10 de enero de 2019

De La Cruz Díaz Adolfo ángel

Murillo Hernández Josafat

Facilitador: Perez Arriaga Juan Carlos

Inbo

Tecnologías para la construcción de software

**Contenido**

[Introducción 2](#_Toc532999586)

[Especificación de los requerimientos. 3](#_Toc532999587)

[Diagrama de casos de uso. 4](#_Toc532999588)

[Descripción de los casos de uso 5](#_Toc532999589)

[Modelo de dominio 24](#_Toc532999590)

[Justificación de framework. 25](#_Toc532999591)

[Seguridad. 27](#_Toc532999592)

[Análisis estático. 28](#_Toc532999593)

[Conclusiones. 29](#_Toc532999594)

Introducción

En el presente documento se presenta el proyecto de preguntas y respuestas llamado “Inbo”, el cual tiene como propósito el poder registrar un quiz completo que conlleva sus respectivas preguntas y respuestas a cada pregunta, manteniendo una persistencia de los datos y con una facilidad de uso basado en la interfaz y con la posibilidad de jugar el quiz de manera local mediante conexión a la misma red.

Se divide en varias categorías, lo que son los requerimientos, que vienen con el diagrama de casos de uso y descripción de casos de uso, se describe el modelo del dominio utilizado, se justifica la selección de frameworks utilizados, se mencionan los apartados de seguridad que utiliza el sistema, se proporciona un enlace directo a la documentación del sistema, se obtienen los resultados del análisis estático utilizando la herramienta sonar cube donde se explican las vulnerabilidades, las malas prácticas y los posibles bugs y finalmente se muestran las conclusiones de cada integrante con lo que cada uno ha aprendido y los retos que se ha encontrado durante el desarrollo.

Especificación de los requerimientos.

Para poder obtener unos requerimientos del sistema lo suficientemente realistas nos enfocamos en una aplicación similar al trabajo solicitado, esta aplicación se llama “Kahoot!”, la cual presenta funcionamientos parecido. Tomas en cuenta las necesidades que se nos expusieron durante la fase de elección del proyecto, donde se nos mencionó que es lo que se necesita para que se un quiz de verdad, que son el poder jugarlo con otros equipos conectados de manera local, el tener la certeza que la información almacenada no se pierda, mantener una persistencia en los datos, y el hecho de que la aplicación no sea difícil de utilizar.

El presente proyecto “Inbo” muestra una forma cómoda de registrar un quiz, poder jugar con otros jugadores mediante una conexión a red, tiene las funcionalidades de enviar un correo electrónico a cualquier persona que se registre en el sistema, proporcionándole una contraseña generada de manera aleatoria por el sistema, esta contraseña podrá modificarse más tarde si así lo desea el usuario, posee el factor de internacionalización en los idiomas inglés y español, tiene una interfaz gráfica muy agradable, lo que promueve que el usuario no se agobie con los colores o con las transiciones entre pantallas.

En cuanto a los requerimientos encontrados, estos se mostrarán en más detalle en las siguientes secciones, con el diagrama de casos de uso y sus respectivas descripciones, de igual manera mostrando el modelo de dominio que se utilizó para el desarrollo del sistema.

Diagrama de casos de uso.

En la figura 1. se presenta la vista general del sistema con los posibles actores que interactúan con el mismo, como se puede apreciar en la imagen se contemplaron tres actores, de los cuales dos heredan la funcionalidad de uno que es considerado genérico, ya que cualquiera que herede de este puede manejar todas las funcionalidades que se le han asignado. Durante la asignación de funcionalidades para cada actor se contemplaron algunos extends e includes, haciendo referencia a que se puede realizar ya sea de manera opcional o de manera obligada.

