



Memoria PFC Videojuegos Educativos Juego online de preguntas y respuestas

Alberto Javier Vilches Ratón
2º ciclo de Ingeniería en Informática

Jordi Duch Gavalrà y Heliodoro Tejedor Navarro

6 de Enero de 2015

© Alberto Javier Vilches Ratón

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Juego online de preguntas y respuestas
Nombre del autor:	Alberto Javier Vilches Ratón
Nombre del consultor:	Jordi Duch Gavalrà y Heliodoro Tejedor Navarro
Fecha de entrega (mm/aaaa):	01/2015
Área del Trabajo Final:	Videojuegos educativos
Titulación:	2º ciclo de Eng. Informática
Resumen del Trabajo (máximo 250 palabras):	
<p>Videojuego online de preguntas y respuesta es una aplicación web que se puede jugar desde un navegador.</p> <p>Incorpora varios juegos que se crean automáticamente a partir de la información de países, banderas y capitales del mundo que hay en la wikipedia.</p> <p>Los jugadores deben responder una serie de preguntas tipo tests en las que se ofrecen varias respuestas en la que solo una es la correcta.</p> <p>Al acabar el juego obtendrá una puntuación y podrá ganar premios y subir puestos en un ranking con otros jugadores.</p>	
Abstract (in English, 250 words or less):	
<p>Game online answer & questions is a web application that you can play using a internet browser.</p> <p>The game load information from the wikipedia and creates his owns games about world countries and flags.</p> <p>Players have to answer with four questions but only one of them is the right one.</p> <p>When the player finishes the match, the game gives score and prizes to the player, so he can scale through a global ranking to the top.</p>	
Palabras clave (entre 4 y 8):	
Online, tests, juego, gamification	

Índice

1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	1
1.4 Planificación del proyecto.....	1
1.5 Breve resumen de productos obtenidos	2
2. Sobre el proyecto	3
2.1 Motivación del jugador.....	3
2.2 Diseño interno	3
2.3 Interfaz de usuario	5
2.4 Tecnologías utilizadas	6
2.5 Estructura del código fuente.....	6
2.6 Configuración y requisitos	7
2.6 Ejecución	7
2.7 Utilización del juego.....	7
2.8 Mecánica del juego.....	8
2.9 Tipos de juego	8
2.10 Obtención de preguntas para cada partida	9
2.11 Sistema de puntuación	9
2.12 Desbloqueo de premios (gamification)	10
3. Conclusiones	11
4. Glosario	12
5. Bibliografía.....	13
6. Anexos.....	¡Error! Marcador no definido.

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

La necesidad a cubrir es crear un juego online de preguntas y respuestas en el que se puedan crear juegos de manera automática a partir de scrappers. De esta manera, ya no será necesario crear cada pregunta y respuesta individualmente, sino que todas éstas se cargarán a partir de un proceso automatizado llamado web scrapping, que los lee de fuente de datos mayor.

1.2 Objetivos del Trabajo

El objetivo del trabajo realizado, como producto final, es que los jugadores aprendan conceptos y datos de un tema en concreto. Para ello, se le presentará al jugador preguntas con varias respuestas de las cuales solo una será correcta. Al fallar, se indicará la respuesta correcta y el jugador recordará la próxima vez que juegue consiguiendo con esto que aumenten sus conocimientos.

El objetivo personal consiste en aprender tecnologías de desarrollo nuevas, como el framework de desarrollo Grails y hacer scrapping de un sitio web como la Wikipedia.

1.3 Enfoque y método seguido

El enfoque consiste en desarrollar un producto nuevo apoyándose en frameworks existentes que permiten la creación rápida de aplicaciones web, como Grails.

Esta estrategia nos permite centrarnos exclusivamente en la lógica del producto sin tener que preocuparnos por la infraestructura http y acceso a base de datos. Un framework como Grails, basado en modelo-vista-controlador y con Hibernate para la persistencia de datos, es un sistema de creación de aplicaciones maduro, basado en Java, y por lo tanto, multiplataforma.

1.4 Planificación del proyecto

Hito nº 1: entrega del 12 de octubre.

