

ROBERTO RODRÍGUEZ CARBONELL

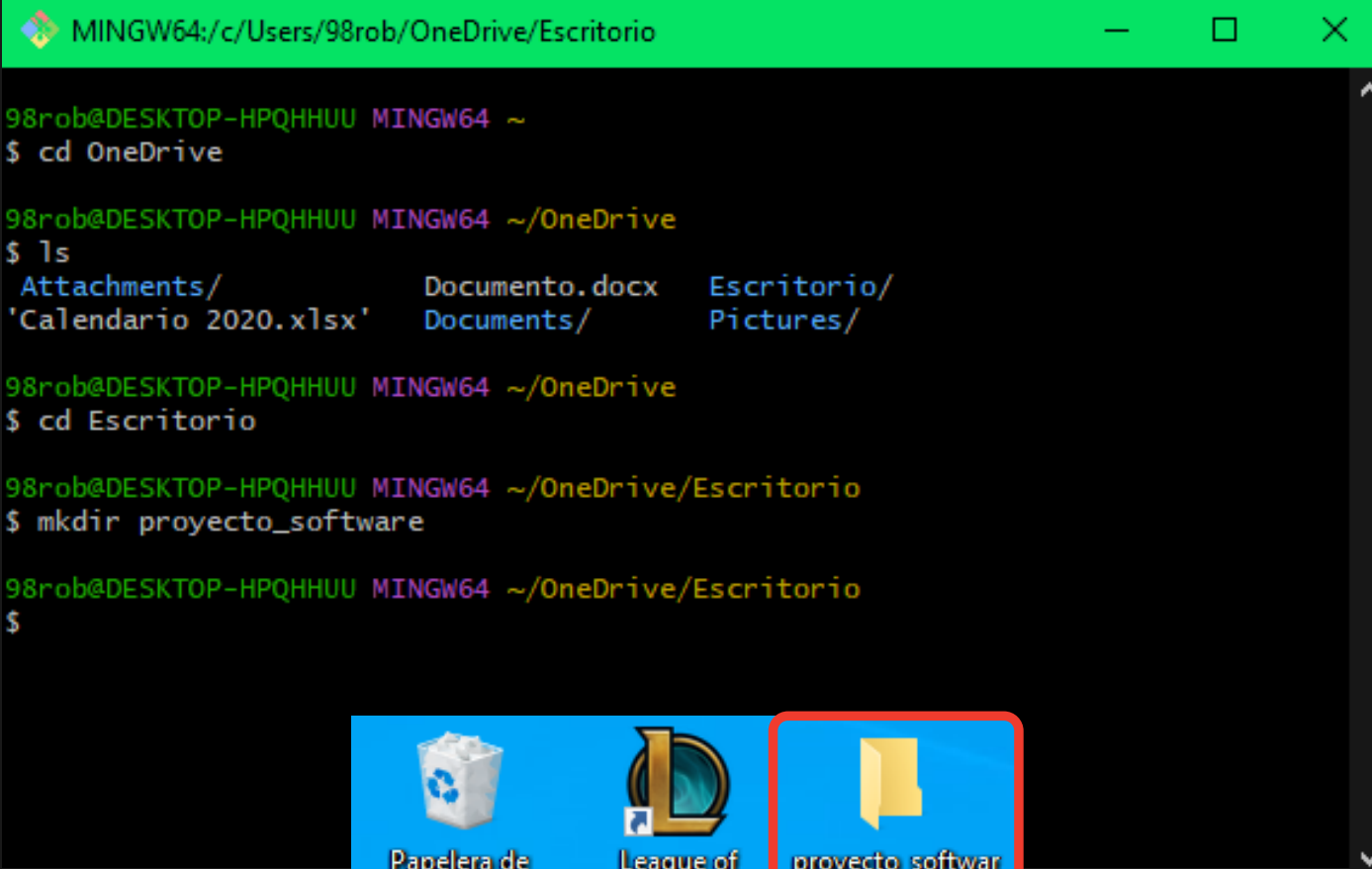
PRÁCTICA 2

GIT

1.- CREAMOS UN DIRECTORIO DE TRABAJO QUE SE LLAME PROYECTO_SOFTWARE.

Lo primero que tenemos que hacer es dirigirnos a la carpeta del directorio creado o que queramos crear a través del comando "cd".

Una vez ubicado en el directorio correspondiente creo la carpeta de nombre "proyecto_software" con el comando "mkdir nombreDelDirectorio".



```
MINGW64:/c/Users/98rob/OneDrive/Esitorio
98rob@DESKTOP-HPQHHUU MINGW64 ~
$ cd OneDrive

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive
$ ls
Attachments/      Documento.docx    Esitorio/
'Calendario 2020.xlsx' Documents/         Pictures/

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive
$ cd Esitorio

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Esitorio
$ mkdir proyecto_software

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Esitorio
$
```

The screenshot also shows a taskbar at the bottom with icons for 'Papera de reciclaje', 'League of Legends', and a folder named 'proyecto_software' which is highlighted with a red border.

