# Trabajo Práctico de Bases de Datos

### 1. Introducción

El proyecto integra el manejo de elementos en dos tipos de bases de datos diferentes: relacional y NoSQL. La interfaz desarrollada permite a los usuarios interactuar de manera eficiente con ambas bases de datos a través de una API REST. Este informe describe la arquitectura y funcionalidades implementadas.

## 2. Descripción del Proyecto

## 2.1. Arquitectura

- Frontend: Utilizamos HTML, CSS y JavaScript para la gestión de elementos.
- Backend: Utilizamos el framework de Django en Python para conectar el frontend y las bases de datos.

#### Bases de Datos:

- Relacional: Para almacenar elementos con esquemas bien definidos, utilizamos MySQL.
- NoSQL: Para elementos con datos más flexibles, utilizamos MongoDB

#### 2.2. Funcionalidades

#### Gestión de Elementos Relacionales:

• Crear, leer, editar y eliminar elementos con atributos nombre, descripción y valor.

#### Gestión de Elementos NoSQL:

• Similar a la relacional, pero con un campo adicional metadata, que nos permite guardar datos dinámicos en formato JSON.

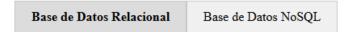
## 3. Demostración Técnica

- 3.1. Funcionamiento de la Base de Datos Relacional
  - Creación



Lectura

## CRUD con JavaScript y Django - Bases de Datos



## Base de Datos Relacional (SQLite)

### **Agregar Producto**



- Producto1 Este es el producto 1 \$100.00 Editar Eliminar
- Actualización



## CRUD con JavaScript y Django - Bases de Datos

Base de Datos Relacional Base de Datos NoSQL

## Base de Datos Relacional (SQLite)

#### **Agregar Producto**

Nombre Descripción Valor Agregar Producto

#### Lista de Productos

- Producto1 Este es el producto 1 \$1000000.00 Editar Eliminar
- Eliminación



## CRUD con JavaScript y Django - Bases de Datos

Base de Datos Relacional Base de Datos NoSQL

## Base de Datos Relacional (SQLite)

#### **Agregar Producto**

Nombre Descripción Valor Agregar Producto

### 3.2. Funcionamiento de la Base de Datos No Relacional

Creación



Base de Datos NoSQL Base de Datos Relacional

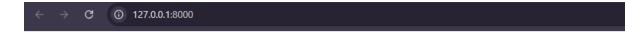
## Base de Datos NoSQL (MongoDB)

#### **Agregar Producto**



- Producto 42 Este es un producto con información especial \$42000 Editar Eliminar
- Actualización





## CRUD con JavaScript y Django - Bases de Datos

Base de Datos Relacional Base de Datos NoSQL

### Base de Datos NoSQL (MongoDB)

#### Agregar Producto

Nombre Descripción Valor Metadata (JSON Agregar Producto

#### Lista de Productos

- Producto 42 Este es un producto con información especial cambiada \$42000000 Editar Eliminar
- Eliminación



## CRUD con JavaScript y Django - Bases de Datos

Base de Datos Relacional Base de Datos NoSQL

## Base de Datos NoSQL (MongoDB)

#### **Agregar Producto**

Nombre Descripción Valor Metadata (JSON Agregar Producto

### 3.2. Funciones Clave del Sistema

#### Interacción con la API REST:

• Métodos HTTP (GET, POST, PUT, DELETE) para operaciones CRUD.

#### Validación de Datos:

- Validación de metadata como JSON válido en NoSQL.
- Manejo de errores y alertas informativas al usuario.

### Seguridad:

• Protección contra ataques CSRF mediante el uso de tokens CSRF en solicitudes.

## 4. Conclusión

El proyecto muestra cómo combinar bases de datos relacionales y NoSQL en un entorno integrado y flexible, que permite adaptarse a los diferentes datos aprovechando sus diferencias.