



## **Informe Técnico – Plan de Trabajo para Construcción de Software**

**PROYECTO:** SISTEMA DE INVENTARIO

**PROGRAMA DE FORMACIÓN:** ANÁLISIS Y DESARROLLO DE SOFTWARE

**APRENDICEZ:**

- *KAREN NIETO*
- *CAMILO CUADRADO*

## 2. Introducción

El desarrollo del Sistema de Inventario se fundamenta en el paradigma orientado a objetos y en las prácticas de integración continua, con el fin de garantizar un proceso de construcción de software eficiente, organizado y de alta calidad.

Este informe presenta el plan de trabajo técnico relacionado con la selección y justificación de herramientas de versionamiento, automatización e integración continua que soportarán la evolución del proyecto durante sus etapas de diseño, desarrollo, pruebas y despliegue.

## 3. Objetivo

Establecer un plan técnico de trabajo que incluya la selección, justificación y aplicación de herramientas de control de versiones e integración continua para el desarrollo del Sistema de Inventario, garantizando trazabilidad, organización y calidad en el proceso de construcción del software.

### 3.1 Objetivos Específicos

1. **Identificar y describir las herramientas de control de versiones adecuadas** para la construcción del Sistema de Inventario, evaluando sus beneficios y aplicaciones dentro del proceso de desarrollo.
2. **Seleccionar la plataforma de repositorio remoto** que permita almacenar, organizar y gestionar el código fuente garantizando trazabilidad y trabajo colaborativo.
3. **Definir el flujo de trabajo de versionamiento** (Git Flow o Feature Branching) que se implementará durante el desarrollo del proyecto, con el fin de mantener una estructura ordenada y controlada en las actualizaciones del software.
4. **Establecer las herramientas de integración continua** que permitirán automatizar procesos como compilación, pruebas y validación del código.
5. **Diseñar un plan de trabajo técnico** que integre versionamiento, integración continua y buenas prácticas del paradigma orientado a objetos para guiar el proceso de construcción del software.

#### 4. Características del Software a Desarrollar (Resumen técnico)

El Sistema de Inventario tendrá como propósito principal administrar productos, existencias, movimientos de entrada/salida y reporte de estados. Se basa en:

- Paradigma **orientado a objetos**, con clases como *Producto*, *Categoría*, *Usuario*, *MovimientoInventario*, etc.
- Arquitectura multicapa (presentación, lógica, datos).
- Base de datos relacional.
- Módulos:
  - Gestión de productos
  - Gestión de usuarios
  - Reportes
  - Movimientos de inventario

Esta estructura hace indispensable el uso de herramientas que permitan mantener organización, trazabilidad y control del código fuente.

#### 5. Selección de Herramientas de Versionamiento e Integración Continua

##### 5.1 Control de versiones

- **Herramienta seleccionada: Git**
  - Es el estándar industrial para control de versiones.
  - Permite trabajo colaborativo mediante ramas, commits y fusiones.
  - Soporta repositorios locales y remotos.
  - Facilita la trazabilidad de cambios.
- **Repositorio remoto seleccionado: GitHub**
  - Permite almacenar versiones del proyecto en la nube.
  - Facilita integración con herramientas DevOps.
  - Ofrece funciones como issues, proyectos y documentación.

- Permite aplicar flujos de trabajo como *Git Flow* o *Feature Branching*.

## 5.2 Estrategia de Versionamiento

Se utilizará el flujo: **Feature Branching**

- Rama principal: main
- Rama de desarrollo: develop
- Ramas por funcionalidad: feature/nombre-funcion
- Ramas de corrección: hotfix/nombre-correccion

Este flujo evita conflictos y garantiza estabilidad de la rama principal.

## 5.3 Integración Continua (CI)

- Se utilizará:

### GitHub Actions

- Automatiza pruebas, compilación y análisis del proyecto.
- Permite ejecutar flujos cada vez que se hace un push o pull request.
- No requiere servidores propios.
- **Pipeline propuesto:**
  1. Compilar proyecto
  2. Ejecutar pruebas automatizadas
  3. Validar estilo y calidad de código
  4. Generar reporte
  5. Notificar resultados en GitHub

## 5.4 Otras herramientas complementarias

Tipo de herramienta	Selección	Función
IDE	Visual Studio Code	Edición del código, extensiones para Git
Gestión de dependencias	npm / Maven (según lenguaje)	Administrar librerías
Seguimiento de tareas	GitHub Projects / Issues	Planificar backlog y tareas
Documentación	GitHub Wiki	Documentos técnicos y manuales

## 6. Plan de trabajo técnico

### Fases

#### 1. Configuración inicial del repositorio

- Crear repositorio en GitHub
- Configurar ramas principales
- Agregar README técnico

#### 2. Diseño del modelo orientado a objetos

- Definir clases
- Relaciones
- Diagramas UML

#### 3. Desarrollo iterativo

- Programación por sprints
- Creación de ramas por funcionalidad
- Pull requests con revisión

#### 4. Integración continua

- Implementación del pipeline con GitHub Actions
- Automatización de pruebas

#### 5. Versionamiento y despliegue

- Versiones usando etiquetas (tags)
- Release por hitos del proyecto

## Selección del Lenguaje de Programación y Base de Datos

### 6.1 Lenguaje de Programación Seleccionado

#### ✓ Java

El desarrollo del Sistema de Inventario se realizará completamente en **Java**, debido a que:

- Permite aplicar de manera directa el paradigma orientado a objetos.

- Es un lenguaje estable, seguro y ampliamente utilizado en aplicaciones empresariales.
  - Ofrece múltiples librerías y frameworks que facilitan la conexión con bases de datos.
  - Permite construir tanto la lógica del negocio como la interfaz del sistema (por ejemplo, usando JavaFX).
  - Es compatible con herramientas de integración continua como GitHub Actions.
- 

## 6.2 Base de Datos Seleccionada

### ✓ MySQL

La base de datos seleccionada para el proyecto es **MySQL**, porque:

- Es relacional y adecuada para manejar el inventario, productos, usuarios y movimientos del sistema.
  - Es gratuita, estable y compatible con Java.
  - Permite el uso de herramientas como **MySQL Workbench** para modelar y administrar los datos.
- 

## 6.3 Justificación de la Selección Tecnológica

- Java permite mantener una estructura completamente orientada a objetos, alineada con los requerimientos del proyecto.
- La integración entre Java y MySQL es simple mediante JDBC o librerías ORM.
- Ambas tecnologías son robustas, confiables y amplias en documentación.
- Facilitan el versionamiento e integración continua usando Git y GitHub.

## **7. Conclusiones**

La aplicación del paradigma orientado a objetos, junto con las prácticas de integración continua y el uso de Git y GitHub como herramientas de control de versiones, garantiza un proceso de desarrollo organizado, colaborativo y con altos estándares de calidad.

El plan técnico presentado establece una ruta clara para la construcción, seguimiento y mantenimiento del Sistema de Inventario, asegurando trazabilidad, automatización y control durante todas las fases del proyecto.