

**IES AUGUSTO GONZÁLEZ DE LINARES**

**DEPARTAMENTO DE INFORMÁTICA**

**PROYECTO VAX**

PROYECTO DICIEMBRE 2023 2º CURSO

GRADO SUPERIOR DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA

2022/2023

Barrios Fernández, María Carmen

Díez de Paulino, Albano

Espinosa García, Daniel

Gutiérrez Valverde, Ramiro

Índice

1. **ANÁLISIS** …………………………………………………………………………………………………………………………………………2

[1.1.Requisitos del proyecto 2](#_Toc134376937)

[1.2.Planteamiento de la solución 3](#_Toc134376938)

[**2. DISEÑO** 6](#_Toc134376939)

[2.1.Diseño lógico de la base de datos 6](#_Toc134376940)

[2.2.Diseño de la interfaz gráfica de java 8](#_Toc134376941)

[2.3.Diseño web 10](#_Toc134376942)

[2.4.Odoo 11](#_Toc134376943)

[**3. DESARROLLO** 15](#_Toc134376944)

[3.1.Hardware 15](#_Toc134376945)

[3.2.Software 16](#_Toc134376946)

[3.2.1.Software Empleado 16](#_Toc134376947)

[3.2.2.Software Alternativo 17](#_Toc134376948)

[3.3.Diseño físico de la base de datos 18](#_Toc134376949)

[3.4.Código java 19](#_Toc134376950)

[3.5.Código HTML+ CSS+ RSS 24](#_Toc134376951)

[**4. PRUEBAS** 27](#_Toc134376952)

[4.1.Pruebas unitarias sobre código java (JUnit5) 27](#_Toc134376953)

[**5. DESPLIEGUE** 29](#_Toc134376954)

[5.1.Despliegue Web 29](#_Toc134376955)

[5.2.Despliegue aplicación 29](#_Toc134376956)

[**6. PLANTEAMIENTO FUTURO** 29](#_Toc134376957)

[6.1.Ideas para continuar el desarrollo 29](#_Toc134376958)

[**7. Control de Versiones del Proyecto** 30](#_Toc134376959)

[7.1.Git 30](#_Toc134376960)

[7.2.Opinión de los integrantes del grupo 31](#_Toc134376961)

[**8. Índice Ilustraciones** 32](#_Toc134376962)

[**9. Bibliografía** 33](#_Toc134376963)

[**10. Anexos** 34](#_Toc134376964)

[10.1.Anexo I Sentencias SQL usadas en el proyecto 34](#_Toc134376965)

[10.2.Anexo II Pruebas sobre diferentes Sistemas Informáticos 49](#_Toc134376976)

[10.3.Anexo III Documentación FOL 79](#_Toc134376992)

1. **ANÁLISIS**

# Requisitos del proyecto

Empresa simulada de distribución de videojuegos Indie.

Hay una licitación del gobierno de Cantabria que plantea, en concurrencia competitiva, ayudar con un importe monetario de 30.000€ a la *start-up* tecnológica especializada en el desarrollo de aplicaciones que alcance una mayor puntuación en la realización de un proyecto viable de distribución de videojuegos.

El objetivo es desarrollar una plataforma para videojuegos que permita iniciar sesión, cargar y descargar juegos, así como gestionar los juegos de cada usuario. También será necesario desarrollar un videojuego inicial para dicha plataforma. El equipo planteará su actividad de empresa a través de un sistema de gestión empresarial.

Las bases para el desarrollo del reto son:

* Cada empresa deberá elegir un nombre y un logo que se utilizará durante su actividad corporativa.
* Cada miembro de la empresa deberá de firmar un contrato/declaración responsable de obligado cumplimiento.
* Se creará una plataforma que constará de un cliente de escritorio y un servidor.

