Arquitectura del sistema - WEB

# Editor:

Reinaldo Espinel Torres

Plataforma de Orquestación para Automatizaciones RPA

**27/05/2024**

# Detalles del documento

|  |  |
| --- | --- |
| Seguridad | Publico |
| Version | Version 1.1 |
| Editor | Reinaldo Espinel Torres |

# Resumen

Este documento incluye los requerimientos del Sistema.

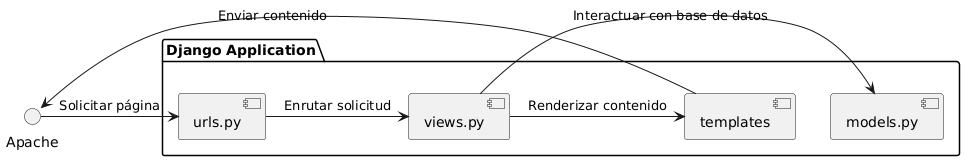
# Introducción

Este documento describe la arquitectura del orquestador de WEB desarrollado en Django. Incluye una visión general del sistema, diagramas de arquitectura, componentes principales, interacciones y especificaciones técnicas detalladas.

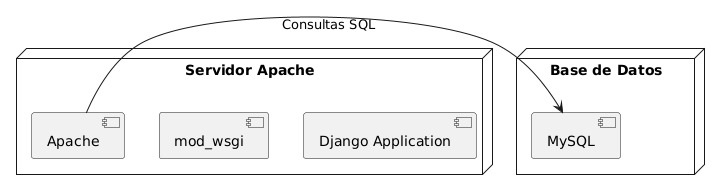
# Descripción General de la Arquitectura

La arquitectura del sistema se basa en un modelo de aplicación web típica, estructurada en varios componentes y capas. El siguiente diagrama proporciona una visión general de los principales componentes del sistema y sus interacciones.

**Diagrama de componentes**



**Diagrama de despliegue**



1. **Componentes del Sistema**

A continuación, se describen los principales componentes del sistema y sus responsabilidades.

**Servidor Apache**

**• Descripción:** Un servidor web que maneja las solicitudes HTTP/S entrantes y ejecuta la aplicación Django**.**

**• Rol:** Actúa como un proxy inverso y servidor de aplicaciones WSGI, sirviendo contenido estático y dinámico**.**

**• Dependencias:**

* **Django Application:** La aplicación web que maneja**.**
* **mod\_wsgi**: Módulo Apache para servir aplicaciones Django**.**

**Django Application**

**• Descripción: El núcleo de la aplicación web que incluye varios componentes.**

**• Rol: Procesa la lógica de negocio, maneja la interacción con la base de datos, y renderiza las plantillas HTML.**

**• Componentes:**

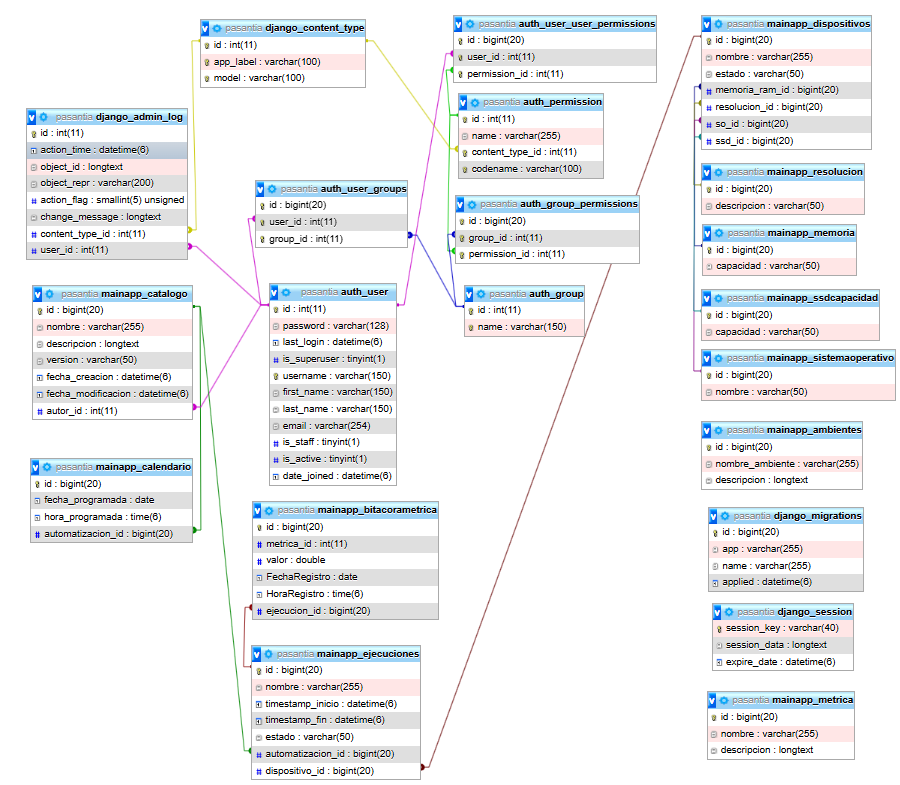
* **urls.py:** Define las rutas y conecta URLs con vistas.
* **views.py:** Controla la lógica de la aplicación, manejando solicitudes y respuestas.
* **models.py:** Define la estructura de la base de datos y maneja las interacciones con ella.
* **templates:** Contiene las plantillas HTML que se renderizan para el usuario.
* **forms.py:** Define formularios para recibir y validar datos del usuario.

