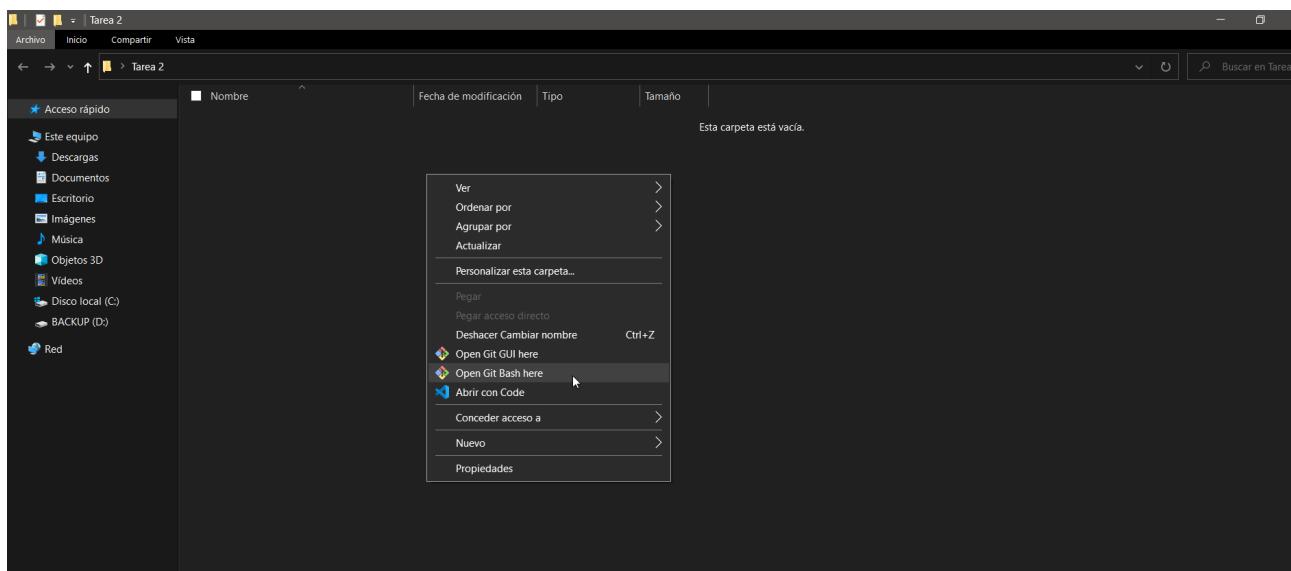


TAREA 2 SOBRE GIT Y GITHUB

Bienvenidos al maravilloso mundo de Git. En esta práctica vamos a poner a prueba los conocimientos básicos sobre los comandos y como plus vamos a tener que hacer algo de desarrollo web.

- **Clona el repositorio <https://github.com/EnriquePicasso/ActGIT2.git> en tu equipo. Este repositorio contiene la cheat-sheet de GIT para tener los comandos a mano.**

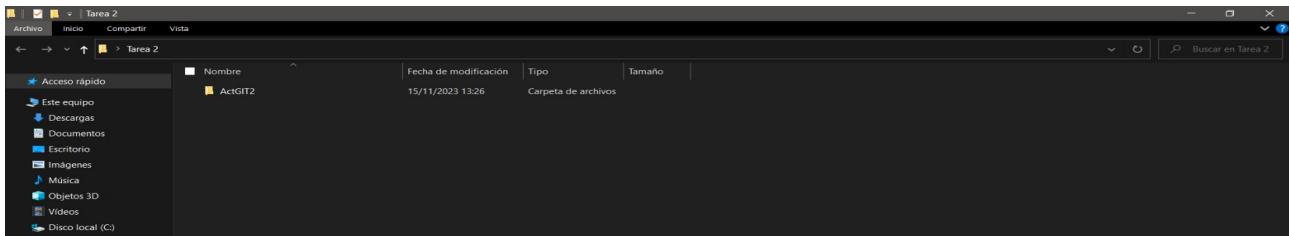
Para comenzar nuestro proyecto deberemos clonar en una carpeta de nuestra elección un repositorio que está en GitHub y que contiene las instrucciones del ejercicio. Cuando estemos situados en la carpeta en cuestión daremos al botón derecho del ratón y pincharemos sobre “Open Git Bash Here”.



Una vez tengamos desplegado nuestro terminal de git usaremos la instrucción git clone repositorio que queremos clonar.

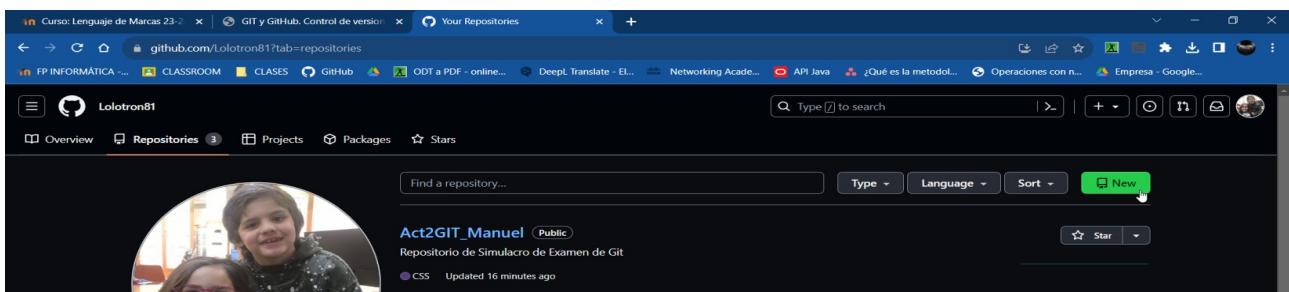
A screenshot of a terminal window titled 'MINGW64;/c/Users/baske/Desktop/Tarea 2'. The command \$ git clone https://github.com/EnriquePicasso/ActGIT2.git is being typed in. The output shows the cloning process: 'Cloning into 'ActGIT2'...', 'remote: Enumerating objects: 3, done.', 'remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 3', 'Receiving objects: 100% (3/3), 747.90 KiB | 2.61 MiB/s, done.' The prompt '\$' is visible at the bottom.

Y veremos que en nuestra carpeta hasta ahora vacía ha aparecido un repositorio



- **Crea un nuevo repositorio local llamado Act2GIT_Tunombre**

Para esta tarea tenemos dos opciones. Crear una carpeta directamente dentro de la carpeta que ya tenemos, junto al repositorio clonado, o bien nos vamos a GitHub y creamos un repositorio nuevo. Esta opción es más cómoda. Nos iremos a nuestro GitHub y pincharemos sobre nuevo repositorio.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * Repository name *

Lolotron81 / Act2Git_Tunombre

Act2Git_Tunombre is available.

Great repository names are short and memorable. Need inspiration? How about [scaling-system](#) ?

Description (optional)

Tarea 2 de Git y GitHub

Public Private

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

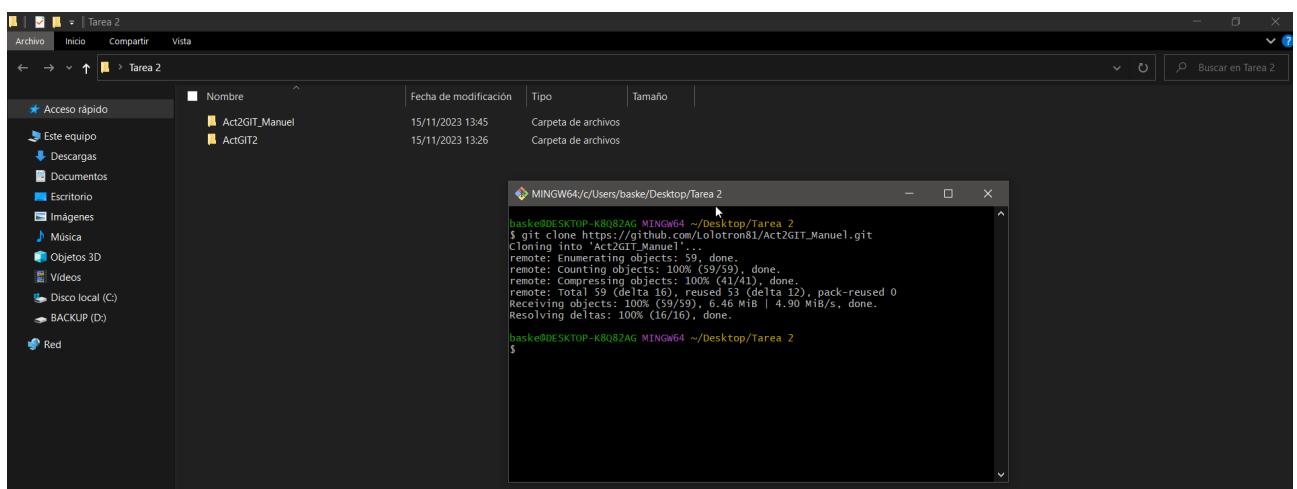
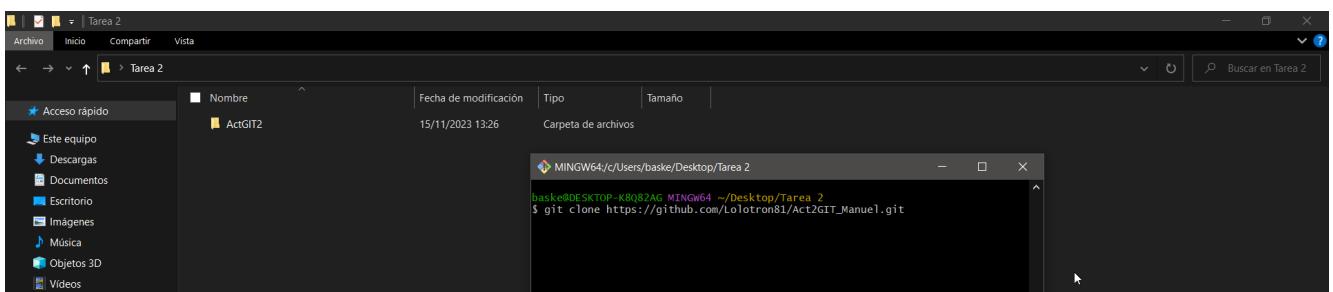
A license tells others what they can and can't do with your code. Learn more about licenses.