El **cliente** se ejecutará en un ordenador de la clase y tendrá una interfaz gráfica que permita:

* Registrar nuevos usuarios a la plataforma: la información de los usuarios se almacenará en una base de datos remota instalada en un servidor.
* Iniciar sesión de un usuario en la plataforma. Un usuario que inicia sesión puede realizar diferentes acciones en función de su rol (jugador, desarrollador o invitado):
  + Visualizar los juegos que hay en el servidor.
  + Visualizar los juegos que tiene en su biblioteca.
  + Descargar un juego almacenado en el servidor en una carpeta local y añadirlo a su biblioteca.
  + Lanzar un juego descargado en una carpeta local.
  + Además, si es un usuario desarrollador:
    - Añadir un nuevo juego al servidor: se indicará información como el nombre, una imagen representativa, desarrollador, género, etc.
    - Subir al servidor un juego.
* Otras funcionalidades que se pueden añadir:
  + Registro de partidas.
  + Registro de puntos.
  + Notificaciones a los usuarios.
  + Etc.

Se implementarán al menos dos **servidores** instalados en una o varias máquinas virtuales (a elección del alumnado):

* Un **servidor** de Base de Datos con toda la información relacionada con los usuarios y juegos.
* Un **servidor** de descarga y subida de videojuegos implementado en Java utilizando

*sockets*. Este servidor administrará una o varias carpetas que contendrán:

* + Un juego o varios juegos de ejemplo en un formato *zip*. El zip contendrá el

ejecutable (.exe) y todos los archivos necesarios para su ejecución.

* + Las imágenes de las carátulas de los videojuegos.

También se tendrá en cuenta:

* Se creará un juego de ejemplo para la plataforma utilizando el motor **Unity**.
* Los servidores permitirán la conexión y descarga de juegos al mismo tiempo por varios usuarios.
* Se estudiará la viabilidad de uso de un programa ERP para gestionar la empresa junto con los módulos necesarios para ello.

Herramientas software a utilizar:

* GitLab
* NetBeans
* Unity
* PostgreSQL
* VirtualBox o VMware
* Odoo ERP

# Planteamiento de la solución

Ante el problema que se nos plantea hemos tomado la solución de desarrollar una aplicación sobre lenguaje “Java”, con las librerías “JavaFX” de la empresa “Gluon”, relacionado con el sistema gestor de bases de datos relacionadas “MySQL”, ya que nos es más familiar este sistema gestor que PostreSQL, si en un futuro el cliente exige PostreSQL se puede realizar una migración sin problemas.

Todos los servicios necesarios serán montados sobre un servidor de AWS o Google Cloud en la versión final, pero para el desarrollo usaremos un maquina virtual con S.O Xubuntu sobre VirtualBox.

Dicho servidor contendrá tanto los servidores necesarios para la comunicación de la aplicación del cliente(Imágenes y Archivos), la base de datos y el SGE de nuestra empresa(Odoo).

Para ello el servidor debe de contar con los siguiente requisitos:

* JRE Y JVM 17
* MySQL Server 8.2.0
* Servidor Apache 2.0
* Servidor Odoo Community 16

Las funciones que consideramos núcleo de la App son:

* Interfaz gráfica que permita visitar la tienda de juegos sin iniciar sesión(Usuario Invitado).
* Interfaz gráfica que permita registrar nuevos usuarios(Jugadores y Desarrolladores) y nuevos juegos(Solo para Desarrolladores).
* Interfaz grafica que permita ver que juegos tienes comprados si inicias sesión con tu usuario.
* Servidor de Bases de Datos que almacene toda la información para hacer persistente la aplicación.
* Servidor con un SGE para la empresa.
* Interfaz Responsive.

Las funciones que consideramos principales son:

* Servidor-Cliente para la descarga de juegos.
* Una forma de ejecutar el juego que te descargas.
* Cargar Imágenes y textos para el usuario(Avatar, Alias, etc.) y los juegos (Titulo, regulación, precio ,etc.)

Las funciones que consideramos secundarias son:

* Interfaz con ayuda para el usuario.
* Sistema de recuperación de contraseñas.
* Logotipo de empresa(Café Con Palito).
* TitleBar Personalizada.
* Filtros en tienda y biblioteca.

Las funciones que consideramos realizar en un futuro desarrollo:

* Añadir más idiomas a la aplicación.
* Ayudas Visuales y Auditivas (TTS, Cambios de colores para daltónicos, Aumento del tamaño de letra).
* Ajustes Personalizados por usuario.
* Sistema de Categorías para la interfaz de la biblioteca.
* Añadir un botón que envié a la página web de la empresa.
* 100% de compatibilidad con otros S.O que no sean Windows
* Crear una aplicación móvil.

