

APLICACIÓN DE GIPHUB PARA CREAR UN REPOSITORIO

Anthony Marcatoma (*Autor*)
Facultad de Ingeniería Mecánica
Escuela Politécnica
Quito, Ecuador
antony.marcatoma@epn.edu.ec
GR1

Resumen—Se elaboro un foro en Github con el fin de subir el código de proyecto final de programació, se creó un enlace y compartió el url git además se elaboró un informe de como uso Git. Para la personalización se usó Markdown para poner estilos de lectura en el código.

Palabras claves—

I. INTRODUCCION

Git es un software de control de versiones distribuido, libre y de código abierto, diseñado para la eficiencia, la confiabilidad y la compatibilidad del mantenimiento de versiones de aplicaciones, especialmente aquellas con un gran número de archivos de código fuente.

Git se caracteriza por realizar control de versiones, esto quiere decir que puede realizar las siguientes acciones:

- **Rastrear los cambios:** Puedes ver quién hizo qué cambio, cuándo se realizó y por qué.
- **Revertir cambios:** Si cometes un error, puedes volver a una versión anterior del proyecto.
- **Colaborar con otros:** Puedes compartir tu código con otros y trabajar juntos en el mismo proyecto.

Git tiene muchas ventajas sobre otros sistemas de control de versiones, como:

Eficiencia: Git es muy rápido y eficiente, incluso para proyectos grandes.

- **Confianza:** Git es muy confiable y rara vez pierde datos.
- **Compatibilidad:** Git es compatible con muchos sistemas operativos y lenguajes de programación.
- **Gratuito y de código abierto:** Git es gratuito y de código abierto, lo que significa que cualquiera puede usarlo y modificarlo.

II. USO DE GITHUB

1. Creacion de un usuario

Primero entramos a la página de Github es necesario crear un usuario, despues tendremos acceso a las herramientas de la aplicación.



Fig 1. Cuenta de Github.

2. Elaboracion del repositorio

Dentro del programa se da click en nuevo repositorio, agregamos un título y una descripción sobre el repositorio.

Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * AntonyMarcatoma / Repository name *

Great repository names are short and memorable. Need inspiration? How about [bug-free-meme](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

fig 2. Repositorio creado.

3. Publicación del código

Al crear el repositorio se debe seleccionar la opción importar código, luego aparece un editor de texto donde se colocará el código del proyecto.

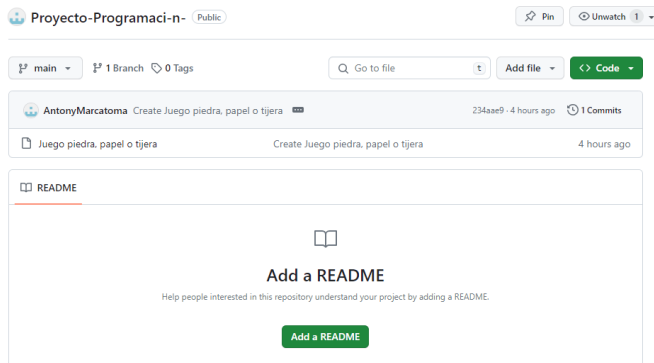


Fig. 3. Repositorio que incluye código del videojuego.

III. CONCLUSIONES

- Al crear un repositorio en GitHub, se obtienen una serie de beneficios y ventajas que pueden mejorar significativamente el desarrollo y la gestión de proyectos de software.
- Permite realizar un seguimiento de los cambios realizados en el código a lo largo del tiempo, lo que facilita revertir errores o recuperar versiones anteriores.

- ofrece una serie de herramientas y ventajas que pueden mejorar significativamente el proceso de desarrollo.

IV. RECOMEDACIONES

- Usa una nomenclatura clara y descriptiva para tus archivos y carpetas. Esto facilitará la navegación y comprensión del código por parte de otros colaboradores.
- Escribe comentarios claros y concisos en tu código. Esto ayudará a otros desarrolladores a entender cómo funciona tu código y por qué se ha escrito de esa manera.
- Utiliza las etiquetas y las releases para organizar tu código. Esto te ayudará a identificar fácilmente versiones específicas de tu código y a distribuirlas a tus usuarios.

REFERENCES

- [1] <https://git-scm.com/>
- [2] <https://github.com/>
- [3] <https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository>