Informe Técnico de Plan de Trabajo para Construcción de Software

David Felipe Morales

María Fernanda Ibáñez Benavides

Dilan Alexander Robayo

Juan Sebastian Florez

Julio Roberto Galvis

Servicio Nacional de Aprendizaje

Análisis y desarrollo de Software

2024

Bogotá D.

**Indice**

[**Introduccion**](#_93o6pnj21pjg) **3**

[**Objetivo**](#_t5u50ammoyjx) **4**

[**Selección de herramientas de versionamiento**](#_i1jcgprcirb5) **5**

[**Características Clave de Git**](#_g5cjbkxm2lye) **6**

[1. Distribuido y Descentralizado](#_951h7hz2mlqo) 6

[2. Rastreo Preciso de Cambios](#_iw8ex1d73psw) 6

[3. Ramificación Eficiente](#_7646dczup71) 6

[4. Fusiones Simplificadas](#_68knpks81ei5) 6

[**Ventajas de Usar Git en el Desarrollo de Software**](#_2jfl9wwuyv13) **7**

[1. Historial Completo y Rastreable](#_u2jyjydsu8xo) 7

[2. Desarrollo Colaborativo sin Fricción](#_ayi1693ixkia) 7

[3. Reversión Sencilla y Segura](#_naggsrlxsy1w) 7

[4. Flujo de Trabajo Flexible](#_a13w5zb5ngyt) 7

[5. Integración con Plataformas de Colaboración](#_5r12hlkebkrp) 8

[**Desventajas de usar GitHub**](#_2nvl0icbmf4y) **8**

[1. Curva de Aprendizaje](#_w8n6k4qw49qf) 8

[2. Dependencia de la Línea de Comandos](#_u9qyergkckik) 8

[3. Limitaciones de Privacidad en la Versión Gratuita](#_bbdd4pq9qn1u) 8

[4. Costos para Equipos Grandes](#_aedvl3387vh) 9

[5. Restricciones de Almacenamiento](#_g4bmm271beo) 9

[6. Fallas o Caídas del Servicio](#_a6huw6t67v13) 9

[7. Exposición Pública de Errores](#_udx221ymbm3c) 9

[**Repositorio de TWF en GitHub**](#_ms67hwhtjctt) **10**

[**Conclusiones**](#_7axmirslp2jk) **12**

# Introduccion

En el presente informe técnico se describe el plan de trabajo para la construcción del software The World Food, un sistema diseñado para la gestión de recetas gastronómicas a nivel nacional que permitirá a las personas acceder a diferentes recetas con sus debidas instrucciones de preparación y respectivos ingredientes.Es por ello que el objetivo principal es el que el proyecto sea eficiente y efectiva a la hora de satisfacer la necesidad de los usuarios finales. En este contexto del proyecto, es importante la gestión eficaz de las versiones esto para que se garantice la integridad y consistencia del código fuente, es por eso que para la elaboración de este informe se han evaluado diversas opciones de versionamiento y se ha seleccionado github como la herramienta más adecuada para nuestro proyecto. En capítulos posteriores a esta introducción se encontrará la explicación de todo lo que implica el uso de github, sus ventajas, desventajas , características principales, etc.

# Objetivo

The World food plantea utilizar la integración continua como práctica fundamental para garantizar la disponibilidad del espacio compartido en la cual todo el grupo participante podrá gestionar el trabajo propio por medio del sistema de control de versiones.

Es por eso que este documento explicará el funcionamiento y las debidas características de lo que este sistema de gestión de versiones github pueda ofrecer.

# Selección de herramientas de versionamiento

Github es un sistema de control de versiones distribuido desarrollado por Linus Torvalds en 2005. Se diseñó originalmente para gestionar el desarrollo del kernel de Linux, pero su flexibilidad y eficiencia lo convirtieron en una herramienta ampliamente adoptada en la industria del software. En esencia, Git permite rastrear los cambios en el código fuente a lo largo del tiempo, lo que facilita la colaboración entre desarrolladores y la gestión del ciclo de vida del software.

# Características Clave de Git

## 1. Distribuido y Descentralizado

A diferencia de los sistemas de control de versiones centralizados, en los que se almacena una única copia del repositorio, Git es distribuido y como desarrolladores tenemos una copia completa del historial de cambios. Esto nos permite trabajar de forma independiente y fusionar nuestros cambios en el repositorio principal de manera eficiente.

