

PROYECTO DAM - ANINEXUS

Contenido

Introducción.....	2
Contexto	2
Presentación.....	2
Objetivos	2
Análisis y diseño de la aplicación	3
Análisis de requisitos funcionales	3
Introducción.....	3
Tabla numerada de requisitos.....	3
Diagrama de casos de uso	4
Introducción.....	4
Diagramas y descripción	4
Diseño de la interfaz.....	6
Introducción.....	6
Diagrama y descripción.....	6
Estructura de datos usada	13
Introducción.....	13
Diagramas y descripción	13
Fase de pruebas.....	14
Pruebas de validación de código	14
pruebas de implementación en distintos soportes	14
Implementación	14
Tecnologías usadas en el desarrollo del proyecto	14
Puesta en marcha. Despliegue	15
Guía de despliegue del proyecto	15
Requisitos previos	15
Pasos	16
Generacion APK.....	19
Aplicación en el ámbito empresarial.....	19
Idea de negocio	19
Planificación de necesidades de RRHH	20
Planificación de necesidades de producción.....	20
Planificación de la Prevención de Riesgos Laborales	20
Previsión de inversión y financiación	20

Estudio de mercado y marketing operativo	21
Conclusión.....	22
Valoración personal del trabajo realizado	22
Posibles ampliaciones.....	22
Bibliografía y webgrafía	22
Anexo	23
Manual de usuario.....	23
Requisitos previos	23
Pasos	23
Código completo del proyecto	28

Introducción

Contexto

Hoy en día, es más frecuente ver anime en la TV o en “streaming”. Casi 1 millón de personas disfrutan de este contenido. A pesar de eso, no existe ninguna app que te muestre los animes que hay cada temporada, junto con información detallada de éstos. Por eso, he decidido crear esta aplicación.

Presentación

AniNexus está diseñado para que fanáticos del anime (coloquialmente conocidos como “otakus”) puedan consultar los animes que cada temporada (primavera, verano, otoño e invierno) se emiten tanto en TV como “streaming”. Además, permite conocer información técnica de este mundo: información del estudio que lo realiza (junto con la de todos los estudios de animación del mercado), los nombres de los directores que los realizan y los géneros y demografías existentes. Aparte, también se puede guardar en favoritos los animes que decidas seguir cada temporada.

Objetivos

Los objetivos de la aplicación son:

- Proporcionar una app donde fanáticos del anime descubran fácilmente los animes que se estrenan cada temporada.
- Acercar a los usuarios al mundo técnico del anime.

Análisis y diseño de la aplicación

Análisis de requisitos funcionales

Introducción

En esta sección, se describen los requisitos funcionales que sirven para el correcto funcionamiento de la aplicación. Estos requisitos son imprescindibles, ya que representan las funcionalidades que debe tener la app para satisfacer las necesidades de los usuarios y el objetivo del proyecto.

Tabla numerada de requisitos

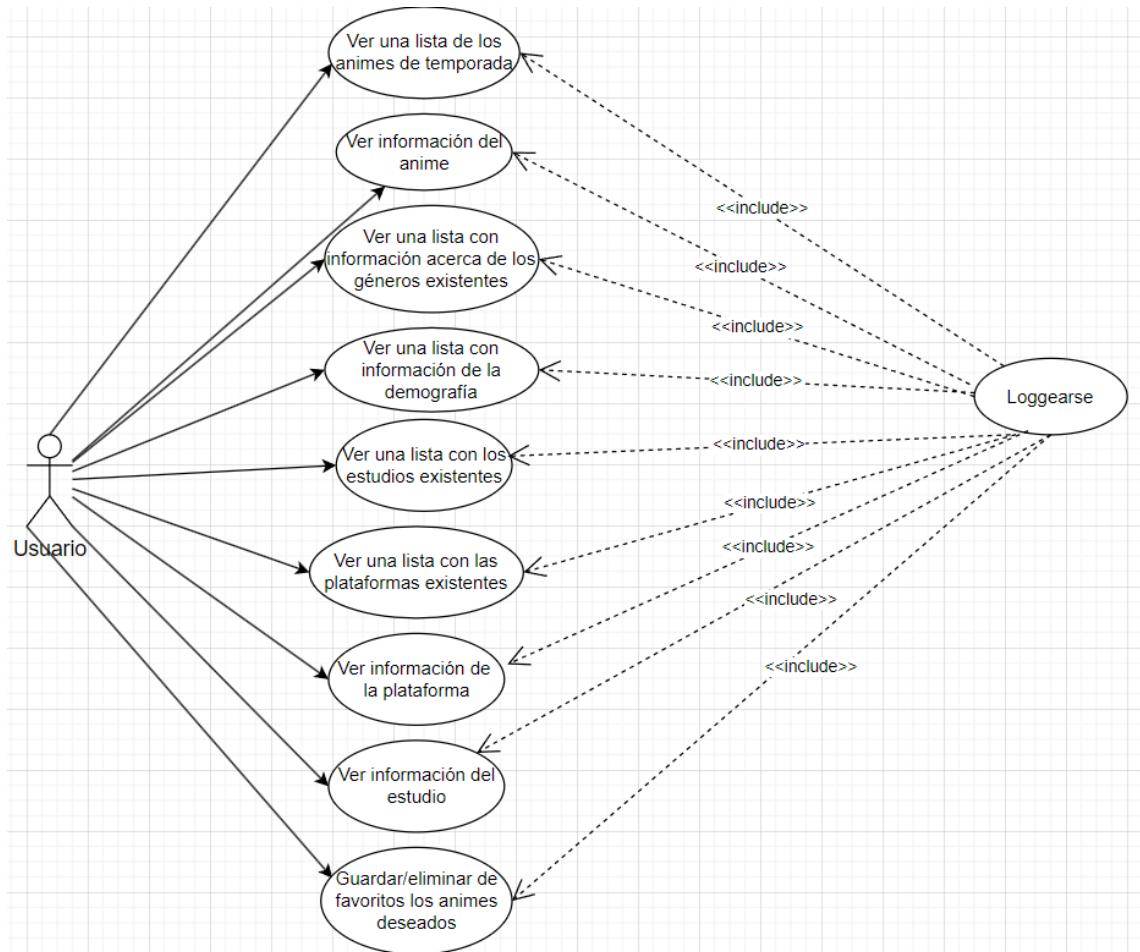
Requisito funcional	Descripción
Registrar usuarios	Permitir que los nuevos usuarios se registren a partir de una serie de campos obligatorios.
Iniciar sesión	Iniciar sesión usando el nombre de usuario y la contraseña.
Eliminar cuenta	Permitir a los usuarios que se quieran ir eliminar su cuenta a través de la introducción de un correo.
Visualizar animes	Una vez que el usuario se ha logueado, mostrar una serie de animes de temporada.
Acceder al anime seleccionado	Al pinchar sobre un anime, acceder a la información que tiene (demografía, género, temporada/s, etc.).
Acceder a la temporada del anime seleccionado	Al hacer clic en una temporada, acceder a dicha temporada y ver su información (estudio, número de episodios, director, etc.).
Acceder al estudio de la temporada seleccionada	Al seleccionar el estudio que realizó la temporada, acceder a él para observar toda su información (sede, trabajadores si los tiene, su web, etc).
Visualizar estudios existentes	Al seleccionar en el menú los estudios, poder ver un listado de estudios con todos los estudios existentes.
Acceder al estudio seleccionado	Al cliquear en un estudio de la lista, acceder a él para observar toda su información (sede, trabajadores si los tiene, su web, etc).
Visualizar demografías existentes	Al seleccionar en el menú las demografías, ver todas las demografías existentes.
Visualizar géneros existentes	Al hacer clic en el apartado de géneros en el menú, poder ver todos los géneros existentes
Visualizar plataformas existentes	Al ir en el menú a la parte de plataformas, observar todas las plataformas de “streaming”.
Acceder a la plataforma seleccionada	Al clicar en una plataforma, ver todos sus datos (sede, url, etc.).
Guardar y eliminar animes de favoritos	Al acceder a un anime, poder guardar en favoritos, lo que llevará al anime a una página “Favoritos”. Al volver a pulsar en ese botón, desmarcarlo de favoritos.
Buscar animes y estudios	Mediante una barra de búsqueda, poder buscar animes y estudios mediante coincidencias