Al solo disponer de 15 semanas de trabajo la organización del proyecto es la siguiente:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Tabla 1 - Diagrama de Gantt

# DISEÑO

# Diseño lógico de la base de datos

El diseño lógico de la base de datos del proyecto no es igual que el proporcionado, ya que consideramos al inicio del proyecto que muchas de las tablas del diseño no eran necesarias ,así que pasamos a un diseño de solo 3 tablas pero lo hemos ido cambiando desde la idea original de 3 tablas (usuario, juego, biblioteca) que hicimos al inicio del reto, ya que al añadir más funcionalidades a la aplicación de java nos encontramos con la necesidad de almacenar nuevos datos que no estaban previstos al inicio, así que el diagrama relacional final es el siguiente:

Diagrama

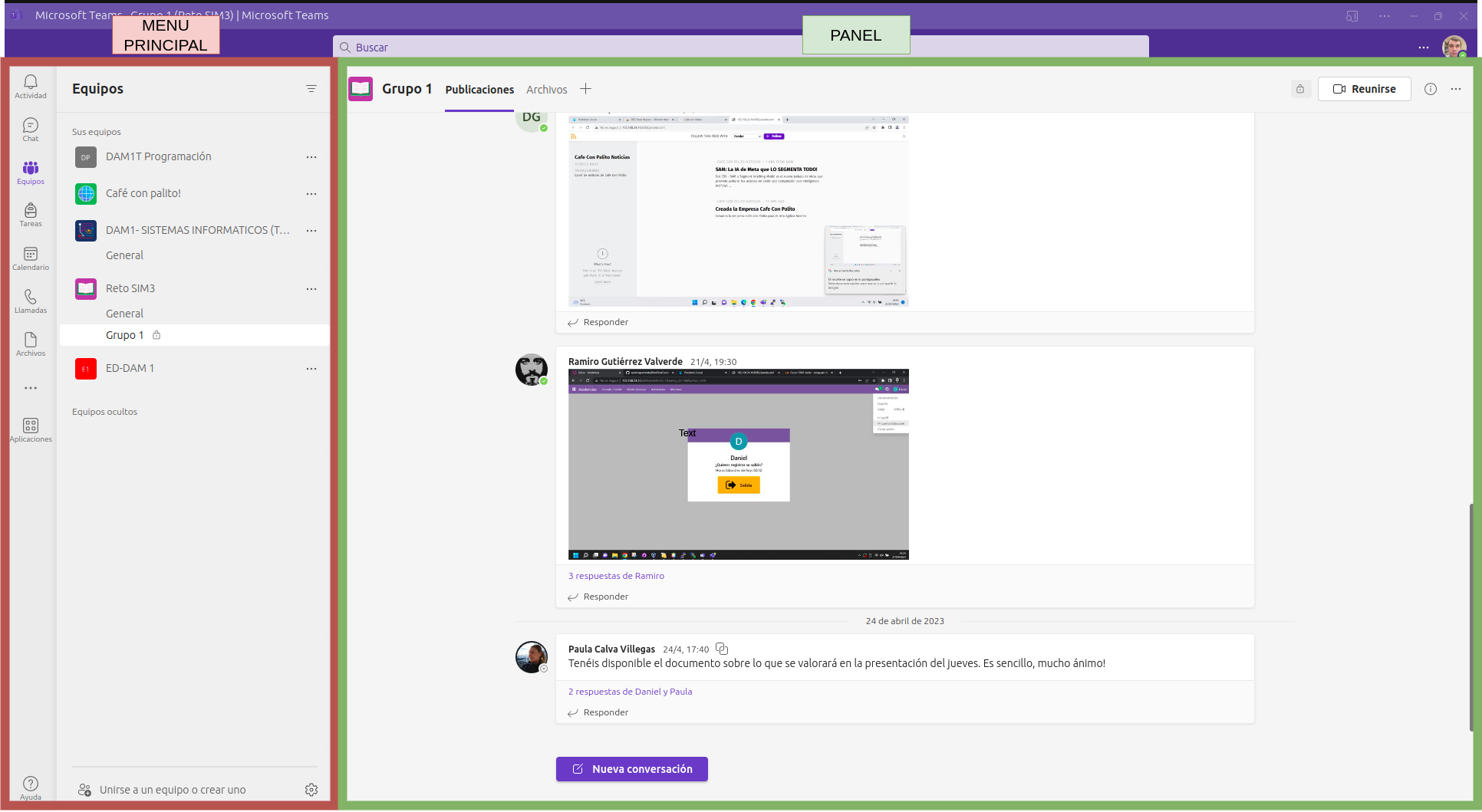
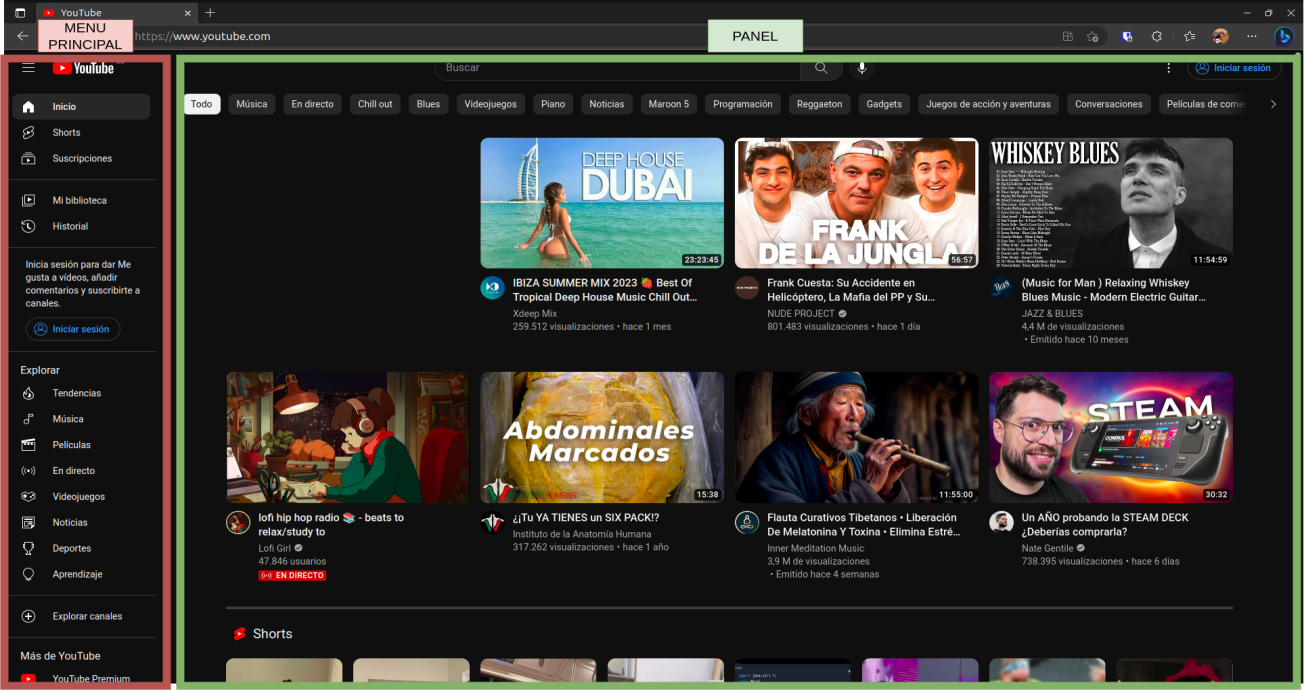
Descripción generada automáticamente

Diagrama 1 – Modelo Relacional

# Diseño de la interfaz gráfica de java

El diseño inicial que planteamos para la interfaz de java está basado en las aplicaciones o aplicaciones web modernas como por ejemplo Teams.

Dichas aplicaciones mantienen un estilo minimista, con una barra vertical a la izquierda que realiza la función de menú principal que nos permite interactuar con todas las funciones de la aplicación y un panel que abarca el resto de la pantalla con las funciones que llamas desde el menú de la izquierda.