## 2. **Rastreo Preciso de Cambios**

Git realiza un seguimiento preciso de los cambios realizados en cada archivo a lo largo del tiempo. Esto nos permite como desarrollador ver el historial completo de cambios, incluidas las diferencias entre versiones y la posibilidad de regresar a versiones anteriores en caso de problemas.

## 3. **Ramificación Eficiente**

Git hace que la creación y gestión de ramas branches sea sencillo. Como desarrollador podemos crear ramas para trabajar con nuevas características o arreglos sin afectar la rama principal. Esto fomenta el desarrollo paralelo y facilita la colaboración en equipos grandes.

## 4. **Fusiones Simplificadas**

La capacidad de fusionar cambios entre diferentes ramas es una característica clave de Git. Como desarrollador podemos combinar cambios de manera eficiente, lo que facilita la integración continua y evita conflictos de código.

# Ventajas de Usar Git en el Desarrollo de Software

## 1. Historial Completo y Rastreable

Git permite rastrear y documentar cada cambio realizado en el código a lo largo del tiempo. Esto facilita la identificación de quién hizo qué cambio y cuándo, lo que es esencial para resolver problemas y mantener la transparencia en el desarrollo.

## 2. Desarrollo Colaborativo sin Fricción

Con Git, varios desarrolladores pueden trabajar en paralelo en diferentes características o soluciones sin interferir en el trabajo de los demás. Las ramas y las fusiones simplifican la colaboración y permiten una integración continua más fluida.

## 3. Reversión Sencilla y Segura

En caso de errores o problemas, Git nos permite como desarrolladores retroceder a versiones anteriores de manera rápida y segura. Esto reduce el riesgo de impactos negativos en el proyecto y facilita la corrección de errores.

## 4. Flujo de Trabajo Flexible

Git se adapta a diferentes flujos de trabajo, desde el modelo de ramificación estándar hasta enfoques más avanzados como Gitflow o GitHub Flow. Esto permite a los equipos adaptar Git a sus necesidades específicas.

## 5. Integración con Plataformas de Colaboración

Plataformas como GitHub ofrecen características de seguimiento de problemas, revisión de código y colaboración en proyectos. Git se integra perfectamente con estas herramientas, facilitando la gestión de proyectos y el trabajo en equipo.

# Desventajas de usar GitHub

Algunas de las desventajas que podemos presentar al momento de usar los repositorios de git serían: Aquí tienes otras desventajas de GitHub:

## 1. Curva de Aprendizaje

Para desarrolladores nuevos o sin experiencia en control de versiones, GitHub puede ser difícil de aprender, especialmente al manejar conceptos avanzados como ramas, fusiones o rebases.

## 2. Dependencia de la Línea de Comandos

Aunque existen interfaces gráficas, muchos comandos avanzados de Git requieren el uso de la línea de comandos, lo que puede resultar incómodo para algunos usuarios.

## 3. Limitaciones de Privacidad en la Versión Gratuita

Los repositorios privados en GitHub están limitados en la versión gratuita, lo que puede no ser ideal para proyectos que requieran confidencialidad sin coste.

## 4. **Costos para Equipos Grandes**

Aunque GitHub ofrece planes gratuitos, los planes premium para organizaciones grandes pueden ser costosos, especialmente si se requieren funciones avanzadas como análisis de seguridad y automatización.

## 5. Restricciones de Almacenamiento

GitHub impone límites en el tamaño de los repositorios y archivos, lo que puede ser problemático para proyectos que manejan archivos grandes o un número masivo de archivos.