This will set main as the default branch. Change the default name in your settings.

(i) You are creating a public repository in your personal account.

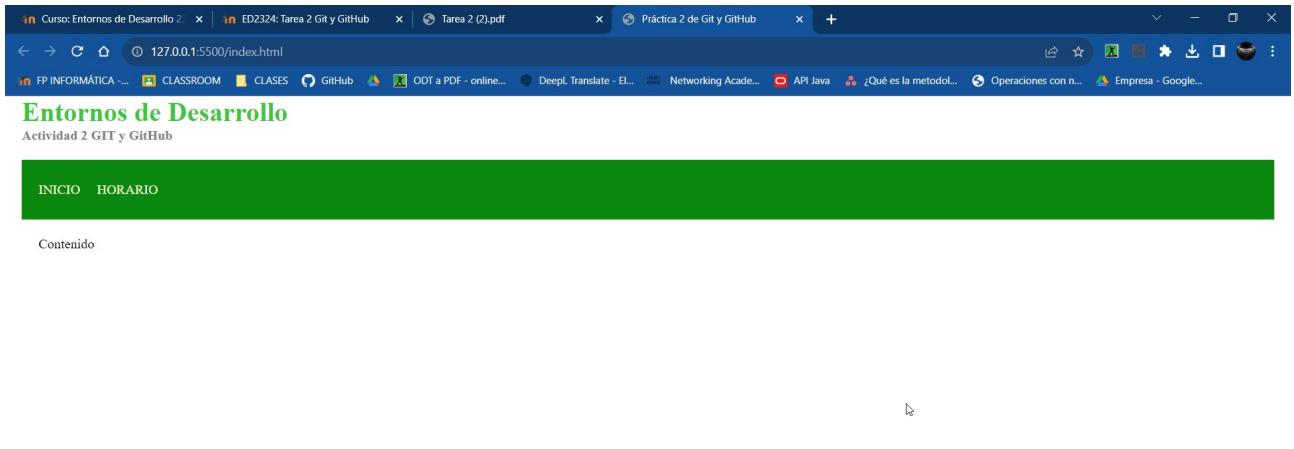
Create repository

Una vez hayamos creado nuestro repositorio en GitHub entraremos en él y hay un botón verde que pone “code”. Si pinchamos sobre él en el desplegable podremos copiar la ruta de nuestro repositorio. La copiamos y faremos el mismo proceso que al principio. Volvemos a nuestra carpeta que contiene el repositorio con los ejercicios e invocamos al Git Bash.



- Crea una página web con una cabecera, un menú en la parte superior que contenga las opciones de Inicio y Horario y un footer con el texto “Desarrollado por IES Pablo Picasso”.

Por medio del potente editor de texto Visual Studio Code creamos una pequeña web que se ajuste a lo que se nos pide en el enunciado dando como resultado:



Pequeño ejemplo del código utilizado:

```

File Edit Selection View Go Run Terminal Help < > Página Web
OPEN EDITORS index.html ...
PÁGINA WEB > images > styles.css
> styleshorario.css > stylesobremi.css
horario.html index.html sobremi.html

```

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>Práctica 2 de Git y GitHub</title>
    <link rel="stylesheet" href="./styles/styles.css">
</head>
<body>
    <header>
        <h1 class="letras"> Entornos de Desarrollo </h1>
        <h4 class="letras"> Actividad 2 GIT y GitHub </h4>
    </header>
    <nav class="barraNavegacion">
        <div class="inicio">
            <a href="./index.html" target="_blank"> INICIO </a>
        </div>
        <div class="horario">
            <a href="./horario.html" target="_blank"> HORARIO </a>
        </div>
        <div class="mi">
            <a href="./sobremi.html" target="_blank"> SOBRE MI </a>
        </div>
    </nav>
    
    <footer>
        <div class="pie">
            <h3> Desarrollado por IES Pablo Picasso </h3>
        </div>
    </footer>
</body>
</html>

```

```

# styles.css
# styles.css > # barraNavegacion
1 * {
2     margin: 0;
3 }
4 .letras {
5     color: #rgb(60, 199, 60);
6     margin-left: 20px;
7 }
8 .letras2 {
9     color: #grey;
10    margin-left: 20px;
11 }
12 a {
13     text-decoration: none;
14     color: #beige;
15 }
16 .barraNavegacion {
17     background-color: #rgb(11, 137, 14);
18     display: flex;
19     justify-content: left;
20     align-items: center;
21     flex-direction: row;
22     height: 10vh;
23     margin: 20px;
24     padding: 20px;
25 }
26
27
28
29
30
31
32
33
34
35
36
37

```

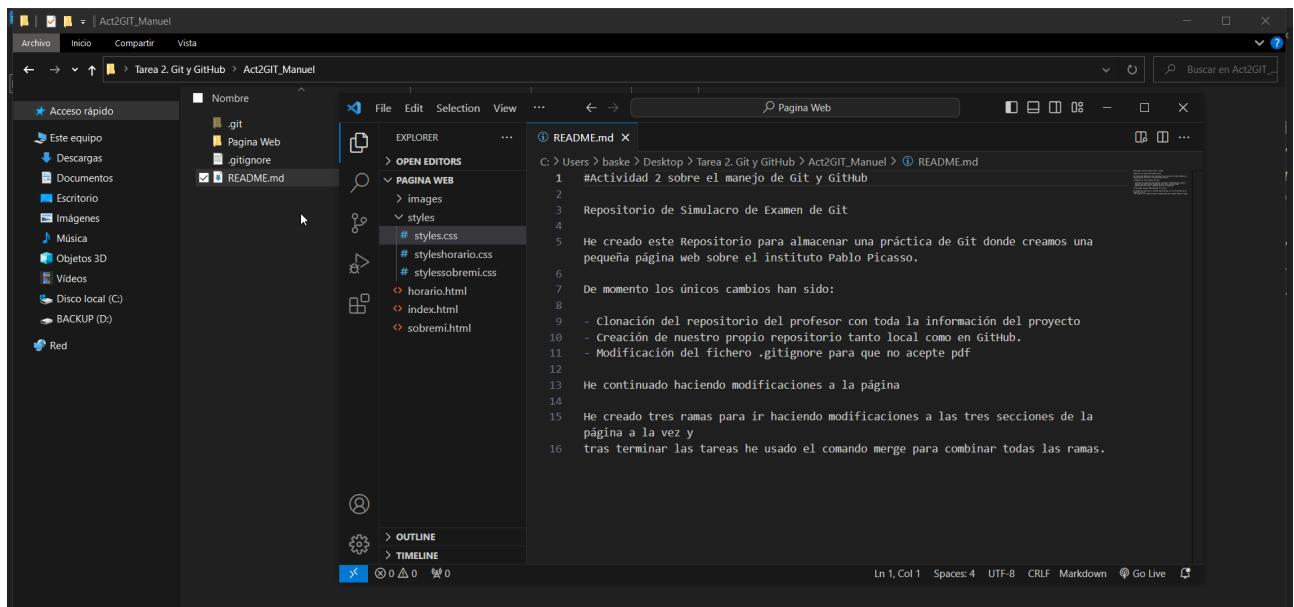
A git repository was found in the parent folders of the workspace or the open file(s). Would you like to open the repository?

Source: Git (Extension)

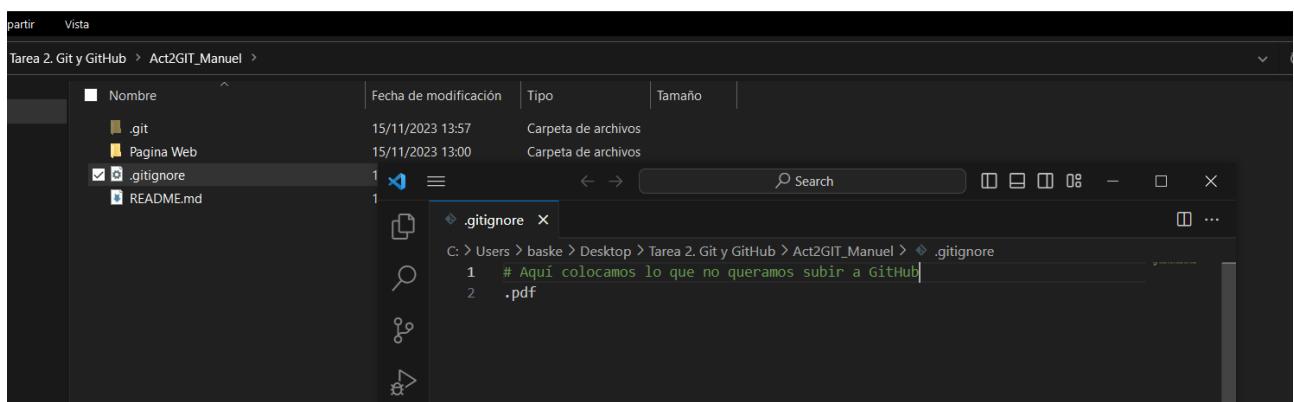
Yes Always Never

- Añade un `readme.md` con el contenido necesario para que el proyecto esté bien documentado en GitHub. Y un archivo `.gitignore` para no subir los `.pdf`

Ya dentro de nuestro repositorio y donde hemos creado la web creamos un fichero que por ejemplo sea de texto (`fichero.txt`) y lo renombramos a `README.md`. Una vez lo tengamos creado lo abrimos con Visual Studio Code y usando el lenguaje de marca `MarkDown` usamos dicho fichero como una especie de manual sobre nuestro trabajo.



También se nos pide que creamos un fichero `-gitignore`. Este archivo sirve para restringir todo tipo de archivos según nuestras preferencias. Para crearlo recurrimos a un fichero de texto y le cambiamos el nombre dejándolo simplemente como `.gitignore`. Editamos este archivo con el visual Studio Code y simplemente escribimos lo que queremos ignorar para que no se suba a GitHub. En este caso vamos a restringir los pdfs por lo que deberemos escribir `.pdf`



- **Realiza un commit para guardar el estado inicial del proyecto.**

Vamos a dejar constancia de todos los cambios que ya hemos hecho. Para poder usar el comando commit previamente usaremos el comando add de la siguiente forma:

- `git add .` → Esto subirá todos los cambios a la zona de Staging que es el paso previo para poder subir a GitHub
- `git commit -m "mensaje"` → Con esta expresión dejamos un pequeño mensaje contando los cambios que hemos hecho de forma resumida

