**Licenciatura en Ingeniería de Software**

**Tecnologías para la Construcción de Software**

**Mtro. Juan Carlos Pérez Arriaga**

**Proyecto Q&A**

**Puxka Acosta Domínguez**

**Eduardo Rosas Rivera**

**5° Semestre**

**07/01/2019**

# Creacion del Repositorio

Liga: <https://github.com/EduardoRR0798/Juego-P-R>

**Requerimientos**

* Cuando el usuario se acabe de registrar, el sistema deberá enviar un correo electrónico de confirmación al correo del usuario
* El usuario con una cuenta registrada y un usuario invitado podrá iniciar sesión en el sistema.
* El usuario sin una cuenta podrá participar en un juego si se le envía una invitación
* El jugador podrá registrar sus preguntas para un futuro juego.
* El jugador podrá agregar 4 imágenes correspondientes a sus respuestas y una imagen como pregunta.
* Solo cuando el jugador que hizo el juego quiera empezarlo, el sistema deberá permitir a los demás jugadores empezar el juego.
* Cuando el jugador responda una pregunta, el sistema deberá calificar la respuesta inmediatamente
* El jugador podrá contestar cada pregunta en un tiempo correspondiente a 15 segundos.
* El jugador que contestó primero la pregunta podrá obtener más puntos que los demás que también contestaron la pregunta correctamente.
* El jugador podrá monitorear el número de respuestas correctas que tiene en el juego.
* Cualquier jugador podrá participar en el chat del juego.
* Al contestar una pregunta, el sistema deberá mostrar una gráfica con el avance que el jugador lleva al momento en la partida.
* El jugador podrá cambiar el idioma del sistema antes de iniciar sesión.
* La creación de la gráfica de respuestas no tomará más de 1.0 segundo
* Los datos para registrarse sólo se usarán en el sistema.
* Las contraseñas de los jugadores registrados no se almacenarán como tal, sino sus referencias.
* Si un usuario intenta registrarse con datos registrados anteriormente, el sistema deberá impedir su registro.

# Modelo de clases





**Prototipo**

Ver archivo prototipo.vp

**Descripción de casos de uso**

|  |  |
| --- | --- |
| **ID:** | CU01 |
| **Nombre:** | Registrarse |
| **Autor(es):** | Eduardo Rosas Rivera |
| **Actor(es):** | Invitado |
| **Descripción:** | Permite a un invitado registrarse en la base de datos. |
| **Precondiciones:** | 1. El invitado no está registrado en el sistema. 2. El invitado se encuentra en la ventana de *Login*. |
| **Flujo Normal:** | 1. El invitado presiona el botón “Registrarse”. 2. El sistema muestra la pantalla de *Registro* con los datos a solicitar (nombre, correo, contraseña). 3. El Usuario llena los datos solicitados y presiona “Registrar”.   [Flujo alterno A.3: El Usuario presiona “Cancelar”]   1. El sistema valida los datos.   [Flujo alterno A.4: No se llenaron todos los campos]   1. El sistema guarda los datos en la base de datos y muestra el mensaje “Se ha registrado al usuario exitosamente”.   [Excepción E.5: No se pudo registrar en la base de datos]   1. El sistema envía un correo electrónico de solicitud de confirmación. 2. Fin de caso de uso. |
| **Flujos Alternos:** | A.3 El Usuario presiona “Cancelar”  A.3.1 El sistema vuelve a la ventana *Login*.  A.3.2 Fin de caso de uso.  A.4 No se llenaron todos los campos.  A.4.1 El sistema muestra el mensaje “Llene todos los campos.  A.4.2 Regresa al paso 3. |
| **Excepciones:** | E.5 No se pudo registrar en la base de datos.  E.5.1 El sistema muestra un mensaje de error.  E.5.2 Fin de caso de uso. |
| **Postcondiciones:** | El usuario queda registrado en la base de datos. |
| **Entradas:** | Nombre, Correo electrónico, contraseña. |
| **Salidas:** | Correo de solicitud de información. |
| **Incluye:**  **(relación Include)** |  |
| **Extiende:**  **(relación Extend)** |  |

|  |  |
| --- | --- |
| **ID:** | CU02 |
| **Nombre:** | Iniciar sesión |
| **Autor(es):** | Eduardo Rosas Rivera |
| **Actor(es):** | Jugador, Invitado |
| **Descripción:** | Este caso de uso le permite al jugador o invitado ingresar al sistema. |
| **Precondiciones:** | * + - * El jugador o invitado no ha iniciado sesión. * El jugador o invitado está registrado en la base de datos. * El invitado tiene un código de acceso al juego y un usuario. * El jugador o invitado se encuentran en la ventana de *Login* |
| **Flujo Normal:** | * + - 1. El jugador o invitado ingresa los datos nombre y contraseña.   [Flujo alterno A.1 Acceder como invitado]   * + - 1. El jugador o invitado presiona “Ingresar”.       2. El sistema valida los datos.   [Flujo alterno A.2: Datos inválidos]   * + - 1. El sistema muestra el mensaje “Iniciando sesión”.   [Excepción E.4: Error de ingreso.]  5. El sistema muestra la ventana *Menú Principal.*   * + - 1. Fin de caso de uso. |
| **Flujos Alternos:** | A.2 Acceder como Invitado  A.2.1 El invitado ingresa el código de invitado.  A.2.2 El invitado presiona la casilla “Soy invitado”.  A.2.3 Avanza al paso 2.  A.3 Hay datos inválidos.  A.3.1 El sistema muestra el mensaje “Nombre o contraseña inválidos”.  A.3.2 Regresa a paso 1. |
| **Excepciones:** | E.3 Error de ingreso  E.3.1 El sistema muestra un mensaje de error.  E.3.2 Fin de caso de uso. |
| **Postcondiciones:** | El jugador o invitado tiene acceso al sistema. |
| **Extiende:** |  |
| **Incluye:** |  |

|  |  |
| --- | --- |
| **ID:** | CU03 |
| **Nombre:** | Cerrar sesión |
| **Autor(es):** | Eduardo Rosas Rivera |
| **Actor(es):** | Jugador o invitado |
| **Descripción:** | Este caso de uso le permite al jugador o invitado cerrar su sesión. |
| **Precondiciones:** | * El jugador o invitado inicio sesión en el sistema. * El jugador se encuentra en la ventana *Menú Principal.* |
| **Flujo Normal:** | 1. El jugador o invitado presiona el botón “Cerrar sesión”. 2. El sistema cierra la sesión de jugador o invitado. 3. Fin de caso de uso. |
| **Flujos Alternos:** | N/A |
| **Excepciones:** | N/A |
| **Postcondiciones:** | * La sesión del JUGADOR o INVITADO queda cerrada. |
| **Incluye:** |  |
| **Extiende:** |  |