1. **Especificaciones técnicas**

**Tecnologías utilizadas:**

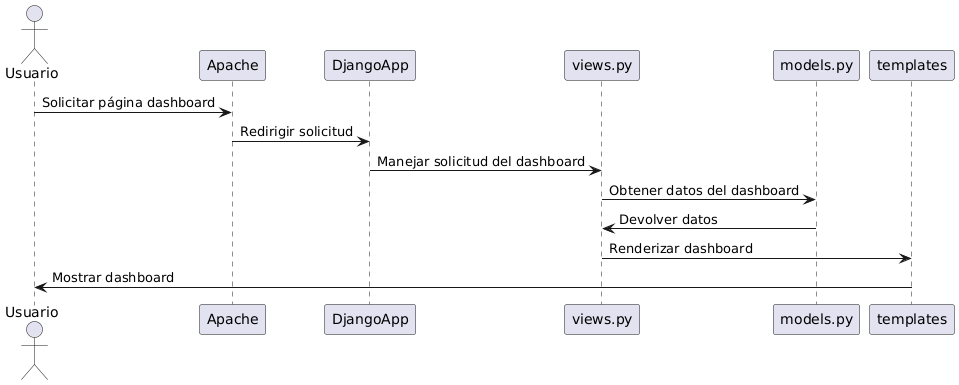
* **Lenguaje de Programación**: Python
* **Framework**: Django
* **Servidor**: Apache (para servir tanto la aplicación como el servidor web).
* **Base de Datos**: MySQL