```

MINGW64:/c/Users/baske/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ git add .
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ git commit -m "Nuestro Proyecto ha comenzado"
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ |

```

- **Etiqueta el repositorio como V1.0**

Para poder poner etiquetas hacemos uso del comando tag seguido del nombre que queremos dar a nuestra etiqueta.

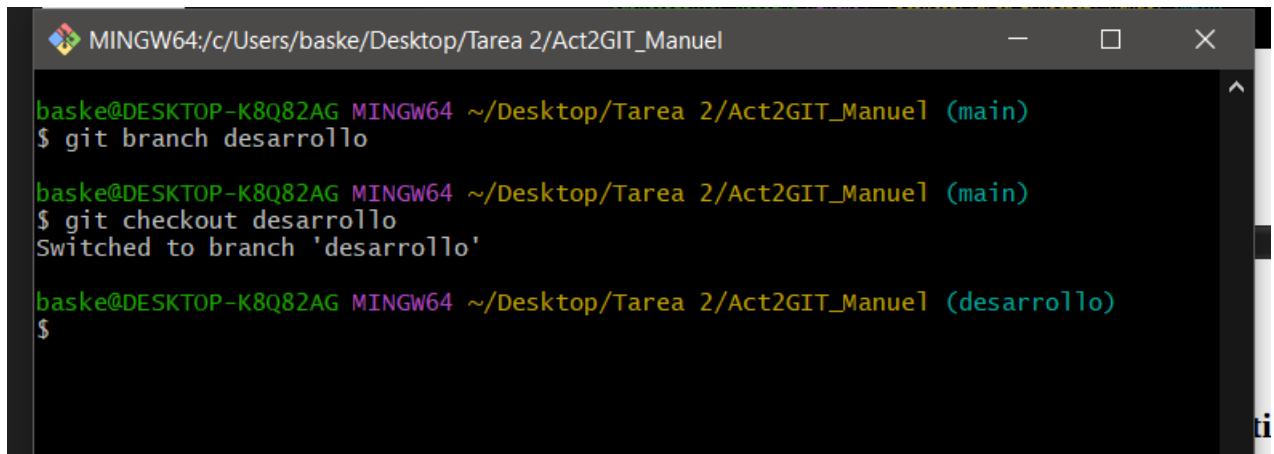
```

MINGW64:/c/Users/baske/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ git add .
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ git commit -m "Nuestro Proyecto ha comenzado"
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ git tag V1.0
fatal: tag 'V1.0' already exists
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ git tag V1.0

```

- Crea una rama llamada desarrollo desde la rama principal. Continúa por la rama de desarrollo y crea la página Horario, donde tendrás una tabla con tu horario de clase. Registra esos cambios en el repositorio local.

Para crear una rama usaremos el comando branch seguido del nombre de la rama y cuando queramos cambiar entre ramas usaremos el comando checkout.



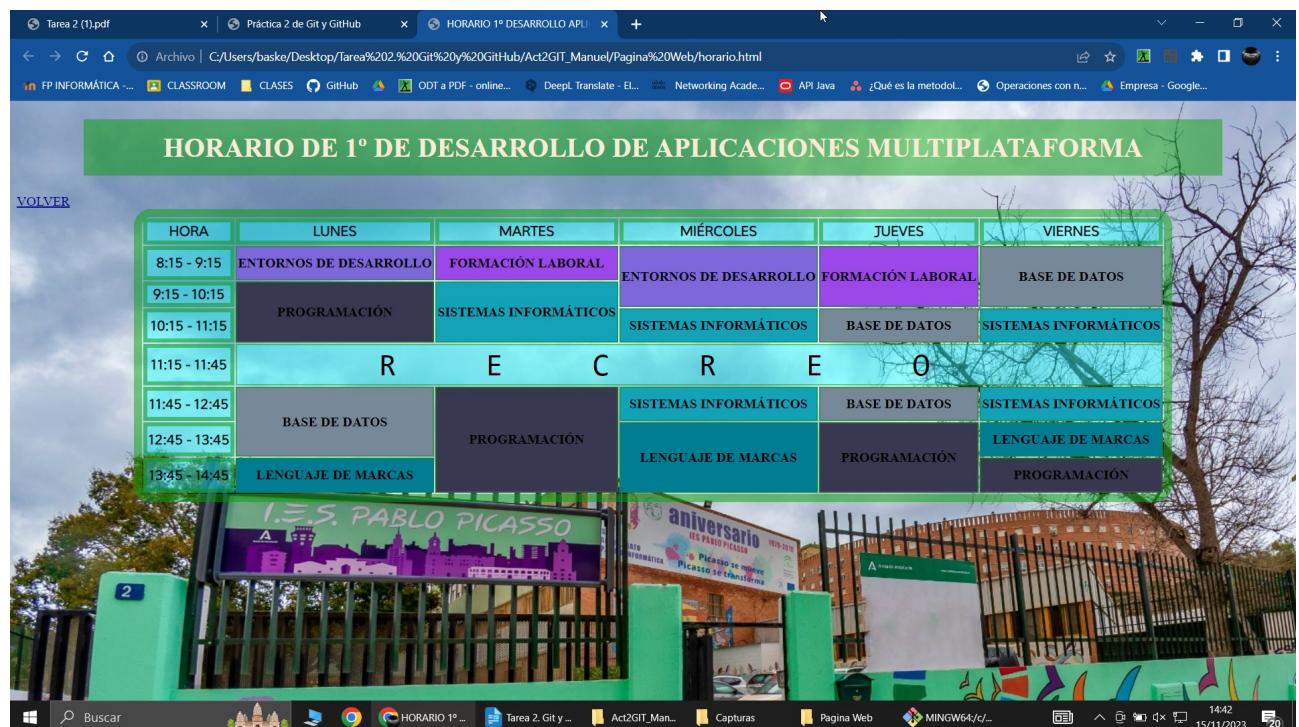
```
MINGW64:/c/Users/baske/Desktop/Tarea 2/Act2GIT_Manuel
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ git branch desarrollo

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2/Act2GIT_Manuel (main)
$ git checkout desarrollo
Switched to branch 'desarrollo'

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2/Act2GIT_Manuel (desarrollo)
$
```

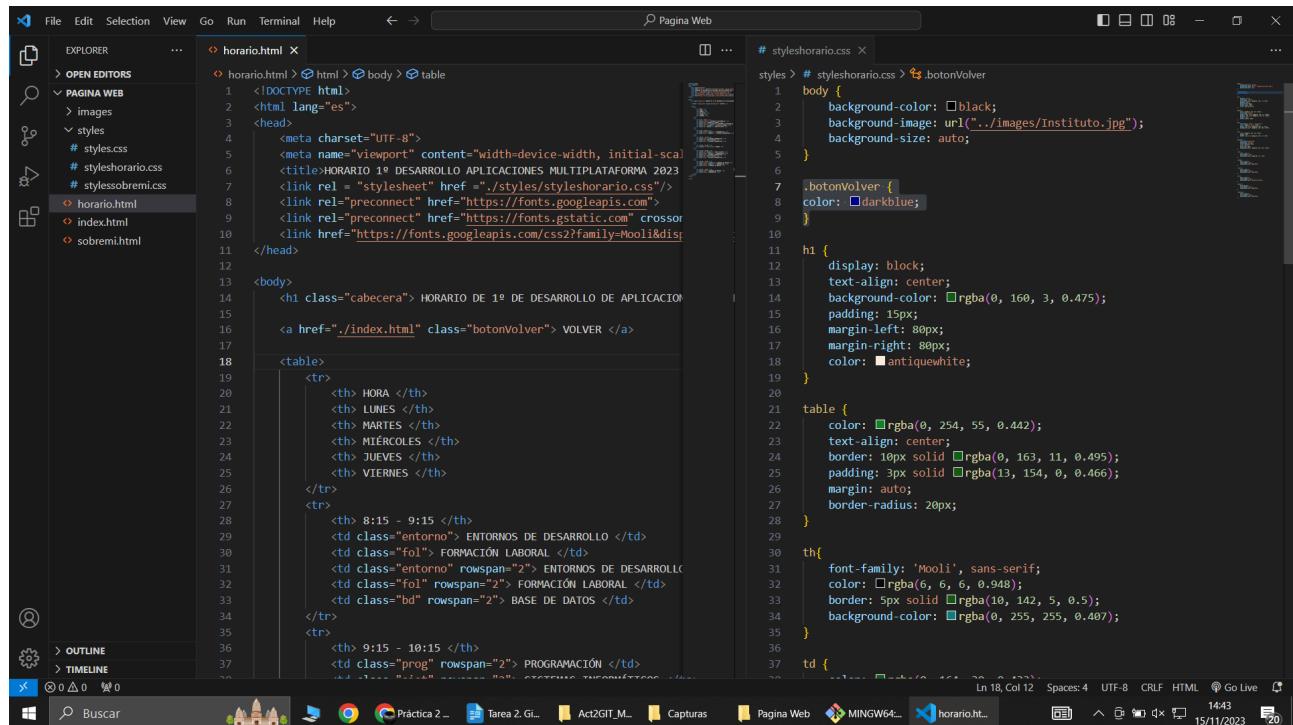
Estando sobre esta rama crearemos nuevamente con Studio Code una segunda página web que contendrá un horario del actual curso de 1º de Desarrollo de Aplicaciones Multiplataforma.

Así ha quedado:



HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
8:15 - 9:15	ENTORNOS DE DESARROLLO	FORMACIÓN LABORAL	ENTORNOS DE DESARROLLO	FORMACIÓN LABORAL	
9:15 - 10:15					BASE DE DATOS
10:15 - 11:15	PROGRAMACIÓN	SISTEMAS INFORMÁTICOS	SISTEMAS INFORMÁTICOS	BASE DE DATOS	SISTEMAS INFORMÁTICOS
11:15 - 11:45	R	E	C	R	O
11:45 - 12:45	BASE DE DATOS		SISTEMAS INFORMÁTICOS	BASE DE DATOS	SISTEMAS INFORMÁTICOS
12:45 - 13:45		PROGRAMACIÓN	LENGUAJE DE MARCAS	PROGRAMACIÓN	LENGUAJE DE MARCAS
13:45 - 14:45	LENGUAJE DE MARCAS				PROGRAMACIÓN

Y ésta es una pequeña muestra del código:



```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>HORARIO 1º DESARROLLO APLICACIONES MULTIPLAFORMA 2023</title>
    <link rel="stylesheet" href="./styles/stylehorario.css"/>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin="anonymous">
    <link href="https://fonts.googleapis.com/css2?family=Mooli&display=block">
</head>
<body>
    <h1 class="cabecera"> HORARIO DE DESARROLLO DE APLICACIONES</h1>
    <a href="./index.html" class="botonVolver"> VOLVER </a>
    <table>
        <tr>
            <th> HORA </th>
            <th> LUNES </th>
            <th> MARTES </th>
            <th> MIÉRCOLES </th>
            <th> JUEVES </th>
            <th> VIERNES </th>
        </tr>
        <tr>
            <th> 8:15 - 9:15 </th>
            <td class="entorno"> ENTORNOS DE DESARROLLO </td>
            <td class="fol"> FORMACIÓN LABORAL </td>
            <td class="entorno" rowspan="2"> ENTORNOS DE DESARROLLO </td>
            <td class="fol" rowspan="2"> FORMACIÓN LABORAL </td>
            <td class="bd" rowspan="2"> BASE DE DATOS </td>
        </tr>
        <tr>
            <th> 9:15 - 10:15 </th>
            <td class="prog" rowspan="2"> PROGRAMACIÓN </td>
        </tr>
    </table>
</body>
```

```
# stylehorario.css
body {
    background-color: black;
    background-image: url("../images/Instituto.jpg");
    background-size: auto;
}

.botonVolver {
    color: darkblue;
}

h1 {
    display: block;
    text-align: center;
    background-color: rgba(0, 160, 3, 0.475);
    padding: 15px;
    margin-left: 80px;
    margin-right: 80px;
    color: antiquewhite;
}

table {
    color: rgba(0, 254, 55, 0.442);
    text-align: center;
    border: 1px solid rgba(0, 163, 11, 0.495);
    padding: 3px solid rgba(13, 154, 0, 0.466);
    margin: auto;
    border-radius: 20px;
}

th {
    font-family: 'Mooli', sans-serif;
    color: rgba(6, 6, 6, 0.948);
    border: 5px solid rgba(10, 142, 5, 0.5);
    background-color: rgba(0, 255, 255, 0.407);
}

td {
```

Para registrar los cambios en nuestro repositorio local usaremos el comando git add . para subir los cambios a la zona de staging y hacemos un nuevo commit registrando los cambios.

- **Crea otra rama llamada feature1 desde desarrollo o Añade un nuevo elemento de menú que sea “Sobre Mí” y crea una página con algún dato sobre ti (Gustos, fecha de nacimiento...) o Guarda esos cambios en el repositorio local.**

Usamos el comando branch para crear la rama y a continuación usamos el comando checkout para situarnos en la rama recién creada.



```
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel
$ git branch feature1
fatal: a branch named 'feature1' already exists

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel
$ git checkout feature1
Switched to branch 'feature1'

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (feature1)
$ |
```

Ya estando en nuestra rama feature vamos a crear una tercera página web con algo de información sobre mí. Así ha quedado:



y aquí una pequeña muestra del código:

A screenshot of the Visual Studio Code (VS Code) interface. On the left, the Explorer sidebar shows files like 'sobremi.html', 'index.html', and 'horario.html'. The main editor area displays the content of 'sobremi.html', which includes a title, meta information, and a section with a heading and some text. To the right, the 'stylessobremi.css' file is open in the Editor tab, containing CSS code for styling the page. The bottom status bar shows the date as 15/11/2023 and the time as 16:13.

Tras terminar hacemos git add . Y git commit -m “” para dejar los cambios terminados en nuestro repositorio local.

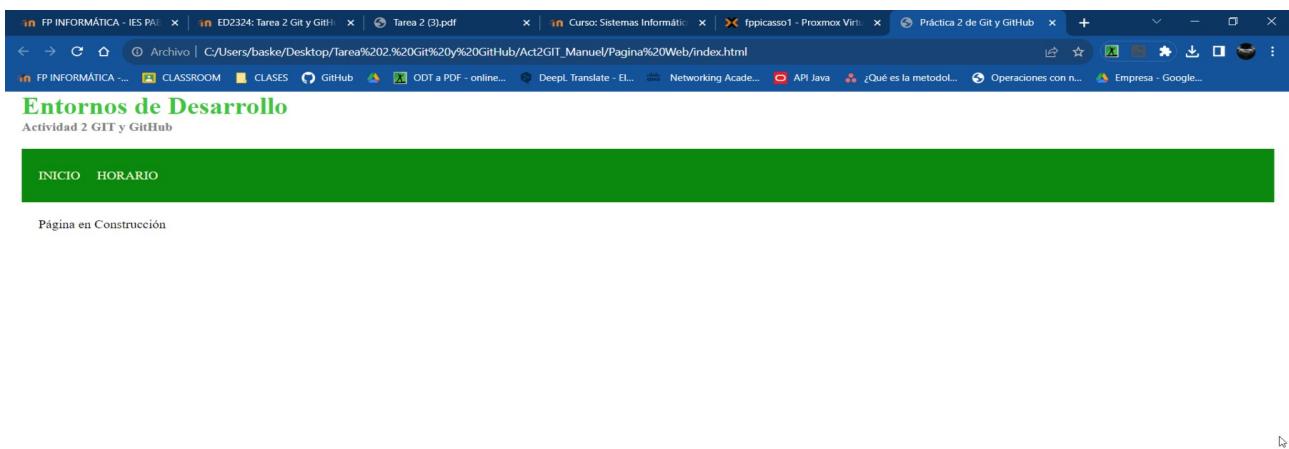
- Vuelve a la rama main y modifica el contenido de la Home con un mensaje indicando que la página está en construcción.

Con el comando checkout volvemos a nuestra rama main

```
MINGW64:/c/Users/baske/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (feature1)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (main)
$ |
```

Cambiamos la página principal y cambiamos el contenido.

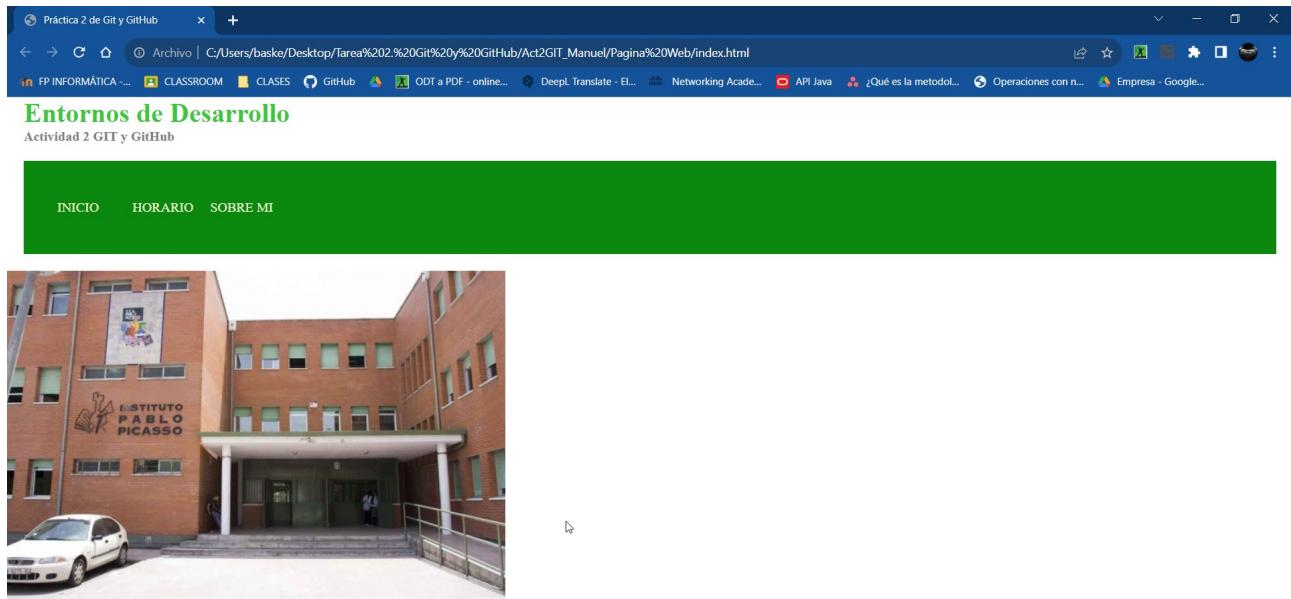


Aquí una pequeña Muestra del código

