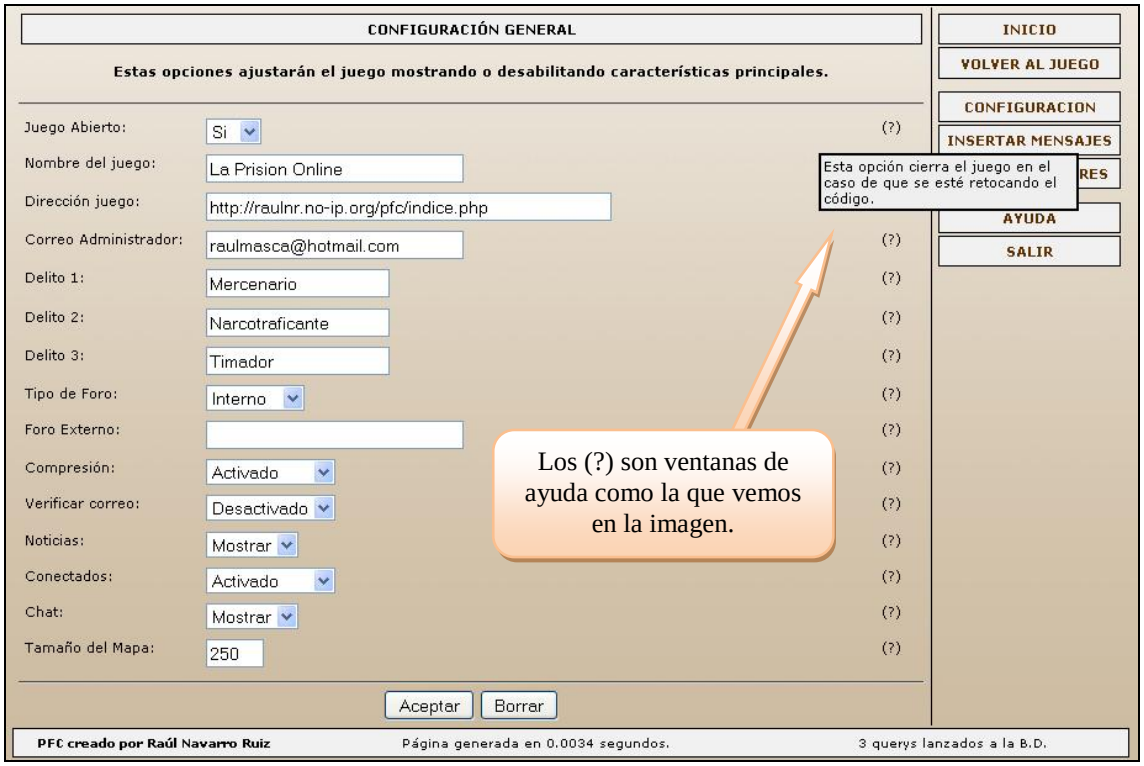


Imagen 16.6. Vista de la pantalla de la configuración dentro del menú de administrador

Cuando hayamos hecho los cambios oportunos aceptaremos, el botón borrar sirve para deshacer los últimos cambios.

16.4.2. Insertar mensajes

Con este link podemos acceder fácilmente a crear un nuevo mensaje que lo verán todos los jugadores cuando accedan a las zonas, es una forma sencilla de informar a los jugadores sobre novedades, cambios, concursos que se pudieran realizar tanto periódicamente como esporádicamente.

Imagen 16.7. Vista de la pantalla de inserción de mensajes dentro del menú de administrador

16.4.3. Editar Jugadores

Este es una de las herramientas más potentes del menú del administrador juntamente con la configuración del juego, aquí se puede ver información detallada de los jugadores y acceder a esta información para modificarla como el administrador considere oportuno.

