

DOCUMENTACIÓN ENTREGABLE2

(Parte 1)

Arquitectura del Software (EII Oviedo)
Trivial3b

1-. Planteamiento del problema

Tras la creación de la aplicación que permitía procesar las preguntas para almacenarlas en una base de datos, la empresa NoGame tiene el objetivo de desarrollar un juego para ordenadores de sobremesa

El juego constará de una GUI (Graphical User Interface) que debe cumplir los siguientes objetivos:

- Permitir al usuario “loguearse” antes de comenzar a jugar para que se almacenen sus estadísticas personales. Puede haber tantos jugadores como colores tenga el tablero
- Mostrar un tablero que guíe la experiencia de juego del usuario, siguiendo las reglas básicas de los juegos tipo Trivial
- Los administradores podrán acceder a una serie de estadísticas organizadas por preguntas o categorías

Cabe destacar que el juego debe coger las preguntas de la base de datos construida en la primera fase. Además, la empresa no exige que el tablero sea redondo, aunque quiere que sea independiente de la funcionalidad para poder cambiar el “Look & Feel” en ejecución.

2-. Metodología usada

Tal y como se ha explicado en la asignatura de Arquitectura del Software, se va a utilizar el método ADD (Attribute-Driven Design) y la norma SEI (ANSI/IEEE 1471).

3-. Identificación de los Interesados

En nuestro caso, los interesados en el sistema son:

- La empresa NoGame
- Equipo de desarrolladores
- Usuarios del juego
- Administradores del juego

3.1-. La empresa NoGame

Se trata del equipo de directivos de la empresa de videojuegos NoGame, que ha ideado el desarrollo del juego y son los responsables de su financiación

Sus objetivos son:

- Conseguir que el coste del proyecto no sea excesivo
- Hacer que el sistema sea lo más escalable posible para poder aumentar la funcionalidad o los tipos de juegos que soporte el sistema sin volver a gastar dinero

3.2-. Equipo de desarrolladores

Este es el equipo responsable de desarrollar el sistema

Sus objetivos son:

- El sistema debe ser lo más mantenible posible, para evitar realizar grandes esfuerzos en caso de que la empresa quiera ampliar la funcionalidad

3.3-. Usuarios del juego

Se trata del conjunto de usuarios que han descargado el juego y cuyos objetivos son:

- Pasar un rato divertido gracias a una buena experiencia de juego
- Uso fácil e intuitivo
- Ver sus estadísticas de juego

3.4-. Administradores del juego

Son un grupo de trabajadores de la empresa que se encarga de llevar el control de las estadísticas que se van generando

Sus objetivos son:

- Poder ver con facilidad las estadísticas de juego, organizandolas por preguntas, por categorías, por usuarios...

3.5-. Lista actualizada de los Interesados (Stakeholders)

Código	Stakeholder	Intereses
ST-01	La empresa NoGame	Bajo coste de desarrollo Posibilidad de ampliar los objetivos de la aplicación
ST-02	Equipo de desarrolladores	Sistema fácil de mantener y ampliar
ST-03	Usuarios del juego	Buena experiencia de juego Sistema intuitivo y sencillo
ST-04	Administradores del juego	Facilidad para acceder a las estadísticas

4-. Identificación de los Atributos de Calidad

Para el sistema descrito se han identificado los siguientes atributos de calidad:

- Disponibilidad
 - Las preguntas guardadas en la base de datos deben estar disponibles en todo momento
- Modificabilidad
 - Facilidad de cambio de las reglas de juego, el tablero, el número de jugadores...
 - Facilidad para cambiar la apariencia de la interfaz del juego en tiempo de ejecución
- Usabilidad
 - El administrador debe poder ver las estadísticas de juego de forma clara y sencilla
 - Facilidad de uso del juego por parte del usuario, de forma que todas las opciones estén claramente explicadas
- Seguridad
 - Se debe contar con alguna medida de seguridad para evitar perder las preguntas ya procesadas y almacenadas en la base de datos
 - Debe existir una sincronización en el acceso a la BD de preguntas y de estadísticas en el servidor para evitar datos inconsistentes
 - Los usuarios clientes no pueden acceder a la funcionalidad exclusiva de los administradores (p.e. acceso a las estadísticas)
- Testabilidad
 - Debe ser posible probar y depurar la lógica del juego sin necesidad de usar la interfaz gráfica de usuario (GUI)
- Adaptabilidad
 - Se pretende que el juego funcione en diferentes plataformas (ordenadores con diferentes sistemas operativos, dispositivos móviles...)
 - Algunos de los módulos del juego pueden ser reutilizados en el futuro
- Time to market
 - La empresa quiere que el tiempo de desarrollo sea lo más bajo posible