Web 1 - Pantalla Principal Teams

Web 2 - Pantalla Principal Youtube

Con las referencias que buscamos para el diseño, nos quedamos con la siguiente base para la fase 3 (Desarrollo).

# Mockup

Para llegar a un diseño de interfaz atractivo hicimos varios bocetos iniciales con las premisas anteriores, pero el definitivo fue el siguiente:



Diagrama 2 – Mockup

# Wireframe

Tras tener un mockep, procedimos a realizar un Wireframe para tener más claro el diseño de la interfaz.

Interfaz de usuario gráfica

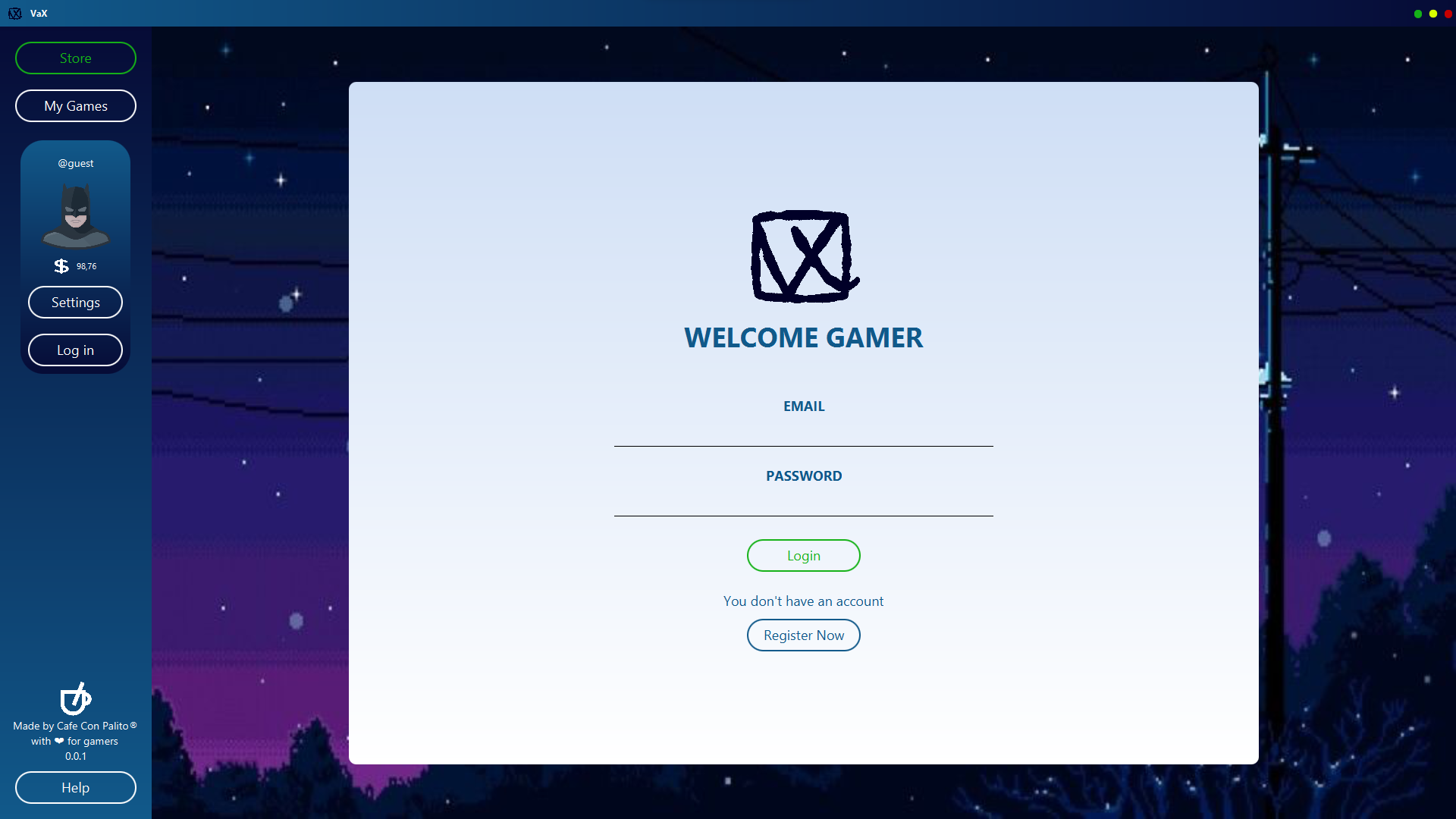
Descripción generada automáticamente

Diagrama 3 – Modelo UI 2

Este modelo fue presentado a los desarrolladores frontend de Lis Data Solutions S.L y nos plantearon las siguientes sugerencias para mejorar el diseño:

* Crear una titlebar personalizada.
* Eliminar el menú de navegación de la parte superior.
* Usar degradados desde azul oscuro a azul claro para mejorar visualmente los diferentes paneles.
* Si un panel se encuentra sobre otro invertir el degradado.
* Usar el color verde para los botones importantes que se encuentren sobre fondo oscuro.
* Usar el color blanco para los botones importantes que se encuentren sobre fondo oscuro.
* Usar el azul claro para los botones que se encuentren sobre fondo blanco.
* Eliminar el footer.

Partiendo desde nuestro diseño y aplicando las sugerencias, el diseño final de la aplicación es el siguiente.



Software 1 – UI Proyecto Vax

# Estructura de Servicios

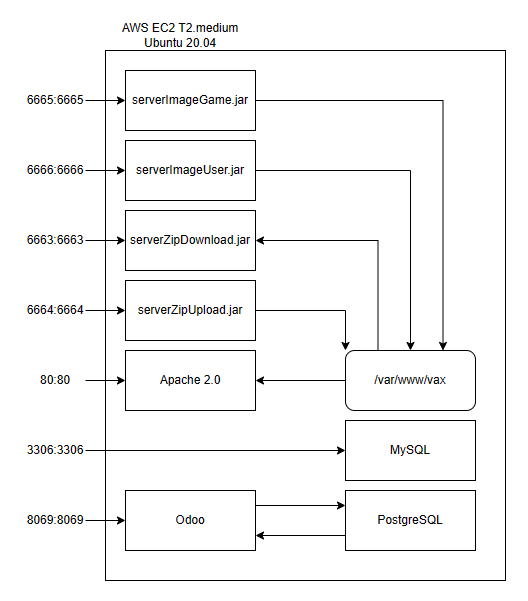
Para la estructura del proyecto nos hemos decantado por la arquitectura de microservicios. Ya que las ventajas que aporta respecto a la arquitectura monolítica superan a las desventajas.