Documento de diseño en el que se propone una definición inicial del proyecto a realizar.

Hito nº 2: entrega 16 de Noviembre. Primera versión del juego.

Desarrollar el modelo de datos y clases de dominio en el que guardar usuarios, juegos, partidas y puntuación.

Programar la versión inicial de la aplicación web Grails con el sistema de juego completo: selección de juego, secuencia de preguntas y respuestas, tiempo máximo para responder a cada pregunta.

Desarrollar un algoritmo de puntuación y de obtención de preguntas.

Desarrollar el web-scrapper que se conecte a la Wikipedia y que cree automáticamente las preguntas y respuestas a partir de la información de países: nombre de país, capital y bandera.

Elaboración de vídeo que presente el estado del desarrollo.

Entrega nº 3: 21 de diciembre. Versión jugable

-Trás un intento fallido de adaptar el juego a multijugador con websockets-

Gamification: programar en la aplicación un sistema de premios que se otorgan a los usuarios a medida que avanzan en el juego.

Crear un ranking de jugadores que reúna en una tabla la puntuación global de todos los jugadores.

Elaboración de vídeo que presenta el estado del desarrollo y como instalar y probar el juego.

Entrega nº 4 y final: 11 de enero. Código fuente, documentación y vídeo.

Corrección de bugs.

Elaboración de la memoria del proyecto y entrega del binario WAR desplegable en una aplicación web.

Elaboración de vídeo que presenta el estado del desarrollo y como instalar y probar el juego.

1.5 Breve sumario de productos obtenidos

pfc-avilches.war

Aplicación Web compatible con contenedor de Servlets como Apache Tomcat.

2. Sobre el proyecto

2.1 Motivación del jugador

A continuación se detallan cuales serán los puntos en los que se centrará el juego para motivar al jugador y que el juego tenga éxito.

Superación

La motivación se consigue con una interfaz gráfica agradable, dando puntos por respuestas correctas que luego el jugador puede guardar en el sistema y cuyo nombre aparecerá en una tabla de records. Esto debería motivar al jugador a volver a jugar para superarse a si mismo o a otros jugadores.

Recompensar la persistencia

Una puntuación creciente en la que dos respuestas correctas seguidas puntúan más que dos respuestas correctas sueltas, motivarán al usuario a repetir la partida con el fin de obtener más rachas de respuestas consecutivas.

Reconocimiento público: premios

Los jugadores podrán recibir premios como recompensa a medida que juega. Por ejemplo, si ha jugado 10 partidas se le da un premio o si acaba una partida sin un fallo. El objetivo es mostrar estos premios junto con el jugador de manera que le invite a volver a jugar hasta que los consiga.

Dificultad de las preguntas

La batería de preguntas y respuestas será determinante para que la dificultad del juego sea variable. A mayor cantidad y variedad, menos repetitivo será el juego y más posibilidades de acierto o error. El juego repetirá siempre un porcentaje fijo de preguntas al jugador que este haya fallado anteriormente. Con esto se garantiza que el jugador no se desmotive: una pregunta repetida tiene más posibilidades de acierto que una pregunta nueva, ya que el jugador podría recordar cuál era la respuesta correcta.

Presión

Finalmente, para aumentar la dificultad y para evitar que el jugador haga trampas buscando la respuesta en internet, habrá un tiempo limitado para responder cada pregunta. Si el jugador no responde en ese tiempo límite, perderá los puntos de esa pregunta.

2.2 Diseño interno

Internamente, el juego está dividido en las siguientes entidades y sistemas.

Usuario

Puede ser un jugador o un administrador que configura el sistema.

Jugador: puede registrarse, jugar y ver records

Administrador: además de jugar, puede crear nuevos juegos.

Están representados por la clase de dominio y tabla Usuario

Batería de preguntas y respuestas

Conjunto de datos que se utilizarán para los juegos. Serán cargados por un proceso externo que los obtiene de internet (Wikipedia, etc).

Están representados por la clase de dominio y tabla Pregunta y RespuestaPosible.

Sistema de carga de preguntas

Los scrappers son scripts que navegan por la Wikipedia descargando y procesando el html de sus páginas con el fin de crear baterías de preguntas y respuestas.

Está representado por la clase BackofficeController, que posee los action que se deben invocar para hacer la carga de juegos.

Juego

Consiste en la configuración de un juego y está formado por

- Un nombre (el que verá el jugador por ejemplo “Capitales (difícil)”)
- La batería de preguntas y respuestas a utilizar
- El número de preguntas a hacer (10, 20, 50...)
- El tipo de juego: heterogéneo u homogéneo. El número de respuestas incorrectas a mostrar por pregunta (2, 4, 8...)
- Estado (activo / inactivo)

Está representado por la clase de dominio y tabla Juego.

Se amplía esta información en el punto 2.9

Algoritmo de obtención de preguntas de cada juego

Durante la partida el sistema debe mostrar al usuario preguntas de la batería. Con el fin de mantener la dificultad equilibrada durante el juego, el sistema revisará las preguntas anteriores que ha respondido el usuario e intentará mostrar preguntas nuevas siempre, y si todas han sido respondidas, intentará mostrar las preguntas con más fallos antes que las que tienen menos fallos.

Se amplía esta información en el punto 2.10

Algoritmo de puntuación

Define el sistema en el que se asignan puntos al jugador en función de las respuestas correctas durante una partida.

Se amplía esta información en el punto 2.11

Está desarrollado en la clase PartidaService, en el método responde().

Partida

Es una tupla formada por un usuario y un juego a la que se añade la fecha en la que se juega la partida, la puntuación obtenida y el estado actual.

Está representado por la clase de dominio y la tabla Partida.

Premios disponibles

Son todos y cada uno de los premios que un jugador puede recibir. Para recibir un premio es necesario cumplir unos requisitos que todavía están por definir. El jugador podrá ver todos los premios que hay y una explicación sobre como conseguirlos.

Está representado por el enumerado TipoPremio y el servicio PremioService que contiene la lógica para otorgar premios.

Se amplía esta información en el punto 2.12

Premio otorgado

Es una tupla formada por un jugador y un premio disponible junto con la fecha en la que recibe el premio.

2.3 Interfaz de usuario

Página inicial

Al entrar, arriba a la derecha podrá introducir el login / password, para identificarse. Si ya está logado, aparecerá su nombre, los puntos totales y premios individuales recibidos. También habrá un botón de registrarse para crear nuevos usuarios.

En la parte central aparecerán todos los juegos creados y a la derecha los premios recibidos.

El usuario, solo si está logado en el sistema, elegirá un juego y, una vez confirmado que realmente desea jugar, empieza la partida.

Página de juego

Hay un marcador arriba con el número de preguntas pendientes, y va disminuyendo a cada paso hasta que llega a 0, momento en el que finaliza la partida.

En la parte de superior central se muestra en grande la pregunta y, opcionalmente, una imagen. Abajo aparecerán las respuestas disponibles.

Si el jugador acierta la respuesta, la respuesta correcta parpadea en verde. Tras 3 segundos, se pasa a la siguiente pregunta.

Si el jugador se equivoca, la respuesta incorrecta se pone en rojo parpadeando y la correcta en verde.

Un contador abajo a la derecha pone el tiempo que tiene el jugador para responder que disminuye. Cuando llega a 0, se desactivan los botones de respuesta y se resalta la respuesta correcta parpadeando. Tras 3 segundos, se pasa a la siguiente respuesta. Por debajo funcionaría igual que si el jugador elige una respuesta incorrecta.

Final de la partida

Cuando se llega al final, se muestra el número de puntos.

Se mostrará un ranking de usuarios con su puntuación en el que se pone la puntuación acumulada total y un porcentaje sobre el total posible ganada.

2.4 Tecnologías utilizadas

Para jugar será necesario una conexión a internet y un navegador.

- Grails como framework MVC web.
- Base de datos embebida H2
- Jquery y Twitter Bootstrap para diseño y css.
- Groovy y Htmlunit para el scrapping y generación de todas las preguntas.

2.5 Estructura del código fuente

La aplicación está desarrollada usando Grails 2.3.11. El código fuente está situado en la carpeta codigo-fuente. Al ser una aplicación Grails, debe tener una estructura predeterminada formada por:

```
▼ codigo-fuente
  ▼ grails-app
    ► conf
    ► controllers
    ► domain
    ► i18n
    ► migrations
    ► services
    ► taglib
    ► utils
    ► views
  ► scripts
  ► web-app
  ► wrapper
  application.properties
  grailsw
  grailsw.bat
```

grails-app/conf

Carpeta donde se almacenan los ficheros de configuración. Es importante el fichero DataSource.groovy donde se define el pool de conexiones JDBC contra la base de datos. En esta primera versión, está configurada para funcionar con H2, una base de datos embebida Java que almacena los datos en memoria. La ventaja es que no requiere instalar ni configurar nada. La desventaja es que al no escribir en disco los datos, estos son destruidos al apagar la aplicación.

grails-app/domain

grails-app/controller

grails-app/views

Grails usa el patrón MVC (model-view-controller) y en estas tres carpetas se ubican el código fuente del modelo de datos (domain), controladores (controller) y vistas.

(views, en formato gsp, groovy server pages)

web-app

Aquí se almacenan los recursos web que debe cargar el navegador, como css y javascript.

2.6 Configuración y requisitos

Java

No es necesario configurar nada ni tener Grails 2.3.11 instalado, tan solo un JDK con el ejecutable "java" o "java.exe" en el path del sistema. Con Java instalado, la aplicación es completamente autónoma por si misma y se descarga Grails y todas las dependencias necesarios de internet (requiere por lo tanto conexión a internet).

Puerto 8080 libre

La aplicación arranca un Tomcat que escucha por el puerto 8080, por lo que dicho puerto debe estar libre o dará un error de puerto ocupado. Se puede cambiar el puerto de arranque en el fichero grails-app/conf/BuildConfig.groovy modificando la siguiente línea:

```
grails.server.port.http = "8081" // Si se quiere cambiar el puerto a 8081
```

2.6 Ejecución

Usando un servidor contenedor de Servlets como Tomcat, copiar el fichero war entregado en la carpeta webapps.

Opcionalmente, se puede arrancar directamente a partir del código fuente con los siguientes comandos:

Situados en la carpeta codigo-fuente, se debe ejecutar el siguiente comando en Linux/Mac:

```
./grailsw run-app
```

O el siguiente comando en Windows:

```
grailsw.bat run-app
```

2.7 Utilización del juego

El juego arranca con una base de datos completamente vacía. Durante el bootstrap se crea un usuario para poder jugar cuyo login y password es:

Login: player
Password: player

Dado que al arrancar no hay juegos, estos deben ser creados llamando a un controlador específico que se encargará de conectarse a la Wikipedia y crearlos. Para ello, se debe visitar las siguientes url:

<http://localhost:8080/pfcuoc/backoffice/wikipedia>
<http://localhost:8080/pfcuoc/backoffice/demo>

La carga de juegos de la Wikipedia mostrará en los logs los países que se van cargando.

Una vez creados los juegos, se debe visitar la siguiente url

<http://localhost:8080/pfcuoc/>

Se introduce en la parte superior el login y password del usuario player/player y se puede elegir cualquier juego para jugar.

2.8 Mecánica del juego

- Cada partida es completamente funcional: se puede iniciar, responder preguntas fallando o correctamente y al acabar muestra una puntuación.
- Al fallar muestra la respuesta fallida en rojo y la respuesta correcta en verde.
- Al acertar muestra la respuesta en verde y un mensaje indicando el incremento de la puntuación.
- La puntuación es acumulable durante la partida y se aplica un factor de corrección llamado bonificación al acabar (se explica más adelante).
- El sistema avisa con un mensaje motivador cuando se aciertan preguntas en racha.
- Hay un tiempo limitado para responder cada pregunta. Si no se responde en ese tiempo, se da la pregunta por fallada.

