

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	3
Objetivo	4
Arquitectura de hardware (Diagrama de despliegue)	5
Arquitectura de software (Diagrama de componentes)	6
Herramientas necesarias:	6
Elección de Sistema de Versionamiento	9
Git local y git remoto	9
Características Clave de Git	9
1. Distribuido y Descentralizado	9
2. Rastreo Preciso de Cambios	9
3. Ramificación Eficiente	10
4. Fusiones Simplificadas	10
Ventajas de Usar Git en el Desarrollo de Software	10
1. Historial Completo y Rastreable	10
2. Desarrollo Colaborativo sin Fricción	10
3. Reversión Sencilla y Segura	10
4. Flujo de Trabajo Flexible	11
5. Integración con Plataformas de Colaboración	11
Desventajas de usar GitHub	11
1. Curva de Aprendizaje	11
2. Dependencia de la Línea de Comandos	12
3. Limitaciones de Privacidad en la Versión Gratuita	12
4. Costos para Equipos Grandes	12
5. Restricciones de Almacenamiento	12
6. Fallas o Caídas del Servicio	12
7. Exposición Pública de Errores	12
Repositorio de TWF en GitHub	13
Conclusiones	15

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.

Arquitectura de hardware (Diagrama de despliegue)

Diagrama de despliegue The World Food

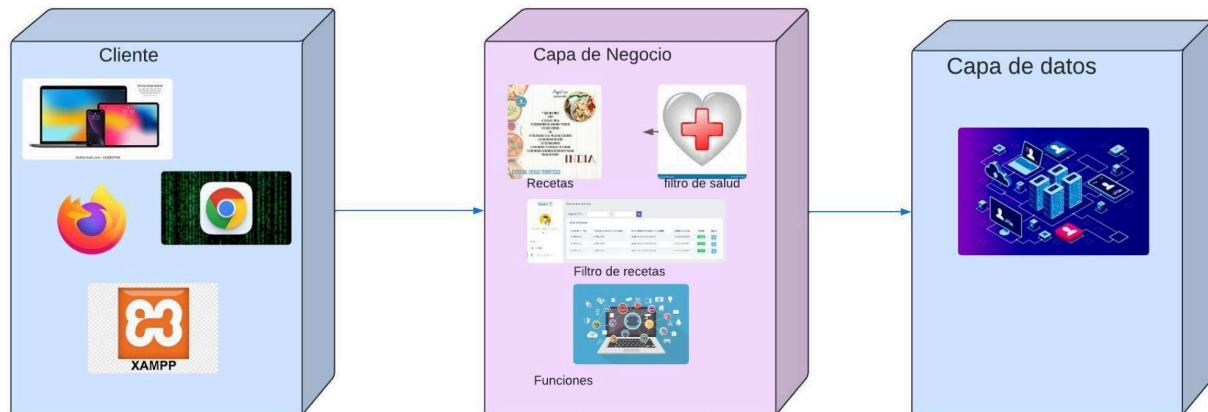
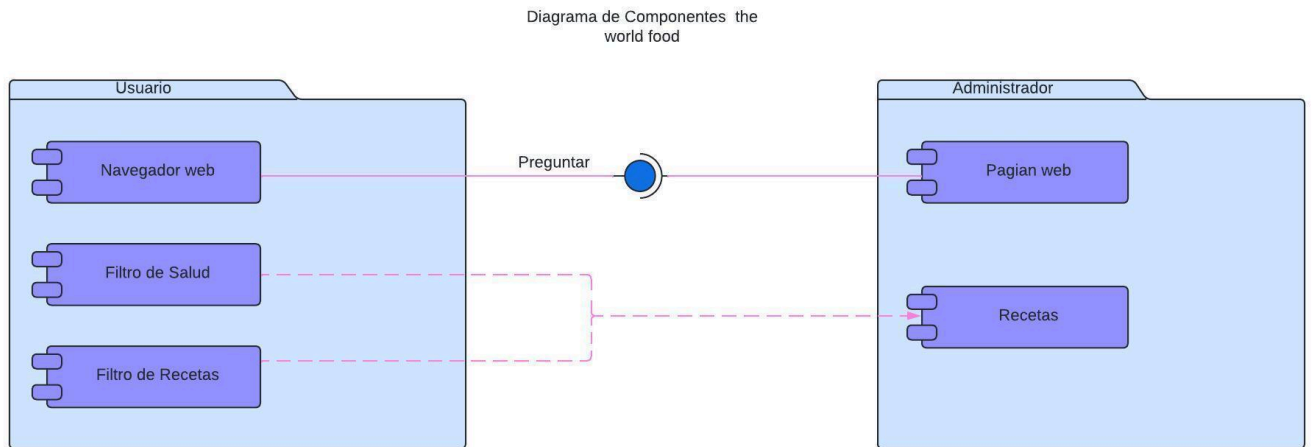


Diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta. En este caso The world food presenta 3 nodos en las cuales se nombran cliente, capa de negocio y capa de datos. En el nodo cliente se presenta qué dispositivos serán compatibles para acceder a la página, así mismo se especifica desde qué navegadores podrán acceder y que programa se utilizará para el Procesamiento de base de datos. La capa de negocio muestra la razón de ser del sistema, todo lo que la página proporcionará en modo funciones para que el usuario interactúe con lo que la página proporciona. Y por último la capa de datos es donde ira almacenada toda la base de datos que tendrá el sistema, es decir, la información de los usuarios, las recetas, las instrucciones, vídeos,etc.