COMANDOS:

```
cd OneDrive/Esitorio
mkdir proyecto_software
```

2.- INICIALIZAMOS UN REPOSITORIO VACÍO. ¿EN QUÉ RAMA ESTAMOS? ¿QUÉ COMANDO NOS PERMITE CONOCER ESE DATO?

Iniciamos el repositorio con el comando "git init".

Siempre que iniciemos un repositorio, la primera rama que se nos va a crear va a ser la rama "master", como se nos muestra en la fotografía.

Para ver en qué rama nos encontramos podemos usar el comando "git branch".

```
MINGW64:/c:/Users/98rob/OneDrive/Escritorio/proyecto_software
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio
$ ls
DiscoDuro.lnk*  PracticaGIT/  'Visual Studio Code.lnk'*  proyecto_software/
Discord.lnk*   Spotify.lnk*  desktop.ini

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio
$ cd proyecto_software/

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software
$ git init
Initialized empty Git repository in C:/Users/98rob/OneDrive/Escritorio/proyecto_software/.git/

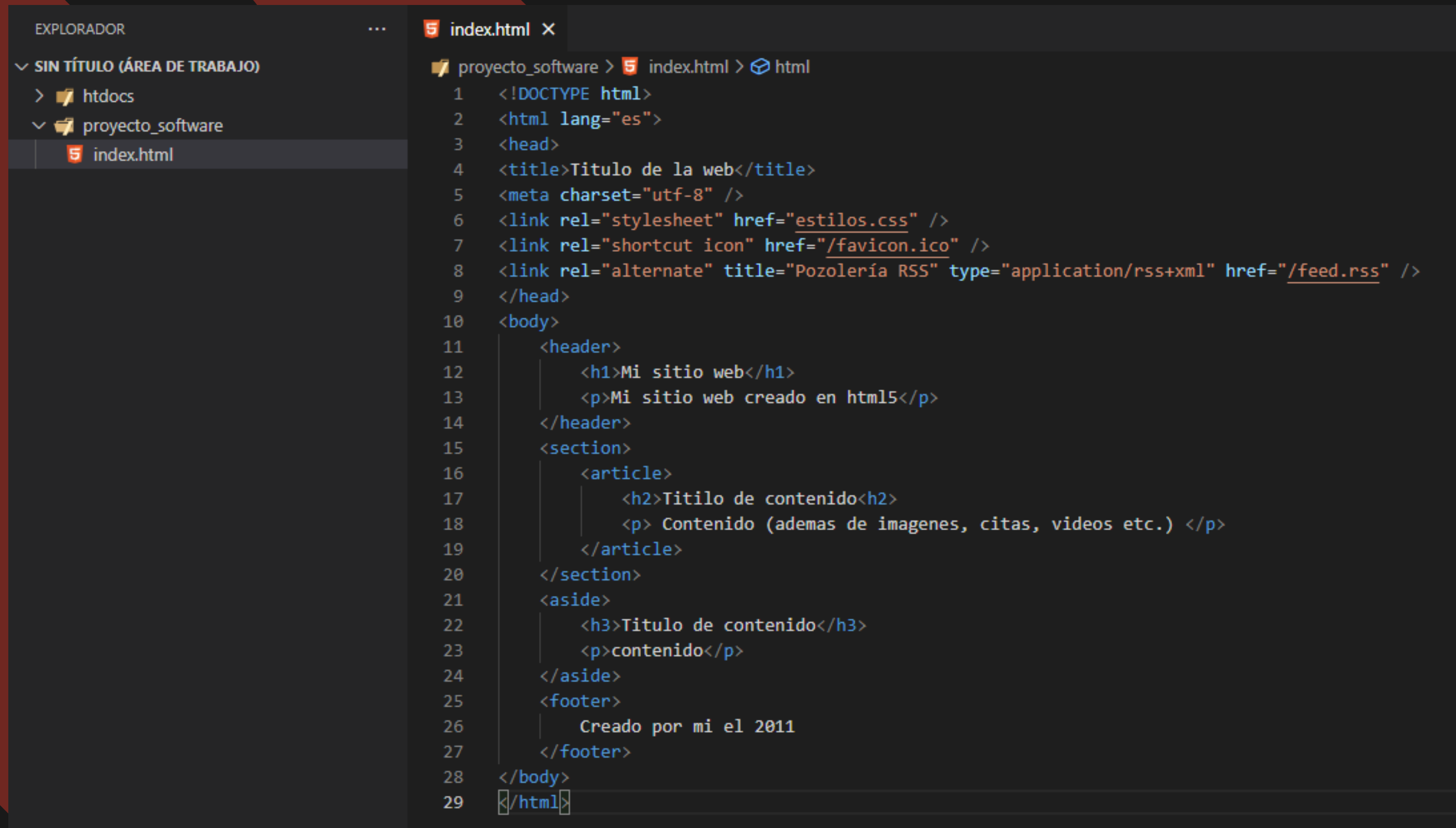
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$
```

COMANDOS:

git init

git branch

3.- CREAMOS UN ARCHIVO CON NOMBRE INDEX.HTML CON EL SIGUIENTE CONTENIDO.