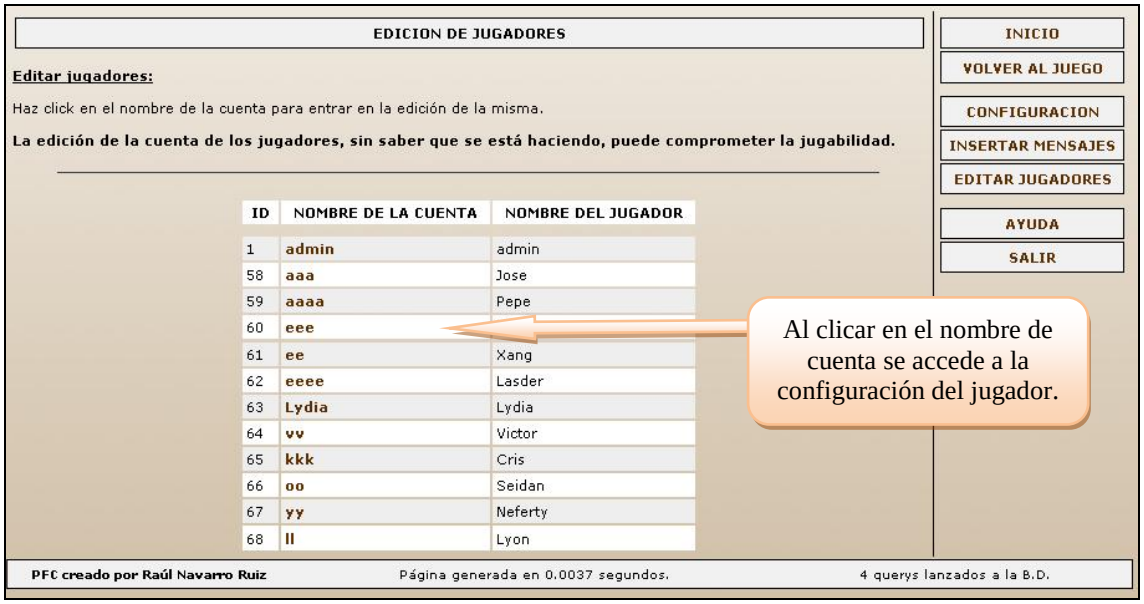


Imagen 16.8. Vista de la pantalla de edición de jugadores dentro del menú de administrador

En esta captura se pueden ver los nombres de los jugadores así como sus nombres de cuenta, clicando en el nombre de cuenta se accede a la configuración de la cuenta del jugador.

Si hacemos clic en el jugador con nombre de cuenta *eee* podemos acceder a toda su información, en la siguiente captura solo muestro una parte de esta información, pero se pueden editar todos los aspectos de la cuenta exceptuando los 6 primeros.

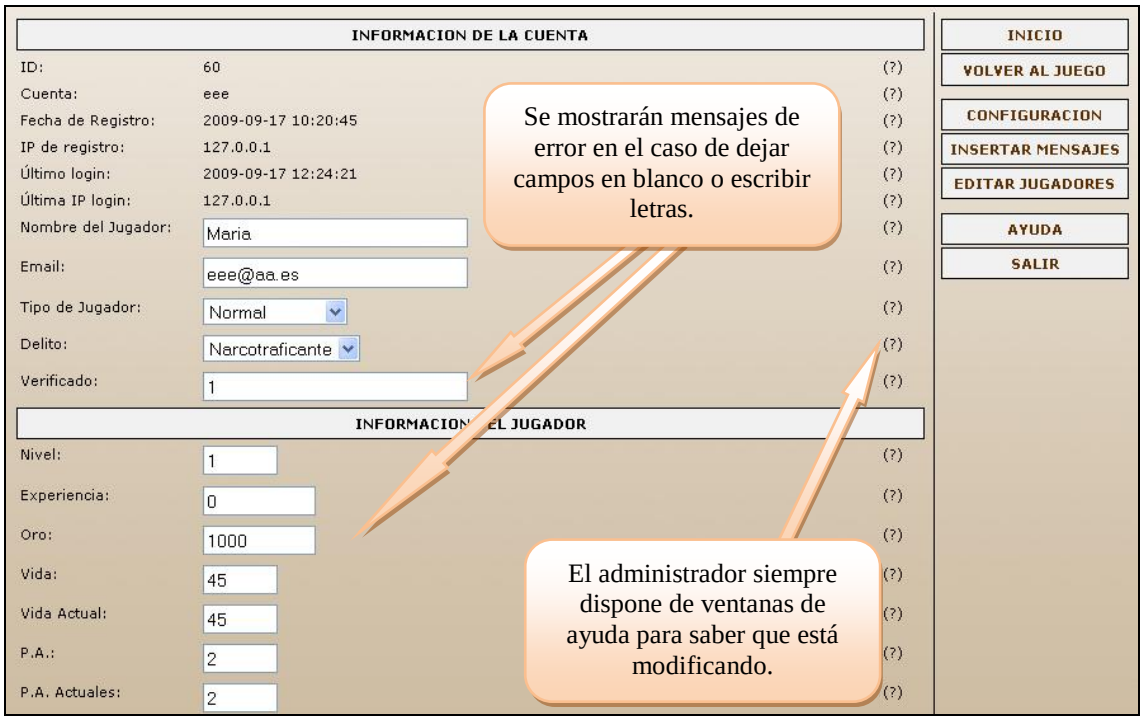


Imagen 16.9. Vista de la pantalla de edición de los diferentes jugadores.

