

# Sobre Git y GitHub

natetas



# Índice general

<b>1. Definiciones</b>	<b>5</b>
1.1. Git . . . . .	6
1.2. GitHub . . . . .	6
1.3. Repositorio (repo) . . . . .	6
1.4. Commit . . . . .	6
1.5. Ramas (branches) . . . . .	6
1.5.1. Rama principal (main o master) . . . . .	6
1.5.2. Ramas secundarias . . . . .	6
1.6. Clonar (clone) . . . . .	7
1.7. Pull . . . . .	7
1.8. Push . . . . .	7
1.9. Clave SSH . . . . .	7
1.10. Archivo .md . . . . .	7
1.11. Sintaxis de Markdown . . . . .	7
<b>2. Comandos</b>	<b>9</b>
2.1. Navegación y manejo de archivos. . . . .	9
2.2. Manejo de repositorios de GitHub. . . . .	10
<b>3. Para acceder a los repositorios.</b>	<b>11</b>
3.1. Antes de acceder. . . . .	11
3.2. Para clonar el repositorio. . . . .	13



## Capítulo 1

# Definiciones

## 1.1. Git

Es un sistema de control de versiones que te permite rastrear cambios en archivos a lo largo del tiempo. Te permite trabajar en diferentes versiones de un proyecto sin perder el historial de cambios.

## 1.2. GitHub

Es una plataforma en la nube que aloja *repositorios Git*. Permite almacenar proyectos en línea, colaborar con otros y acceder a los archivos desde cualquier computadora.

## 1.3. Repositorio (repo)

Es una carpeta que contiene todos los archivos del proyecto, junto con el historial de cambios. Puede estar de manera local o en *GitHub* (remoto).

## 1.4. Commit

Es un "*punto de guardado*" en el historial del proyecto. Cada vez que haces un *commit*, guardas una versión de los archivos en ese momento.

## 1.5. Ramas (branches)

Es una línea de desarrollo independiente dentro de un *repositorio*.

Cada *rama* tiene su propio historial de *commits*, lo que permite trabajar en nuevas características, experimentos o correcciones de errores sin afectar la versión principal del proyecto.

### 1.5.1. Rama principal (main o master)

Es la *rama* predeterminada y suele contener la versión estable y funcional del proyecto.

Todos los cambios importantes (como nuevas versiones) se fusionan en esta *rama*.

### 1.5.2. Ramas secundarias

Son *ramas* que se crean a partir de la *rama principal* (o de otra *rama*) para trabajar en tareas específicas.

## 1.6. Clonar (clone)

Descargar un repositorio remoto (de *GitHub*) a una computadora local.

## 1.7. Pull

Traer cambios desde un *repositorio remoto* a un repositorio local.

## 1.8. Push

Enviar cambios desde un *repositorio local* a un repositorio remoto.

## 1.9. Clave SSH

Una *clave SSH* es un método de autenticación seguro que permite a una computadora comunicarse con servicios remotos (como GitHub) sin necesidad de usar contraseñas. Consiste en un par de archivos:

- **Clave privada:** Es un archivo que se guarda solo en tu computadora. Nunca debes compartirla, ya que es como una llave maestra que da acceso a tus repositorios.
- **Clave pública:** Es un archivo que puedes compartir con servicios como GitHub. GitHub usa esta clave para verificar que tienes la clave privada correspondiente.

## 1.10. Archivo .md

Un *archivo .md* es un archivo de texto plano que usa la *sintaxis Markdown*

## 1.11. Sintaxis de Markdown

*Markdown* es un lenguaje de marcado ligero que permite formatear texto de manera sencilla, como añadir títulos, listas, enlaces, imágenes, código, etc., sin necesidad de usar HTML o herramientas complejas.





# Capítulo 2

## Comandos

### 2.1. Navegación y manejo de archivos.

Comando	Descripción
<code>pwd</code>	Muestra la ruta del directorio actual.
<code>ls</code>	Lista los archivos y carpetas en el directorio actual.
<code>ls -l</code>	Lista los archivos y carpetas en formato largo (con detalles como permisos y tamaño).
<code>ls -a</code>	Lista todos los archivos, incluyendo los ocultos (que empiezan con <code>.</code> ).
<code>cd &lt;carpeta&gt;</code>	Cambia al directorio especificado.
<code>cd ..</code>	Sube un nivel en la estructura de directorios.
<code>cd ~</code>	Cambia al directorio home del usuario.
<code>mkdir &lt;nombre&gt;</code>	Crea una nueva carpeta con el nombre especificado.
<code>rmdir &lt;nombre&gt;</code>	Elimina una carpeta vacía.
<code>touch &lt;archivo&gt;</code>	Crea un archivo vacío o actualiza su fecha de modificación.
<code>cp &lt;origen&gt;&lt;destino&gt;</code>	Copia un archivo o carpeta.
<code>mv &lt;origen&gt;&lt;destino&gt;</code>	Mueve o renombra un archivo o carpeta.
<code>rm &lt;archivo&gt;</code>	Elimina un archivo.
<code>rm -r &lt;carpeta&gt;</code>	Elimina una carpeta y su contenido de forma recursiva.
<code>cat &lt;archivo&gt;</code>	Muestra el contenido de un archivo.
<code>less &lt;archivo&gt;</code>	Muestra el contenido de un archivo página por página.
<code>echo "texto"</code>	Muestra texto en la terminal o lo redirige a un archivo.
<code>echo "texto» &lt;archivo&gt;</code>	Guarda el texto en un archivo (sobrescribe).
<code>echo "texto»»&lt;archivo&gt;</code>	Añade el texto al final de un archivo.