```
<html>
  <head>
    <title>Entornos de Desarrollo</title>
    <link href="styles.css" type="text/css" rel="stylesheet"/>
    <link href="stylehorario.css" type="text/css" rel="stylesheet"/>
    <link href="stylesobremi.css" type="text/css" rel="stylesheet"/>
  </head>
  <body>
    <header>
      <h1>Entornos de Desarrollo</h1>
      <h2>Actividad 2 GIT y GitHub</h2>
    </header>
    <nav>
      <div class="inicio">
        <a href="./index.html" target="_blank">INICIO</a>
      </div>
      <div class="horario">
        <a href="./horario.html" target="_blank">HORARIO</a>
      </div>
      <div class="mi">
        <a href="./sobremi.html" target="_blank">SOBRE MI</a>
      </div>
    </nav>
    <p>Página en Construcción</p>
    <footer>
      <div>
        <h3>Desarrollado por IES Pablo Picasso</h3>
      </div>
    </footer>
  </body>
</html>
```

- **Vuelve a la rama de desarrollo y Añade una imagen del instituto en el contenido de la página de Inicio**

Volvemos a la rama de desarrollo y desde ahí modificamos la página principal cambiando el texto por una imagen. Así ha quedado:



Y el código modificado:

The screenshot shows a Microsoft Edge browser window with the following details:

- File Edit Selection View Go Run Terminal Help**
- Pagina Web** in the address bar.
- EXPLORER** sidebar:
 - OPEN EDITORS**
 - PAGINA WEB** (selected)
 - images**
 - styles**
 - # styles.css
 - # styleshorario.css
 - # stylesobremi.css
 - horario.html**
 - index.html** (selected)
 - sobremi.html**
- OUTLINE** sidebar:
 - TIMELINE**
- Buscar** search bar at the bottom.
- index.html** content (copied from the image):

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Pagina Web</title>
        <link href="styles.css" rel="stylesheet">
        <link href="styleshorario.css" rel="stylesheet">
        <link href="stylesobremi.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h1 class="letras"> Entornos de Desarrollo </h1>
            <h4 class="letras2"> Actividad 2 GIT y GitHub </h4>
        </header>
        <nav class="barraNavegacion">
            <div class="inicio">
                <a href=".index.html" target="_blank"> INICIO </a>
            </div>
            <div class="horario">
                <a href=".horario.html" target="_blank"> HORARIO </a>
            </div>
            <div class="mi">
                <a href=".sobremi.html" target="_blank"> SOBRE MI </a>
            </div>
        </nav>
        
        <footer>
            <div class="pie">
                <h3> Desarrollado por IES Pablo Picasso </h3>
            </div>
        </footer>
    </body>
</html>
```

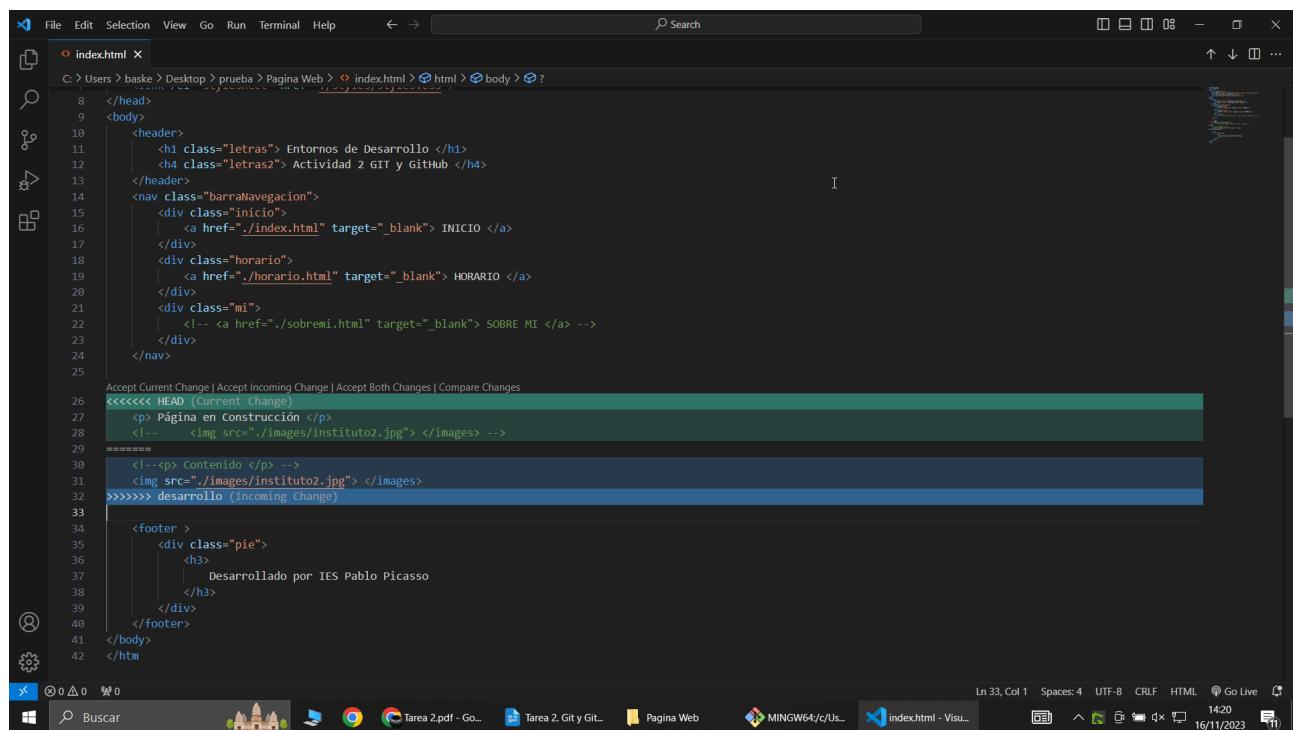
Bottom status bar: Ln 27, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚡ Port: 1646

- **Realiza un commit para guardar esos cambios Realiza una fusión (merge) entre la rama desarrollo y la rama feature1 y etiqueta el estado del proyecto como V2.0. Fusiona la rama desarrollo con la principal, donde deben aparecer todos los cambios que hemos realizado. Etiqueta el estado de la rama como V3.0**

Para hacer un merge de la rama de desarrollo y la rama feature1 nos situamos en desarrollo y escribimos:

git merge feature1. Una vez hecho subiremos los cambios al staging area y de ahí haremos un commit indicando que nuestro proyecto ha alcanzado la versión 2.0

Para unir las ramas main y desarrollo haremos exactamente lo mismo salvo que en este caso surgirá un conflicto ya que hemos tocado en un mismo sitio desde dos ramas y se nos abrirá Visual Studio Code donde solventaremos el conflicto eligiendo cómo queremos que se quede nuestra página web.



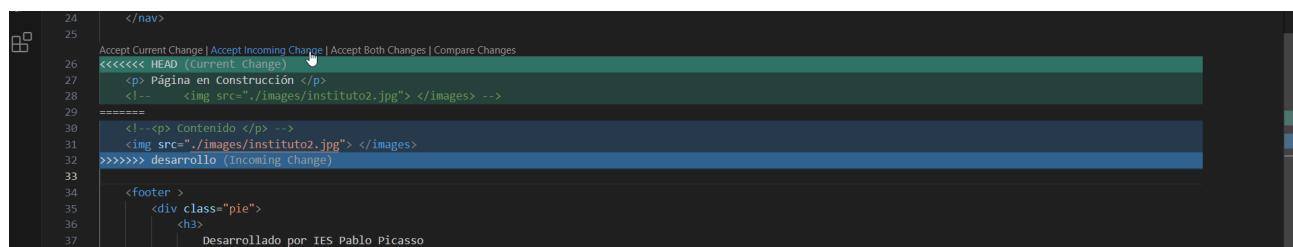
```

File Edit Selection View Go Run Terminal Help ⏎ ⏎ Search
C:\Users\baske\Desktop\prueba\Página Web>index.html>html>body>
8  </head>
9  <body>
10 <header>
11   |<h1 class="letras">Entornos de Desarrollo</h1>
12   |<h4 class="letras2">Actividad 2 GIT y GitHub</h4>
13 </header>
14 <nav class="barraNavegacion">
15   <div class="inicio">
16     |<a href=".index.html" target="_blank">INICIO</a>
17   </div>
18   <div class="horario">
19     |<a href=".horario.html" target="_blank">HORARIO</a>
20   </div>
21   <div class="mi">
22     |<!--<a href=".sobreMi.html" target="_blank">SOBRE MI</a>-->
23   </div>
24 </nav>
25
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<< HEAD (Current Change)
26 <p>Página en Construcción</p>
27 <!--</images>-->
28 =====
29 <!--<p>Contenido</p>-->
30 </images>
31 >>>>> desarrollo (Incoming Change)
32
33 <footer>
34   <div class="pie">
35     |<h3>
36       |Desarrollado por IES Pablo Picasso
37     </h3>
38   </div>
39 </footer>
40 </body>
41 </html>
42

```

Ln 33, Col 1 Spaces: 4 UTF-8 CRLF HTML ⏎ Go Live 14:20 16/11/2023

Vamos a quedarnos sólo con la foto y eliminaremos los dos comentarios anteriores. Para ello pinchamos sobre Accept Incoming Change.

