Grado en Ingeniería Informática del software

Grupo: Trivial 1A  
Sergio Cueto López de Bustamante  
Ignacio Rodríguez Vázquez  
Yenni Ramos Martínez  
Diego Jaular Ortigueira  
Robert Stefanita Ene  
Alejandro García Torriello  
Isabel Del Álamo Rancaño  
Francisco Gil Gala

Documentación entregable 2

Arquitectura del Software



Contenido

[Planteamiento del Problema 3](#_Toc416292508)

[Requisitos Funcionales 3](#_Toc416292509)

[Requisitos No Funcionales 4](#_Toc416292510)

[Identificación de los Interesados (Stakeholders) 4](#_Toc416292511)

[Atributos de Calidad 5](#_Toc416292512)

[Atributos de calidad e Interesados 6](#_Toc416292513)

[Escenarios de Calidad 7](#_Toc416292514)

[Vistas 8](#_Toc416292515)

Este documento incluye:

* Una identificación de stakeholders, atributos y escenarios de calidad para el problema propuesto, junto con la relación que guardan.
* La arquitectura optada que soluciona los requisitos pedidos, explicada y detallada.
* El manual de usuario, indicando lo necesario para que se pueda manejar la aplicación.
* Un manual de sistema, indicando aspectos importantes de configuración y restricciones para el correcto funcionamiento de la misma

# Planteamiento del Problema

Se pide desarrollar un juego que funcione como una aplicación tradicional en ordenadores de sobremesa. Se pide que cumpla con los siguientes requisitos:

* El juego deberá tomas las preguntas de la base de datos que se ha desarrollado en la primera fase. Se mostrará el juego con una interfaz gráfica.
* El tablero oficial cuenta con una rueda de 6 radios, se podrá presentar una versión simplificada.
* Los jugadores se podrán registrar guardando usuario y contraseña.
* La máquina ira generando tiradas de dado para cada jugador, les preguntaran a que casilla quieren moverse y mostrará la pregunta que corresponda según la categoría de la casilla.
  + Si un jugador acierta->se le asigna el quesito correspondiente a la casilla en la que este.
  + Si un jugador no acierta->el jugador pierde el turno y pasa a ser el turno del siguiente jugador.
* Finalmente para que un jugador gane deberá ir a la casilla final con todos los quesitos, es decir con el pastel completo.
* La compañía pide que cada persona pueda ver sus estadísticas de preguntas acertadas y falladas para que sea una aplicación de aprendizaje.
* Se pide que este separado claramente la visualización del juego con el resto de la funcionalidad. Los usuarios pueden modificar la apariencia de la aplicación cuando ellos lo deseen.

# Metodología utilizada

Como se ha visto en clase de Arquitectura del Software el método que ha sido utilizado es el Atribute-Driven-Design (ADD).

El Atribute-Driven-Design (ADD) es un método que se basa en descomponer los procesos en los atributos de calidad que el software deberá cumplir. Se definen los atributos de calidad a partir de los requisitos no funcionales finalizado con el modelado de los escenarios de calidad.

# Requisitos Funcionales

1. El número de jugadores que podrán participar en cada partida será como mínimo 1 y como máximo 6.
2. Posibilidad de registro como usuario y autenticación ante el sistema mediante nombre de usuario y contraseña.
3. Guardado de estadísticas (porcentaje de preguntas acertadas y falladas clasificadas por categorías y por usuarios) de todas las partidas de los usuarios para fines educativos.
4. Funcionamiento de la aplicación en un futuro en diferentes plataformas y para varias visualizaciones de la misma.

# Requisitos No Funcionales

* 1. Debe poder correr en paralelo varias instancias de la aplicación a la vez.
  2. Se le debe poder comunicar a la aplicación por consola que formato va a leer.

# Identificación de los Interesados (Stakeholders)

* Responsables de NoGame: Se trata de los equipos directivos de la empresa, son responsables de los presupuestos y toman las decisiones que comprometen fondos de dicho presupuesto.
* Equipo de desarrollo del proyecto Trivial: Este equipo será el responsable de desarrollar el sistema resultante de la arquitectura.
* Responsables de la información: Este equipo se encargará de introducir la información con las preguntas que utilizarán los distintos juegos, además de comprobar su veracidad y actualizarlas en caso de cambios.
* Usuario cliente: Serán los encargados de jugar con la aplicación.

|  |  |  |
| --- | --- | --- |
| Código | Stakeholders | Intereses |
| ST-01 | Responsables de NoGame | 1. Bajo coste económico de desarrollo. El desarrollo del proyecto debe ser corto y con un coste reducido. 2. Posibilidad de que el juego sea multiplataforma. 3. Posibilidad de crear más juegos con formato preguntas/respuestas, extendiendo el que se está diseñando actualmente. |
| ST-02 | Equipo de desarrollo del proyecto Trivial | 1. Baja complejidad en los algoritmos de aleatorización de las preguntas. |
| ST-03 | Responsables de la información | 1. Proyecto rentable, esto es, que permita ser desarrollado por el precio establecido con un grado de rentabilidad que haga atractivo el desarrollo. 2. Permitir la ampliación de la aplicación con nuevas partes de manera sencilla al proyecto. 3. Máxima facilidad y rapidez a la hora de introducir preguntas, preferiblemente con tratamientos por lotes y no individualmente. 4. Máxima facilidad y rapidez para buscar una determinada pregunta. 5. Máxima facilidad y rapidez para actualizar los datos de una pregunta. 6. Poder mantener la integridad y coherencia en la información. |
| ST-04 | Usuario Cliente | 1. 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 |
|  | | |