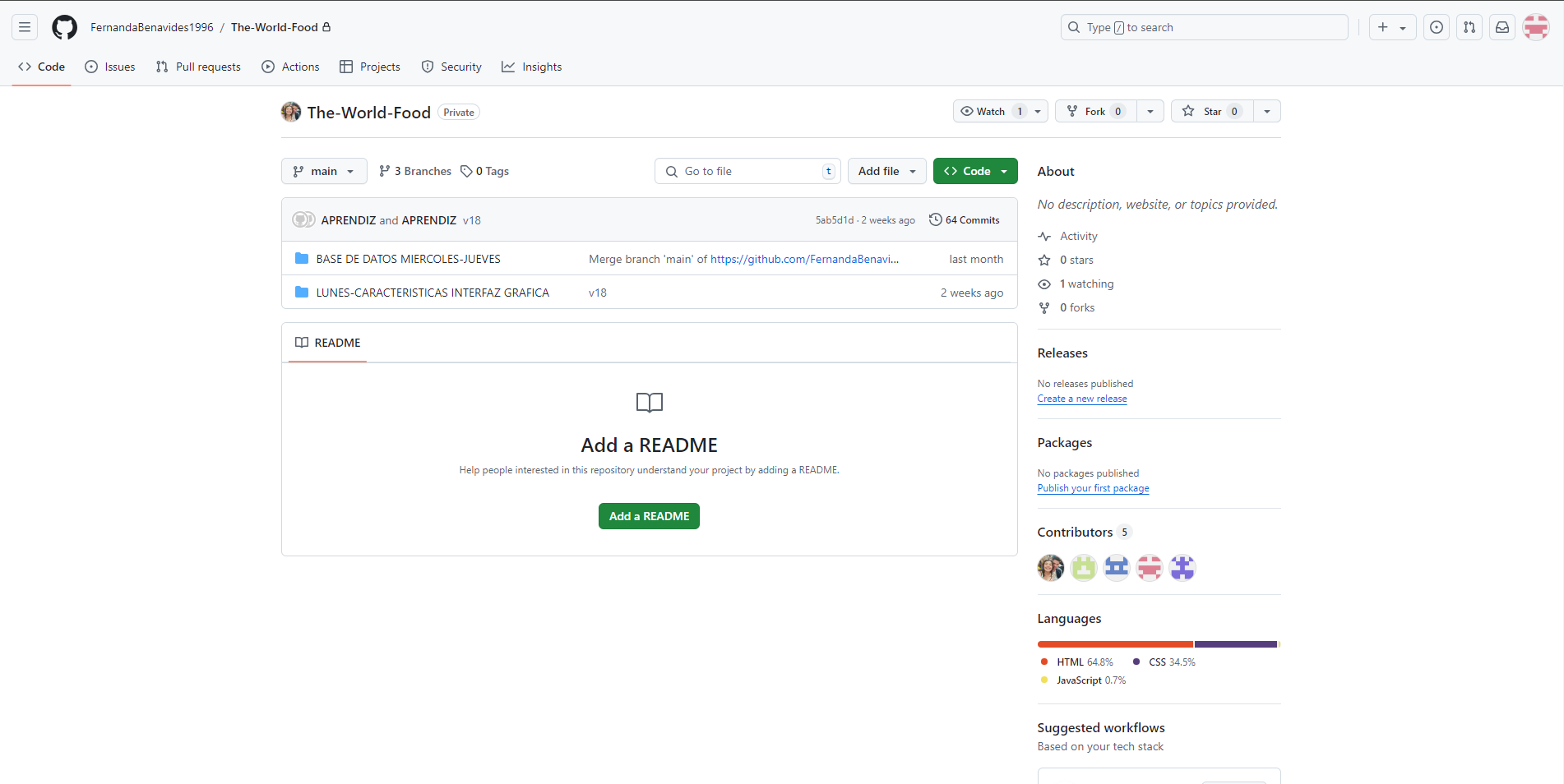
## 6. Fallas o Caídas del Servicio

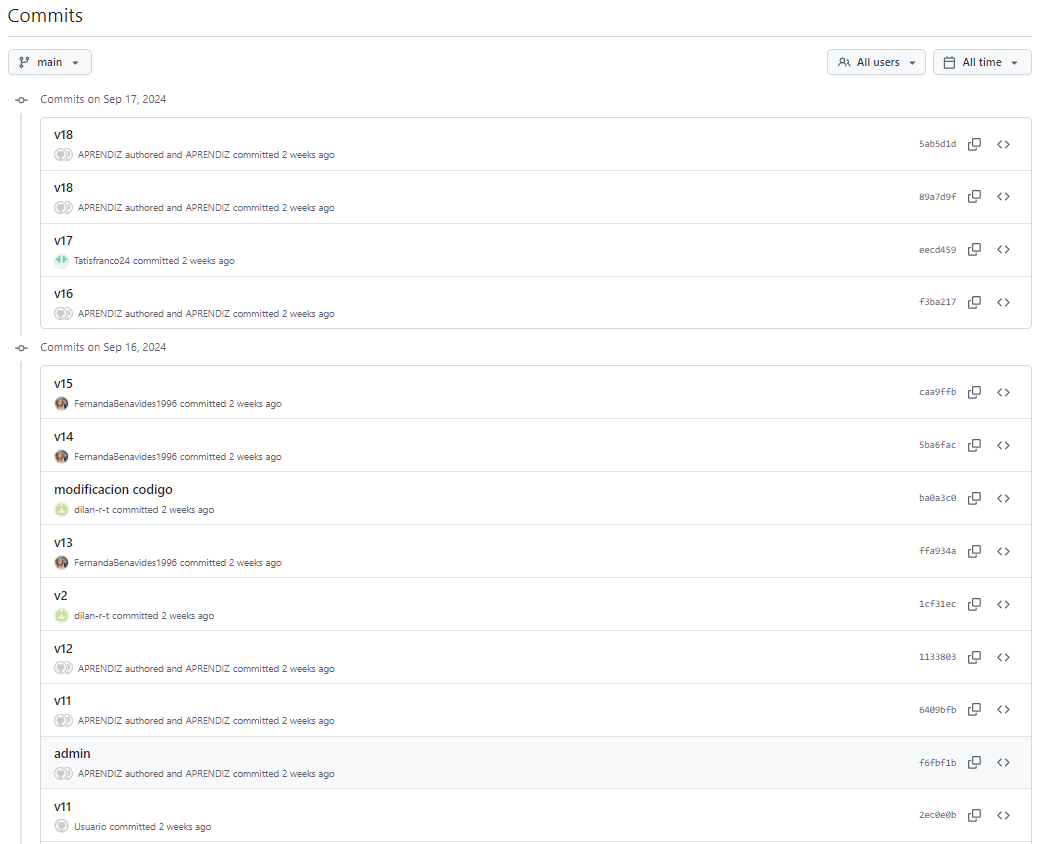
Aunque es raro, GitHub no está exento de interrupciones de servicio, lo que podría afectar la colaboración en tiempo real si la plataforma no está disponible temporalmente.

