

UAX FP
Universidad
Alfonso X el Sabio

Cuadro de Mando Interactivo para una Clínica Dental
2º Administración de Sistemas Informáticos en Red Presencial

Conesa de Quadros, Guillermo
Tutor del TFG

DEDICATORIA (OPCIONAL)

ÍNDICES

Instalación y Configuración de WSL con Ubuntu 24.04.1 LTS.....	20
Habilitar WSL en Windows.....	20
Instalar Ubuntu 24.04.1 LTS desde Microsoft Store	21
Crear Usuario y Configuración Básica	21
Cambiar el Nombre del Equipo.....	21
Verificar la Versión del Kernel.....	22
Verificar Arquitectura y CPU	22
Verificar Versión de Ubuntu.....	22
Instalación y Configuración de MySQL	23
Actualizar el sistema	23
Instalar MySQL Server.....	23
Verificar la instalación.....	23
Iniciar y Habilitar MySQL.....	23
Acceso a MySQL y Creación del Usuario.....	24
Iniciar sesión en MySQL	24
Crear Usuario MySQL para ConeDental:.....	24
Crear la Base de Datos.....	24
Acceder a la Base de Datos.....	25
Permitir Acceso Remoto a MySQL	25
Creación de la Base de Datos en MySQL	26
Creación de las Tablas.....	26
Clinicas.....	26
Pacientes.....	26
Doctores	26
Material Dental	27
Prótesis Dentales	27
Empleados.....	27
Proveedores Dentales.....	28
Tratamientos	28

Medicamentos	28
Tratamientos_Medicamentos	28
Pacientes_Medicamentos	29
Citas	29
Verificación de la Base de Datos	30
Inserción de Datos en la Base de Datos	31
Clínicas	31
Pacientes	31
Material Dental	32
Prótesis Dentales	33
Empleados	33
Proveedores Dentales	34
Instalación de Python y Virtual Environment	34
Crear un entorno virtual	34
Instalación de Flask y -CORS	34
Desarrollo de la API con Flask y Flask-CORS	35
Creación el archivo app.py	35
Ejecutar la API Flask	37
Activar el entorno virtual	37
Iniciar la API	37
Probar la API	38
Código HTML	39
Estructura Head	39
Bootstrap y Chart.js	39
Sidebar de Navegación	39
Tarjetas de Resumen	40
Gráficos con Chart.js	41
Tablas Interactivas	41
Conexión con JavaScript	41
CSS	42
Estilos Generales	42
Sidebar	42

Contenido Principal	42
Estilos de Tarjetas	42
Estilos de Gráficos	43
Estilos de Tablas	43
Código JavaScript	44
Cargar Datos al Iniciar la Página	44
Obtener Datos de la API	44
Cargar Información en las Tarjetas	44
Gráfico de Citas por Doctores	44
Gráfico de Pacientes por Tratamientos	45
Citas	45
Pacientes en la Página	46
Gráficos de Medicamentos por Pacientes y Tratamientos	46
Gráfico de barras con la cantidad de medicamentos por paciente	46

ABSTRACT

Este documento presenta el desarrollo de un Cuadro de Mando Interactivo para la gestión de la clínica dental ConeDental. El sistema está basado en una arquitectura cliente-servidor, con un backend en Flask y MySQL que proporciona una API para gestionar clínicas, pacientes, doctores, citas, materiales y otros datos clave. El frontend, desarrollado con HTML, CSS y JavaScript, ofrece una interfaz visual con gráficos dinámicos, tablas interactivas y alertas automáticas para optimizar la toma de decisiones.

El sistema incluye un mecanismo de recopilación automática de datos y notificaciones inteligentes sobre citas, stock de materiales y disponibilidad del personal. Su diseño escalable y personalizable permite adaptarlo a diferentes necesidades de la gestión clínica.

This document presents the development of an Interactive Dashboard for managing the ConeDental clinic. The system is based on a client-server architecture, with a Flask and MySQL backend providing an API to manage clinics, patients, doctors, appointments, materials, and other key data. The frontend, built with HTML, CSS, and JavaScript, offers a visual interface with dynamic charts, interactive tables, and automated alerts to enhance decision-making.

The system includes an automated data collection mechanism and smart notifications for appointments, material stock, and staff availability. Its scalable and customizable design allows adaptation to various needs in clinic management in the dental sector

JUSTIFICACIÓN DEL PROYECTO

En el ámbito odontológico, una gestión eficiente de una clínica es fundamental para garantizar un servicio de calidad a los pacientes y optimizar los recursos disponibles. Sin embargo, muchas clínicas aún dependen de métodos tradicionales, como registros en papel o sistemas informáticos desactualizados, lo que dificulta la organización de citas, el control de stock de materiales y la supervisión del personal. La motivación principal de este proyecto es desarrollar un Cuadro de Mando Interactivo que permita centralizar y visualizar en tiempo real toda la información relevante de una clínica dental, facilitando la toma de decisiones basada en datos.

Actualmente, existen soluciones comerciales para la gestión de clínicas dentales, que ofrecen funciones similares como Grafana. No obstante, estas aplicaciones suelen ser costosas, poco personalizables y requieren licencias de uso, lo que puede limitar su accesibilidad para pequeñas y medianas clínicas. A diferencia de estas herramientas, el Cuadro de Mando Interactivo de ConeDental está diseñado para ser una solución escalable y adaptable, permitiendo una integración completa con la base de datos de la clínica y personalización según sus necesidades específicas.

El sistema está dirigido a clínicas dentales y profesionales de la odontología que buscan mejorar la gestión de su práctica a través de una plataforma moderna y accesible. Al ser un sistema interactivo con alertas, facilita la identificación de problemas en la operativa diaria, como la falta de materiales o la sobrecarga de citas, contribuyendo a una administración más eficiente. En comparación con otras soluciones existentes, este proyecto destaca por su enfoque en la automatización, visualización intuitiva y accesibilidad sin costos de licencia, lo que lo convierte en una alternativa viable y competitiva para clínicas en crecimiento.

INTRODUCCIÓN

El Cuadro de Mando Interactivo de ConeDental optimiza la gestión clínica con un sistema centralizado y automatizado. Entre sus principales funciones destacan:

Gestión de Clinicas, Pacientes, Doctores, etc: Permite administrar datos de manera estructurada y accesible.

Control de Citas: Visualización y organización eficiente del calendario de consultas.

Supervisión de Inventario: Monitorea materiales dentales y medicamentos con alertas de stock bajo.

Visualización de Datos en Tiempo Real: Usa gráficos dinámicos y tablas interactivas para facilitar la toma de decisiones.

Automatización de Alertas: Notificaciones sobre eventos críticos como citas próximas o escasez de insumos.

Para un funcionamiento óptimo, el sistema debe cumplir con los siguientes requisitos:

Base de Datos MySQL con estructuras optimizadas para gestionar clínicas, citas y materiales.

Backend en Flask (Python) con una API RESTful para procesar y servir datos.

Frontend en HTML, CSS y JavaScript, con fetch() para actualizar información en tiempo real.

Gráficos en Chart.js para representación visual de estadísticas clave.

Sistema de Alertas Automáticas para notificar sobre eventos relevantes.

OBJETIVOS

R01 – El sistema debe ejecutarse en Ubuntu

- R01F01 – El servidor debe estar instalado y configurado en Ubuntu.
 - R01F01T01 – Instalar Ubuntu Server y configurar dependencias necesarias (Python, MySQL, Flask, etc.).
 - R01F01T01P01 – Verificar la instalación ejecutando `python3 --version` y `mysql --version`.
- R01F02 – El sistema debe iniciar automáticamente al arrancar Ubuntu.
 - R01F02T01 – Configurar systemd para ejecutar Flask como un servicio.
 - R01F02T01P01 – Reiniciar el servidor y comprobar que el servicio está activo con `systemctl status flaskapp`.

R02 – La base de datos debe estar implementada en MySQL

- R02F01 – La base de datos debe contener las tablas necesarias para la gestión de la clínica.
 - R02F01T01 – Crear las tablas clínicas, pacientes, doctores, citas, materiales, medicamentos, proveedores, etc.
 - R02F01T01P01 – Ejecutar `SHOW TABLES;` en MySQL para verificar la creación de las tablas.
- R02F02 – Se debe garantizar la integridad y relaciones entre los datos.
 - R02F02T01 – Definir claves primarias, foráneas y restricciones adecuadas.
 - R02F02T01P01 – Insertar datos de prueba y verificar restricciones (`INSERT`, `SELECT`, `DELETE`).

R03 – El backend debe estar desarrollado en Flask

- R03F01 – Implementar una API RESTful con Flask.
 - R03F01T01 – Crear endpoints para CRUD de clínicas, pacientes, doctores, citas, materiales y medicamentos.
 - R03F01T01P01 – Probar los endpoints con Postman (GET, POST, PUT, DELETE).
- R03F02 – Gestionar la autenticación con tokens JWT.

- R03F02T01 – Implementar login y generación de tokens JWT.
- R03F02T01P01 – Hacer una solicitud con y sin token para verificar el acceso.

R04 – Implementación de CORS para permitir el acceso al backend

- R04F01 – Permitir solicitudes desde el frontend al backend.
 - R04F01T01 – Configurar flask-cors para permitir el acceso desde distintos orígenes.
 - R04F01T01P01 – Hacer una solicitud fetch() desde JavaScript y verificar que no hay errores de CORS.
- R04F02 – Restringir el acceso a dominios específicos si es necesario.
 - R04F02T01 – Configurar Flask-CORS para aceptar solo solicitudes desde un dominio autorizado.
 - R04F02T01P01 – Intentar hacer una solicitud desde un dominio no permitido y verificar el bloqueo.

R05 – Desarrollo del frontend en HTML, CSS y JavaScript

- R05F01 – Diseñar una interfaz responsive para la gestión de la clínica.
 - R05F01T01 – Crear una página principal con navegación en HTML y CSS.
 - R05F01T01P01 – Abrir index.html en un navegador y comprobar la carga correcta de estilos y estructura.
- R05F02 – Implementar la conexión con el backend usando JavaScript (fetch()).
 - R05F02T01 – Obtener y mostrar datos dinámicamente desde la API Flask.
 - R05F02T01P01 – Inspeccionar la consola del navegador para verificar la respuesta de la API.