Diagrama de casos de uso

Introducción

Los casos de uso son las descripciones de las acciones que va a tener una aplicación. Son importantes debido a que nos proporcionan información para poder crear la estructura del flujo de trabajo.

Diagramas y descripción



- Ver información del anime: El usuario puede ver la información del anime seleccionado.
- Ver una lista de los animes de temporada: Al usuario se le muestra una lista con los animes de esa temporada (primavera, verano, otoño e invierno).
- Ver una lista con información acerca de los géneros existentes: Al usuario se le muestra una lista con los géneros existentes.
- Ver una lista con información acerca de la demografía: Al usuario se le muestra una lista con las demografías existentes.
- Ver una lista con los estudios existentes: Al usuario se le muestra una lista con los estudios existentes.
- Ver información del estudio: El usuario puede ver la información del estudio seleccionado, ya sea pinchando en la lista de estudios, o en la información de un anime

- Ver una lista con las plataformas existentes.: Al usuario se le muestra una lista con las plataformas de “streaming” existentes.
- Ver información de la plataforma. El usuario puede ver la información de la plataforma seleccionada pinchando en un elemento de la lista de plataformas.
- Guardar/eliminar de favoritos los animes deseados: Los usuarios pueden guardar en favoritos los animes deseados y, una vez allí, eliminarlos de dicha página.



- Introducir estudios: El administrador irá introduciendo los estudios que vayan surgiendo.
- Eliminar animes de temporada: El administrador eliminará los animes de esa temporada unos días antes de acabar dicha temporada.
- Introducir animes de temporada: El administrador introducirá los animes de temporada unos días antes de comenzar la temporada.
- Eliminar animes de favoritos: Al final de la temporada, eliminar los animes que cada usuario tiene en favoritos.
- Editar animes de temporada: El administrador editará los animes en caso de equivocarse a la hora de introducirlos.
- Editar estudio: El administrador editará los estudios en caso de equivocarse a la hora de introducirlos.

- Introducir plataformas: El administrador irá introduciendo las plataformas que vayan surgiendo.
- Introducir géneros: El administrador irá introduciendo los géneros que vayan surgiendo.
- Editar géneros: El administrador editarán los géneros en caso de equivocarse a la hora de introducirlos.
- Editar plataformas: El administrador editarán las plataformas en caso de equivocarse a la hora de introducirlas.
- Eliminar estudios: El administrador eliminará un estudio que no se debía de introducir.
- Eliminar plataformas: El administrador eliminará una plataforma que no se debía de introducir.
- Eliminar géneros: El administrador eliminará un género que no se debía de introducir.

Diseño de la interfaz

Introducción

El diseño de la interfaz es un aspecto clave en el desarrollo de aplicación, ya que condiciona como el usuario va a interactuar con el sistema. Un buen diseño tiene múltiples ventajas: facilita la usabilidad, mejora la experiencia de usuario, atrae a más gente por ser visualmente atractiva.

Al realizar la interfaz se han considerado los siguientes aspectos:

- Accesibilidad: Diseñar la interfaz para que todo tipo de personas puedan acceder a la app.
- Usabilidad: Asegurar que los usuarios puedan acceder a todos los lugares eficientemente, sin pérdidas.
- Estética: Utilizar un diseño limpio, con colores para que sea lo más atractivo posible visualmente.

