



Paradigmas de la programación

Hernández Ceseña Iván Fernando

Práctica 0.

Git y Markdown

29 de Mayo del 2024

En este reporte se ven los fundamentos sobre git y github, así como lo que es un lenguaje de marcado como Markdown y una demostración de los comandos básicos de git para el control de archivos y versiones.

```
### Github
Github es una plataforma de desarrollo colaborativo basado en Git. Git es un sistema de control de versiones que permite a los desarrolladores gestionar y realizar un seguimiento de cambios en el código fuente durante el desarrollo de software.

### Como crear cuenta en GitHub
1. Accede a Github en github.com
2. Haz clic en "Sign up".
3. Crea tu cuenta ingresando los datos requeridos(nombre de usuario, correo electronico y contraseña).
4. Verifica tu correo electronico mediante el enlace que te enviara github.
5. Personaliza tu perfil añadiendo una descripción y foto de perfil.
6. Crear repositorio en GitHub

### Como crear repositorio en GitHub
1. **Crear un repositorio nuevo**:
   - Haz clic en el símbolo "+" en la esquina superior derecha de cualquier página de GitHub.
   - Selecciona "New repository" (Nuevo repositorio) del menú desplegable.
   - Ingresa un nombre para tu repositorio. Por ejemplo, "mi-proyecto".
   - Opcionalmente, añade una descripción corta del proyecto.
   - Elige si quieres que el repositorio sea público o privado.
   - Los repositorios públicos son visibles para todos. Los privados solo pueden ser vistos y colaborados por ti y los colaboradores que invites.
   - Puedes añadir un archivo README, que es una buena práctica para describir tu proyecto.
2. **Opciones adicionales**:
   - Puedes añadir un archivo de gitignore (ignorar archivos específicos, como los generados automáticamente) y una licencia de código abierto.
3. **Crear el repositorio**:
4. Haz clic en "Create repository" (Crear repositorio).
```

```
## Comandos basicos en git
### Iniciar sesion en git
```bash
git config --global user.name "Nombre de usuario"

git config --global user.email "
```

### Clonar repositorio
```bash
git clone <url_repositorio>
```

### Agregar archivo
```bash
git add nombre_del_archivo
```

### Confirmar cambios
```bash
git commit -m "Descripcion del cambio"
```

### Subir cambios al repositorio
```bash
git push origin <nombre_de_la_rama>
```

### Actualizar repositorio local
```bash
git pull origin <nombre_de_la_rama>
```
```

```
## Comandos utiles para terminal git
```

```
```bash
```

```
git status
```

```
git log
```

```
git branch
```

```
git checkout
```

```
git merge
```

```
git reset
```

```
```
```

```
### Crear directorio
```

```
```bash
```

```
mkdir <nombre_del_directorio>
```

```
```
```

```
### Crear archivo
```

```
```bash
```

```
touch <nombre_del_archivo>
```

```
```
```

```
### Eliminar archivo
```

```
```bash
```

```
rm <nombre_del_archivo>
```

```
```
```

```
### Eliminar directorio
```

```
```bash
```

```
rm -r <nombre_del_directorio>
```

```
```
```

```
### Cambiar directorio
```

```
```bash
```

```
cd <nombre_del_directorio>
```

```
```
```

```
### Regresar un directorio arriba
```

```
```bash
```

```
cd ..
```

```
```
```

Regresar un directorio arriba

```
```bash
```

```
cd ..
```

```
```
```

Mostrar directorio actual

```
```bash
```

```
pwd
```

```
```
```

Listar archivos en el directorio actual

```
```bash
```

```
ls
```

```
```
```