



- GRADO SUPERIOR EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA. MODALIDAD A DISTANCIA.
- TEAMDER, UNA APLICACIÓN PARA CONECTAR A JUGADORES EN LINEA.
- Tutor: Francisco Javier Navazo.
- Bernardo Manuel Da Silva Fernández.
- Diego Gómez Moreno.
- Gianluca Ruggeri Márquez.
- 20 de mayo de 2023. 2do Año.

Agradecimientos

Agradecemos de manera muy especial a nuestro tutor Francisco Javier Navazo quien nos ha prestado todo su tiempo, atención y soporte durante estos agitados meses mientras desarrollábamos “Teamder” y nos ha brindado su apoyo y conocimientos siempre que lo hemos necesitado.

Además, cabe destacar que, gracias a la ayuda de la Universidad Europea de Madrid en facilitarnos el acceso al IDE IntelliJ Ultimate. Nos ha ayudado mucho en el desarrollo de nuestra aplicación.

Gracias a la profesora Pilar por instruirnos en la materia de APA 7 y por su presentación la cual nos ha ayudado a construir de manera correcta nuestra documentación.

Tabla de Contenidos

Resumen	1
Abstract	2
Justificación del Proyecto	3
Objetivos	4
Objetivo General:	4
Objetivos Específicos:	4
Tecnologías y Herramientas Empleadas	7
Tecnologías	7
Herramientas	8
Enumeración de Requisitos	10
Requisitos funcionales:	10
Requisitos no funcionales:	11
Diagrama Modelo Entidad-Relación.	13
Diagrama de Gantt.	16
Diagrama de Casos de Uso.	17
Diagrama de Clases UML.	18
Diagrama de Flujo	24
Metodología Empleada	25
Metodología Ágil:	25
Presupuesto.....	26
Diseño	27
Prototipo	27
Logotipo y Diseño de Marca	32
Interfaces Finales	33
Conclusiones.....	37

Webgrafía y Bibliografía.....	39
-------------------------------	----

Índice de Figuras

Figura 1. Diseñador de Interfaz de Usuario	10
Figura 2. Encriptación de Contraseñas.	13
Figura 3. Diagrama de la Base de Datos (Modelo Entidad/Relación).	14
Figura 4. Diagrama de Gantt.....	16
Figura 5. Diagrama de Casos de Uso.	18
Figura 6. Diagrama de Clases UML (Completo).	19
Figura 6.1. Diagrama de Clases UML (Parte 1).....	20
Figura 6.2. Diagrama de Clases UML (Parte 2).....	21
Figura 6.3. Diagrama de Clases UML (Parte 3).....	22
Figura 6.4. Diagrama de Clases UML (Parte 4).....	24
Figura 7. Diagrama de flujo de navegación.....	24
Figura 8. Primeras Impresiones del "home".	28
Figura 9. Prototipo de logos, posibles opciones.	29
Figura 10. Rueda del color.	31
Figura 11. Correcciones en Photoshop.	32
Figura 12. Interfaz Final de Inicio de Sesión.	33
Figura 13. Interfaz Final de Registro.....	34
Figura 14. Producto final vista "HOME".	35
Figura 15. Interfaz Final de las Salas.....	35
Figura 16. Interfaz Final de las Opciones de Sala.	36

Resumen

El proyecto de fin de grado titulado "Teamder: una aplicación para conectar a jugadores en línea" aborda la problemática de la comunicación segura y efectiva entre usuarios durante el juego en línea. La justificación se basa principalmente en la necesidad de mejorar la experiencia de juego para el usuario, al proporcionar salas de juego con chat de voz y texto estables, sistemas de gestión de amistades e invitaciones a salas de chat. El objetivo general del proyecto es desarrollar una aplicación de escritorio completa, funcional y estable que promueva la colaboración y comunicación entre los jugadores. Se utiliza una metodología de desarrollo Agile y se emplean tecnologías como JavaFX, MySQL, protocolos TCP, GitHub, Java, Trello, IntelliJ IDEA y Photoshop.

"Teamder" busca mejorar la experiencia de juego de los usuarios al proporcionar una herramienta que facilita la comunicación efectiva y estable, promoviendo la colaboración y la socialización entre los jugadores. Se busca lograr esto a través de una interfaz gráfica de usuario atractiva, un sistema de registro seguro, chats de texto y voz seguros y estables, así como un sistema de gestión de amistades y salas de juegos.

El proyecto también identifica posibles mejoras y ampliaciones, como la integración con cuentas de Google para facilitar el registro e inicio de sesión, y la integración con redes sociales para facilitar la búsqueda de jugadores. Estas mejoras podrían aumentar el valor de la aplicación y brindar una mejor experiencia al usuario final.

En resumen, el proyecto "Teamder" se centra en el desarrollo una aplicación de escritorio que mejora la comunicación y colaboración entre jugadores en línea. A través de una interfaz intuitiva, un sistema de registro seguro y funciones de chat y gestión de amistades se busca proporcionar a los usuarios una experiencia emocionante y enriquecedora. El proyecto destaca la importancia de lograr una comunicación fluida, segura y satisfactoria entre los jugadores en línea.

Abstract

The undergraduate project entitled "Teamder: an application to connect online players" addresses the issue of secure and effective communication between users during online gaming. The justification is mainly based on the need to improve the gaming experience for users by providing stable voice and text chat rooms, friendship management systems, and chat room invitations. The overall objective of the project is to develop a comprehensive, functional, and stable desktop application that promotes collaboration and communication among players. An Agile development methodology is employed, and technologies such as JavaFX, MySQL, TCP protocols, GitHub, Java, Trello, IntelliJ IDEA, and Photoshop are utilized.

"Teamder" aims to enhance the gaming experience for users by providing a tool that facilitates effective and stable communication, promoting collaboration and socialization among players. This is achieved through an attractive user interface, a secure registration system, secure and stable text and voice chats, as well as a friendship and game room management system.

The project also identifies potential improvements and expansions, such as integrating with Google accounts for easier registration and login, and integrating with social networks to facilitate player search. These enhancements could increase the value of the application and provide a better experience for end users.

In summary, the "Teamder" project focuses on developing a desktop application that improves communication and collaboration among online players. Through an intuitive user interface, a secure registration system, and chat and friendship management features, the project aims to provide users with an exciting and enriching experience. The project highlights the importance of achieving smooth, secure, and satisfactory communication among online players.

Justificación del Proyecto

En la actualidad, muchos jugadores se pueden encontrar muchas dificultades al momento de comunicarse de manera efectiva y segura durante el juego en línea, esto puede afectar de forma negativa a su experiencia de juego. Por lo que nuestra aplicación aborda esta problemática al ofrecer una alternativa en la que los jugadores tienen a su disposición salas de juego, con chat de voz y de texto estables, sistemas de gestión de amistades e invitaciones de unión a salas.

La justificación de nuestro proyecto, "Teamder", radica en la necesidad de mejorar la experiencia de juego en línea entre jugadores, promoviendo la colaboración y comunicación entre ellos. En la actualidad, existen diversas plataformas de juegos en línea, pero no todas ofrecen una comunicación segura y fluida entre jugadores.

El objetivo completo de este proyecto es desarrollar una aplicación de escritorio intuitiva, completa, funcional y estable para la comunicación y colaboración entre usuarios jugadores en línea. Para lograrlo, el equipo de desarrollo de "Teamder" seguirá una metodología de desarrollo Agile, incorporación de tecnologías conocidas como: JavaFx, MySQL, protocolos TCP, Github, Java, Trello, IntelliJ IDEA y Photoshop. De las cuales también obtendrán conocimientos más profundos y prácticos de cada una de las tecnologías anteriormente nombradas.

En conclusión, el proyecto tiene como finalidad ofrecer una mejora en la experiencia de juego de los usuarios, poniendo a disposición una herramienta que facilite un sistema de comunicación efectiva y estable que promueva la colaboración y socialización entre jugadores. Esto, a través de una interfaz grafica de usuario atractiva, un sistema de registro seguro, chats de texto y de voz seguros y estables, así como un sistema de gestión de amistades y salas de juegos. Para finalmente ofrecer a los usuarios una experiencia de juego mas satisfactoria y emocionante.

Objetivos

Objetivo General:

Desarrollar una aplicación de escritorio dirigida principalmente a jugadores, que permita a los usuarios registrados conectar entre sí y mantener conversaciones de voz y de texto a tiempo real a través de salas de chat, así como disfrutar de una experiencia de juego en grupos con sus amigos, mediante sistema de chat de voz y texto estable y seguro, implementando un sistema de solicitudes de amistad, proporcionando una interfaz grafica de usuario atractiva e intuitiva y un sistema de registro de usuarios seguro y confiable. Para que, de esta forma, el usuario final tenga una experiencia de comunicación en línea fluida, segura y satisfactoria, promoviendo la interacción y colaboración entre jugadores.

Objetivos Específicos:

1. Crear una interfaz gráfica de usuario atractiva e intuitiva que permita a los usuarios poder navegar y utilizar la aplicación con facilidad.
2. Proporcionar a los jugadores un sistema de registro de usuarios seguro y confiable en el manejo de sus contraseñas, mediante contraseñas encriptadas con un algoritmo "SHA-256".
3. Implementar un sistema de chat de texto y chat de voz estable y seguro, que permita a los usuarios comunicarse entre ellos de una manera cómoda y efectiva.
4. Integrar un sistema de gestión de amistades que permita a los usuarios agregar a otros usuarios a su lista de amistades. Y un sistema de invitaciones a salas de juegos, que permita a los usuarios tener una experiencia en grupo con sus amigos.

Introducción

En el contexto actual de los juegos en línea, uno de los elementos clave para tener una experiencia de juego enriquecedora y gratificante es la comunicación fluida y la interacción con otros jugadores. Así que, con este objetivo en mente se ha desarrollado este proyecto de fin de grado titulado “Teamder, una aplicación para conectar a jugadores en línea”. El cual se basa principalmente en la creación de una aplicación de escritorio que permita a los usuarios registrados conectarse entre sí para mantener conversaciones de voz y texto en tiempo real a través de salas de chat, y que así puedan disfrutar de gratificantes experiencias de juego con sus amigos.

El objetivo principal de “Teamder” es ofrecer a los jugadores una aplicación segura y confiable que impulse las relaciones y colaboraciones entre usuarios en línea. Y para lograrlo se han planteado una serie de objetivos específicos. En primer lugar, se ha desarrollado una interfaz gráfica atractiva e intuitiva que permite a los jugadores utilizar la aplicación de una forma cómoda y sencilla.

En cualquier plataforma de comunicación en línea, una de las preocupaciones más importante es la seguridad. Por lo que se ha implementado un sistema de registro de nuevos usuarios seguro y confiable que utiliza un sistema de encriptación mediante el uso de algoritmos “SHA-256”. Por lo que se garantiza la protección de las contraseñas de los usuarios, dando a los mismos tranquilidad y confidencialidad.

Una parte vital de la experiencia de juego de los usuarios es la comunicación entre ellos. Por lo que se ha desarrollado un sistema de chat de voz y chat de texto estable y seguro, lo que permite a los usuarios comunicarse entre ellos fomentando la colaboración y el trabajo en equipo durante el juego.

También, “Teamder” incluye un sistema de solicitudes de amistades, permitiendo así a los usuarios, agregar otros jugadores a su lista de amistades y ampliar sus conexiones en línea. A su vez, se ha integrado un sistema de invitaciones a salas de juegos, brindando a los jugadores la oportunidad de jugar y disfrutar de emocionantes experiencias de juego junto a sus amigos.

En resumen, “Teamder” es un proyecto que tiene como objetivo complementar la diversión del juego en línea con una comunicación efectiva y segura entre jugadores. Gracias a una interfaz grafica de usuario intuitiva, un sistema de registro de usuarios seguro y confiable, diversas funciones de chat y gestión de amistades y salas, se pretende proporcionar a los jugadores una experiencia emocionante y enriquecedora. A lo largo de este documento, se irán detallando los aspectos técnicos y proceso de desarrollo de esta aplicación, así como los resultados obtenidos y las conclusiones a las que se han llegado.

Tecnologías y Herramientas Empleadas

Tecnologías

La elección de las tecnologías adecuadas para nuestro proyecto es crucial para su correcto desarrollo. Después de investigar y basándonos en nuestros conocimientos actuales, hemos seleccionado cuidadosamente las tecnologías que consideramos más adecuadas para nuestras necesidades específicas:

- Base de datos: MySQL como nuestro principal gestor de bases de datos debido a su eficiencia y fácil manejo; además de ser el sistema gestor de bases de datos relacional que tratamos y estudiamos en estos dos años de curso.

"MySQL es un sistema de gestión de bases de datos relacional de código abierto, que se utiliza comúnmente en aplicaciones web y de servidor. MySQL es altamente escalable y eficiente, y puede manejar grandes cantidades de datos con facilidad." (Gupta, 2020, p. 38).

- Interfaz gráfica: JavaFX es nuestra elección para la interfaz gráfica de usuario debido a su facilidad de uso y capacidad para crear interfaces atractivas y personalizadas. Además de ser sumamente intuitivo a la hora de crear la interfaz con elementos de todo tipo, utilizando como principal herramienta Scene Builder de la cual hablaremos en detalle más adelante.

"JavaFX es una biblioteca de software libre de Oracle utilizada para crear aplicaciones interactivas de escritorio, móviles y web. Es una plataforma de desarrollo de software para la creación de interfaces gráficas de usuario que

ofrece una amplia variedad de componentes gráficos personalizables, efectos visuales y animaciones. Además, se integra con otros componentes de Java, como Swing, para facilitar la creación de aplicaciones más complejas.”

- **Protocolos de comunicación:** Utilizamos conexiones de tipo TCP para realizar el intercambio de mensajes entre usuarios, ya sean de voz o texto, conseguimos este protocolo como el más efectivo a la hora de desarrollar el código para las comunicaciones entre usuarios.

En resumen, hemos elegido estas tecnologías después de una investigación detallada y basándonos en nuestros conocimientos actuales para asegurarnos de que nuestro proyecto sea desarrollado con las herramientas adecuadas para lograr nuestros objetivos.

Herramientas

1. **GitHub:** Se utilizó en el desarrollo de “Teamder” para el control de versiones, permitiéndonos trabajar de manera colaborativa, rastrear y revertir cambios cuando era necesario. Además, GitHub también ofrece funcionalidades como solicitudes de extracción y revisión de código, lo cual ha facilitado el trabajo en equipo y aseguró la calidad del código.
2. **IntelliJ IDEA:** Fue nuestra herramienta principal para escribir el código de “Teamder” debido a sus características de autocompletado inteligente, análisis de código, y su potente sistema de depuración que nos ayudaba a encontrar los errores y solucionar los problemas de manera más eficiente.

3. **Scene Builder:** Se utilizó en el desarrollo de "Teamder" para diseñar y desarrollar la interfaz de usuario de la aplicación debido a que es muy intuitivo y permitía que, al guardar los cambios hechos en la interfaz, se tradujeran a código FXML que luego podíamos utilizar a la hora de crear funcionalidades, acceso a vistas mediante botones, etc.
4. **Trello:** Fue clave para el desarrollo de Teamder gestionar las tareas del proyecto, seguir su progreso y mejorar la colaboración y la comunicación entre nosotros, aumentó nuestra organización y tiempos estimados para realizar nuestras tareas tanto individuales como grupales.
5. **Java:** Se utilizó como el lenguaje de programación principal en el desarrollo de Teamder debido a su compatibilidad con múltiples sistemas operativos y su sólido soporte para la programación orientada a objetos respaldada con años de fiabilidad, que facilita la creación de aplicaciones. Además de ser el lenguaje principal en este curso, hemos querido utilizarlo para demostrar todo lo aprendido y seguir formándonos en Java.
6. **Photoshop:** Fue utilizado para realizar las modificaciones pertinentes a nuestro logotipo para que fuera apto y lo más cercano a lo que buscábamos, esta herramienta fue una parte que puede no parecerlo, pero fue sumamente importante ya que aquí se crea de manera definitiva el logo que va a representar nuestra aplicación de cara al cliente final y que consideramos es crucial para incitar a la gente a usarla.

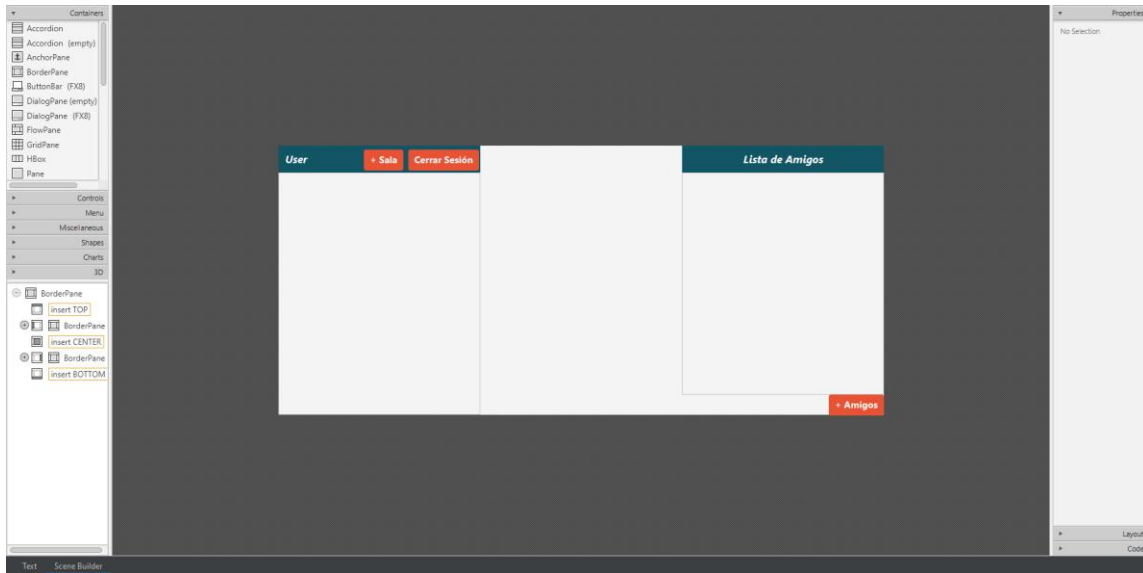


Figura 1. Diseñador de Interfaz de Usuario.

Enumeración de Requisitos

Requisitos funcionales:

Registro de usuario: Los usuarios deben poder registrarse en la aplicación proporcionando detalles como nombre de usuario, correo electrónico y contraseña.

Inicio de sesión: Los usuarios deben poder iniciar sesión en sus cuentas usando sus credenciales registradas.

Agregar amigos: Los usuarios deben poder agregar a otros usuarios a su lista de amigos.

Chat de texto y voz: Los usuarios deben poder comunicarse con sus amigos a través de mensajes de texto y llamadas de voz.

Creación y búsqueda de salas de juegos: Los usuarios deben poder crear salas de juegos e invitar a amigos a unirse mediante una invitación que llega a tiempo real.

Invitar a amigos a salas de juegos: Los usuarios deben poder invitar a sus amigos a unirse a sus salas de juegos.

Almacenamiento de mensajes de chat: Los mensajes de chat en las salas de chat deben ser almacenados en la base de datos para que sean accesibles en futuras sesiones.

Requisitos no funcionales:

Seguridad: La aplicación debe implementar medidas de seguridad adecuadas, incluyendo la encriptación de contraseñas y la protección de los datos del usuario.

Rendimiento: La aplicación debe responder rápidamente a las acciones del usuario, sin retrasos notables.

Disponibilidad: La aplicación debe estar disponible para su uso en cualquier momento, con un tiempo mínimo de inactividad.

Escalabilidad: La aplicación debe ser capaz de manejar un aumento en el número de usuarios sin sufrir una disminución en el rendimiento.

Usabilidad: La aplicación debe ser fácil de usar, con una interfaz de usuario intuitiva y amigable.

Finalmente queremos complementar el punto de seguridad describiendo uno de los aspectos mas importantes de la implementación de esta en nuestra aplicación.

El proceso de encriptación de la contraseña empieza con la generación de un "salt" gracias a la clase "SecureRandom" de la librería "java.security". Este salt es un numero aleatorio seguro de 16 bytes y será utilizado para reforzar la encriptación de la contraseña.