Diagrama y descripción

Móvil IOS/Android



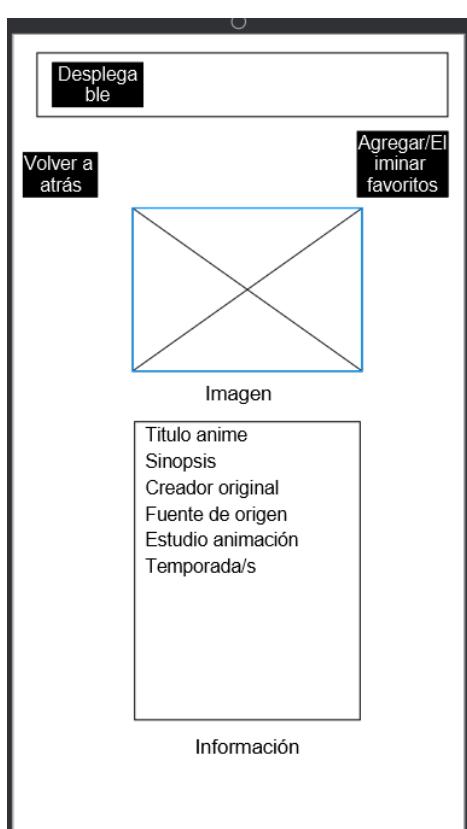
Inicio sesión

A wireframe diagram of a registration form. It consists of several input fields: "Nombre usuario", "Nombre y apellidos", "Email" (aligned with a date field "05/11/1998"), "Contraseña", and "Repetir contraseña". At the bottom is a single action button labeled "Registrarse".