The image shows a code editor interface with a dark theme. On the left, the 'EXPLORADOR' (Explorer) sidebar shows a project structure with folders 'htdocs' and 'proyecto_software'. Inside 'proyecto_software', the file 'index.html' is selected. The main editor area shows the content of 'index.html' with line numbers 1 through 29. The code is an HTML5 boilerplate with a title 'Titulo de la web', a meta charset of 'utf-8', and links to 'estilos.css', 'favicon.ico', and 'feed.rss'. The body contains a header with 'Mi sitio web', a main section with an article titled 'Titilo de contenido' and a paragraph, an aside with a title 'Titulo de contenido' and a paragraph, and a footer with 'Creado por mi el 2011'.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <title>Titulo de la web</title>
5 <meta charset="utf-8" />
6 <link rel="stylesheet" href="estilos.css" />
7 <link rel="shortcut icon" href="/favicon.ico" />
8 <link rel="alternate" title="Pozolería RSS" type="application/rss+xml" href="/feed.rss" />
9 </head>
10 <body>
11 <header>
12 <h1>Mi sitio web</h1>
13 <p>Mi sitio web creado en html5</p>
14 </header>
15 <section>
16 <article>
17 <h2>Titilo de contenido</h2>
18 <p> Contenido (ademas de imagenes, citas, videos etc.) </p>
19 </article>
20 </section>
21 <aside>
22 <h3>Titulo de contenido</h3>
23 <p>contenido</p>
24 </aside>
25 <footer>
26 Creado por mi el 2011
27 </footer>
28 </body>
29 </html>
```

4.- SI EJECUTO LA ORDEN GIT STATUS, ¿QUÉ OBTENGO?

Cuando ejecutemos el comando "git status" nos va a aparecer el estado en el que se encuentra en repositorio. Si tenemos algún cambio sin añadir, nos indica en qué documento no se han añadido los cambios.

```
MINGW64:/c/Users/98rob/OneDrive/Escritorio/proyecto_software

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

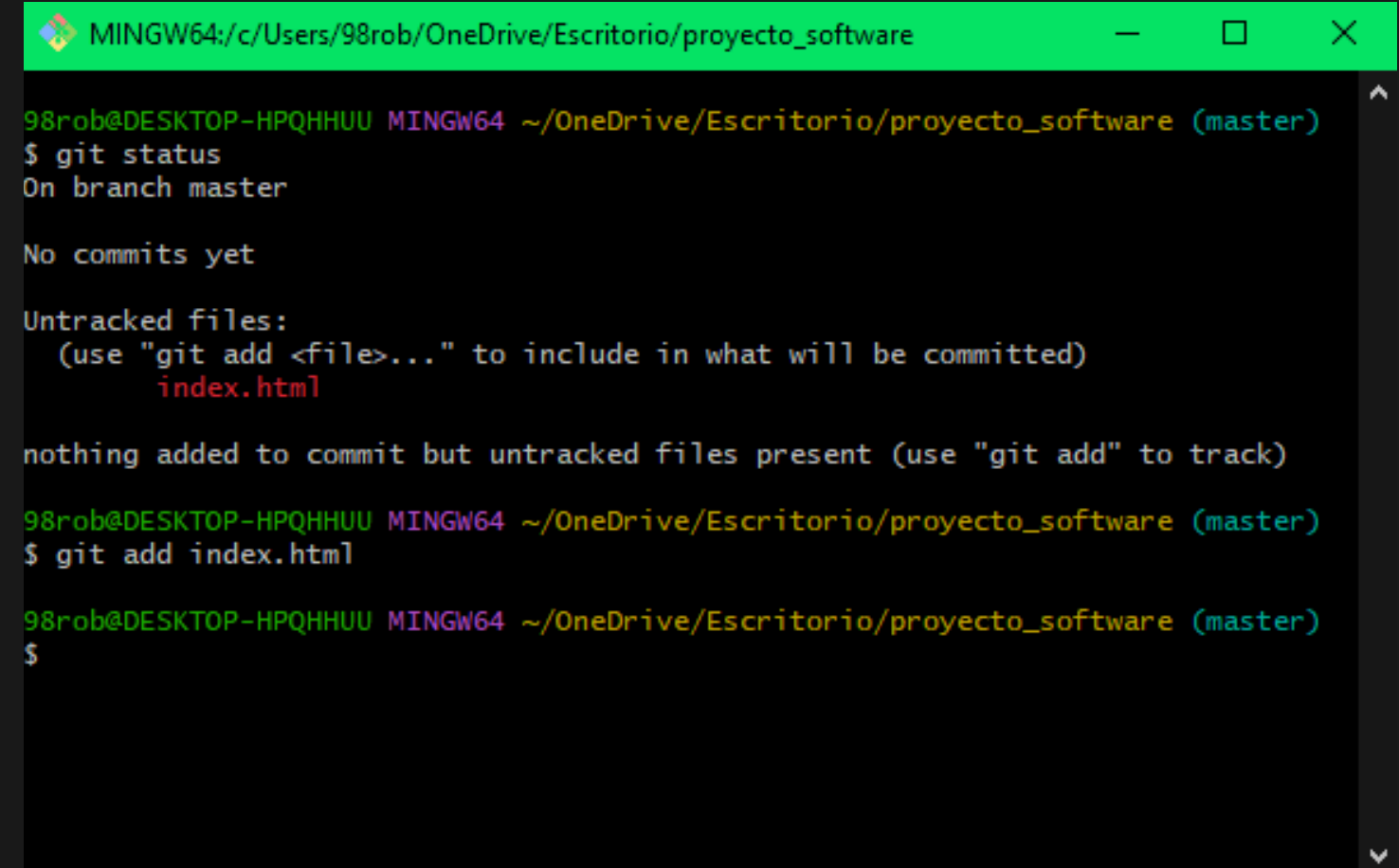
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$
```

COMANDOS:

git status

5.- AÑADIMOS EL FICHERO INDEX.HTML AL ÁREA DE PREPARACIÓN.

En la imagen nos muestra que no hemos añadido el archivo "index.html" por lo que, para añadirlo, tenemos que introducir el siguiente comando: "git add index.html".

A screenshot of a terminal window with a green title bar. The title bar text is 'MINGW64:/c/Users/98rob/OneDrive/Escritorio/proyecto_software'. The terminal shows the following commands and output:

```
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git add index.html

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$
```

COMANDOS:

```
git add index.html
```

6.- CREO UN COMMIT CON EL SIGUIENTE MENSAJE "COMMIT INICIAL".

Ahora voy a crear un "commit".

La sentencia "commit" sirve para dar por finalizado una unidad de trabajo.

Para hacer un "commit" hay que escribir el siguiente comando: "git commit -m Commit inicial".

```
MINGW64:/c/Users/98rob/OneDrive/Escritorio/proyecto_software
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git add index.html

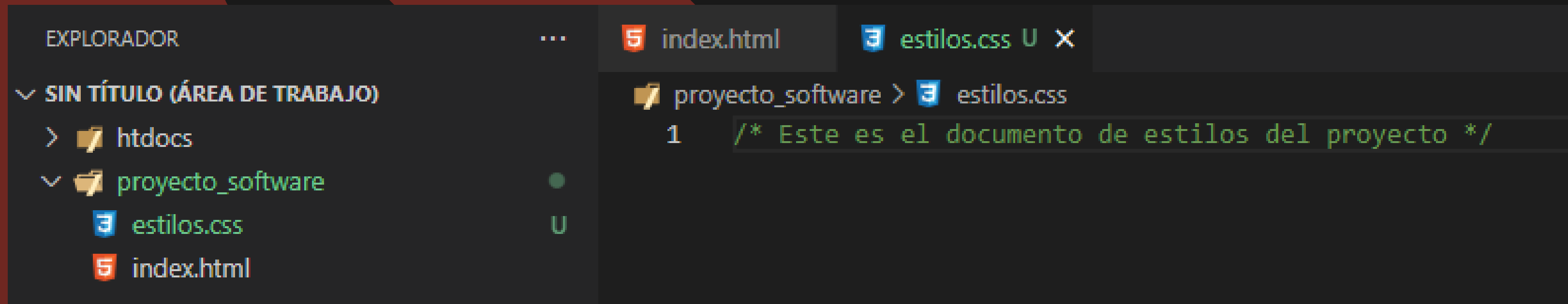
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git commit -m "Commit inicial"
[master (root-commit) 29bae7c] Commit inicial
1 file changed, 29 insertions(+)
create mode 100644 index.html

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$
```

COMANDOS:

```
git commit -m "Commit inicial".
```

7.- CREO EL FICHERO ESTILOS.CSS E INSERTO EL SIGUIENTE CONTENIDO.



8.- MODIFICO EL FICHERO INDEX.HTML Y MODIFICO:

Titulo de la web

POR EL SIGUIENTE CONTENIDO:

Proyecto Software

```
4 | <title>Proyecto Software</title>
```

9.- SI EJECUTO LA ORDEN GIT STATUS, ¿QUÉ OBTENGO?

Con el comando "git status" observamos el estado del git, y vemos si al algún cambio que no hayamos dado por finalizado.