R06 – Visualización de datos con Chart.js

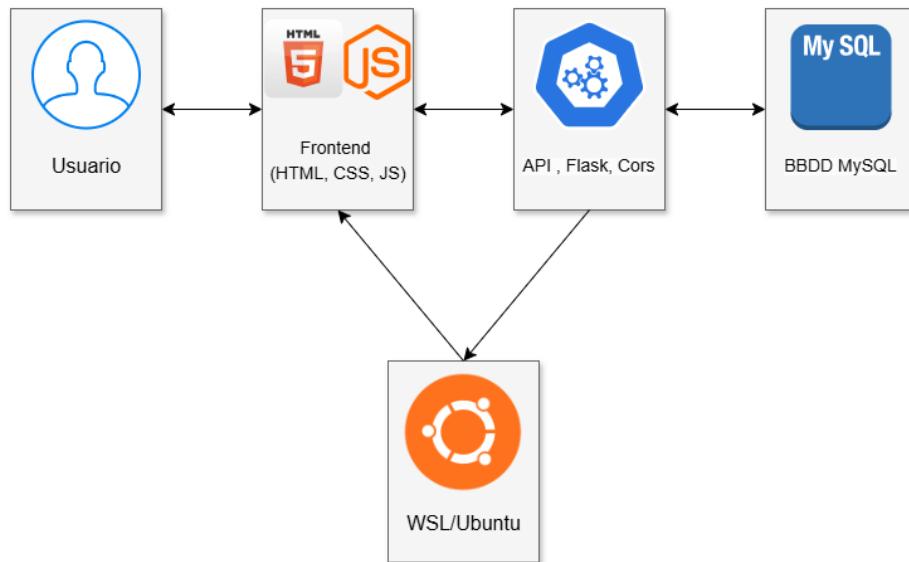
- R06F01 – Representar estadísticas con gráficos dinámicos.
 - R06F01T01 – Implementar gráficos para citas por doctor y pacientes por tratamiento.
 - R06F01T01P01 – Comparar los datos mostrados en los gráficos con los valores en la base de datos.
- R06F02 – Permitir la actualización dinámica de los gráficos.

- R06F02T01 – Actualizar gráficos en tiempo real al recibir nuevos datos.
- R06F02T01P01 – Simular la creación de una nueva cita y comprobar la actualización del gráfico.

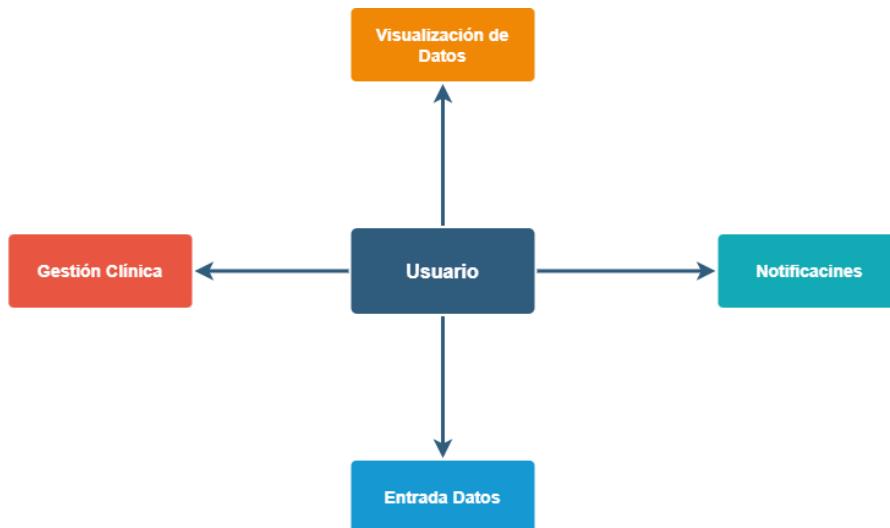
DESCRIPCIÓN

Arquitectura

1. **El usuario** interactúa con la interfaz web (Frontend en HTML, CSS y JavaScript).
2. **El frontend** envía solicitudes a la **API Flask (Backend)** para obtener y gestionar datos.
3. **El backend en Flask** procesa las solicitudes y accede a la **Base de Datos MySQL** para recuperar o actualizar información.
4. **El servidor Ubuntu** aloja el backend y la base de datos, asegurando la conectividad y procesamiento.
5. **El frontend** recibe los datos procesados y actualiza la interfaz en tiempo real, mostrando información y gráficos dinámicos.



Casos de uso.

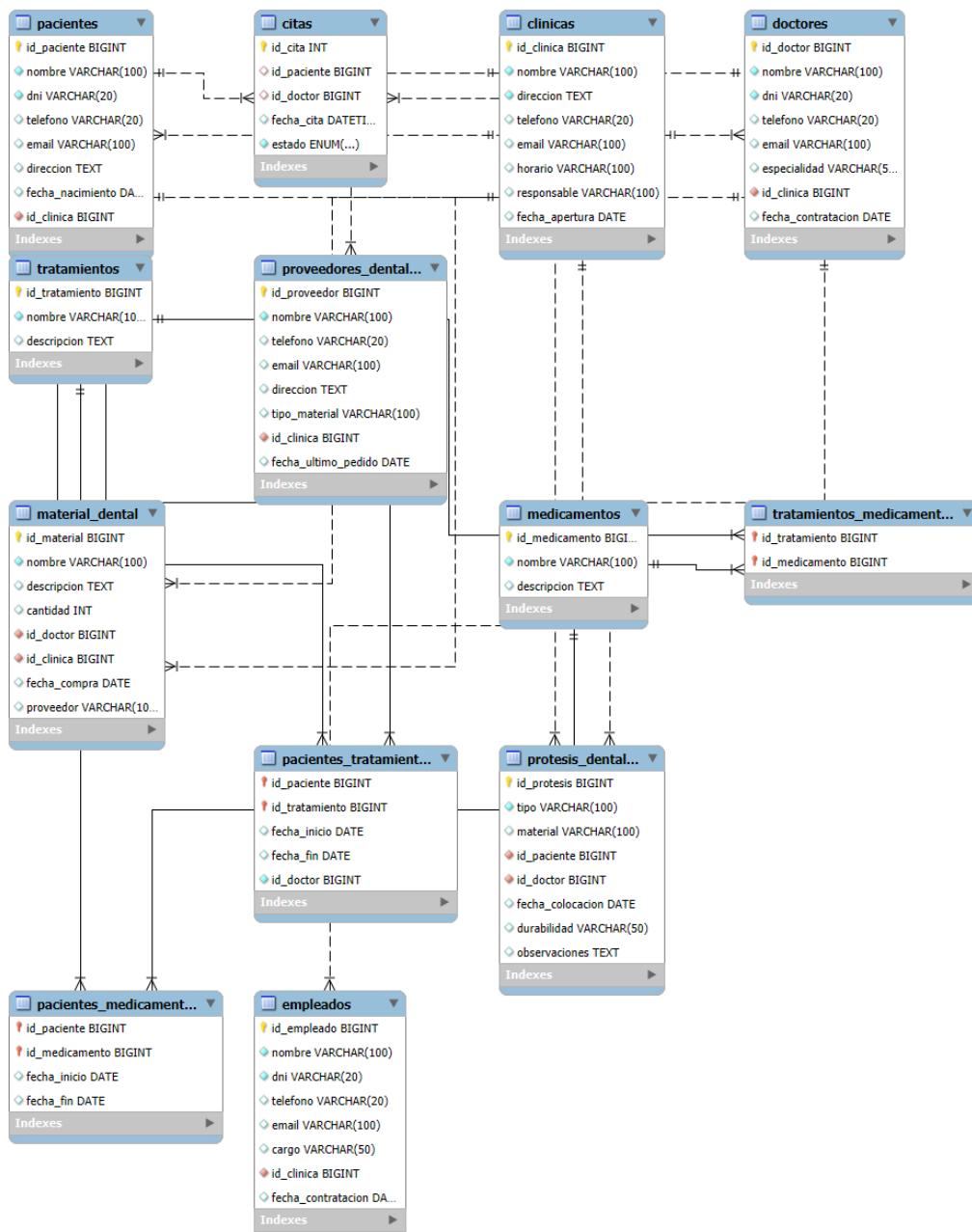


Categoría	Usuario	Visualización de Datos	Notificaciones	Entrada de Datos
Descripción	Autenticación	Dashboard	Alertas automáticas	Registro de Información
Funcionalidad	El usuario inicia sesión en el sistema.	Muestra estadísticas con gráficos dinámicos.	Genera alertas sobre citas y stock bajo.	Permite ingresar nuevos datos en el sistema.
Precondiciones	Usuario registrado en la BD.	Usuario autenticado.	Usuario autenticado.	Usuario autenticado.
Postcondiciones	Acceso concedido o denegado.	Datos cargados y mostrados.	Notificación enviada.	Datos guardados correctamente.
Datos de Entrada	Usuario, contraseña.	Solicitudes al backend.	Eventos críticos detectados.	Información del paciente, cita, material, etc.
Datos de Salida	Mensaje de éxito o error.	Gráficos generados.	Mensaje de alerta.	Confirmación de operación.

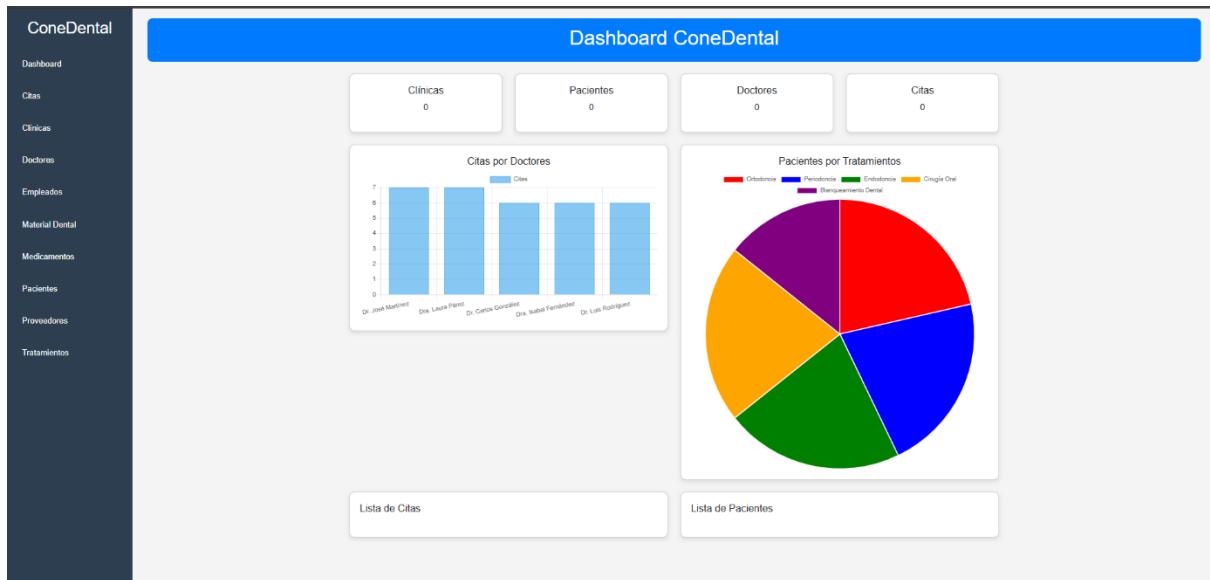
Tablas	usuarios	N/A	citas, materiales, pacientes	Varios (dependiendo del módulo)
Interfaces	LoginInterface	DashboardInterface	AlertaInterface	FormularioInterface

DISEÑOS

Diagrama de la base de datos. Con detalle de campos.