Registrarse



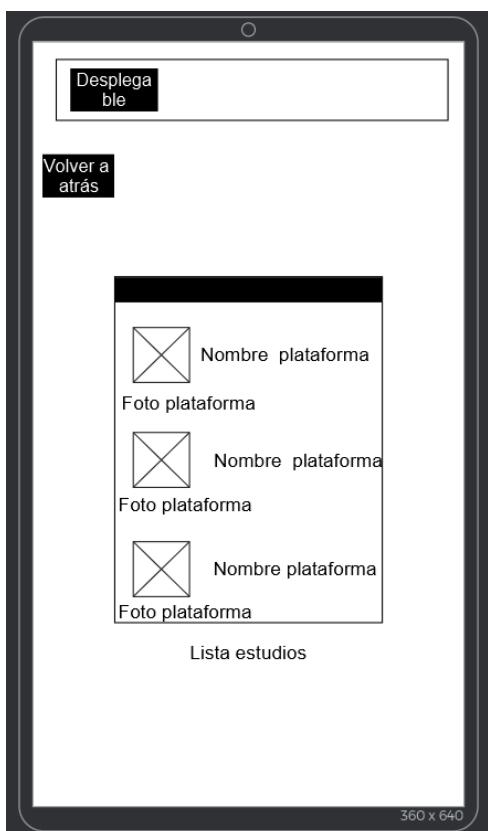
Página inicio



Página anime



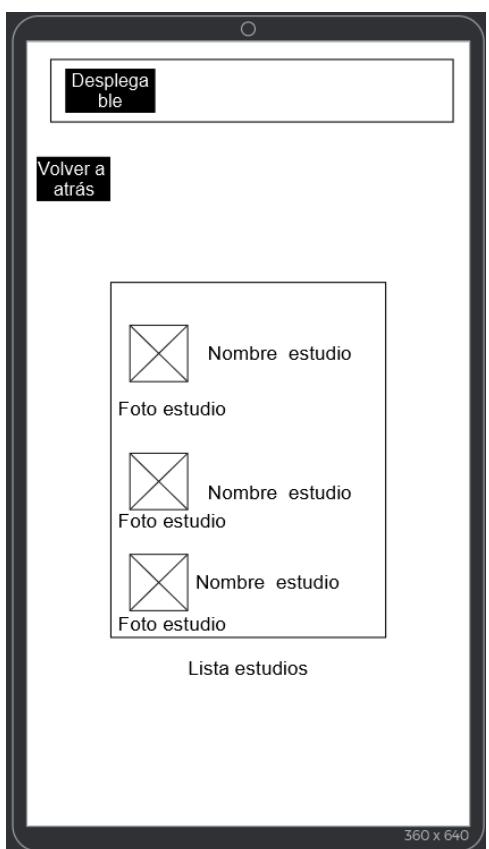
Página temporada



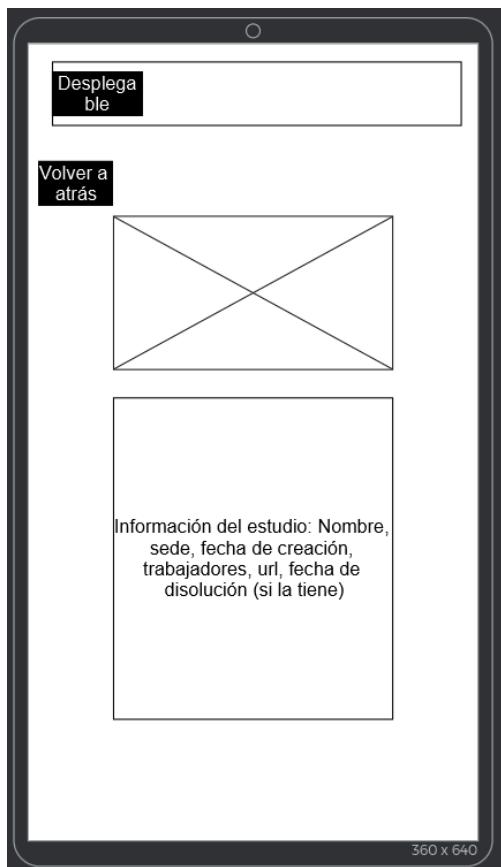
Página listado plataformas



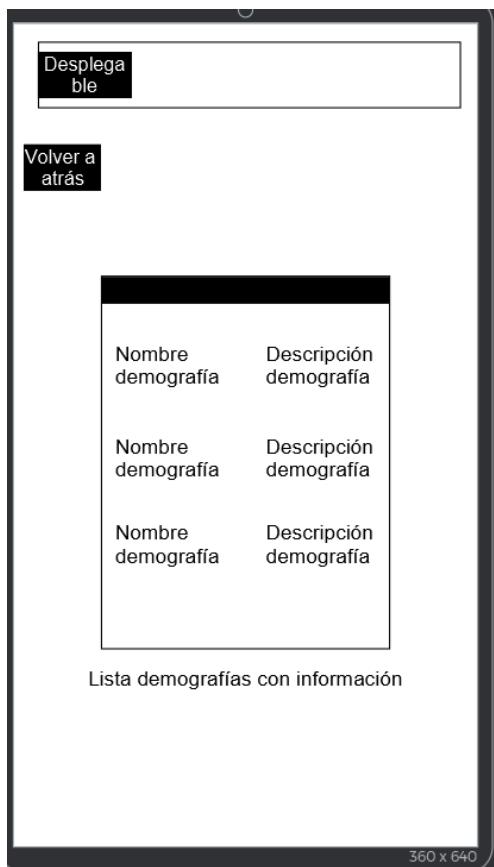
Página plataforma



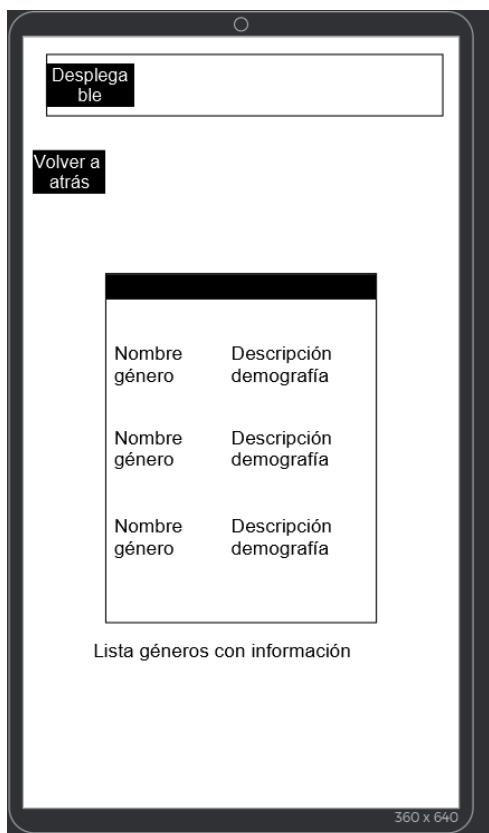
Página lista estudios



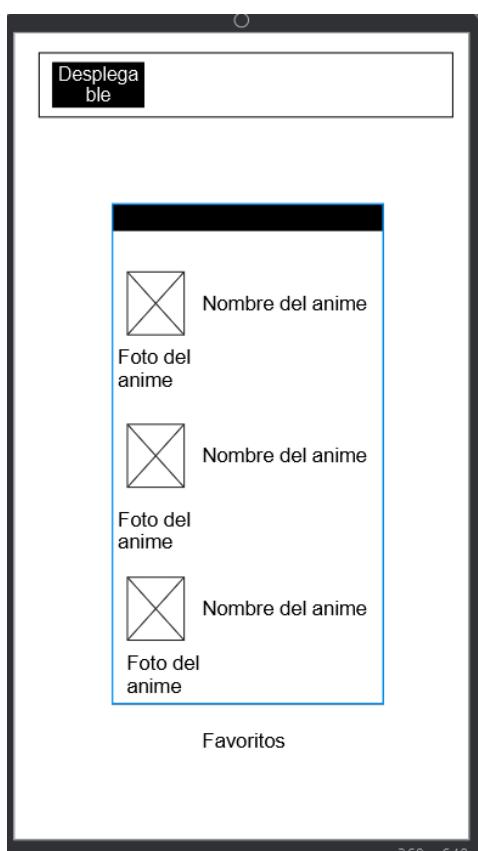
Página estudio



Página lista demografías



Página lista géneros



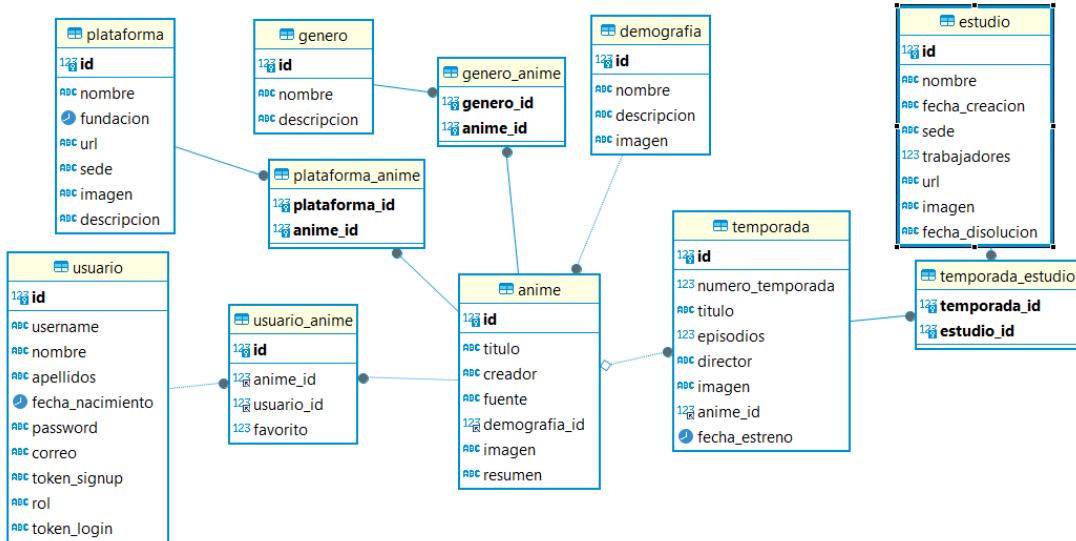
Favoritos

Estructura de datos usada

Introducción

Planificar la estructura de los datos es un factor clave a la hora de desarrollar una aplicación, ya que define como se almacenará, gestionará y recuperará la información. Una estructura de datos bien diseñada va a permitir evitar redundancias, inconsistencias de los datos, reducirá los tiempos de procesamiento y garantizará un acceso a los datos de forma eficiente.

