



Memoria de proyecto

[Trip And Váter app]

Daniel Merlo Espino
IES MONTE NARANCO, OVIEDO, ASTURIAS.

Índice de la memoria DAM

Contenido

Índice de la memoria DAM.....	1
Introducción	2
Contexto.....	2
Presentación.....	3
Objetivos.	3
Análisis y diseño de la aplicación	4
Análisis de requisitos funcionales	4
Introducción	4
Diagrama de casos de uso.....	4
Diseño de interfaz	5
Introducción	5
Diagramas y descripción.....	9
Estructuras de datos usadas	9
Introducción	9
Diagramas y descripción	10
Fase de pruebas	¡Error! Marcador no definido.
Pruebas de validación de código.....	¡Error! Marcador no definido.
Pruebas de implantación en distintos soportes.....	¡Error! Marcador no definido.
Implementación	10
Tecnologías utilizadas en el desarrollo del proyecto	10
IDEs (Entornos de Desarrollo Integrados).....	10
Frontend.....	11
Backend	11
Bases de Datos	12
Puesta en marcha. Despliegue	13
Guía de despliegue del proyecto.....	13
Arquitectura en el servidor:	13
Conexión con el servidor de proyectos.....	13
Despliegue en el servidor.....	14
Compilación de APK Android	17
APLICACIÓN EN EL AMBITO EMPRESARIAL.....	¡Error! Marcador no definido.

Idea de negocio	¡Error! Marcador no definido.
Planificación de necesidades de RRHH	¡Error! Marcador no definido.
Planificación de necesidades de producción	¡Error! Marcador no definido.
Planificación de la Prevención de Riesgos Laborales	¡Error! Marcador no definido.
Previsiones de inversión y financiación	¡Error! Marcador no definido.
Estudio de mercado y marketing operativo.....	¡Error! Marcador no definido.
CONCLUSIÓN	18
Valoración Personal del Trabajo Realizado.....	18
BIBLIOGRAFÍA Y WEBGRAFÍA	19
ANEXO	19
I. MANUAL DE USUARIO.....	19
II. CÓDIGO COMPLETO DEL PROYECTO (Enlace a GitHub).....	19

Introducción

El presente documento detalla el desarrollo y puesta en marcha de una aplicación móvil enfocada para ayudar a las personas a encontrar baños públicos de manera rápida y fácil. Que sea especialmente útil cuando se está de viaje, en una ciudad desconocida o en una situación de apuro donde necesite encontrar un baño cercano.

Más que un buscador de aseos se pretende que sea una referencia de los sitios mejor valorados por los usuarios, enfocada a encontrar servicios públicos (aseos) cerca de la ubicación donde te encuentres o mostrarte un ranking de los sitios con mejores aseos cuando planeas ir algún lugar. Que Permita filtrar servicios por zona, accesibilidad, los que son gratuitos y los que no, si tienen cambiadores de bebés, etc. Así como puntuarlos por limpieza y el estado sus instalaciones. Siempre promoviendo un buen uso y la conservación de estos.

Contexto

En la actualidad, existen diversas aplicaciones que abordan esta temática, pero muchas de ellas son poco intuitivas y no ofrecen información relevante, limitándose a mostrar unos pocos puntos en un mapa sin mayor referencia. La idea de este proyecto surge al observar una necesidad real para la cual no se ofrecen suficientes soluciones.

Cuando estamos fuera de casa, de viaje o en una zona desconocida, surge la necesidad de encontrar un aseo. Esto es especialmente crítico para personas con necesidades específicas, como quienes requieren aseos accesibles, cambiadores para bebés, o simplemente desean encontrar un lugar en buenas condiciones que asegure su limpieza. Según Human Rights Watch, el acceso a un baño es un derecho global:

"El derecho al saneamiento emana del derecho a un estándar de vida adecuado y reconoce a todas las personas el derecho a acceder a servicios sanitarios que brinden privacidad, garantice la dignidad, y sean físicamente accesibles, asequibles, seguros e higiénicos en un ámbito social y culturalmente aceptable."

Este proyecto pretende satisfacer esa necesidad, brindando una solución accesible, moderna y sostenible para todas las personas que necesiten encontrar un aseo público o de un establecimiento en buenas condiciones, con el respaldo de la comunidad y la tecnología

Presentación

Esta memoria tiene como propósito documentar el desarrollo del proyecto “**TripAndVáter**” cuyo objetivo principal, como se mencionó antes, es el facilitar el acceso a información más detallada sobre aseos y sus ubicaciones. Este trabajo responde a la necesidad de tener información más precisa a la hora de buscar un aseo, intentando aportar una solución innovadora y efectiva.

A lo largo de este documento, se describen los antecedentes, objetivos, análisis y diseño, implementación y puesta en marcha, así como los resultados obtenidos y conclusiones, con el fin de proporcionar una visión clara y estructurada del trabajo realizado.

Objetivos.

- **Promover el buen uso, higiene y conservación de los aseos públicos y de los establecimientos:**
El proyecto busca incentivar la responsabilidad compartida entre usuarios, propietarios y la comunidad para garantizar que los aseos se mantengan en buenas condiciones. Se incentivará la participación de los usuarios mediante la posibilidad de reportar problemas y compartir reseñas sobre la limpieza y estado de los aseos.
- **Facilitar el acceso a información sobre localización, accesibilidad y el estado en que se encuentran los aseos:**
La aplicación proporcionará una plataforma que permita encontrar aseos cercanos a la ubicación del usuario, mostrando detalles clave como accesibilidad, disponibilidad de cambiadores y la calificación general del aseo. Esta información será actualizada en tiempo real, en la medida de lo posible, y estará acompañada de rutas para llegar al aseo de manera rápida y eficiente.
- **Crear la base de datos más completa de España de aseos públicos o de establecimientos, alimentada y moderada por la comunidad:**
Uno de los objetivos centrales es formar la mayor base de datos sobre aseos públicos y de establecimientos en España. Esta base de datos será alimentada tanto por los usuarios, que podrán añadir y revisar aseos, como por la comunidad en general, que podrá validar la información a través de reseñas y reportes.
- **Promover la sostenibilidad en el uso de los aseos públicos:**
Se fomentará el uso de aseos que adopten medidas sostenibles, destacando aquellos que implementen prácticas ecológicas como el ahorro de agua y el uso de productos respetuosos con el medio ambiente. Los usuarios podrán filtrar los resultados para

encontrar este tipo de aseos, ayudando así a promover una mayor conciencia medioambiental.

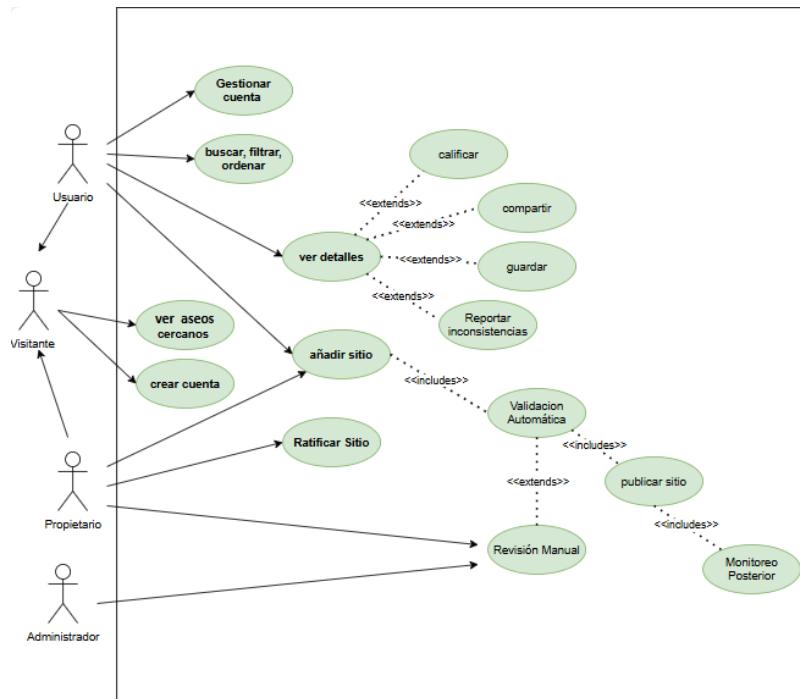
Análisis y diseño de la aplicación

Análisis de requisitos funcionales

Introducción

El análisis de requisitos funcionales es el proceso de identificar, documentar y validar las funciones específicas que un sistema debe realizar para satisfacer las necesidades del usuario. Es un paso crucial en el desarrollo de software y sistemas, asegurando que el producto final cumpla con las expectativas y sea útil para los usuarios.

Diagrama de casos de uso



Actores:

- Usuario:** Este actor representa a un usuario registrado en el sistema, que tiene acceso a más funcionalidades que otros actores (como el visitante).
- Visitante:** El visitante es un usuario no registrado que tiene funciones limitadas dentro del sistema.
- Propietario:** Representa una entidad externa al sistema, empresa o establecimiento público o privado donde se ubican los aseos, implicada en la gestión de estos y las condiciones de uso.

Casos de Uso (Funciones del sistema):

- **Crear Cuenta:** Caso de uso para que los visitantes se registren en el sistema.
- **Mostrar aseos cercanos:** muestra a los usuarios o visitantes los aseos próximos a su ubicación actual
- **Buscar, filtrar, ordenar aseos:** Permite a los usuarios una búsqueda más detallada en ubicaciones, aplicar distintos filtros como limpieza, gratuidad, accesibilidad etc. y ordenar por calificaciones.
- **Ver detalles:** Muestra todos los detalles acerca del sitio.
- **Calificar:** Los usuarios pueden calificar los sitios por distintos criterios que formaran un ranking.
- **Guardar:** Permite almacenar una lista de los sitios favoritos.

Proceso para añadir ubicaciones:

Actores:

1. **Usuario/Propietario:** El actor que añade el sitio.
2. **Comunidad de Usuarios:** En algunos casos, los usuarios ayudan a verificar la existencia o precisión de la información.

Caso de Uso:

Añadir Sitio: Este es el punto de partida donde el usuario o propietario introduce la información del lugar. Se obtiene de forma automática la dirección, se realiza una serie de validaciones básicas en el formulario de datos imprescindible para poder añadir el aseo (nombre, dirección, latitud y longitud obtenidas de las coordenadas, formato de datos, etc.).

Diseño de interfaz

Introducción

El diseño de la interfaz se planteó desde un principio como algo sencillo pero atractivo, fácil de usar e intuitivo, con pocas pantallas y acceso rápido a las diferentes funciones de la aplicación también con mensajes que guían al usuario en todo momento para que la experiencia sea fluida y agradable.



Pantalla de inicio

¡Añadir nuevo!