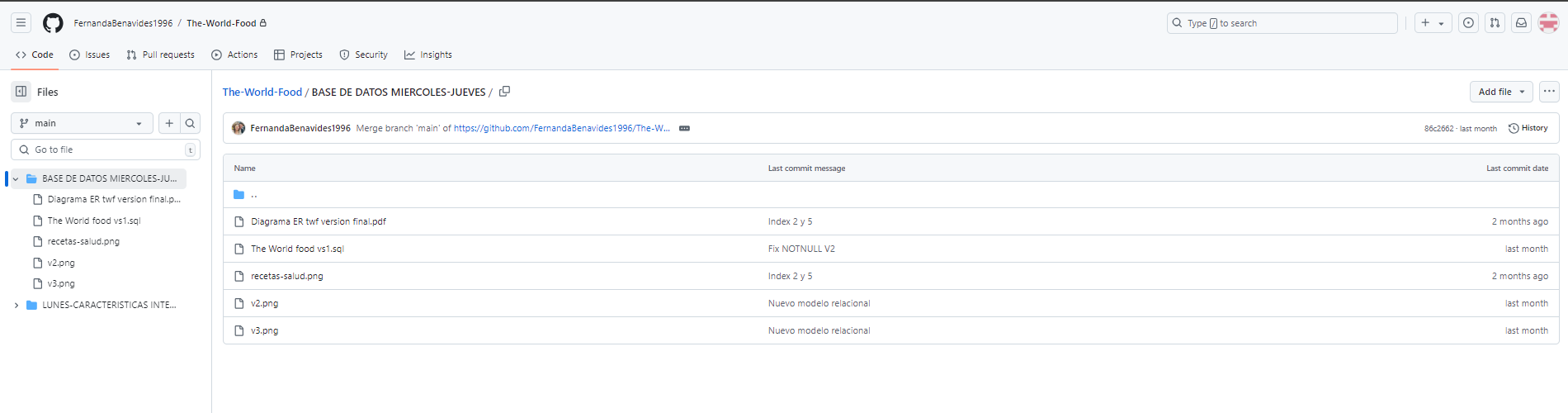
## 7. Exposición Pública de Errores

En repositorios públicos, cualquier error o malentendido en el código está visible para toda la comunidad, lo que podría ser una preocupación para algunos equipos o empresas que prefieren mantener su desarrollo más privado.

# Repositorio de TWF en GitHub







# Conclusiones

En conclusión, el informe técnico destaca la importancia de una adecuada gestión de versiones en el desarrollo del software The World Food. Se ha seleccionado GitHub como la herramienta más adecuada debido a su capacidad para rastrear cambios, facilitar la colaboración y permitir flujos de trabajo flexibles. Git ofrece ventajas significativas en términos de eficiencia, transparencia y seguridad, esenciales para el éxito del proyecto. Sin embargo, también se reconocen algunas desventajas, como la curva de aprendizaje, las limitaciones en la versión gratuita y las posibles interrupciones del servicio, que deben ser consideradas para una implementación efectiva.