Diagramas y descripción



- Tabla anime
 - o Es la tabla principal que almacena los datos básicos del anime, como título, creador, fuente, imagen, resumen, y si es un favorito.
 - o Está relacionada con otras tablas como genero_anime, demografia, temporada, y plataforma_anime.
- Tabla genero
 - o Contiene información sobre los géneros de anime, con atributos como id, nombre y descripción.
 - o Está relacionada con la tabla genero_anime, que vincula géneros con animes específicos (relación de muchos a muchos).
- Tabla demografia
 - o Define las características demográficas relacionadas con un anime, como su público objetivo (descripción e imagen).
- Tabla temporada
 - o Describe las temporadas de un anime con atributos como numero_temporada, titulo, episodios, director, y fecha_estreno.
 - o Se relaciona con anime (una temporada pertenece a un anime) y con la tabla temporada_estudio.
- Tabla estudio

- Incluye información sobre los estudios de animación, con atributos como nombre, fecha_creacion, sede, trabajadores, imagen, y fecha_disolucion.
 - Se relaciona con las temporadas a través de la tabla temporada_estudio.
- Tabla plataforma
 - Representa las plataformas donde se pueden ver los animes, con atributos como nombre, fundacion, url, e imagen.
 - Relacionada con plataforma_anime para indicar qué animes están disponibles en cada plataforma (relación de muchos a muchos).
- Tabla usuario
 - Contiene información sobre los usuarios del sistema, con atributos como usuario, nombre, apellidos, fecha_nacimiento, correo, y contraseña.
- Tabla usuario_anime:
 - Conecta un usuario con los animes, lo que permite que cada usuario solo vea los animes que guardó en favoritos, y no los de otros usuarios.

Fase de pruebas

Pruebas de validación de código

Realicé tres tipos de pruebas:

- Pruebas unitarias:
 - Comprobé que la autenticación al registrarse, iniciar sesión y borrar la cuenta funcionasen correctamente.
 - Comprobé que al marcar el botón de favorito me lleva ese anime a la página de favoritos, y al pulsar de nuevo ese botón me lo elimina.
- Pruebas de integración:
 - Validé que al seleccionar un anime, un estudio, una temporada, una plataforma de “streaming” de sus listas correspondientes me lleva a esos elementos y se puede ver toda su información.
- Pruebas de sistema:
 - Simulé el flujo completo que realiza un usuario. Se registra, luego inicia sesión, lo que le lleva a la página principal (listado de animes). Manejé el menú para que me lleve a distintos sitios. Luego cerré sesión, lo que me lleva a la pantalla de “login”. Por último, borré la cuenta.

pruebas de implementación en distintos soportes

- Probé a llevar el proyecto al navegador con el modo desarrollador y se me adapta a todos los dispositivos.
- Probé a llevar la app a un emulador de Android (Android Studio). Se ejecuta correctamente.

Implementación

Tecnologías usadas en el desarrollo del proyecto

Las tecnologías que usé para realizar el proyecto son las siguientes:

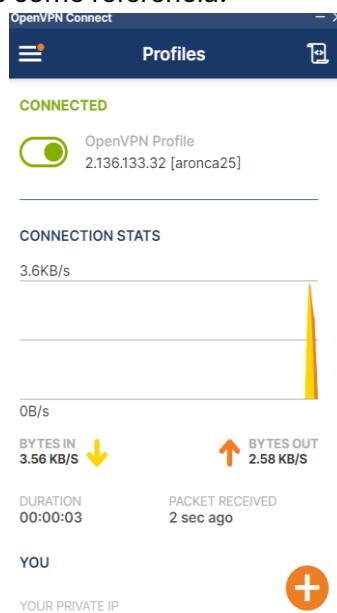
- Lenguaje backend: Java. Decidí usar este lenguaje debido a mi conocimiento sobre él.
- Framework backend: Usé Spring boot, por su integración, por su facilidad a la hora de desarrollar el backend y por su interfaz.
- Entorno backend: IntelliJ Idea. Me decanté por este entorno por su facilidad a la hora de escribir código y sus ayudas.
- Base de datos: MariaDB. Debido a que es la que, personalmente, más me gusta.
- Lenguaje frontend: JavaScript. Decidí usar este lenguaje por ser el lenguaje de frontend que más conozco.
- Framework frontend: Angular e Ionic. Por ser frameworks muy intuitivos, con clases de estilos predefinidos, para realizar aplicaciones con bien diseño y funcionales.
- Entorno frontend: Visual Studio Code. Me decanté por este entorno por su facilidad a la hora de escribir código gracias a sus extensiones.
- Tecnologías usadas para guardar el proyecto en la nube: GitHub. Decidí usar esta tecnología debido a que es la que más conozco.

Puesta en marcha. Despliegue

Guía de despliegue del proyecto

Requisitos previos

- Descargargar OpenVPN Connect.
- Escoger la opción “File”.
- Pulsar en “Browse” y escoger el archivo con extensión OVPN facilitado.
- Establecer un nombre como referencia.



- Conectarnos, mediante la CMD, vía SSH al servidor de proyectos. En mi caso ssh [aronca25@192.168.7.160](ssh://aronca25@192.168.7.160). Inicialmente la contraseña coincide con el nombre de usuario.

```

aronca25@proyecto: ~      + | 
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-138-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of vie 13 jun 2025 15:27:07 UTC

System load: 0.0          Processes:           217
Usage of /: 45.4% of 27.45GB   Users logged in:     0
Memory usage: 35%          IPv4 address for ens160: 192.168.7.160
Swap usage: 1%             Swap usage: 0

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 73 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

*** System restart required ***
Last login: Wed Jun 11 14:17:21 2025 from 192.168.7.150
aronca25@proyecto:~$ |

```

- Cambiar la contraseña por una personalizada.

