Digitalització d’un Taller Mecànic Connectat

Mòdul: 1665 Digitalització

**Nom de l’equip**: Jordi Arabia, Cristian Moyano y Denís Ramos

**Data de lliurament**: XX/05/2025

# Índex

[Índex 2](#_Toc197945044)

[1 Introducció 3](#_Toc197945045)

[1.1 Objetivos del proyecto 3](#_Toc197945046)

[1.2 Componentes del Sistema 3](#_Toc197945047)

[2 Anàlisi Inicial 4](#_Toc197945048)

[2.1 Necesidades del cliente 4](#_Toc197945049)

[3 Model de Dades 5](#_Toc197945050)

[3.1 Diagrama ER 5](#_Toc197945051)

[3.2 Esquema relacional 6](#_Toc197945052)

[3.3 Script SQL 6](#_Toc197945053)

[4 Desenvolupament de la Solució 9](#_Toc197945054)

[4.1 API 9](#_Toc197945055)

[4.2 Web 9](#_Toc197945056)

[4.3 Scripts 9](#_Toc197945057)

[5 Infraestructura de Sistemes 10](#_Toc197945058)

[6 Problemes i Solucions 11](#_Toc197945059)

[7 Conclusions 12](#_Toc197945060)

[8 Annexos 13](#_Toc197945061)

# Introducció

Nuestro objectivo en este proyecto es crear un sistema digital para la gestión de clientes y citas en un taller mecánico. El sistema se desarrollará utilizando una web, una API y una base de datos, con el fin de automatizar tareas básicas y mejorar la experiencia del cliente.

## Objetivos del proyecto

El proyecto tiene como objetivo principal mejorar la eficiencia y la experiencia del cliente en los talleres mecánicos. Para lograr esto, se han definido los siguientes objetivos:

* **Crear un sistema digital para la gestión de clientes y citas en un taller mecánico:** El objetivo principal de este proyecto es desarrollar un sistema que permita a los clientes gestionar sus citas y servicios de manera fácil y accesible en línea.
* **Automatizar tareas básicas relacionadas con la gestión de clientes y citas:** El sistema debe ser capaz de automatizar tareas como la gestión de citas, la notificación de los clientes y la gestión de la información de los clientes.
* **Mejorar la experiencia del cliente a través de una interfaz de usuario amigable y accesible en línea:** El sistema debe ser fácil de usar y permitir a los clientes acceder a la información de sus citas y servicios en cualquier momento y desde cualquier lugar.

## Componentes del Sistema

* Una web de gestión de clientes y citas para que los clientes puedan acceder a la información de sus citas y servicios en línea.
* Una API (Application Programming Interface) para permitir la comunicación entre la web y la base de datos.
* Una base de datos segura y organizada para registrar y gestionar la información de los clientes y sus citas.

# Anàlisi Inicial

**(Quines necessitats té el client? Quins problemes solucioneu?)** **Ejemplo:**

**El client necessita gestionar de manera eficient els seus clients i vehicles, oferir la possibilitat de demanar cita online i tenir còpies de seguretat de les seves dades.**

## Necesidades del cliente

* **Gestión eficiente de clientes y vehículos:** El cliente necesita un sistema que le permita mantener una base de datos actualizada de sus clientes y vehículos, incluyendo información como contactos, historiales de mantenimiento y reparaciones.
* **Solicitud de cita en línea:** El cliente necesita la capacidad de permitir a sus clientes solicitar cita en línea, lo que les permitirá planificar sus visitas de manera más eficiente.
* **Copias de seguridad de datos:** El cliente necesita un sistema que le permita tomar copias de seguridad de sus datos, lo que les permitirá proteger sus datos en caso de una falla o pérdida de información.

# Model de Dades

## Diagrama ER

Diagrama

El contenido generado por IA puede ser incorrecto.

## Esquema relacional

Diagrama

El contenido generado por IA puede ser incorrecto.