|  |  |
| --- | --- |
| **ID:** | CU04 |
| **Nombre:** | Registrar Pregunta |
| **Autor(es):** | Eduardo Rosas Rivera |
| **Actor(es):** | Jugador |
| **Descripción:** | El jugador registra una pregunta en el sistema. |
| **Precondiciones:** | * El jugador esta registrado en el sistema. * El jugador se encuentra en la ventana *Menú Principal.* |
| **Flujo Normal:** | 1. El jugador presiona el botón “Registrar pregunta”. 2. El sistema muestra la pantalla de registro de pregunta con los datos a solicitar (pregunta, 4 respuestas, categoría, no set, respuesta correcta). 3. El jugador llena los datos.   [Flujo alterno A.3: El jugador ingresa una imagen como pregunta].  [Flujo alterno B.3: El jugador ingresa imágenes como respuestas].   1. El jugador selecciona la opción correspondiente a la respuesta correcta en el comboBox. 2. El jugador presiona el botón “+” en la esquina superior derecha para añadir la PREGUNTA. 3. El jugador selecciona la categoría y el número de set cuando haya terminado de registrar por lo menos dos PREGUNTA. 4. El jugador presiona “Registrar” 5. El sistema valida los datos. 6. El sistema guarda los datos en la base de datos.   [Excepción E.8: Error de registro]   1. Fin de caso de uso. |
| **Flujos Alternos:** | A.3 El jugador ingresa una imagen como pregunta.  A.3.1 El jugador presiona el botón “+” junto al campo de texto de la pregunta.  A.3.2 El sistema abre una ventana de búsqueda de archivos con solo imágenes.  A.3.3 El sistema selecciona la imagen a agregar.  A.3.4 El sistema bloquea el campo de texto de la pregunta.  A.3.5 Paso 3 Flujo normal.  B.3 El jugador ingresa imágenes como respuestas.  B.3.1 Para cada respuesta el usuario presiona el botón “+” ubicado por debajo del campo de texto para cada respuesta,  B.3.2 El sistema abre una ventana de búsqueda de archivos con solo imágenes.  A.3.3 El sistema selecciona la imagen a agregar.  A.3.4 Avanza al paso 5. |
| **Excepciones:** | E.8 Error de registro.  E.8.1 el sistema un mensaje de error.  E.8.2 fin de caso de uso. |
| **Postcondiciones:** | La pregunta queda registrada en la base de datos. |
| **Entradas:** | Pregunta o imagen, 4 respuestas o 4 imágenes, categoría, set, respuesta correcta. |
| **Salidas:** | Registro en la base de datos. |
| **Incluye:**  **(relación Include)** |  |
| **Extiende:**  **(relación Extend)** |  |

|  |  |
| --- | --- |
| **ID:** | CU05 |
| **Nombre:** | Iniciar partida |
| **Autor(es):** | Eduardo Rosas Rivera |
| **Actor(es):** | Jugador |
| **Descripción:** | El JUGADOR inicia una PARTIDA |
| **Precondiciones:** | * El JUGADOR tiene por lo menos un set registrado en la base de datos. * El jugador se encuentra en la ventana *Iniciar partida.* |
| **Flujo Normal:** | 1. El jugador presiona el botón “Iniciar”. 2. El sistema recupera las categorías de los SET PREGUNTA registrados al momento y los enlista. 3. El sistema muestra la pantalla de inicio de partida solicitando los datos para la partida (categoría, nombre) 4. El jugador selecciona una categoría e ingresa un nombre. 5. El jugador presiona “Iniciar”.   [Flujo alterno A.4: el jugador presiona “cancelar”]  5. El sistema crea la ventana *Esperar jugadores.*   1. El sistema espera la solicitud de todos los jugadores para unirse a la partida y muestra en una lista a todos los jugadores que se hayan unido. 2. El jugador presiona “Iniciar”. 3. El sistema crea la ventana *Envió de Preguntas*. 4. El sistema envia todas las PREGUNTA dentro del SET PREGUNTA. 5. El jugador presiona “Iniciar” 6. El sistema da inicio a la partida creando la ventana *Responder pregunta.* 7. Fin de caso de uso. |
| **Flujos Alternos:** | A.4 El JUGADOR presiona “Cancelar”  A.4.1 El sistema cierra la ventana y vuelve a *Menú principal.*  A.4.2 Fin de caso de uso. |
| **Excepciones:** |  |
| **Postcondiciones:** | La partida es iniciada. |
| **Entradas:** | Categoría, nombre de la partida. |
| **Salidas:** | La partida se guarda en la base de datos. |
| **Incluye:**  **(relación Include)** |  |
| **Extiende:**  **(relación Extend)** |  |

|  |  |
| --- | --- |
| **ID:** | CU06 |
| **Nombre:** | Enviar mensaje |
| **Autor(es):** | Puxka Acosta Domínguez |
| **Actor(es):** | Jugador, invitado |
| **Descripción:** | Este caso de uso le permite al jugador o invitado enviar un mensaje a través del chat. |
| **Precondiciones:** | 1. El jugador o invitado ha entrado al juego. 2. El jugador o invitado están conectados a la red. 3. La partida ha iniciado. |
| **Flujo Normal:** | 1. El jugador o invitado escribe un mensaje y presiona el botón “Enviar”.   [Flujo alterno A.1: El jugador presiona el botón “Enviar” sin haber escrito nada.]   1. El sistema muestra el mensaje en el chat.   [Excepción E.3: Se pierde conexión con el servidor]   1. Fin de caso de uso. |
| **Flujos Alternos:** | A.1. El JUGADOR o INVITADO presiona el botón “Enviar” sin haber escrito nada.  A.1.1 El sistema valida el mensaje.  A.1.2 El sistema no emite el mensaje.  A.1.1 Regresa al paso 1 del flujo normal. |
| **Excepciones:** | E.3. Se pierde conexión con el servidor  E.3.1 El sistema no emite el mensaje.  E.3.2 Fin de caso de uso. |
| **Postcondiciones:** | El mensaje fue enviado. |

|  |  |
| --- | --- |
| **ID:** | CU07 |
| **Nombre:** | Responder pregunta |
| **Autor(es):** | Puxka Acosta Domínguez |
| **Actor(es):** | JUGADOR, INVITADO |
| **Descripción:** | Este caso de uso le permite al jugador o invitado responder una pregunta del juego. |
| **Precondiciones:** | 1. El jugador o invitado ha entrado al juego. 2. El jugador o invitado están conectados a la red. 3. La partida ha iniciado. |
| **Flujo Normal:** | 1. El sistema muestra la PREGUNTA y sus RESPUESTA y solicita seleccionar una RESPUESTA. 2. El jugador o invitado selecciona una RESPUESTA.   [Flujo alterno A.2: El jugador o invitado no selecciona una respuesta]   1. El sistema califica las RESPUESTA seleccionadas y muestra en una gráfica del puntaje acumulado de los JUGADOR.   [Excepción E.3: Se pierde conexión con el servidor]   1. El sistema avanza a la siguiente pregunta en caso de haber una. 2. El sistema muestra una pantalla con el nombre del ganador y un botón para volver a *Menú Principal*. 3. El jugador o invitado presiona el botón “Volver” 4. Fin de caso de uso. |
| **Flujos Alternos:** | A.2. El jugador o invitado no selecciona una respuesta.  A.2.1 El sistema califica la respuesta con puntaje -1.  A.2.1 Regresa al paso 1. |
| **Excepciones:** | E.3. Se pierde conexión con el servidor  E.3.1 Fin de caso de uso. |
| **Postcondiciones:** |  |

|  |  |
| --- | --- |
| **ID:** | CU08 |
| **Nombre:** | Enviar invitación |
| **Autor(es):** | Puxka Acosta Dominguez |
| **Actor(es):** | Jugador |
| **Descripción:** | Este caso de uso le permite al jugador enviar una invitación a un usuario sin cuenta usuario. |
| **Precondiciones:** | 1. El jugador ha iniciado sesión. 2. El jugador o invitado están conectados a la red. |
| **Flujo Normal:** | 1. El jugador presiona el botón “Invitar” 2. El sistema crea la ventana “Invitar”. 3. El sistema solicita un correo electrónico. 4. El jugador ingresa un correo electrónico.   [Flujo alterno A.3: el jugador presiona “Cancelar”]   1. El sistema envía una invitación al correo electrónico ingresado.   [Excepción E.5: Se pierde conexión con el servidor]   1. Fin de caso de uso. |
| **Flujos Alternos:** | A.3. El jugador presiona “Cancelar”  A.3.1 Fin de caso de uso |
| **Excepciones:** | E.5. Se pierde conexión con el servidor  E.5.1 El sistema muestra un mensaje de error.  E.5.3 Fin de caso de uso. |
| **Postcondiciones:** | * El jugador envió una invitación * El invitado podrá entrar al juego. |