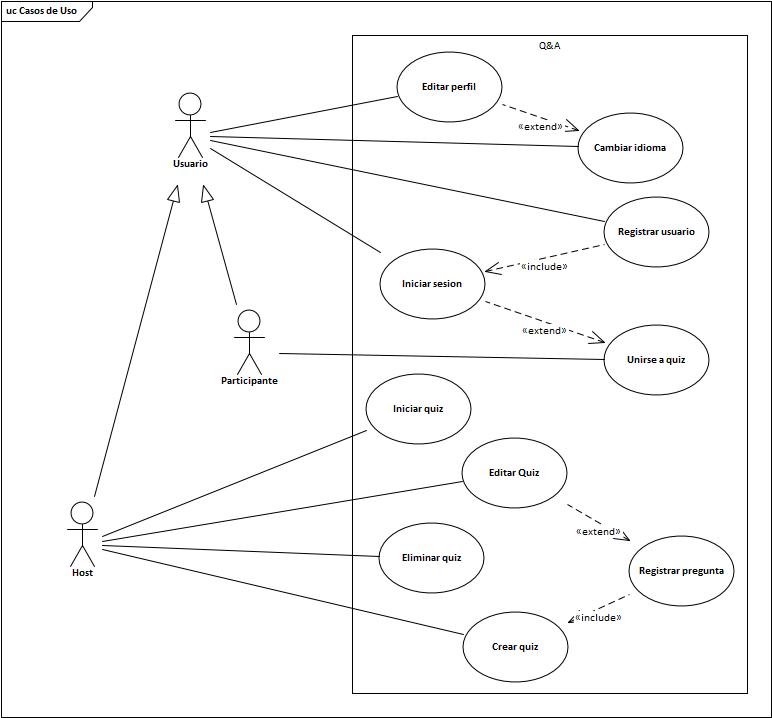


Figura . Diagrama de casos de uso

Descripción de los casos de uso

En esta sección se muestran a detalle los casos deuso que se eligieron, dando el camino positivo más viable como el flujo normal, algunos inconvenientes que no afectan al sistema como los flujos alternos y los fallos dentro del sistema como las excepciones.

|  |  |
| --- | --- |
| **ID:** | QA-01 |
| **Nombre:** | Crear Quiz |
| **Autor(es):** | Josafat Murillo Hernández |
| **Fecha de creación:** | 8 de septiembre de 2018 |
| **Fecha de actualización:** | 19 de diciembre de 2018 |
| **Descripción:** | El host podrá mantener una persistencia en los datos de cualquier quiz que haya creado, dando una facilidad en su creación y pudiendo posponer la adicion de preguntas al quiz. |
| **Actor(es):** | Host |
| **Frecuencia de uso:** | Mínimo una vez al día. |
| **Disparador:** | El actor selecciona la opción “Crear quiz”. |
| **Precondiciones:** | PRE-01: Debe existir una sesión de usuario iniciada. |
| **Flujo normal:** | 1. El sistema despliega una pantalla con información necesaria para crear un quiz.  * Título de Quiz * Descripción * (Agregar imagen)  1. El host ingresa los datos solicitados y selecciona la opción continuar. (ver FA-01, EX-01) 2. El sistema pasa al CU QA-02 3. Termina el caso de uso |
| **Flujos alternos:** | FA-01: Se cancela la creación del quiz.   1. El host selecciona la opción “Regresar”. 2. El sistema despliega un mensaje de confirmación. 3. El host confirma el mensaje. 4. El sistema despliega la pantalla principal, eliminando todo avance del quiz. 5. Termina el caso de uso. |
| **Excepciones:** | EX-01: Se pierde la conexión con la base de datos.   1. El sistema despliega un mensaje indicando que no se puede proseguir con la operación. 2. El host confirma el mensaje. 3. El sistema regresa a la pantalla principal. 4. Termina el caso de uso.   EX-02: Se pierde la conexión con el servidor.   1. El sistema despliega un mensaje indicando que hubo una perdida de conexión. 2. El host confirma el mensaje. 3. El sistema regresa a la pantalla principal. 4. Termina el caso de uso |
| **Postcondiciones:** | POST-01: Se cambia la pantalla a la de “Preguntas”  POST-02: Se agrega un nuevo quiz a la pantalla principal. |
| **Reglas de negocio:** | Ninguna |
| **Incluye (relación Include):** | QA-02 |
| **Extiende (relación extends):** | Ninguno |
| **Prioridad:** | Alta |

|  |  |
| --- | --- |
| **ID:** | QA-02 |
| **Nombre:** | Registrar pregunta |
| **Autor(es):** | Josafat Murillo Hernández |
| **Fecha de creación:** | 8 de septiembre de 2018 |
| **Fecha de actualización:** | 19 de diciembre de 2018 |
| **Descripción:** | El host podrá mantener los datos de cada pregunta mediante la persistencia en la base de datos, para futuras consultas, ediciones o incluso eliminación de estas. |
| **Actor(es):** | Host |
| **Frecuencia de uso:** | Cada vez que se cree un quiz |
| **Disparador:** | PRE-01: El actor selecciona la opción “Registrar pregunta”. |
| **Precondiciones:** | Debe existir un quiz creado o iniciado |
| **Flujo normal:** | 1. El sistema despliega una pantalla con información necesaria para poder agregar una pregunta al quiz.  * Pregunta * Respuesta 1 * Respuesta 2 * Respuesta 3 * Respuesta 4 * Tiempo por pregunta  1. El actor ingresa la información solicitada y selecciona la opción “Guardar quiz” (ver FA-01, FA-03, EX-01) 2. El sistema valida que todos los campos estén llenados y guarda el quiz. 3. Termina el caso de uso |
| **Flujos alternos:** | FA-01: Se cancela la creación del quiz.   1. El actor selecciona la opción “Cancelar”. 2. El sistema despliega un mensaje de confirmación. 3. El actor confirma el mensaje. (ver FA-02) 4. El sistema despliega la pantalla principal, eliminando todo avance del quiz. 5. Termina el caso de uso.   FA-02: Se deniega la cancelación del quiz.   1. El actor deniega el mensaje. 2. El sistema regresa al flujo normal 2.   FA-03: Se agrega otra pregunta.   1. El actor selecciona la opción “Registrar otra pregunta”. 2. El sistema regresa al flujo normal 1. |
| **Excepciones:** | EX-01: Se pierde la conexión con la base de datos.   1. El sistema despliega un mensaje indicando que no se puede proseguir con la operación. 2. El actor confirma el mensaje. 3. El sistema regresa a la pantalla principal. 4. Termina el caso de uso. |
| **Postcondiciones:** | POST-01: El quiz creado es visible en la pantalla “Mis Quizes”. |
| **Reglas de negocio:** | Ninguna |
| **Incluye (relación Include):** | Ninguno |
| **Extiende (relación extends):** | Ninguno |
| **Prioridad:** | Alta |