## 2.2. Manejo de repositorios de GitHub.

Comando	Descripción
<code>git init</code>	Inicializa un repositorio Git.
<code>git clone &lt;URL&gt;</code>	Clona un repositorio remoto.
<code>git status</code>	Muestra el estado del repositorio.
<code>git add &lt;archivo&gt;</code>	Añade archivos al área de preparación.
<code>git commit -m "Mensaje"</code>	Guarda los cambios en el historial.
<code>git push origin &lt;rama&gt;</code>	Sube cambios al repositorio remoto.
<code>git pull origin &lt;rama&gt;</code>	Trae cambios desde el repositorio remoto.
<code>git branch &lt;nombre&gt;</code>	Crea una nueva rama.
<code>git checkout &lt;rama&gt;</code>	Cambia a una rama específica.
<code>git merge &lt;rama&gt;</code>	Fusiona una rama con la rama actual.
<code>git log</code>	Muestra el historial de commits.
<code>git diff</code>	Muestra diferencias entre cambios.
<code>git remote add origin &lt;URL&gt;</code>	Añade un repositorio remoto.
<code>git tag &lt;nombre&gt;</code>	Crea una etiqueta (tag).
<code>git reset &lt;archivo&gt;</code>	Elimina un archivo del área de preparación.
<code>git checkout - &lt;archivo&gt;</code>	Descarta cambios en un archivo.
<code>git remote -v</code>	Muestra los repositorios remotos configurados.
<code>git branch -d &lt;rama&gt;</code>	Elimina una rama local.

Cuadro 2.1: Comandos útiles de Git

## Capítulo 3

# Para acceder a los repositorios.

### 3.1. Antes de acceder.

Sigue estos pasos para generar una **clave SSH** en tu computadora y añadirla a GitHub:

#### 1. Genera la clave SSH

- Abre la terminal en tu laptop.
- Ejecuta el siguiente comando para generar una **clave SSH**:

```
ssh-keygen -t ed25519 -C "tu-email@ejemplo.com"
```

- Presiona **Enter** para aceptar la ubicación predeterminada de la clave.

#### 2. Añade la clave SSH a tu agente SSH

- Ejecuta los siguientes comandos para asegurarte de que el agente SSH esté activo y añadir tu clave:

```
eval "$(ssh-agent -s)"  
ssh-add ~/.ssh/id_ed25519
```

#### 3. Copia la clave pública

- Usa el siguiente comando para mostrar tu clave pública en la terminal:

```
cat ~/.ssh/id_ed25519.pub
```

#### 4. Añade la clave pública a GitHub

- Ve a la pestaña **SSH and GPG keys** en GitHub (dentro de Settings).
- Haz clic en **New SSH key**.

- Dale un título descriptivo (por ejemplo, "Mi Laptop").
- Pega la clave pública en el campo **Key**.
- Haz clic en **Add SSH key**.

#### 5. Verifica la conexión SSH

- En la terminal, ejecuta:

```
ssh -T git@github.com
```

- Si todo está configurado correctamente, verás un mensaje como:

```
Hi ElNatitis! You've successfully authenticated  
, but GitHub does not provide shell access.
```

## 3.2. Para clonar el repositorio.

Una vez que se configuró la **clave SSH** en tu computadora y esta se añadió a GitHub (Sección 3.1), puedes acceder a tus repositorios, usando SSH, siguiendo los siguientes pasos.

### 1. Obtén la URL SSH de tu repositorio

- Ve a la página de tu repositorio en GitHub.
- Haz clic en el botón verde `Code`.
- Selecciona la pestaña SSH y copia la URL. Se verá algo como esto:

```
git@github.com:ElNatitis/oach.git
```

### 2. Clona de repositorio

- Abre la terminal y ejecuta el siguiente comando:

```
git clone git@github.com:ElNatitis/oach.git
```

- Esto descargará el repositorio en una carpeta con el nombre del repositorio (oach en este caso).

### 3. Navega a la carpeta del repositorio

Usa el siguiente comando para entrar a la carpeta del repositorio:

```
cd oach
```