**Instalación y Configuración de WSL con Ubuntu 24.04.1 LTS**

**Habilitar WSL en Windows**

**PowerShell** como administrador.

wsl –install

A computer screen with white text

AI-generated content may be incorrect.

**Instalar Ubuntu 24.04.1 LTS desde Microsoft Store**

* **Microsoft Store** en Windows.
* **Ubuntu 24.04.1 LTS**.

A screenshot of a computer

AI-generated content may be incorrect.

**Crear Usuario y Configuración Básica**

* **Nombre de usuario:** admin\_conedental
* **Contraseña:** Pa$$w0rd

**Cambiar el Nombre del Equipo**

sudo hostnamectl set-hostname PCconedental

**A screenshot of a computer

AI-generated content may be incorrect.**

**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)

A screenshot of a computer

AI-generated content may be incorrect.

**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

**A screenshot of a computer

AI-generated content may be incorrect.**

**Acceso a MySQL y Creación del Usuario**

**Iniciar sesión en MySQL**

sudo mysql -u root -p

**Crear Usuario MySQL para ConeDental**:

CREATE USER 'admin\_conedental'@'localhost' IDENTIFIED BY 'Pa$$w0rd';

GRANT ALL PRIVILEGES ON \*.\* TO 'admin\_conedental'@'localhost' WITH GRANT OPTION;

FLUSH PRIVILEGES;

**Crear la Base de Datos**

CREATE DATABASE conedental;

SHOW DATABASES;

A screenshot of a computer

AI-generated content may be incorrect.

**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

A screenshot of a computer

AI-generated content may be incorrect.

**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;

A screenshot of a computer

AI-generated content may be incorrect.

DESCRIBE pacientes; DESCRIBE medicamentos; DESCRIBE pacientes\_medicamentos;

A screenshot of a computer

AI-generated content may be incorrect.

**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

**A screenshot of a computer program

AI-generated content may be incorrect.**

**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)

**A screenshot of a computer

AI-generated content may be incorrect.**

**Ejecutar la API Flask**

**Activar el entorno virtual**

source venv/bin/activate

**Iniciar la API**

python app.py

A screenshot of a computer program

AI-generated content may be incorrect.

**Probar la API**

http://127.0.0.1:5000/clinicas

A screenshot of a computer

AI-generated content may be incorrect.

**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>

**A screenshot of a computer program

AI-generated content may be incorrect.**

**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 screen shot of a computer program

AI-generated content may be incorrect.

**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: 'Citas',

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']

}]

}

});

A screen shot of a computer program

AI-generated content may be incorrect.

**Resumen** **Estructura y Componentes del Dashboard**

A screenshot of a computer

AI-generated content may be incorrect.

**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.