Diagrama 4 – Arquitectura de los servicios del Proyecto VaX

# Software

# Software Empleado

Para el proyecto hemos empleado software gratuito y a poder ser de software libre ya que ese tipo de licencias se amoldan a nuestra filosofía de empresa, a continuación se muestra el listado del software:

* Java:
  + **JDK (Java Development Kit)** : Versión 17 yOpenJDK17.
  + **IDE:** Apache Netbeans: Versión 19 y Versión 17.
* Bases de datos:
  + **Sistema gestor de bases de datos:** MySQL
  + **Entorno grafico:** DBeaver
* Sistemas:
  + **Software de Virtualización :** Oracle VM Virtual Box
  + **Sistema de despliegue:** AWS
  + **Servidor Web:** Apache 2.0
  + **Sistema de Gestión Empresarial:** Odoo
* Entornos de desarrollo
  + **Control de Versiones:** Git | Github | GitLab
  + **Entorno Grafico:** SourceTree

# Software Alternativo

Dejamos software alternativo para si nosotros u otra persona/empresa desea continuar el desarrollo, pero con otro software pueda hacerlo.

* Java:
  + **JDK (Java Development Kit)** : Versión 21 yOpenJDK21.
  + **IDE:** Eclipse | intellij idea
* Bases de datos:
  + **Sistema gestor de bases de datos:** PostreSQL | MariaDB
  + **Entorno grafico:** PGAdmin
* Sistemas:
  + **Software de Virtualización :** VMware
  + **Sistema de despliegue:** Google Cloud | Azure
  + **Servidor Web:** Apache 2.0
  + **Sistema de Gestión Empresarial:** Odoo
* Entornos de desarrollo
  + **Control de Versiones:** Bit | Bitbucket
  + **Entorno Grafico:** GitKraken

# Diseño físico de la base de datos

Partiendo del diseño lógico realizado en el apartado 2.1 (Diseño lógico de la base de datos) hemos realizado el siguiente diseño físico.



