

INFORME DE ARQUITECTURA 4+1

Angeles Scheduler



Contenido

[**1. Introducción 2**](#_heading=h.jpbg8t1zihjb)

[2. Vista de Desarrollo 3](#_heading=h.1fob9te)

[Descripción del flujo de interacción: 3](#_heading=h.8nwii49vxjnv)

[3. Vista de Procesos 4](#_heading=h.3znysh7)

[4. Vista Física 5](#_heading=h.2et92p0)

[Despliegue físico: 5](#_heading=h.g1g3cq322wof)

[Descripción del flujo de despliegue: 5](#_heading=h.b0wcq7150436)

[5. Escenarios (Casos de Uso) 6](#_heading=h.tyjcwt)

[Escenario 1: Registro de una Venta 6](#_heading=h.60w86icubf0e)

[Escenario 2: Registro de una Cita 6](#_heading=h.pdh4btrmal2u)

[6. Conclusión 7](#_heading=h.3dy6vkm)

[7. Anexos 7](#_heading=h.1t3h5sf)

# 1. Introducción

Este documento describe la arquitectura del sistema **Ángeles Scheduler**, que utiliza el modelo 4+1 vistas adaptado para gestionar ventas y citas en línea. El sistema está diseñado para ser escalable, seguro y accesible, brindando una experiencia optimizada tanto para usuarios finales como para los equipos de desarrollo y operaciones.

#### **Objetivos del sistema:**

* **Facilitar la gestión de ventas y citas:** Proporcionar una plataforma intuitiva para registrar y gestionar transacciones de manera eficiente.
* **Garantizar la seguridad de los datos y transacciones:** Asegurar la integridad y confidencialidad de la información mediante protocolos seguros como HTTPS y WebSocket.
* **Ofrecer alta disponibilidad y escalabilidad:** Diseñar el sistema para soportar múltiples usuarios concurrentes y garantizar un funcionamiento ininterrumpido.
* **Optimizar la experiencia de usuario:** Ofrecer una interfaz fluida e interactiva que permita realizar procesos clave con facilidad.

#### **Interesados:**

* **Desarrolladores y diseñadores (Vista Lógica y de Desarrollo):**Encargados de la implementación del frontend y backend, además de diseñar componentes para la interacción del sistema con los usuarios.
* **Equipos de infraestructura y operaciones (Vista Física):**Responsables del despliegue en **Vercel**, monitoreo de **Supabase** y configuración de pipelines de integración y entrega continua (CI/CD).
* **Arquitectos de software y probadores (Vista de Procesos y Escenarios):**Supervisan el diseño general del sistema, los flujos de datos y realizan pruebas para garantizar la calidad del software.
* **Clientes y usuarios finales (Vista de Escenarios):**Usuarios que interactúan con la plataforma para gestionar ventas y citas, beneficiándose de un sistema eficiente y fácil de usar.

# 2. Vista de Desarrollo

Esta vista muestra cómo se organiza el código del sistema.

* **Interfaz (Frontend)**:
  + Aplicación web llamada "WEB Ángeles Scheduler", accesible por los usuarios a través de un navegador web.
  + La comunicación con el servidor se realiza mediante el protocolo HTTPS.
* **Servidor (Backend)**:
  + Implementado en **Vercel**, incluye las siguientes capas:
    - **Capa de Vista**: Gestiona las interacciones y respuestas hacia la interfaz de usuario.
    - **Capa de Controlador**: Maneja las peticiones del frontend y coordina la lógica del negocio.
    - **Capa de Modelo**: Se encarga de interactuar con la base de datos a través de APIs REST y Websockets.
* **Base de Datos**:
  + Implementada utilizando **Supabase** para la persistencia de datos.
  + Admite comunicación bidireccional con el servidor mediante APIs REST y Websockets.

### **Descripción del flujo de interacción:**

* Los usuarios interactúan con la interfaz web (WEB Ángeles Scheduler).
* La interfaz envía solicitudes al servidor mediante HTTPS.
* El servidor procesa las solicitudes en sus capas respectivas:
  + Vista para gestionar la presentación.
  + Controlador para manejar la lógica.
  + Modelo para acceder y actualizar datos en la base de datos.
* El servidor se comunica con la base de datos a través de **APIs REST** y **Websockets**, garantizando una sincronización eficiente de los datos.

**Ver Anexo A**

# 3. Vista de Procesos

Esta vista describe el comportamiento dinámico del sistema en tiempo de ejecución.

**Registro de Ventas**:

* **Inicia**: Cuando el usuario solicita crear una venta desde la vista.
* **Flujo**:
  1. La vista envía una **solicitud de nueva venta** al controlador.
  2. El controlador procesa los datos de la venta y los envía al modelo.
  3. El modelo guarda los datos en la base de datos y recibe una **confirmación de los datos guardados**.
  4. El modelo envía una **respuesta de venta creada** al controlador.
  5. El controlador actualiza la vista.
* **Finaliza**: Cuando el usuario recibe una notificación de que la venta ha sido creada exitosamente.