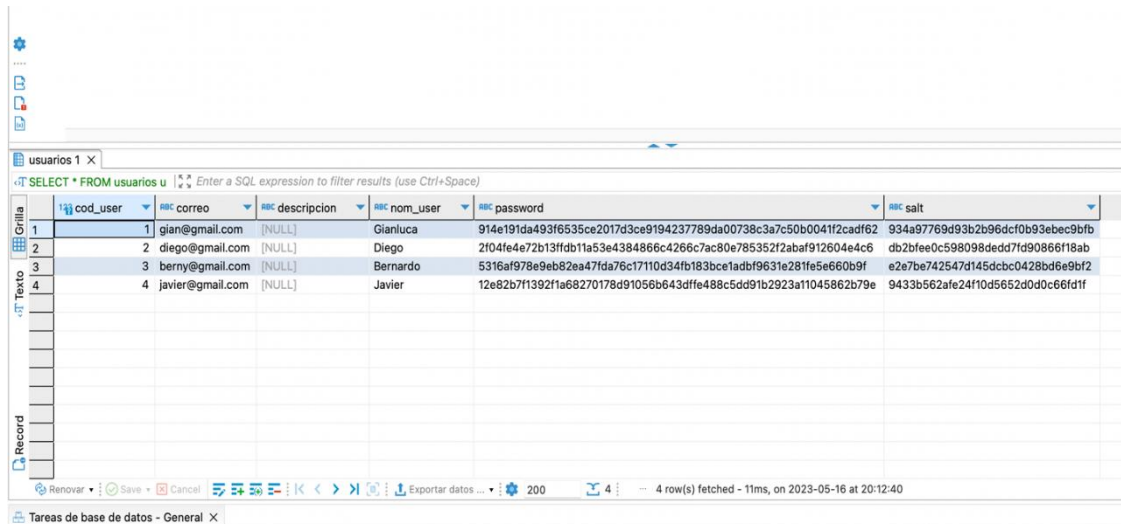
Una vez generado el salt, el proceso sigue con la incorporación de un MessageDigest que utiliza un algoritmo de encriptación "SHA-256" y a este se le agrega el salt al proceso de encriptación. Para luego calcular el hash de la combinación de la salt y la contraseña, y así conseguir finalmente un arreglo de bytes que se convertirá a un bloque hexadecimal para almacenarlo en la base de datos como el "password" del usuario. Además también se guarda en la base de datos el correspondiente "salt" de la contraseña convertido a un bloque hexadecimal.

Cabe destacar que este sistema de cifrado es irreversible, es decir, no se puede generar de nuevo la contraseña a partir de la contraseña y/o el salt almacenados en la base de datos. Por lo que al momento de evaluar si la contraseña es correcta para el usuario que quiere iniciar sesión, se deberá recuperar el salt almacenado para ese usuario en la base de datos. Luego generar con el mismo proceso una contraseña encriptada, pero esta vez utilizando el salt recuperado de la base de datos. De esta forma, lo que queda es evaluar si el bloque hexadecimal de la contraseña almacenada en la base de datos es igual al bloque hexadecimal de la contraseña que ha introducido el usuario.

Finalmente, una vez evaluadas las dos contraseñas encriptadas, dejaremos o no entrar al usuario a la aplicación.

Todo este proceso hace la función de protección de la contraseña original para evitar cualquier ataque de fuerza bruta a la contraseña del individuo por parte de hackers o malware. Esto nos ayuda a asegurar la integridad y seguridad de nuestros usuarios.

En la siguiente figura tenemos un ejemplo de registros de la tabla de usuarios, en la que podemos observar como se encuentran almacenada la contraseña y su correspondiente "salt".



usuarios 1 X

SQL: `SELECT * FROM usuarios u` Enter a SQL expression to filter results (use Ctrl+Space)

	cod_user	correo	descripcion	nom_user	password	salt
1	1	gian@gmail.com	[NULL]	Gianluca	914e191da493f6535ce2017d3ce9194237789da00738c3a7c50b0041f2cadf62	934a97769d93b2b96dcf0b93ebec9bfb
2	2	diego@gmail.com	[NULL]	Diego	2f04fe4e72b13f1db11a53e4384866c4266c7ac80e785352f2abaf912604e4c6	db2bfee0c598098dedd7fd90866f18ab
3	3	berny@gmail.com	[NULL]	Bernardo	5316af978e9eb82ea47fda76c17110d34fb183bce1adb9631e281fe5e660b9f	e2e7be742547d145dcbc0428bd6e9bf2
4	4	javier@gmail.com	[NULL]	Javier	12e82b7f1392f1a68270178d91056b643dffe488c5dd91b2923a11045862b79e	9433b562afe24f10d5652d0d0c66fd1f

Renovar Save Cancel Exportar datos 200 4 4 row(s) fetched - 11ms, on 2023-05-16 at 20:12:40

Tareas de base de datos - General X

Figura 2. Encriptación de Contraseñas.

Diagrama Modelo Entidad-Relación.

En el siguiente diagrama Entidad/Relación, se puede apreciar como las tablas se relacionan entre ellas según las interacciones que cumplen. Cada una de ellas son de carácter auto-incremental debido a que son tablas que siempre van a cargar un dato nuevo y se va a almacenar de manera automática según lo que el usuario solicite.

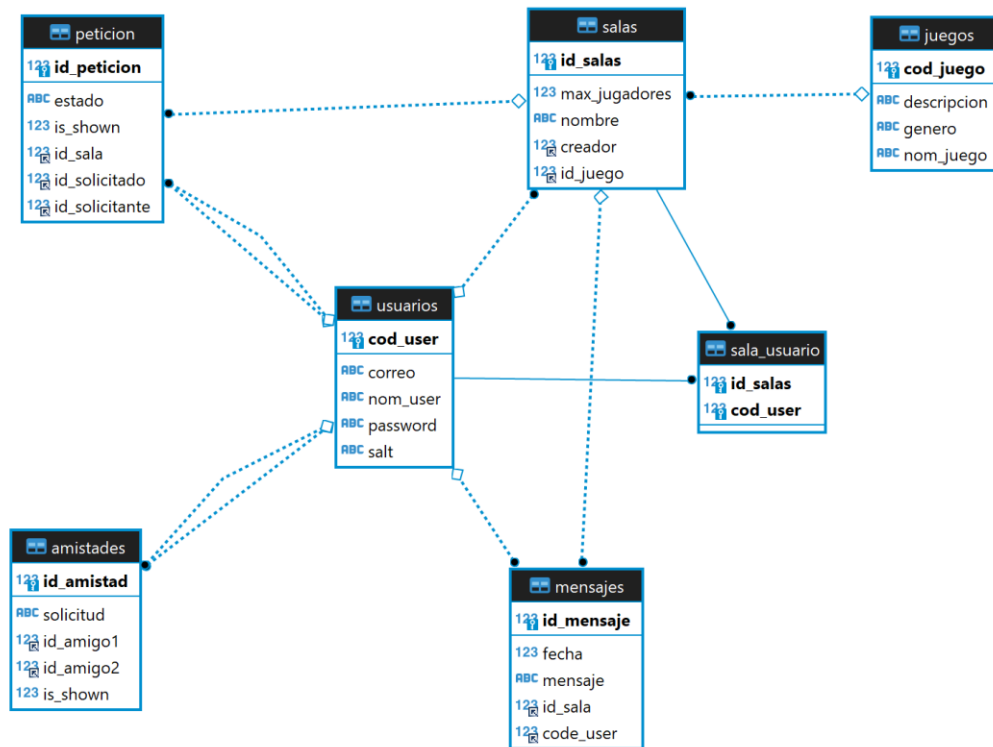


Figura 3. Diagrama de la Base de Datos (Modelo Entidad/Relación).

La tabla “usuarios” es la principal debido a que el núcleo de nuestra aplicación posee las columnas necesarias para guardar los datos del usuario y que sean únicos e irrepetibles, los cuales serán almacenados una vez el usuario ingrese sus registros al inicio de la aplicación.

- En torno a la tabla “usuarios” giran el resto de las tablas;
- La tabla “amistades” es una de las más importantes ligadas a “usuarios”, porque esta aplicación busca la interacción entre los usuarios para mejorar la experiencia de juego. Con un valor “pendiente” por defecto en la columna “solicitud” hasta que el usuario que haya recibido la solicitud la acepte, rechace o bloquee al usuario solicitante.

- La tabla "sala_usuario" es crucial, ya que es la encargada de que los usuarios puedan añadir a sus amigos a la sala y una vez estos dentro, puedan hablar tanto por mensajería como por voz, almacenando a los usuarios dentro de la sala, aunque la aplicación se cierre. Solo por elección del creador de la sala, podrá sacar a las personas que están dentro.
- "Salas", va a determinar el máximo de jugadores permitidos en la conexión entre los jugadores, llevará como título principal un nombre que quieran darle los usuarios a su libre albedrío y el juego que hayan escogido en común para reunirse, de la mano con la sala "mensajes" va a ser capaz de almacenar cada una de las conversaciones, aunque se cierre la aplicación.
- La tabla "mensajes" es la encargada de comunicar a nuestros usuarios.

Es importante aclarar que dentro de la columna que estará siempre visible será solo para mensajes escritos.
- Por último, pero no menos importante, tenemos la tabla "juegos" que está directamente conectada con "salas" debido a que, al crear una sala, uno de los requisitos para que esta función sea posible, es seleccionar un juego entre la amplia variedad que tenemos de juegos online, de modo que la lista de salas que aparezca a la izquierda de la pantalla, sea más ordenada.

Diagrama de Gantt.

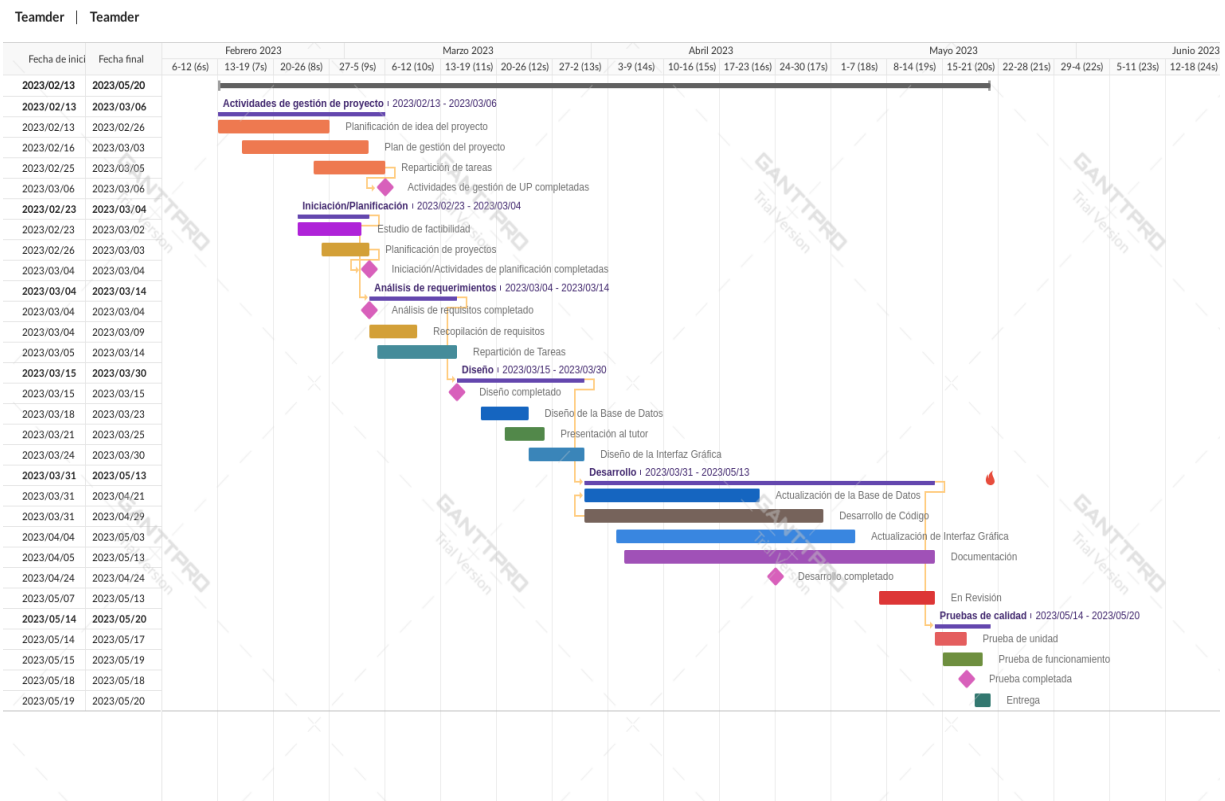


Figura 4. Diagrama de Gantt.