|  |  |
| --- | --- |
| **ID:** | QA-03 |
| **Nombre:** | Eliminar Quiz |
| **Autor(es):** | Josafat Murillo Hernández |
| **Fecha de creación:** | 8 de septiembre de 2018 |
| **Fecha de actualización:** | 19 de diciembre de 2018 |
| **Descripción:** | El actor podrá eliminar cualquier quiz previamente creado de manera rápida, lo que conlleva liberar espacio y mantener un orden en la ventana de los quizzes. |
| **Actor(es):** | Host |
| **Frecuencia de uso:** | Mínimo una vez al día. |
| **Disparador:** | El actor selecciona la opción “Mis Quizes”. |
| **Precondiciones:** | PRE-01: Debe existir por lo un quiz para eliminar en la pantalla “Mis Quizes” |
| **Flujo normal:** | 1. El sistema despliega todos los quizes creados 2. El actor selecciona la opción “Eliminar Quiz” de un quiz en específico. 3. El sistema despliega un mensaje de confirmación 4. El actor confirma el mensaje. 5. El sistema elimina el quiz 6. Termina el caso de uso. |
| **Flujos alternos:** | FA-01: Se deniega la cancelación de la eliminación.   1. El actor deniega el mensaje. 2. El sistema regresa al flujo normal 2. |
| **Excepciones:** | EX-01: Se pierde la conexión con la base de datos.   1. El sistema despliega un mensaje indicando que no se puede proseguir con la operación. 2. El actor confirma el mensaje. 3. El sistema regresa a la pantalla principal. 4. Termina el caso de uso. |
| **Postcondiciones:** | POST-01: El quiz eliminado desaparece de la pantalla “Mis Quizes”. |
| **Reglas de negocio:** | Ninguna |
| **Incluye (relación Include):** | Ninguno |
| **Extiende (relación extends):** | Ninguno |
| **Prioridad:** | Baja |

|  |  |
| --- | --- |
| **ID:** | QA-04 |
| **Nombre:** | Iniciar Quiz |
| **Autor(es):** | Josafat Murillo Hernández |
| **Fecha de creación:** | 8 de septiembre de 2018 |
| **Fecha de actualización:** | 19 de diciembre de 2018 |
| **Descripción:** | El actor podrá iniciar un quiz previamente creado por él mismo, generando un código de acceso que puede ser enviado por correo electrónico a los participantes, lo tardado es esperar a los invitados pero el inicio es rápido con un tiempo de 10 a 30 segundos de espera para iniciar la partida. |
| **Actor(es):** | Host |
| **Frecuencia de uso:** | Mínimo una vez al día. |
| **Disparador:** | PRE-01: El actor selecciona la opción “Mis Quizes”. |
| **Precondiciones:** | Debe existir quiz en la pantalla “Mis Quizes”. |
| **Flujo normal:** | 1. El sistema despliega todos los quizes creados. 2. El actor selecciona el quiz a jugar. 3. El sistema despliega una pantalla donde se puede ingresar el correo electrónico a las personas que serán las participantes. 4. El actor ingresa los correos electrónicos de los participantes y selecciona la opción “Aceptar”. (ver FA-01, EX-01) 5. El sistema envía un correo electrónico con un código generado para el quiz. (ver FA-03) 6. El actor al considerar que hay suficientes participantes (mínimo 1), selecciona la opción “Iniciar Quiz”. (ver FA-04) 7. El sistema prosigue con el quiz previamente configurado en su creación, al finalizar el juego se abre una sala de chat y se muestra una gráfica con los puntajes, tanto de los puntos obtenidos por pregunta como un conteo global. (ver FA-05) 8. Termina el caso de uso. |
| **Flujos alternos:** | FA-01: Se cancela el quiz.   1. El actor selecciona la opción “Cancelar”. 2. El sistema despliega un mensaje de confirmación. 3. El actor confirma el mensaje. 4. El sistema despliega la pantalla principal tanto para el host como para los invitados que hayan ingresado, perdido validez del código para los invitados que aún no habían ingresado. 5. Termina el caso de uso.   FA-02: Se deniega la cancelación del quiz.   1. El actor deniega el mensaje. 2. El sistema regresa al flujo normal 4.   FA-03: Correo electrónico invalido.   1. El sistema despliega un mensaje indicando que el correo electrónico ingresado esta erróneo 2. El actor confirma el mensaje. 3. El sistema regresa al flujo normal 4.   FA-04: Terminar sesión de quiz   1. El actor selecciona la opción “Finalizar quiz” 2. El sistema redirige tanto al host como a los invitados a la página principal 3. Termina el caso de uso   FA-05: Volver a jugar   1. El actor selecciona la opción “Volver a jugar” 2. El sistema regresa al flujo normal 5. |
| **Excepciones:** | EX-01: Se pierde la conexión con la base de datos.   1. El sistema despliega un mensaje indicando que no se puede proseguir con la operación. 2. El actor confirma el mensaje. 3. El sistema regresa a la pantalla principal. 4. Termina el caso de uso. |
| **Postcondiciones:** | POST-01: El quiz previamente jugado muestra la puntuación de los jugadores |
| **Reglas de negocio:** | Ninguna |
| **Incluye (relación Include):** | QA-07 |
| **Extiende (relación extends):** | Ninguno |
| **Prioridad:** | Alta |