Una vez se hayan modificado los campos oportunos simplemente aceptaremos los cambios y volveríamos a la página de edición de jugadores. Siempre podríamos ir a un link de la derecha para ir directamente a la acción que quisiéramos realizar.

Parte VI: Balances y Conclusiones

17. Planificación Final y Análisis económica

En este apartado trataré de mostrar los costes económicos del proyecto y la variación final del tiempo estimado y el tiempo real.

17.1. Análisis de Costes

Para el desarrollo de este proyecto de final de carrera se han usado alrededor de 600 horas. Podemos hacer la siguiente división por tareas:

- **Búsqueda e información de las tecnologías a usar:** Esta etapa se compone de dos etapas, la primera de ellas fue saber que tecnología debía usar para desarrollar el proyecto y la segunda etapa fue el aprender esa tecnología. Fueron un cúmulo de tecnologías, puesto que fue el aprender PHP, las CSS, XHTML, SQL y Gestionar la base de Datos.

- **Diseño:** Esta parte corresponde a como quería desarrollar el juego, sus características, sus funcionalidades y en general todo lo relacionado con el aspecto visual del juego, la caracterización de los delitos, las armas, las tablas de progresión de experiencia, las habilidades de los jugadores, etc. Ha sido una parte en continuo desarrollo, por lo que ha supuesto un esfuerzo elevado en tiempo.

- **Implementación:** Consiste en el desarrollo del proyecto a nivel de programación. Este apartado iría de la mano con el primero, es decir, que el aprendizaje de las tecnologías también ha sido a medida de que el proyecto iba avanzando. Ha sido una tarea ardua, sobre todo al principio, una vez tuve desarrollados la estructura inicial fue ir poniendo piezas encima para obtener el resultado actual.

- **Implantación:** Básicamente esta tarea fue la instalación de los programas a usar y su configuración. Hay que tener en cuenta que puede que la instalación no fuera especialmente complicada, pero en este apartado se tiene también en cuenta el tiempo que se tardo en la creación de la base de datos.

- **Testeo:** El correcto funcionamiento de la aplicación ha sido una máxima durante todo este tiempo. Se quería evitar en todo momento errores del juego, la manipulación no autorizada en la URL[24], pero sobretodo que el juego fuera estable y balanceado, es decir que los combates fueran equilibrados, el uso de las armas adecuado al nivel del jugador, etc.…

- **Documentación:** Básicamente es el desarrollo de la memoria del proyecto final de carrera y las ayudas dentro del juego. Ha requerido un número elevado de horas debido a la complejidad de la realización de la memoria. Las ayudas

[24] URL significa Uniform Resource Locator, es decir, localizador uniforme de recurso. El URL combina el nombre del ordenador que proporciona la información, el directorio donde se encuentra, el nombre del archivo y el protocolo a usar para recuperar los datos.

dentro del juego son la página de ayuda propiamente dicha y ciertas ayudas internas como en la página de registro o en la página del menú de administrador.

	Tecnologías	Diseño	Implementación	Implantación	Testeo	Documentación	Total
Tiempo (horas)	96	112	215	16	80	80	**599**
%	16,0%	18,7%	35,9%	2,7%	13,4%	13,4%	**100%**

En el siguiente gráfico podemos observar de una forma más clara como fue la dispersión temporal del coste temporal para cada recurso, aunque es un valor aproximado puesto que no se contabilizaron las horas de forma exhaustiva, si que nos puede dar una idea del tiempo que se tuvo que dedicar a cada apartado.

