**Trivial5b**

15

**Entrega 1**

**Arquitectura del software**

Skateholders

La identificación de los stakeholders de la aplicación es un paso a tener en cuenta. Estos stakeholders son aquellas personas o grupo de personas que van a interactuar de alguna forma con la aplicación. Las personas que hemos encontrado que van a utilizar la aplicación son:

1. Responsables de la empresa NoGrame

Se trata de las personas que financian la aplicación y los que se encargaron del proyecto, son los que establecen los requisitos generales que debe tener el proyecto para poder salir al mercado. Además, buscan que el coste y tiempo de desarrollo del proyecto sea lo más ajustado posible.

Como último, son los encargados del lanzamiento, promoción y aceptación de la aplicación. Para ello, podrán publicitar el juego en el mayor número de medios y plataformas posibles.

2. Equipo de desarrolladores del juego

Este equipo será el responsable de desarrollar el sistema resultante de la arquitectura. Para ello, crearán un Parser de GIFT y otro de JSON para las preguntas. Este grupo de personas desean que la aplicación sea por etapas, de esta forma en un futuro, se podrá convertir a más formatos. Las etapas en las que están divididas son dos:

1. Paso de GIFT a un formato intermedio.
2. Paso de este formato intermedio a un formato de salida.

Por último, este grupo de personas busca construir una aplicación sencilla, de baja intensidad tecnológica siendo un sistema fluido y predecible para los usuarios. No es necesario que el proceso de conversión se realice de forma interactiva.

3. Encargados de creación de preguntas

Desean hacer una base de preguntas suficiente para poder alimentar los diferentes juegos de la compañía. Los bancos de preguntas deben estar en formato GIFT. También, supervisan la ejecución de las etapas de generación del archivo JSON que posteriormente se almacenará en una base de datos.

4. Operador encargado del procesamiento de los datos

Controlará la ejecución de las dos etapas de ejecución de la aplicación. La primera etapa analiza los ficheros con las preguntas y genera el formato intermedio. La segunda toma las preguntas en dicho formato intermedio y las almacenará en la base de datos. También tiene que decidir cuándo ejecuta cada etapa e incluso podrá automatizar dicha ejecución para que se realice cada cierto tiempo. Este procesamiento de los datos podrá realizarse de forma manual o automática. Por último, garantizará que la conversión es correcta, de esta forma facilitara la depuración del proceso de carga de preguntas, sin utilizar la interfaz gráfica.

5. Usuarios del juego

Se trata de los usuarios del juego, los clientes que van a usar las aplicaciones. Deberán ser capaces de contestar al mayor número de preguntas de forma sencilla y eficiente.

6. Gestor de la base de datos

Se encargará de supervisar la seguridad en el acceso a los datos. Diseñará una base de datos fácil de mantener y escalable, dado que el número de usuarios puede crecer en cualquier momento. Además la empresa en cualquier momento puede querer crear nuevas aplicaciones que usen esa misma base de datos.

|  |  |  |
| --- | --- | --- |
| Código | Stakeholder | Interés |
| ST-01 | Responsable de NoGame. | Bajo coste de desarrollo, esto es, el desarrollo del proyecto debe ser corto y con un coste reducido. Bajo coste del proceso de cálculo: procesar grandes cantidades de datos a un coste muy reducido. |
| ST-02 | Desarrolladores del juego. | Baja intensidad tecnológica en el proyecto, con el fin de obtener un proceso de desarrollo controlado y predecible. Proyecto rentable, eso es, que permita ser desarrollado por el precio establecido con un grado de rentabilidad que haga atractivo el desarrollo. |
| ST-03 | Encargados de las preguntas. | Gran variedad de preguntas y una amplia cobertura de las mimas. |
| ST-04 | Encargados del procesamiento de datos. | Separación del proceso por fases y clara diferenciación de esas fases. Estas fases pueden ser utilizadas en otras aplicaciones. |
| ST-05 | Usuario cliente. | Facilidad de instalación y uso. Confiabilidad en la aplicación: que no provoque averías y que sea segura. Que no solicite aperturas de puertos al ordenador del usuario. |
| ST-06 | Gestor base de datos. | Facilidad frente al acceso a la base de datos y a su modificación y a la comunicación entre esta y las aplicaciones que deseen usarla. |

Requisitos funcionales

Los requisitos funcionales asociados a esta parte de la práctica son:

1. Las preguntas de Trivial y sus posibles respuestas estarán contenidos en un fichero con formato GIFT.
2. El sistema debe leer ficheros en formato GIFT y traducirlos a formato JSON.
3. El sistema debe almacenar las preguntas en formato JSON en una base de datos, preferiblemente MongoDB.
4. La aplicación debe adaptarse para permitir la traducción de otros formatos distintos de GIFT en el futuro.
5. La aplicación debe poder ejecutarse en dos etapas diferenciadas:
   1. La primera etapa será de traducción de GIFT al formato intermedio JSON.
   2. La segunda etapa será de almacenamiento de este formato intermedio en la base de datos.
6. Se debe permitir que cada etapa se ejecute en un momento concreto elegido por un operador de la compañía.
7. La conversión debe poder automatizarse para que se realice en cualquier momento y cada cierto tiempo.
8. La aplicación debe permitir las siguientes opciones de entrada:
   1. Elegir nombre del fichero a cargar.
   2. Elegir formato de la entrada.
   3. Elegir nombre del fichero de salida.
   4. Elegir formato del fichero de salida.
9. La aplicación debe permitir visualizar los resultados intermedios de la conversión con el fin de localizar errores.