|  |  |
| --- | --- |
| **ID:** | CU09 |
| **Nombre:** | Unirse a partida |
| **Autor(es):** | Puxka Acosta Domínguez |
| **Actor(es):** | Jugador o invitado |
| **Descripción:** | Este caso de uso le permite al jugador o invitado entrar a una partida hecha por otro jugador |
| **Precondiciones:** | 1. El jugador ha iniciado sesión. 2. La partida no ha iniciado. 3. El jugador o invitado están conectados a la red. |
| **Flujo Normal:** | 1. El jugador o invitado ingresa a la ventana *Unirse.* 2. El sistema solicita seleccionar una CATEGORÍA y PARTIDA. 3. El jugador o invitado selecciona una CATEGORÍA y una PARTIDA y presiona “Unirse”   [Flujo alterno A.3: El jugador presiona “Cancelar”]   1. El sistema muestra crea la ventana *Jugadores conectados.* 2. El sistema crea la ventana *Responder pregunta* cuando el jugador que creo la partida presiona “Iniciar”   [Excepción E.4: Se pierde conexión con el servidor]   1. Fin de caso de uso. |
| **Flujos Alternos:** | A.3. El jugador o invitado presiona “Cancelar”  A.3.1 El sistema vuelve a *Menú Principal.*  A.3.1 Fin de caso de uso |
| **Excepciones:** | E.2. Se pierde conexión con el servidor  E.2.1 Fin de caso de uso. |
| **Postcondiciones:** | El jugador o invitado esperará el inicio de la PARTIDA. |

|  |  |
| --- | --- |
| **ID:** | CU10 |
| **Nombre:** | Cambiar idioma |
| **Autor(es):** | Puxka Acosta Domínguez |
| **Actor(es):** | Jugador, invitado |
| **Descripción:** | Este caso de uso le permite al jugador o invitado cambia el idioma de la pantalla en la que se encuentre |
| **Precondiciones:** | 1. El jugador o invitado se encuentran en la ventana *Login.* |
| **Flujo Normal:** | 1. El jugador o invitado presiona el comboBox “Idioma”. 2. El sistema despliega las opciones “Español”, “English”. 3. El jugador selecciona la opción “English”.   [Flujo alterno A.3: El jugador presiona la opción “Español”]   1. El sistema cambia el idioma de la pantalla de Español a Inglés.   [Excepción E.2: Se pierde conexión con el servidor]   1. Fin de caso de uso. |
| **Flujos Alternos:** | A.3. El jugador o invitado presiona la opción “Español”  A.3.1 El sistema no cambia el idioma de la ventana.  A.3.1 Fin de caso de uso |
| **Excepciones:** |  |
| **Postcondiciones:** | La ventana cambió de idioma |

**Selección del Framework de persistencia**

El framework de persistencia utilizado para este proyecto fue JPA. Este mismo fue seleccionado por la compatibilidad que posee con Java pues, es su API de persistencia, por lo tanto cumple completamente con el estandar necesario. Este framework aprovecha la orientacion a objetos proveyendo mayor facilidad para persistir datos reduciendo el esfuerzo en las consultas y en la persistencia, ya que genera la ventaja de poder trabajar directamente con objetos de java con soporte a base de datos.

**Documentacion del codigo**

Ver carpeta PACKAGES.

**Manejo seguro de datos**

El proyecto contempla el manejo seguro de datos validando que ningun dato pueda ser persistido de forma nula o con caracteres que representen un espacio en blanco, asi mismo realiza la persistencia de la cuentas de los jugadores registrados almacenando solo la referencia de la contraseña por medio de un hashmap SHA-512, asi mismo antes de poder realizar cualquier accion el programa valida que las entradas no sean nulas. Para los registros de usuarios, el programa valida que no se hayan registrado usuarios con el mismo nombre ni el mismo correo electronico.

Para evitar disminucion en el rendimiento de la base de datos no se realizara el alamacenamiento de ninguna imagen, unicamente se guardara la ruta de dicha imagen para su posterior consulta.

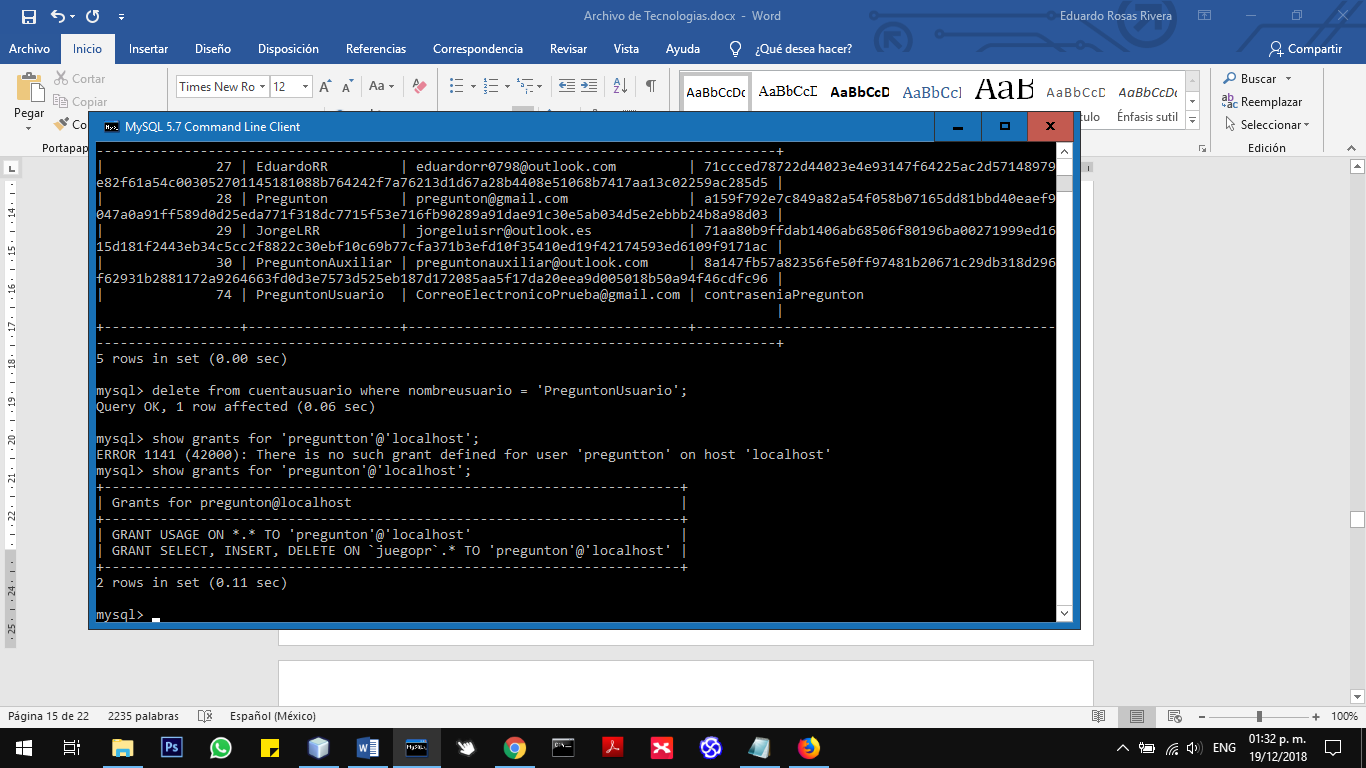
**Seguridad en la base de datos**

Para fines de este proyecto fue necesaria la creacion de un nuevo usuario, de este modo se evitaria el uso de root, el cual tiene todos los privilegios sobre la base de datos.

