

# **DOCUMENTACIÓN ENTREGABLE1**

## **(Parte 1)**

### **1-. Planteamiento del problema**

Una empresa de videojuegos llamada “NoGame” tiene el objetivo de crear un juego de preguntas y respuestas similar al Trivial. Para ello, quieren construir una aplicación que lea las preguntas de un fichero, las procese y las almacene en una base de datos.

El primer paso del proceso es leer los enunciados y preguntas de un fichero con un formato concreto, GIFT. Aunque es posible que se quiera ampliar para procesar ficheros XML como QTI.

En segundo lugar, el procesamiento de las preguntas debe garantizar que no hay errores y por ello, se prescinde de IGU. La aplicación se ejecutará en dos partes controladas por un operador de la compañía. En la primera parte se deben pasar las preguntas a un formato intermedio (JSON) y en la segunda almacenarlas en la BD.

- Podrían realizarse ambas partes de forma automatizada cada cierto tiempo.

La empresa tiene pensado utilizar MongoDB como SGBD.

### **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
- Operador de la compañía
- Equipo de desarrolladores

#### **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-. Operador de la compañía**

Es uno de los trabajadores de la empresa que se encarga de llevar el control del proceso de análisis de los ficheros de preguntas y volcado a la base de datos

Sus objetivos son:

- El sistema debe permitir depurar el proceso para poder ver qué preguntas son erróneas.
- En caso de tener que usar más de un formato de fichero de entrada, el operador está interesado en poder configurar la aplicación sin mucho esfuerzo

#### **3.3-. 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

### 3.4-. 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	Operador de la compañía	Sencillez a la hora de descubrir errores Facilidad de configuración del sistema
ST-03	Equipo de desarrolladores	Sistema fácil de mantener y ampliar

### 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 del algoritmo de procesado de las preguntas (parser) para utilizar otros tipos de formato, tanto de entrada como de salida
  - Existe la posibilidad de querer ampliar el sistema para ser usado en distintas plataformas y distintos tipos de juegos de preguntas
- Usabilidad
  - Aunque el sistema no debe tener interfaz interactivo, el operador encargado de controlar el procesado de los datos debe poder realizar su tarea de manera sencilla, especialmente si debe alternar varios formatos de ficheros
- Rendimiento
  - Aunque no se exige que el procesado sea rápido, se debe garantizar la correcta conversión de las preguntas
- Seguridad
  - Se debe contar con alguna medida de seguridad para evitar perder las preguntas ya procesadas y almacenadas en la base de datos
- Testabilidad
  - El sistema debe ser fácil de probar, concretamente debe poder depurarse el proceso de análisis del fichero de entrada, para saber qué preguntas son incorrectas

#### 4.1-. Listado de atributos de calidad

Código	Descripción	Tipo de Atributo
AT001	Disponibilidad de la base de datos 24x7	Disponibilidad
AT002	Posibilidad de uso de varios formatos de entrada o salida	Modificabilidad
AT003	Escalabilidad del sistema, ya que es posible que se quieran ampliar los objetivos del sistema en el futuro	Modificabilidad
AT004	Facilidad a la hora de configurar el procesado de los ficheros y los distintos formatos	Usabilidad
AT005	Garantía de que las preguntas almacenadas den la BD son correctas	Rendimiento
AT006	Medidas de prevención contra la pérdida de los datos de las preguntas ya procesadas	Seguridad
AT007	Facilidad de depuración del proceso de carga de preguntas	Testabilidad

#### 4.2-. Atributos de calidad relacionados con los Interesados

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

<b>Atributos vs Interesados</b>	<b>ST-01</b>	<b>ST-02</b>	<b>ST-03</b>
AT001	X		X
AT002	X	X	X
AT003	X		X
AT004		X	X
AT005	X	X	
AT006	X		X
AT007		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 AT006
2	Directivos de la empresa NoGame	Modificación de los objetivos del juego	Desarrollo	Sistema	Realizar los cambios sin alterar demasiado el sistema	(Correcto diseño del programa para minimizar el tiempo y la cantidad de código a cambiar)	AT003
3	Ficheros fuente de preguntas	Utilizar un fichero con un formato distinto a GIFT	Explotación	Parser	Las preguntas se convierten a JSON sin ningún problema		AT002 AT003
4	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
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	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
7	JUnit	Ejecutar las pruebas	Desarrollo	Parser y BD	Permite al desarrollador localizar errores en el sistema		AT005 AT007
							AT004

## 6-. Primer acercamiento a la solución