

Trivial5b

Javier Camacho Suárez

Carla Costa Suárez

Adela Gutiérrez Inés

Sergio López González

Daniel Mera López

Alejandro Rico González

Pablo Rodríguez Valdés

Marcos Ruiz Villameà

Cristóbal Solar Fernández

Entrega 1

Arquitectura del software



15

Contenido

1. Manual de usuario	3
2. Stakeholders.....	4
2.1. Responsables de la empresa NoGame.....	4
2.2. Equipo de desarrolladores del juego	4
2.3. Encargados de creación de preguntas.....	4
2.4. Operador encargado del procesamiento de los datos.....	5
2.5. Usuarios del juego	5
2.6. Gestor de la base de datos.....	5
3. Requisitos funcionales	6
4. Requisitos no funcionales.....	7
4.1. Testabilidad	7
4.2. Integridad.....	7
4.3. Modularidad	7
4.4. Integración.....	7
4.5. Modificabilidad	7
4.6. Disponibilidad	7
4.7. Robustez	8
4.7. Usabilidad.....	8
5. Atributos de calidad.....	9
6. Escenarios de calidad.....	10
7.1. Diagrama de paquetes	12
7.2. Diagrama de clases.....	12
7.3. Diagrama de componentes	13
7.4. Diagrama de despliegue.....	13

1. Manual de usuario

En esta primera parte se desea construir una sencilla aplicación que lea ficheros con los enunciados y respuestas de las preguntas, procese dichas preguntas y las almacene en una base de datos. Los archivos con los enunciados y las respuestas se tomarán de bancos de preguntas ya existentes. Estos bancos pueden estar en formato GIFT o XML. Para poder almacenar las preguntas en una base de datos, primero deben estar en un formato apto para ella. En este caso el sistema de bases de datos escogido es MongoDB y el formato en el que las preguntas deben estar es JSON.

La aplicación, en este caso, obtiene un fichero (XML o GIFT), transforma las preguntas a un formato intermedio y este se transforma en un objeto JSON. Para poder ejecutar el programa desde el eclipse los archivos con las preguntas a transformar deben estar en el directorio Data, ya que el sistema obtiene los archivos de ahí.

Para comenzar la ejecución del programa, arrancamos el programa desde la clase Extractor y aparecerá en la consola un pequeño menú.

```
Wellcome to Trivial Extractor
=====
|  SELECCIONE UNAOPCION  |
=====
| Options:                |
|      1. GIFT-JSON       |
|      2. XML-JSON        |
|      3. Salir           |
=====
Escoja una de las acciones disponibles:
```

Figura 1. Menú de la aplicación

El menú consta de dos opciones principales:

1. GIFT-JSON: se encarga de convertir el fichero con preguntas GIFT a formato JSON.
2. XML-JSON: se encarga de convertir el fichero con preguntas XML a formato JSON.

En función de la opción escogida el programa imprimirá todas las preguntas del formato original, en formato JSON.

2. Stakeholders

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, Tabla 1. Las personas que hemos encontrado que van a utilizar la aplicación son:

2.1. Responsables de la empresa NoGame

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.

2.2. 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 desea que la aplicación sea por etapas, de forma que 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.

2.3. 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.

2. 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 facilitará la depuración del proceso de carga de preguntas, sin utilizar la interfaz gráfica.

2. 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.

2. 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 mismas.
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.

Tabla 1. Representación de los Stakeholders

3. 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:
 - 5.1. La primera etapa será de traducción de GIFT al formato intermedio JSON.
 - 5.2. 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:
 - 8.1. Elegir nombre del fichero a cargar.
 - 8.2. Elegir formato de la entrada.
 - 8.3. Elegir nombre del fichero de salida.
 - 8.4. Elegir formato del fichero de salida.
9. La aplicación debe permitir visualizar los resultados intermedios de la conversión con el fin de localizar errores.

4. 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.

4.1. 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.

4.2. Integridad

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

4.3. 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 independiente y que a la vez sea capaz de coexistir con otros sistemas menores.

4.4. 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.

4.5. 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.

4.6. 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.

4.7. 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.

4.7. 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.

5. 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, Tabla 2. 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

Tabla 2. Atributos de calidad

Los diferentes atributos de calidad son de interés para alguno de los Stakeholders. La Tabla 3 muestra dicha relación.

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				

Tabla 3. Relación Stakeholders y atributos de calidad

6. 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. En la Tabla 4 se encuentran todos los escenarios encontrados junto con las siguientes características:

1. **Fuente del estímulo:** Quién o qué 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.

