

# Trivial3b

## Entregable 3



Joel Alcalá Pérez	UO226321
Raúl Fernández Morán	UO180365
Daniel García García	UO231763
Sergio Llana Pérez	UO229817
Jairo López Soto	UO218230
Cristian Javier Navia Santos	UO229778
José Manuel García Fernández	UO231490

# Índice

## 1 - Manual de Usuario

1.1 - Comenzar a jugar	(Pág. )
1.1 - Opciones del usuario	(Pág. )
1.2 - Opciones del administrador	(Pág. )

## 2 - Documentación

2.1 - Planteamiento del problema	(Pág. )
2.2 - Metodología usada	(Pág. )
2.3 - Identificación de los interesados	(Pág. )
2.4 - Identificación de los Atributos de Calidad	(Pág. )
2.5 - Escenarios de calidad	(Pág. )
2.6 - Descripción de la solución	(Pág. )
2.7 - Vistas	(Pág. )

3 - Documentación CASE (Vistas)	(Pág. )
---------------------------------	---------

## **Documentación (Parte 1)**

### **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 decidió desarrollar un juego para ordenadores de sobremesa. Ahora, debido a su potencial, quiere que sus usuarios puedan jugar a través de la web.

El juego reutilizará toda la lógica del juego de la fase anterior, únicamente cambiando la capa superior de la arquitectura (la vista).

En cuanto al juego, los jugadores pueden iniciar partidas con otros jugadores en tiempo real, teniendo en cuenta que los jugadores pueden abandonar en medio de la partida

- Dos de los puntos más importantes del juego son la escalabilidad del sistema debido a muchos usuarios y rápidos tiempos de respuesta

Además, la empresa quiere asegurar la separación entre lógica de negocio e interfaz del usuario y apostar por un diseño Web responsivo.

### **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
  - El servidor de juego debe estar disponible 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
  - La página Web debe tener un diseño responsivo que permita adaptarse a cualquier tipo de pantalla, tablet o dispositivo móvil
  - Algunos de los módulos del juego pueden ser reutilizados en el futuro

- Rendimiento
  - Los tiempos de respuesta al internacional con la interfaz de usuario deben ser lo más cortos posibles
- 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	Disponibilidad del servidor de juego 24x7	Disponibilidad
AT003	Posibilidad de cambio de las reglas de juego, tipo de tablero, número de jugadores...	Modificabilidad
AT004	Facilidad de cambio de la apariencia de la interfaz del juego en tiempo de ejecución	Modificabilidad
AT005	Facilidad de consulta de las estadísticas por parte del Administrador	Usabilidad
AT006	Sencillez de uso del juego por parte del cliente	Usabilidad
AT007	Medidas de prevención contra la pérdida de los datos de las preguntas ya procesadas	Seguridad
AT008	Sincronización de accesos a la BD para evitar inconsistencias	Seguridad
AT009	Los usuarios no deben poder acceder a la funcionalidad exclusiva del Administrador	Seguridad
AT010	Facilidad de probar y depurar la lógica del juego sin necesidad de usar la interfaz gráfica de usuario (GUI)	Testabilidad
AT011	Página Web con diseño responsivo para que pueda ser accedido desde cualquier tipo de dispositivo	Adaptabilidad
AT012	Posibilidad de necesitar reutilizar alguno de los módulos del juego para futuras variantes	Adaptabilidad
AT013	Tiempos de respuesta cortos	Rendimiento
AT014	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	
AT007	X	X		
AT008	X	X		
AT009	X			
AT010	X	X	X	
AT011			X	X
AT012		X		
AT013	X		X	X
AT014	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	Usuario	Acceso al juego por medio de un navegador	Explotación	Sistema	Ejecución del juego sin problemas	La juego debe estar disponible el 95% del tiempo	AT002
3	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)	AT003 AT004 AT014
4	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	AT003 AT014
5	Usuario	Cambiar la apariencia del tablero	Explotación	Interfaz	Cambio de la imagen de fondo en tiempo de ejecución	< 1,5 segundos	AT004 AT013
6	Administrador del juego	Obtención de estadísticas	Explotación	Sistema	Obtiene los datos de una forma completa y ilustrativa	< 1 hora en comprender el funcionamiento de la aplicación	AT005

7	Usuario	Comenzar a jugar	Explotación	Sistema	Manejo de la aplicación de forma sencilla e intuitiva	< 1 hora en comprender el funcionamiento de la aplicación	AT006
8	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	AT007
9	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	AT008
10	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	AT009
11	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 $\geq 75\%$	AT010
12	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	AT010
13	Usuario	Ejecución del juego en un dispositivo móvil	Explotación	Interfaz	La aplicación se ejecuta sin ningún problema	Interfaz correcta en > 90% de los dispositivos móviles (Tablets, smartphones...)	AT011
14	Directivos de la empresa NoGame	Decisión de crear un nuevo juego de preguntas (aunque con otra temática)	Desarrollo	Sistema	Desarrollo de la nueva interfaz del juego.	Desarrollo en < 1 mes. Reutilización de todos los módulos de lógica de negocio y persistencia	AT012