Requisitos no funcionales

A continuación se describirán los atributos de calidad que tienen mayor importancia en esta primera fase del desarrollo del proyecto. La extracción y transformación de los datos de entrada irán aumentando a medida que vaya avanzando hacia una aplicación de escritorio o hacia una aplicación web, teniendo cada taxonomía unos atributos de calidad distintos a la otra.

Testabilidad

Facilidad para probar y comprobar la fiabilidad del sistema, con garantía de que las preguntas han sido transformadas correctamente en ambas etapas del proceso.

Integridad

Los datos de la aplicación deben estar siempre consistentes, completos y correctos.

Modularidad

El sistema actual será posteriormente una pequeña parte de un sistema más complejo, por lo que ha de estar pensado para que actúe de manera independientemente y que a la vez sea capaz de coexistir con otros sistemas menores.

Integración

Es un atributo importante de nuestro sistema ya que pretendemos incorporarlo en un contexto de aplicación más amplio. La integración de los datos abarca guardar los datos que una aplicación manipula de manera que otras aplicaciones puedan acceder a ellos.

Modificabilidad

El sistema ha de permitir la obtención de los datos de entrada con diferentes formatos por lo que se desea la posibilidad de incluir mayor funcionalidad en este aspecto en el futuro.

Disponibilidad

El operador encargado de la ejecución de las distintas etapas del proceso ha de ser capaz de iniciar cualquiera de ellas en cualquier momento que desee.

Robustez

Es la habilidad de un sistema para tratar los errores durante el proceso de ejecución. En nuestro caso sería, por ejemplo, el hecho de que en el proceso de tratamiento de los datos, tanto lectura como transformación, pudiera existir algún error y el usuario no debe ser consciente de que esto ha ocurrido, de manera que el sistema trate el error y continúe con una ejecución normal.

Usabilidad

No es de mucha importancia, pero queremos que el operador encargado de la ejecución de las distintas etapas cuente con una interfaz sencilla y usable.

Atributos de calidad

Partiendo de los requisitos no funcionales anteriormente comentados, partimos a extraer los atributos de calidad que albergan una mayor importancia en nuestra primera parte del sistema. Además, aparecerán nuevos atributos, de menor importancia pero necesarios.

|  |  |  |
| --- | --- | --- |
| **Código** | **Descripción** | **Tipo de atributo** |
| AT001 | Facilidad de prueba y comprobación de la fiabilidad en las distintas etapas. | Testabilidad |
| AT002 | Datos consistentes, completos y correctos. | Integridad |
| AT003 | Sistema independiente. | Modularidad |
| AT004 | Utilización de los datos en otra aplicación. | Integración |
| AT005 | Posibilidad de añadir mayor funcionalidad añadiendo nuevos formatos de entrada de datos. | Modificabilidad |
| AT006 | Disponibilidad para la ejecución de las etapas en cualquier momento. | Disponibilidad |
| AT007 | Tratamiento de errores durante la ejecución del proceso. | Robustez |
| AT008 | Interfaz sencilla y usable. | Usabilidad |
| AT009 | Bajo tiempo de desarrollo. | Time to market |
| AT010 | Bajo coste de desarrollo. | Relación Coste-Beneficio |

Los diferentes atributos de calidad son de interés para alguno de los Stakeholders. La siguiente tabla muestra la lista de intereses para el proyecto actual:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributos  vs  Interesados | ST-01 | ST-02 | ST-03 | ST-04 | ST-05 | ST-06 |
| AT-01 |  | X |  | X |  |  |
| AT-02 |  |  | X |  |  | X |
| AT-03 |  |  | X |  |  | X |
| AT-04 |  |  | X |  |  | X |
| AT-05 | X |  | X |  |  | X |
| AT-06 |  |  |  | X |  |  |
| AT-07 |  |  |  | X |  |  |
| AT-08 |  |  |  |  | X | X |
| AT-09 | X |  |  |  |  |  |
| AT-10 | X | X |  |  |  |  |