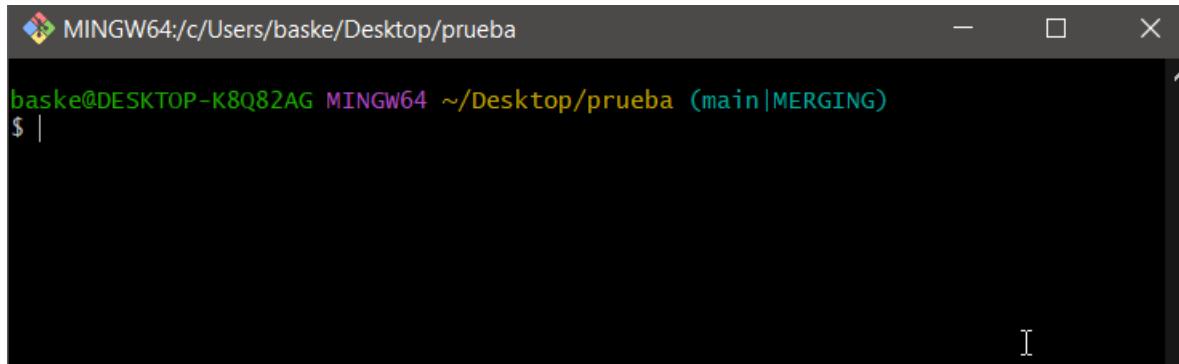


```

2d
25 </nav>
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<< HEAD (Current Change)
26 <p>Página en Construcción</p>
27 <!--</images>-->
28 =====
29 <!--<p>Contenido</p>-->
30 </images>
31 >>>>> desarrollo (Incoming Change)
32
33 <footer>
34   <div class="pie">
35     |<h3>
36       |Desarrollado por IES Pablo Picasso
37     </h3>

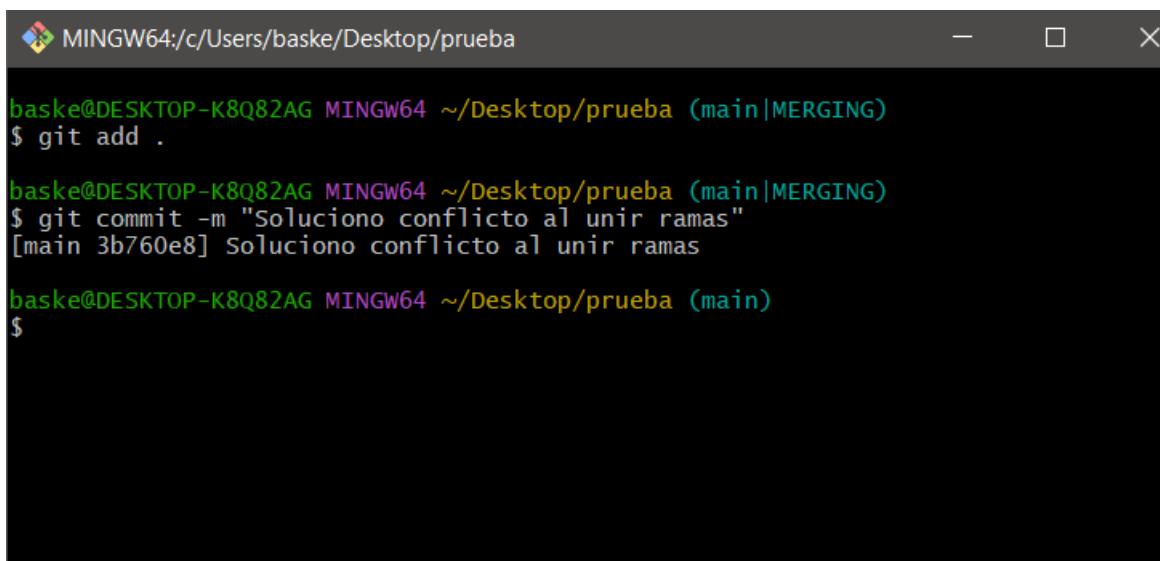
```

Una vez solventado el conflicto con Visual Studio Code volvemos a Git y vemos que junto a la palabra main aparece MERGING.



```
MINGW64:/c/Users/baske/Desktop/prueba
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/prueba (main|MERGING)
$ |
```

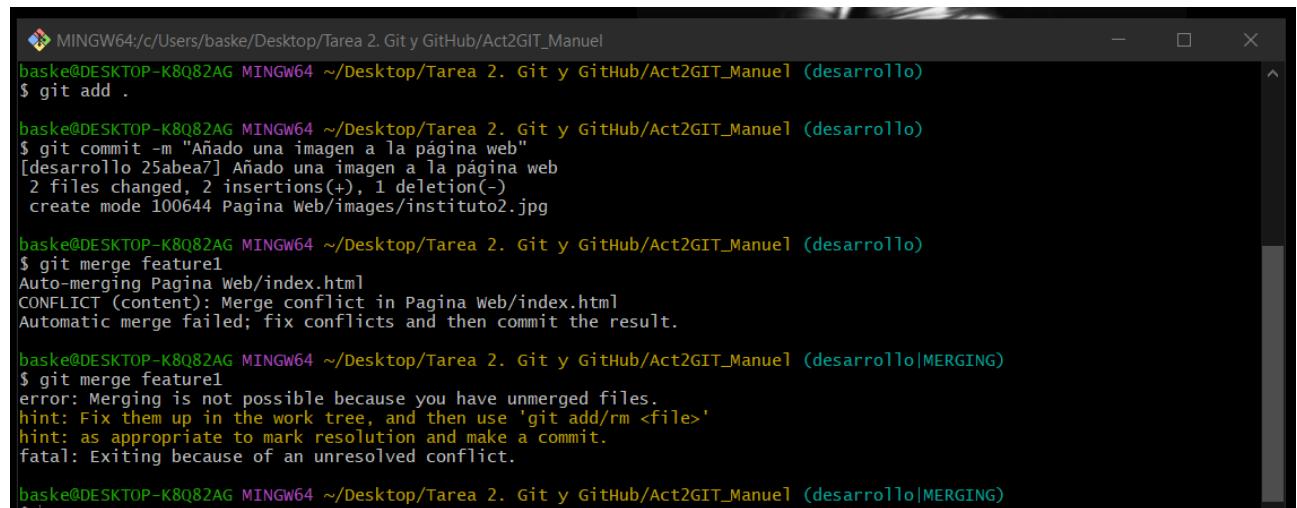
Para terminar de unir las ramas debemos hacer un commit.



```
MINGW64:/c/Users/baske/Desktop/prueba
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/prueba (main|MERGING)
$ git add .

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/prueba (main|MERGING)
$ git commit -m "Soluciono conflicto al unir ramas"
[main 3b760e8] Soluciono conflicto al unir ramas

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/prueba (main)
$
```



```
MINGW64:/c/Users/baske/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (desarrollo)
$ git add .

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (desarrollo)
$ git commit -m "Añado una imagen a la página web"
[desarrollo 25abea7] Añado una imagen a la página web
2 files changed, 2 insertions(+), 1 deletion(-)
create mode 100644 Pagina Web/images/instituto2.jpg

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (desarrollo)
$ git merge feature1
Auto-merging Pagina Web/index.html
CONFLICT (content): Merge conflict in Pagina Web/index.html.
Automatic merge failed; fix conflicts and then commit the result.

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (desarrollo|MERGING)
$ git merge feature1
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'.
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (desarrollo|MERGING)
```

Una vez resuelto el conflicto ya las tres ramas quedan unidas y guardo todos los cambios añadiendo una nueva etiqueta que se llamará V3.0.

Voy a mostrar un pequeño resumen de cómo han quedado mis commits y mis tags:

The screenshot shows a Windows desktop environment. In the top left, there's a File Explorer window titled 'Act2GIT_Manuel' showing files like '.git', 'Pagina Web', '.gitignore', and 'README.md'. To its right is another File Explorer window showing a folder structure. Below these is a terminal window with the following command history:

```
baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (main)
$ git log --oneline
56fc27a (HEAD -> main, origin/main, origin/HEAD) Nuevos añadidos a la página web
63a679b (feature) Modifico el fichero README.md
193b1a8 (tag: v3.0) Combino mis tres ramas
096de09 (tag: v2.0, origin/desarrollo, desarollo) Resuelvo conflicto
25abea7 Añado una imagen a la página web
87e9748 Modifico contenido de la página
ffcc542a (origin/feature1, feature) Anado página web con contenido
8496121 Anada una nueva rama con otra página web
542325d Anada una rama con el horario
073a71d (tag: v1.0) Comienzo del Proyecto
6871dfc Initial commit

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (main)
$ git tag
V1.0
V2.0
V3.0

baske@DESKTOP-K8Q82AG MINGW64 ~/Desktop/Tarea 2. Git y GitHub/Act2GIT_Manuel (main)
$ ~
```

Y aquí el GitHub:

The screenshot shows a GitHub repository page for 'Act2GIT_Manual'. The repository is public and has 3 branches and 3 tags. The main branch is not protected. The repository description is: 'Repositorio de Simulacro de Examen de Git'. It contains several commits from 'Loltron81' and a README.md file. The README.md file content includes:

```
#Actividad 2 sobre el manejo de Git y GitHub

Repository de Simulacro de Examen de Git

He creado este Repositorio para almacenar una práctica de Git donde creamos una pequeña página web sobre el instituto Pablo Picasso.

De momento los únicos cambios han sido:

• Clonación del repositorio del profesor con toda la información del proyecto
• Creación de nuestro propio repositorio tanto local como en GitHub.
• Modificación del fichero .gitignore para que no acepte pdf

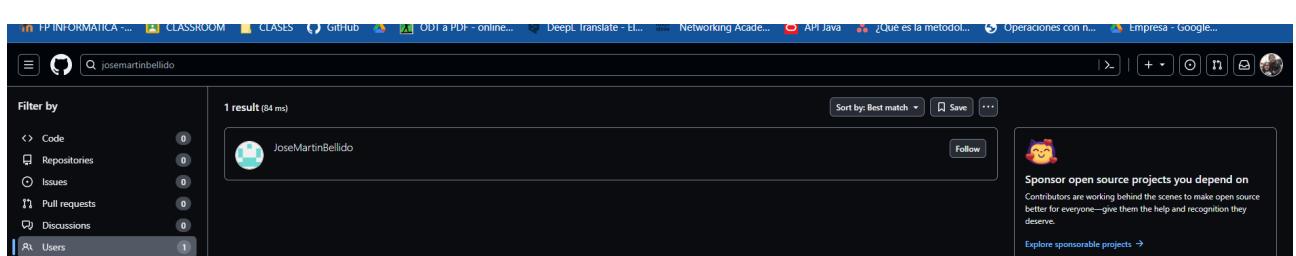
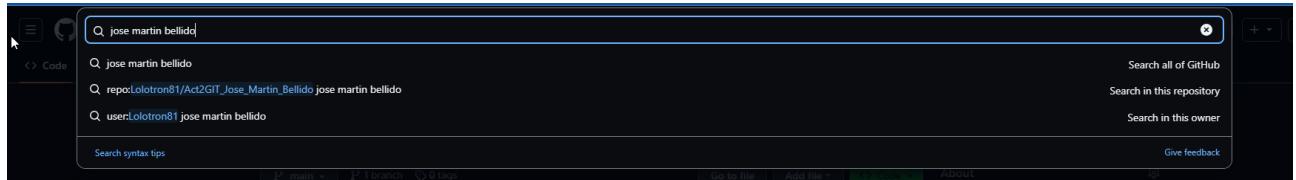
He continuado haciendo modificaciones a la página

He creado tres ramas para ir haciendo modificaciones a las tres secciones de la página a la vez y tras terminar las tareas he usado el comando merge para combinar todas las ramas.
```