|  |  |
| --- | --- |
| **­­ID:** | QA-05 |
| **Nombre:** | Registrar usuario |
| **Autor(es):** | Adolfo Ángel de la Cruz Díaz |
| **Fecha de creación:** | 08 de septiembre de 2018 |
| **Fecha de actualización:** |  |
| **Descripción:** | Permite al usuario registrarse en el sistema para poder tener todas las funciones disponibles para usuarios registrados, tardando en total del proceso un tiempo de 5 min. |
| **Actor(es):** | Usuario |
| **Frecuencia de uso:** | Al menos 1 vez al mes |
| **Disparador:** | El actor selecciona “Registrarse” en la página de Login |
| **Precondiciones:** | --- |
| **Flujo Normal:** | 1. El sistema despliega en pantalla una ventana con dos campos de texto en los que el usuario deberá introducir su correo electrónico y nombre de usuario. 2. El Usuario introduce su correo electrónico y nombre de usuario, seleccionando “Aceptar” una vez hecho. 3. El sistema registra al usuario en la base de datos, enviando de manera consecuente a la dirección recientemente introducida un correo electrónico de confirmación junto con la contraseña generada por el sistema para que el usuario pueda iniciar sesión. 4. El sistema vuelve a la página de Login.   Fin del caso de uso. |
| **Flujos Alternos:** | --- |
| **Excepciones:** | EX1: Error en la conexión con la base de datos.   1. El sistema muestra un mensaje informando al usuario que el sistema no está disponible de manera temporal para realizar esa acción. 2. El usuario selecciona “Aceptar”. 3. Vuelve al paso 4 del flujo normal.   EX2: Error en la conexión con el servicio de correo.   1. El sistema muestra un mensaje al usuario informándole de una falla al tratar de enviar el correo de confirmación, pudiendo envíarlo nuevamente presionando el botón de “Reenviar correo de confirmación”. 2. El usuario selecciona aceptar. |
| **Postcondiciones:** | POST-1: Los datos del usuario están registrados en la base de datos.  POST-2: Un correo de confirmación fue enviado a la dirección de correo del usuario. |
| **Reglas de negocio:** | --- |
| **Incluye:**  **(relación Include)** | --- |
| **Extiende:**  **(relación Extend)** | --- |
| **Prioridad:** | Alta |

|  |  |
| --- | --- |
| **­­ID:** | QA-06 |
| **Nombre:** | Iniciar Sesión. |
| **Autor(es):** | Usuario |
| **Fecha de creación:** | 8 de septiembre de 2018 |
| **Fecha de actualización:** | --- |
| **Descripción:** | Permite al usuario ingresar al sistema para hacer uso de sus funciones disponibles, un tiempo mínimo de 10 a 30 segundos. |
| **Actor(es):** | Usuario |
| **Frecuencia de uso:** | Al menos 1 vez al día |
| **Disparador:** | El sistema se ejecuta |
| **Precondiciones:** | --- |
| **Flujo Normal:** | 1. El sistema despliega la pantalla de Login con dos campos pidiendo los siguientes datos del usuario.    * 1. Nombre de Usuario.      2. Contraseña. 2. El usuario introduce los datos requeridos, seleccionando Aceptar para ingresar (Ver FA1, FA2). 3. El sistema verifica que el usuario ingresado esté registrado en el sistema con la contraseña ingresada. (Ver FA3, EX1) 4. El sistema despliega el Dashboard (Página Principal). 5. Fin del caso de uso. |
| **Flujos Alternos:** | FA1: El usuario no está registrado.   1. El usuario selecciona la opción Registrarse. 2. Inicia caso de uso Registrar usuario (QA-05). 3. Vuelve al paso 1 del flujo normal.   FA2: El usuario es un invitado.   1. El usuario selecciona la opción Ingresar como invitado. 2. Inicia caso de uso Unirse a quiz (QA-07). 3. Fin del caso de uso.   FA3: El nombre de usuario o la contraseña son incorrectos.   1. El sistema muestra un mensaje informando a usuario de que el nombre de usuario o la contraseña introducida son incorrectos. 2. El usuario selecciona aceptar. 3. Vuelve al paso 2 del flujo normal. |
| **Excepciones:** | EX1: Error en la conexión con la base de datos.   1. El sistema muestra un mensaje informando al usuario que el sistema no está disponible de manera temporal para realizar esa acción. 2. El usuario selecciona “Aceptar”. 3. Vuelve al paso 2 del flujo normal. |
| **Postcondiciones:** | --- |
| **Reglas de negocio:** | Ninguna |
| **Incluye:**  **(relación Include)** | QA-07 |
| **Extiende:**  **(relación Extend)** | QA-05 |
| **Prioridad:** | Alta |