Código	Descripción	Tipo de Atributo
AT001	Disponibilidad de la base de datos 24x7	Disponibilidad
AT002	Posibilidad de cambio de las reglas de juego, tipo de tablero, número de jugadores...	Modificabilidad
AT003	Facilidad de cambio de la apariencia de la interfaz del juego en tiempo de ejecución	Modificabilidad
AT004	Facilidad de consulta de las estadísticas por parte del Administrador	Usabilidad
AT005	Sencillez de uso del juego por parte del cliente	Usabilidad
AT006	Medidas de prevención contra la pérdida de los datos de las preguntas ya procesadas	Seguridad
AT007	Sincronización de accesos a la BD para evitar inconsistencias	Seguridad
AT008	Los usuarios no deben poder acceder a la funcionalidad exclusiva del Administrador	Seguridad
AT009	Facilidad de probar y depurar la lógica del juego sin necesidad de usar la interfaz gráfica de usuario (GUI)	Testabilidad
AT010	Posibilidad de querer ejecutar el juego en diferentes plataformas, ya sean otros sistemas operativos, dispositivos móviles...	Adaptabilidad
AT011	Posibilidad de necesitar reutilizar alguno de los módulos del juego para futuras variantes	Adaptabilidad
AT012	Tiempo de desarrollo corto	Time to market

4.1-. Atributos de calidad relacionados con los Interesados

Todos los atributos de calidad son de interés para alguno de los Stakeholders:

Atributos vs Interesados	ST-01	ST-02	ST-03	ST-04
AT001			X	X
AT002	X	X		
AT003		X	X	
AT004				X
AT005			X	
AT006	X	X		
AT007	X	X		
AT008	X	X		
AT009		X		
AT010	X		X	
AT011	X	X		
AT012	X	X		

5-. Escenarios de Calidad

Los escenarios de calidad del sistema son los siguientes:

Nº	Fuente	Estímulo	Entorno	Artefacto	Respuesta	Medida de la respuesta	Atributos de calidad afectados
1	Petición del usuario a la BD	Acceso a la base de datos	Explotación	Base de datos	Acceso a las preguntas almacenadas	La BD debe estar disponible el 95% del tiempo	AT001
2	Directivos de la empresa NoGame	Añadir nuevo tipo de tablero	Desarrollo	Sistema	Se crean los ficheros externalizados del grafo y los botones de la interfaz sin tocar la lógica	< 1 día para realizar los cambios. Ninguna línea de código de lógica modificada (0 líneas)	AT002 AT012
3	Directivos de la empresa NoGame	Modificación de las reglas del juego	Desarrollo	Sistema	Realizar los cambios sin alterar demasiado el sistema	Cambio realizado en el sistema en < 1 semana	AT002 AT012
4	Usuario	Cambiar la apariencia del tablero	Explotación	Interfaz	Cambio de la imagen de fondo en tiempo de ejecución	< 2 segundos, sin necesidad de reiniciar la aplicación	AT003
5	Administrador del juego	Obtención de estadísticas	Explotación	Sistema	Obtiene los datos de una forma completa y ilustrativa		AT004
6	Usuario	Comenzar a jugar	Explotación	Sistema	Manejo de la aplicación de forma sencilla e intuitiva		AT005
7	Fallo en la BD	Pérdida de datos de la preguntas y/o estadísticas almacenadas en la BD	Explotación	Base de datos	Recuperación mediante un backup	No se puede perder > 5% de preguntas y/o estadísticas de juego	AT006
8	Dos o más usuarios	Acceso concurrente a la BD de estadísticas para modificar un valor	Explotación	Base de datos	No se pierden datos ni quedan valores inconsistentes	Ningún error de datos inconsistentes	AT007
9	Usuario	Intento de acceso a la zona de estadísticas del administrador	Explotación	Sistema	El acceso es denegado y se muestra un mensaje de error	Ninguna forma de acceso sin tener los privilegios de administrador	AT008