Nombre del lugar
Enter name

Dirección
selecciona ubicación en el mapa

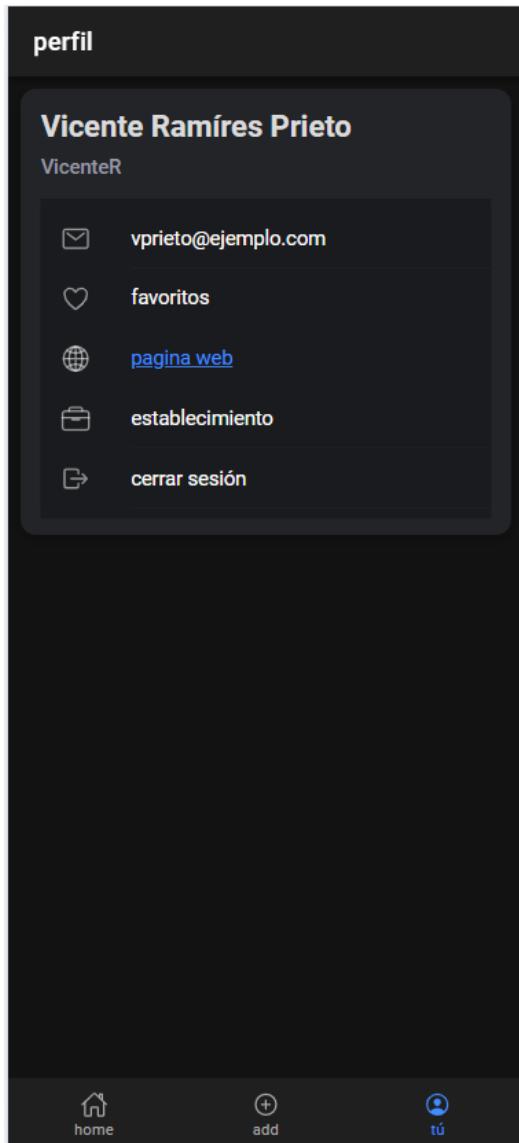
horarios
e.g., 9 AM - 5 PM

acceso
PUBLIC

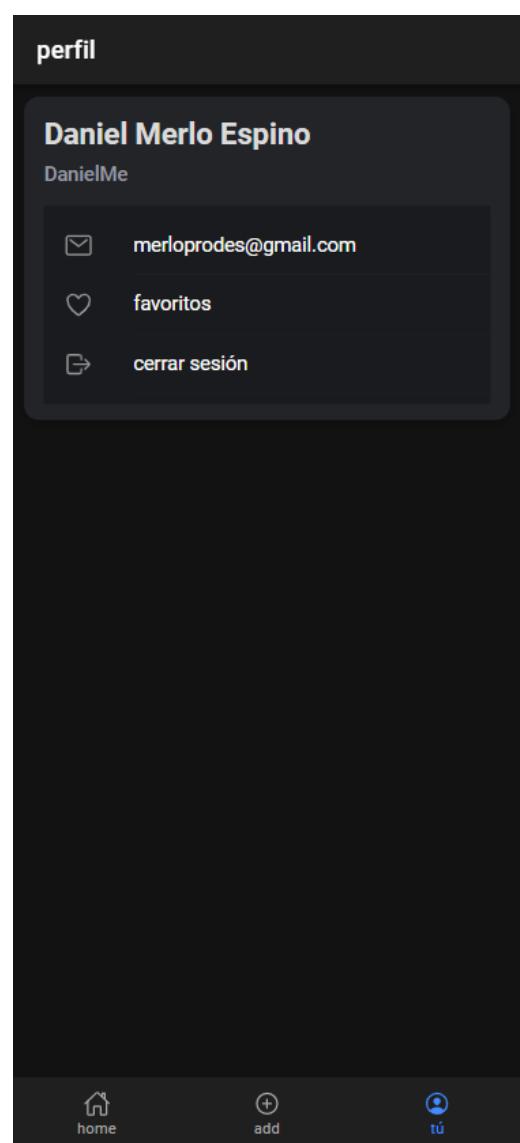
Descripción

home add tú

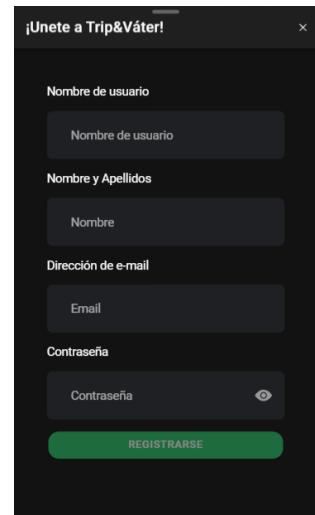
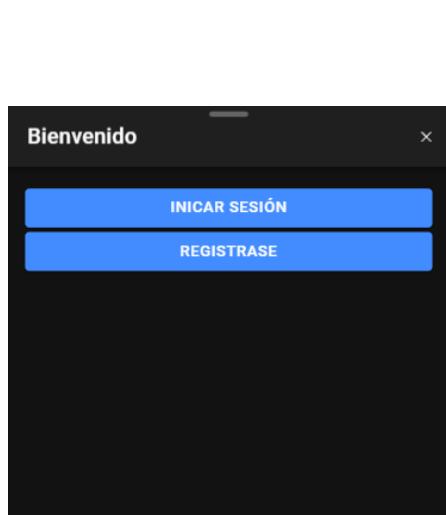
Pantalla para añadir nuevo