```
MINGW64:/c/Users/98rob/OneDrive/Escritorio/proyecto_software
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        estilos.css

no changes added to commit (use "git add" and/or "git commit -a")
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$
```

COMANDOS:

git status

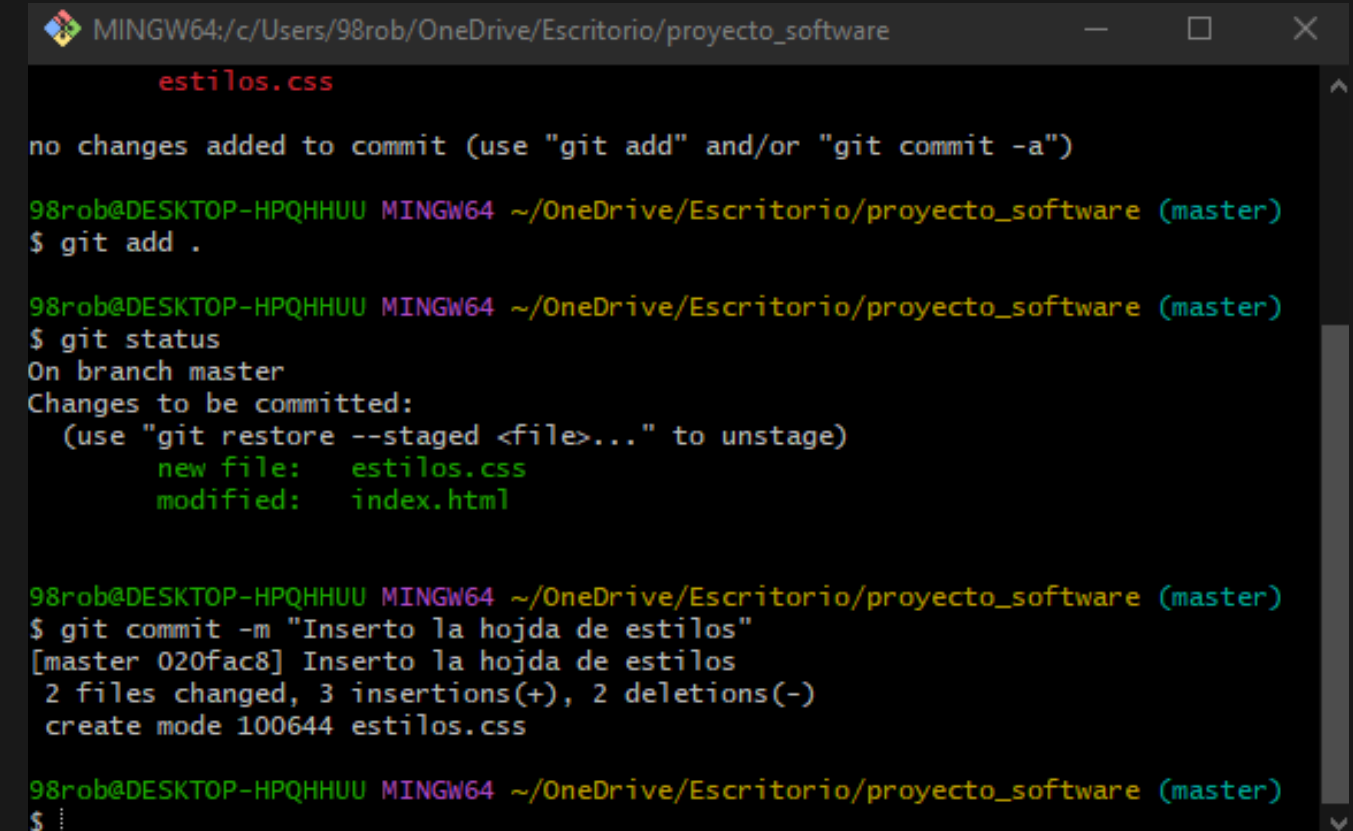
**10.- AGREGO LOS ARCHIVOS NECESARIOS
AL ÁREA DE PREPARACIÓN.**

**11.- CREO UN COMMIT CON EL SIGUIENTE
MENSAJE “INSERTO LA HOJA DE ESTILOS” .**

Con el comando "git add ." añadimos todos los cambios que no hayan sido registrados como finalizados.

Tras eso, he usado el comando "git status" que ya habíamos visto anteriormente para confirmar si los cambios se han realizado con éxito.

Además, también he aprovechado para realizar el "commit".

A screenshot of a terminal window titled 'MINGW64:/c/Users/98rob/OneDrive/Escritorio/proyecto_software'. The terminal shows the following sequence of commands and output: 1. A file named 'estilos.css' is shown at the top. 2. The command 'git add .' is entered, resulting in 'no changes added to commit (use "git add" and/or "git commit -a")'. 3. The command 'git status' is entered, showing 'On branch master' and 'Changes to be committed: (use "git restore --staged <file>..." to unstage) new file: estilos.css modified: index.html'. 4. The command 'git commit -m "Inserto la hojda de estilos"' is entered, resulting in '[master 020fac8] Inserto la hojda de estilos' and '2 files changed, 3 insertions(+), 2 deletions(-) create mode 100644 estilos.css'. 5. The prompt '\$' is shown at the bottom.