Con este diagrama de Gantt, se puede apreciar la línea de tiempo de como cronológicamente se ha desarrollado nuestra aplicación con fecha de inicio 13 de febrero de 2023, terminando el 20 de mayo de 2023. Ordenando meticulosamente cada tarea que hemos llevado a cabo y cuánto ha durado dicha tarea. Esto sirve para llevar un control exhaustivo del desarrollo de la aplicación y ayuda a una mejor organización. Las barras equivalen al tiempo que ha durado la tarea y a un lado de las barras se puede apreciar el nombre.

En la pestaña de arriba se puede ver el número de semanas que tiene cada mes y va acorde con las barras y el tiempo empleado en cada tarea.

Diagrama de Casos de Uso.

En este diagrama de casos de uso podemos observar de una forma sencilla el funcionamiento de nuestra aplicación de cara al usuario. En el que el actor es el "Usuario", y el sistema es nuestra "Aplicación". El usuario tendrá la posibilidad de iniciar sesión (caso en el que esté registrado en el sistema), o podrá registrarse (caso en el que no esté registrado en el sistema). Para luego iniciar sesión ("include"). De esta forma, una vez el usuario haya iniciado sesión podrá acceder al "Home" donde están las funciones de añadir amigos y "crear sala" donde podrá seleccionar un juego, el nombre de la sala y seguido esto, crearla donde aparecerá en un menú a la izquierda de la aplicación. Una vez dentro de la sala podrá:

- Invitar a un amigo (caso en el que el usuario tenga amigos y estos estén conectados).
- Iniciar el chat de voz.
- Iniciar el chat de texto.
- Salir de la sala.

Esta última opción tendrá como resultado la posibilidad de elegir de nuevo otro juego, y de esta forma poder realizar de nuevo los pasos posteriores a la selección de juego.

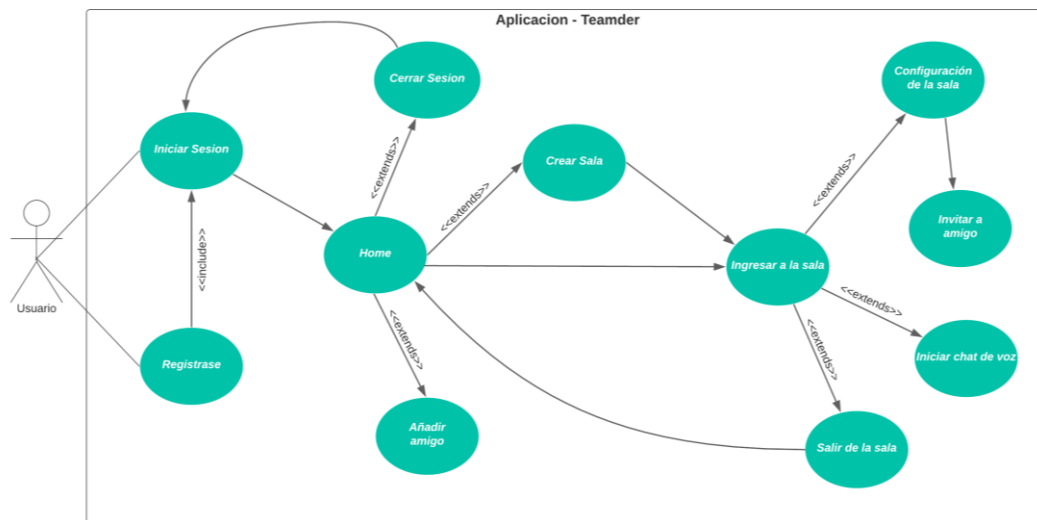


Figura 5. Diagrama de Casos de Uso.

Diagrama de Clases UML.

Este diagrama de clases ayuda a entender mejor la base de datos y como las entidades (clases) están relacionadas e interactúan entre sí. Esto es útil para entender cómo se estructura la aplicación y como fluye la información a través de ella.

Nuestro diagrama de clases ha sido tremendamente útil para planificar la estructura del código antes de realizarlo, poniendo claras las prioridades y las bases de la aplicación. Además de actuar como forma de documentación visual, facilita a otros desarrolladores o personas que entiendan código, cómo está compuesta la aplicación, incluso a nosotros mismos, ya que, si en un futuro se decide cambiar la estructura de la aplicación, podría facilitar la refactorización de esta y ayudarnos a identificar que partes del sistema necesitamos cambiar y como enfocar dichos cambios.

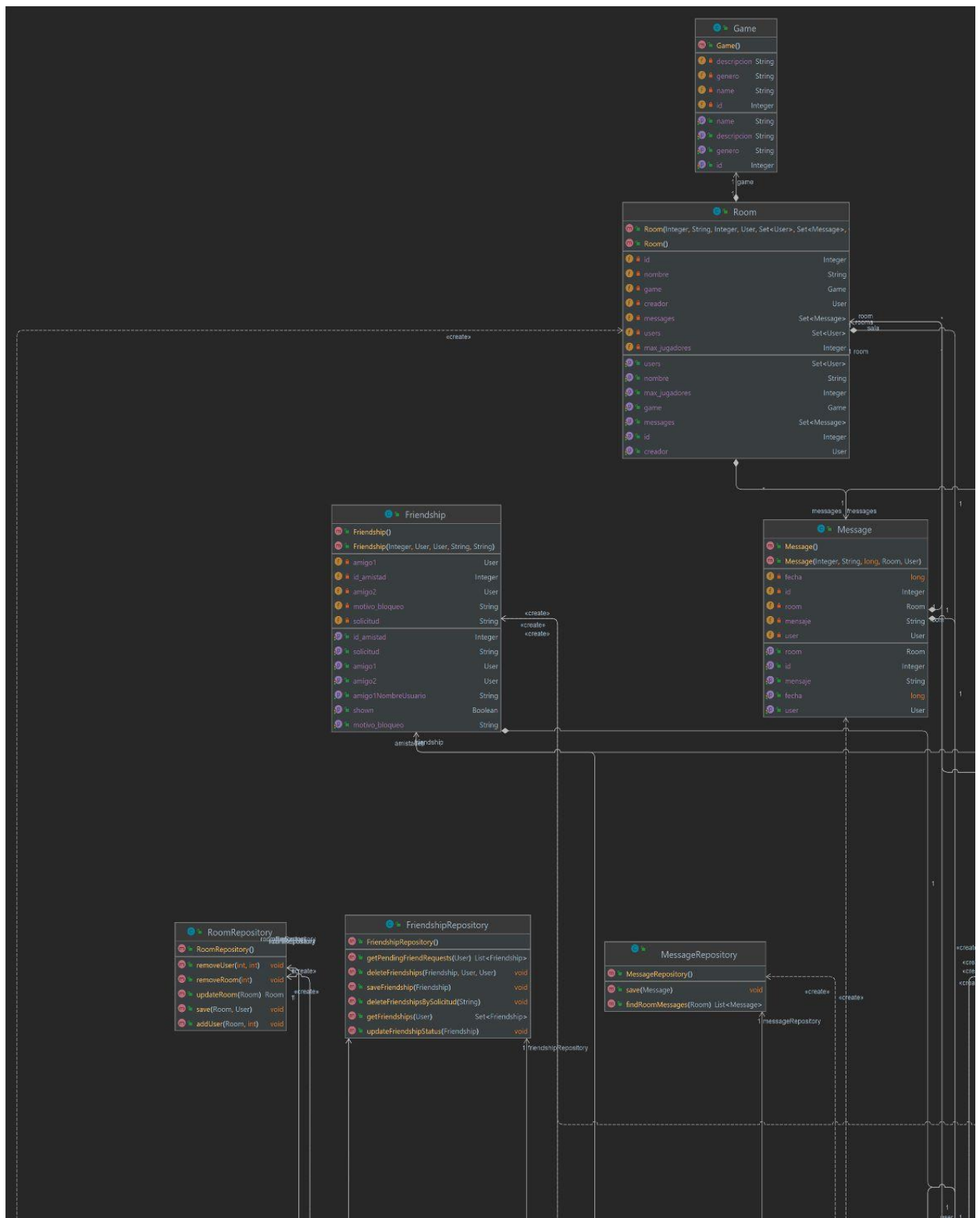


Figura 6.1. Diagrama de Clases UML (Parte 1).

