



Servicio Nacional De Aprendizaje - SENA

Tecnólogo en Análisis y Desarrollo de Software - 2834891

Informe técnico de plan de trabajo para construcción de software. GA7-220501096-AA1-  
EV01

Por

Christofer Jordan Homen

Presentado a

Israel Arbona

Bogotá D.C., 12 - Octubre – 2024

## **Introducción**

El presente informe técnico tiene como objetivo detallar las herramientas y tecnologías de versionamiento que se utilizarán en el desarrollo de la plataforma "Performance Radio". Esta plataforma se enfocará en ofrecer servicios publicitarios personalizados mediante cuñas radiales y espacios publicitarios en la web, facilitando la interacción entre marcas y audiencias a través de la gestión efectiva de anuncios.

---

## **Objetivo**

El objetivo de este informe es especificar las herramientas de versionamiento a utilizar en el desarrollo del software, asegurando un flujo de trabajo eficiente y la integración continua de los componentes del sistema.

---

## Selección de Herramientas de Versionamiento

Para el desarrollo del software de "Performance Radio", se seleccionarán las siguientes herramientas y tecnologías:

### 1. Git:

- **Descripción:** Git es un sistema de control de versiones distribuido que permite a múltiples desarrolladores colaborar en el código de manera eficiente. Es la herramienta más utilizada para gestionar proyectos de software, dado su robusto manejo de ramas y su capacidad para fusionar cambios.
- **Uso:** Se utilizará Git para realizar seguimiento de las modificaciones en el código fuente y para facilitar la colaboración entre los miembros del equipo de desarrollo.

### 2. GitHub/GitLab:

- **Descripción:** Estas plataformas ofrecen almacenamiento remoto para repositorios de Git, así como herramientas de colaboración y revisión de código. Además, permiten la integración de pipelines de CI/CD (Integración Continua/Entrega Continua).
- **Uso:** Se optará por GitHub o GitLab para gestionar el código y las colaboraciones, facilitando el trabajo en equipo y la implementación de revisiones de código mediante "pull requests".

### 3. Jenkins:

- **Descripción:** Jenkins es un servidor de automatización open-source que permite la integración continua y entrega continua del software. Facilita la ejecución automática de pruebas y la implementación de builds.
- **Uso:** Se implementará Jenkins para automatizar el proceso de integración y pruebas del código, asegurando que cada cambio realizado se valide antes de ser integrado al proyecto final.

### 4. Docker:

- **Descripción:** Docker es una plataforma que permite crear, desplegar y gestionar aplicaciones dentro de contenedores. Facilita la creación de entornos de desarrollo consistentes y la implementación de aplicaciones en diversas plataformas.
- **Uso:** Se utilizará Docker para empaquetar la aplicación de "Performance Radio" y sus dependencias, asegurando que funcione de manera uniforme en diferentes entornos.

### 5. Bitbucket:

- **Descripción:** Bitbucket es otra opción de gestión de repositorios que ofrece integración con pipelines de CI/CD y soporte para proyectos privados.
  - **Uso:** Puede ser utilizado para gestionar el código del proyecto y automatizar el proceso de integración continua.
-

## **Conclusión**

La implementación de estas herramientas y tecnologías de versionamiento no solo facilitará el desarrollo de "Performance Radio", sino que también asegurará un flujo de trabajo eficiente y de alta calidad a través de la integración continua. Al utilizar sistemas de control de versiones adecuados y herramientas de automatización, el equipo de desarrollo podrá mejorar la colaboración y la calidad del software, entregando un producto final robusto y eficiente.