```
MINGW64:/c/Users/98rob/OneDrive/Escritorio/proyecto_software
estilos.css

no changes added to commit (use "git add" and/or "git commit -a")

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git add .

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   estilos.css
        modified:   index.html

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ git commit -m "Inserto la hojda de estilos"
[master 020fac8] Inserto la hojda de estilos
2 files changed, 3 insertions(+), 2 deletions(-)
create mode 100644 estilos.css

98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$
```

COMANDOS:

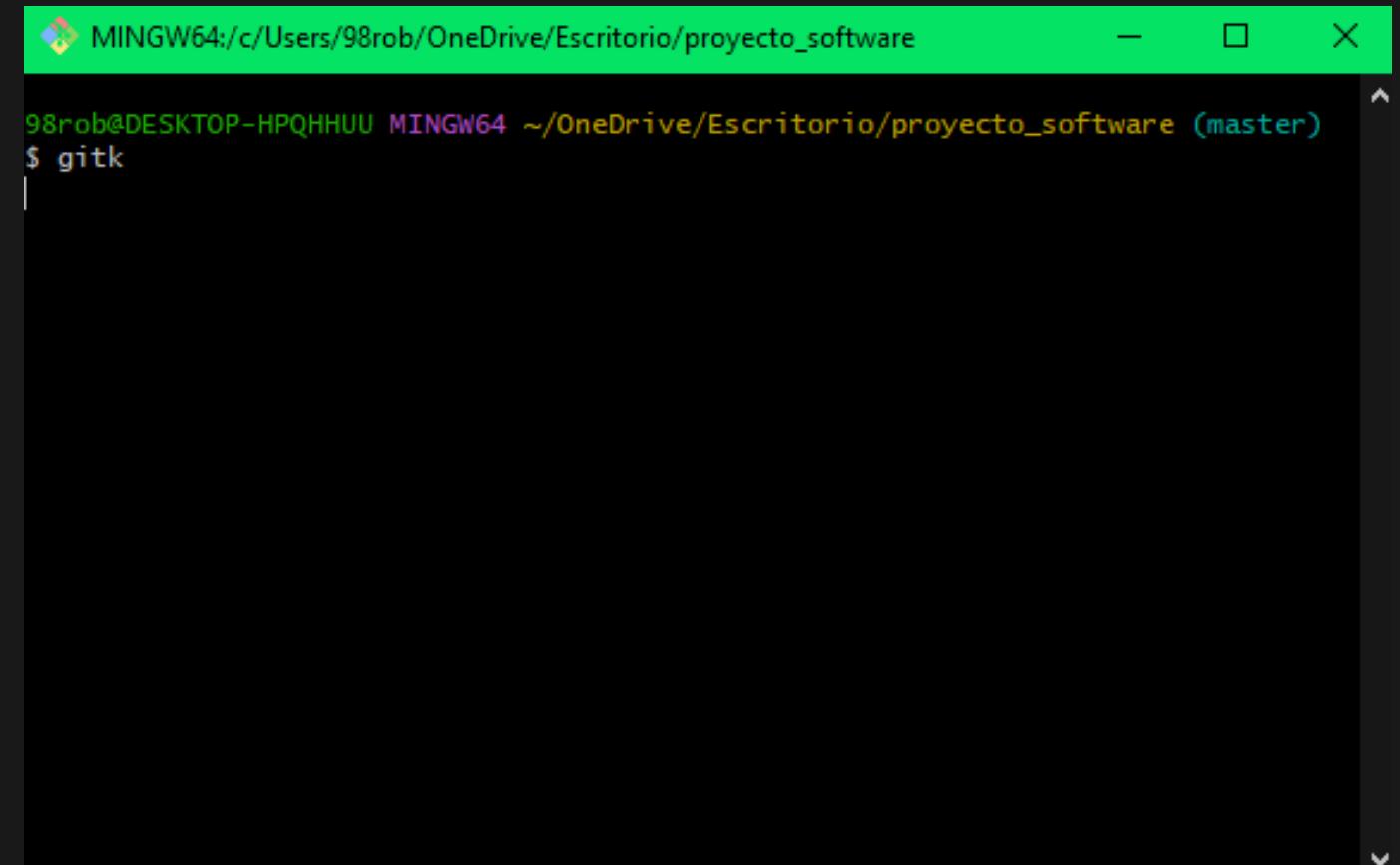
git add .

git commit -m "Inserto las hojas de estilos"

12.- ¿CON QUÉ COMANDO PUEDO VER LOS COMMITS REALIZADOS?

Con el comando "gitk" podemos ver todos los "commits" realizados con todos sus detalles.

Toda esta información nos aparecerá en una ventana nueva que se nos abrirá tras ingresar el comando.

A screenshot of a terminal window with a green title bar. The title bar text is 'MINGW64:/c/Users/98rob/OneDrive/Escritorio/proyecto_software'. The terminal content shows the prompt '98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)' followed by the command '\$ gitk' on a new line. The terminal has a scrollbar on the right side.