Interfaces. Interesa ver la solución en diferentes tamaños o dispositivos.



TECNOLOGÍA



Ubuntu

Sistema operativo basado en Linux, utilizado para servidores y entornos de desarrollo.

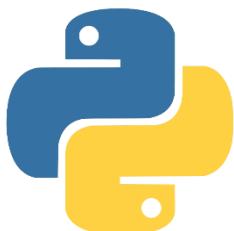
Se utiliza como la plataforma base para alojar el servidor web, la base de datos y la aplicación Flask.



MySQL

Sistema de gestión de bases de datos relacional (RDBMS) que permite el almacenamiento y manejo eficiente de datos

Se usa para gestionar la información de la clínica dental, incluyendo pacientes, doctores, citas, materiales y proveedores.



Python - Flask

Framework web ligero para Python que facilita la creación de aplicaciones y APIs.

Se emplea para desarrollar el backend de la aplicación, gestionar peticiones HTTP y conectar con la base de datos MySQL.



Flask-CORS

Extensión para Flask que permite compartir recursos entre orígenes diferentes (Cross-Origin Resource Sharing)

Se implementa para permitir que el frontend (HTML, CSS, JavaScript) pueda comunicarse con el backend sin restricciones de seguridad.



HTML, CSS y JavaScript

Tecnologías estándar para la creación de interfaces web interactivas y visualmente atractivas.

HTML estructura la interfaz de usuario.
CSS estiliza la interfaz para mejorar la experiencia del usuario.
JavaScript permite la interacción dinámica con el backend mediante **fetch()**.



Chart.js

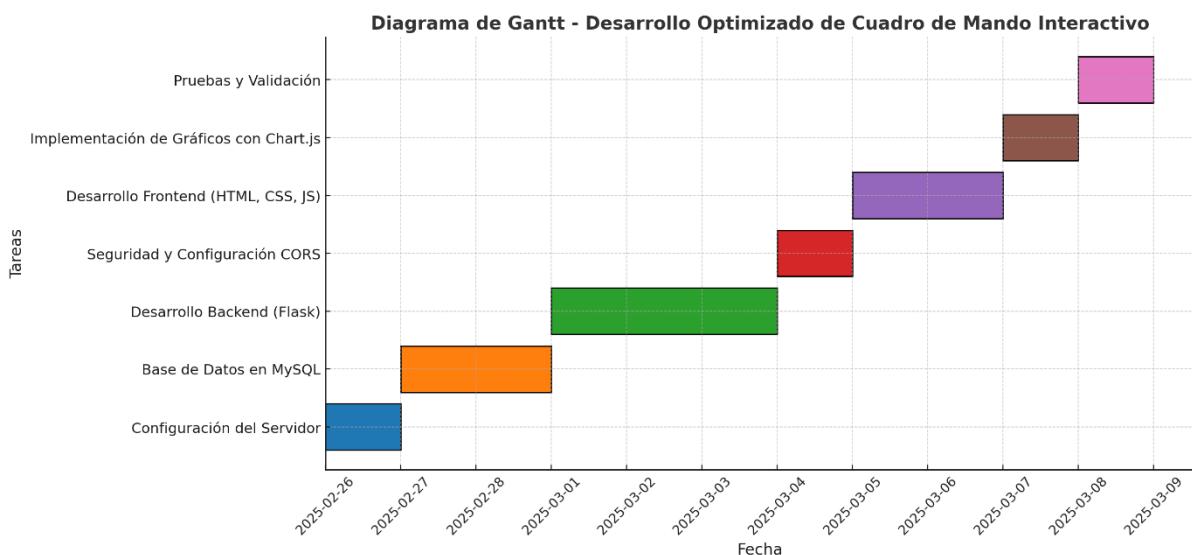
Biblioteca de JavaScript para la visualización de datos mediante gráficos interactivos.

Se utiliza para representar estadísticas clave en el cuadro de mando, como citas por doctor y pacientes por tratamiento.

METODOLOGÍA

Considerando una tarifa de 40€ por hora, el costo total estimado se reduce a 2.400€.

- Configuración del servidor Ubuntu y dependencias en 7 horas, incluyendo la instalación de los paquetes necesarios y la automatización del servicio.
- Implementación de la base de datos en MySQL en 10 horas, con la creación de tablas optimizadas y definición de relaciones.
- Desarrollo del backend con Flask y API RESTful en 13 horas, enfocándose en endpoints eficientes para la gestión de la clínica.
- Integración de seguridad con JWT y configuración de CORS en 6 horas, garantizando el acceso seguro al sistema.
- Desarrollo del frontend en HTML, CSS y JavaScript en 12 horas, asegurando una interfaz intuitiva y responsive.
- Implementación de gráficos dinámicos con Chart.js en 9 horas, facilitando la visualización de datos en tiempo real.
- Pruebas y validación del sistema en 3 horas, verificando el correcto funcionamiento de cada módulo.



README y GIT.

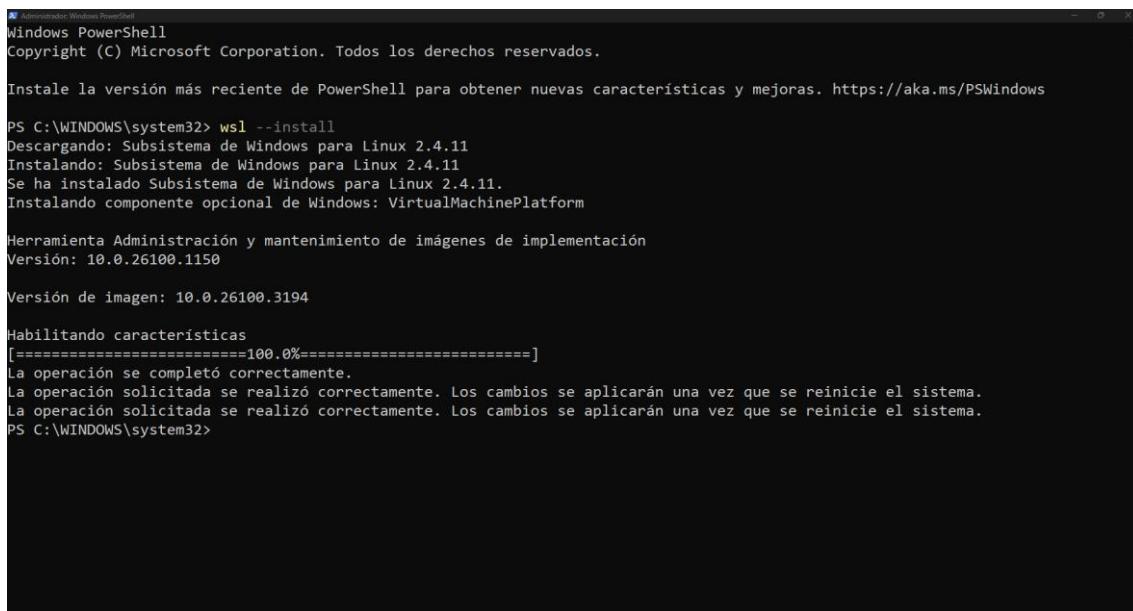
<https://github.com/guillecone/conederal.git>

Instalación y Configuración de WSL con Ubuntu 24.04.1 LTS

Habilitar WSL en Windows

PowerShell como administrador.

```
wsl -install
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install
Descargando: Subsistema de Windows para Linux 2.4.11
Instalando: Subsistema de Windows para Linux 2.4.11
Se ha instalado Subsistema de Windows para Linux 2.4.11.
Instalando componente opcional de Windows: VirtualMachinePlatform

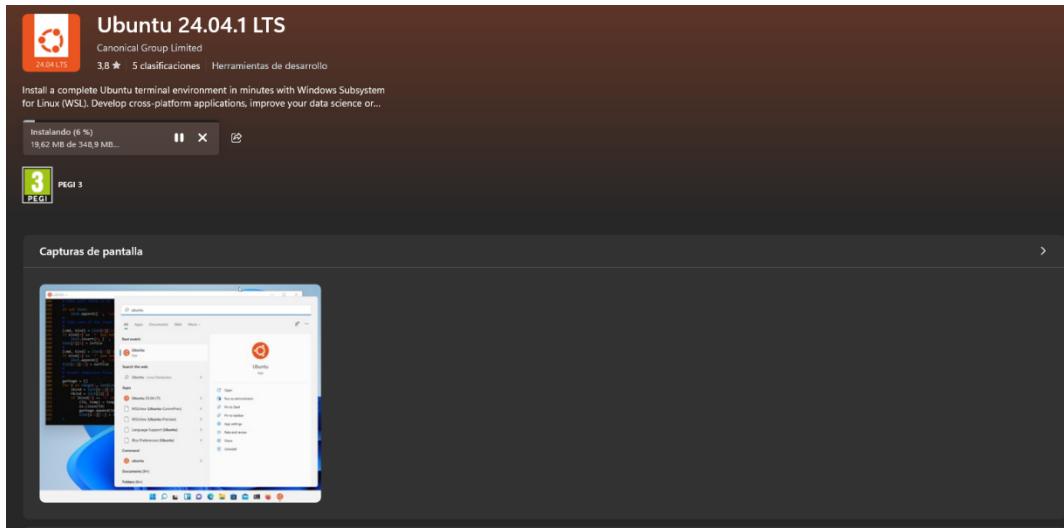
Herramienta Administración y mantenimiento de imágenes de implementación
Versión: 10.0.26100.1150

Versión de imagen: 10.0.26100.3194

Habilitando características
[=====100.0%=====]
La operación se completó correctamente.
La operación solicitada se realizó correctamente. Los cambios se aplicarán una vez que se reinicie el sistema.
La operación solicitada se realizó correctamente. Los cambios se aplicarán una vez que se reinicie el sistema.
PS C:\WINDOWS\system32>
```

Instalar Ubuntu 24.04.1 LTS desde Microsoft Store

- Microsoft Store en Windows.
- Ubuntu 24.04.1 LTS.