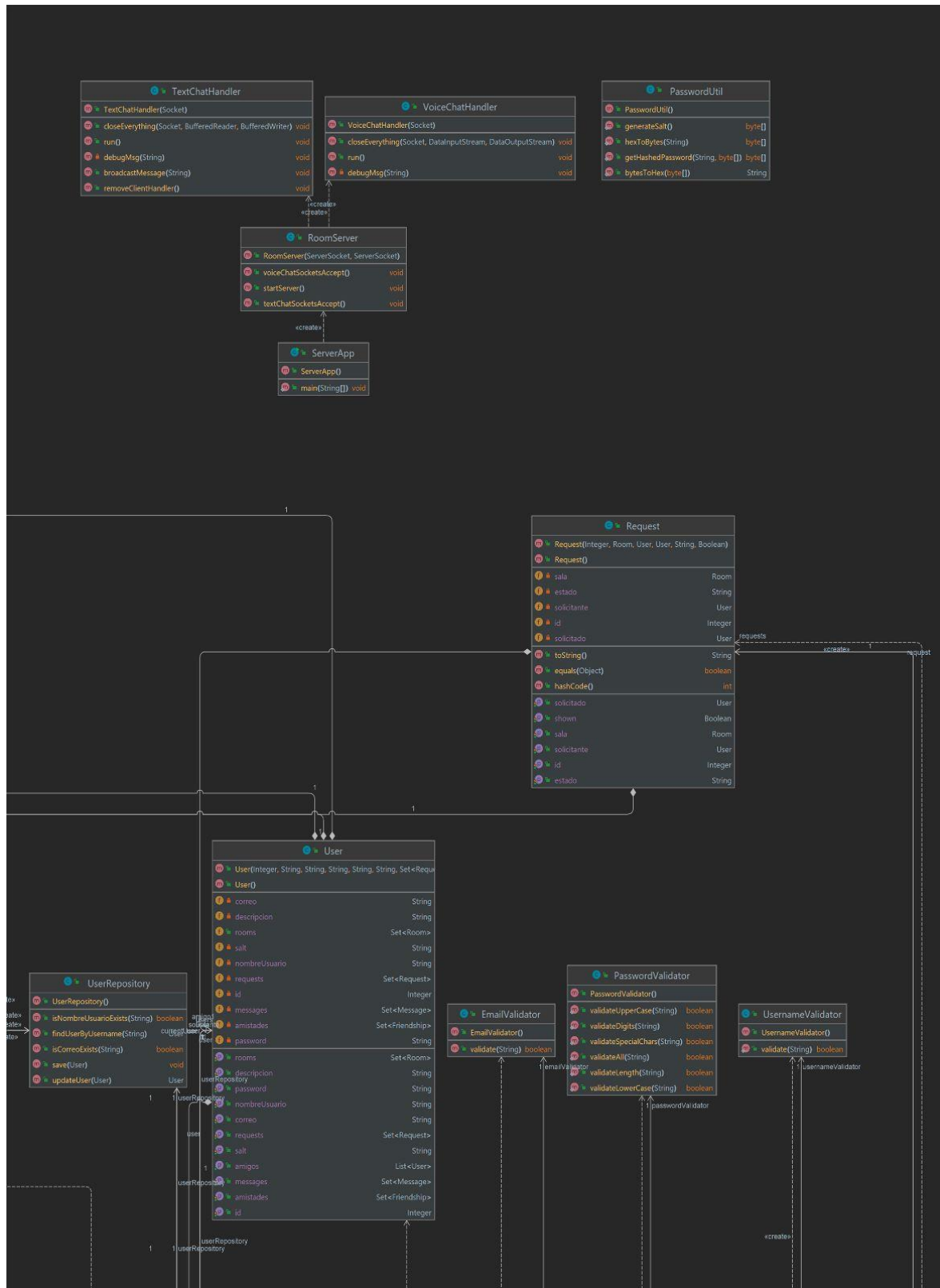


Figura 6.2. Diagrama de Clases UML (Parte 2).

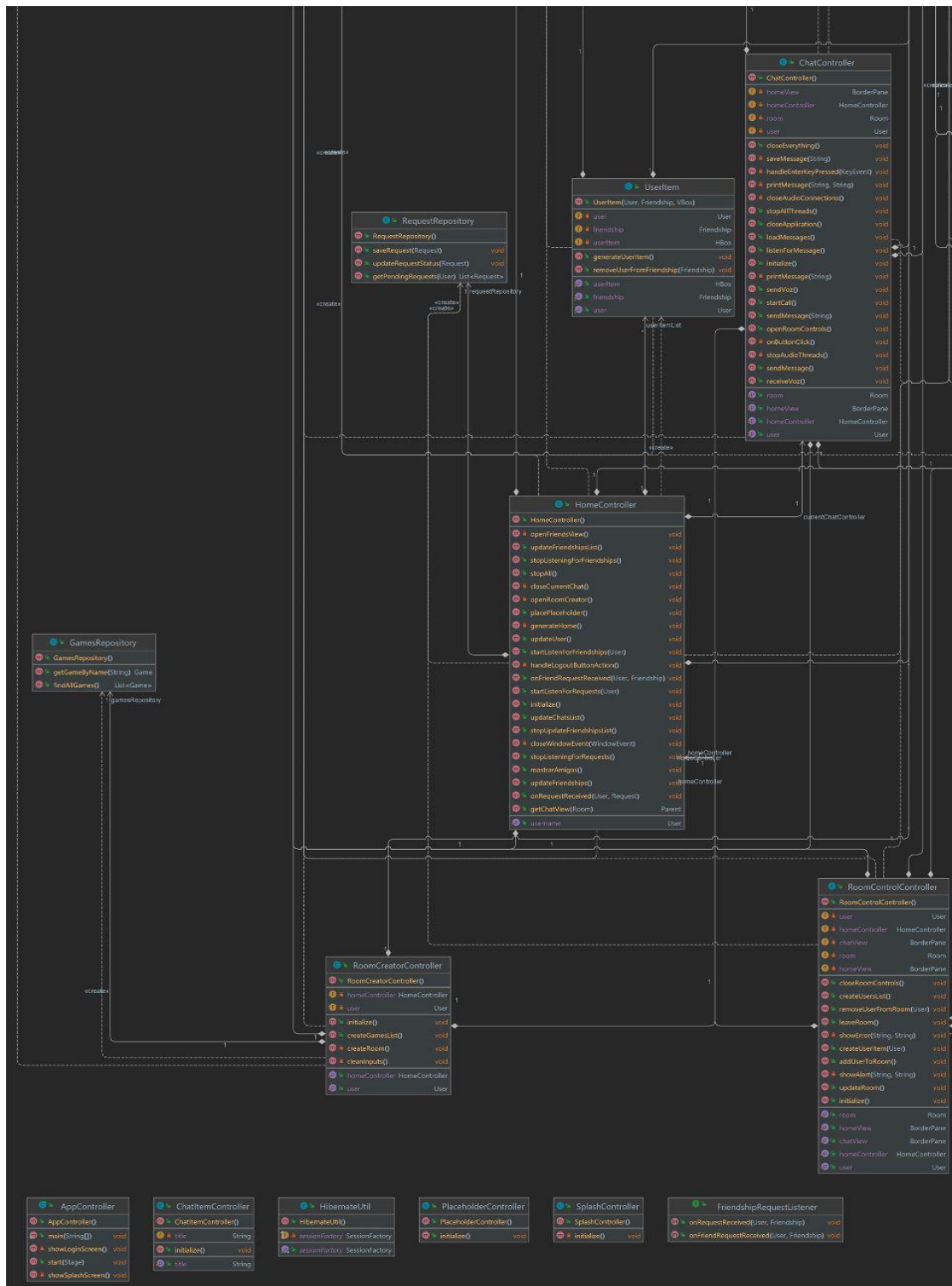


Figura 6.3. Diagrama de Clases UML (Parte 3).



Figura 6.4. Diagrama de Clases UML (Parte 4).

Diagrama de Flujo

Como podemos observar en el diagrama de flujo de navegación, tenemos que al iniciar la aplicación nos encontramos con dos decisiones (registrarse e iniciar sesión). Para registrarse el usuario tendrá que rellenar un formulario con datos y en el caso de que fuera valido se guardarían en la base de datos. Luego tenemos la decisión de iniciar sesión, en la cual de igual forma el usuario ingresara los datos para iniciar sesión y si fuera correcto mostraría la pantalla de home. de home.

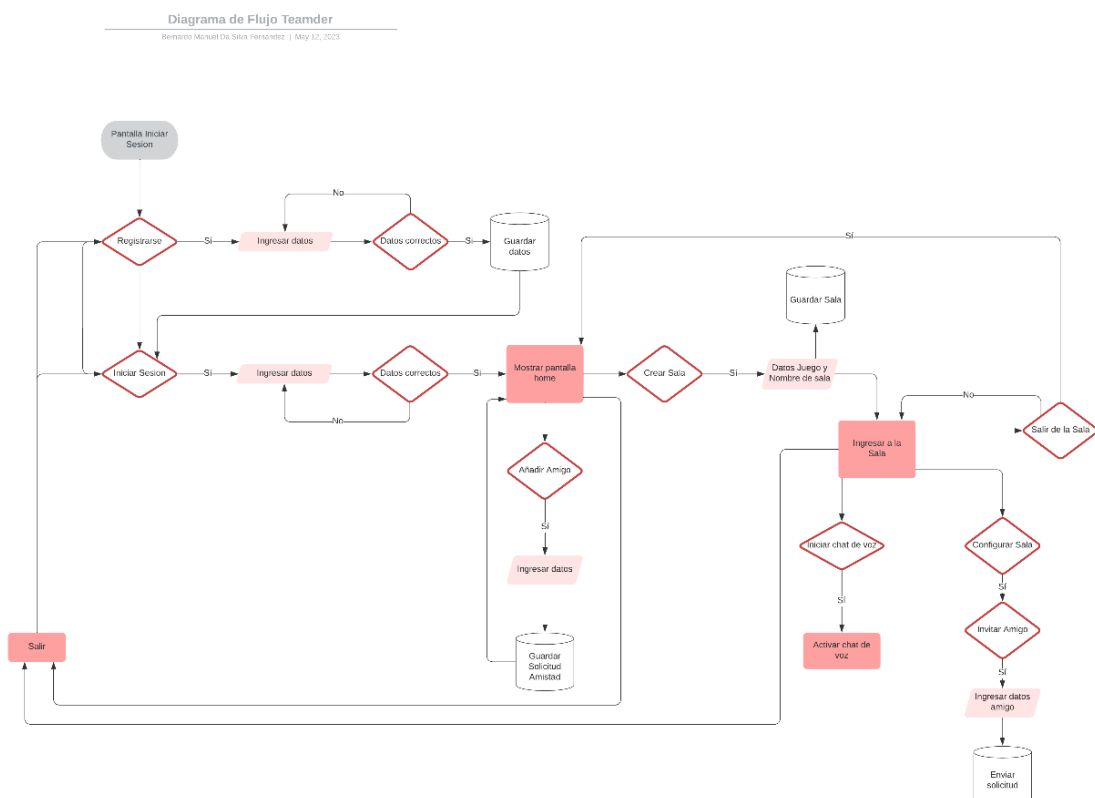


Figura 7. Diagrama de flujo de navegación

Ya dentro del proceso de la pantalla home, nos encontramos con dos decisiones: "Añadir amigo", en el que el usuario tendrá que introducir los datos del amigo a enviar la solicitud y luego esta misma se guardará en la base de datos. Y, "Crear Sala" en la que el usuario selecciona un juego y el nombre de la sala y así poder ingresar a la sala.