Código 1 – Script SQL 1

Texto

Descripción generada automáticamente

Código 2 - Script SQL 2

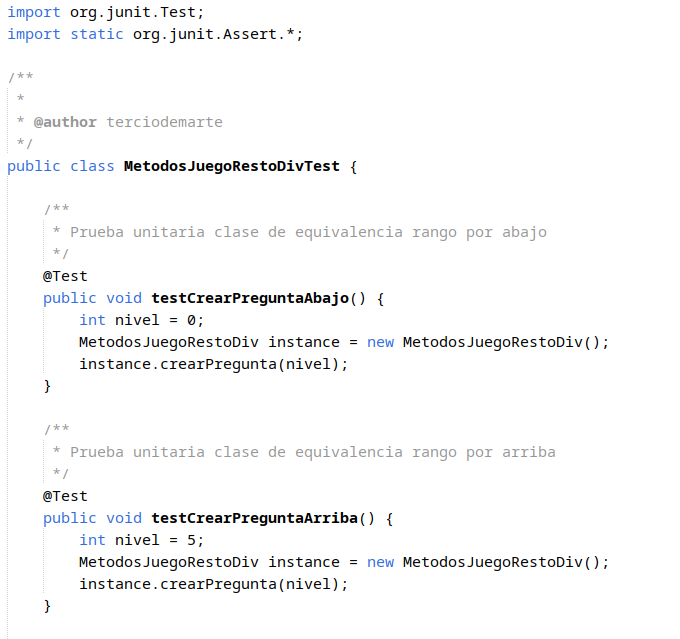
# Código java // FALTA DE HACER

El código java se ha estructurado de forma modular en diferentes paquetes para aumentar la eficiencia y claridad del código y a su vez hemos desarrollado la parte visual de la aplicación como el diseño original porque guarda la modularidad que hemos buscado en todos los apartados del proyecto.

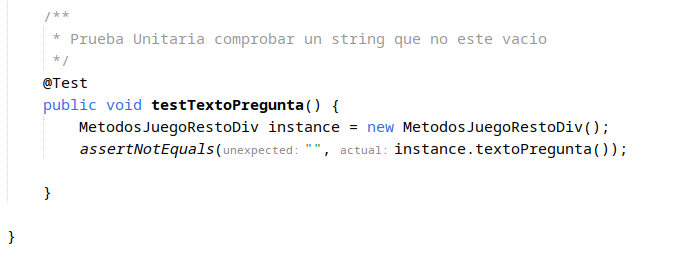
# PRUEBAS // FALTA DE HACER

# Pruebas unitarias sobre código java (JUnit5)

A lo largo del desarrollado hemos realizado pruebas unitarias sobre los métodos públicos de las clases del proyecto para comprobar el correcto funcionamiento de los métodos, a continuación, se muestra diferentes tipos de pruebas unitarias (Booleana, rango y NULL).







# DESPLIEGUE //FALTA DE HACER

# Despliegue Web

Como ya se ha dicho 1.2 (Planteamiento de la solución) la web se ha desplegado en un contenedor de Docker con el servidor web NGNIX, en la que va incluido el fichero XML del RSS, el ejecutable de Windows de la aplicación Agilizando Mentes II para que los usuarios se lo puedan descargar, el manual de usuario de la aplicación, y el javadoc del proyecto para que otro/s desarrollador/es puedan continuar el proyecto y que sepan que función realizan las clases y sus métodos.

# Despliegue aplicación

Hemos decidido que la aplicación se portable ya que no tenemos ninguna dependencia externa a java, solo es necesario tener instalado el JRE, el que ya viene instalado en las distribuciones más populares de GNU/Linux para entornos de escritorio (Ubuntu, Manjaro, Linux Mint) y en las distribuciones de MacOS modernas.