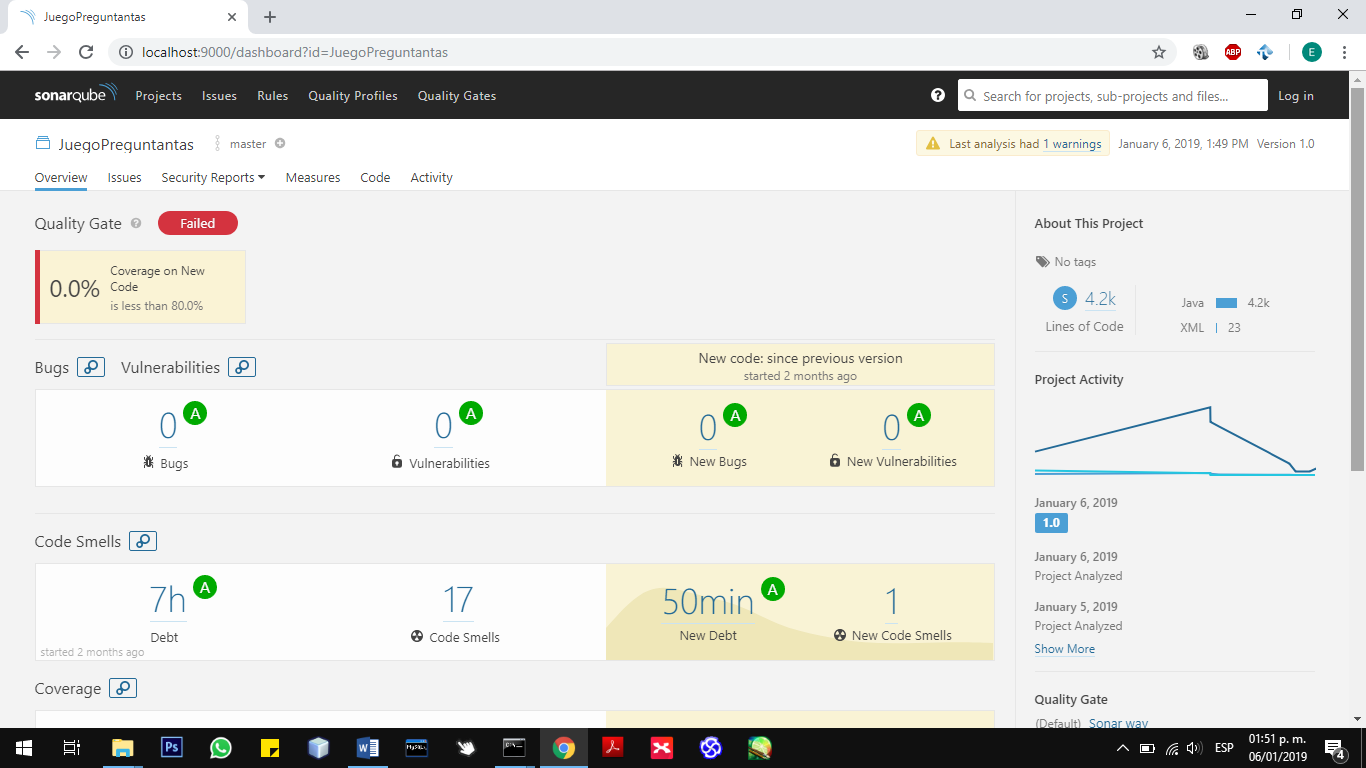
El nuevo usuario se llamó “pregunton” con la contraseña “PR3GUNT0N” al cual se le asignaron permisos unicamente para la base de datos “juegopr”, estos permisos consisten en:

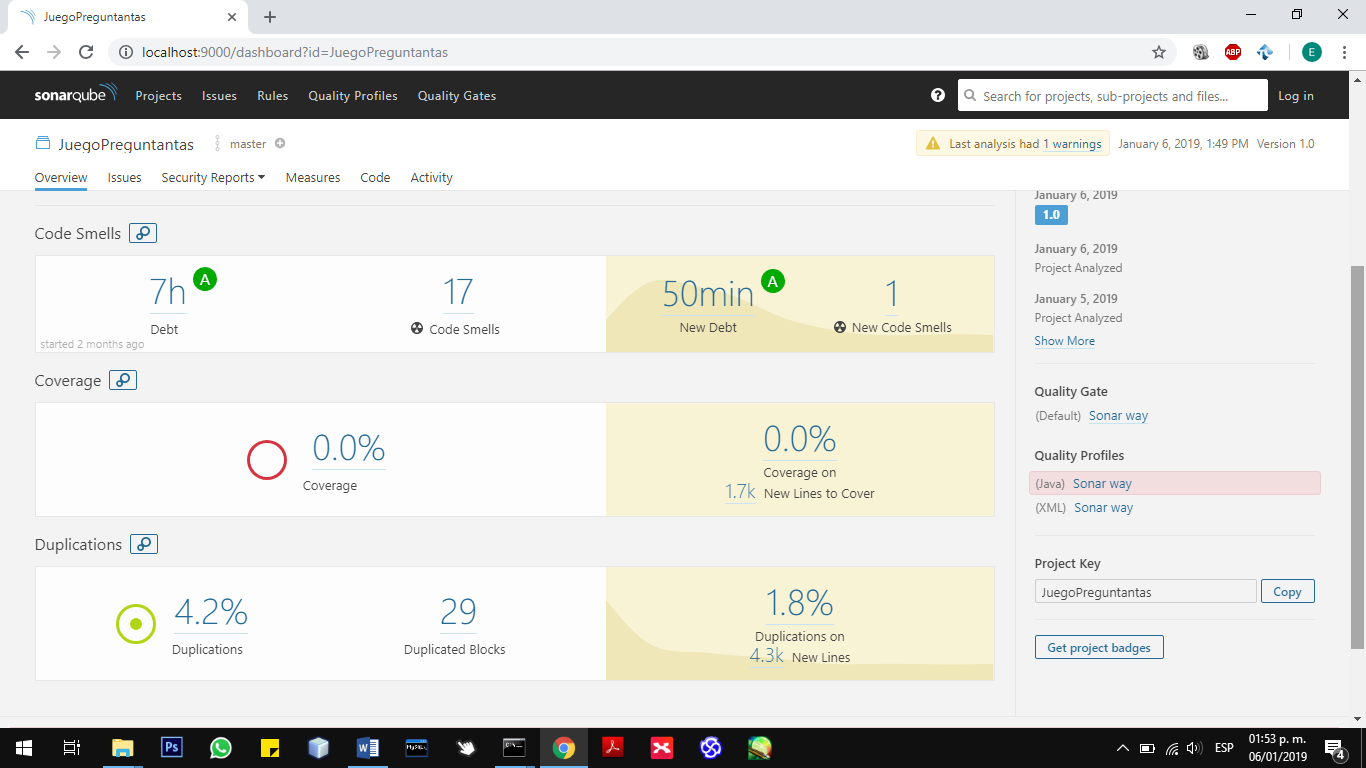
* Consultas de tipo SELECT.
* Consultas de tipo DELETE.
* Consultas de tipo INSERT

Estos permisos fueron asigandos con el proposito de que el usuario no pueda modificar la base de datos en lo que a su estructura y organización se refiere.