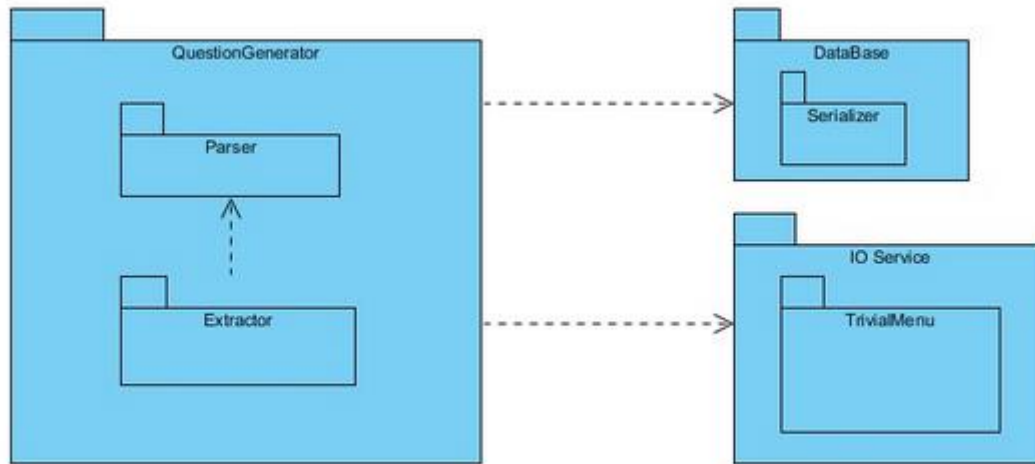
Atributo detectado	Med. de la Respuesta	Respuesta	Artefacto	Entorno	Estímulo	Fuente de Estímulo	Nº
AT001	Pruebas ejecutadas sobre las transformaciones correctas en un tiempo razonable.	Ejecución de las pruebas.	Sistema	Desarrollo	Tratamiento de datos.	Comprobación de que las etapas han finalizado correctamente.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

AT009	4 semanas.	N/A	Sistema	Desarrollo	N/A	Bajo tiempo de desarrollo.	9
AT010	Sin gastos.	N/A	Sistema	Desarrollo	N/A	Bajo coste de desarrollo.	10

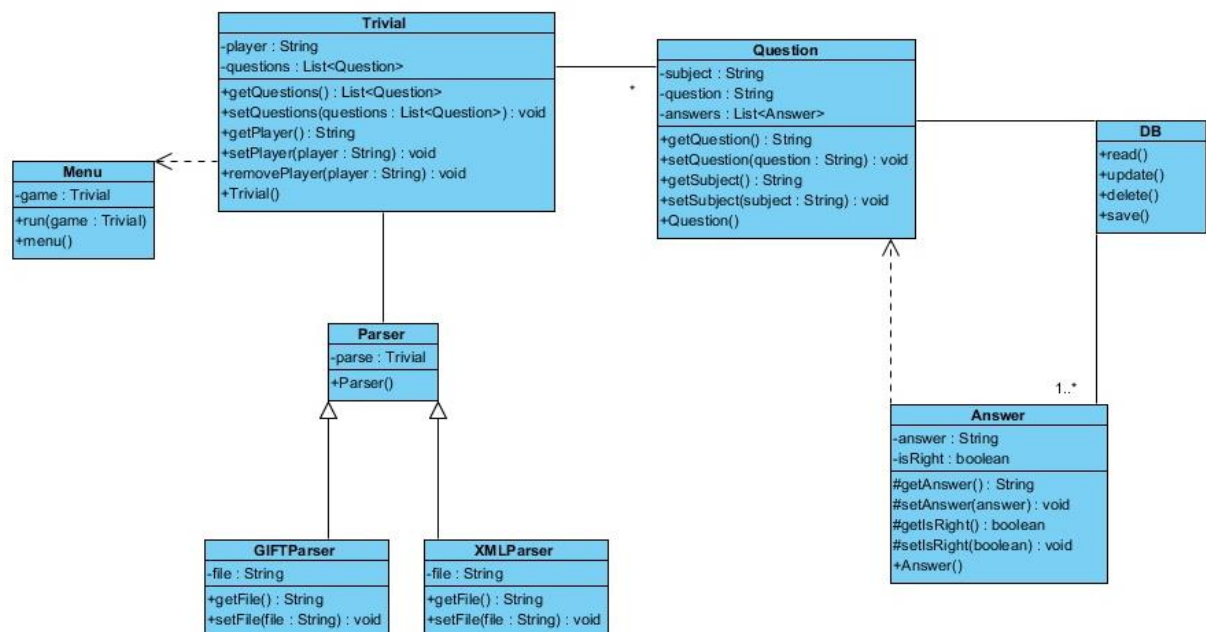
Tabla 4. Escenarios de Calidad

7. Diagramas

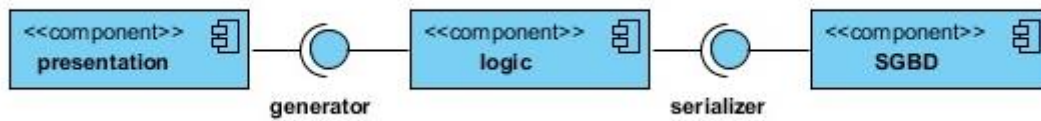
7.1. Diagrama de paquetes



7.2. Diagrama de clases



7.3. Diagrama de componentes



7.4. Diagrama de despliegue