2.9 Tipos de juego

Internamente los juegos se dividen en dos tipos en función de si sus respuestas son compatibles con el resto de preguntas o no:

Homogéneos

Son los juegos en los que hay tantas preguntas como respuestas, y todas las respuestas sirven para todas las preguntas. Por ejemplo, los juegos de países y capitales, o países y banderas. Al mostrar una pregunta de un juego homogéneo, se muestra su respuesta correcta y para las incorrectas, se muestran las respuestas correctas de cualquier otra pregunta.

Heterogéneos

Son los juegos en los que cada pregunta tiene sus propias respuestas (la correcta y varias incorrectas), que son incompatibles con el resto de preguntas.

2.10 Obtención de preguntas para cada partida

Las preguntas de cada partida se obtienen aleatoriamente, dando prioridad siempre a las que no se hayan realizado nunca al usuario actual.

Si se han usado todas, se da prioridad (se le asigna un peso mayor que hace que sea más probable que salgan) a las que se hayan fallado más veces. Para esto último, el número de fallos de una pregunta se obtiene sacando el número de aciertos menos el número de no aciertos. Esto se hace así porque si solo se usara el número de fallos, sería injusto que una pregunta que se haya fallado 5 veces y se haya acertado otras 5, saliera más a menudo (con más probabilidad) que una pregunta que se ha fallado 2 veces y no se ha acertado nunca. Por esto razón, lo que se hace es obtener el número neto de fallos (aciertos – fallos). El peso será como mínimo de 1, de manera que si la pregunta tiene más aciertos que fallos, en vez de dar un peso negativo, da un peso de 1. Ejemplo: si la pregunta tiene 4 fallos y 2 aciertos, tiene un peso de 2. Finalmente se obtiene una pregunta al azar teniendo en cuenta el peso de cada pregunta (a mayor peso, mayor probabilidad). Con el paso del tiempo, si una pregunta que tenía más fallos que aciertos pasa a tener los mismos o menos, dejará de salir con más probabilidad.

2.11 Sistema de puntuación

La puntuación es acumulable: una pregunta acertada es un punto, si se acierta dos veces seguidas, se dan dos puntos, etc. De esta manera, en un juego con 10 preguntas, si se aciertan todas se puede llegar a ganar $1+2+3+4+5+6+7+8+9+10 = 55$ puntos.

La puntuación final es el resultado de los puntos acumulados durante la partida multiplicado por el porcentaje de aciertos entre 10. Ejemplo:

- Acierto 8 preguntas de 20, lo que supone un 40% de aciertos. El bonus multiplicador será por lo tanto 4 ($40 / 10 = 4$)
- De las 8 preguntas respondidas, algunas las he hecho de manera consecutiva, por lo que durante la partida he acumulado 13 puntos
- La puntuación final será $13 \times 4 = 52$ puntos

¿Porqué se hace esto así? Dado que los puntos se acumulan cuando acertamos respuestas consecutivas, sería injusto que la puntuación final sea simplemente la puntuación obtenida durante el juego, ya que pocos aciertos consecutivos podrían una puntuación mayor que muchos aciertos no consecutivos.

Pongamos por ejemplo un juego de 10 preguntas. Un jugador A que acierte 3 preguntas de las 10, pero las haga todas seguidas, ganaría $1+2+3 = 6$ puntos. Otro jugador B que acertase 5 preguntas (dos más que el jugador A), pero las hiciera todas de manera no consecutiva (acierta/fallo 5 veces), obtendría solo 5 puntos (un punto menos que el jugador A). Aplicando el factor corrector de la bonificación, la puntuación de cada jugador quedaría:

Jugador A: $1+2+3 = 6$ puntos, 33% de aciertos = 20 puntos

Jugador B: $1+1+1+1+1 = 5$ puntos, 50% de aciertos = 25 puntos

El jugador B obtiene más puntos que el A ya que su porcentaje de aciertos es mayor.

2.12 Desbloqueo de premios (gamification)

Cuando se finaliza la partida están disponibles hasta cinco premios distintos. Los premios solo se pueden conseguir una vez y se muestran en la home. La idea es motivar al usuario a desbloquearlos y premiar su avance. Es posible añadir fácilmente nuevos premios añadiéndolos al enumerado TipoPremio y modificando el servicio PremioService.