|  |  |
| --- | --- |
| **­­ID:** | QA-07 |
| **Nombre:** | Unirse a quiz |
| **Autor(es):** | Adolfo Ángel de la Cruz Díaz |
| **Fecha de creación:** | 9 de septiembre de 2018 |
| **Fecha de actualización:** | --- |
| **Descripción:** | Permite al usuario unirse al juego de un quiz al que se le ha invitado y está activo, el ingreso a la partida es rápido esperando un tiempo máximo de 1 min. |
| **Actor(es):** | Participante |
| **Frecuencia de uso:** | Al menos 1 vez al mes |
| **Disparador:** | Para usuarios registrados, el participante selecciona la opción “Jugar con código quiz” de la página principal. Para los invitados, el participante selecciona “Entrar como invitado” en la página de Login. |
| **Precondiciones:** | --- |
| **Flujo Normal:** | 1. El sistema despliega en pantalla la página para unirse a un juego, la cual consiste en un solo campo en el que se debe introducir el código quiz que le fue enviado al usuario en la invitación al mismo. 2. El usuario introduce el código, seleccionando “Aceptar” para iniciar el juego. 3. El sistema verifica que el código pertenezca a un quiz activo en espera de participantes (Ver FA1, EX1). 4. El sistema ingresa al usuario a la lista de participantes del quiz.   Fin del caso de uso. |
| **Flujos Alternos:** | FA1: Código inválido.   1. El sistema muestra en pantalla un mensaje informando al usuario que el código introducido es incorrecto o no pertenece a una partida activa de un quiz. 2. El usuario selecciona Aceptar. 3. Vuelve al paso 2 del flujo normal. |
| **Excepciones:** | EX1: Error en la conexión con la base de datos.   1. El sistema muestra un mensaje informando al usuario que el sistema no está disponible de manera temporal para realizar esa acción. 2. El usuario selecciona “Aceptar”. 3. Vuelve al paso 2 del flujo normal. |
| **Postcondiciones:** | POST-1: El usuario está registrado como participante del juego. |
| **Reglas de negocio:** | Ninguna |
| **Incluye:**  **(relación Include)** | QA-04 |
| **Extiende:**  **(relación Extend)** | Ninguna |
| **Prioridad:** | Alta |

|  |  |
| --- | --- |
| **­­ID:** | QA-08 |
| **Nombre:** | Editar Perfil |
| **Autor(es):** | Josafat Murillo Hernández |
| **Fecha de creación:** | 13 de septiembre de 2018 |
| **Fecha de actualización:** | --- |
| **Descripción:** | Permite al usuario modificar datos de su perfil, como su foto, nombre, y correo, de manera sencilla con un mínimo de 2 clicks para la operación. |
| **Actor(es):** | Usuario |
| **Frecuencia de uso:** | Al menos 1 cada 6 meses |
| **Disparador:** | El usuario registrado selecciona la opción configuración en la pantalla principal. |
| **Precondiciones:** | PRE-01: Haber iniciado el sistema e ingresar con una cuenta registrada. |
| **Flujo Normal:** | 1. El sistema despliega la pantalla configuración con campos y opciones varias a modificar, como lo son:  * Cambiar imagen * Nombre de usuario * Cambiar contraseña * Contraseña actual * Contraseña nueva * Repetir contraseña nueva * Correo electrónico  1. El usuario realiza los cambios que quiera efectuar y selecciona la opción guardar. (ver FA-01, FA-02) 2. El sistema valida la información, efectúa los cambios solicitados y se redirecciona a la pantalla principal. 3. Termina el caso de uso |
| **Flujos Alternos:** | FA-01: Nombre de usuario ya registrado.   1. El sistema despliega un mensaje indicando que el nombre de usuario que eligió ya esta ocupado. 2. El usuario confirma el mensaje 3. El sistema regresa al flujo normal 2.   FA-02: Contraseña nuevo no coincide con la repetición de esta.   1. El sistema despliega un mensaje indicando que la contraseña ingresada en el campo “Repetir contraseña nueva” no coincide con el campo “Contraseña nueva”. 2. El usuario confirma le mensaje. 3. El sistema regresa al flujo normal 2. |
| **Excepciones:** | EX1: Error en la conexión con la base de datos.   1. El sistema muestra un mensaje informando al usuario que el sistema no está disponible de manera temporal para realizar esa acción. 2. El usuario selecciona “Aceptar”. 3. Vuelve al paso 2 del flujo normal. |
| **Postcondiciones:** | POST-1: El usuario ve reflejado los cambios en la información desde la página principal |
| **Reglas de negocio:** | Ninguna |
| **Incluye:**  **(relación Include)** | Ninguno |
| **Extiende:**  **(relación Extend)** | Ninguna |
| **Prioridad:** | Media |