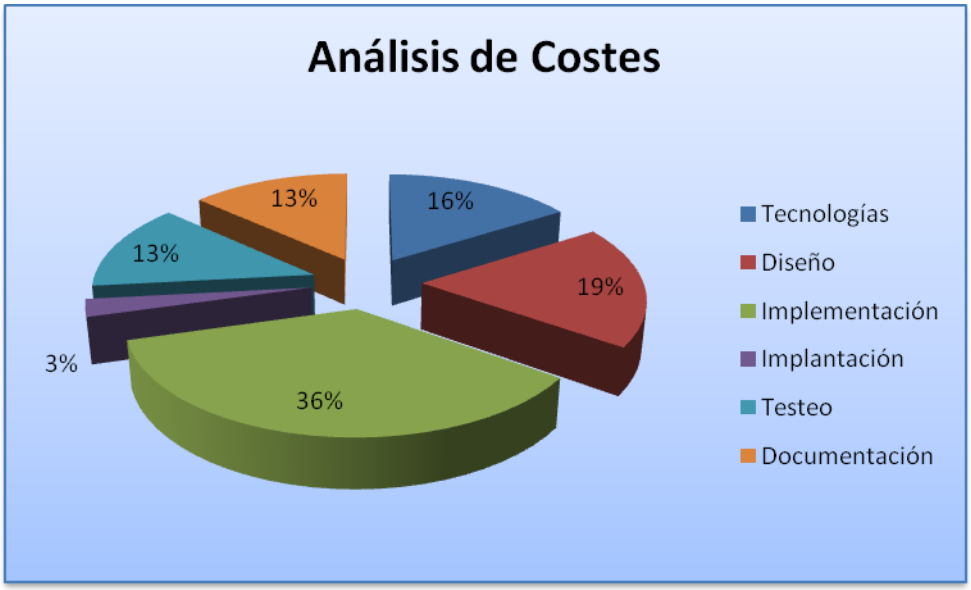


Imagen 17.1. Gráfico que indica el porcentaje de tiempo que ha consumido cada recurso.

17.2. Comparativa entre la estimación inicial y la final

En el capítulo 4 se ha podido observar la planificación inicial del tiempo en desarrollar este proyecto. No he hecho un diagrama GANTT[25] puesto que el proyecto no ha sido desarrollado continuamente, se empezó a principios del año 2009 con el diseño del juego y se finalizo a principios de septiembre del mismo año con las últimas pruebas de integridad.

Este es el balance final entre la estimación realizada y el tiempo final de dedicación.

	Tecnologías	Diseño	Implementación	Implantación	Testeo	Documentación	Total
Tiempo estimado	75	175	350	12	50	100	762
Tiempo real (h)	96	112	215	16	80	80	**599**
Variación (%)	21,88	-56,25	-62,79	25,00	37,50	-25,00	-27,21

En el siguiente gráfico podemos observar de una forma más clara como fue esta variación, aunque es un valor aproximado puesto que no se contabilizaron las horas de forma exhaustiva, si que nos puede dar una idea del tiempo que se tuvo que dedicar a cada apartado.

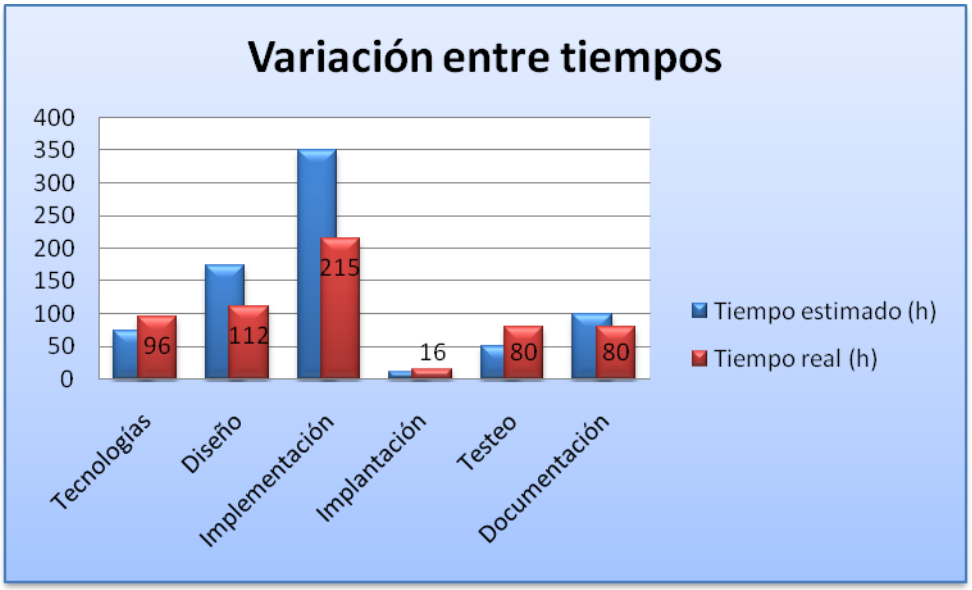


Imagen 17.2. Gráfico que muestra la variación temporal entre el tiempo estimado y el real.