# Atributos de Calidad

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:

|  |  |  |
| --- | --- | --- |
| Código | Descripción | Tipo de Atributo |
| AT001 | Posibilidad de cambio de visualización | Modificabilidad |
| AT002 | Posibilidad de cambio de número de jugadores | Modificabilidad |
| AT003 | Las preguntas guardadas en base de datos deberán estar disponibles durante la partida. | Disponibilidad |
| AT004 | El acceso a los usuarios registrados deberá ser accesible en un tiempo óptimo. | Rendimiento |
| AT005 | El acceso a las preguntas almacenadas en la base de datos deberá producirse en un tiempo óptimo y de una manera correcta. | Rendimiento |
| AT006 | Cada usuario podrá ver solamente sus estadísticas. | Confidencialidad |
| AT007 | La interfaz debe ser intuitiva | Usabilidad |
| AT008 | Consistencia de las preguntas en la base de datos | Seguridad |
| AT009 | El acceso a nuestra aplicación se hará por medio de un registro por ello los jugadores no registrados no podrán jugar. | Seguridad |
| AT010 | Aplicación fácilmente depurable | Testabilidad |
| AT011 | Se pide que este separado claramente la visualización del juego con el resto de la funcionalidad, para que el juego pueda ser utilizado en diferentes plataformas. | Adaptabilidad |
| AT012 | Algunos módulos pueden ser reutilizados en un futuro | Adaptabilidad |
| AT013 | Tiempo de desarrollo de la aplicación que sea corto | Time to market |

# Atributos de calidad e Interesados

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 |
| AT001 | x | x |  | x |
| AT002 | x | x |  | x |
| AT003 | x | x | x | x |
| AT004 | x | x |  | x |
| AT005 | x | x |  | x |
| AT006 | x | x |  | x |
| AT007 | x | x |  | X |
| AT008 | x | x | x |  |
| AT009 | x | x |  | x |
| AT010 |  | x |  |  |
| AT011 | x | x |  |  |
| AT012 | x | x |  |  |
| AT013 | x | x |  |  |

# Escenarios de Calidad

Con toda la información anterior se procederá a definir los escenarios de calidad que influencian esta arquitectura.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Escenario | Fuente  de estímulo | Entorno | Artefacto | Respuesta | Medición  de la respuesta |  | Atributos de calidad afectado |
| 1 | Usuario | Explotación | Interfaz | Cambio de vista y usuarios | Tiempo no superior a 5s |  | AT001, AT002 |
| 2 | Responsables de Información | Explotación | Base de Datos | Disponibilidad de preguntas | Tiempo no superior a 2s |  | AT003 |
| 3 | Usuario | Explotación | Aplicación | Inicio de sesión y desarrollo de partidas rápidas | Tiempo no superior a 1-2s |  | AT004, AT005 |
| 4 | Usuario | Explotación | Interfaz | Listado de estadísticas del usuario en sesión | Tiempo no superior a 1s |  | AT006 |
| 5 | Usuario | Explotación | Interfaz | Transiciones y fluidez entre las vistas de la aplicación | Tiempo no superior a 3-4s |  | AT007 |
| 6 | Responsables de Información | Desarrollo | Base de Datos | Consistencia en los datos (preguntas) de la Base de Datos | Tiempo no superior a 1s |  | AT008 |
| 7 | Usuario | Explotación | Aplicación | Registro de los jugadores en el sistema. | Tiempo del registro del usuario (5-10s) |  | AT009 |
| 8 | Desarrolladores | Desarrollo | Aplicación | Módulos de la aplicación depurables. | Tiempo relativamente corto en la localización del bug. |  | AT010 |
| 9 | Desarrolladores | Desarrollo | Aplicación | Módulos de la aplicación reutilizables para distintas plataformas. | Tiempo de respuesta en localizar el módulo a reutilizar. |  | AT011, AT012 |
| 10 | Responsables de NoGame | Desarrollo | Aplicación | Tiempo de desarrollo corto | Tiempo del desarrollo (un mes) |  | AT013 |

\*\*En el manual de usuario, cuando no hay base de datos (mongodb) hay una base de datos simulada que ira como el culo

# Vistas

|  |  |  |  |
| --- | --- | --- | --- |
| Vistas | Stakeholders | Atributos de calidad | Escenarios |
| Logging | ST-02, ST-03 | AT002, AT005, AT003 | 2 |
| Persistencia | ST-01, ST-02,  ST-03 | AT004, AT002, AT005, AT003 | 2, 4 |
| Vista procesado | ST-02, ST-03 | AT001, AT006, AT002, AT005, AT003 | 1, 2, 3 |