1. Planteamiento del problema

La empresa NoGame dedicada a la creación de videojuegos va a crear una nueva línea de productos basados en juegos de preguntas/respuesta para diversas plataformas. En primer lugar se desea construir una variante del juego Trivial, aunque la idea es que puedan crearse más juegos de preguntas/respuesta similares en el futuro.

En una primera fase, desean hacerse con una base de preguntas suficiente para poder alimentar los diferentes juegos. Para ello, desean construir una sencilla aplicación que lea ficheros con los enunciados y respuestas de las preguntas, procese dichas preguntas indicando si hay errores o son correctas y las almacene en una base de datos.

Los ficheros con los enunciados de las preguntas se tomarán de bancos de preguntas ya existentes. Uno de los bancos de preguntas utiliza el formato GIFT. La aplicación utilizará una representación interna de las preguntas en formato JSON que facilitará su posterior almacenamiento en la base de datos. La compañía está considerando utilizar MongoDb, aunque esta decisión todavía no se ha tomado.

Se desea que la aplicación pueda ejecutarse en dos etapas. Una primera etapa que analiza los ficheros con las preguntas y genera el formato intermedio y otra etapa que toma las preguntas en dicho formato intermedio y las almacena en la base de datos. La ejecución de estas 2 etapas será controlada por un operador de la compañía que podrá decidir cuándo ejecuta cada etapa e incluso podrá automatizar dicha ejecución para que se realice cada cierto tiempo.

Es importante garantizar que la conversión es correcta, facilitando la depuración del proceso de carga de preguntas. Por ese motivo, se buscará una solución que no requiera interfaz interactivo pero que permita varias opciones de entrada (elegir nombre del fichero a cargar, elegir formato de la entrada, elegir nombre del fichero de salida, elegir formato del fichero de salida, etc.) y que permita observar los resultados intermedios de la conversión para chequear que no hay errores.

Puesto que la conversión puede automatizarse para que se realice en cualquier momento, la compañía no requiere que las aplicaciones sean muy eficientes ni que el proceso de conversión se realice de forma interactiva. En la primera fase la compañía solamente requiere la conversión de preguntas en formato GIFT, pero se está pensando que la solución debe admitir otros formatos en el futuro. Por ejemplo, la compañía está pensando procesar también preguntas en formatos XML como el formato QTI.

*Inserta aquí diagrama contextual —————*

2. Metodología usada

Se va a realizar un estudio de arquitectura siguiendo el método de ADD.

ADD es un método de diseño de arquitectura de modelo Top-don, y por lo tanto se basa en la descomposición de procesos en los atributos de calidad que el software ha de cumplir.

Por tanto el primer paso es identificar todos los atributos de calidad del sistema afectados por los requisitos no funcionales y modelarlos como escenarios de calidad.

A partir de estos escenarios y siguiendo las iteraciones del proceso ABC, se va refinando la arquitectura hasta tener una solución madura. Estas iteraciones son:

* + Creación del modelo de negocio para el sistema
  + 􏰀Comprensión de los requisitos
  + 􏰀Creación o Selección de la arquitectura
  + 􏰀Documentar y comunicar la arquitectura
  + 􏰀Análisis o evaluación de la arquitectura
  + 􏰀Implementación del sistema basado en la arquitectura
  + 􏰀Aseguramiento de que la aplicación se ajusta a la arquitectura

Adicionalmente, y principalmente al principio del proceso, se pueden crear diagramas contextuales u otros elementos que servirán para comunicar con los stakeholders en primera instancia. Y otros diagramas que se consideren necesarios.

En un primer momento aparecerán las vistas estructurales (estáticas) y posteriormente se le añadirán vistas dinámicas para añadir información a las anteriores.

Los pasos a seguir por tanto serán:

1. Elegir un módulo a descomponer

2. Refinar el módulo

3. Repetir estos pasos hasta que no se necesario una descomposición más profunda

3. Stakeholders

* **Responsables de la empresa NoGame**

Se trata de los equipos directivos de la corporación, son responsables de los presupuestos y toman las decisiones que comprometen fondos de dicho presupuesto.

Objetivos:

- Bajo coste en el desarrollo de la aplicación

* **Equipo de desarrollo del proyecto**

Equipo responsable de desarrollar el sistema resultante de la arquitectura

Objetivos:

- Proyecto rentable, que permita ser desarrollado con el precio establecido y un grado de rentabilidad que haga atractivo el desarrollo.

- Proyecto portable, que permita el cambio de plataforma sin que esto conlleve la modificación de la mayoría del código.

* **Usuarios de la aplicación**

Personas que utilizarán la aplicación en tiempo de explotación

Objetivos:

- Aplicación intuitiva, con una baja curva de aprendizaje

- Bajo coste en la adquisición del producto