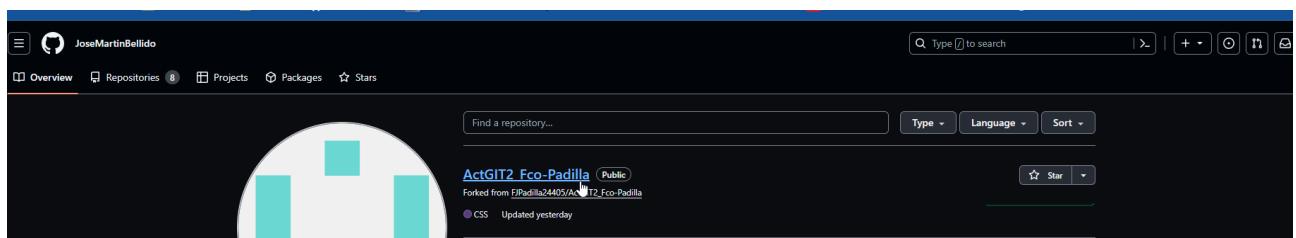
The repository also shows activity, releases, packages, languages (CSS 51.3%, HTML 48.7%), and suggested workflows.

Ahora vamos a proceder a usar la herramienta Fork de GitHub para copiarnos un repositorio de otra persona. Esta copia queda desligada del proyecto original y por tanto podemos hacerle todas las modificaciones que queramos y además por medio de la instrucción Pull Request podemos mandarle los cambios que hemos hecho y si le gustan los introducirá en su proyecto.

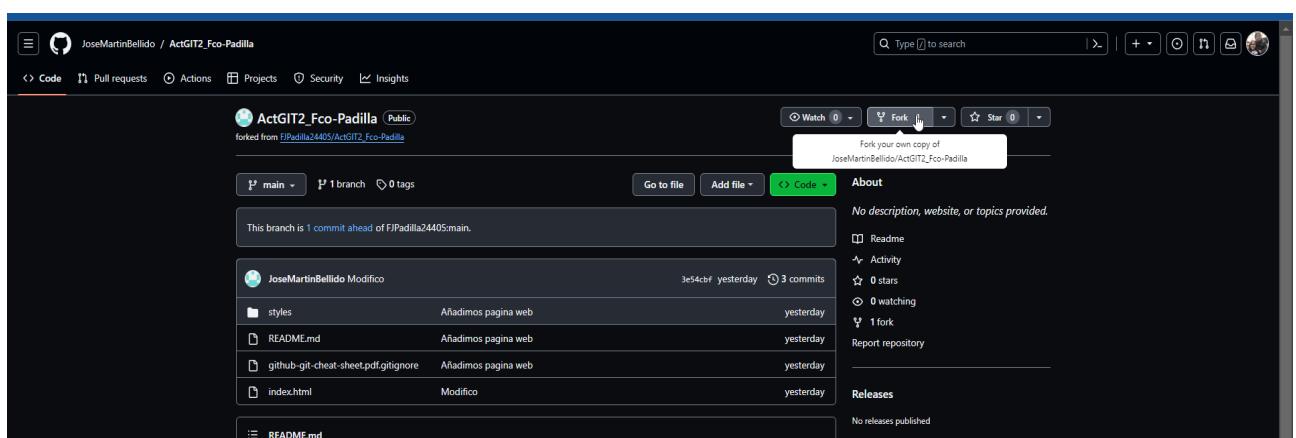
Vamos a buscar al compañero de clase Jose Martín Bellido.



Encontramos el repositorio que nos interesa y entramos.



Debemos ir a la pestaña superior y pinchamos sobre Fork.



Aquí vemos la página para llenar algunos campos y solicitar el fork. Damos una pequeña descripción y por último pinchamos sobre el botón verde Create Fork.

Create a new fork

A **fork** is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

Lolotron81 / ActGIT2_Fco-Padilla
ActGIT2_Fco-Padilla is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)
Modifico la página web de mi Compañero Jose

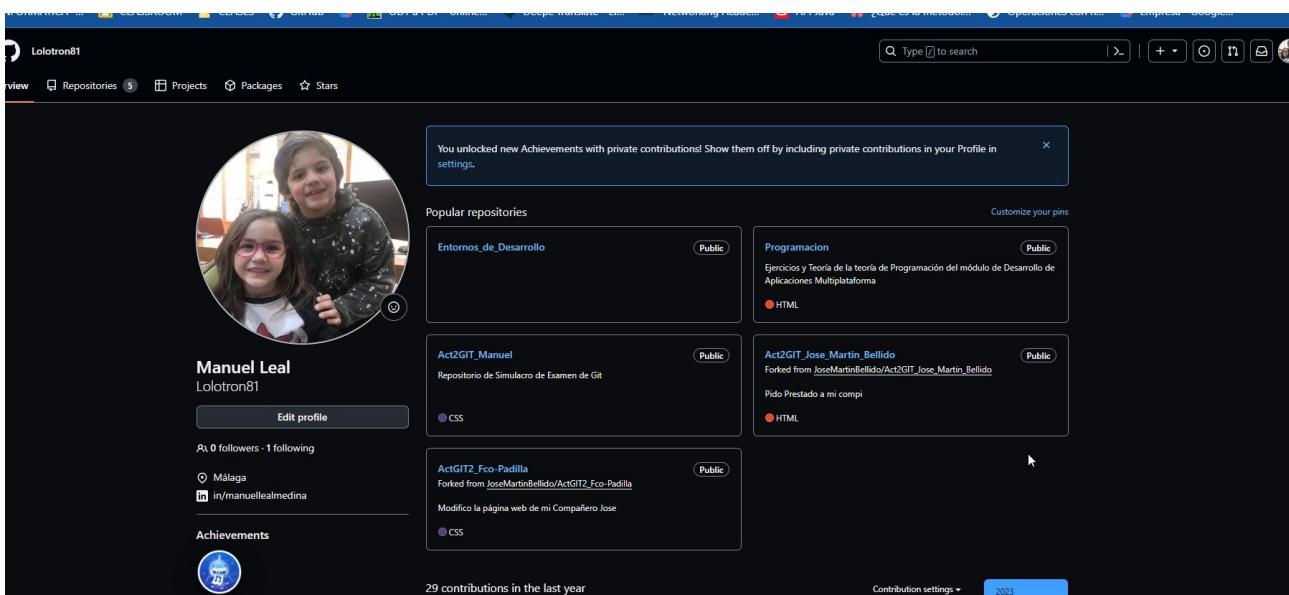
Copy the `main` branch only
Contribute back to [JoseMartinBellido/ActGIT2_Fco-Padilla](#) by adding your own branch. [Learn more.](#)

① You are creating a fork in your personal account.

Create fork



Cuando volvemos a nuestro GitHub vemos que ya aparece.



Accedo al código de su página web con Visual Studio Code y le hago un par de modificaciones a su página principal.

Así lucía su página:

The screenshot shows a browser window with the address bar containing 'C:/Users/baske/Documents/ENTORNOS%20DE%20DESARROLLO/Unidad%2004.%20Optimización,%20Documentación%20y%20Factorización/Tarea%202.%20Git%2...'. The page title is 'Entornos de desarrollo' and the subtitle is 'Actividad 2. Git y Github'. A blue header bar includes 'INICIO', 'HORARIO', and 'SOBRE MÍ'. Below the header, a section titled 'Contenido' states 'La página está en construcción' and features a photograph of a brick building with a car parked in front. The status bar at the bottom right shows the date as 16/11/2023.

The screenshot shows a browser window with the same address bar and title. The main content area now features a large blue banner at the top with the text '© Desarrollado por IES Pablo Picasso'. Below the banner, the 'Actividad 2. Git y Github' section and the 'Contenido' section with the building photo remain. The status bar at the bottom right shows the date as 16/11/2023.

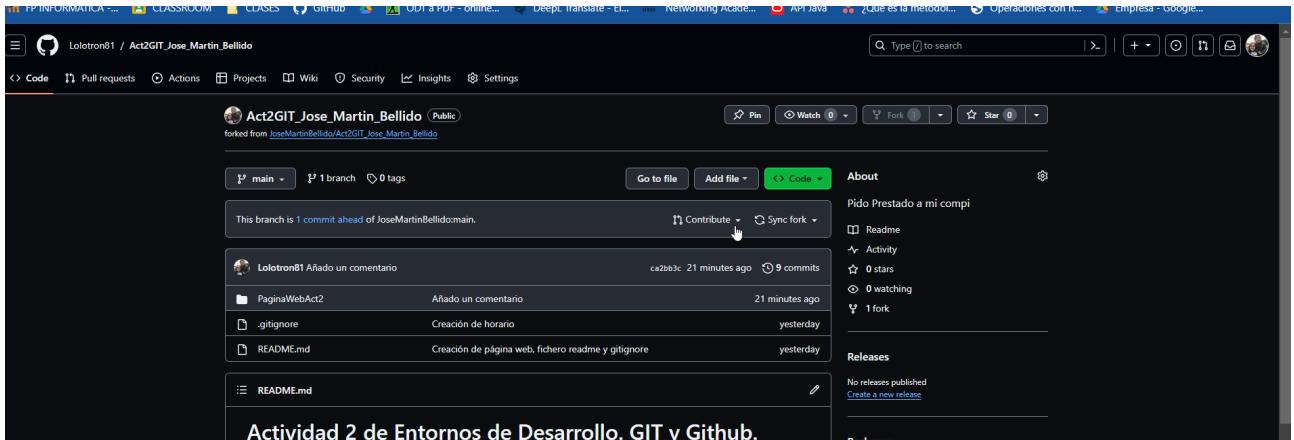
Y así luce ahora:

The screenshot shows a browser window with the same address bar and title. The main content area now features a large blue banner at the top with the text '© Desarrollado por IES Pablo Picasso'. Below the banner, the 'Actividad 2. Git y Github' section and the 'Contenido' section with the building photo remain. A new paragraph of text has been added below the 'Contenido' section: 'En el dibujo de la práctica no aparecen estas dos líneas. Be al Carrefour'. The status bar at the bottom right shows the date as 16/11/2023.