10	Travis y Maven	Ejecución de test mediante integración continua	Explotación	Lógica de negocio	Comprobado el correcto funcionamiento de la lógica de negocio sin necesidad de GUI	El 100% de los test pasan correctamente y hay una cobertura de código $\geq 75\%$	AT009
11	Cucumber	Ejecución de test utilizando historias de usuario	Explotación	Lógica de negocio	Comprobado el correcto funcionamiento de la lógica de negocio sin necesidad de GUI	El 100% de los test pasan correctamente	AT009
12	Directivos de la empresa NoGame	Decisión de ejecutar el juego en otro SO	Desarrollo	Sistema	La aplicación se ejecuta sin ningún problema	Ejecución sin problemas en Windows, Mac y Linux (Siempre que tengan el jdk instalado)	AT010
13	Directivos de la empresa NoGame	Decisión de adaptar la aplicación a la Web	Desarrollo	Sistema	Desarrollo de la nueva interfaz del juego en la Web, reutilizando el código de lógica de negocio	Desarrollo en < 1 mes. Reutilización de todos los módulos de lógica de negocio y persistencia	AT011 AT012

6-. Descripción de la solución

Una vez descritos los atributos de calidad, se ha optado por una aplicación que sigue la arquitectura MVC (Model View Controller)

Tecnologías usadas

En primer lugar, cabe destacar que se ha elegido Java como lenguaje de programación y Eclipse como entorno de desarrollo debido a que todos los integrantes del equipo estamos familiarizados con ambos. De esta forma conseguiremos agilizar el proyecto

Para desarrollar la Interfaz Gráfica de Usuario (GUI) se ha optado por el plugin de Eclipse, WindowBuilder por razones similares al lo explicado en el párrafo anterior

Además, para conseguir cumplir el atributo de calidad de testabilidad (AT009) se van a utilizar las siguientes tecnologías:

- Travis
 - Vamos a utilizar este sistema distribuido de integración continua con la intención de que ejecute los Test de forma automática
- Cucumber
 - Utilizado para realizar pruebas en lenguaje “casi natural” sobre historias de usuario que describen funcionalidad implementada en la parte de lógica de la aplicación

Distribución y Estructura

Tal y como se verá reflejado con más detalle en el Diagrama de Despliegue posteriormente, hemos decidido que el sistema se estructure en 3 capas que se distribuyen en 2 “tiers” (Máquina local del usuario y Servidor de base de datos)

Siguiendo el patrón Model-View-Controller (MVC) el sistema se divide en:

- Interfaz Gráfica de Usuario (GUI) que cumpliría el rol de Vista
- Un Controlador que atiende a las peticiones de información que le hace la Vista y se apoya en el Modelo para resolverlas
- El Modelo, que a su vez hemos dividido en dos: Clases del modelo y Capa de persistencia

Se ha optado por dicha solución porque permite tener totalmente desacopladas las tres capas, pudiendo así hacer cambios en la Interfaz sin afectar a la lógica de negocio

- Para reducir el acoplamiento, se utilizarán los patrones Façade y Factory
- De esta forma se cumplen los requisitos no funcionales de modificabilidad y adaptabilidad

DIAGRAMA CONCEPTUAL

Figura 1 - Diagrama conceptual del problema

Por último, cabe destacar que el prototipo que se va a implementar guardará las estadísticas de juego en una base de datos en local para simplificar el problema. Por tanto, habría dos bases de datos: una local con las estadísticas y otra en un servidor con las preguntas

- Pero en la arquitectura ideada ambas estarían en un servidor para que así, el administrador pueda consultar

7-. Vistas

En la segunda parte de la documentación, se han descrito las siguientes vistas del sistema para una mejor comprensión del sistema. En la siguiente tabla se muestra un índice de las mismas, con los escenarios de calidad que resuelve cada una de ellas (además, aparecen los atributos de calidad y stakeholders relacionados).

Nombre	Tipo	Stakeholders	Atributos de calidad	Escenarios
Contexto	Contexto			
Parser	C&C			
Uploader	C&C			
Despliegue	Deployment			
Paquetes	Package Model			

Interfaz

Views: Módulo que interacciona con el usuario y pinta la GUI (Graphic User Interface)

Módulo que

Lógica

Game: Módulo que funciona como Controlador de la aplicación y que se encarga de delegar en otros módulos para crear y gestionar el tablero. Además, delega en el subsistema de persistencia para pedir preguntas a la BD y almacenar las estadísticas.

Model: Módulo que adapta el problema real a...

Persistencia

Estadísticas: Módulo que se encarga de la gestión (CRUD) de Jugadores y Estadísticas por jugador y por pregunta

Questions: Módulo que se encarga de pedir preguntas de manera aleatoria a la BD Mongo en función de una categoría