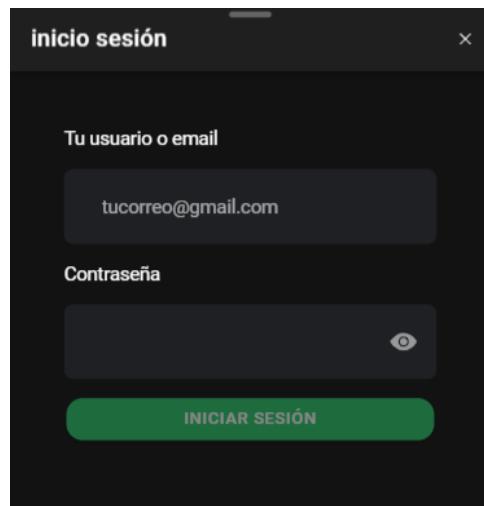
Pantalla perfil Propietario



Pantalla perfil Usuario



Modal de Bienvenida – Modal para Registro de usuario



Modal para Login.

A dark-themed details screen for a cycling lane. It shows a blue header with a city skyline icon and a back arrow. Below is a large blue image of the lane. At the top right are four circular icons: "Cómo ir" (location), "favorito" (heart), "Comparte" (share), and "valora" (star). Below the image, the location is listed as "WC-Carril bici Avenida de El Llano, Carril bici Avenida de El Llano, Distrito Llano, Gijón / Xixón, Asturias / Asturies, 33209, España". It includes "n.e.p", "Horario: n.e.p", and "Acceso: PUBLIC". A "tarifa" section shows a lock icon and "Con tarifa". The "Reseñas" section shows one review from "DanielMe" with a 5-star rating. Navigation icons for "home", "add", and "tú" are at the bottom.

Pantalla de Detalles

A dark-themed review screen. It has a header with "detalles" and "reseña" buttons. Below are five rating sections: "limpieza" (5 stars), "accesibilidad" (5 stars), "Comodidades" (5 stars), "Privacidad" (5 stars), and "Instalaciones" (5 stars). A "commentario" input field is below the ratings. A green "SUBMIT" button is at the bottom. Navigation icons for "home", "add", and "tú" are at the bottom.

Pantalla Reseña

The screenshot shows the Swagger UI interface for the tripAndVater API. At the top, there's a navigation bar with the 'Swagger' logo and a 'Explore' button. Below it, the URL 'v3/api-docs' is displayed along with an 'OAS 3.0' badge. The main content area is titled 'OpenAPI definition' and shows the 'tripAndVater API. Version 1'. It lists four operations under the '/api/toilet' endpoint:

- PUT** /api/toilet/{id}/update/{userId}
- POST** /api/toilet/create/{userId}
- GET** /api/toilet Obtener aseos por filtros
- GET** /api/toilet/detail

Interfaz de OpenApi

Diagramas y descripción

Estructuras de datos usadas

Introducción

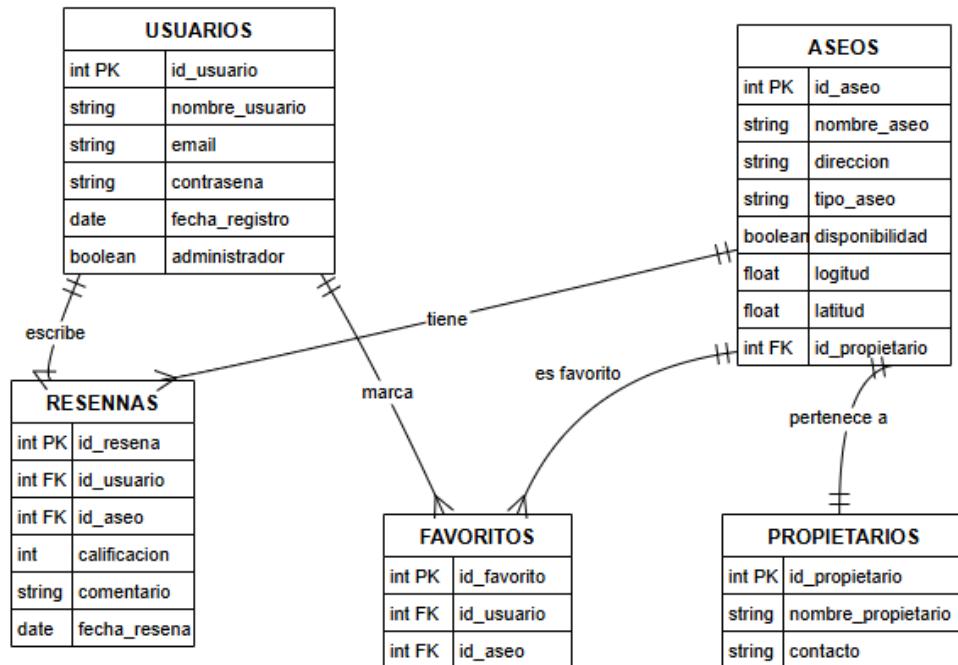
La siguiente imagen muestra la estructura de una base de datos relacional donde definen las entidades y sus relaciones. Es importante destacar que esta estructura puede ser flexible y adaptarse según las necesidades de la aplicación.