Pasos

1. Crear Dockerfile para el frontend.

```

FROM nginx:alpine

# Copiar la configuración personalizada de nginx
COPY nginx.conf /etc/nginx/conf.d/default.conf

# Copiar los archivos compilados localmente
COPY www/ /usr/share/nginx/html/

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

server {
    listen 80;
    server_name aronca25.iesmontenaranco.com;

    root /usr/share/nginx/html;
    index index.html;

    location /assets/ {
        try_files $uri $uri/ =404;
    }

    location / {
        try_files $uri $uri/ /index.html;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg|json)$ {
        expires max;
        add_header Cache-Control "public, max-age=31536000";
    }
}

```

2. Creación del Dockerfile para el backend.

```

# Usar la imagen oficial de OpenJDK 21 como base
FROM eclipse-temurin:21-jdk

# Establecer el directorio de trabajo dentro del contenedor
WORKDIR /app

# Copiar el archivo JAR y el script al contenedor
COPY target/Proyecto-0.0.1-SNAPSHOT.jar app.jar
COPY wait-for-it.sh wait-for-it.sh
COPY uploads/ uploads/

# Dar permisos de ejecución al script
RUN chmod +x wait-for-it.sh

# Exponer el puerto en el que corre la aplicación
EXPOSE 5000

# Comando para ejecutar la aplicación (esperando primero si hace falta)
ENTRYPOINT ["/./wait-for-it.sh", "projeto-mariadb:3306", "--", "java", "-jar", "app.jar"]
|

```

3. Creación del docker-compose.yml.

```

4. version: "3.8"

services:
  mariadb:
    image: mariadb:10.5
    container_name: proyecto-mariadb
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: proyecto
      MYSQL_PASSWORD: root
    ports:
      - "3316:3306" # Exponemos el contenedor en 3316 del host
  volumes:
    - ./init_db:/docker-entrypoint-initdb.d/
    - mariadb_data:/var/lib/mysql
  networks:
    - proyecto-network

  backend:
    build:
      context: ./Proyecto
      dockerfile: Dockerfile
    container_name: proyecto-backend
    restart: always
    depends_on:
      - mariadb
    environment:
      SPRING_DATASOURCE_URL: jdbc:mariadb://proyecto-mariadb:3306/proyecto?useSSL=false&allowPublicKeyRetrieval=true
      SPRING_DATASOURCE_USERNAME: root
      SPRING_DATASOURCE_PASSWORD: root
    ports:
      - "8000:5000" # Puerto externo 8000 → interno 5000
  volumes:
    - ./images:/app/images
    - ./Proyecto/uploads:/app/uploads
  networks:
    - proyecto-network

```

```

frontend:
  build:
    context: ./ proyecto-front
    dockerfile: Dockerfile
    container_name: proyecto-frontend
    restart: always
    ports:
      - "80:80"
    networks:
      - proyecto-network

  volumes:
    mariadb_data:

  networks:
    proyecto-network:

```

5. Compilación del proyecto Ionic con ionic build –prod. Esto genera una carpeta www/

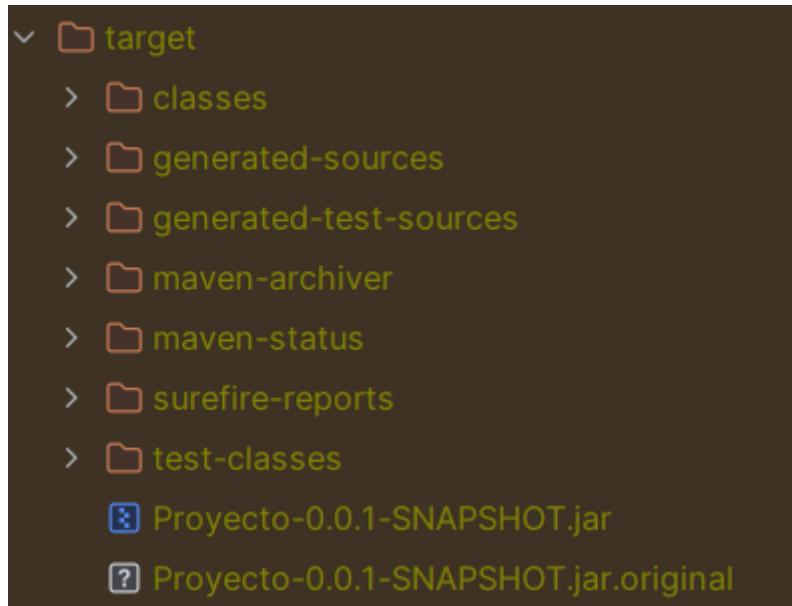
6. Configuración del application.properties.

```

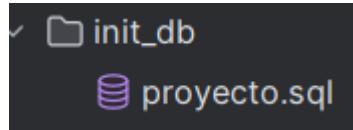
7. spring.application.name=Proyecto
server.port=5000
# Datos del SGBD.
spring.datasource.url=jdbc:mariadb://proyecto-
mariadb:3306/proyecto
#
spring.datasource.url=jdbc:mariadb://localhost:3316/proyecto
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
# configuración de hibernate
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect
.MariaDBDialect
spring.mvc.media-types.json=application/json; charset=UTF-8
spring.jpa.hibernate.naming.physical-
strategy=org.hibernate.boot.model.naming.PhysicalNamingStrate
gyStandardImpl
logging.level.org.springframework=DEBUG
spring.servlet.multipart.enabled=true
spring.servlet.multipart.max-file-size=5MB
spring.servlet.multipart.max-request-size=5MB