**Registro de Citas**:

* **Inicia**: Cuando el usuario solicita registrar una nueva cita desde la vista.
* **Flujo**:
  1. La vista envía una **solicitud de nueva cita** al controlador.
  2. El controlador procesa los datos de la cita y los envía al modelo.
  3. El modelo guarda los datos en la base de datos y recibe una **confirmación de los datos guardados**.
  4. El modelo envía una **respuesta de cita registrada** al controlador.
  5. El controlador actualiza la vista.
* **Finaliza**: Cuando el usuario recibe una notificación de que la cita ha sido registrada exitosamente.

**Anexo B**

# 4. Vista Física

Esta vista muestra la infraestructura sobre la que se despliega el sistema.

### **Despliegue físico:**

1. **Plataforma**:
   * La aplicación "Ángeles Scheduler" está desplegada en **Vercel**, que gestiona tanto el frontend como el backend en un entorno integrado.
2. **Componentes principales**:
   * **Vista, Controlador y Modelo**: Forman parte del sistema central dentro de la plataforma Vercel, procesando las interacciones de los usuarios y gestionando la lógica de negocio.
   * **Vercel Logs y Build**:
     + El sistema de **build** en Vercel gestiona la construcción y despliegue del código a partir de la rama principal del repositorio.
     + Los **logs** proporcionan trazabilidad y monitoreo en tiempo real.
3. **Base de datos**:
   * Implementada en **Supabase**, que permite la comunicación con el modelo mediante **Websockets** y conexiones TCP para manejar datos de manera eficiente y en tiempo real.
4. **Integración Continua/Despliegue Continuo (CI/CD)**:
   * Basado en un repositorio (como GitHub) conectado directamente al sistema de build de Vercel.
   * Las variables de entorno se configuran a través del activador del proceso de despliegue.

### **Descripción del flujo de despliegue:**

* **Repositorio CI/CD**: El código en la rama principal desencadena automáticamente un proceso de despliegue en Vercel.
* **Activador**: Configura variables de entorno críticas y gestiona la integración con otros servicios.
* **Logs y Build**: Proveen un sistema robusto para monitorear, depurar y asegurar el despliegue correcto.

**Ver Anexo C**

# 5. Escenarios (Casos de Uso)

Esta vista integra las anteriores mediante escenarios típicos de uso del sistema.

### **Escenario 1: Registro de una Venta**

1. **Acceso al sistema**:
   * El usuario ingresa a la plataforma **Ángeles Scheduler** y selecciona la opción para registrar una venta (**Vista**).
2. **Consulta de datos**:
   * La **vista** interactúa con el **controlador** para solicitar la creación de una nueva venta.
3. **Procesamiento**:
   * El **controlador** envía la solicitud al **modelo**, que procesa los datos y los guarda en la base de datos alojada en **Supabase**.
4. **Confirmación**:
   * La base de datos confirma la operación a través de un **WebSocket**, garantizando que la transacción fue exitosa.
5. **Notificación al usuario**:
   * El sistema actualiza la vista, mostrando una notificación de que la venta ha sido registrada correctamente.

### **Escenario 2: Registro de una Cita**

1. **Acceso al sistema**:
   * El usuario selecciona la opción para registrar una cita en la plataforma (**Vista**).
2. **Procesamiento inicial**:
   * La **vista** envía una solicitud al **controlador** para registrar la cita.
3. **Interacción con la base de datos**:
   * El **modelo** procesa los datos de la cita y los almacena en **Supabase**.
   * Se confirma el almacenamiento mediante una comunicación **TCP**.
4. **Actualización y notificación**:
   * El sistema actualiza la vista con la información de la cita registrada y notifica al usuario del éxito del proceso..

# 6. Conclusión

El sistema de e-commerce propuesto ha sido diseñado para ofrecer una arquitectura flexible, escalable y segura, capaz de manejar grandes volúmenes de transacciones y proporcionar una experiencia óptima para los usuarios. Las cinco vistas de la arquitectura permiten abordar las distintas preocupaciones de los interesados, asegurando que el sistema sea robusto y esté bien documentado.

# 7. Anexos

#### **Anexo A: Diagrama de Componentes - Vista de Desarrollo**

Consulte el archivo "Diagrama de componentes.png".

#### **Anexo B: Diagrama de Secuencia - Vista de Procesos**

Consulte el archivo "Diagrama de secuencia.png".

#### **Anexo C: Diagrama de Despliegue - Vista Física**

Consulte el archivo "Diagrama de despliegue.png".