Dentro de la sala el usuario se encontrará con cuatro decisiones:

1. Iniciar chat de voz: el usuario podrá activar el chat de voz.
2. Configurar Sala: el usuario podrá decidir añadir un amigo a la sala, enviando una solicitud. Y dicha solicitud al mismo tiempo se almacena en la base de datos.
3. Añadir un amigo: el usuario será capaz de añadir a la sala a algún miembro de su lista de amigos para poder compartir la experiencia de jugar acompañados.
4. Salir de la Sala: El usuario podrá regresar a la pantalla del home nuevamente.

Metodología Empleada

Metodología Ágil:

1. **Iteraciones y entregas incrementales:** En lugar de entregar todo el software de una sola vez al final del proyecto, "Teamder" ha sido desarrollado y entregado en etapas o "sprints". Esto nos permitía a los

integrantes del equipo, obtener feedbacks del tutor en las reuniones pautadas y realizar los cambios pertinentes según las correcciones del nuestro tutor.

2. **Adaptabilidad:** “Teamder” ha sido diseñado con la capacidad de adaptarse y mejorar con el tiempo. Esto es un pilar clave de las metodologías ágiles, que valoran la capacidad de responder a los cambios y las necesidades del usuario.
3. **Colaboración:** “Teamder” se ha desarrollado haciendo un fuerte énfasis en la colaboración, tanto entre los integrantes del equipo como con el tutor. En las metodologías ágiles, la colaboración y la comunicación son fundamentales.
4. **Enfoque en el usuario:** Los desarrolladores de “Teamder” hemos puesto un fuerte énfasis en la experiencia del usuario y la usabilidad. Esto es consistente con la metodología ágil, que valora el software funcional y la satisfacción del cliente.

Presupuesto

Aunque el desarrollo de Teamder no requirió una inversión monetaria en términos de compra de software, hardware o servicios de bases de datos, hubo costos no monetarios significativos asociados con el proyecto. Estos incluyen:

1. **Tiempo de desarrollo:** Los desarrolladores de Teamder hemos invertido una cantidad considerable de tiempo en el diseño, codificación, prueba y

depuración de la aplicación. Si bien este tiempo no tiene un costo monetario directo, representa un costo de oportunidad.

2. **Formación y aprendizaje:** Los integrantes del equipo hemos invertido tiempo en retomar y dominar las tecnologías y herramientas utilizadas en el desarrollo de Teamder, como JavaFX y MySQL. Aunque esto no representa un costo monetario directo, es una inversión significativa en términos de tiempo y esfuerzo.
3. **Infraestructura:** Aunque los tres integrantes del equipo utilizamos herramientas y tecnologías de código abierto y gratuitas, el mantenimiento y la operación de estas herramientas implican costos indirectos. El tiempo y los esfuerzos dedicados al mantenimiento y la administración de estas herramientas son un coste para el proyecto.

En resumen, Teamder no ha requerido una inversión monetaria directa, el proyecto ha incurrido en costos significativos en términos de tiempo y esfuerzo, pero estamos orgullosos de fomentar que el desarrollo de una aplicación no requiere gastos monetarios para funcionar correctamente y dar la talla.

Diseño

Prototipo

En la siguiente figura podemos observar el prototipo de la primera interfaz gráfica para el usuario y las primeras impresiones. Ha habido algunos cambios visuales pero la lógica y las funcionalidades siguen siendo las mismas.

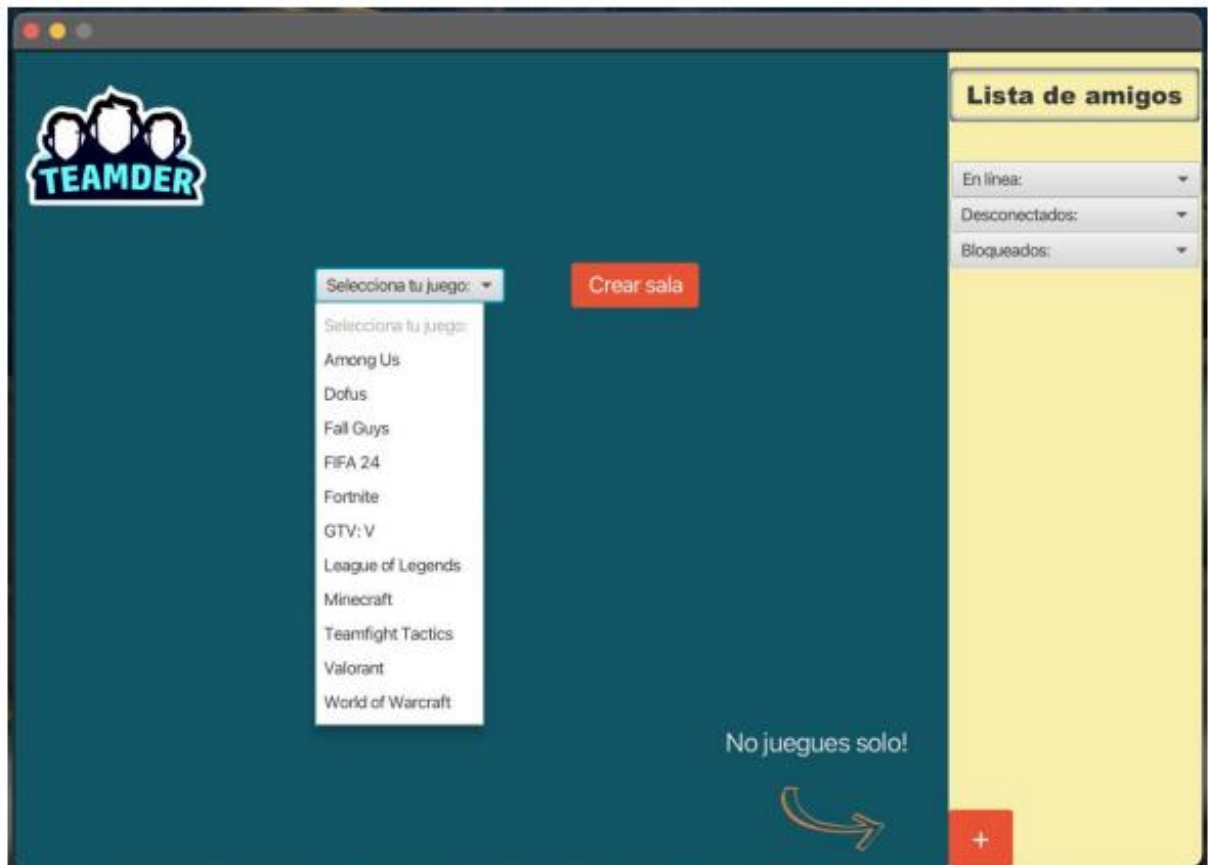


Figura 8. Primeras Impresiones del "home".



Figura 9. Prototipo de logos, posibles opciones: Captura obtenida de la página web de DALL-E (OpenAI, 2021).

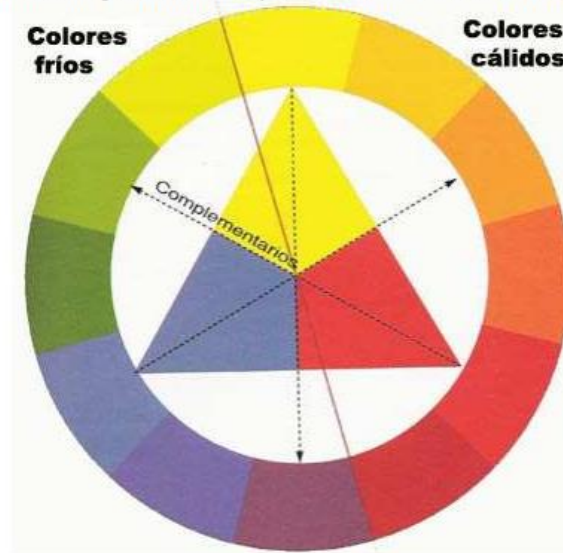
Para el diseño de la interfaz de usuario hemos querido hacerlo lo más sencillo e intuitivo posible para que sea accesible a todo el público incluyendo edad, procedencia, lenguaje, etc.

Utilizamos en general una gama de solo 3 colores, ya que una aplicación debe ser visualmente agradable y no queríamos sobrecargar las vistas. Es una paleta de colores basada en azul verdoso oscuro, naranja y amarillo pálido con una tonalidad tierra.

Hemos realizado una investigación basada en la teoría cromática del color por lo que estos colores nos han parecido los más adecuados por distintas razones:

- 1. En la rueda de colores, el azul verdoso oscuro y el naranja son colores complementarios, por lo que tienen un alto contraste entre sí y atraen la atención del usuario cuando se utilizan juntos.
- 2. El color amarillo pálido tierra, ayuda a suavizar el fuerte contraste de los dos colores mencionados anteriormente por lo que aporta una armonía visual, aunque sean colores tan distintos. En otras palabras, sirve como un equilibrador para que no se haga tan pesado al usuario estar varias horas utilizando la aplicación.
- 3. Según investigaciones realizadas, cada color tiene asociaciones culturales y emocionales diferentes. El azul verdoso es sinónimo de confianza y profesionalidad, mientras que el naranja se asocia con la creatividad y energía, justo lo que queremos fomentar en la comunidad de "Teamder". Pero a su vez el color amarillo es sinónimo de calidez y comodidad, por lo que creemos que esta paleta de colores hace el equilibrio visual perfecto.

Los significados tanto de carácter psicológico como de orden simbólico que tienen, o se le ha dado a los distintos colores, se verá a continuación. No obstante, las características principales que las gamas fría y cálida tienen, hacen que los que correspondan a la primera se relacionen con la tranquilidad, la calma, la pasividad, y, en general, con conceptos de moderación y poco vitales, así como con el invierno y el frío. Sin embargo, en los calientes se aprecian los efectos contrarios: vivacidad, movimiento, alegría, excitación, energía, entusiasmo, y se les relaciona con el verano y el calor.



Significados, códigos y uso cultural de los colores

El lenguaje del color es conocido por artistas y profesionales de campos donde su utilización es fundamental en la elaboración de sus productos, como el diseño, la decoración, la publicidad, etc... Es un medio atractivo que, consciente o subconscientemente actúa sobre el observador, por lo cual sus características son explotadas por los mismos en sus trabajos.