[25] El diagrama de Gantt, gráfica de Gantt es una herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

17.2.1. Conclusiones sobre los datos

- El aprendizaje de las tecnologías y la búsqueda de información ha sido superior al esperado, esto significa que me costó más tiempo familiarizarme con las tecnologías a usar.

- El Coste temporal del diseño fue algo más del 56% inferior a lo previsto, esto fue debido a que tenía una versión del juego estable mucho más rápido de lo que se calculó en un principio.

- La Implementación fue casi un 63% inferior al coste previsto, esto se debió en parte a que las funciones de JavaScript que quise usar ya estaban hechas, no tuve que crearlas de cero, y por lo tanto no tenía la necesidad de aprender un lenguaje con el que no estaba familiarizado.

- En la implementación se suprimieron algunos aspectos como el combate entre jugadores o la creación de clanes. Además comentar que en el tiempo que estimé fue a la alza para evitar quedarme sin tiempo, es por esto que la diferencia es tan grande.

- La instalación de las tecnologías y la creación de la base de datos fue más o menos el tiempo estimado.

- El testeo fue casi el doble de lo previsto puesto que fueron muchos apartados a depurar y era una parte que debía estar completamente verificada.

- La documentación aproximadamente fue el tiempo previsto, esta pequeña variación inferior fue debido a que miré anteriormente como desarrollar la documentación del PFC y fue una gran ventaja.

17.3. Análisis económico

En este apartado se quiere mostrar el coste económico que hubiera tenido el realizar este proyecto, como el coste temporal, es un valor aproximado puesto que los precios son variables y no están fijados en ningún documento a la hora de realizar este estudio.

El equipo físico donde se realizó este proyecto no se tiene en cuenta a la hora de realizar este estudio puesto que ya se disponía previamente.

Los diferentes programas utilizados, XAMPP y el MiniRelay son libres y no se tiene que pagar ninguna licencia.

El coste de la licencia del Windows XP desde donde se realizó el proyecto no se tiene en cuenta debido al mismo motivo dado que el del equipo físico, no obstante se contabilizaría en el caso de que no se tuviera.

Los costes de las licencias de los programas usados no los tengo en cuenta, esto es debido a varios motivos:

- Windows XP: Su licencia se compro cuando compre el equipo donde he realizado el proyecto, así pues, es el mismo caso de porque no se tiene en cuenta el coste del equipo.

- Adobe Dreamweaver CS4: Esta es la última versión, yo usé la Dreamweaver 8 y usé varias versiones de demostración que eran de coste cero.

- Photoshop CS4: Yo usé el CS2, use también versiones de demostración, con lo que el coste fue cero.

En el caso de comprar las licencias indico el coste a continuación:

- La Licencia del Adobe Dreamweaver CS4 es de aproximadamente 273€ [7]
- La Licencia del Adobe Photoshop CS4 es de aproximadamente 478€ [8]
- La Licencia del Windows XP Home Edition es de aproximadamente 159€ [9]

Se utilizó una versión demo de 60 días del Microsoft Office 2007 para el desarrollo de la memoria así que no se contabilizará nada en este aspecto.

El coste de luz y material de oficina, así como el papel, etc. no se tiene en cuenta a la hora de realizar el estudio.

Por lo tanto, después de desglosar estos aspectos, lo único que necesitamos calcular es el coste de los recursos humanos, el proyecto solo lo realizó una persona, pero lo ideal sería dividir las tareas en dos o tres personas, para este estudio se supondrá que tendremos un responsable del proyecto un programador y un analista.

- **Responsable del proyecto:** Será el encargado de supervisar el correcto desarrollo del proyecto, así como de implantarlo y del lanzamiento.

- **Analista:** El encargado de diseñar y analizar las tecnologías a usar.

- **Programador:** El encargado de realizar la implementación del proyecto y realizar pruebas de testeo.

Se suponen los siguientes salarios:

- Responsable del proyecto: 30,000 € brutos al año o 14.98€/hora
- Analista: 26,000 € brutos al año o 12.78€/hora
- Programador: 20,000€ brutos al año o 9.48€/hora

Encargado	Tarea	Días	Horas	Coste
Programador	Formación de las tecnologías	12	96	910,08
Analista	Diseño y análisis	14	112	1431,36
Programador	Implementación	27	215	2038,20
Responsable del proyecto	Implantación	2	16	239,68
Programador	Testeos	10	80	758,40
Responsable del proyecto	Documentación	10	80	1198,40
TOTAL PERSONAL				**6576,12**

En el siguiente gráfico podemos observar de una forma más clara como es este coste en función de la tarea desarrollada, este valor es aproximado debido a que las horas no son exactos completamente ni tampoco los salarios de los empleados.