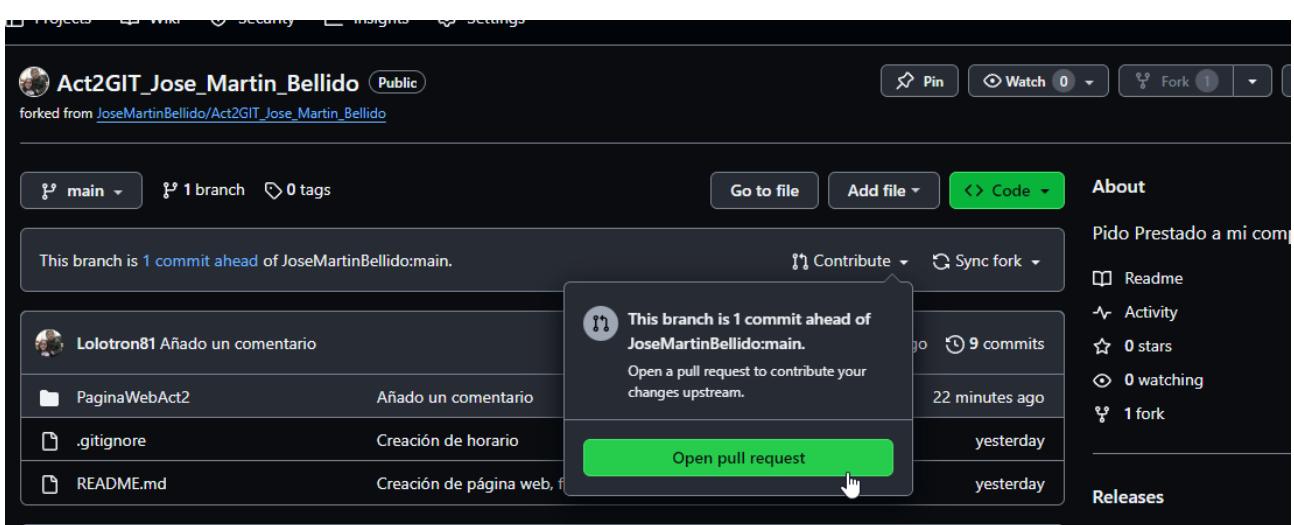
The screenshot shows a browser window with the same address bar and title. The main content area now features a large blue banner at the top with the text '© Desarrollado por IES Pablo Picasso'. Below the banner, the 'Actividad 2. Git y Github' section and the 'Contenido' section with the building photo remain. A new paragraph of text has been added below the 'Contenido' section: 'El IES Pablo Picasso de Pinto comenzó a impartir clases en el curso 92-93. En aquel momento el conjunto de edificios que hoy albergan nuestro instituto todavía no se había terminado de construir; por lo que durante aquel primer año las clases se impartieron en el IES Vicente Aleixandre en horario vespertino, con un Jefe de Estudios que también ejercía las funciones de Director y que posteriormente sería el Director durante el siguiente curso escolar. El Instituto comenzó impartiendo únicamente primero de REM (Reforma de Enseñanzas Medias) a un total de diez grupos con un Claustro de profesores muy reducido de los cuales todavía algunos permanecen en el Centro.' The status bar at the bottom right shows the date as 16/11/2023.

Una vez hecha la modificación toca guardar los cambios en mi repositorio prestado y hacer un commit para que quede constancia del cambio.

Una vez hecho esto volvemos a GitHub y a nuestro repositorio prestado y deberemos pulsar sobre el botón que dice Contribute.

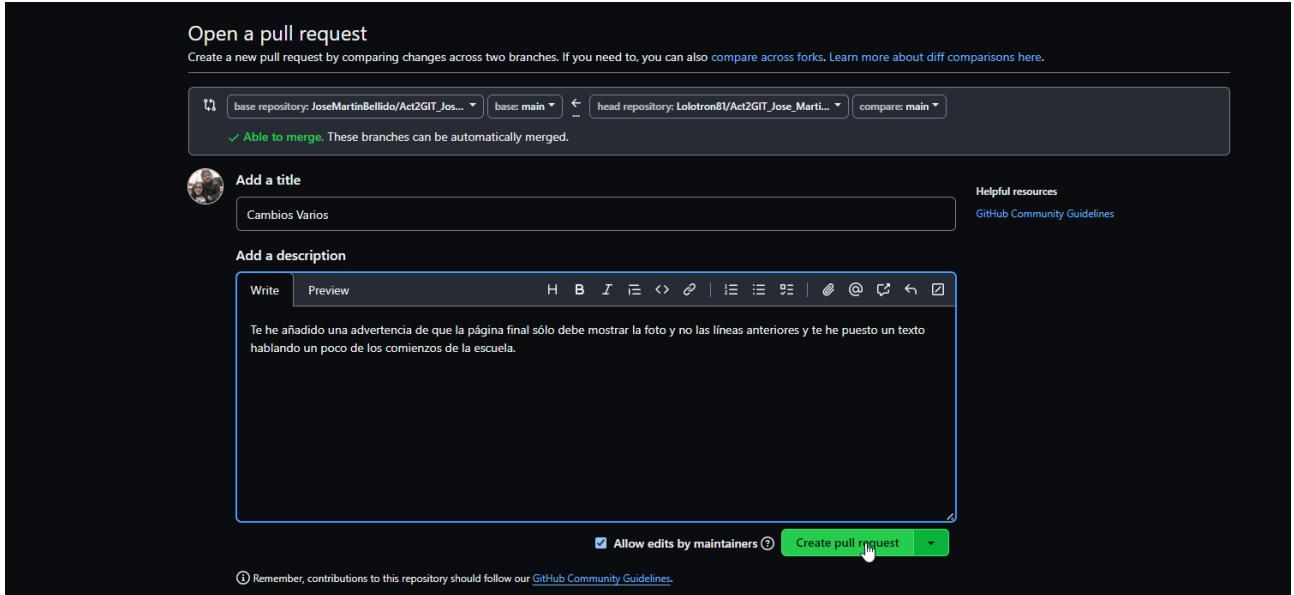


Se nos abre esta pestaña:

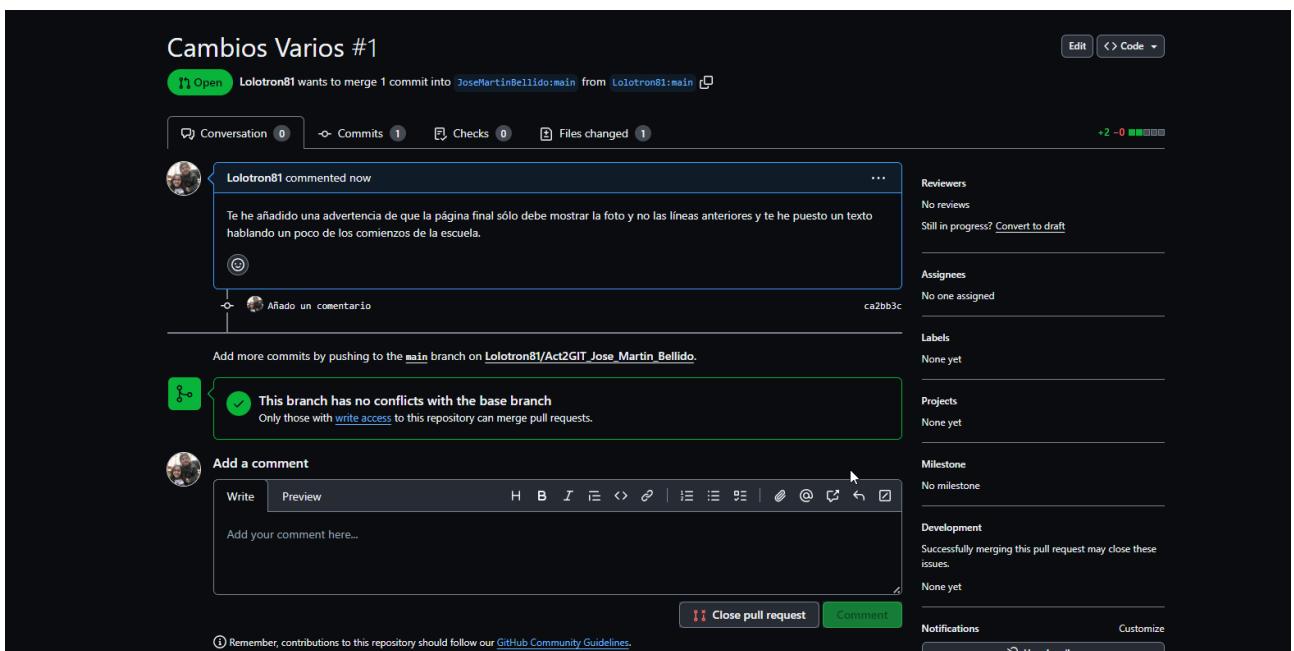


Y pinchamos sobre Open Pull Request.

Se nos abrirá una nueva ventana donde podremos poner un título a nuestro request y una descripción detallando los cambios que hemos realizado. Para finalizar pincharemos sobre el botón verde que indica Create Pull Request.



Nos saldrá una nueva ventana indicándonos que todo ha salido bien.



Pinchamos sobre Close Pull Request y ya tocará esperar a que el compañero actúe en consecuencia.