Escenarios de calidad

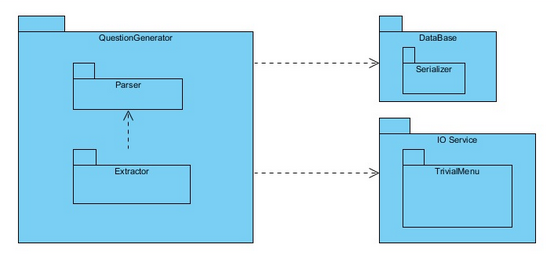
Partiendo de los atributos de calidad obtenidos, nos disponemos a buscar un caso concreto de ese atributo de calidad, es decir, un escenario de calidad. Se mostrará mediante la siguiente tabla con los siguientes campos:

1. **Fuente del estímulo**: quien o que genera el estímulo.
2. **Estímulo:** lo que se quiere llevar a cabo.
3. **Entorno**: condiciones dentro de las cuales se presenta el estímulo.
4. **Artefacto**: parte del sistema que recibe el estímulo.
5. **Respuesta**: actividad que ocurre luego de la llegada del estímulo.
6. **Medida de la Respuesta**: criterio para testear el requerimiento.
7. **Atributo de calidad afectado**: Atributo de calidad relacionado con el escenario.

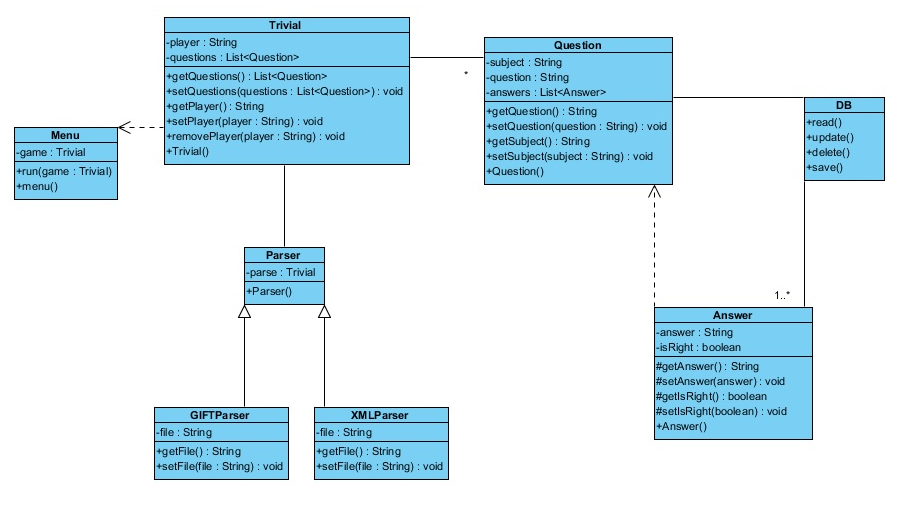
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº** | **Fuente de Estímulo** | **Estímulo** | **Entorno** | **Artefacto** | **Respuesta** | **Med. de la Respuesta** | **Atributo detectado** |
| 1 | Comprobación de que las etapas han finalizado correctamente. | Tratamiento de datos. | Desarrollo | Sistema | Ejecución de las pruebas. | Pruebas ejecutadas sobre las transformaciones correctas en un tiempo razonable. | AT001 |
| 2 | Obtención de datos persistentes. | Acceso a datos persistentes. | Explotación | Base de Datos (mongoDB) | Datos obtenidos correctamente. | Datos completos, consistentes y correctos. | AT002 |
| 3 | Aplicación puede ser utilizada por sistemas externos. | Sistema independiente. | Despliegue | Aplicación | Sistema funcionando en dos sistemas diferentes sin problemas. | N/A | AT003 |
| 4 | Reutilización de los datos tratados en el proceso. | Uso de los datos transformados. | Desarrollo | Sistema | Datos en el formato necesario en la aplicación receptora. | Datos correctos para su utilización. | AT004 |
| 5 | Ampliación de funcionalidad en la obtención de datos de fichero. | Obtención de datos. | Explotación | Sistema | Funcionalidad añadida al sistema con otros formatos. | N/A | AT005 |
| 6 | Ejecución de las diferentes etapas por el operador. | Transformación de datos. | Explotación | Sistema | Proceso de ejecución de la etapa seleccionada. | Ejecución inminente de la etapa seleccionada. | AT006 |
| 7 | Tratamiento de errores durante la ejecución del proceso. | Procesamiento de datos. | Explotación | Sistema | Ejecución normal incluso con errores. | Fluidez en la ejecución. | AT007 |
| 8 | Utilización de la interfaz de selección de etapas por el operador. | Transformación de datos. | Explotación | Sistema | Ejecución de etapas de manera sencilla. | El uso ha de ser rápido e intuitivo. | AT008 |
| 9 | Bajo tiempo de desarrollo. | N/A | Desarrollo | Sistema | N/A | 4 semanas. | AT009 |
| 10 | Bajo coste de desarrollo. | N/A | Desarrollo | Sistema | N/A | Sin gastos. | AT010 |

Diagramas

# DIAGRAMA DE PAQUETES



# DIAGRAMA DE CLASES



# DIAGRAMA DE COMPONENTES

# 

# DIAGRAMA DE DESPLIEGUE

# 