

DIARIO DE DARKGAMBLER

NOTAS GITHUB

Github Notas

Darkgambler

Profesor
WALTER WHITE

March 19, 2022

1 Capítulo 1: Configuraciones

De la siguiente manera se configura el usuario y el email en nuestro entorno git:

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

Para poder ver los valores ya configurados en el entorno se escribe el siguiente comando:

```
git config --global -l
```

Para configurar el end of file se usa el siguiente comando

```
git config --global core.autocrlf true
```

Para agregar credenciales al cache, primero se necesita hacer un push con lo que se solicitará el username y el token como password, luego de que se hizo el push correctamente se debe escribir:

```
git config --global credential.helper cache
```

si estamos en entorno windows entonces se debe escribir lo siguiente:

```
git config --global credential.helper wincred
```

Con el siguiente código se pueden agregar alias, que sirven para sustituir texto de comandos git con el fin de no escribir tanto, por ejemplo si queremos acortar commit -m con el nombre com:

```
git config --global alias.com "commit -m"
```

Para borrar el alias que anteriormente creamos simplemente escribimos:

```
git config --global --unset alias.com
```

Si se desea ver las configuraciones hechas en el git, así como los alias y otras informaciones podemos abrir el archivo de configuraciones en emacs u otro

```
emacs ~/.gitconfig
```

O bien lo podemos ver en la terminal:

```
git config --global --list
```

2 Capítulo 2: Crear un repositorio

Las siguientes son las instrucciones para crear un repositorio:

```
echo "cualquier texto para el readme" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin <url_repositorio_github>
git push -u origin master
```

git init se puede inicializar en un directorio agregando simplemente el nombre del directorio que se quiere crear

```
git init <NOMBRE_DIRECTORIO>
```

Si se desea agregar todo lo que hay se puede usar el comando add de la siguiente manera

```
git add .
```

Si se hizo algún cambio en otra cuenta asociada al branch o en la página de github que queremos actualizar en local debemos escribir lo siguiente:

```
git pull
```

Si queremos revertir el pull hecho entonces primero debemos encontrar el ID commit usando:

```
git log
```

y luego:

```
git reset --hard <commit id>
```

Si queremos ver los cambios que se han hecho al repositorio sin actualizar los archivos locales utilizamos:

```
git fetch
```

3 Capítulo 3: Markdown

Aquí veremos el uso básico de markdown para darle formato a los archivos .md, típicamente los README.

3.1 Saltos de línea

Para realizar un salto de línea simplemente debemos dar dos espacios seguidos de enter.

3.2 Encabezados

Para realizar encabezados se utiliza la almoadilla #, seguida de un espacio, entre más almoadillas se agregen más pequeño es el encabezado.

3.3 Blockquotes

Para mostrar una cita se requiere utilizar el símbolo > por cada renglón de cita.

3.4 Listas

Las listas pueden ser numeradas o no numeradas, para usar las no numeradas se utilizan los siguientes símbolos: * - +, todos estos símbolos harán una lista con puntos gruesos como demarcadores.

Para el caso de una lista numerada se debe usar 1. se puede usar el mismo símbolo para demarcar todos los items de la lista o bien seguir la secuencia 2. 3. ...

3.5 Enlaces e imágenes

Para agregar un enlace se hace lo siguiente:

```
[nombre_que_aparece] (url)
```

Para agregar una imagen debemos:

```
![nombre_imagen] (url)
```

3.6 Agregar Código

Para agregar código se deben usar tres comillas de las inclinadas hacia la izquierda como se muestra:

```
```print( "hola Mundo" )```
```

### 3.7 Cuadros de check

para agregar cuadros vacíos y con check hacemos lo siguiente:

- [x] comprar leche
- [ ] comprar atun

### 3.8 Tablas

Para crear una tabla básica hacemos lo siguiente:

```
first header | second header
----- | -----
cell 1 | cell 2
cell a | cell b
```

## Bibliografía