Arquitectura de software (Diagrama de componentes)



El diagrama de componentes es una representación visual que muestra la estructura y las relaciones entre los diferentes componentes de un sistema. En este caso, el diagrama de componentes de "The World Food" ilustra las principales funcionalidades y módulos del sistema, destacando la interacción entre los usuarios y los administradores.

Herramientas necesarias:

- Entorno de desarrollo: El IDE que se usará para el desarrollo de The World Food (TWF) será Visual Studio Code dado a que es multiplataforma, es decir que tiene compatibilidad con los demás sistemas operativos, cuenta con facilidad de idioma, con un soporte remoto etc.
- **Lenguajes:**
JAVA(posiblemente):

Ventaja: Java permite que el mismo código se ejecute en diferentes plataformas sin modificaciones, gracias a la Java Virtual Machine (JVM).

Desventaja: Java puede ser más lento que otros lenguajes como C o C ++, debido a la sobrecarga de la JVM y la recolección de basura.

HTML:

Ventaja: HTML es el lenguaje estándar para crear páginas web y estructura el contenido en la web, proporcionando la base sobre la cual se construyen y se visualizan todos los elementos

de un sitio web.

Desventaja: Por sí solo, es estático y no puede manejar funciones interactivas o dinámicas.

MySQL:

Ventaja: MySQL es una base de datos relacional muy popular que ofrece una gestión de datos rápida y fiable, ideal para manejar grandes volúmenes de información y consultas complejas con eficiencia.

Desventaja: Escalar MySQL en grandes aplicaciones distribuidas puede ser complicado y requiere optimización avanzada.

CSS and BOOTSTRAP:

Ventaja: CSS permite la separación del contenido y el diseño visual de una página web, facilitando la personalización y el mantenimiento del estilo y la presentación de múltiples páginas a la vez.

Desventaja: Puede volverse complejo de gestionar en proyectos grandes sin una

estructura adecuada.

JavaScript:

Ventaja: JavaScript añade interactividad y dinamismo a las páginas web, permitiendo crear experiencias de usuario más atractivas y reactivas sin necesidad de recargar la página completa.

Desventaja: Puede ser difícil de depurar y mantener en proyectos grandes sin herramientas y buenas prácticas estructuradas.

- Manejador de base de datos:

El posible gestor de la base de datos a utilizar para la aplicación orientada a la web (The World Food) es MySQL. Se considera como la mejor opción ya que este gestor se destaca por su alto rendimiento y escalabilidad, es decir maneja un gran volumen de base de datos y su procesamiento es rápido. Además, ofrece una seguridad robusta lo que garantiza que la información que los usuarios ingresen está protegida por medio del cifrado.

Otra característica por la cual escogemos este gestor es por su compatibilidad con diversos lenguajes de programación (Java, PHP, C++, Python y Ruby). MySQL es un gestor que también permite que este tenga soporte con los diversos sistemas operativos como lo son Windows, Linux, Mac OS X, lo cual es una ventaja ya que, si los desarrolladores manejamos diferentes sistemas operativos, esto no afectará para seguir trabajando en la base de datos.

Otras ventajas que nos ofrece MySQL es que es totalmente gratis, lo que permite un amplio ahorro en términos de licencias.

Elección de Sistema de Versionamiento

Git local y git remoto

<https://github.com/FernandaBenavides1996/The-World-Food>

The screenshot shows the GitHub repository page for 'The-World-Food'. The repository is private and has 3 branches and 64 commits. The main branch is selected. The repository description is 'No description, website, or topics provided.' The repository has 0 stars, 1 watching, and 0 forks. The repository has no releases published and no packages published. The repository has 5 contributors and the languages are HTML (64.8%), CSS (34.5%), and JavaScript (0.7%). The repository has suggested workflows based on the tech stack.

The screenshot shows the 'Branches' page for the 'The-World-Food' repository. The page has tabs for Overview, Yours, Active, Stale, and All. The 'Overview' tab is selected. The page shows a search bar for branches. The 'Default' section shows the 'main' branch, which is the default branch. The 'Active branches' section shows two active branches: 'D.R.' and 'master'. The 'D.R.' branch is 38 commits ahead of 'main' and has 0 pull requests. The 'master' branch is 64 commits ahead of 'main' and has 1 pull request.

Branch	Updated	Check status	Behind / Ahead	Pull request
main	last month		Default	

Branch	Updated	Check status	Behind / Ahead	Pull request
D.R.	last month		38 / 0	
master	2 months ago		64 / 1	

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 Rastreado

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.

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, además este documento presenta información plena acerca de la dinámica de nuestro proyecto y las herramientas que se requieren.