Imagen 17.3. Gráfico que muestra el coste por cada tipo de tarea.

Por lo tanto el coste final es de unos 6.500€ teniendo en cuenta la formación del programador y sin usar ninguna licencia.

Si quisiéramos comprar las licencias y formando al programador el coste aumentaría hasta los 7.500€ aproximadamente (es el coste de las licencias).

18. Análisis Medioambiental

Cuando al principio del documento se plantea la pregunta si este proyecto tiene en cuenta consideraciones medioambientales, se responde afirmativamente.

Es difícil de una forma objetiva decir que aporta este proyecto a la mejora o conservación del medioambiente, no obstante esta sería una posible lista.

- Al ser una aplicación para ser usada íntegramente por internet se puede evitar el usar transportes para poder distribuirla, evita el uso de carburantes como la gasolina o el gasoil.

- No es necesario el uso de un CD-DVD para instalar y jugar a la aplicación debido a que es completamente en línea, evita el uso de todos los procesos derivados en la creación de los CD's DVD's.

- Al tener una ayuda en línea y ventanas constantes de ayuda se evita el uso de un manual de juego, evitando así el uso de papel y la tala de árboles.

- La memoria se imprimió a doble cara, contribuyendo de esta manera al ahorro de papel.

- Al no usar un fondo blanco sobre la aplicación y usar un tono más bien oscuro se ahora consumo eléctrico[26], son pocos *Watts* de potencia pero también se pensó en ello a la hora de crear la aplicación.[10]

[26] Un monitor de 17 pulgadas en modo gráfico (800x600, 85 Hz) todo en negro consume 52 W. El mismo modo gráfico pero todo en blanco consume 61 W.

19. Análisis de Accesibilidad

Este proyecto ha tenido especial cura en la accesibilidad a personas con discapacidades, las que incluyen ceguera o visión deficiente, sordera y pérdida de audición, deficiencias de aprendizaje, limitaciones cognitivas, movilidad reducida, deficiencias del lenguaje, fotosensitividad y las combinaciones de todas estas.

Para poder hacer accesible la aplicación a todas estas personas, se han seguido las Pautas de Accesibilidad de Contenido Web 2.0, se definen tres niveles de conformidad: A (el más bajo), AA y AAA (el más alto).

Para cumplir el nivel AAA, se han de cumplir todas las pautas correspondientes al nivel AAA, todas las pautas correspondientes al nivel AA y todas las pautas correspondientes al nivel A.

Mi proyecto cumple con el nivel AA de la WCGA[27] 2.0 (y por tanto también cumple las pautas del nivel A), hay cuatro pautas generales y cada una se divide en varias sub-pautas. El cumplir con todas las sub-pautas significa que obtienes el nivel de conformidad AAA.

Las cuatro pautas generales son: [11]

- Principio 1: Perceptibilidad - La información y los componentes de la interfaz de usuario deben presentarse a los usuarios de la manera en que puedan percibirlos.

- Principio 2: Operabilidad - Los componentes de la interfaz de usuario y la navegación deben ser operables.

- Principio 3: Comprensibilidad - La información y el manejo de la interfaz de usuario deben ser comprensibles.

- Principio 4: Robustez - El contenido debe ser lo suficientemente robusto como para confiarse en su interpretación por parte de una amplia variedad de agentes de usuario, incluidas las tecnologías asistivas.

Estas son algunas de las pautas que mi proyecto cumple:

- **Pauta 1.1 Alternativas textuales:** Alternativas textuales para todo contenido no textual, de manera que pueda modificarse para ajustarse a las necesidades de las personas (Nivel A)

[27] En inglés **W**eb **C**ontent **A**ccessibility **G**uidelines (WCAG), o Pautas de Accesibilidad de Contenido Web, son unas guías publicadas por la W3C para hacer que el contenido sea accesible a personas con discapacidades.

- **Pauta 1.4.1 Empleo del color:** El color no se emplea como el único medio visual para transmitir una información, indicar una acción, provocar una respuesta o distinguir visualmente un elemento. (Nivel A)

- **Pauta 1.4.3 Contraste (mínimo):** La presentación visual del texto y las imágenes de texto tienen una relación de contraste de al menos 4.5:1, excepto para los siguientes casos: Gran tamaño, incidental, logotipos (Nivel AA)

- **Pauta 2.1.1 Teclado:** Toda funcionalidad del contenido es operable a través de una interfaz de teclado sin que exista un límite de tiempo específico para realizar las pulsaciones de las teclas, excepto para cuando la funcionalidad subyacente requiere una interacción del usuario para la que es relevante no solo los puntos finales de su movimiento sino también la ruta que traza el mismo. (Nivel A)

- **Pauta 2.3.1 Tres destellos o por debajo del umbral:** Las páginas web no contienen nada que destelle más de tres veces en cualquier periodo de un segundo, o el destello está por debajo de los umbrales de destello general y de destello rojo. (Nivel A)

- **Pauta 2.4.6. Encabezados y etiquetas:** Los encabezados y las etiquetas describen el tema o propósito pertinente. (Nivel AA)

- **Pauta 3.3.3 Sugerencia tras error:** Si se detecta automáticamente un error de entrada de datos y se pueden determinar las sugerencias apropiadas para la corrección, entonces se proporcionan las sugerencias al usuario, a menos que esto pudiese poner en riesgo la seguridad o el propósito del contenido. (Nivel AA)

- **Pauta 4.1.1 Interpretación:** Para contenido que se haya implementado empleando un lenguaje de marcado, los elementos cuentan con etiquetas completas de cierre y apertura, se han anidado correctamente, no contienen atributos duplicados y cualquier ID es único, excepto donde la especificación permita excepciones. (Nivel A)

20. Conclusiones y Mejoras

Estas conclusiones complementan a los puntos 14.2.3 y 17.2.1 que han sido unos puntos con conclusiones especificas sobre los test realizados y sobre los tiempos de respuesta.

20.1. Conclusiones Personales

Durante la elaboración del Proyecto Final de Carrera (PFC) he llegado a varias conclusiones de las que destaco:

- El objetivo personal de crear un juego de ordenador que tuviera unas condiciones mínimas de jugabilidad, seguridad y diversión ha sido completado y mejorado con creces en las expectativas iníciales que tenia.

- Es fundamental hacer un buen análisis del problema y utilizar las herramientas adecuadas aunque esto te lleve más tiempo, luego te soluciona muchos problemas.

- Igual de importante que el análisis del problema es el realizar un buen diseño del juego, el proyecto podría haber consistido en apretar un botón que tirase una moneda y si saliera cara ganases experiencia y si saliera cruz no, pero no tendría ni profundidad, ni diversión.

- Se añadieron algunas funcionalidades nuevas al juego a medida que se desarrollaba y que no estaban planificadas en el diseño original, esto me permitió aportar un valor añadido al juego así como aprender a hacer frente a cambios no planeados en el desarrollo de un proyecto.

- Es importante el punto de vista de los usuarios que jugarán a mi aplicación, se tuvo en cuenta su opinión para modificar algún aspecto del juego que previamente era de otra manera, a fin de cuentas es para ellos quien va dirigida la aplicación.

- Aunque realizar un PFC es un camino largo, costoso y no exento de dificultades la valoración final mirando el camino recorrido es más que positiva.

20.2. Conclusiones Técnicas

- El uso de plantillas para mostrar las páginas ha sido todo un acierto, de una forma muy sencilla se pueden añadir nuevas funcionalidades al proyecto.

- Se ha obtenido un conocimiento más profundo del PHP i XHTML, el uso de un lenguaje de programación u otro fue determinante a la hora de realizar la aplicación.

- He aprendido a utilizar el JavaScript y los CSS (páginas de estilo en cascada), anteriormente conocía su existencia pero nunca las había usado.

- He mejorado los conocimientos que tenia con el SGBD MySQL y uno de los aspectos destacables en este punto ha sido evitar el uso de subconsultas para mejorar el rendimiento global.

- Limitar la maquetación a XHTML 1.0 ha supuesto tener mucha cura en el código para poder pasar los estándares, así como en la maquetación de las páginas de estilos en cascada (CSS) también se tuvo cura para pasar los estándares.

- He aprendido a usar y el servidor web Apache así como a usar la compresión en las páginas generadas para optimizar los recursos.

20.3. Mejoras

Como se explicó en el punto 13.2 en el desarrollo de la aplicación se ha usado una programación modular, haciendo que se puedan añadir funcionalidades en un futuro sin tener que modificar todo el código.