```

8. Compilación y construcción del JAR mediante mvn clean package.



9. Exportar tu base de datos para generar un archivo SQL.



10. Subir todos los archivos mediante el comando scp.
11. Una vez que tienes los archivos en tu máquina hacer docker compose up -d.
12. Acceder a las URLs del frontend y backend. Las más son, respectivamente:
 - a. <https://aronca25.iesmontenaranco.com>
 - b. <http://aronca25.iesmontenaranco.com:8000>

Generacion APK

1. Ejecutamos ionic build –prod.
2. Ejecutamos npx cap copy Android. Copia la carpeta www/ al proyecto Android/.
3. Ejecutamos npx cap sync Android. Sincroniza plugins y assets.
4. Ejecutamos npx cap open Android. Abre Android Studio con el proyecto ya listo para ejecutarlo en el móvil.
5. En Android Studio vamos a Build -> Build Bundle(s) / APK(s) -> Build APK(s). Esto genera la APK en la carpeta Android de Ionic.

Aplicación en el ámbito empresarial

Idea de negocio

La idea de negocio se centra en el desarrollo y la gestión de una aplicación móvil especializada en el anime. La aplicación proporcionará una base de datos exhaustiva y actualizada sobre series de anime, incluyendo sinopsis, fichas técnicas, temporadas, géneros, personajes, productores y reseñas.

La plataforma también fomentará la interacción entre los usuarios mediante recomendaciones personalizadas y herramientas de gestión de listas. También incluirá chats privados y grupos donde los usuarios puedan comunicarse entre sí.

Planificación de necesidades de RRHH

Para garantizar el correcto funcionamiento de la aplicación y la calidad del servicio, se requiere la formación de un equipo multidisciplinario:

- Desarrolladores de software (3 personas): Encargados de la creación, actualización y mantenimiento de la aplicación.
- Diseñadores UX/UI (2 personas): Responsables de la experiencia del usuario y el diseño visual de la aplicación.
- Gestores de contenido (4 personas): Especializados en recopilar y verificar información de series de anime.
- Especialistas en marketing (2 personas): Diseño de estrategias para atraer usuarios y fidelizarlos.
- Moderadores de la comunidad (3 personas): Supervisión de la interacción entre usuarios.
- Equipo de soporte técnico (2 personas): Resolución de problemas técnicos.

Planificación de necesidades de producción

- Infraestructura tecnológica: Servidores cloud para almacenar y gestionar grandes volúmenes de información.
- Espacio de trabajo: Oficina con estaciones de trabajo equipadas con tecnología de última generación y acceso a internet de alta velocidad.
- Colaboradores externos: Contratación temporal de traductores y analistas especializados para la clasificación inicial de contenidos.

Planificación de la Prevención de Riesgos Laborares

- Riesgos ergonómicos: Implementación de mobiliario de trabajo ergonómico y pausas activas para prevenir lesiones musculares.
- Riesgos psicosociales: Programas de bienestar laboral para manejar el estrés y la carga laboral.
- Capacitaciones: Talleres de seguridad informática para prevenir riesgos relacionados con la gestión de datos sensibles.

Previsión de inversión y financiación

La inversión inicial requerida para este proyecto se estima en 140.000 euros, desglosados en los siguientes rubros:

- Desarrollo de la aplicación: 50.000 euros, que incluyen el diseño, programación, pruebas y lanzamientos iniciales de la plataforma.
- Infraestructura tecnológica: 30.000 euros destinados a servidores cloud, licencias de software, y equipos de hardware necesarios.
- Gastos operativos del primer año: 20.000 euros para cubrir costos como salarios, mantenimiento técnico y actividades promocionales.
- Alquiler de una parcela en una nave industrial: 40.000 euros anuales, incluyendo un espacio para oficinas con áreas dedicadas al comedor y zonas comunes. Este

coste incluye mobiliario, equipamiento de cocina básica y servicios como electricidad, agua y climatización para garantizar un entorno laboral cómodo y funcional.

La financiación se estructurará en tres vías principales:

1. Recursos propios: Aportación inicial de 30.000 euros por parte de los fundadores, garantizando un compromiso financiero directo.
2. Inversores externos: Obtención de 70.000 euros mediante rondas de inversión con capitalistas interesados en el sector tecnológico y de entretenimiento.
3. Crédito empresarial: Solicitud de un préstamo bancario por 40.000 euros, con condiciones de devolución a mediano plazo.

Además, se contempla la generación de ingresos a través de suscripciones premium, publicidad dentro de la aplicación y colaboraciones estratégicas con distribuidores de anime, lo que permitirá garantizar la sostenibilidad económica del proyecto.

Estudio de mercado y marketing operativo

El estudio de mercado se centra en analizar la creciente popularidad del anime, especialmente entre jóvenes y adultos jóvenes. Según estudios recientes, el público objetivo abarca personas de entre 12 y 40 años, que representan el segmento más activo en plataformas digitales. Además, se ha detectado una tendencia al alza en el consumo de contenido relacionado con el anime, impulsada por servicios de streaming y eventos culturales globales.

Se identifican competidores como MyAnimeList y AniList, Crunchyroll, pero el diferencial radica en la supervisión activa de la comunidad y la actualización constante de contenidos. Esto permite ofrecer un producto más personalizado y seguro.

En cuanto al marketing operativo, las estrategias se estructuran de la siguiente manera:

- Estrategias de atracción:
 - o Campañas en redes sociales como Instagram, TikTok y YouTube, aprovechando influencers del sector anime.
 - o Creación de contenido interactivo, como encuestas, trivias y eventos temáticos, para aumentar el interés de los usuarios.
 - o Colaboraciones con convenciones de anime y tiendas especializadas para promocionar la aplicación.
- Estrategias de retención:
 - o Implementación de un sistema de gamificación, ofreciendo recompensas como insignias, puntos y contenido exclusivo para usuarios activos.
 - o Actualizaciones constantes con nuevas funciones, reseñas y guías de episodios.
 - o Creación de una sección de recomendaciones personalizadas basada en el historial de visualización de cada usuario.

