

# Integración de Perfil de Estudiante en GitHub

Marco Antonio Gómez Angel

September 3, 2025

## 1 Datos generales

1. Nombre del perfil: Marco-Angel
2. Programa y semestre: Ingeniería Electronica; 5-6 semestre.
3. Usuario de GitHub: [usuario](#)
4. Repositorio de perfil: [README.md](#)

## 2 Objetivo del documento

Describir el proceso para crear un perfil original en GitHub y registrarlo en el repositorio central de la clase, aportando evidencias y enlaces.

## 3 Paso a Paso

### 3.1 Creación/Actualización del repositorio de perfil

1. Buscar GitHub
2. Registrarse con el correo de la Universidad.
3. Crear una contraseña personal
4. Crear el repositorio <usuario>/<usuario> en GitHub.
5. Elaborar un `README.md` original con bio, habilidades y proyectos.
6. Subir activos (imágenes) y configurar topics/visibilidad.

## 4 Evidencias

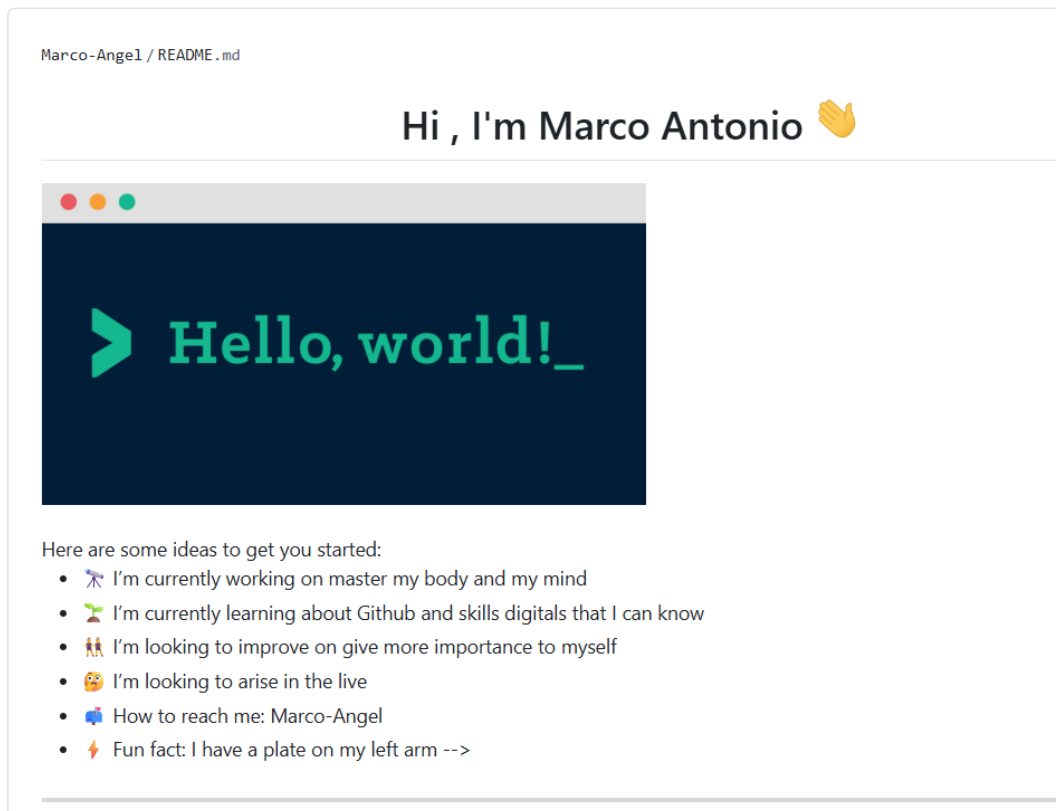


Figure 1: primera evidencia

Marco-Angel / README.md

in main

Cancel ch

Edit

Preview

Spaces

```

6 Here are some ideas to get you started:
7
8 - 🦾 I'm currently working on master my body and my mind
9 - 🦾 I'm currently learning about Github and skills digitals that I can know
10 - 🦾 I'm looking to improve on give more importance to myself
11 - 🦾 I'm looking to arise in the live
12 - 🦾 How to reach me: Marco-Angel
13 - ⚡ Fun fact: I have a plate on my left arm
14 -->
15 <hr>
16
17 <hr>
18
19 <p align="center">
20 <i><b>Profile Visitor Count...</b></i><br><br>
21 <br><br>
22 
23 </p>
24
25 <hr>
26
27
28
29 <!--
30 ****Marco-Angel is a 🦾 special 🦾 repository because its "README.md" (this file) appears on your Github profile
31
32 Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use |esc| then tab to move to the next interactive element on the page.

```

Figure 2: segunda evidencia



Figure 3: evidencia tres

## 5 GIT

Git es un sistema de control de versiones distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad.[descripción](#)

### 5.1 Aspectos Basicos

Git crea una confirmación. Una confirmación es una instantánea de todos los archivos en un momento dado. Si un archivo no ha cambiado de una confirmación a la siguiente, Git usa el archivo almacenado anteriormente. Este diseño difiere de otros sistemas que almacenan una versión inicial de un archivo y mantienen un registro de las diferencias a lo largo del tiempo.[descripción](#)

### 5.2 Ventajas de Git

- Desarrollo simultáneo
