Puntos a desarrollar

## Descripción del proyecto

NoGame, empresa dedicada a la creación y al desarrollo de videojuegos, se dispone a crear una nueva colección de productos que estarán basados en preguntas y respuestas y que servirán para distintas plataformas. Como primer videojuego de la colección, esta empresa quiere desarrollar uno similar al Trivial.

Para ello, se va a realizar una aplicación cuyo objetivo será la lectura de un fichero de las preguntas y respuestas que van a formar parte del videojuego, de forma que esta aplicación compruebe si hay algún tipo de error en las mismas, y por último, que almacene la información en una base de datos.

La colección de preguntas/respuestas que forman parte del producto a desarrollar se tomará de bancos ya existentes en formato GIFT (cabe la posibilidad de que se amplíe en un futuro a la lectura de los datos de un fichero XML como QTI) y, posteriormente, la aplicación desarrollada para manejar el funcionamiento del videojuego, utilizará una representación interna de las preguntas en formato JSON para facilitar el posterior almacenamiento de los datos en la BD. El SGBD que la empresa pretende utilizar en primera instancia es MongoDB.

La empresa NoGame quiere llevar a cabo el desarrollo del videojuego en dos etapas:

* + 1. Se analizan las preguntas de un fichero y se almacenan en un formato intermedio.
    2. Posteriormente se toman las preguntas en el formato intermedio y se almacenan en la BD.
* *Cabe destacar la necesidad de que exista una conversión correcta de la información en estos pasos, por lo que se prescindirá del uso de un interfaz interactivo.*

## Metodología utilizada (no hace falta incluirlo)

Tal y como se nos ha indicado en las clases de teoría de la asignatura, se va a seguir el método ADD (Atribute-Driven Design) acompañado de la norma del SEI (ANSI/IEEE 1471, 2000).

## Identificación de stakeholders (Descripción)

* Cliente(NoGame) :

Es la empresa NoGame, dueña del proyecto, es quizá el stakeholder con más poder dentro del proyecto ya que es quien autoriza y tiene la última palabra siempre, su deseo debe de verse satisfecho ante la solución informática que se le proporciona.

Entre sus objetivos están:

* 1. Obtener una aplicación sencilla pero funcional.
  2. La aplicación debe suponerle un valor añadido.
  3. Obtener la aplicación en el plazo estipulado.
* Desarrolladores(Equipo 2a):

Forman el núcleo de operaciones en la construcción del software , son los responsables de codificar, probar y mantener el código fuente del sistema a desarrollar.(¿desplegar, instalar, migrar datos ?)

Entre sus objetivos están:

* 1. Estabilidad técnica y de requisitos, es decir, que no haya demasiados cambios en los requisitos.
  2. El proyecto tiene que ser rentable.
  3. Realización a largo plazo, gozar de tiempo suficiente para la implementación.
* Administrador(Operador de la empresa):

Es el usuario el que ejecuta e interactúa con la aplicación indicándole los parámetros necesarios cuando esta no se ejecuta de forma automática.

Algunos de sus objetivos son:

* 1. Interactuar con una interfaz fácil de comprender y utilizar.
  2. No tener que estar resolviendo supuestos fallos de la aplicación.
* Arquitectos del sistema (Equipo 2a):

El equipo encargado de documentar, estructurar, diseñar y mantener el sistema informático que debe solucionar de la mejor forma posible, el proyecto propuesto por el cliente.

Entre sus objetivos están:

* 1. Garantizar un sistema reutilizable y de fácil mantenimiento.
  2. Encontrarse con unos requisitos fáciles de solucionar, es decir , que no sean peticiones imposibles de llevar a cabo.
  3. Proyecto rentable, atractivo y que aporte a su experiencia.

## Identificación inicial de los atributos de calidad

Para la aplicación que se va a desarrollar se han identificado los siguientes atributos de calidad:

* **Disponibilidad**: La aplicación debe poder ejecutarse en cualquier momento, pero sin las exigencias de un sistema 24/7.
* **Modificabilidad:** 
  + Facilidad de cambio del código de la aplicación para poder realizar cambios como la procedencia de los datos de una nueva fuente o el cambio en el funcionamiento interno de la misma.
  + Escalabilidad del producto de forma que en el futuro se puedan realizar las modificaciones pertinentes sin excesiva complejidad.
* **Rendimiento:** Se debe garantizar el correcto manejo de los datos de la aplicación, sin importar ni el tiempo de procesado de la aplicación ni que el proceso de conversión de datos no se realice de forma interactiva.
* **Seguridad:** Los datos en bruto y los datos procesados no tienen restricciones especiales de seguridad, pero los datos almacenados en la base de datos no deben poder ser accedidos por cualquiera.
* **Testabilidad:** 
  + Debe conseguirse un sistema sencillo para probar de forma que se garantice el correcto funcionamiento del mismo.
  + Debe garantizarse que la conversión al formato empleado por la aplicación es correcta y no se pierde la información en el proceso.
* **Usabilidad:** Debe garantizarse una facilidad de uso de la aplicación aun cuando esta no conste, como en este caso, de una interfaz de usuario interactiva.

## Primer acercamiento a la solución y riesgos

Una vez estudiados los requisitos y tras una serie de propuestas de solución, se opta por desarrollar un traductor en dos etapas mediante el patrón arquitectónico de *Pipe-and-filter*.

La primera etapa analizará los ficheros con las preguntas y generará un formato intermedio. Ésta etapa contendrá un analizador léxico y un analizador sintáctico que validarán el fichero de entrada y ayudarán a depurar las erratas que pueda contener. El formato intermedio será un AST (*Abstract Syntax Tree*).

La segunda etapa tomará las preguntas en formato intermedio y las procesará para poder almacenarlas en una base de datos. Estará compuesta por un *Visitor* que genere el formato final y un pequeño controlador que se encargará de añadir las preguntas a la base de datos.

Ésta solución facilita, mediante la sustitución de filtros, tratar diferentes formatos de entrada (y de salida, de ser necesario). También permite observar los resultados intermedios y una posible ejecución independiente de ambas etapas. Así la empresa tendrá una herramienta que podrá adaptar con relativa facilidad a todos sus productos similares.

## Riesgos relacionados con la solución

## Lista actualizada de stakeholders

|  |  |  |
| --- | --- | --- |
| Código | Stakeholder | Intereses |
| ST-01 |  |  |
| ST-02 |  |  |

1. Lista de atributos de calidad

|  |  |  |
| --- | --- | --- |
| Código | Descripción | Tipo de Atributo |
| AT001 | Disponibilidad razonable del sistema para procesar los archivos | Disponibilidad |
| AT002 | Facilidad de cambio de la fuente de entrada de los archivos. | Modificabilidad |
| AT003 | Facilidad de cambio del parseador o los algoritmos de este. | Modificabilidad |
| AT004 | Escalabilidad del sistema, es posible que en el futuro aumente el volumen de datos a procesar. | Modificabilidad |
| AT005 | Se debe asegurar la correcta realización de todos los trabajos. | Rendimiento |
| AT006 | Seguridad de acceso a la base de datos. | Seguridad |
| AT007 | Facilidad para probar la fiabilidad del sistema. | Testabilidad |
| AT008 | Garantía de conversión correcta al formato deseado de los archivos. | Testabilidad |
| AT009 | Facilidad al indicar los archivos a convertir | Usabilidad |

1. Atributos de calidad e interesados

|  |  |  |
| --- | --- | --- |
| Atributos  Stakeholders | ST-01 | ST-02 |
| AT001 |  |  |
| AT002 |  |  |

1. Descripción del negocio de la solución
2. Escenarios de calidad

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nº Escenario | Fuente de estímulo | Estímulo | Entorno | Artefacto | Respuesta | Medición de respuesta | Atributo de calidad |
| 1 |  |  |  |  |  |  | AT--- |
| 2 |  |  |  |  |  |  | AT--- |

1. Vistas (incluir en todas justificación)

### Vista del sistema (diagrama global de componentes)

### Diagrama de componentes (uno para cada subsistema- conectar siempre con puertos)

### Diagrama de paquetes

### Diagrama de despliegue

### Diagrama de secuencia

### Diagrama de clases (recomendó no utilizarlo)