Diferentes códigos visuales se basan precisamente en la adjudicación de un determinado significado de cada color; así, en el tráfico el rojo simboliza peligro o prohibición; el amarillo, precaución, y el verde es el utilizado para permitir la circulación (lo vemos en los semáforos y también en las señales).

En la industria, los distintos colores simbolizan bien aparatos, conducciones o piezas de maquinarias según a lo que corresponda cada cual. En topografía, los distintos signos que se utilizan en planos y mapas han de estar coloreados también según un código establecido que impida la confusión. Los arquitectos y ingenieros igualmente identifican los trazados de esquemas eléctricos, de fontanería o gas, según determinados colores cuya lectura ha de ser conocida por quienes han de realizarlos.

Figura 10. Rueda del color.

Extraído de: ESCOLA D'ART I SUPERIOR DE DISSENY DE VIC. (2020).
Psicología del color.

Logotipo y Diseño de Marca

El logotipo de "Teamder" fue creado con la ayuda de la IA "DALL-E" la cual pertenece al grupo "OpenAI" y se encarga de generar imágenes a partir de descripciones textuales. Nos ha sido de gran ayuda ya que es un logo libre de derechos de autor, creado al momento y muy original. Ahorra el tiempo de diseñar un logo por cuenta propia y dinero a la hora de pagar a alguien para que se encargue del diseño. Simplemente se le indica la descripción de cómo se quiere la imagen, y la diseña en cuestión de segundos siguiendo los parámetros que se le han indicado al pie de la letra.

Luego de generar el logo, pasamos a darle una serie de modificaciones utilizando Photoshop. Como fue la separación del fondo para utilizar únicamente el logo. Y finalmente, colocar el texto "TEAMDER" en el centro del logo.

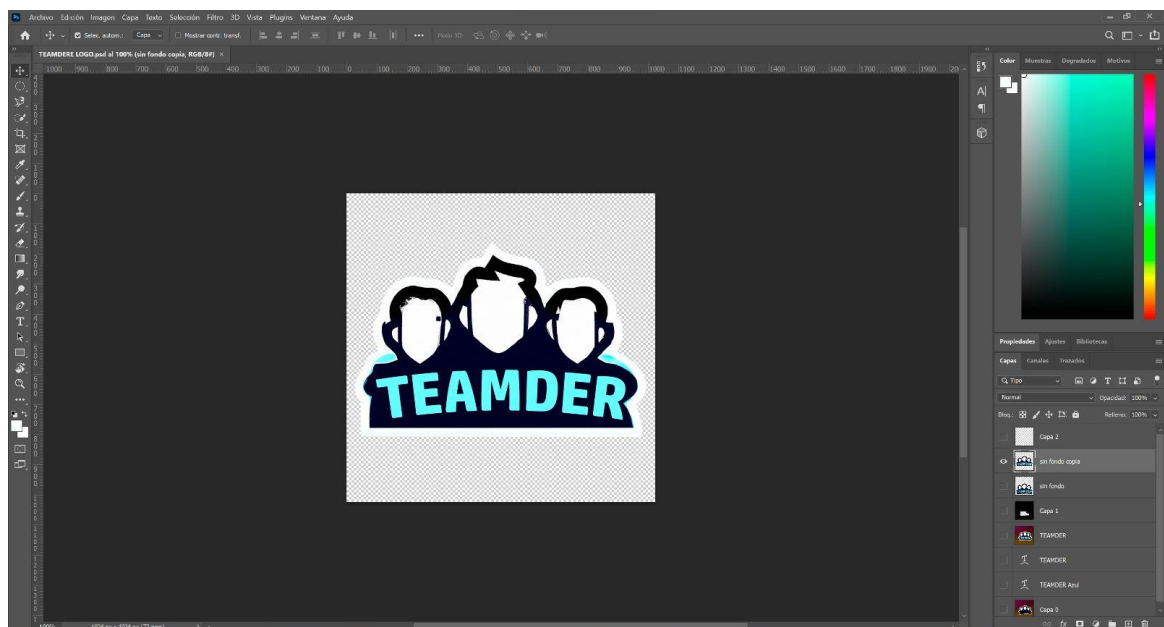


Figura 11. Correcciones en Photoshop.

Interfaces Finales

El diseño del inicio de sesión fue una de las pocas vistas o “prototipos” que llegaron para quedarse en nuestra aplicación desde el principio, fue la primera UI creada y mantuvo su diseño desde el inicio del proyecto, ya que consideramos que cumple con los principios de usabilidad, es intuitivo, de fácil entendimiento y es agradable a la vista.

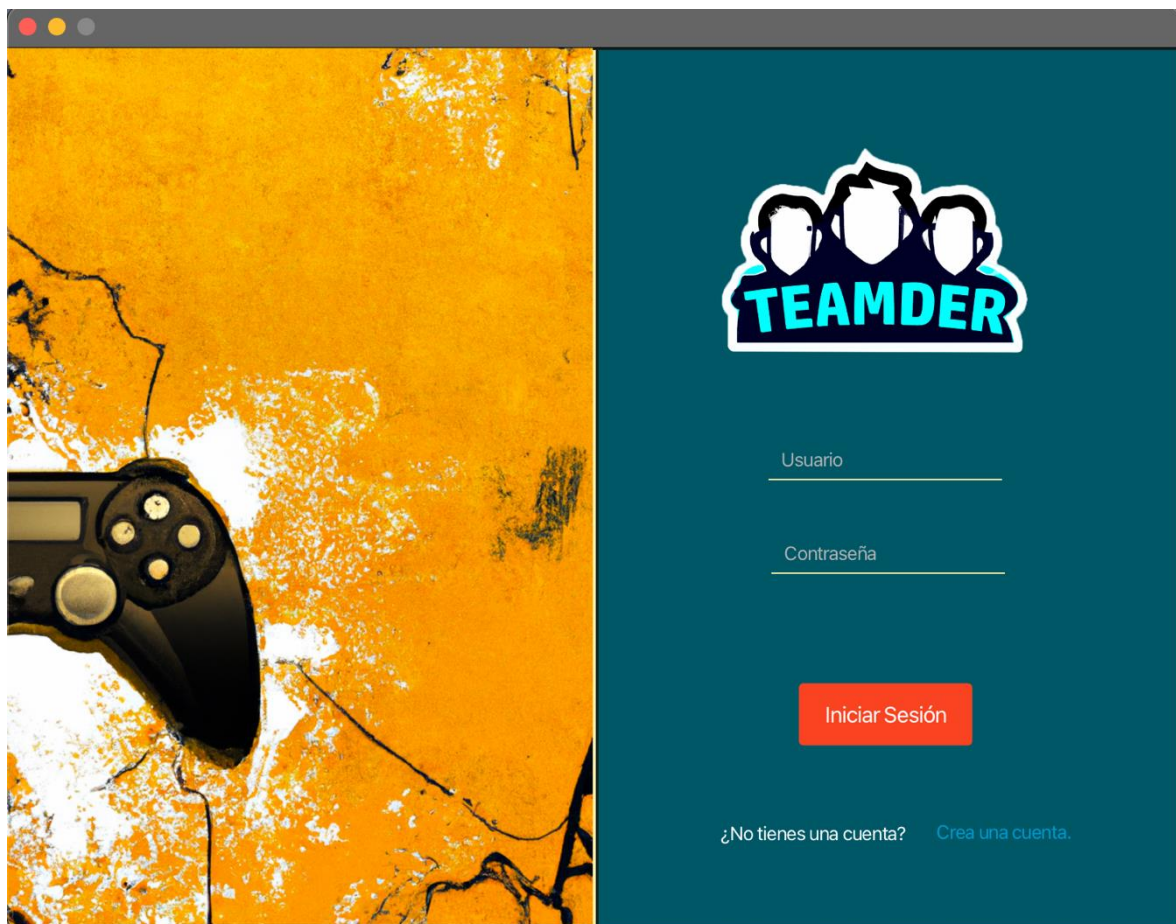


Figura 12. Interfaz Final de Inicio de Sesión.

La vista del Registro del usuario, junto con el inicio de sesión, quedó implantada desde el principio debido a su fácil usabilidad y comprensión, cada uno de los campos que deben ser rellenado para que el usuario cree una cuenta

en nuestra aplicación, están validados y tienen una serie de requerimientos y exigencias para que el usuario goce de la mayor seguridad posible a la hora de conectarse, como por ejemplo el ingreso de caracteres especiales, mayúsculas, minúsculas y números para la contraseña.

Figura 13. Interfaz Final de Registro.

La figura representada a continuación, es la interfaz final del "HOME" de "Teamder", donde se podrá añadir y coleccionar las amistades, crear nuevas salas para jugar con amigos, cerrar sesión para dar paso a otra cuenta o ingresar a las salas que ya se tiene creadas, con nuevas funcionalidades para que la experiencia de juego sea de otro nivel.

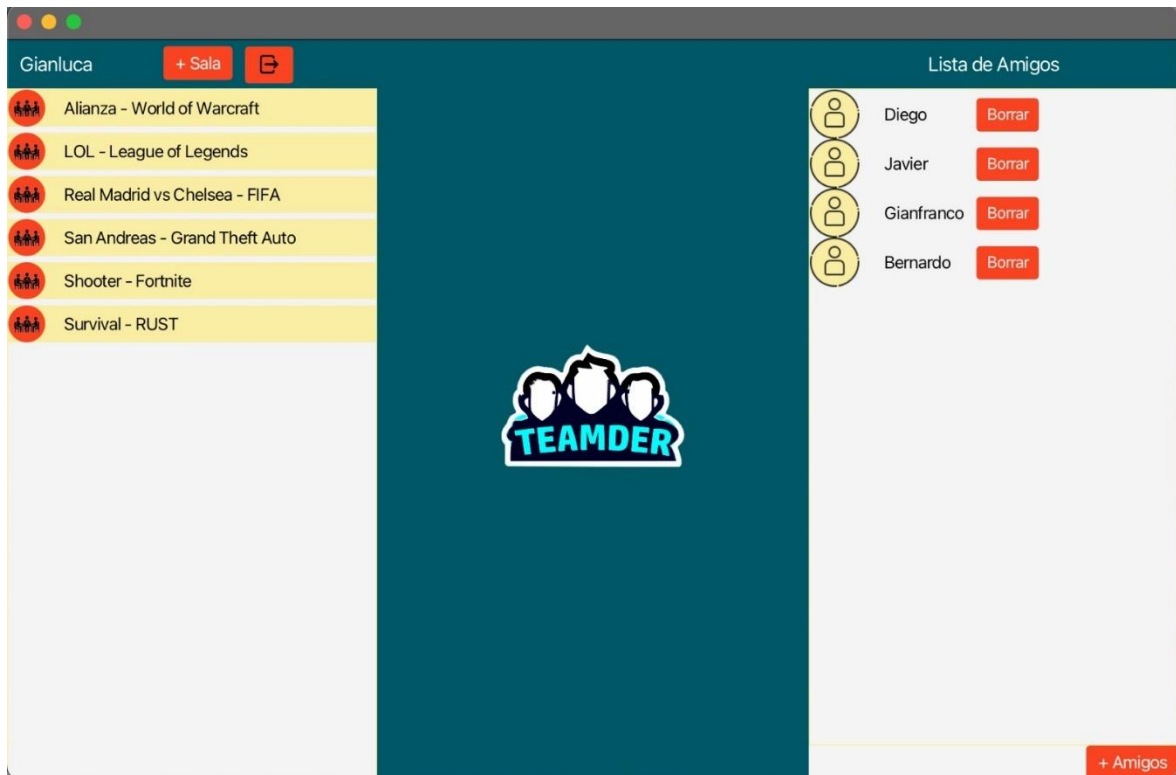


Figura 14. Producto final vista "HOME".