|  |  |
| --- | --- |
| **­­ID:** | QA-09 |
| **Nombre:** | Cambiar idioma |
| **Autor(es):** | Josafat Murillo Hernández |
| **Fecha de creación:** | 13 de septiembre de 2018 |
| **Fecha de actualización:** | --- |
| **Descripción:** | Permite al usuario modificar el idioma por default que la aplicación en cualquier momento que este quiera, solo con la interacción de 2 clicks y un tiempo variable de carga del nuevo idioma pero sin sobrepasar el máximo de 1 min.. |
| **Actor(es):** | Usuario |
| **Frecuencia de uso:** | Al menos 1 vez al mes |
| **Disparador:** | Para todo tipo de usuario, selecciona la opción configuración desde la pantalla principal. |
| **Precondiciones:** | PRE-01: Haber iniciado el sistema e ingresar con una cuenta registrada. |
| **Flujo Normal:** | 1. El sistema despliega la pantalla configuración con las opciones de idiomas en la parte derecha de la pantalla. 2. El usuario selecciona el idioma que quiera dentro de los ofrecidos y selecciona la opción guarda. 3. EL sistema realiza los cambios necesarios de lenguaje y se redirecciona a la pantalla principal 4. Termina el caso de uso. |
| **Flujos Alternos:** | --- |
| **Excepciones:** | EX1: Error en la conexión con la base de datos.   1. El sistema muestra un mensaje informando al usuario que el sistema no está disponible de manera temporal para realizar esa acción. 2. El usuario selecciona “Aceptar”. 3. Vuelve al paso 2 del flujo normal. |
| **Postcondiciones:** | POST-1: La aplicación cambia todos los textos al idioma seleccionado. |
| **Reglas de negocio:** | Ninguna |
| **Incluye:**  **(relación Include)** | Ninguno |
| **Extiende:**  **(relación Extend)** | Ninguna |
| **Prioridad:** | Alta |

Modelo de dominio

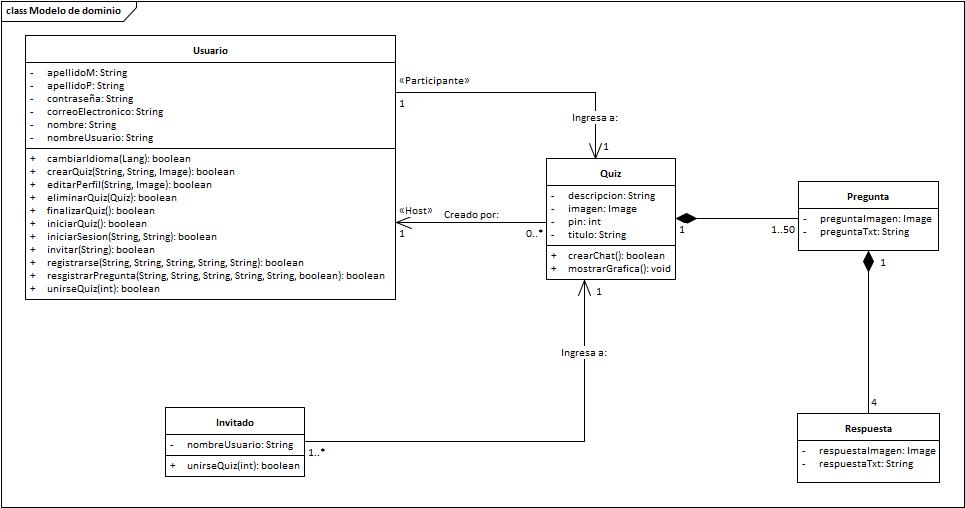
La figura 2. muestra el dominio del sistema, las clases que se manejan en el sistema y las entidades que se controlan con el framework utilizado, también delimitando la cantidad de preguntas y respuestas por quiz.