- Versiones de lanzamiento más rápidas
- Integración incorporada
- Sólido soporte técnico de la comunidad

- Git funciona con cualquier equipo
- Solicitudes de incorporación de cambios
- Directivas de rama

## 6 GITHUB

### 6.1 ¿Que es GITHUB?

GitHub es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso. [acerda de github](#)

### 6.2 Recursos de GitHub

1. "GitHub Hello World quickstart exercise" (Ejercicio de inicio rápido GitHub Hola mundo): tutorial en línea que le permite practicar los conceptos básicos de Git mediante GitHub.
2. "GitHub Guides" (Guías de GitHub): la documentación principal de GitHub
3. "GitHub learning resources" (Recursos de aprendizaje de GitHub): otros recursos útiles de GitHub
4. "Glossary" (Glosario): útil glosario de términos de Git y GitHub.
5. "GitHub Student developer pack" (Paquete de desarrollador de GitHub Student): acceso gratuito para alumnos a las mejores herramientas de desarrollo

[acerca de los recursos](#)

## 7 GITLAB

### 7.1 ¿Que es ¿GitLab?

GitLab es una plataforma integral diseñada para el desarrollo de software y DevOps . Inicialmente, era un gestor de repositorios Git basado en la web que permitía a los equipos colaborar en la programación. Sin embargo, desde entonces ha evolucionado hasta convertirse en una solución totalmente integrada para gestionar todo el ciclo de vida del desarrollo de software (SDLC). [acerca de gitlab](#)

### 7.2 Utilidades

1. Manejar el control de versiones
2. Automatizar las canalizaciones de CI/CD
3. Gestionar tareas del proyecto
4. Monitorizar el rendimiento

### 7.3 Características

1. Los repositorios de GitLab sirven como base para la gestión del código fuente basado en Git.
2. CI/CD automatizan los procesos de prueba, compilación e implementación de código mediante archivos de configuración YAML sencillos que definen pipelines para automatizar estas tareas.
3. Auto DevOps simplifica el proceso de CI/CD al ofrecer plantillas preconfiguradas para crear, probar e implementar aplicaciones.
4. GitLab también destaca en la gestión de proyectos, ofreciendo herramientas como tableros de incidencias, hitos, etiquetas y diagramas de evolución para facilitar los flujos de trabajo ágiles.

[acerca de gitlab](#)

## **7.4 Ventajas de GitLab**

1. Plataforma todo en uno
2. Colaboración y gestión de equipos
3. Escalabilidad y flexibilidad
4. Seguridad y cumplimiento