

Plan de despligue

Angeles Scheduler



11 de noviembre de 2024

Equipo 2

Felipe Concha – Jimmy Muñoz – Cristian Molina

Contenido

[1. Introducción 3](#_Toc184684653)

[2. Objetivos del Plan de Despliegue 3](#_Toc184684654)

[3. Equipo de Despliegue 3](#_Toc184684655)

[3. Requisitos Previos 4](#_Toc184684656)

[3.1. Requisitos técnicos 4](#_Toc184684657)

[3.2. Requisitos funcionales del sistema 4](#_Toc184684658)

[4. Flujo General del Despliegue 4](#_Toc184684659)

[4.1. Proceso inicial 4](#_Toc184684660)

[4.2. Despliegue continuo 4](#_Toc184684661)

[5. Pasos Detallados del Despliegue 5](#_Toc184684662)

[5.1. Configuración del Repositorio GitHub 5](#_Toc184684663)

[5.2. Configuración en Vercel 5](#_Toc184684664)

[5.3. Validación Previa al Despliegue 5](#_Toc184684665)

[5.4. Despliegue en Producción 5](#_Toc184684666)

[5.5. Post-Despliegue 5](#_Toc184684667)

[6. Conclusión 6](#_Toc184684668)

# 1. Introducción

El presente informe describe el Plan de Despliegue diseñado para garantizar la implementación exitosa de Angeles scheduler. Este plan establece las estrategias, actividades y recursos necesarios para la puesta en marcha del sistema, asegurando su correcta operatividad y aceptación por parte de los usuarios finales.

# 2. Objetivos del Plan de Despliegue

1. Garantizar un despliegue eficiente, seguro y escalable del sistema web.
2. Facilitar actualizaciones continuas mediante la integración automatizada de cambios desde GitHub.
3. Asegurar la disponibilidad del sistema para usuarios finales en un entorno de producción confiable.

# **3. Equipo de Despliegue**

| **Rol** | **Responsabilidades** | **Nombre** |
| --- | --- | --- |
| Líder de Proyecto | Supervisar y coordinar todas las actividades de despliegue. | Felipe Concha |
| Desarrollador | Configurar el entorno de producción y resolver incidencias. | Cristian Molina |
| Tester | Realizar pruebas finales en el entorno de producción. | Jimmy Muñoz |

# 3. Requisitos Previos

## 3.1. Requisitos técnicos

| **Requisito** | **Descripción** |
| --- | --- |
| Repositorio GitHub | Código fuente alojado en repositorio accesible |
| Cuenta en Vercel | Configurada y vinculada al repositorio |
| Framework | Next.js |
| Base de datos | Supabase |
| Lenguaje | TypeScript |

**Configuración de variables de entorno:**

* Claves para la conexión a la base de datos (Supabase URL y clave secreta)
* Cualquier otra API o clave utilizada por el sistema

## 3.2. Requisitos funcionales del sistema

* Gestión de empleados, clientes, citas, ventas y comisiones
* Registro de servicios y generación de reportes detallados
* Autenticación segura basada en correo electrónico y contraseña

# 4. Flujo General del Despliegue

## 4.1. Proceso inicial

* Configuración del repositorio en GitHub
* Configuración de Vercel como plataforma de despliegue

## 4.2. Despliegue continuo

Cada vez que se realiza un push o se aprueba un Pull Request en la rama principal, Vercel ejecuta un despliegue automático.

# 5. Pasos Detallados del Despliegue

## 5.1. Configuración del Repositorio GitHub

**Verificación del código fuente:**

* Confirma que la aplicación funciona correctamente en un entorno local
* Asegúrate de incluir un archivo .env.example con las variables de entorno necesarias

**Configuración de ramas:**

* Establece main como la rama principal para despliegues en producción
* Crea ramas separadas para desarrollo y nuevas características

## 5.2. Configuración en Vercel

**Vinculación del Repositorio:**

* Inicia sesión en Vercel y selecciona "New Project"
* Conecta el repositorio correspondiente desde GitHub

**Configuración del Proyecto:**

* Framework Preset: Next.js
* Variables de Entorno: Configuración del archivo .env.local
* Dominio Personalizado (opcional)

## 5.3. Validación Previa al Despliegue

* Pruebas de Previsualización
* Verificación de funcionalidades clave
  + Acceso al sistema
  + Gestión de empleados, clientes, citas y ventas
  + Generación de reportes

## 5.4. Despliegue en Producción

* Push a la Rama Principal
* Monitoreo del Despliegue

## 5.5. Post-Despliegue

* Pruebas finales en producción
* Configuración de alertas en Vercel

# 6. Conclusión

El despliegue del sistema en Vercel utilizando GitHub como repositorio garantiza un flujo de trabajo eficiente y automatizado. La integración continua asegura que los cambios realizados por los desarrolladores sean revisados y aplicados rápidamente al entorno de producción, manteniendo la calidad del sistema y la satisfacción del usuario final.