```
MINGW64:/c/Users/98rob/OneDrive/Escritorio/proyecto_software
98rob@DESKTOP-HPQHHUU MINGW64 ~/OneDrive/Escritorio/proyecto_software (master)
$ gitk
```

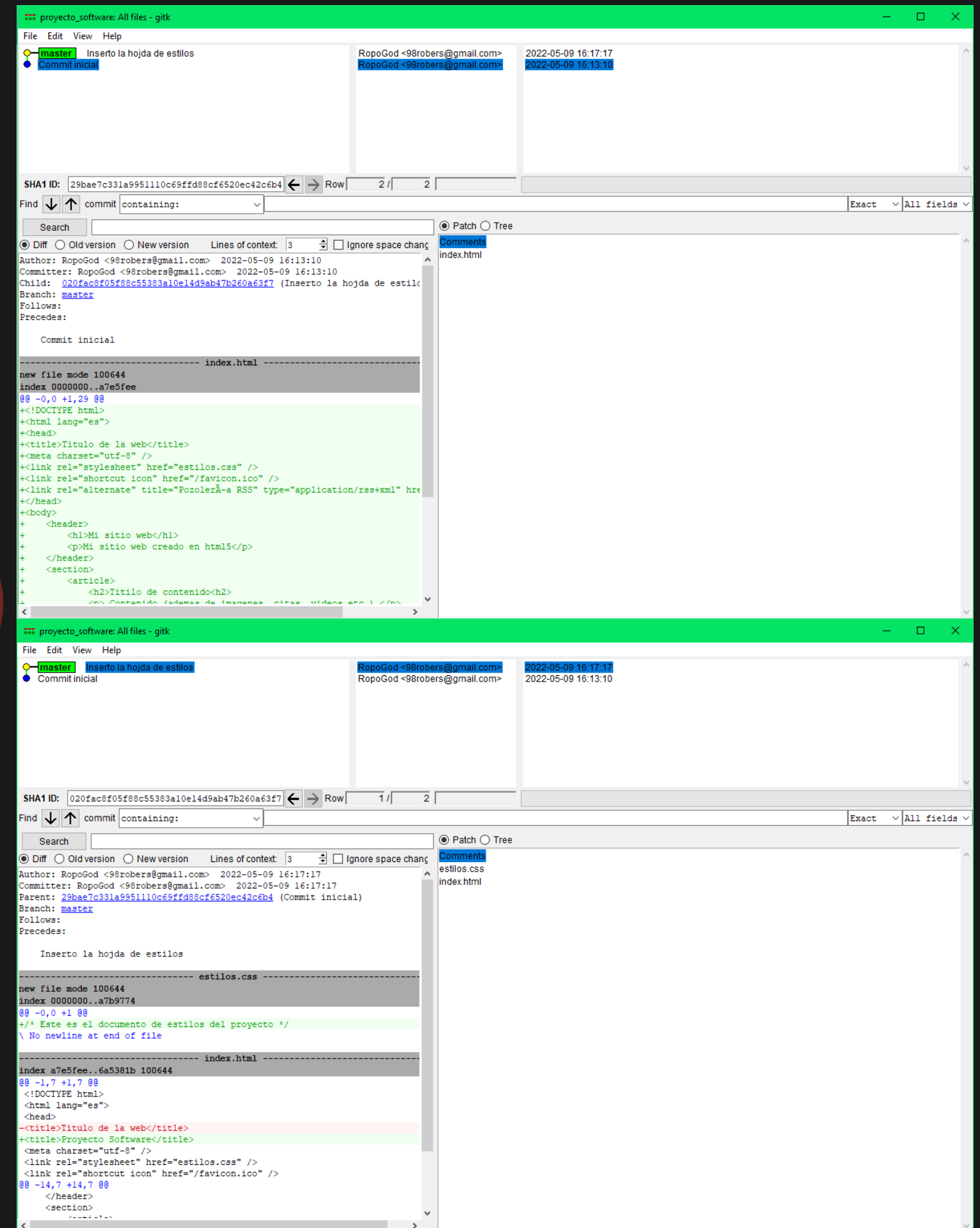
COMANDOS:

gitk

12.- ¿CON QUÉ COMANDO PUEDO VER LOS COMMITS REALIZADOS?

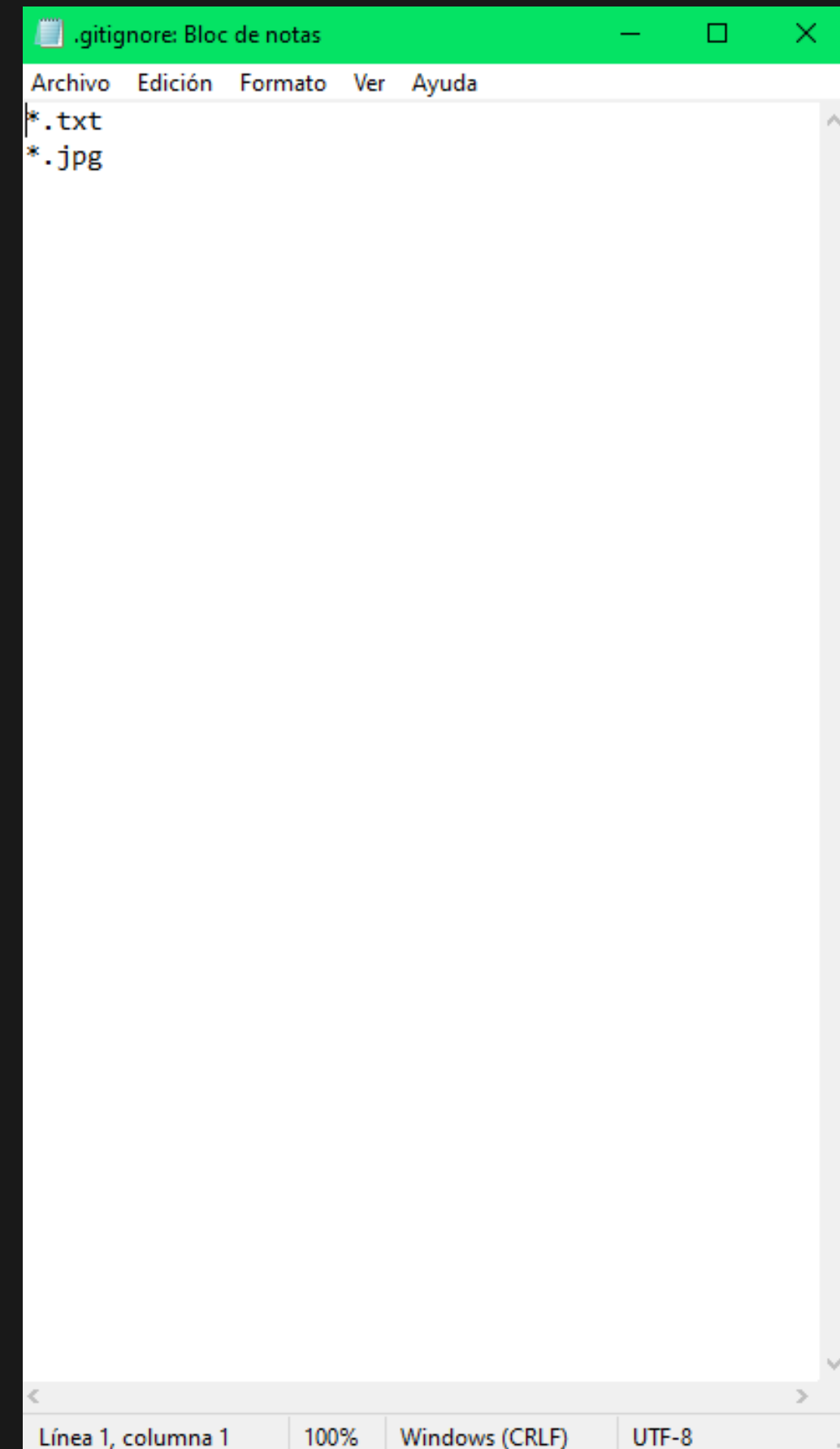
Con el comando "gitk" podemos ver todos los "commits" realizados con todos sus detalles.

Toda esta información nos aparecerá en una ventana nueva que se nos abrirá tras ingresar el comando.



13.- CONFIGURA EL REPOSITORIO PARA QUE NO TENGA EN CUENTA LAS SIGUIENTES EXTENSIONES DE FICHEROS: .TXT Y .JPG.

Para esto, tenemos que crear un archivo con el nombre ".gitignore". Dentro de este archivo introducimos lo que queremos que git ignore. En mi caso introducimos lo que aparece en la fotografía.



```
.gitignore: Bloc de notas
Archivo Edición Formato Ver Ayuda
*.txt
*.jpg
Línea 1, columna 1 100% Windows (CRLF) UTF-8
```

The background features a dark gray field with a large, light gray geometric shape on the left side, consisting of a vertical line and a diagonal line meeting at a point. A stylized, dark gray 'F' logo is positioned on the left, partially overlapping the light gray shape. The word 'FINN' is written in a bold, red, sans-serif font across the center of the image.

FINN