Obtención de datos de aseos públicos:

OpenStreetMap (OSM)

- **Descripción:** OpenStreetMap es una fuente global de datos geográficos que incluye aseos públicos. Los datos se actualizan por la comunidad,
- **API :** [Overpass API](#)

Diagramas y descripción



Implementación

Tecnologías utilizadas en el desarrollo del proyecto

IDEs (Entornos de Desarrollo Integrados)

IntelliJ IDEA

Descripción: Es un IDE desarrollado por JetBrains, diseñado principalmente para el desarrollo en lenguajes como Java, Kotlin. Ofrece una potente integración con herramientas de construcción y soporte para frameworks modernos.

Uso en la aplicación: Se utiliza para el desarrollo del backend en **Java** y el framework **Spring Boot**.

Sitio web: <https://www.jetbrains.com/idea/>

Visual Studio Code

Descripción: Este es un editor de código fuente ligero y potente desarrollado por Microsoft, con soporte para depuración, control de versiones y extensiones para lenguajes y frameworks como **TypeScript**, **Angular**, **Ionic**, entre otros.

Uso en la aplicación: Se utiliza para el desarrollo del frontend con tecnologías como **HTML**, **CSS**, **TypeScript**, y **Angular**.

Sitio web: <https://code.visualstudio.com/>

Frontend

HTML, CSS

Descripción: HTML (HyperText Markup Language) y CSS (Cascading Style Sheets) son las tecnologías base para la estructura y diseño visual de las interfaces web.

Uso en la aplicación: Se utiliza para estructurar y estilizar el contenido en las páginas del frontend de la aplicación.

Sitio web: <https://developer.mozilla.org/en-US/docs/Web/HTML>

TypeScript

Descripción: Un lenguaje de programación basado en JavaScript que añade tipos estáticos opcionales, mejorando la robustez del código.

Uso en la aplicación: Se utiliza para desarrollar la lógica del frontend en combinación con Angular.

Sitio web: <https://www.typescriptlang.org/>

Angular

Descripción: Un framework de desarrollo web frontend basado en TypeScript, que facilita la creación de aplicaciones web de una sola página (SPA) mediante componentes reutilizables.

Uso en la aplicación: Se utiliza para la creación del frontend y la gestión de las vistas y componentes de la aplicación.

Sitio web: <https://angular.io/>

Ionic

Descripción: Un framework de código abierto para el desarrollo de aplicaciones móviles híbridas que utiliza tecnologías web como **HTML, CSS, y JavaScript**.

Uso en la aplicación: Permite la creación de la aplicación móvil a partir del mismo código que el frontend, utilizando Angular.

Sitio web: <https://ionicframework.com/>

Capacitor

Descripción: Una herramienta de código abierto que permite que las aplicaciones web creadas con Ionic y otras tecnologías se ejecuten como aplicaciones nativas en dispositivos móviles.

Uso en la aplicación: Capacitor permite la integración de funcionalidades nativas de dispositivos móviles en la aplicación híbrida.

Sitio web: <https://capacitorjs.com/>

Backend

Java

Descripción: Un lenguaje de programación orientado a objetos ampliamente utilizado para el desarrollo de aplicaciones empresariales.

Uso en la aplicación: Se implementa en el desarrollo del backend para manejar la lógica de negocio de la aplicación.

Sitio web: <https://www.oracle.com/java/>

Spring Boot

Descripción: Un framework basado en Java para crear aplicaciones backend rápidas y seguras, con configuraciones mínimas.

Uso en la aplicación: Spring Boot se utiliza para el desarrollo de la API y la gestión del backend.

Sitio web: <https://spring.io/projects/spring-boot>

JPA / Hibernate

Descripción: JPA (Java Persistence API) es una especificación para el mapeo de datos a objetos Java, e Hibernate es su implementación más popular.

Uso en la aplicación: Se utiliza para el mapeo de las entidades de la base de datos en el backend, facilitando la interacción con el sistema gestor de bases de datos.

Sitio web: <https://hibernate.org/>

Bases de Datos

MySQL maría db

Descripción: Sistemas gestor de bases de datos relacionales de código abierto, conocido por su robustez, escalabilidad y soporte para transacciones complejas.

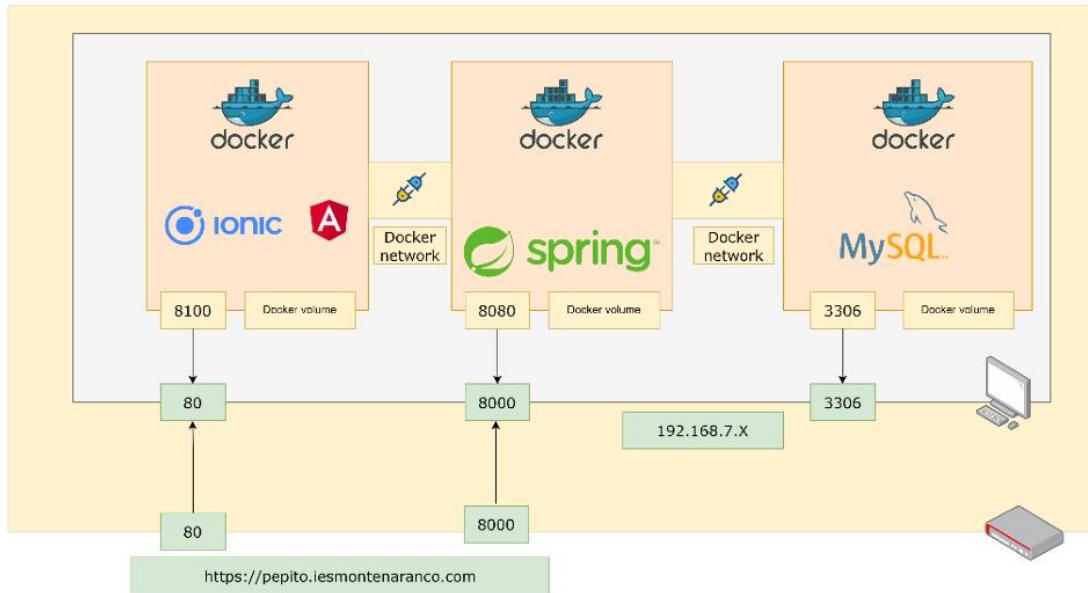
Uso en la aplicación: Se utiliza persistir la información de los usuarios, aseos, reseñas y otros datos relevantes de la aplicación.