Figura 2. Modelo de dominio

Justificación de framework.

En esta sección se presenta una justificación del porque se utilizó cierto framework para este sistema, dando una breve descripción del framework, las ventajas que este ofrece y en que se aplicó dentro del sistema.

**Persistencia JPA.**

Java Persistence API por sus siglas en inglés JPA. Es un framework desarrollado para la plataforma de java que maneja datos relacionales.

Tiene como principal objetivo el no perder las ventajas del manejo de objeto durante la interacción con la base de datos.

Maneja diferentes tipos de clases para conseguir una conexión con la base de datos y poder almacenar cualquier tipo de información.

Maneja las entidades o entitites, que son clases que representan una tabla en la base de datos, cada instancia de estas entidades corresponde a un registro en la base de datos, por lo genera algunas entidades se relacionan con otras mediante los metadatos.

*Ventajas:*

-Permite un desarrollo más rápido.

-Permite trabajar con la base de datos mediante entidades y no con querys.

-Ofrece un paradigma 100% orientado a objetos.

-Mejora el mantenimiento del sistema.

*Desventajas:*

-No ofrece todas las funcionalidades que podría ofrecer una consulta nativa.

-Puede representar una curva de aprendizaje más grande.

*Aplicación dentro del sistema.*

En nuestro sistema se aplicó solo para la conexión y registro en la base datos, para ello se colocó la base de datos dentro del apartado de servicios y se creó un paquete META-INF que contiene la persistencia de la base de datos, e crearon las entidades de manera automática y se utilizaron controladores, en un principio se pensó en utilizar las named querys pero por falta de conocimiento no se consiguió establecer una buena consulta, es por eso que se optó por utilizar los controladores, se crearon funciones a partir de estos controladores para diferenciar cada consulta por separado, de los CRUD de cada entidad, así se mantiene un orden en cuanto a funcionalidades se refiere y facilita la modificación de algunos elementos.

**Conexión en red RMI.**

Durante el curso se expusieron diferentes modos de conexión entre los vistos están, los sockets, el node y el RMI, a continuación, se expone por qué se utilizó este método de conexión por sobre los demás, cabe aclarar que RMI se utilizó en todo el sistema, no hubo combinación de métodos de conexión.

*Socket:*

Son un mecanismo que nos permite establecer un enlace entre dos programas que se ejecutan de manera independiente, hace uso de dos clases la llamada *Socket* que es del lado del cliente y *ServerSocket* que es del lado del servidor.

*Node:*

No es un framework, no es una herramienta y tampoco es un lenguaje, es un entorno de desarrollo que maneja su propia máquina virtual para estar disponible en diferentes plataformas.

*RMI (Remote Method Invocation):*

Es un mecanismo ofrecido por Java para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de Java y proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java.

Con los conceptos antes mencionados, que no abarcan a mucho detalle lo que es cada cosa, se puede obtener la razón del porque utilizamos RMI, es porque al estar basada y diseñada exclusivamente para aplicaciones en Java ya proporciona facilidades para su implementación en cualquier programa de Java, por lo que fue más fácil implementarla en el sistema sin mayores complicaciones.

Seguridad.

El sistema “Inbo” lo consideramos seguro debido a los siguientes factores:

-Aunque sea obligatorio esto, el manejo de excepciones evita que se obstruya el funcionamiento del sistema, por lo que mantiene su integridad durante la ejecución y ayuda al usuario a comprender que está mal y como podría solucionarlo.

-Maneja las entradas del sistema, mediante ciertos validadores se obstruye el ingreso a datos no válidos, se verifica si los campos están vacíos o son iguales a null.

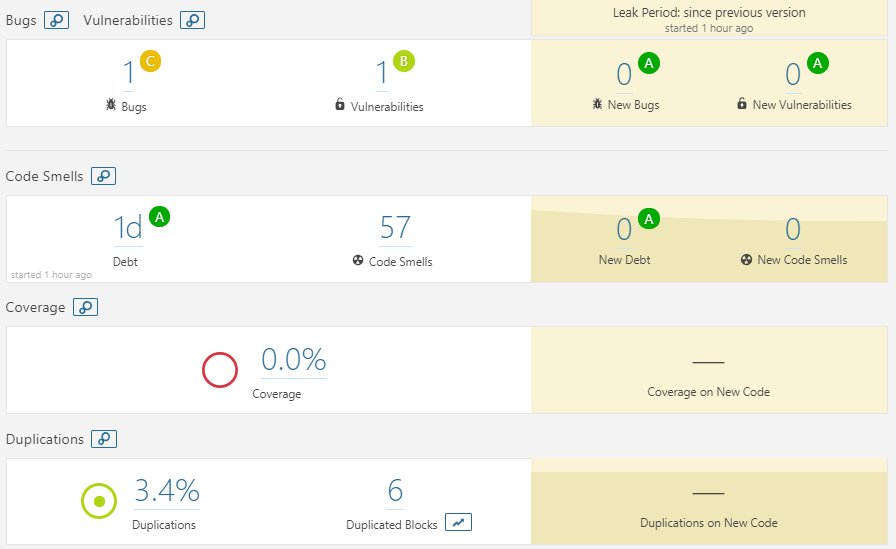
-Se limita el acceso de los datos, para evitar complicaciones se utilizó un sistema de restricciones que evitan que el sistema pueda sobrecargarse, en el caso de los campos de texto, se limita el número de caracteres disponibles, y en el caso de registrar algo en la base de datos se limita a que debe estar la información solicitada antes de realizar la operación.