Crear Usuario y Configuración Básica

- **Nombre de usuario:** admin_conedental
- **Contraseña:** Pa\$\$w0rd

Cambiar el Nombre del Equipo

```
sudo hostnamectl set-hostname PCconederal
```

```
admin_conederal@PCconederal:~$ whoami
admin_conederal
admin_conederal@PCconederal:~$
```

Verificar la Versión del Kernel

```
uname -r
5.15.167.4-microsoft-standard-WSL2
```

Verificar Arquitectura y CPU

lscpu

```
Arquitectura: x86_64
Modos de operación: 32-bit, 64-bit
Núcleos físicos: 4
Hilos totales: 8
Modelo CPU: 11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz
```

Verificar Versión de Ubuntu

```
lsb_release -a
Sistema Operativo: Ubuntu 24.04.1 LTS (Noble Numbat)
```

```
admin_condenatal@PCcondenatal: ~
5.15.167.4-microsoft-standard-WSL2
admin_condenatal@PCcondenatal: ~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         39 bits physical, 48 bits virtual
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:  0-7
Vendor ID:             GenuineIntel
Model name:            11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz
CPU Family:            6
Model:                 140
Thread(s) per core:   2
Core(s) per socket:   4
Socket(s):            1
Stepping:              2
BogoMIPS:              4992.02
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
                        pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm
                        constant_tsc arch_perfmon rep_good nopl xtTopology tsc_reliable no
                        nstop_tsc cpuid pn1 pclmu1gdd1 vms sse3 fma cx16 pdcm pcid sse4_1
                        sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c r
                        drand hypervisor lahf.lm abm 3dnowprefetch invpcid_single ssbd ibr
                        s ibpb stibp ibrs enhanced tpr_shadow vmmi ept vpid ept_ad fsgsbas
                        e tsc_adjust bni1 avx2 smep bmi2 erms invpcid avx512f avx512dq rds
                        eed adv smap avx512fma clflushopt clwb avx512cd sha_ni avx512bw a
                        vx512vl xsavesopt xsaves xgetbv1 xsaves avx512vmbi umip avx512_vbmi
                        2 gfn1 waes vpclmulqdq avx512_vnni avx512_bitalg avx512_vpocndq
                        rdpid movdir64b fsrm avx512_vp2intersect md_clear flush_ll
                        d arch_capabilities

Virtualization Features:
Virtualization:          VT-x
Hypervisor vendor:       Microsoft
Virtualization type:     full
Caches (sum of all):
L1d:                    192 KiB (4 instances)
L1i:                    128 KiB (4 instances)
L2:                     5 MiB (4 instances)
L3:                     8 MiB (1 instance)
Vulnerabilities:
Gather data sampling:   Not affected
Itnlb multithit:        Not affected
L1tf:                   Not affected
Mdts:                   Not affected
Meltdown:               Not affected
Mmio stale data:        Not affected
Reg file data sampling: Not affected
Retbleed:                Mitigation; Enhanced IBRS
Spec rstack overflow:   Not affected
Spec store bypass:       Mitigation; Speculative Store Bypass disabled via prctl and seccomp
                        p
Spectre v1:              Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Spectre v2:              Mitigation; Enhanced / Automatic IBRS; IBPB conditional; RSB filling; PDRSB-eIBRS SW sequence; BHI SW loop; KVM SW loop
Srbds:                  Not affected
Txn async abort:         Not affected
admin_condenatal@PCcondenatal: ~
```

Instalación y Configuración de MySQL

Actualizar el sistema

```
sudo apt update && sudo apt upgrade -y
```

Instalar MySQL Server

```
sudo apt install mysql-server -y
```

Verificar la instalación

```
mysql -version
mysql Ver 8.0.41-0ubuntu0.24.04.1 for Linux on x86_64 ((Ubuntu))
```

Iniciar y Habilitar MySQL

```
sudo systemctl start mysql
sudo systemctl enable mysql
sudo systemctl status mysql
```

```
admin_condenatal@PCcondenatal: $ sudo systemctl status mysql
● mysql.service - MySQL Community Server
  Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset:)
  Active: active (running) since Thu 2025-03-06 12:29:17 CET; 22h ago
    Main PID: 156258 (mysqld)
   Status: "Server is operational"
     Tasks: 51 (limit: 9373)
    Memory: 448.3M
      CGroup: /system.slice/mysql.service
              └─156258 /usr/sbin/mysqld

Mar 06 12:29:16 PCcondenatal systemd[1]: Starting mysql.service - MySQL Comm...
Mar 06 12:29:17 PCcondenatal systemd[1]: Started mysql.service - MySQL Comm...
Lines 1-12/12 (END) .. skipping...
● mysql.service - MySQL Community Server
  Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enab...
  Active: active (running) since Thu 2025-03-06 12:29:17 CET; 22h ago
    Main PID: 156258 (mysqld)
   Status: "Server is operational"
     Tasks: 51 (limit: 9373)
    Memory: 448.3M
      CGroup: /system.slice/mysql.service
              └─156258 /usr/sbin/mysqld

Mar 06 12:29:16 PCcondenatal systemd[1]: Starting mysql.service - MySQL Community S...
Mar 06 12:29:17 PCcondenatal systemd[1]: Started mysql.servic...
e - MySQL Community Se...
```

Acceso a MySQL y Creación del Usuario

Iniciar sesión en MySQL

```
sudo mysql -u root -p
```

Crear Usuario MySQL para ConeDental:

```
CREATE USER 'admin_conederal'@'localhost' IDENTIFIED BY 'Pa$$w0rd';
GRANT ALL PRIVILEGES ON *.* TO 'admin_conederal'@'localhost' WITH GRANT
OPTION;
FLUSH PRIVILEGES;
```

Crear la Base de Datos

```
CREATE DATABASE conederal;
SHOW DATABASES;
```

```
admin_conederal@PCconederal:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 554
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| conederal |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)
```

Acceder a la Base de Datos

```
USE conedental;
```

Permitir Acceso Remoto a MySQL

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Modificamos la línea:

```
bind-address = 0.0.0.0
```

```
GNU nano 7.2                               /etc/mysql/mysql.conf.d/mysqld.cnf
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user          = mysql
# pid-file     = /var/run/mysqld/mysqld.pid
# socket       = /var/run/mysqld/mysqld.sock
# port         = 3306
# datadir      = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir        = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address      = 0.0.0.0
#
# * Fine Tuning
#
key_buffer_size   = 16M
# max_allowed_packet = 64M
# thread_stack      = 256K
# thread_cache_size = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options = BACKUP

# max_connections     = 151
# table_open_cache    = 4000
```

Otorgar privilegios de acceso remoto al usuario

```
GRANT ALL PRIVILEGES ON *.* TO 'admin_conedental'@'%' IDENTIFIED BY
'Pa$$w0rd' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Creación de la Base de Datos en MySQL

Creación de las Tablas

Clinicas

```
CREATE TABLE clinicas (
    id_clinica BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    direccion TEXT NOT NULL,
    telefono VARCHAR(20),
    email VARCHAR(100),
    horario VARCHAR(100),
    responsable VARCHAR(100),
    fecha_apertura DATE
);
```

Pacientes

```
CREATE TABLE pacientes (
    id_paciente BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    dni VARCHAR(20) UNIQUE NOT NULL,
    telefono VARCHAR(20),
    email VARCHAR(100),
    direccion TEXT,
    fecha_nacimiento DATE,
    id_clinica BIGINT UNSIGNED NOT NULL,
    FOREIGN KEY (id_clinica) REFERENCES clinicas(id_clinica) ON DELETE CASCADE
);
```

Doctores

```
CREATE TABLE doctores (
    id_doctor BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    dni VARCHAR(20) UNIQUE NOT NULL,
    telefono VARCHAR(20),
    email VARCHAR(100),
    especialidad VARCHAR(50),
    id_clinica BIGINT UNSIGNED NOT NULL,
    fecha_contratacion DATE,
    FOREIGN KEY (id_clinica) REFERENCES clinicas(id_clinica) ON DELETE CASCADE
);
```

Material Dental

```
CREATE TABLE material_dental (
    id_material BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT,
    cantidad INT CHECK (cantidad >= 0),
    id_doctor BIGINT UNSIGNED NOT NULL,
    id_clinica BIGINT UNSIGNED NOT NULL,
    fecha_compra DATE,
    proveedor VARCHAR(100),
    FOREIGN KEY (id_doctor) REFERENCES doctores(id_doctor) ON DELETE CASCADE,
    FOREIGN KEY (id_clinica) REFERENCES clinicas(id_clinica) ON DELETE CASCADE
);
```

Prótesis Dentales

```
CREATE TABLE protesis_dentales (
    id_protesis BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    tipo VARCHAR(100) NOT NULL,
    material VARCHAR(100),
    id_paciente BIGINT UNSIGNED NOT NULL,
    id_doctor BIGINT UNSIGNED NOT NULL,
    fecha_colocacion DATE,
    durabilidad VARCHAR(50),
    observaciones TEXT,
    FOREIGN KEY (id_paciente) REFERENCES pacientes(id_paciente) ON DELETE CASCADE,
    FOREIGN KEY (id_doctor) REFERENCES doctores(id_doctor) ON DELETE CASCADE
);
```

Empleados

```
CREATE TABLE empleados (
    id_empleado BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    dni VARCHAR(20) UNIQUE NOT NULL,
    telefono VARCHAR(20),
    email VARCHAR(100),
    cargo VARCHAR(50),
    id_clinica BIGINT UNSIGNED NOT NULL,
    fecha_contratacion DATE,
    FOREIGN KEY (id_clinica) REFERENCES clinicas(id_clinica) ON DELETE CASCADE
);
```

Proveedores Dentales

