

# **PROYECTO CALIDAD DE SOFTWARE GESTIÓN DE MESAS CAFÉ/BAR**

Branndon Schneyder Suarez Jiménez

ADSO, Centro de Servicios Financieros, Sena

Ficha: 3197813

Andrés Ricardo Vallejo Cifuentes

05 de diciembre de 2025

## **PROYECTO CALIDAD DE SOFTWARE**

Aprendiz: Branndon Schneyder Suarez Jimenez.

Ficha: 3197813

Nombre del proyecto: Control de mesas.

Lenguaje Programación: Java Spring Boot.

Software a realizar: Control de mesas para café/bar.

## **1. Descripción del Desarrollo**

Este proyecto se desarrolló utilizando el framework Spring Boot, una tecnología que permite crear aplicaciones Java empresariales robustas y de rápida configuración.

Tecnologías Principales: Se utilizó Java con Spring Boot para el backend.

Gestión de Dependencias: La estructura del proyecto y la gestión de librerías se manejó a través de la configuración de los archivos pom.xml (Maven), asegurando la correcta integración de módulos como Spring Data JPA, Spring Security, y el driver de MySQL.

Persistencia y Base de Datos: Los datos se gestionan a través de Spring Data JPA (Hibernate), conectado a una base de datos MySQL externa. Las configuraciones de conexión, parámetros y otras configuraciones de entorno se establecieron en el archivo application.properties.

Funcionalidad Central (Autenticación): El sistema incluye un módulo de Login basado en Spring Security que valida credenciales de usuario y gestiona los roles (ADMIN y EMPLEADO) para un acceso seguro y diferenciado a los recursos de la aplicación.

Estructura del Proyecto: El desarrollo siguió el patrón MVC (Modelo-Vista-Controlador) para una clara separación de responsabilidades.

## **2. Resultados del análisis de calidad.**

Se empleó la herramienta SonarQube para realizar un análisis estático y metódico del código, asegurando el cumplimiento de las buenas prácticas de programación Java Spring boot y las convenciones de nomenclatura.

El análisis inicial identificó varias code smells (olores de código) relacionados con la convención de nombres y la duplicación de código. Todos los hallazgos se corrigieron para mejorar la mantenibilidad y la legibilidad del proyecto.

### 3. Tabla Comparativa de Errores Corregidos

Se solucionan solamente problemas de nivel alto, siendo estos:

Archivo Principales	Elemento Problemático	Problema Reportado por SonarQube	Solución Implementada
DetallePago.java	enum MetodoPago	Nombres de constantes no cumplen con UPPER_SNAKE_CASE.	Se renombraron los miembros del enum (Efectivo, Neki) a EFECTIVO, NEKI, etc.
Usuario.java	public Usuario() {}	Constructor vacío sin comentario o implementación.	Se añadió un comentario (// Constructor requerido por JPA...) para justificar su vacío, cumpliendo con la regla S1186.
AdminInitializer.java	Cadena "ADMIN"	Duplicación de la cadena literal "ADMIN" más de 3 veces.	Se extrajo el valor a una constante estática final: private static final String ROL_ADMIN = "ADMIN";

#### **4. Conclusiones Personales**

El uso de SonarQube resultó ser una parte crucial del proceso de desarrollo, ya que nos obligó a aplicar rigurosamente las convenciones de Java y a mejorar la calidad del código más allá de la simple funcionalidad. Trabajar con la eliminación de cadenas literales duplicadas (S1192) y la normalización de constantes (S115) hizo que el código fuera más legible y fácil de mantener. Esto tiene un propósito importante, preparando para futuros proyectos

Esta experiencia refuerza la importancia de integrar herramientas de calidad de código desde etapas tempranas, no solo para la detección de bugs, sino para la construcción de un software sostenible a largo plazo.