

# Manual de Instalación

---

## 1. Introducción

### 1.1 Propósito de este documento

Este documento proporciona instrucciones detalladas para la instalación del sistema. Cubre todos los pasos necesarios para instalar y configurar correctamente el sistema, asegurando que funcione adecuadamente en el entorno previsto.

### 1.2 Identificación

El sistema se identifica como una solución integral para la gestión y operación de la tienda en línea Rosita. Este documento es relevante para la versión 1.0 del sistema.

### 1.3 Información de referencia

- Manual del Usuario: Proporciona una guía completa sobre el uso del sistema para administradores y usuarios finales.
- Documentación Técnica: Incluye detalles sobre la arquitectura del sistema, APIs y otros aspectos técnicos.

### 1.4 Puntos de contacto

Para obtener asistencia adicional, póngase en contacto con el equipo de soporte en:

- Correo electrónico: [joraazam@espol.edu.ec](mailto:joraazam@espol.edu.ec), [lizpevacacela@espol.edu.ec](mailto:lizpevacacela@espol.edu.ec), [alenocot@espol.edu.ec](mailto:alenocot@espol.edu.ec)

## 2. Plan de Instalación

### 2.1 Alcance

Este plan de instalación cubre todos los aspectos necesarios para la correcta implementación del sistema en el entorno designado.

### 2.2 Entorno

El backend se desplegará en render.com y el frontend en Netlify. El entorno incluye servidores configurados para ejecutar Django y Angular 17.

### 2.3 Tareas

Las tareas incluyen la preparación del entorno, la instalación del backend y frontend, y la verificación de la correcta operación del sistema.

## 2.4 Seguridad

Se deben aplicar todas las medidas de seguridad necesarias durante la instalación, incluyendo mantener credenciales seguras.

## 2.5 Criterios de Entrada y Salida

Criterios de entrada: todos los pre-requisitos de software y hardware deben estar cumplidos. Criterios de salida: el sistema debe estar completamente instalado y operativo.

## 2.6 Procedimiento de Respaldo

Se recomienda realizar un respaldo completo del entorno existente antes de iniciar el proceso de instalación.

## 2.7 Procedimiento de Prueba de Instalación

Después de la instalación, se deben realizar pruebas para garantizar que todos los componentes funcionen correctamente.

# 3. Requisitos del Sistema

## 3.1 Descripción General de la Instalación del Sistema

Esta sección describe los requisitos generales para la instalación del sistema.

## 3.2 Requisitos de Software

Incluye Node.js, Angular CLI, Python y Django, entre otros.

## 3.3 Requisitos de Hardware

Servidor con al menos 4 GB de RAM y 2 núcleos de CPU para backend,y frontend.

## 3.4 Sistemas Operativos

El sistema es compatible con los siguientes sistemas operativos:

- Windows 10 o superior
- Ubuntu 20.04 LTS o superior
- macOS Catalina o superior

## 3.5 Soporte de Base de Datos

El backend soporta las siguientes bases de datos:

- PostgreSQL 13 o superior
- SQLite 3 (solo para desarrollo)

### 3.6 Problemas Conocidos

No hay problemas conocidos en esta versión.

## 4. Instalación del Sistema

### 4.1 Requisitos Previos

Asegúrese de que todos los requisitos de software y hardware estén cumplidos antes de comenzar la instalación. Esto incluye la instalación de Node.js, Angular CLI, Python y la base de datos.

### 4.2 Preparación para instalar componentes

Descargue y prepare todos los archivos necesarios para la instalación, incluyendo el código fuente del backend y frontend desde los repositorios correspondientes.

### 4.3 Inicio de la Instalación

Inicie la instalación del backend y frontend según las instrucciones proporcionadas.

#### 4.3.1 Instalacion del Frontend

Este manual proporciona los pasos necesarios para instalar tanto el frontend como el backend de la tienda online. Asegúrate de seguir cada sección de manera secuencial para garantizar una instalación correcta.

El frontend es la interfaz de usuario de la tienda online. Está desarrollado en Angular y proporciona una experiencia interactiva para los clientes y administradores de la tienda.

### Proceso de Instalación

#### Pre-requisitos

Antes de comenzar, asegúrate de tener instaladas las siguientes herramientas:

- Node.js: Versión 20.14
- NPM: Versión 10.7
- Angular CLI: Versión 17.3.11

#### Instalación

1. Clonar el proyecto

```
git clone https://github.com/StoreDevApps/ft_admin.git
```

2. Instalar dependencias

```
npm install
```

### Otros comandos de administración

1. Para iniciar en modo desarrollo

```
ng serve
```

2. Para compilar archivos para desplegar

```
ng build
```

### Otros comandos de desarrollo

1. Crear componente:

```
ng generate component componentes/Componente
```

2. Crear componentes para la página de administración:

```
ng generate component componentes/pages/admin/admin-Componente
```

3. Crear componentes para la página de trabajadores:

```
ng generate component componentes/pages/worker/trabajadores-Componente
```

4. Crear componentes para la página de clientes:

```
ng generate component componentes/pages/user/clientes-Componente
```

5. Crear componentes para la página sin login:

```
ng generate component componentes/pages/public/public-Componente
```

6. Crear servicios:

```
ng generate service services
```

### 4.3.2 Instalacion del Frontend

El backend es la API que gestiona la lógica del negocio, la base de datos, y la autenticación para la tienda online. Está desarrollado en Django.

#### Pre-requisitos

Antes de comenzar, asegúrate de tener instaladas las siguientes herramientas:

- Python: Versión 3.12.4

- Pip: El instalador de paquetes para Python.

1. Clonar repositorio

```
git clone https://github.com/StoreDevApps/bk_store.git
```

2. Instalar virtualenv si no está instalado

```
pip install virtualenv
```

3. Crear un entorno virtual

```
python -m virtualenv env
```

4. Activar el entorno virtual

- En Windows:

```
.\env\Scripts\activate
```

- En Linux/MacOS:

```
source env/bin/activate
```

5. Instalar los requerimientos

```
pip install -r requirements.txt
```

6. Migrar la base de datos

```
python manage.py migrate
```

7. Iniciar el servidor de Django

```
python manage.py runserver
```

### Otros comandos de administración

1. Aplicar los cambios de los modelos creados:

```
python manage.py makemigrations
```

2. Crear superusuarios para Django Admin:

```
python manage.py createsuperuser
```

3. Ejecutar tests:

```
python manage.py test
```

4. Actualizar el archivo `requirements.txt` después de instalar nuevos paquetes:

```
pip freeze > requirements.txt
```

## 5. Apéndices

### Glosario de Términos

Definiciones de términos clave utilizados en este documento:

- Backend: Parte del sistema que gestiona la lógica de negocio, bases de datos y la funcionalidad del servidor.
- Frontend: Parte del sistema que interactúa con el usuario, proporcionando la interfaz gráfica y la experiencia de usuario.
- Render.com: Servicio de alojamiento utilizado para desplegar aplicaciones backend.
- Netlify: Plataforma utilizada para desplegar y alojar aplicaciones frontend estáticas.

### Acrónimos y Abreviaturas

Lista de acrónimos y sus significados:

- CLI: Command Line Interface (Interfaz de Línea de Comandos).
- API: Application Programming Interface (Interfaz de Programación de Aplicaciones).
- RAM: Random Access Memory (Memoria de Acceso Aleatorio).