**Analisis estatico de codigo**

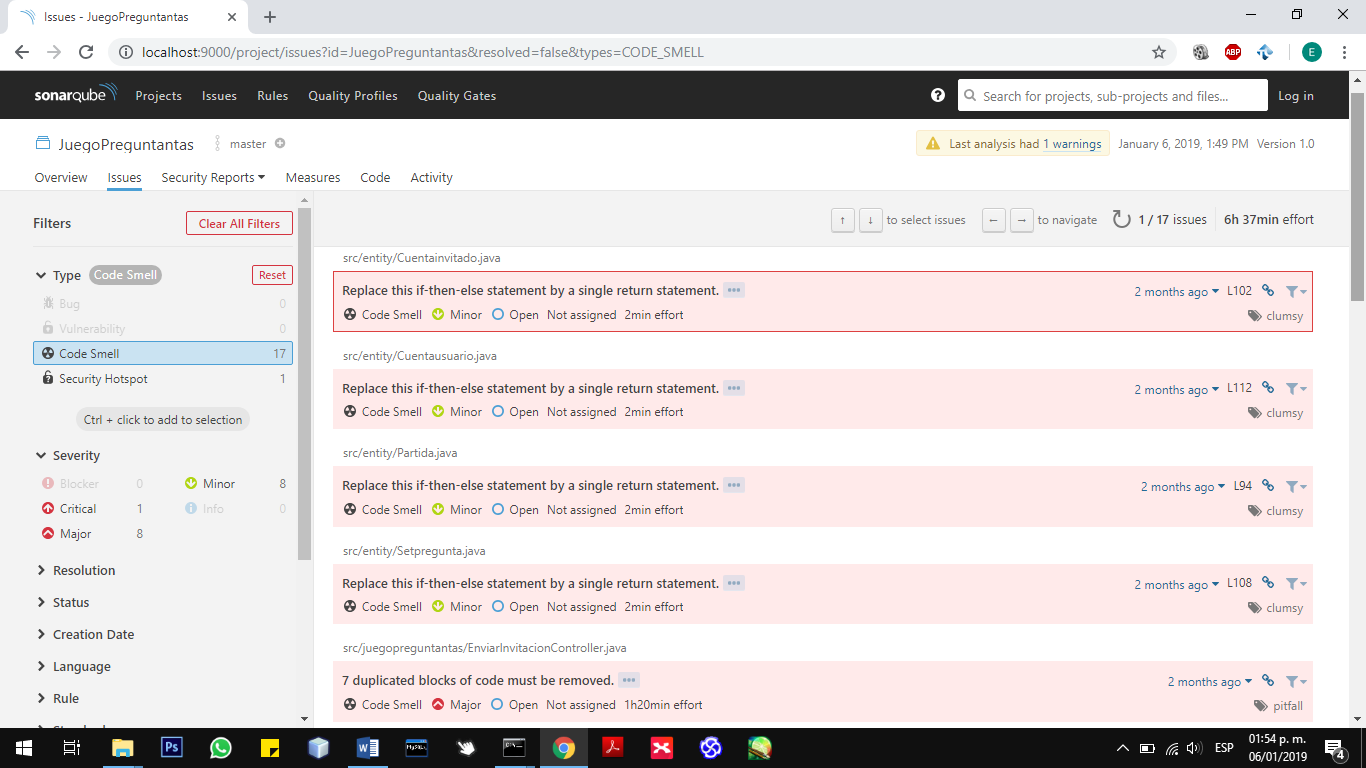


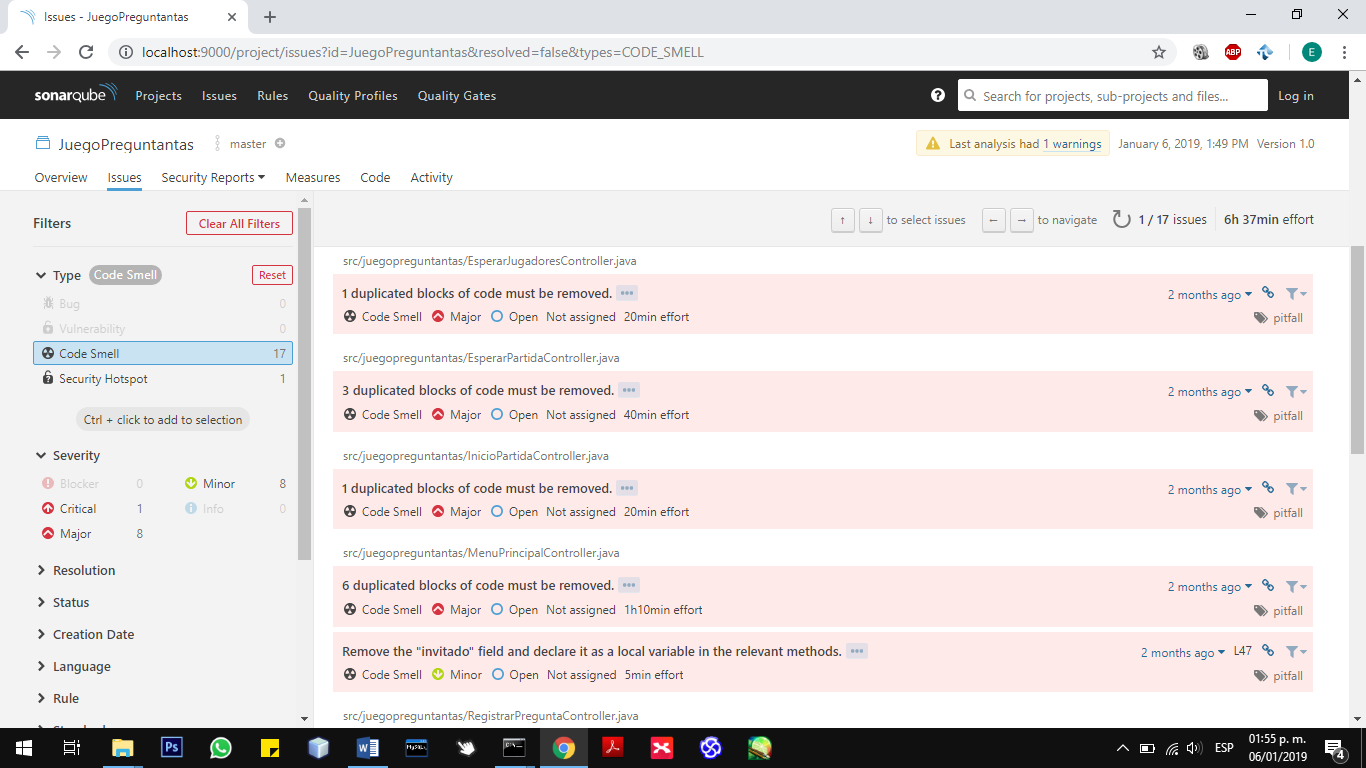


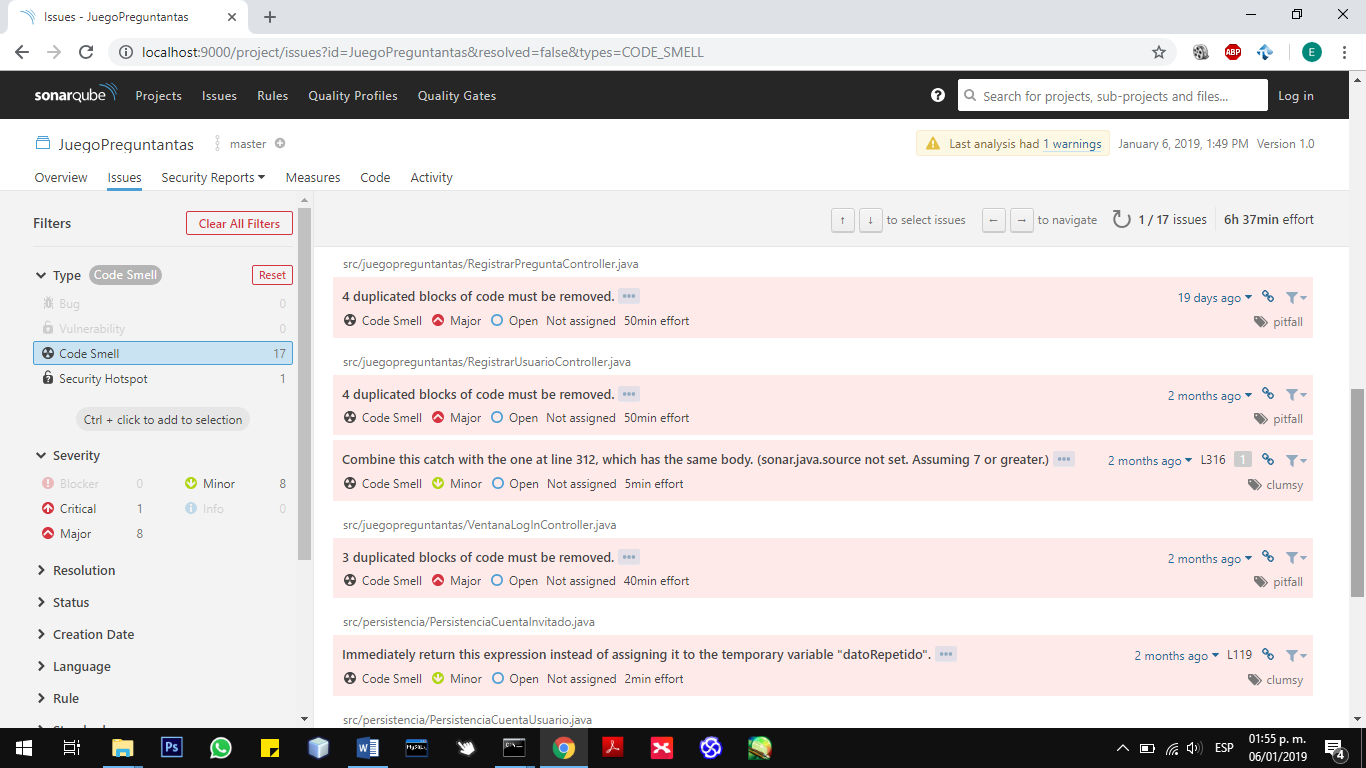
*Analisis Estatico de Codigo por SonarQube*

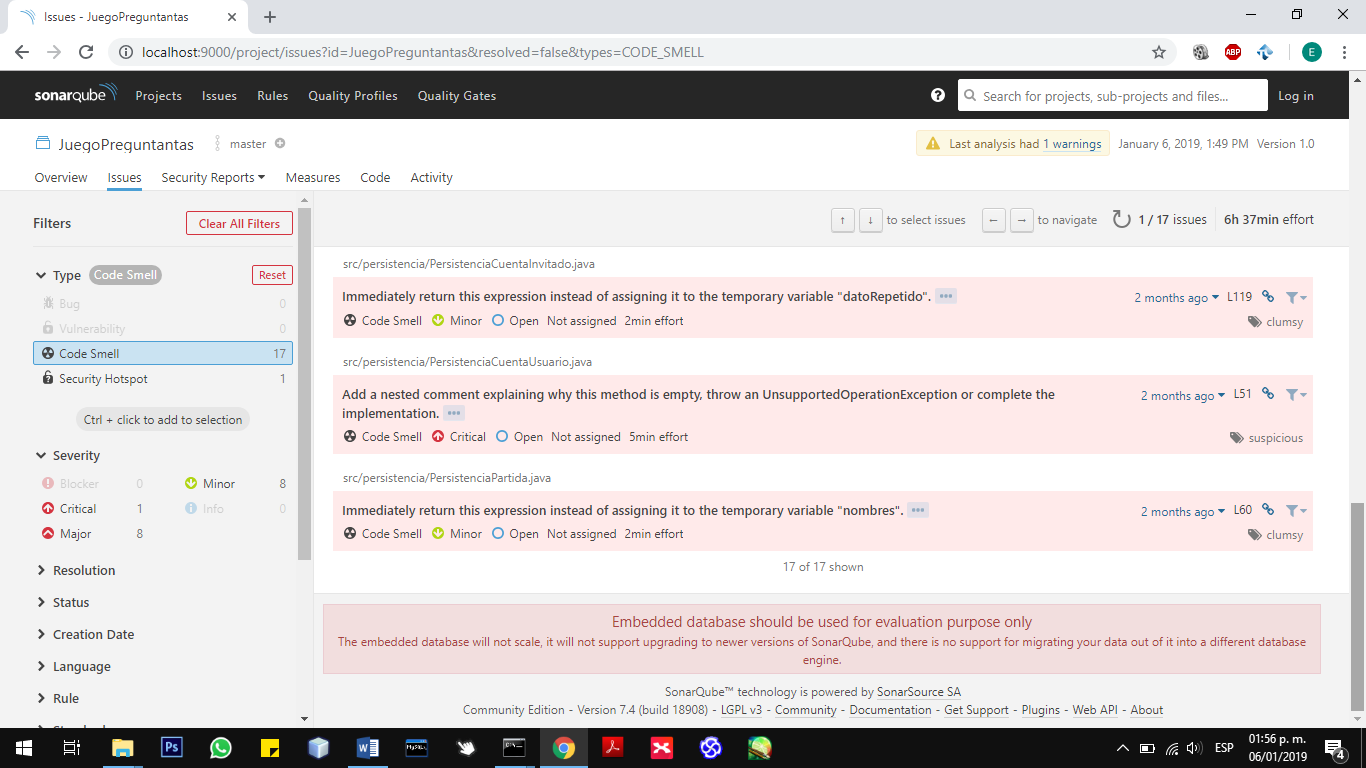
El analisis de codigo estatico expresa la ausencia de vunerabilidades y bugs, asimismo refleja la existencia de 1.8% de duplicaciones dentro del codigo (mismo que se compone de 4192 lineas de codigo) , las cuales consistencen en codigo para abrir ventanas tras la presion de un boton. Y la presencia de 17 code smells. El analisis arroja un 0% de covertura en las pruebas, sin embargo, estas si fueron realizadas.

Los code smells restantes fueron:









**Conclusiones**

**Eduardo**

El proyecto requirio el uso de muchas tecnologias para su implementacion, el uso de cada una de ellas redujo considerablemente el esfuerzo necesario para construirlo, sin embargo, estas mismas tecnologias requerian un tiempo necesario en su curva de aprendizaje, como fue el caso del uso de JPA y Sonarqube, herramientas en las que fue necesario dedicar mas tiempo a su aprendizaje que las demas, como SceneBuilder o NetBeans.

Tambien es importante hacer enfasis en el uso de las tecnologias que permiten la construccion colaborativa, como GitHub, de este modo me pude poner deacuerdo con mi compañero de equipo para la division de responsabilidades y ayudó a la realizacion de revisiones de codigo para el aumento de la calidad.

