

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

¿Añadir Time-To-Market y Coste-Beneficio?

4.1-. Listado de atributos de calidad

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

4.2-. Atributos de calidad relacionados con los Interesados

Todos los atributos de calidad son de interés para alguno de los Stakeholders, tal y como refleja la siguiente tabla:

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

5-. Primer acercamiento a la solución

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

La estructura de la aplicación se dividiría en 3 capas totalmente diferenciadas

- Interfaz Gráfica de Usuario (GUI) que cumpliría el rol de Vista
- Capa de negocio que cumple los roles de Modelo y Controlador
- Capa de persistencia

Cabe destacar que se ha optado por dicha solución porque permite tener totalmente desacopladas las tres capas, permitiendo hacer cambios en la Interfaz sin afectar a la lógica de negocio. De esta forma se cumplen los requisitos no funcionales de modificabilidad

Además...

DIAGRAMA CONCEPTUAL

Figura 1 - Diagrama conceptual del problema

6-. 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 AT006
2	Ficheros fuente de preguntas	Utilizar un fichero con un formato distinto a GIFT	Explotación	Parser	Las preguntas se convierten a JSON sin ningún problema	El nº de preguntas fallidas se mantiene*	AT002 AT003
3	Directivos de la empresa NoGame	Modificación de los objetivos del juego	Desarrollo	Sistema	Realizar los cambios sin alterar demasiado el sistema	Cambio realizado en el sistema en < 1 semana	AT003
4	Operador de la compañía	Utilización de la aplicación	Explotación	Sistema	Un nuevo operador de la compañía no debe tener problema en comprender el funcionamiento del sistema	< 30 minutos para aprender a manejar todas las opciones	AT004
5	Pregunta expresada de forma incorrecta	Intentar parsear una pregunta incorrecta	Explotación	Parser	La pregunta no debe ser almacenada en la BD	No hay ninguna pregunta mal formada en la BD (0%)	AT005
6	JUnit	Ejecutar las pruebas	Desarrollo	Parser y BD	Permite al desarrollador localizar errores en el sistema	No se detecta ningún fallo y hay una cobertura > 75%	AT005 AT007
7	Fallo en la BD	Pérdida de datos de la preguntas almacenadas en la BD	Explotación	Base de datos	Poder recuperarse utilizando un Backup	No se puede perder más del 5% de las preguntas	AT006
8	Pregunta expresada de forma incorrecta	Intentar parsear una pregunta incorrecta	Explotación	Parser	Se debe mostrar un error que explique lo sucedido	No se afecta ninguna de las otras preguntas correctas (100%)	AT007

* Se refiere a que, el número de errores a la hora de convertir dichas preguntas a JSON no aumenta con respecto a si el fichero fuese GIFT

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	ST-01, ST-02, ST-03	AT003, AT004	3, 4
Parser	C&C	ST-01, ST-02, ST-03	AT002, AT003, AT005, AT007	2, 5, 6, 8
Uploader	C&C	ST-01, ST-03	AT001, AT006	1, 6, 7
Despliegue	Deployment	ST-01, ST-02, ST-03	AT001, AT004, AT006	1, 4, 7
Paquetes	Package Model	ST-01, ST-02, ST-03	AT002, AT003, AT005, AT007	2, 3, 6, 8