Sitio web: <https://mariadb.org/>

Puesta en marcha. Despliegue

Guía de despliegue del proyecto

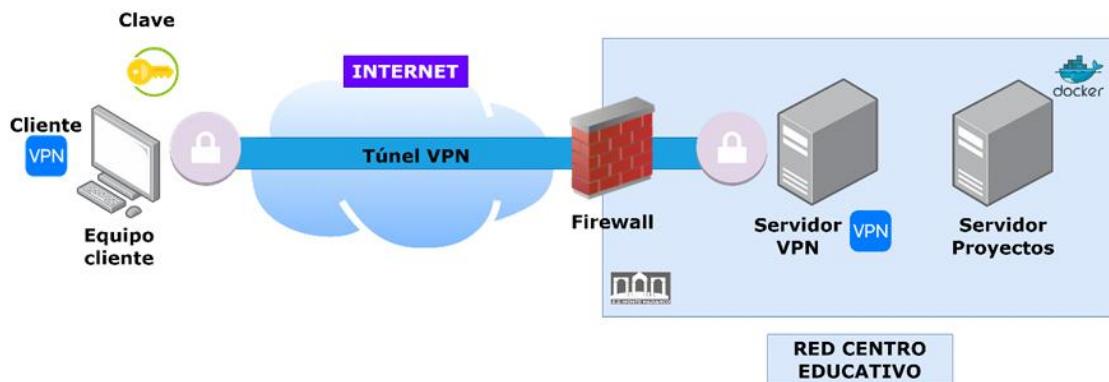
Arquitectura en el servidor:



Conexión con el servidor de proyectos

La conexión se realiza a través de VPN al servidor OpenVPN que tiene en el centro educativo.

Esta conexión VPN permite establecer un canal de comunicación seguro a través de Internet desde mi ubicación hacia el instituto. Utilizando un cliente VPN y una clave que previamente me proporcionaron se crea un túnel seguro que permite conectarse a la red del instituto.



Una vez conectado al servidor del instituto por VPN, en mi caso decidí crear una clave SSH para para agilizar la conexión y poder posteriormente automatizar el proceso de despliegue.

Estos son los pasos que seguí para configurar la clave SSH:

En el servidor:

1. Crear el directorio `~/.ssh/`

```
mkdir -p ~/.ssh
```

Esto crea la carpeta si no existe.

2. Establecer los permisos correctos

```
chmod 700 ~/.ssh
```

Esto asegura que solo mi usuario pueda acceder a la carpeta.

3. Crear el archivo `authorized_keys`

```
touch ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

Esto crea el archivo donde se guarda la clave pública.

Para verificar que todo es correcto:

```
ls -ld ~/.ssh ~/.ssh/authorized_keys
```

Debería mostrar los siguientes permisos:

```
drwx----- 2 .ssh
-rw----- 1 authorized_keys
```

Desde el equipo local:

1. Generar una clave SSH (si no se tiene una)

```
ssh-keygen -t ed25519 -C "mi-conexion-github"
```

- En mi caso, cuando preguntó dónde guardar la clave, presioné **Enter** (para que usará la ruta por defecto: `~/.ssh/id_ed25519`).
- También dejé la contraseña en blanco.

2. Verificar si se ha creado la clave pública

```
cat ~/.ssh/id_ed25519.pub
```

3. copiar la clave pública al archivo `authorized_keys` en el servidor.

```
cat ~/.ssh/id_ed25519.pub | ssh danielme25@IP_DEL_SERVIDOR "cat >> ~/.ssh/authorized_keys"
```

Yo como ya estaba conectado al servidor abrí el fichero con **nano** `~/.ssh/authorized_keys`

Despliegue en el servidor

Para lograr esta arquitectura se llevaron a cabo los siguientes pasos:

1. Creación de 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;"]

```

2. Creación de Dockerfile para el backend

```

FROM eclipse-temurin:17-alpine

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

# Copiar el archivo JAR de la aplicación al contenedor
COPY target/TripAndVater-0.0.1-SNAPSHOT.jar app.jar

# Exponer el puerto en el que corre la aplicación
EXPOSE 8080
EXPOSE 587

# Comando para ejecutar la aplicación
ENTRYPOINT ["java", "-jar", "app.jar"]