```
CREATE TABLE proveedores_dentales (
    id_proveedor BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    telefono VARCHAR(20),
    email VARCHAR(100),
    direccion TEXT,
    tipo_material VARCHAR(100),
    id_clinica BIGINT UNSIGNED NOT NULL,
    fecha_ultimo_pedido DATE,
    FOREIGN KEY (id_clinica) REFERENCES clinicas(id_clinica) ON DELETE
CASCADE
);
```

Tratamientos

```
CREATE TABLE tratamientos (
    id_tratamiento BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT
);
```

Medicamentos

```
CREATE TABLE medicamentos (
    id_medicamento BIGINT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT
);
```

Tratamientos_Medicamentos

```
CREATE TABLE tratamientos_medicamentos (
    id_tratamiento BIGINT UNSIGNED NOT NULL,
    id_medicamento BIGINT UNSIGNED NOT NULL,
    PRIMARY KEY (id_tratamiento, id_medicamento),
    FOREIGN KEY (id_tratamiento) REFERENCES tratamientos(id_tratamiento) ON
DELETE CASCADE,
    FOREIGN KEY (id_medicamento) REFERENCES medicamentos(id_medicamento) ON
DELETE CASCADE
);
```

Pacientes_Tratamientos

```
CREATE TABLE pacientes_tratamientos (
    id_paciente BIGINT UNSIGNED NOT NULL,
    id_tratamiento BIGINT UNSIGNED NOT NULL,
    fecha_inicio DATE,
    fecha_fin DATE,
    PRIMARY KEY (id_paciente, id_tratamiento),
    FOREIGN KEY (id_paciente) REFERENCES pacientes(id_paciente) ON DELETE
CASCADE,
    FOREIGN KEY (id_tratamiento) REFERENCES tratamientos(id_tratamiento) ON
DELETE CASCADE
);
```

Pacientes_Medicamentos

```
CREATE TABLE pacientes_medicamentos (
    id_paciente BIGINT UNSIGNED NOT NULL,
    id_medicamento BIGINT UNSIGNED NOT NULL,
    fecha_inicio DATE,
    fecha_fin DATE,
    PRIMARY KEY (id_paciente, id_medicamento),
    FOREIGN KEY (id_paciente) REFERENCES pacientes(id_paciente) ON DELETE
CASCADE,
    FOREIGN KEY (id_medicamento) REFERENCES medicamentos(id_medicamento) ON
DELETE CASCADE
);
```

Citas

```
CREATE TABLE citas (
    id_cita INT AUTO_INCREMENT PRIMARY KEY,
    id_paciente BIGINT UNSIGNED,
    id_doctor BIGINT UNSIGNED,
    fecha_cita DATETIME,
    estado ENUM('Pendiente', 'Completada', 'Cancelada') NOT NULL,
    FOREIGN KEY (id_paciente) REFERENCES pacientes(id_paciente) ON DELETE
CASCADE,
    FOREIGN KEY (id_doctor) REFERENCES doctores(id_doctor) ON DELETE
CASCADE
);
```

Verificación de la Base de Datos

```
SHOW TABLES;
```

```
admin_conederal@PCconederal:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 556
Server version: 8.0.41-Ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE conederal;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_conederal |
+-----+
| citas
| clinicas
| doctores
| empleados
| material_dental
| medicamentos
| pacientes
| pacientes_medicamentos
| pacientes_tratamientos
| protesis_dentales
| proveedores_dentales
| tratamientos
| tratamientos_medicamentos |
+-----+
13 rows in set (0.01 sec)
```

```
DESCRIBE pacientes; DESCRIBE medicamentos; DESCRIBE pacientes_medicamentos;
```

```
mysql> describe pacientes_medicamentos;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_paciente | bigint unsigned | NO | PRI | NULL |
| id_medicamento | bigint unsigned | NO | PRI | NULL |
| fecha_inicio | date | YES | | NULL |
| fecha_fin | date | YES | | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> describe pacientes;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_paciente | bigint unsigned | NO | PRI | NULL | auto_increment |
| nombre | varchar(100) | NO | | NULL |
| dni | varchar(20) | NO | UNI | NULL |
| telefono | varchar(20) | YES | | NULL |
| email | varchar(100) | YES | | NULL |
| direccion | text | YES | | NULL |
| fecha_nacimiento | date | YES | | NULL |
| id_clinica | bigint unsigned | NO | MUL | NULL |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> describe medicamentos;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_medicamento | bigint unsigned | NO | PRI | NULL | auto_increment |
| nombre | varchar(100) | NO | | NULL |
| descripcion | text | YES | | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Inserción de Datos en la Base de Datos

Clínicas

3 clínicas ubicadas en Madrid.

```
INSERT INTO clinicas (nombre, direccion, telefono, email, horario,
responsable, fecha_apertura) VALUES
('ConeDental - Centro', 'Calle Gran Vía, 45, Madrid', '911234567',
'centro@conedental.com', 'Lunes a Viernes 9:00-20:00', 'Dr. Juan
Pérez', '2015-03-10'),
('ConeDental - Norte', 'Avenida de la Ilustración, 12, Madrid',
'912345678', 'norte@conedental.com', 'Lunes a Sábado 10:00-18:00',
'Dra. María López', '2018-07-05'),
('ConeDental - Sur', 'Calle de Alcalá, 300, Madrid', '913456789',
'sur@conedental.com', 'Lunes a Viernes 8:00-19:00', 'Dr. Carlos
Fernández', '2020-01-15');
```

Pacientes

30 pacientes, 10 por clínica.

```
INSERT INTO pacientes (nombre, dni, telefono, email, direccion,
fecha_nacimiento, id_clinica) VALUES
```

Pacientes en ConeDental - Centro (id_clinica = 1)

```
('Ana Gómez', '12345678A', '600123456', 'ana.gomez@mail.com', 'Calle
Mayor, 5, Madrid', '1985-04-12', 1),
('Luis Fernández', '23456789B', '601234567',
'luis.fernandez@mail.com', 'Calle Arenal, 22, Madrid', '1990-07-23',
1),
('Marta Sánchez', '34567890C', '602345678', 'marta.sanchez@mail.com',
'Calle Preciados, 10, Madrid', '1992-11-15', 1),
...
```

Pacientes en ConeDental - Norte (id_clinica = 2)

```
('Isabel Romero', '11223344A', '610123456', 'isabel.romero@mail.com',
'Calle Serrano, 20, Madrid', '1991-01-05', 2),
('Diego Molina', '22334455B', '611234567', 'diego.molina@mail.com',
'Paseo Castellana, 100, Madrid', '1987-07-21', 2),
...
```

Pacientes en ConeDental - Sur (id_clinica = 3)

```
('Andrea Montes', '11112222A', '620123456', 'andrea.montes@mail.com',
'Calle Vallecas, 14, Madrid', '1994-06-24', 3),
('Jesús Cabrera', '22223333B', '621234567', 'jesus.cabrera@mail.com',
'Calle Méndez Álvaro, 7, Madrid', '1986-05-20', 3);
```

Doctores

12 doctores, 4 por clínica.

```
INSERT INTO doctores (nombre, dni, telefono, email, especialidad,
id_clinica, fecha_contratacion) VALUES

('Dr. Álvaro Serrano', '32165498A', '630123456',
'alvaro.serrano@mail.com', 'Ortodoncia', 1, '2015-04-12'),
('Dra. Carmen Ríos', '43216589B', '631234567',
'carmen.rios@mail.com', 'Implantología', 1, '2017-07-23'),
...
('Dr. Ricardo Torres', '76549812E', '634567890',
'ricardo.torres@mail.com', 'Ortodoncia', 2, '2016-06-30'),
('Dra. Laura Fernández', '87650923F', '635678901',
'laura.fernandez@mail.com', 'Periodoncia', 2, '2019-09-12'),
...
('Dr. Manuel Gómez', '10983256I', '638901234',
'manuel.gomez@mail.com', 'Implantología', 3, '2013-05-07'),
('Dra. Alicia Ortega', '21094367J', '639012345',
'alicia.ortega@mail.com', 'Estética Dental', 3, '2018-08-29');
...
```

Material Dental

30 materiales distribuidos en las clínicas.

```
INSERT INTO material_dental (nombre, descripcion, cantidad,
id_doctor, id_clinica, fecha_compra, proveedor) VALUES

('Punción e infusión', 'Material para punción e infusión', 50, 1, 1,
'2023-01-10', 'Proveedor A'),
('Guantes Médicos', 'Guantes médicos de un solo uso', 200, 1, 1,
'2023-02-15', 'Proveedor B'),
...
('Cobertura quirúrgica', 'Cobertura para procedimientos quirúrgicos',
80, 5, 2, '2023-01-18', 'Proveedor K'),
...
('Cabinas de seguridad y de gases', 'Cabinas de seguridad y de gases
para procedimientos médicos', 3, 9, 3, '2023-01-22', 'Proveedor U');
```

Prótesis Dentales

30 prótesis asignadas a pacientes y doctores.

```
INSERT INTO protesis_dentales (tipo, material, id_paciente,
id_doctor, fecha_colocacion, durabilidad, observaciones) VALUES
```

Prótesis en ConeDental - Centro (id_clinica = 1)

```
('Corona dental de metal-porcelana', 'Metal-porcelana', 1, 1, '2023-01-12', '5 años', 'Prótesis fija de alta durabilidad'),  
...
```

Prótesis en ConeDental - Norte (id_clinica = 2)

```
('Puente dental 3 piezas', 'Metal-porcelana', 6, 11, '2023-01-25', '7 años', 'Puente dental para restauración de piezas'),  
...
```

Prótesis en ConeDental - Sur (id_clinica = 3)

```
('Implante dental', 'Titanio', 13, 2, '2023-06-05', 'Indefinido',
'Implante dental unitario en maxilar superior');  
...
```

Empleados

21 empleados distribuidos entre las clínicas.

```
INSERT INTO empleados (nombre, dni, telefono, email, cargo, id_clinica,
fecha_contratacion) VALUES
```

Empleados en ConeDental - Centro (id_clinica = 1)

```
('Ana Martínez', '12345678A', '612345678',
'ana.martinez@conedental.com', 'Auxiliar', 1, '2023-01-10'),  
...
```