##### **Esquema de Base de Datos**

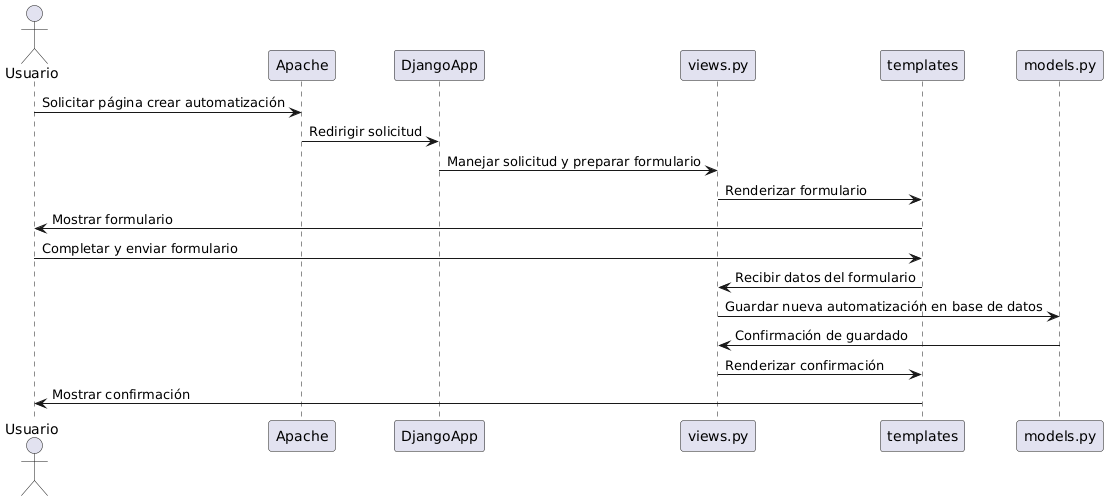


1. **Diagrama de secuencia**

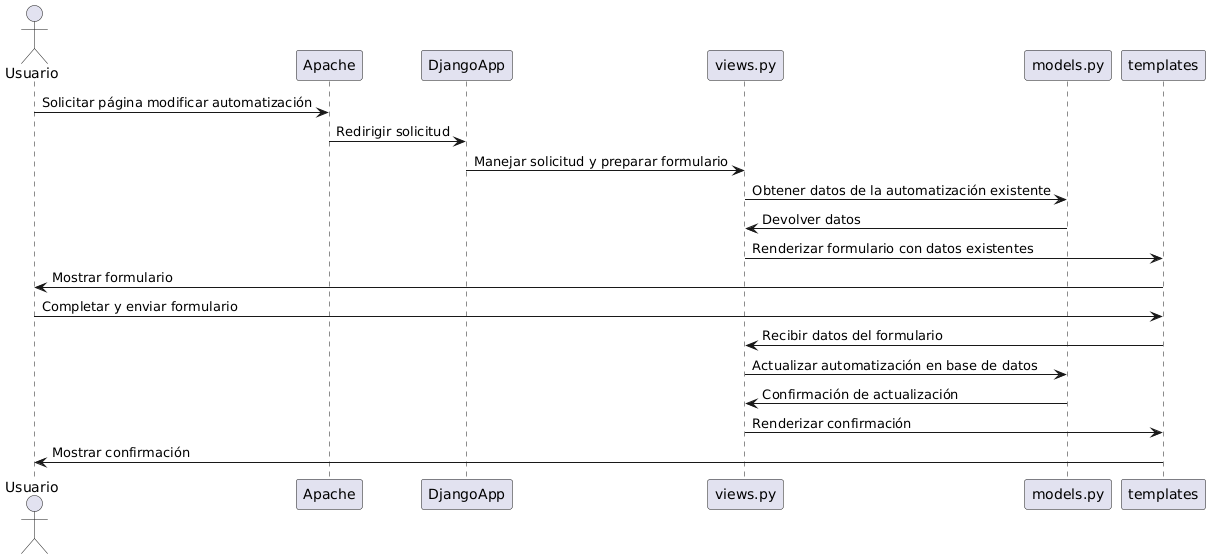
**Visualización del Dashboard**



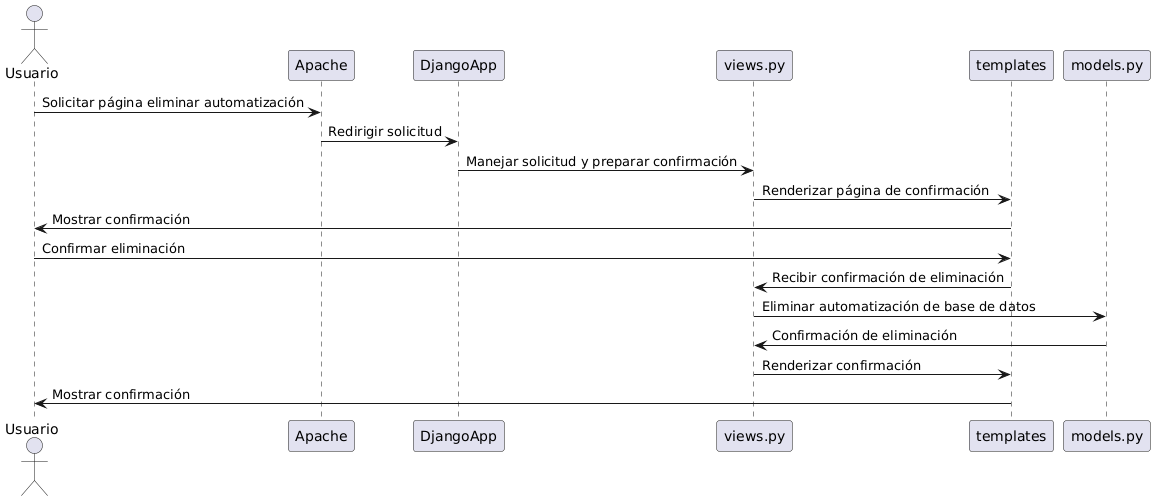
**Creación de una Nueva Automatización**



**Modificación de una Automatización**



**Eliminación de una Automatización**



**Visualización de Detalles de una Automatización**