## Script SQL

DROP DATABASE IF EXISTS `taller-mecànic`;

CREATE DATABASE `taller-mecànic`;

USE `taller-mecànic`;

-- CREATE TABLES

-- Creación de la tabla 'clients'

CREATE TABLE `Clients` (

    `IdClient`  *INT*, -- Clave primaria

    `nom`       *VARCHAR*(64) NOT NULL,

    `DNI`       *CHAR*(9) NOT NULL,

    `telefon`   *VARCHAR*(20) NOT NULL,

    `email`     *VARCHAR*(64) NOT NULL

);

-- Creación de la tabla 'cita'

CREATE TABLE `Cita` (

    `IdCita`    *INT*, -- Clave primaria

    `IdClient`  *INT* NOT NULL, -- Clave foránea hacia 'clients'

    `data`      *DATE* NOT NULL,

    `servei`    *VARCHAR*(64) NOT NULL,

    `IdVehicle` *INT* -- Clave foránea hacia 'Vehicles'

);

-- Creación de la tabla 'Vehicles'

CREATE TABLE `Vehicles` (

    `IdVehicle` *INT*, -- Clave primaria

    `IdClient`  *INT*, -- Clave foránea opcional hacia 'clients'

    `matricula` *VARCHAR*(20) NOT NULL,

    `model`     *VARCHAR*(64) NOT NULL,

    `any`       *INT* NOT NULL

);

-- PRIMARY KEYS

ALTER TABLE `Clients`

    MODIFY `IdClient` *INT* AUTO\_INCREMENT,

    ADD CONSTRAINT PK\_IdClient PRIMARY KEY (`IdClient`);

ALTER TABLE `Cita`

    MODIFY `IdCita` *INT* AUTO\_INCREMENT,

    ADD CONSTRAINT PK\_IdCita PRIMARY KEY (`IdCita`);

ALTER TABLE `Vehicles`

    MODIFY `IdVehicle` *INT* AUTO\_INCREMENT,

    ADD CONSTRAINT PK\_IdVehicle PRIMARY KEY (`IdVehicle`);

-- FOREIGN KEYS (Cambiar lo de abajo)

ALTER TABLE

    ADD CONSTRAINT FK\_Cita\_Cliente FOREIGN KEY (`IdClient`)

        REFERENCES clients(`IdClient`),

    ADD CONSTRAINT FK  FOREIGN KEY (`IdVehicle`)

REFERENCES Vehicles(`IdVehicle`)

-- vehicles

FOREIGN KEY (`IdClient`) REFERENCES clients(`IdClient`)

-- UNIQUE

ALTER TABLE `Clients`

    ADD CONSTRAINT UQ\_Client\_DNI UNIQUE (`DNI`),

    ADD CONSTRAINT UQ\_Client\_Email UNIQUE (`email`);

ALTER TABLE `Vehicles`

    ADD CONSTRAINT UQ\_Vehicle\_Matricula UNIQUE (`matricula`);

# Desenvolupament de la Solució

## API

* Llenguatge utilitzat: Python
* Framework: Flask
* Funcionalitats:
  + GET /vehicles
  + POST /appointments
  + GET /appointments

## Web

* HTML/CSS/JS Bàsic
* Incloure captures de pantalles (en format portàtil i mòbil)
* Descriure operativa bàsica per demanar cita

## Scripts

* Script de còpia de seguretat de la base de dades
* Script de comprovació d'estat dels contenidors

# Infraestructura de Sistemes

**(Explicació de quins contenidors heu creat i com s'han configurat.)**

**(Explicació dels scripts elaborats i demostració de funcionament.)**

*Exemple:*

* Contenidor per MariaDB
* Contenidor per API Flask
* Contenidor per pàgina web
* Execució backup de BDs
* Execució comprovació d’estat dels contenidors.

# Problemes i Solucions

**(Quins problemes heu tingut i com els heu resolt.)**

*Exemple:*

Vam tenir problemes per connectar l'API amb MariaDB i vam resoldre-ho revisant el docker-compose.yml.

# Conclusions

**(Què heu après amb aquest projecte? Què milloraríeu?)**

*Exemple:*

Hem après a connectar una API amb una base de dades i desplegar serveis en contenidors Docker. Milloraríem la validació d’errors a la web.

# Annexos

* Enllaços a repositori de codi
* Captures de pantalla del projecte funcionant
* Fitxers addicionals