**Documento de Arquitectura Questik**

**Atributos de Calidad**

1. **Disponibilidad (Escalabilidad horizontal):** : Se trata de la capacidad de un servicio, de unos datos o de un sistema, a ser accesible y utilizable por los usuarios (o procesos) autorizados cuando estos lo requieran.
   1. **Escenario:** Esto se logra gracias al loadbalancer que se encuentra dentro de azure, este nos proporciona una baja latencia y un alto rendimiento de la aplicación. El load balancer, distribuye el trafico y las peticiones que llegan al front end a las instancias en el back end.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Origen** | **Estimulo** | **Ambiente** | **Respuesta** | **Medida** |
| Usuarios | Varias peticiones a la aplicación | Sobrecarga de peticiones | Questik puede responder a las diversas peticiones | Disponibilidad del 98% |

1. **Seguridad (Autorización y Autenticación):** Medida de la habilidad del sistema para resistir usos no autorizados, sin dejar de prestar el servicio a los usuarios legítimos
   1. **Escenario:** El administrador es la única persona que puede iniciar un juego, un usuario común, no tiene acceso a esto, no pueden ni ver los cuestionarios disponibles, ni iniciar el cuestionario creado por el administrador, esto se logra con un mecanismo de inicio de sesión que brinda acceso a datos y privilegios dentro de la plataforma al usuario.  
      En caso de que un usuario intente forzar el acceso mediante un url, esta lo redirecciona a la página principal.

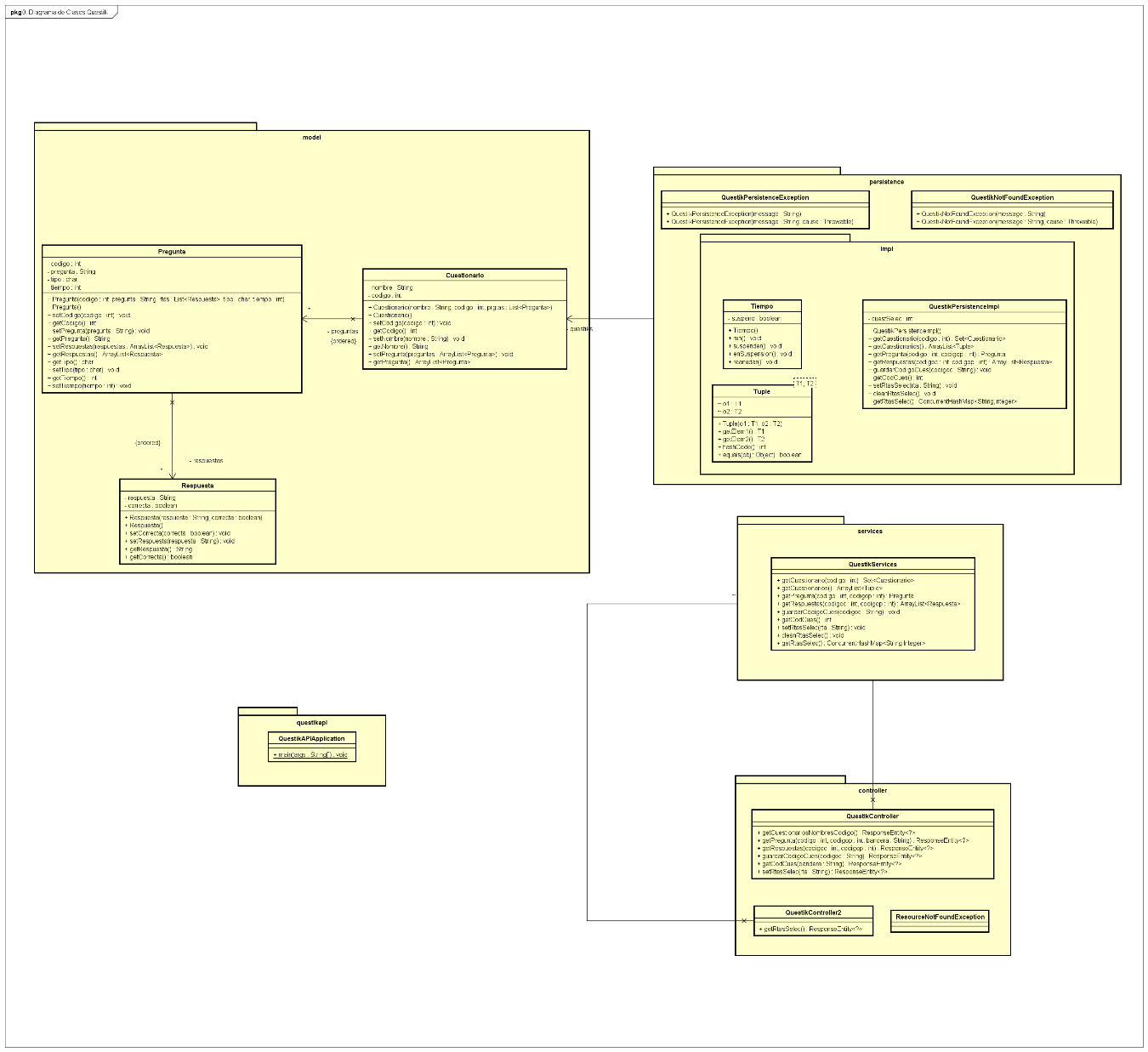
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Origen** | **Estimulo** | **Ambiente** | **Respuesta** | **Medida** |
| Usuarios | Intento de acceso no autorizado a una interfaz | Falta de permisos a un usuario | Questik no permite el acceso | Se redirige a la página principal |

