

**BEJI**

## **Plan de instalación**

Versión: 0100

Fecha: 30/09/2025



# 1. Introducción

El Sistema de Inventario para Almacén de Calzado (BEJI) es una solución integral diseñada para gestionar productos, usuarios, ventas, proveedores. Incluye componentes web y móviles que permiten el control eficiente del negocio tanto en tienda física como en dispositivos móviles.

Este plan describe las actividades necesarias para instalar, configurar y desplegar el sistema en sus entornos productivos, asegurando disponibilidad, seguridad e integridad de los datos.

Se siguen estándares de mejores prácticas (ITIL, ISO 20000) y se establecen lineamientos para una transición controlada de desarrollo a producción.

## 1.1 Objetivo

### ***1.1.1 Técnicos:***

- Instalar y configurar los componentes del sistema:
- Backend Web: Laravel
- Frontend Web: Angular
- Backend Móvil: Spring Boot
- Frontend Móvil: Android
- Base de Datos: MySQL
- Integrar con repositorio GitHub para despliegue controlado

### ***1.1.2 Operativos:***

- Minimizar el tiempo de inactividad durante la migración.
- Validar funcionamiento con pruebas manuales y automatizadas.

### ***1.1.3 Institucionales:***

- Garantizar SLA de disponibilidad del 99,5% en horario laboral.
- Capacitar usuarios finales en el uso del sistema.

## 2. Alcance

Incluye:

✓ **Componentes:**

- Backend Laravel (PHP 8.2, Composer 2.8)
- Frontend Angular (10.9.0)
- Backend Spring Boot (Java 17)
- App Móvil Android (SDK 33)
- Base de datos MySQL 8.0
- Repositorio GitHub

✓ **Entornos:**

- Servidor de producción (AWS EC2 / On-Premise).
- Dispositivos móviles (Android 8.0+).

✓ **Procesos:**

- Despliegue continuo (GitHub Actions / scripts Bash)
- Pruebas de integración post-instalación

Excluye:

- Migración de datos históricos de otros sistemas
- Desarrollo de nuevas funcionalidades
- Soporte a dispositivos iOS

## 3. Responsabilidades

Rol	Tareas
Administrador de Sistemas	Configuración de servidores, firewall, certificados SSL
DevOps	Despliegue de Laravel, Angular y Spring Boot.
DBA	Configuración de MySQL, backups y restauración
Líder de Proyecto	Coordinación con usuarios y aprobación de despliegue

## 4.FASE 1: PREPARACIÓN DEL ENTORNO

(Duración: 2 días)

### Objetivo

- Asegurar que todos los requisitos previos estén cumplidos antes del despliegue.

### 4.1 VERIFICACIÓN DE HARDWARE

- Servidor de Aplicaciones:  
CPU: 8 núcleos  
RAM: 16 GB  
SSD: 100 GB
- Servidor de BD:  
250 GB SSD
- Conectividad:  
Latencia < 50ms  
Ancho de banda mínimo 100 Mbps

### 4.2 INSTALACIÓN DE SOFTWARE BASE

Componentes criticos:

Software	Versión	Instrucciones
Composer	2.8	<a href="#">Descarga e instalación Composer</a>
Node.js	18	<a href="#">Descarga e instalación Node.js</a>
Java	24	<a href="#">Descarga e instalación de java</a>
Intellij Community Edition	3.4.1	<a href="#">Descarga e instalación de intellij Community</a>
MySQL	8.0	<a href="#">Descarga e instalación de MySQL WORKBENCH</a>
Git	2.50.0	<a href="#">Descarga e instalación Git</a>

## 4.3 CONFIGURACIÓN INICIAL DE BASE DE DATOS (Responsable: DBA)

Pasos:

1. Crear base de datos:
  - CREATE DATABASE beji
  - USE beji
2. Script de las tablas de base de datos:
  - [Script](#)

## 4.4 PREPARACIÓN DE REPOSITORIOS (Responsable: DevOps)

1. Instalar [Git](#)
2. Configuración global del usuario de git  
En el bash de git configurar "user.name" y "user.email"
3. Clonar el repositorio:  
→ <https://github.com/Beji2024/Proyecto.git>

## 4.5 RESERVA DE VENTANA DE MANTENIMIENTO (Responsable: Líder de Proyecto)

Comunicación:

- Notificar usuarios finales
  - Fecha: 01/07/2025.
  - Horario: 20:00 - 23:59
  - Coordinar con áreas afectadas.

## Fase 2: DE CONFIGURACIÓN (1 día)

Componentes	Archivos clave	Parámetros Críticos
Backend Laravel	archivo.env	APP_URL=http://127.0.0.1. DB_CONNECTION = mysql DB_HOST=127.0.0.1 DB_PORT= 3306 DB_DATABASE=beji DB_USERNAME=root DB_PASSWORD= JWT_SECRET= Clave generada
App Móvil	application-prod.properties	server.port=8080 spring.datasource.url=jdbc:mysql://servidor.mysql:3306/beji spring.datasource.username=root spring.datasource.password=

## FASE 4: VERIFICACIÓN (1 DÍA)

Prueba	Criterio Éxito	Herramienta
Conexión Laravel – DB	Responde consulta SELECT	MysqlWotkbench + Postman
Consumo API Angular	Al consultar productos, usuarios, proveedores, ventas se obtiene respuesta JSON	Postman
API Spring-boot MySQL	El backend móvil devuelve datos de productos ventas usuarios y proveedores al emulador Android	Postman
Autenticación JWT	Usuario puede iniciar sesión correctamente desde web	Postman + Frontend Angular

## FASE 5: CONTINGENCIAS

Riesgo	Mitigación
Error de conexión a MySQL	Verificar credenciales en .env y application.properties.
Error en Laravel (API web)	Revisar logs (storage/log/laravel.log)
Error en Spring-boot	Revisar logs.
Vulnerabilidad de seguridad detectada	Revocar Tokens JWT, actualizar dependencias.