```

3. Orquestación de servicios en un docker-compose.yml

```

services:
    # Frontend (Ionic/Angular)
    frontend:
        build:
            context: ./frontend
            dockerfile: Dockerfile
        container_name: panaurum-frontend
        restart: always
        ports:
            - "80:80"
        networks:
            - panaurum-network

    mysql:
        image: mariadb:10.4.32
        container_name: panaurum-mysql
        restart: always
        environment:
            MYSQL_ROOT_PASSWORD: root
            MYSQL_DATABASE: panaurum_db
            MYSQL_USER: panaurum
            MYSQL_PASSWORD: Naranco.24
        ports:
            - "3306:3306"
        volumes:
            - ./initdb:/docker-entrypoint-initdb.d/
            - mysql_data:/var/lib/mysql
        networks:
            - panaurum-network

    # Backend (Spring Boot)
    backend:
        build:
            context: ./backend
            dockerfile: Dockerfile
        container_name: panaurum-backend
        restart: always
        depends_on:
            - mysql
        environment:
            SPRING_PROFILES_ACTIVE: prod # <- Activa el perfil "prod"
            SPRING_DATASOURCE_URL: jdbc:mysql://mysql:3306/panaurum_db?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC
            SPRING_DATASOURCE_USERNAME: panaurum
            SPRING_DATASOURCE_PASSWORD: Naranco.24
            APP_DATA_RUN: false
        ports:
            - "8080:8080"
        networks:
            - panaurum-network
        volumes:
            - ./images:/app/images # Montar imágenes dentro del contenedor
    volumes:
        mysql_data:
    networks:
        panaurum-network:

```

4. Construcción y puesta en marcha de los contenedores

Para el despliegue construcción y puesta en marcha de los contenedores cree los siguientes **scripts** donde se detallan los pasos para construir el proyecto, crear copia de seguridad copiar los nuevos cambios y reconstruir los contenedores.

Deploy-frontend.sh:

```

$ deploy-frontend.sh
You, hace 5 días | 1 author (You)
1 #!/bin/bash You, hace 5 días • Se añade script de despliegue a la raiz del proy...
2 USER=danielme25
3 HOST=192.168.7.161
4 KEY=~/ssh/id_ed25519
5 LOCAL_DIR="/c/Users/laura/Desktop/TripAndVaterApp/TripAndVater-frontend-DanielMerlo/www"
6 REMOTE_DIR="/home/danielme25/despliegue-proyecto/frontend/www"
7 BACKUP_DIR="/home/danielme25/despliegue-proyecto/frontend/backups"
8 TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
9
10 echo "🔨 Construyendo la aplicación Ionic..."
11 cd /c/Users/laura/Desktop/TripAndVaterApp/TripAndVater-frontend-DanielMerlo
12 ionic build --prod
13
14 if [ $? -ne 0 ]; then
15   echo "✖ Error en la construcción de Ionic. Abortando."
16   exit 1
17 fi
18
19 echo "📦 Haciendo copia de seguridad del frontend actual (si existe)..."
20 ssh -i "$KEY" "$USER@$HOST" << EOF
21   mkdir -p "$BACKUP_DIR"
22   if [ -d "$REMOTE_DIR" ] && [ "$(ls -A $REMOTE_DIR)" ]; then
23     cp -r "$REMOTE_DIR" "$BACKUP_DIR/www_backup_$TIMESTAMP"
24     echo "👉 Backup creado en: $BACKUP_DIR/www_backup_$TIMESTAMP"
25   else
26     echo "[!] No hay archivos actuales para respaldar."
27   fi
28 EOF
29
30 echo "⚡ Limpiando el directorio remoto..."
31 ssh -i "$KEY" "$USER@$HOST" "rm -rf $REMOTE_DIR/*"
32
33 echo "☁ Copiando archivos desde $LOCAL_DIR ..."
34 scp -i "$KEY" -r "$LOCAL_DIR"/* "$USER@$HOST:$REMOTE_DIR"
35
36 if [ $? -eq 0 ]; then
37   echo "✅ Archivos copiados correctamente. Reconstruyendo contenedores..."
38
39 ssh -i "$KEY" "$USER@$HOST" << EOF
40   cd /home/danielme25/despliegue-proyecto
41   docker-compose down
42   docker-compose build --no-cache
43   docker-compose up -d
44 EOF
45
46 else
47   echo "✖ Error al copiar archivos."
48 fi