1. **Mantenibilidad (Inspección continua):** Capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
   1. **Escenario:** Esto se logra gracias a sonarcloud, la cual es una plataforma digital que nos ofrece diferentes pruebas de penetración que inspeccionan una aplicación web o un software, esto nos da un análisis específico del código, brindando soluciones a diversos problemas que se encuentren en el código.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Origen** | **Estimulo** | **Ambiente** | **Respuesta** | **Medida** |
| Desarrollo | Modificación en el código | Entorno de desarrollo | Análisis de SonarCloud en el código | Calificaciones obtenidas en SonarCloud |

**Diagramas**

1. **Diagrama de Clases**



1. **Diagrama de Componentes**

Gráfico, Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente con confianza media

1. **Diagrama Entidad-Relación**

Diagrama

Descripción generada automáticamente

1. **Diagrama de Despliegue**

Diagrama

Descripción generada automáticamente

1. **Diagramas de Casos de Uso**
   1. **Usuario**

**Diagrama

Descripción generada automáticamente**

* 1. **Administrador**

**Diagrama

Descripción generada automáticamente**

1. **Diagrama de Secuencia**

**Diagrama

Descripción generada automáticamente**

**Drivers Arquitectónicos**

1. **Requerimientos funcionales**
   1. El administrador podrá ver sus cuestionarios realizados e iniciar un juego con los mismos
   2. Los usuarios pueden ingresar a una partida y acceder a la sala de espera
   3. El administrador podrá ingresar a la plataforma con su correo y contraseña
   4. Los usuarios podrán utilizar el poder de estadística
   5. Los usuarios podrán responder la pregunta carrera
   6. Los usuarios pueden ver la tabla de puntajes después de cada partida
2. **Requerimientos no funcionales**
   1. El sistema debe ser capaz de operar adecuadamente con hasta un total de 100 usuarios
   2. El sistema debe poder controlar una sobrecarga de peticiones al sistema
   3. Los permisos de acceso al sistema solo podrán ser cambiados por el desarrollador.
   4. La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.
   5. El sistema debe poseer interfaces gráficas bien formadas.
   6. El sistema debe tener una disponibilidad del 98% de las veces en que un usuario intente accederlo.
3. **Supuestos Arquitectónicos**
   1. Se asume que los usuarios cuentan con equipos de computo con sistemas operativos Windows o Mac, y que estos tienen acceso a internet.
   2. El internet del usuario debe ser estable para la correcta realización del juego
   3. El administrador debe saber sus credenciales y tener el conocimiento de cómo funciona el juego.
   4. Los usuarios deben conocer los diferentes tipos de preguntas y sus condiciones