Por ultimo, me gustaria añadir que en la actualidad es muy importante aprovechar todas las tecnologias que se tengan al alcance de la mano y que sean adecuadas para construir software de calidad, sin embargo, estas siempre afectaran de buena y mala forma al proyecto.

**Puxka**

Este trabajo me suposo un reto, ya que algunas funciones (como la de enviar un corrreo electrónico) la tuve que investigar en internet, lo cual suena a algo a lo que debería de estar acostumbrado a estas alturas de la carrera pero normalmente era para dudas pequeñas o errores que no comprendía, nunca para algo tan nuevo.

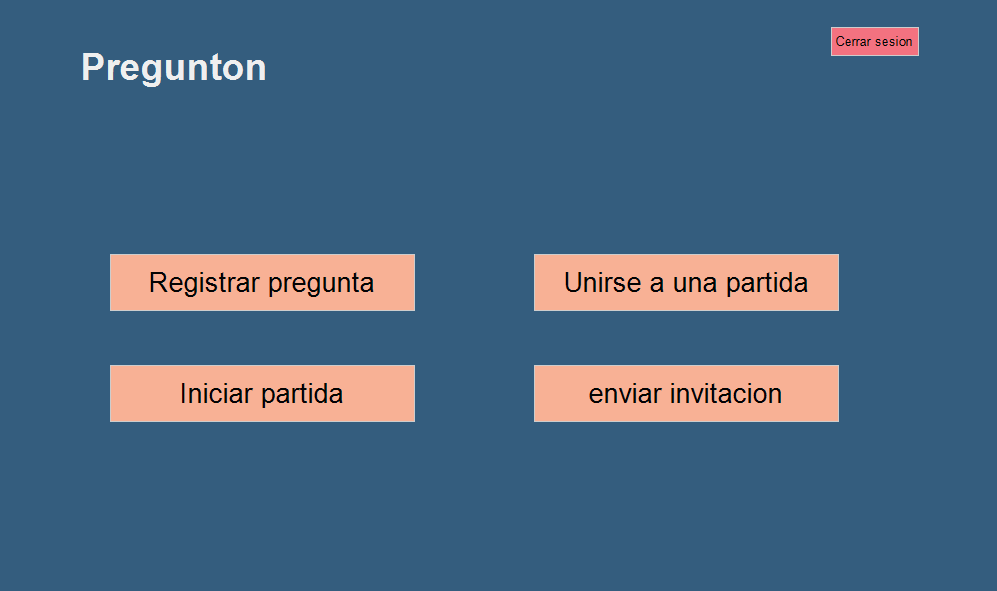
El tiempo fue de los principales enemigos, por que apesar de hacer el trabajo en equipo, hubo momentos en los que ninguno de los dos supo como avanzar. La parte que principalmente ocupo de mi tiempo fue la parte principal: el juego, ya que al trabajar con el socket, cometí muchos errores. Apesar de que yo fui el que hizo la actividad del chat por socket.io, muchos de los errores que tuve con ete fueron a raíz de que no había acabado de comprender el funcionamiento del socket y a causa de esto me preguntaba constantemente "¿Pero por qué no funciona?" cuando el principal error fueron mis suposiciones sobre lo que se supone que debía de hacer y no la realidad.

Hasta ahorita este es el proyecto más dificil en la carrera, por lo que aprendí muchas más cosas que en proyectos de semestres anteriores y en este proyecto también apliqué muchos conocimientos adquiridos en el semestre, como el uso los múltiples tipos de debugging, uso de frameworks y socket, seguridad, etc. Además, reforcé los conocimientos de la EE de Construcción de software sobre las buenas prácticas, a las cuales hice lo posible por apegarme.