Para el entorno Windows el usuario debe descargar el JRE porque no viene instalado, pero en el manual de usuario hemos advertido de dicho problema y hemos planteado una solución.

# PLANTEAMIENTO FUTURO

# Ideas para continuar el desarrollo

Aparte de las ideas a futuro que planteamos en el apartado 1.2 (Planteamiento de la solución) queremos dejar otra serie de ideas para mejorar los sistemas externos a la aplicación de java.

* Implementar el almacenado del proyecto en AWS
* Implementar la Base de datos en el servidor del instituto o en la nube.
* Dar un aspecto más visual a la web de la empresa.
* Continuar publicando noticias para dar visibilidad a la empresa.

# Control de Versiones del Proyecto

# Git

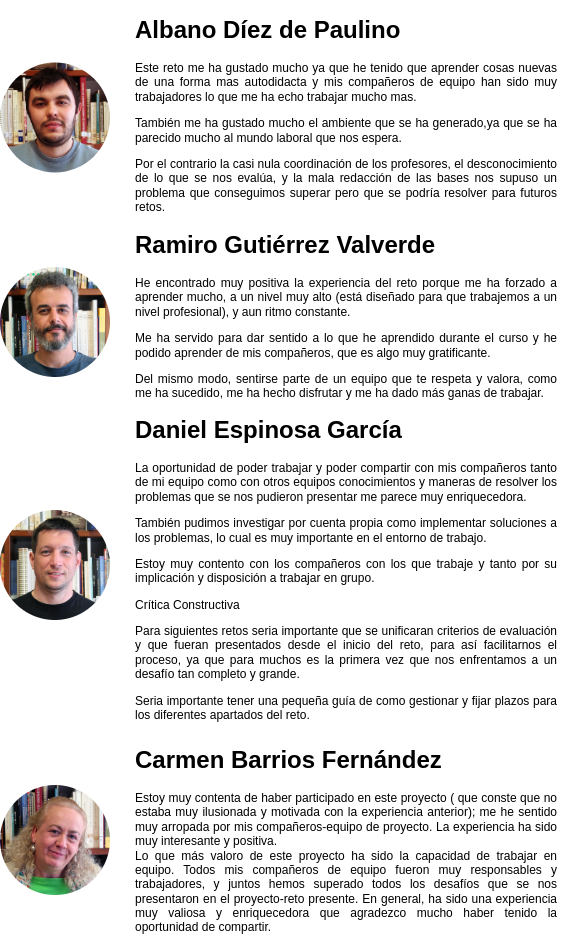
A lo largo del proyecto hemos usado todos los miembros del equipo la herramienta de Git para el control de versiones y GitLab para poder tener un repositorio en la nube para una mejor sincronización de todos los repositorios de los integrantes del equipo.

Gráfico, Gráfico de dispersión, Gráfico de burbujas

Descripción generada automáticamente Para mejor control hemos usado tres repositorios, dos para desarrollar juegos y otro para el proyecto VaX, y a su vez cada proyecto con diferentes ramas siguiente el siguiente principio:

Diagrama 5 – Git Flow

# Opinión de los integrantes del grupo // Falta de Hacer



# Índice Ilustraciones // Falta de Hacer

# Bibliografía

dev.mysql, 2023. [En línea]   
Available at: https://dev.mysql.com/doc/refman/8.0/en/

digitalocean.com, 2023. *digitalocean.com.* [En línea]   
Available at: https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04

digitalocean, 2023. *digitalocean.com.* [En línea]   
Available at: https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04

hub.docker, 2023. *hub.docker.com.* [En línea]   
Available at: https://hub.docker.com/search?type=image&q=

Oracle, s.f. [En línea]   
Available at: https://docs.oracle.com/javase/8/docs/api/

stackoverflow, 2023. *stackoverflow.com.* [En línea]   
Available at: https://stackoverflow.com/

w3schools, 2023. *w3schools.com.* [En línea]   
Available at: https://www.w3schools.com/

wiki.alpinelinux, 2023. *wiki.alpinelinux.org.* [En línea]   
Available at: https://wiki.alpinelinux.org/wiki/Main\_Page