Las posibles mejoras serian muy sencillas de introducirse en la aplicación simplemente será necesario el tiempo para desarrollarlas. Como posibles mejoras se pueden enumerar muchas puesto que un a un MMORPG se le pueden añadir nuevas funcionalidades aumentado así su jugabilidad.

- Creación de un nuevo tipo de combate además del existente de jugador contra bot (PvE), a un combate orientado a jugador contra jugador (PvP).

- Creación de clanes de jugadores con sus respectivos líderes y sistema de habla privado así como una página del ranking de todos los clanes.

- Introducción de un sistema de mensajería privada al jugador.

- Creación de paredes en el mapa para hacerlo más realista, haciendo que unas salas en concreto sean más difíciles de llegar que otras.

- Creación de un sistema de Quest[28] en el juego.

- Introducción de publicidad y banners al juego, seria sencillo puesto que todo el diseño está estructurado de una forma sencilla y ágil.

- Creación de un sistema de micropagos para obtener mejoras en el juego a cambio de dinero real (armas exclusivas, promociones de subir experiencia más rápido durante unos días, etc.).

- Optimizar la accesibilidad para que cumpla con el nivel AAA de la WCGA 2.0

[28] Son tareas que el jugador debe completar para obtener principalmente experiencia, oro u objetos.

Bibliografía

[1] *Blog personal de gerardooscarjt* [En Línea] [Consulta 3 de Octubre del 2009] Disponible en: http://gerardooscarjt.wordpress.com/2008/10/18/mysql-motores-de-almacenamiento/

[2] *Página de Apache friends* [En Línea] [Consulta 4 de Octubre del 2009] Disponible en: http://www.apachefriends.org/en/xampp-windows.html

[3] *Comunidad Linux Los Ángeles* [En Línea] [Consulta 13 de Octubre del 2009] Disponible en: http://www.linuxla.cl/?p=4708&cpage=1

[4] *Market Share* [En Línea] [Consulta 11 de Octubre del 2009] Disponible en: http://marketshare.hitslink.com/report.aspx?qprid=0&qptimeframe=M&qpsp=128

[5][6] *Test de velocidad de la conexión a internet* [En Línea] [Consulta 16 de Octubre del 2009] Disponible en: http://www.testdevelocidad.es/

[7] [8] *Página de adobe* [En Línea] [Consulta 15 de Octubre del 2009] Disponible en: http://www.adobe.com/products

[9] *Portal de compras amazon* [En Línea] [Consulta 5 de Octubre del 2009] Disponible en: http://www.amazon.com/

[10] *Página personal de Daniel Clemente* [En Línea] [Consulta 15 de Octubre del 2009] http://www.danielclemente.com/consumo/#toc16

[11] *Pautas de Accesibilidad de Contenido Web 2.0* [En Línea] [Consulta 11 de Enero del 2010] http://www.codexexempla.org/traducciones/pautas-accesibilidad-contenido-web-2.0.htm#intro-layers-guidance

Anexo I: Consultas

Para la realización del proyecto se han consultado otras páginas web que por su naturaleza no lo pongo dentro del punto de la bibliografía, pero que la complementan.

Web gratuita de JavaScripts para usar en tu página web [En Línea]
Disponible en: http://www.dynamicdrive.com/

CASAS, B., *Como documentar un PFC.* [En Línea]
Disponible en: http://www.lsi.upc.edu/~bcasas/docencia/pfc/manual_PFC.pdf

La normativa de EPSEVG relacionada con el PFC [En Línea]
Disponible en: http://www.epsevg.upc.edu/normatives/docs/Normativa-academica-delsestudis-de-1r-i-2n-cicle-de-EPSEVG.pdf

Com presentar una bibliografia, com citar un text. Barcelona, UPC. [En Línea]
Disponible en: http://bibliotecnica.upc.es/fullsacces/fulls/Serie5_4_Catala.pdf

Tutorial de PHP W3 Schools [En Línea]
Disponible en: http://www.w3schools.com/php

Página de referencias del PHP [En Línea] Disponible en: http://www.php.net

Tipografia del treballls, Servei de Llengües i Terminologia de la UPC [En Línea]
Disponible en: http://www.upc.edu/slt/itineraris/gram_est/gram_est.htm

MySQL 5.1: Reference Manual MySQL AB, 2008. [En Línea]
Disponible en: http://dev.mysql.com/doc/refman/5.1/en/index.html

Web para usada para validar el código XHTML y CSS [En Línea]
Disponible en http://validator.w3.org/