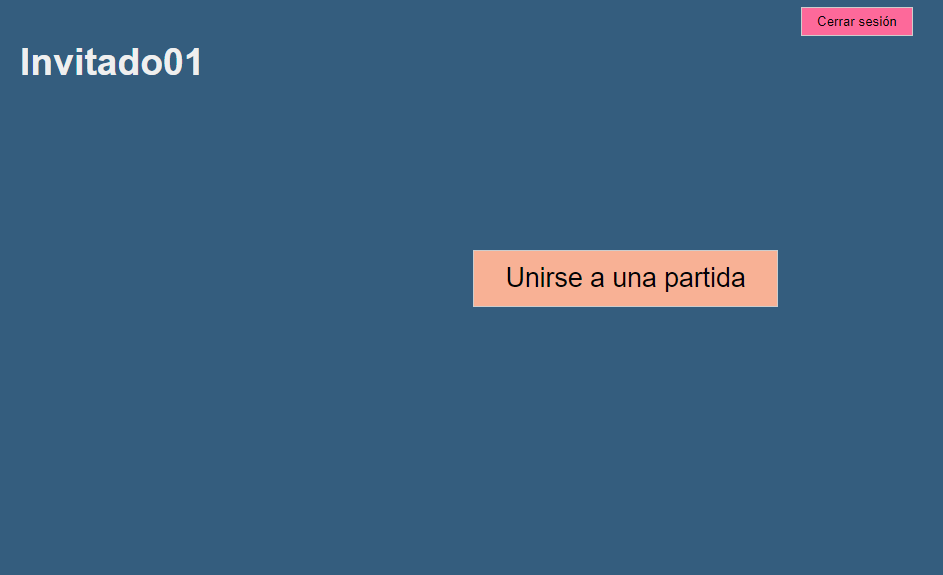
# Prototipos



Ventana *Login*



Ventana *Menú Principal*

**

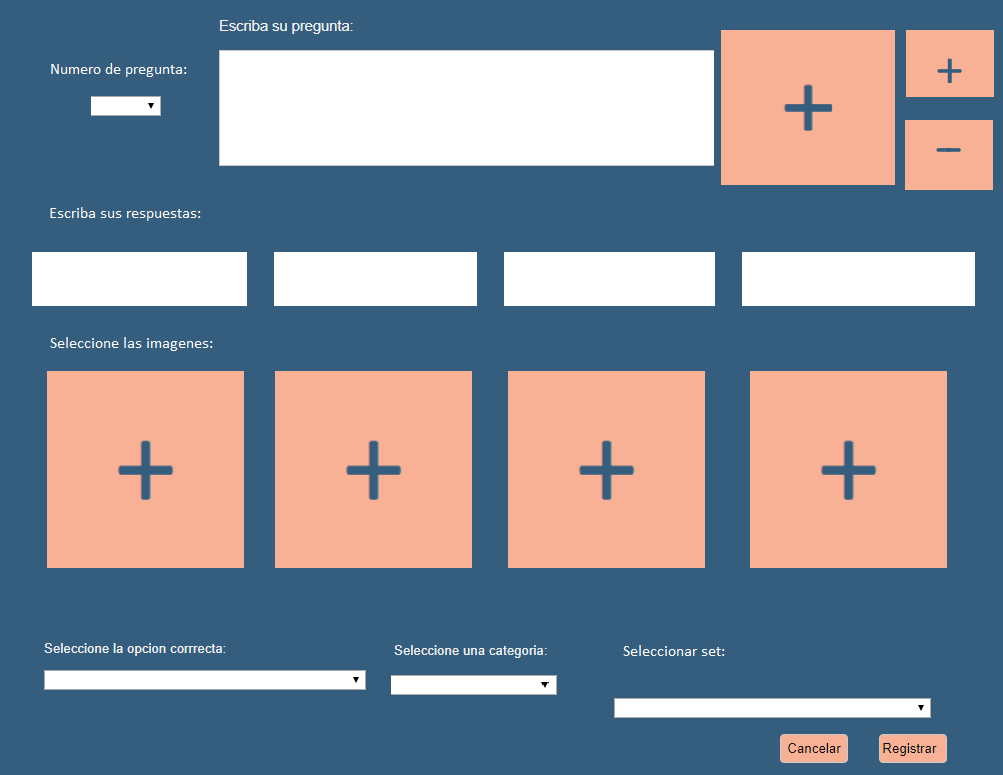
Ventana *Menú Principal Invitado*



Ventana *Jugadores Conectados*



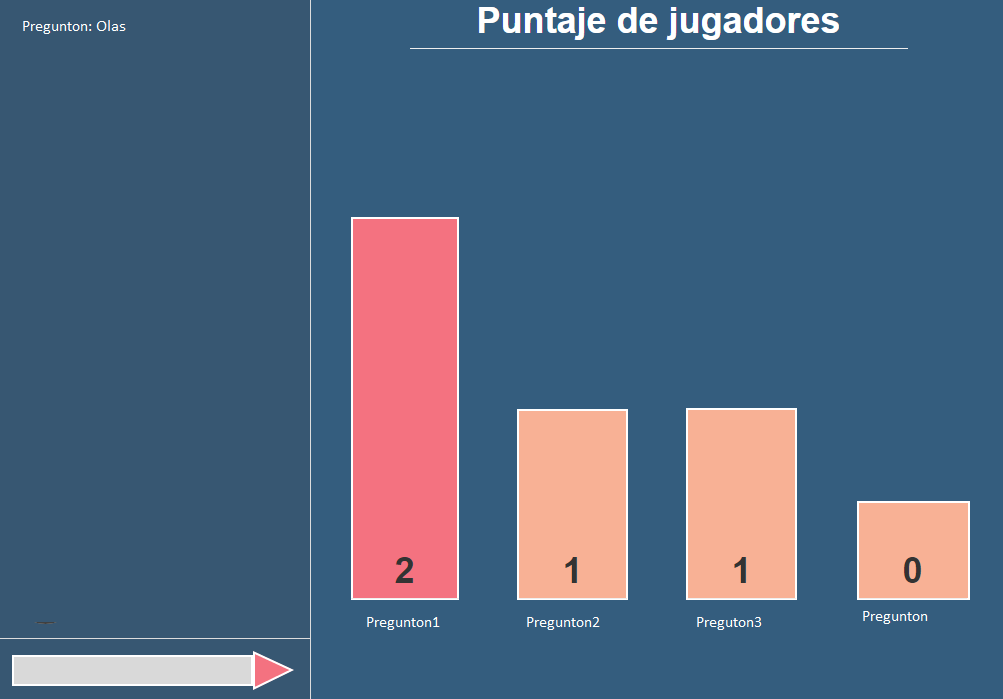
Ventana *Iniciar Partida*



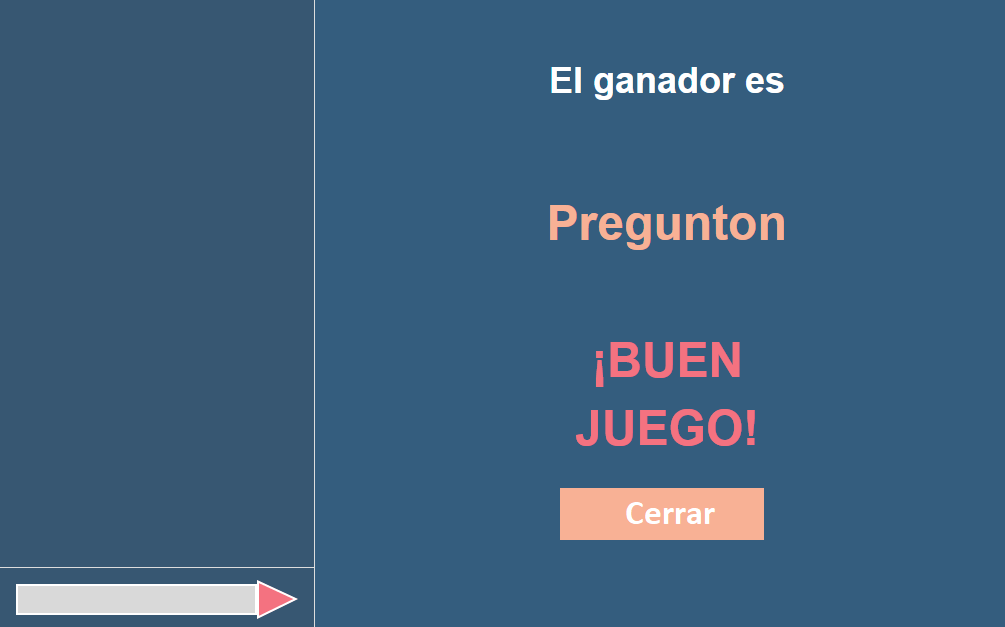
Ventana *Registrar Pregunta*



Ventana *Registro*



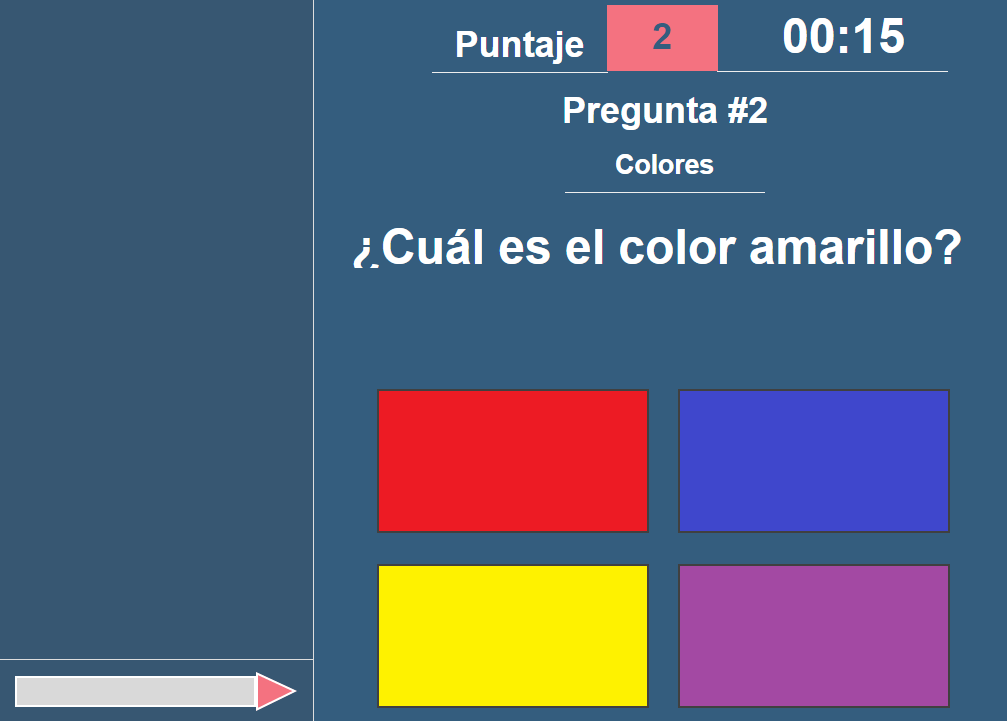
*Grafica de puntajes*

**

Ventana *Ganador*

**

Ventana *Invitar*

**

Ventana *Responder Pregunta*

# Referencias

Doug R., Matt S. (2007). Use Case Driven Object Modeling with UML. United States of America. Editorial Apress.

Karl W., Joy B. (2013). Software Requirements. United States of America. Editorial Microsoft