Empleados en ConeDental - Norte (id_clinica = 2)

```
('José Fernández', '78901234G', '612345684',
'jose.fernandez@conedental.com', 'Auxiliar', 2, '2023-01-15'),  
...
```

Empleados en ConeDental - Sur (id_clinica = 3)

```
('Raquel Martín', '34567890M', '612345690',
'raquel.martin@conedental.com', 'Auxiliar', 3, '2023-01-20');
```

```
...
```

Proveedores Dentales

6 proveedores para todas las clínicas.

```
INSERT INTO proveedores_dentales (nombre, telefono, email, direccion,
tipo_material, id_clinica, fecha_ultimo_pedido) VALUES
('Proveedores Dental S.A.', '912345678',
'contacto@proveedordental.com', 'Calle Falsa 123, Madrid', 'Material
médico de trasplante capilar', 1, '2023-03-10'),
('Dental Equipos Global', '967890123',
'contacto@dentalequiposglobal.com', 'Calle Mayor 303, Madrid',
'Esterilización y autoclave', 3, '2023-06-10');
...
...
```

Instalación y Configuración de Phyton, Flask y CORS

Instalación de Python y Virtual Environment

```
sudo apt install python3 python3-pip python3-venv -y
```

Crear un entorno virtual

```
python3 -m venv venv
source venv/bin/activate
```

Instalación de Flask y -CORS

```
pip install flask flask-cors mysql-connector-python
```

```
(venv) admin_condenatal@PCcondenatal:/var/www/html$ pip install flask-cors
Requirement already satisfied: flask-cors in ./venv/lib/python3.12/site-packages (5.0.1)
Requirement already satisfied: flask>=0.9 in ./venv/lib/python3.12/site-packages (from flask-cors) (3.1.0)
Requirement already satisfied: Werkzeug>=0.7 in ./venv/lib/python3.12/site-packages (from flask-cors) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in ./venv/lib/python3.12/site-packages (from flask>=0.9->flask-cors) (3.1
.6)
Requirement already satisfied: itsdangerous>=2.2 in ./venv/lib/python3.12/site-packages (from flask>=0.9->flask-cors)
(2.2.0)
Requirement already satisfied: click>=8.1.3 in ./venv/lib/python3.12/site-packages (from flask>=0.9->flask-cors) (8.1.
8)
Requirement already satisfied: blinker>=1.9 in ./venv/lib/python3.12/site-packages (from flask>=0.9->flask-cors) (1.9.
0)
Requirement already satisfied: MarkupSafe>=2.1.1 in ./venv/lib/python3.12/site-packages (from Werkzeug>=0.7->flask-c
ors) (3.0.2)
```

Desarrollo de la API con Flask y Flask-CORS

Creación el archivo app.py

```
nano app.py
```

App.py:

```
from flask import Flask, jsonify, request
from flask_cors import CORS # Habilitar CORS
import mysql.connector

app = Flask(__name__)
CORS(app) # Permitir accesos desde frontend

# ♦ Función para conectar con MySQL
def get_db_connection():
    return mysql.connector.connect(
        host='localhost',
        user='admin_conedental',
        password='Pa$$w0rd',
        database='cone_dental'
    )

# ♦ API: Obtener Clínicas
@app.route('/clinicas', methods=['GET'])
def get_clinicas():
    connection = get_db_connection()
    cursor = connection.cursor(dictionary=True)
    cursor.execute('SELECT * FROM clinicas')
    clinicas = cursor.fetchall()
    cursor.close()
    connection.close()
    return jsonify(clinicas)

...

# ♦ Ejecutar la API
if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```



The screenshot shows a code editor with two tabs: 'index.html' and 'app.py'. The 'app.py' tab is active and displays the following Python code:

```
1  from flask import Flask, jsonify, request
2  from flask_cors import CORS # Importar CORS
3  import mysql.connector
4
5  app = Flask(__name__)
6
7  # Habilitar CORS para todas las rutas
8  CORS(app)
9
10 # 🔘 Función para conectar con MySQL
11 def get_db_connection():
12     return mysql.connector.connect(
13         host='localhost',
14         user='admin',
15         password='Pa$$w0rd',
16         database='conedental'
17     )
18
19 # 🔘 API: Obtener Clínicas
20 @app.route('/clinicas', methods=['GET'])
21 def get_clinicas():
22     connection = get_db_connection()
23     cursor = connection.cursor(dictionary=True)
24     cursor.execute('SELECT * FROM clinicas')
25     clinicas = cursor.fetchall()
26     cursor.close()
27     connection.close()
28     return jsonify(clinicas)
29
30 # 🔘 API: Obtener Pacientes
31 @app.route('/pacientes', methods=['GET'])
32 def get_pacientes():
33     connection = get_db_connection()
34     cursor = connection.cursor(dictionary=True)
35     cursor.execute('SELECT * FROM pacientes')
36     pacientes = cursor.fetchall()
37     cursor.close()
38     connection.close()
39     return jsonify(pacientes)
40
41 # 🔘 API: Obtener Doctores
42 @app.route('/doctores', methods=['GET'])
43 def get_doctores():
44     connection = get_db_connection()
45     cursor = connection.cursor(dictionary=True)
46     cursor.execute('SELECT * FROM doctores')
47     doctores = cursor.fetchall()
48     cursor.close()
```

Ejecutar la API Flask

Activar el entorno virtual

```
source venv/bin/activate
```

Iniciar la API

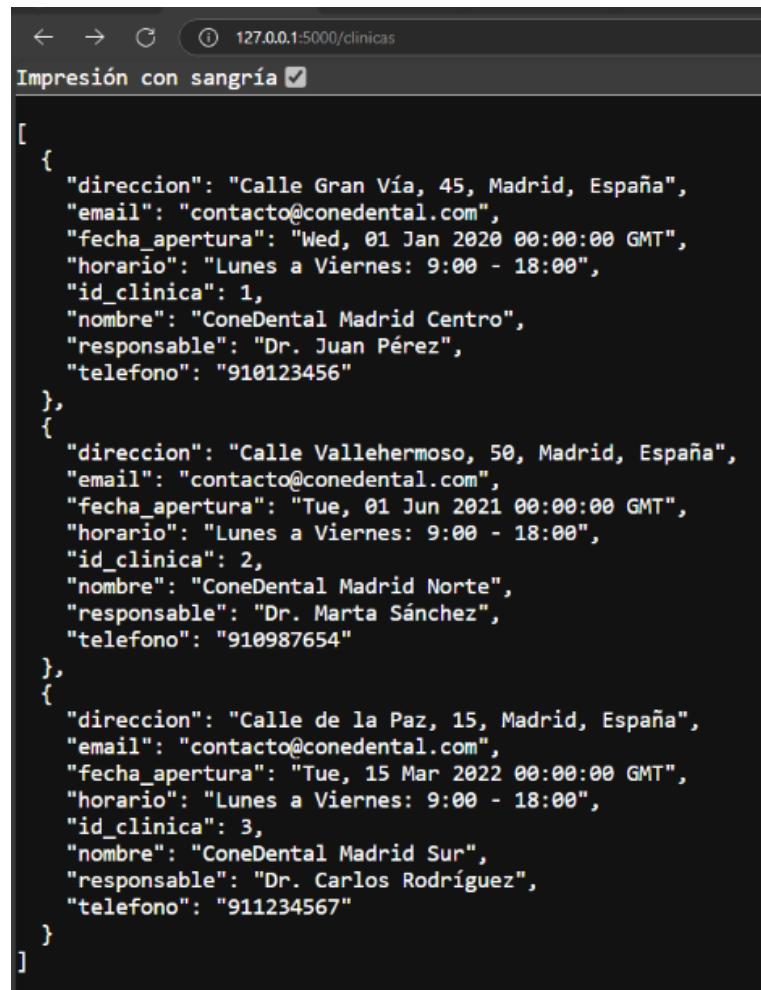
```
python app.py
```

```
(venv) admin_condenatal@PCcondenatal:/var/www/html$ python3 app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.28.89.50:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 851-520-613
127.0.0.1 - - [06/Mar/2025 18:15:11] "GET /clinicas HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2025 18:16:47] "GET /clinicas HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2025 18:19:47] "GET /clinicas HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2025 18:23:23] "GET /clinicas HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2025 18:24:45] "GET /clinicas HTTP/1.1" 200 -
 * Detected change in '/var/www/html/app.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 851-520-613
 * Detected change in '/var/www/html/app.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 851-520-613
127.0.0.1 - - [06/Mar/2025 18:30:07] "GET /citas_por_doctores HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2025 18:30:07] "GET /clinicas HTTP/1.1" 500 -
Traceback (most recent call last):
  File "/var/www/html/venv/lib/python3.12/site-packages/mysql/connector/connection_cext.py", line 365, in _open_connection
    self._cmysql.connect(**cnx_kwargs)
_mysql.connector.MySQLInterfaceError: Authentication plugin 'mysql_native_password' cannot be loaded: it is already loaded
The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "/var/www/html/venv/lib/python3.12/site-packages/flask/app.py", line 1536, in __call__
    return self.wsgi_app(environ, start_response)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/var/www/html/venv/lib/python3.12/site-packages/flask/app.py", line 1514, in wsgi_app
    response = self.handle_exception(e)
           ^^^^^^^^^^^^^^^^^^^^^^
  File "/var/www/html/venv/lib/python3.12/site-packages/flask_cors/extension.py", line 176, in wrapped_function
    return cors_after_request(app.make_response(f(*args, **kwargs)))
           ^^^^^^^^^^^^^^^^^^^^^^
  File "/var/www/html/venv/lib/python3.12/site-packages/flask/app.py", line 1511, in wsgi_app
    response = self.full_dispatch_request()
           ^^^^^^^^^^^^^^^^^^^^^^
  File "/var/www/html/venv/lib/python3.12/site-packages/flask/app.py", line 919, in full_dispatch_request
    rv = self.handle_user_exception(e)
```

Probar la API

<http://127.0.0.1:5000/clinicas>



The screenshot shows a browser window with the URL `127.0.0.1:5000/clinicas` in the address bar. A checkbox labeled "Impresión con sangría" is checked. The page content displays a JSON array of three clinic objects:

```
[{"id_clinica": 1, "nombre": "ConeDental Madrid Centro", "responsable": "Dr. Juan Pérez", "telefono": "910123456", "direccion": "Calle Gran Vía, 45, Madrid, España", "email": "contacto@conedental.com", "fecha_apertura": "Wed, 01 Jan 2020 00:00:00 GMT", "horario": "Lunes a Viernes: 9:00 - 18:00"}, {"id_clinica": 2, "nombre": "ConeDental Madrid Norte", "responsable": "Dr. Marta Sánchez", "telefono": "910987654", "direccion": "Calle Vallehermoso, 50, Madrid, España", "email": "contacto@conedental.com", "fecha_apertura": "Tue, 01 Jun 2021 00:00:00 GMT", "horario": "Lunes a Viernes: 9:00 - 18:00"}, {"id_clinica": 3, "nombre": "ConeDental Madrid Sur", "responsable": "Dr. Carlos Rodríguez", "telefono": "911234567", "direccion": "Calle de la Paz, 15, Madrid, España", "email": "contacto@conedental.com", "fecha_apertura": "Tue, 15 Mar 2022 00:00:00 GMT", "horario": "Lunes a Viernes: 9:00 - 18:00"}]
```

Código HTML .

Estructura Head

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <title>Dashboard - ConeDental</title>
        <link rel="stylesheet" href="styles.css">
</head>
<body>
    <!-- Contenido aquí -->
</body>
</html>
```

Bootstrap y Chart.js

```
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap
      .min.css">
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<link rel="stylesheet" href="styles.css">
```

Sidebar de Navegación

```
<nav class="sidebar">
    <div class="sidebar-header">
        <h2>ConeDental</h2>
    </div>
    <ul class="sidebar-menu">
        <li><a href="#">Dashboard</a></li>
        <li><a href="#">Citas</a></li>
        <li><a href="#">Clínicas</a></li>
        <li><a href="#">Doctores</a></li>
        <li><a href="#">Empleados</a></li>
        <li><a href="#">Material Dental</a></li>
        <li><a href="#">Medicamentos</a></li>
        <li><a href="#">Pacientes</a></li>
        <li><a href="#">Proveedores</a></li>
        <li><a href="#">Tratamientos</a></li>
    </ul>
</nav>
```

Tarjetas de Resumen

```

<div class="container mt-4">
  <div class="row">
    <div class="col-md-3">
      <div class="card info-card">
        <h5>Citas</h5>
        <p id="total-citas">0</p>
      </div>
    </div>
  </div>
</div>

```

index.html

```

back2 > index.html > html > body > nav.sidebar > ul.sidebar-menu > li
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Dashboard - ConeDental</title>
7    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
8    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
9    <link rel="stylesheet" href="styles.css">
10 </head>
11 <body>
12
13  <!-- Sidebar de Navegación -->
14  <nav class="sidebar">
15    <div class="sidebar-header">
16      <h2>ConeDental</h2>
17    </div>
18    <ul class="sidebar-menu">
19      <li><a href="#">Dashboard</a></li>
20      <li><a href="#">Citas</a></li>
21      <li><a href="#">Clínicas</a></li>
22      <li><a href="#">Doctores</a></li>
23      <li><a href="#">Empleados</a></li>
24      <li><a href="#">Material Dental</a></li>
25      <li><a href="#">Medicamentos</a></li>
26      <li><a href="#">Pacientes</a></li>
27      <li><a href="#">Proveedores</a></li>
28      <li><a href="#">Tratamientos</a></li>
29    </ul>
30  </nav>
31
32  <!-- Contenido Principal -->
33  <div class="content">
34    <header class="header">
35      <h1>Dashboard ConeDental</h1>
36    </header>
37
38    <!-- Sección de Tarjetas de Resumen -->
39    <div class="container mt-4">
40      <div class="row">
41        <div class="col-md-3">
42          <div class="card info-card">
43            <h5>Citas</h5>
44            <p id="total-citas">0</p>
45          </div>
46        </div>
47        <div class="col-md-3">
48          <div class="card info-card">

```

Gráficos con Chart.js

```
<div class="col-md-6">
    <div class="card chart-card">
        <h5>Citas por Doctores</h5>
        <canvas id="chart-citas"></canvas>
    </div>
</div>
```

Tablas Interactivas

```
<div class="card table-card">
    <h5>Lista de Medicamentos por Pacientes</h5>
    <table class="table">
        <thead>
            <tr>
                <th>Paciente</th>
                <th>Medicamento</th>
            </tr>
        </thead>
        <tbody id="tabla-pacientes-medicamentos"></tbody>
    </table>
</div>
```

Conexión con JavaScript

```
<script src="script.js"></script>
```

CSS

Estilos Generales

```
body {  
    font-family: 'Arial', sans-serif;  
    background-color: #f4f4f4;  
    margin: 0;  
    padding: 0;  
    display: flex;  
}
```

Sidebar

```
.sidebar {  
    width: 250px;  
    background: #2c3e50;  
    color: white;  
    height: 100vh;  
    padding: 20px;  
    position: fixed;  
}
```

Contenido Principal

```
.content {  
    margin-left: 260px;  
    padding: 20px;  
    width: calc(100% - 260px);  
}
```

Estilos de Tarjetas

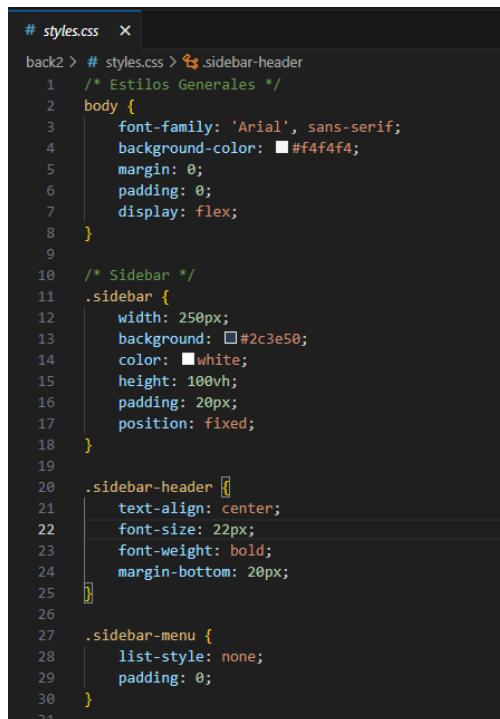
```
.info-card {  
    background: white;  
    padding: 20px;  
    border-radius: 10px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
    text-align: center;  
    font-size: 18px;  
}
```

Estilos de Gráficos

```
.chart-card {  
    background: white;  
    padding: 20px;  
    border-radius: 10px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
    text-align: center;  
    margin-bottom: 20px;  
}
```

Estilos de Tablas

```
.table-card {  
    background: white;  
    padding: 20px;  
    border-radius: 10px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
    margin-bottom: 20px;  
}  
  
.table th, .table td {  
    text-align: center;  
    vertical-align: middle;  
}
```



A screenshot of a code editor window titled '# styles.css'. The code is written in CSS and defines styles for a sidebar. It includes rules for the body, sidebar, sidebar-header, and sidebar-menu. The sidebar has a fixed position, a width of 250px, and a background color of #2c3e50. The sidebar-header is centered, has a font size of 22px, bold weight, and a margin-bottom of 20px. The sidebar-menu has no list style and zero padding.

```
# styles.css x  
back2 > # styles.css > ⚒ sidebar-header  
1  /* Estilos Generales */  
2  body {  
3      font-family: 'Arial', sans-serif;  
4      background-color: ■#f4f4f4;  
5      margin: 0;  
6      padding: 0;  
7      display: flex;  
8  }  
9  
10 /* Sidebar */  
11 .sidebar {  
12     width: 250px;  
13     background: ■#2c3e50;  
14     color: ■white;  
15     height: 100vh;  
16     padding: 20px;  
17     position: fixed;  
18  }  
19  
20 .sidebar-header [  
21     text-align: center;  
22     font-size: 22px;  
23     font-weight: bold;  
24     margin-bottom: 20px;  
25 ]  
26  
27 .sidebar-menu {  
28     list-style: none;  
29     padding: 0;  
30  }  
31
```

Código JavaScript

Cargar Datos al Iniciar la Página

```
document.addEventListener('DOMContentLoaded', () => {
```

Obtener Datos de la API

```
function fetchData(endpoint, elementId) {
    fetch(`http://127.0.0.1:5000/${endpoint}`)
        .then(response => response.json())
        .then(data => {
            console.log(endpoint, data);
            document.getElementById(elementId).textContent =
            data.length;
        })
        .catch(error => console.error(`Error al cargar ${endpoint}:`, error));
}
```

Cargar Información en las Tarjetas

```
fetchData('clinicas', 'total-clinicas');
fetchData('pacientes', 'total-pacientes');
fetchData('doctores', 'total-doctores');
fetchData('citas', 'total-citas');
fetchData('empleados', 'total-empleados');
fetchData('material_dental', 'total-material');
fetchData('medicamentos', 'total-medicamentos');
fetchData('proveedores_dentales', 'total-proveedores');
```

Gráfico de Citas por Doctores

```
fetch('http://127.0.0.1:5000/citas_por_doctores')
    .then(response => response.json())
    .then(data => {
        console.log("Citas por Doctores:", data);
        const ctx = document.getElementById('chart-citas').getContext('2d');
        new Chart(ctx, {
            type: 'bar',
```

```

        data: {
          labels: data.map(item => item.doctor),
          datasets: [
            {
              label: 'Cititas',
              data: data.map(item => item.total_citas),
              backgroundColor: 'rgba(54, 162, 235, 0.6)'
            }
          ]
        });
      })
      .catch(error => console.error('Error al cargar citas por
doctores:', error));
    
```

Gráfico de Pacientes por Tratamientos

```

fetch('http://127.0.0.1:5000/pacientes_por_tratamientos')
  .then(response => response.json())
  .then(data => {
    console.log("Pacientes por Tratamientos:", data);
    const ctx = document.getElementById('chart-
tratamientos').getContext('2d');
    new Chart(ctx, {
      type: 'pie',
      data: {
        labels: data.map(item => item.tratamiento),
        datasets: [
          {
            label: 'Pacientes',
            data: data.map(item => item.total_pacientes),
            backgroundColor: ['#FF6384', '#36A2EB',
'#FFCE56']
          }
        ]
      }
    });
  })
  .catch(error => console.error('Error al cargar pacientes por
tratamientos:', error));
    
```