- Canales de distribución: La aplicación estará disponible en Google Play y App Store, optimizando su visibilidad a través de estrategias de ASO (App Store Optimización).
- Métricas de éxito: El éxito de la estrategia será medido a través de indicadores clave como el número de descargas, tiempo promedio de permanencia en la aplicación, y la tasa de conversión de usuarios gratuitos a suscriptores premium.

Con estas acciones, se espera posicionar la aplicación como un referente global para los fanáticos del anime, promoviendo una experiencia única, informativa y entretenida.

Conclusión

Valoración personal del trabajo realizado

Este proyecto me ha permitido afianzar conocimiento del manejo de Spring Boot. También he adquirido nuevos conocimientos con Ionic, a la hora de trabajar con componentes, servicios y demás elementos. Aun así, considero que aun me falta mejora en el apartado de diseño, ya que no soy igual de bueno a la hora de hacer bonita una aplicación como de crearla y hacerla funcional.

Possibles ampliaciones

He aquí lo que quiero implementar a futuro:

- Chats de grupo y privados, para que los diferentes usuarios puedan comunicarse entre sí.
- Una página principal donde se haga un ramdom de todos los animes existentes y por cada entrada a la aplicación aparezcan 10 animes aleatorios de todos los animes existentes hasta la fecha.
- La página de los animes de temporada no va a ser la principal, sino una más.
- Hacer que según la edad de los usuarios se muestren unos animes u otros.

Bibliografía y webgrafía

- Udemy: *Udemy, Inc.* Plataforma de aprendizaje en línea que ofrece cursos de programación y tecnología.
URL: <https://www.udemy.com>
- ChatGPT: *OpenAI*. Modelo de lenguaje basado en IA, utilizado para soporte técnico y redacción.
URL: <https://openai.com/chatgpt>
- Wikipedia: *Wikipedia, The Free Encyclopedia*. Enciclopedia en línea colaborativa que ofrece información sobre una amplia variedad de temas, incluyendo tecnología, frameworks y metodologías de desarrollo.
URL: <https://www.wikipedia.org>

- Grupo Dinamo: *Grupo Dinamo*. Artículo sobre estrenos de temporada de otoño 2024, que proporciona información actualizada sobre tendencias y novedades relacionadas con el contenido audiovisual.
URL: <https://www.grupodinamo.com.co/2008/03/chart-estrenos-temporada-de-otoño-2024.html>
- **Lorenzo Rubio, Iván.** *Introducción a AWS*. Documento en formato PDF.
Contacto: ivanlr@educastur.org

Anexo

Manual de usuario

Este manual tiene el objetivo de guiar a los nuevos usuarios en el uso de la app AniNexus.

Requisitos previos

El único requisito previo es estar conectado a Internet.

Pasos

Para manejar la aplicación, realiza los siguientes pasos:

1. Regístrate si eres nuevo.



2. Inicia sesión



prueba2



3. Te llevará a la pantalla principal, donde se verán los animes de esa temporada.



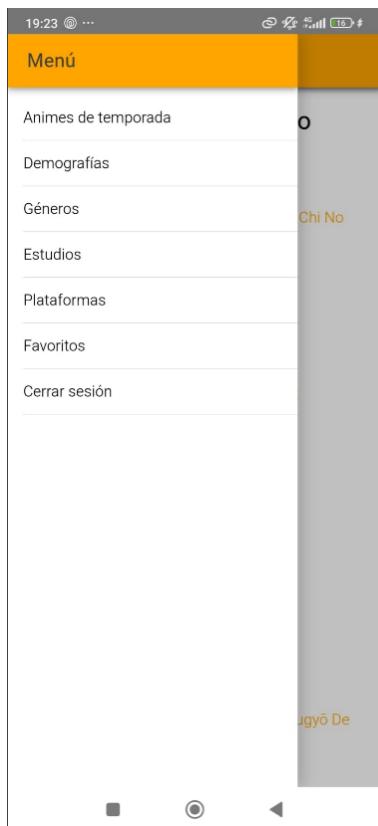
4. Si haces clic en uno de ellos te lleva a ese anime.



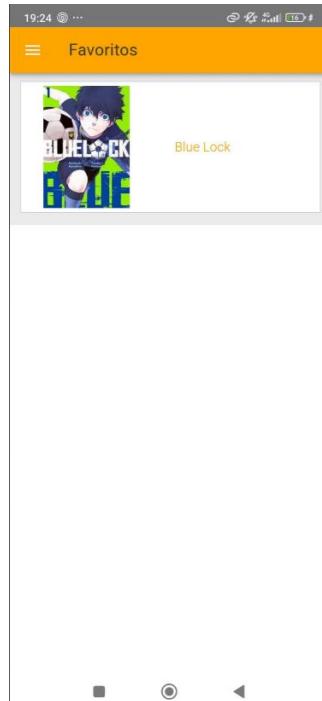
5. Si clicas en la temporada del anime te lleva a esa temporada.



6. En el menú puedes acceder a la lista de demografías, de géneros, de plataformas y de estudios. En estudios y plataformas los pasos a seguir son los mismos que en el listado de animes.



7. Si haces clic en el corazón que hay en el anime (arriba a la derecha) te guarda el anime en favoritos. Te lo quita si lo vuelves a pulsar.



8. Cuando cierras sesión te lleva a la pantalla de login.
9. Con la sesión iniciada, si pulsas en borrar cuenta y pones el correo, se te elimina la cuenta.



NOTA: Los usuarios creados son:

- USER:

- pepepe/Hola123@
 - franro/Total7mod@
- ADMIN:
- admin/Admin123@

Código completo del proyecto

El código subido a GitHub se encuentra en esta url: https://github.com/DAM2A-PDAM-24-25/Carballo_Alvarez_Aron_AniNexus_Junio