Análisis estático.

Para la realización de este análisis estático se utilizó la herramienta SonarQube, la cual genera un breve informe de los bugs, vulnerabilidades, o code smells que pueden encontrarse en el sistema. Cabe aclarar que los cade smells indicados sobre el apartado de persistencia son originados por el mismo framework JPA, con lo que podemos decir con toda certeza que la herramienta detectó esto como falsos positivos. Ahora analizando lo que queda del sistema tenemos como resultado lo siguiente:

Se han obtenido diversos bugs que han sido solucionados todos, cada bug representaba un trabajo de solución significativo, pero en su mayoría fueron solucionados de manera rápida y sin afectar otras funciones, en cuanto a vulnerabilidades se han resuelto todas las que indicaba la herramienta, por el uso de del printstack en una excepción y por el uso de la palabra password por lo que podemos decir que son falsos negativos ya que en nuestro caso no los consideramos graves, por lo que se espera que no se encuentre alguna durante la ejecución del programa, hablando de los code smells, se han mencionado varios, muchos de ellos por el framework de persistencia pero aun así se han encontrado otros cuantos code smells pertenecientes a la lógica del programa, se han analizado estos code smells y se han encontrado que son falsos positivos por lo que no se consideraron mucho para su solución, durante la evaluación del análisis no se pudieron determinar falsos negativos por lo que no se posee la certeza absoluta de que el sistema carezca de estos.

A continuación, se presenta el resultado del proyecto InboRMI que solo contiene al servidor, como se puede observar en la figura se obtuvieron 57 code smells, 1 vulnerabilidad y 1 bug, para esto, no consideramos que son graves y que no afectan al sistema, por lo que consideramos que son falsos positivos, ya que ese bug que detecta sonarqube no es considerado un bug como tal y la vulnerabilidad tampoco, en cuanto a los code smells que nos han remarcado se puede sacar como conclusión que son igualmente falsos positivos ya que son provenientes del JPA, tanto de las entidades como de los controladores, así mismo los bloques duplicados son de los mismos controladores y solo uno es nuestro de parte de las imágenes thumbnail.



Conclusiones.

*Adolfo:*

Mi conclusión personal es que JPA es un framework que facilita la persistencia de los datos para los programas en Java, sin embargo, puede llegar a ser un tanto complejo cuando las consultas hacen referencia a entidades asociadas. Por otro lado, el modelo entidad-servidor es un modelo altamente popular, y que hoy en día se implementa en casi todos los sistemas actuales. Esto, entonces, nos orilla a aprender más sobre estas tecnologías y renovarnos junto a ellas.

*Josafat:*

En lo personal me pareció más sencillo el JPA que el DAO, ya que prácticamente tienes todo ya hecho solo es cuestión de manera lo que el framework te proporciona a tu favor, sin embargo, no pude comprender del todo las consultas JPQL que maneja JPA, por lo que tardé mucho tiempo en intentar comprender este tipo de consultas que al final no pude asimilarlas, es por eso que me dedique a solo utilizar las funciones del framework y a adaptar las a lo que yo necesitaba, eso del lado de la persistencia, sobre las conexiones, a pesar de que los sockets son una versión más básica de RMI me parecieron una opción viable para establecer una conexión, sin embargo RMI ya tiene ciertas ventajas para con los sockets y es por eso que me resulto más fácil comprenderlo, en cambio node me pareció difícil de aprender ya que no estoy acostumbrado a utilizar scripts, antaño yo programaba html en bloc de notas pero con las nuevas herramientas que te facilitan el desarrollo de html se me fue olvidando el cómo usar las etiquetas, que son muy similares a la de los scripts, es por eso y por falta de tiempo que no conseguí comprender ente método de conexión. Como algo que pude aprender en el desarrollo del sistema, es que no sé nada de fxml, aunque tenga mejor edición que swing, me acostumbre demasiado a utilizar swing que me costó demasiado adaptarme a fxml. Finalmente, como lecciones aprendidas, me quedo con lo básico de las conexiones y con las facilidades de la persistencia, aún me queda mucho por aprender sobre esto pero con la marcha conseguiré hacer un sistema sin la necesidad de ver tutoriales en internet.