Citas

```

fetch('http://127.0.0.1:5000/citas')
  .then(response => response.json())
  .then(data => {
    console.log("Citas:", data);
    const listaCitas = document.getElementById('lista-citas');
    listaCitas.innerHTML = '';

    data.forEach(cita => {
      const li = document.createElement('li');
      li.textContent = `📅 ${cita.fecha_hora} - Paciente:
${cita.id_paciente} - Motivo: ${cita.motivo}`;
      listaCitas.appendChild(li);
    });
  });
    
```

```

    })
    .catch(error => console.error('Error al cargar las citas:',
error));
}
  
```

Pacientes en la Página

```

fetch('http://127.0.0.1:5000/pacientes')
  .then(response => response.json())
  .then(data => {
    console.log("Pacientes:", data);
    const listaPacientes = document.getElementById('lista-
pacientes');
    listaPacientes.innerHTML = '';

    data.forEach(paciente => {
      const li = document.createElement('li');
      li.textContent = `${paciente.nombre} - DNI:
${paciente.dni} - Tel: ${paciente.telefono}`;
      listaPacientes.appendChild(li);
    });
  })
  .catch(error => console.error('Error al cargar los pacientes:', error));
}
  
```

Gráficos de Medicamentos por Pacientes y Tratamientos

Gráfico de barras con la cantidad de medicamentos por paciente

```

const pacientesMedicamentosChart = new
Chart(document.getElementById('chart-pacientes-
medicamentos').getContext('2d'), {
  type: 'bar',
  data: {
    labels: pacientesMedicamentos.map(item => `Paciente
${item.id_paciente}`),
    datasets: [{
      label: 'Cantidad de Medicamentos',
      data: pacientesMedicamentos.map(item =>
item.id_medicamento),
      backgroundColor: 'rgba(54, 162, 235, 0.6)'
    }]
  }
});
  
```

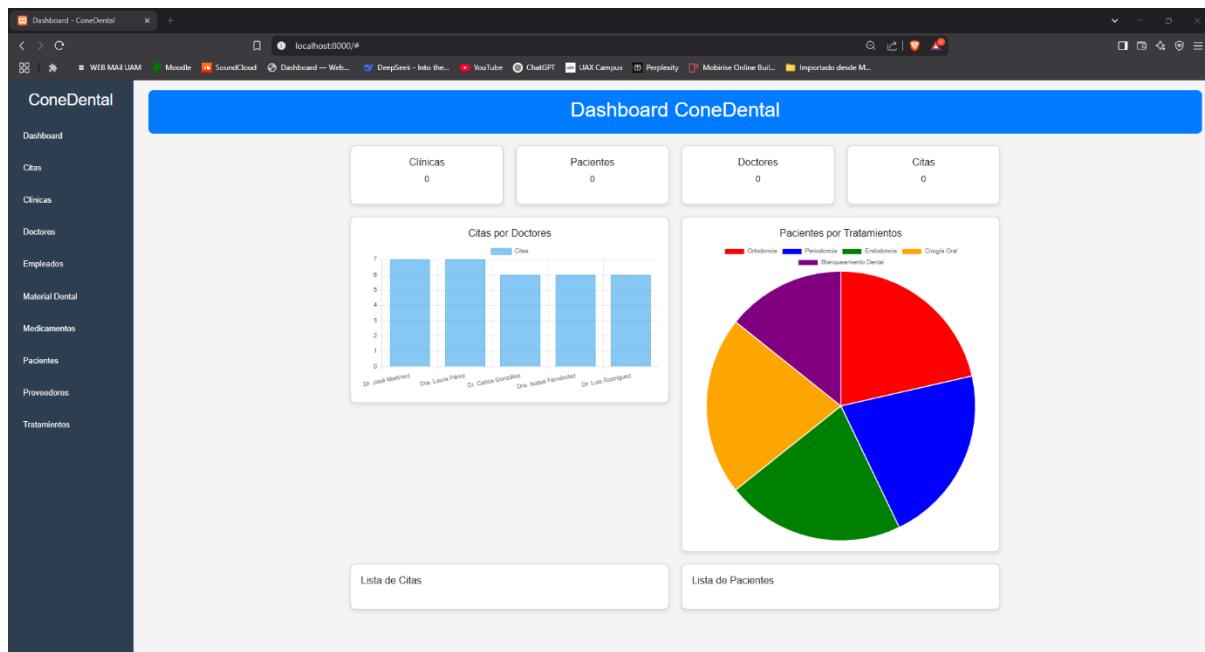
Gráfico de pastel con medicamentos por tratamiento.

```
const tratamientosMedicamentosChart = new
Chart(document.getElementById('chart-tratamientos-
medicamentos').getContext('2d'), {
    type: 'pie',
    data: {
        labels: tratamientosMedicamentos.map(item => `Tratamiento
${item.id_tratamiento}`),
        datasets: [
            data: tratamientosMedicamentos.map(item =>
item.id_medicamento),
            backgroundColor: ['#FF6384', '#36A2EB', '#FFCE56',
'#4CAF50', '#FF9F40']
        ]
    }
});
```



```
JS scriptjs  x
back2 > JS scriptjs > ...
1  document.addEventListener('DOMContentLoaded', () => {
2
3      // Cargar datos de la API
4      function fetchData(endpoint, elementId) {
5          fetch(`http://127.0.0.1:5000/${endpoint}`)
6              .then(response => response.json())
7              .then(data => {
8                  console.log(endpoint, data);
9                  document.getElementById(elementId).textContent = data.length;
10             })
11             .catch(error => console.error(`Error al cargar ${endpoint}:`, error));
12         }
13
14     // Llamadas a la API
15     fetchData('clinicas', 'total-clinicas');
16     fetchData('pacientes', 'total-pacientes');
17     fetchData('doctores', 'total-doctores');
18     fetchData('citas', 'total-citas');
19     fetchData('empleados', 'total-empleados');
20     fetchData('material_dental', 'total-material');
21     fetchData('medicamentos', 'total-medicamentos');
22     fetchData('proveedores_dentales', 'total-proveedores');
23
24 );
25
26 document.addEventListener('DOMContentLoaded', () => {
27     fetch('http://127.0.1:5000/citas_por_doctores')
28         .then(response => response.json())
29         .then(data => {
30             console.log("Citas por Doctores:", data);
31             const ctx = document.getElementById('chart-citas').getContext('2d');
32             new Chart(ctx, {
33                 type: 'bar',
34                 data: {
35                     labels: data.map(item => item.doctor),
36                     datasets: [
37                         {
38                             label: 'Citas',
39                             data: data.map(item => item.total_citas),
40                             backgroundColor: 'rgba(54, 162, 235, 0.6)'
41                         }
42                     ];
43                 });
44             .catch(error => console.error('Error al cargar citas por doctores:', error));
45         });
46     });
47
```

Resumen Estructura y Componentes del Dashboard



Menú de Navegación (Sidebar)

En el lateral izquierdo se encuentra un menú de navegación con accesos directos a diferentes secciones del sistema. Este menú permite a los usuarios desplazarse entre las distintas funcionalidades, como la gestión de citas, doctores, empleados, materiales dentales, medicamentos y proveedores.

Tarjetas de Resumen

Debajo del encabezado se presentan cuatro tarjetas informativas que muestran el número total de clínicas, pacientes, doctores y citas registradas en el sistema. Estas tarjetas permiten una rápida visualización del estado general de la clínica.

Visualización de Datos

El dashboard incluye gráficos dinámicos generados con Chart.js, los cuales representan información clave:

Gráfico de Barras: "Citas por Doctores"

Este gráfico muestra la distribución de citas entre los doctores de la clínica. Cada barra representa a un doctor, y la altura de la barra indica el número de citas asignadas.

Gráfico de Pastel: "Pacientes por Tratamientos"

Este gráfico segmenta la cantidad de pacientes según los tratamientos que han recibido. Cada segmento del gráfico representa un tratamiento diferente, identificado con un color distinto.

Listados de Información

En la parte inferior del dashboard hay dos secciones destinadas a listar datos detallados:

Lista de Citas

Debería mostrar información detallada sobre las citas agendadas, incluyendo la fecha, el paciente asignado y el motivo de la cita.

Lista de Pacientes

Se espera que muestre el nombre, DNI y número de teléfono de los pacientes registrados en la clínica.

TRABAJOS FUTUROS

- Filtros dinámicos (por fecha, paciente, doctor)
- Alertas y notificaciones (bajo stock de medicamentos, citas pendientes)
- Optimización del rendimiento (carga más rápida, caché)
- Agregar más gráficos
- Mejorar el diseño
- Optimizar la base de datos

CONCLUSIONES

Conclusión profesional del proyecto.

REFERENCIAS

- Bootstrap. (s.f.). Introduction to Bootstrap. Bootstrap. Recuperado el 7 de marzo de 2025, de <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- Chart.js. (s.f.). Getting Started. Chart.js. Recuperado el 7 de marzo de 2025, de <https://www.chartjs.org/docs/latest/>
- Mozilla Developer Network (MDN). (s.f.). Using Fetch. MDN Web Docs. Recuperado el 7 de marzo de 2025, de https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
- MySQL. (s.f.). MySQL 8.0 Reference Manual. MySQL Documentation. Recuperado el 7 de marzo de 2025, de <https://dev.mysql.com/doc/refman/8.0/en/>
- Python Software Foundation. (s.f.). Flask Documentation (2.3.x). Python.org. Recuperado el 7 de marzo de 2025, de <https://flask.palletsprojects.com/en/2.3.x/>
- W3Schools. (s.f.). CSS Flexbox Guide. W3Schools. Recuperado el 7 de marzo de 2025, de https://www.w3schools.com/css/css3_flexbox.asp
- MDN Web Docs. (s.f.). Introduction to CORS. Mozilla Developer Network. Recuperado el 7 de marzo de 2025, de <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
- Microsoft. (s.f.). WSL Documentation: Windows Subsystem for Linux. Microsoft Learn. Recuperado el 7 de marzo de 2025, de <https://learn.microsoft.com/en-us/windows/wsl/>
- DigitalOcean. (s.f.). How To Install and Use MySQL on Ubuntu 20.04. DigitalOcean Tutorials. Recuperado el 7 de marzo de 2025, de <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04>
- FreeCodeCamp. (s.f.). A Beginner's Guide to REST APIs. FreeCodeCamp. Recuperado el 7 de marzo de 2025, de <https://www.freecodecamp.org/news/rest-api-tutorial-restful-web-services-for-beginners/>