## 6-. Descripción de la solución

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

- Además, también sigue el estilo Cliente-Servidor

### Tecnologías usadas

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

- También se utilizan HTML5, CSS y Scala para las vistas

Para facilitar la construcción de la aplicación Web siguiendo el patrón MVC, hemos utilizado el Framework de desarrollo rápido “Play”. Además, nos da herramientas para crear una API REST, controlar la seguridad...

Por último, con el objetivo de cumplir el atributo de calidad de Testabilidad, hemos utilizado las siguientes tecnologías:

- Travis
  - Vamos a utilizar este sistema distribuido de integración continua con la intención de que ejecuta los Test de forma automática
- Cucumber
  - Utilizado para hacer pruebas en lenguaje “casi natural” sobre historias de usuario que describen funcionalidad implementada en la parte de lógica de la aplicación
- Pruebas JUnit
  - Pruebas comunes de Java con las que probamos la lógica de negocio sin depender de las vistas de la aplicación gracias al patrón MVC

### 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 3 tiers (Cliente, Servidor Heroku y Servidor de base de datos)

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

- Conjunto de vistas HTML de la aplicación Web con las que interactuará el cliente
- Un conjunto de controladores (uno para la navegación entre vistas, otro para el juego...) que atienden a las peticiones de información apoyandose 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 las vistas sin afectar a la lógica de negocio

- De esta forma se cumplen los requisitos no funcionales de modificabilidad y adaptabilidad

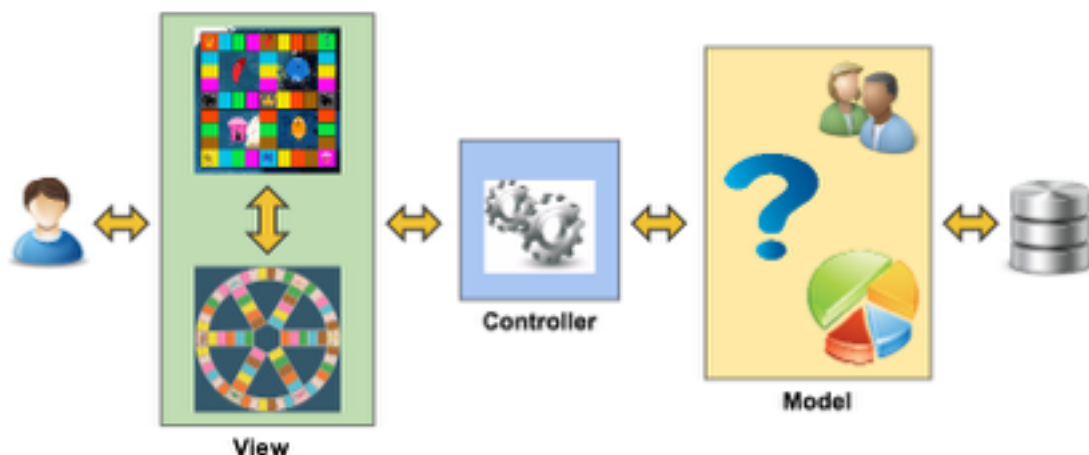


Figura 1 - Diagrama conceptual del problema



La aplicación estará desplegada en el servidor Heroku, que también contendrá la base de datos PostgreSQL de estadísticas y jugadores. Por otro lado, la base de datos de preguntas estarán en el servidor MongoLab

## **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).

<b>Nombre</b>	<b>Tipo</b>	<b>Stakeholders</b>	<b>Atributos de calidad</b>	<b>Escenarios</b>
Context	Contexto	ST-01, ST-02, ST-03	AT002, AT003, AT010, AT011, AT012	2, 4, 12, 13
View	C&C	ST-01, ST-02, ST-03, ST-04	AT002, AT003, AT004, AT005, AT008, AT012	2, 4, 5, 6, 9
Controller	C&C	ST-01, ST-02	AT002, AT008, AT009, AT012	3, 9, 10, 11
Model	C&C	ST-01, ST-02, ST-04	AT004, AT007	5, 8
Deployment	Despliegue	ST-01, ST-02, ST-03, ST-04	AT001, AT006, AT007, AT010, AT011, AT012	1, 7, 8, 12, 13
Packages	Paquetes	ST-01, ST-02	AT002, AT012	2, 3
BPM	BPM			