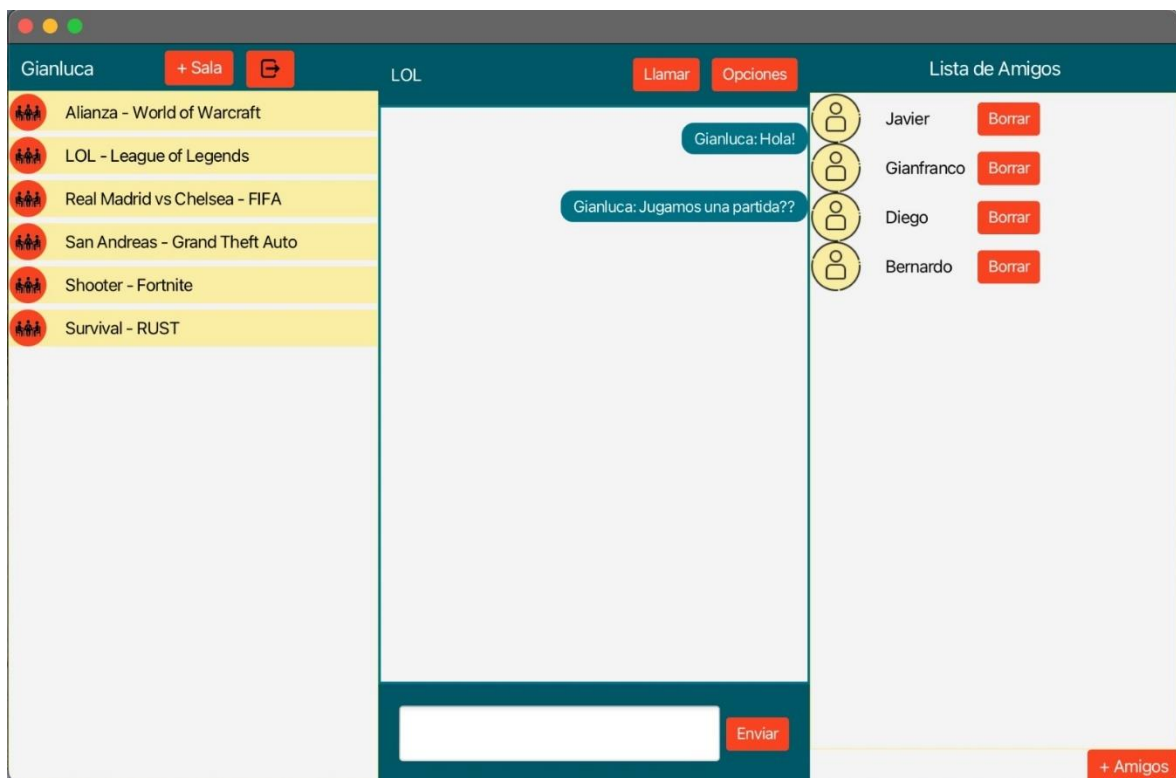


Figura 15. Interfaz Final de las Salas.

Invitar a tus amigos a las salas ahora es fácil y rápido, basta con escribir su nombre, pulsar el botón de añadir y se le enviará una solicitud a la persona que deberá Aceptar o Rechazar. Si la persona acepta, se le creará automáticamente en su "home", la sala en cuestión con el respectivo nombre que ya tiene, se podrá entrar y gozar de todas las funcionalidades que esta trae, como chatear, llamar y opciones como Llamar a los integrantes que yacen en ella o abandonar la sala para que se elimine de la lista del "home".

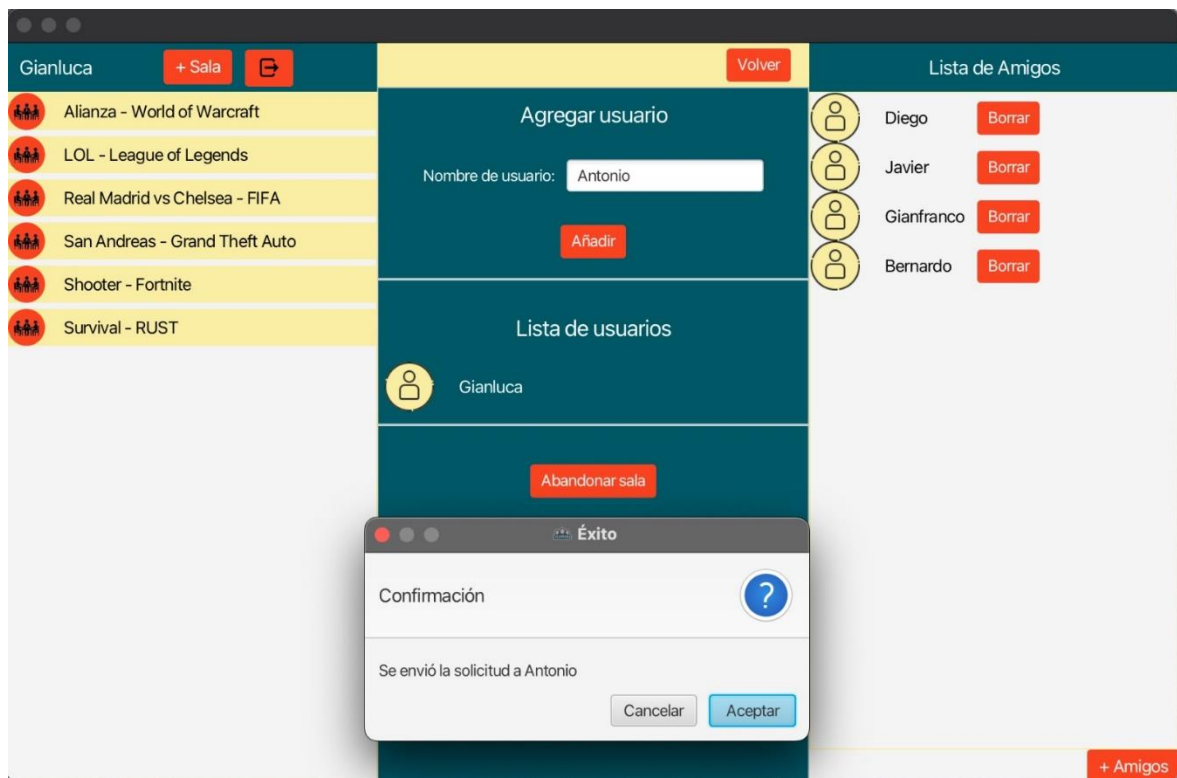


Figura 16. Interfaz Final de las Opciones de Sala.

Conclusiones

Proceso de trabajo:

Lo que más ayudó en el desarrollo de este proyecto fue la gestión eficiente del mismo a través de la metodología Agile. Porque proporcionó una mayor flexibilidad y adaptación a los cambios que surgieron en este durante su desarrollo. También el uso de herramientas como Github para el control de versiones, IntelliJ IDEA como entorno de desarrollo principal y Trello para la gestión y división de tareas correspondientes al desarrollo del proyecto. Además del empleo de tecnologías acertadas como JavaFX, MySQL y protocolos TCP. De esta forma el equipo de desarrollo de "Teamder" cumplió con los plazos establecidos, mantuvo la comunicación y organización en todo momento y superaron los desafíos técnicos.

Ampliaciones y posibles mejoras:

Lo más difícil en el desarrollo de este proyecto fue el limitado tiempo disponible. Incluso logrando los objetivos planteados, existen ampliaciones y posibles mejoras que podrían implementarse en un futuro. Como pueden ser, la integración con cuentas de Google, para permitir al usuario registrarse e iniciar sesión utilizando una cuenta de Google. También, la integración a redes sociales para facilitar la búsqueda de jugadores. Estas mejoras podrían aumentar el valor de la aplicación y brindar una mejor experiencia al usuario final.

Reflexión final:

En este proyecto se desarrolló una aplicación de escritorio dirigida principalmente a jugadores, que permite a los usuarios registrados conectar entre sí y mantener conversaciones de voz y de texto a tiempo real a través de salas de chat, así como disfrutar de una experiencia de juego en grupos con sus amigos, mediante sistema de chat de voz y texto estable y seguro, implementando un sistema de solicitudes de amistad, proporcionando una

interfaz gráfica de usuario atractiva e intuitiva y un sistema de registro de usuarios seguro y confiable. Lo más importante del desarrollo de este proyecto fue lograr que el usuario final tenga una experiencia de comunicación en línea fluida, segura y satisfactoria, promoviendo la interacción y colaboración entre jugadores.

Webgrafía y Bibliografía.

"Gupta, S. K. (2020). MySQL for Beginners: A Step-by-Step Guide to Learn MySQL. BPB Publications."

"Oracle Corporation. (2019). What is JavaFX? Recuperado el 19 de marzo de 2023, de <https://www.oracle.com/java/technologies/javafx.html>. "

"Escola d'Art i Superior de Disseny de Vic. (2020). Psicología del color. Campus Vic de l'ESDAP Catalunya, Rambla de Sant Domènec, 24 , 08500 Vic. Recuperado el 16 de mayo de 2023, de <https://perio.unlp.edu.ar/catedras/iddi/wp-content/uploads/sites/125/2020/04/Psicologia-del-color.pdf>."

"OpenAI. (2021). DALL-E: Creating Images from Text. <https://openai.com/research/dall-e>."