



Manual de Intalación

I.E.S GRAN CAPITAN

GRADO FORMATIVO DE GRADO SUPERIOR
EN APLICACIONES MULTIPLATAFORMA.

PROYECTO INTEGRADO

Autores:

- Antonio Jesús Martínez Díaz – 2º DAM
- Rocío Zuara Jiménez – 2º DAM
- Lorena Barea Rot – 2º DAM

INDICE:

1.Introducción_____	1
2.Requisitos del sistema_____	1
3.Instalación_____	1

1.INTRODUCCION:

Lavadero Sepúlveda es un sistema integral y multiplataforma diseñado para la gestión completa de citas en lavaderos de vehículos. Este manual proporciona instrucciones detalladas de un instalación y configuración.

2. REQUISITOS PREVIOS:

Aplicación Escritorio:

- **Sistema Operativo:** Windows 10 o superior o Linux
- **Procesador:** 1 GHZ o superior
- **Memoria:** 2 GB o superior
- **Java 17**
- **Programas para crear BD y lanzar la APP:** MySQL Workbench | IDE (IntelliJ, Eclipse, VS Code...)

Aplicación Móvil:

- **Sistema Operativo:** Android
- **Conexión a internet** para sincronizar datos.
- **Programas para emular y crear el puente:** Android Studio y Ngrok

3. INSTALACIÓN:

Escritorio:

1.Crear la base de datos en MySQL Workbench:

Antes de arrancar la aplicación, es necesario crear una base de datos vacía. No se deben crear tablas manualmente: la aplicación al lanzarla se encargará de eso.

- 1.Abrir MySQL Workbench.
- 2.Conectarse al servidor MySQL local utilizando tu usuario (por ejemplo, root).
- 3.Abrir una nueva pestaña de consultas (Query).
- 4.Ejecutar el siguiente comando SQL:

```
CREATE DATABASE lavadero_db;
```

Verificar que ahora aparece el esquema lavadero_db en la lista de bases de datos.

2. Arrancar la aplicación desde un IDE con (Spring Boot):

Una vez creada la base de datos, se procede a abrir el proyecto y pulsar el botón Run, para lanzar la aplicación.

Aplicación Móvil:

La aplicación móvil necesita conectarse al servidor backend desarrollado en Spring Boot para poder funcionar (login, lectura de datos, reservas, etc.).

Cuando el backend está instalado en un entorno local —es decir, ejecutándose en tu propio ordenador— el móvil no puede acceder directamente a **http://localhost:8080**, porque esa dirección solo es accesible desde el mismo equipo.

Para solucionar esto, utilizamos ngrok, una herramienta que crea un túnel seguro desde Internet hasta tu máquina local, permitiendo que la app móvil acceda al backend como si estuviera en un servidor real.

```
ngrok
👉 Put your secrets in vaults and (re)use them to transform traffic: https://ngrok.com/r/secrets

Session Status      online
Account             Antonio Jesus Martinez (Plan: Free)
Version             3.33.1
Region              Europe (eu)
Latency             77ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://goosewinged-quinton-petaloid.ngrok-free.dev -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p90
                   24     0     0.05   0.03   8.22   92.70

HTTP Requests
-----
17:44:09.371 CET GET /css/styles.css          200
17:44:09.439 CET GET /images/llanta_sucia.jpg      200
17:44:09.402 CET GET /images/llanta_limpia.jpg     200
17:44:09.444 CET GET /images/limpio.gif          200
17:44:09.485 CET GET /images/sucio.gif           200
17:44:09.484 CET GET /images/limpio2.gif         200
17:44:09.444 CET GET /images/sucio2.gif          200
17:44:09.395 CET GET /images/Logo.jpg            200
17:44:08.901 CET GET /galeria                    200
17:44:06.652 CET GET /images/Logo.jpg            200
```

1. Lanzar aplicación:

- Tener el backend levantado con Spring Boot, escuchando normalmente en `http://localhost:8080/`.

2. Crear la URL pública temporal con ngrok:

1. Abrir una terminal y ejecutar:

ngrok http 8080

2. ngrok mostrará una URL en pantalla, por ejemplo:

<https://goosewinged-quinton-petaloid.ngrok-free.dev>

Esa será la dirección que usará la app móvil para comunicarse con tu backend.

3. Configurar la URL ngrok dentro de la app móvil

Cada vez que generes una URL nueva con ngrok, debes actualizarla en dos archivos del proyecto Android:

3.1 RetrofitClient.kt

Aquí se definen las URLs para los distintos entornos:

```
// URLs para diferentes entornos
private const val NGROK_URL = "https://goosewinged-quinton-petaloid.ngrok-free.dev"
private const val PRODUCTION_URL = "http://lavaderosepulveda.servehttp.com:8080/"
```

3.2 MainActivity.kt

En el arranque de la app se configura qué URL se usará

```
private fun configurarEntornoInicial() {
    val ngrokUrl = "https://goosewinged-quinton-petaloid.ngrok-free.dev"
```

4. Ejecutar la app móvil en Android Studio

Una vez cambiadas las URLs:

1. Abrir Android Studio y seleccionar Run > Run app
2. Elegir un emulador o un teléfono conectado por USB

La app se iniciará en modo Debug, utilizando la URL de ngrok y pudiendo comunicarse correctamente con el backend local.

Acceso a la Aplicación de Escritorio mediante Ngrok:

Para permitir el acceso externo a la aplicación de escritorio cuando el backend se ejecuta en un entorno local, se utiliza un túnel seguro proporcionado por la herramienta ngrok.

1. Tras tener ejecutando la aplicación desarrollada en Spring Boot en su el entorno local, abrimos el terminal o símbolo del sistema y ejecutamos el comando para iniciar el túnel con ngrok y obtener la Url publica:

ngrok http 8080

2. Acceda a la aplicación de escritorio introduciendo dicha URL en cualquier navegador web compatible. Teniendo en cuenta que la URL generada por ngrok es temporal y variará cada vez que se reinicie el túnel. Por lo que es necesario obtener y utilizar la nueva URL en cada sesión.