1. Recluta, con el icono de una mochila.png. Se desbloquea al acabar el primer juego, sin importar la puntuación
2. Veterano, con el icono de un barco. Se desbloquea al terminar diez juegos.
3. Excelente, con el icono de un cubo de Rubik.png. Se desbloquea cuando se acaba un un juego con la máxima puntuación posible.
4. Cumlaude, con el icono de una orla.png. Se desbloquea cuando se acaban cinco juegos con la máxima puntuación posible.
5. Top, con el icono de una escalera. Se desbloquea cuando se consigue ser el primero del ranking al menos una vez.

3. Conclusiones

Lecciones aprendidas

Ha sido muy gratificante utilizar el framework de desarrollo Grails para la elaboración de aplicaciones web ya que permite, usando una única herramienta, crear un WAR desplegable en cualquier servidor de aplicaciones con MVC y acceso a datos.

La creación de scrappers que procesan páginas de la Wikipedia para extraer información ha sido muy interesante ya que abre las puertas a otros tipos de proyectos para el análisis de información en internet.

Reflexión crítica

El videojuego en si mismo era demasiado sencillo como para desarrollar el proyecto de fin de carrera , por lo que ha sido necesario complementarlo con características secundarias que lo enriquecieran técnicamente y funcionalmente.

Una de ellas era la implementación de un sistema de juegos multijugador con websockets, pero ha sido imposible.

De todas formas, creo que se han llegado a cumplir los objetivos porque el videojuego es completamente funcional y tiene el resto características secundarias que se sugirieron como gamification, ranking y los scrappers para cargar las preguntas.

Análisis crítico sobre la planificación del proyecto

Se ha seguido correctamente la planificación hasta la penúltima entrega que fue necesario retrasarla con el fin de que incluyera las características solicitadas de ranking y gamificación. El motivo fue la perdido de tiempo intentando hacer el juego multijugador.

Líneas de trabajo futura abiertas

1. Añadir más premios para enriquecer el juego tanto por la satisfacción que estos pueden proporcionar al usuario, como el reto que supone programarlos, ya que cada premio tiene unas reglas completamente distintas al resto y necesita su propio desarrollo.
2. Añadir más baterías de preguntas y respuestas a partir de nuevos scrappers. El motor del juego funciona a partir de las preguntas y respuestas que hay en su base de datos. Creando nuevos scrappers que alimenten dicha base de datos se pueden crear nuevos juegos sin tener que modificar el sistema.

4. Glosario

Gamification

Consiste en añadir reglas que para que la actividad realizada, en este caso responder preguntas tipo test, se asimile a un juego o a una actividad lúdica, pues consiguen la participación y el compromiso por parte de los usuarios a través de una sucesión de retos que han de superar.

Framework MVC

Es una herramienta para ayudar al desarrollo que se basa en el patrón modelo–vista–controlador (MVC), el cual separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Ranking

Listado de jugadores y puntos ordenado de manera descendiente por la puntuación de cada jugador, de manera que el primero del ranking es el que más puntos tiene.

5. Bibliografía

No se ha consultado ningún libro para la realización del proyecto, sino que toda la información obtenida se ha hecho a través de sitios web, los cuales a continuación se detallan.

Manual de Grails para la creación de un proyecto desde cero:

<http://grails.org/doc/latest/guide/gettingStarted.html>

Autores: Graeme Rocher, Paul King, Jeff Scott

Tutorial de Gorm para modelar las clases de dominio de la base de datos:

<http://timsporvic.github.io/GORM-Recipes/>

Autor: Tim Sporcic

Varios tutoriales de Grails juntos que abordan cada uno de ellos un punto clave para desarrollar una aplicación Grails. Ha sido necesario leer (varias veces) todos y cada uno de los tutoriales aquí citados:

<http://grails.asia/grails-tutorial-for-beginners/>

Autores: varios

Tutorial de HtmlUnit para crear scrappers para procesar la Wikipedia

<http://htmlunit.sourceforge.net/gettingStarted.html>

Autores: Mike Bowler y Christian Sell

Como leer tablas con HtmlUnit

<http://htmlunit.sourceforge.net/table-howto.html>

Autores: Mike Bowler y Christian Sell