```

Deploy-backend.sh:

```

1  #!/bin/bash
2  USER=danielme25
3  HOST=192.168.7.161
4  KEY=~/ssh/id_ed25519
5  JAR_NAME="TripAndVater-0.0.1-SNAPSHOT.jar"
6  LOCAL_FILE="./target/$JAR_NAME"
7  REMOTE_TARGET_DIR="/home/$USER/despliegue-proyecto/backend/target"
8  REMOTE_BACKUP_DIR="/home/$USER/despliegue-proyecto/backend/backups"
9  # 1. Compilar el proyecto sin ejecutar tests
10 echo "💡 Compilando el proyecto con Maven...""
11 mvn clean package -DskipTests
12 # 2. Verificar si el archivo .jar se generó
13 if [ ! -f "$LOCAL_FILE" ]; then
14   echo "✖ ERROR: El archivo $LOCAL_FILE no existe o no se encuentra."
15   exit 1
16 fi
17 # 3. Crear backup remoto si el .jar ya existe
18 echo "📦 Haciendo copia de seguridad del .jar anterior (si existe)..."
19 ssh -i "$KEY" "$USER@$HOST" bash << EOF
20   mkdir -p "$REMOTE_BACKUP_DIR"
21   if [ -f "$REMOTE_TARGET_DIR/$JAR_NAME" ]; then
22     TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
23     cp "$REMOTE_TARGET_DIR/$JAR_NAME" "$REMOTE_BACKUP_DIR/TripAndVater-0.0.1-SNAPSHOT_\$TIMESTAMP.jar"
24     echo "👉 Backup creado: TripAndVater-0.0.1-SNAPSHOT_\$TIMESTAMP.jar"
25   else
26     echo "ℹ️ No hay .jar previo que respaldar."
27   fi
28 EOF
29 # 4. Limpiar el directorio remoto
30 echo "⚡ Limpiando el directorio remoto..."
31 ssh -i "$KEY" "$USER@$HOST" "rm -rf $REMOTE_TARGET_DIR/*"
32
33 # 5. Copiar nuevo .jar al servidor
34 echo "👉 Copiando $LOCAL_FILE al servidor..."
35 scp -i "$KEY" "$LOCAL_FILE" "$USER@$HOST:$REMOTE_TARGET_DIR/"
36
37 # 6. Verificar resultado y reiniciar contenedores si OK
38 if [ $? -eq 0 ]; then
39   echo "✅ Archivo copiado correctamente. Reconstruyendo contenedores..."
40   ssh -i "$KEY" "$USER@$HOST" << EOF
41     cd /home/$USER/despliegue-proyecto
42     docker-compose down
43     docker-compose build --no-cache
44     docker-compose up -d
45 EOF
46 else
47   echo "✖ Error al copiar el archivo."
48 fi

```

Compilación de APK Android

Añadir Capacitor

Capacitor se usa para trabajar con las capacidades nativas del dispositivo:

```

ionic build
ionic cap init

```

Agregar una Plataforma (Android)

```

ionic cap add android

```

Hay que tener especial cuidado con las versiones de capacitor, node etc. Puede generar muchos problemas al compilar.

- **Probar la aplicación localmente:**

Sirve para ejecutar en local e ir viendo los cambios y ediciones en el momento.

```
ionic serve
```

- **Pruebas en un emulador o dispositivo físico:**

Primero hay que sincronizar los archivos:

```
ionic build  
ionic cap sync
```

- **Sincronizar y abrir en Capacitor:**

```
ionic build  
ionic cap sync  
ionic cap open android
```

CONCLUSIÓN

Valoración Personal del Trabajo Realizado.

El proyecto **Trip and Váter App** intenta ser una solución innovadora y útil para resolver la necesidad de encontrar aseos públicos o de establecimientos de manera rápida y eficiente. Mediante el desarrollo de una aplicación híbrida con **Ionic, Angular, y Capacitor**, y el soporte de un backend basado en **Spring Boot** y **Docker**, he procurado implementar una herramienta funcional, escalable y accesible.

En el ámbito personal y profesional, el desarrollo de esta aplicación ha permitido aplicar conocimientos técnicos adquiridos durante la formación, así como intentar mejorar habilidades de análisis, planificación y resolución de problemas. Como posibles ampliaciones, se podrían integrar nuevas funcionalidades, como notificaciones, interacción entre usuarios, soporte multilenguaje, En general un sistema que haga más atractivo el uso de la aplicación y la experiencia de usuario.

Por otra parte, me gustaría señalar que, a pesar de los contratiempos que surgieron durante todo el proceso y de que me hubiese gustado alcanzar la totalidad de los objetivos planteados al inicio, creo que el camino recorrido ha sido valioso en sí mismo. Cada obstáculo, aunque a veces frustrante, me permitió aprender, adaptarme a circunstancias que no siempre pude

prevenir o tener bajo control.

Si bien no todo salió como lo había imaginado, el esfuerzo invertido y los logros obtenidos son un reflejo del compromiso y la dedicación puestos en este proyecto. A veces, no alcanzar la meta exacta nos enseña más que si el camino hubiese sido sencillo, y estoy convencido de que lo logrado hasta ahora sienta una base sólida para continuar mejorando.

BIBLIOGRAFÍA Y WEBGRAFÍA

Bibliografía:

Carlo Blé Jurado (2022). *Código Sostenible. Cómo programar código fácil de mantener.*

Webgrafía:

- Mozilla Developer Network. (2022). Documentación de HTML. <https://developer.mozilla.org/es/docs/Web/HTML>
- J. (2023). Guía Completa de Angular para Principiantes. Angular Docs. <https://angular.io/guide/quickstart>
- Human Rights Watch. (2021). *El Derecho al Saneamiento*. <https://www.hrw.org/es/news/2021/03/22/el-derecho-al-saneamiento>
- [Documentación Oficial de Ionic](#):
- [Documentación Oficial de Angular](#)
- [Documentación Oficial de Capacitor](#):
- [Documentación Oficial de Spring Boot](#):
- [Documentación Oficial de MySQL](#):

ANEXO

I. MANUAL DE USUARIO

II. CÓDIGO COMPLETO DEL PROYECTO (Enlace a GitHub)

Repositorios de Git:

Organización: <https://github.com/DAM2A-PDAM-24-25>

Backend: <https://github.com/DAM2A-PDAM-24-25/TripAndVater-backend-DanielMerlo>

FrontEnd: <https://github.com/DAM2A-PDAM-24-25/TripAndVater-frontend-DanielMerlo>

Enlaces a la APP:

Frontend:

<https://danielme25.iesmontenaranco.com/tabs/home>

OpenApi:

<https://danielme25.iesmontenaranco.com:8000/swagger-ui/index.html#/>