

TAREA UNIDAD 5

Documentación y control de versiones



CFGs DAW. Módulo: Despliegue de Aplicaciones Web (2023/24)
Dayro Morales Cruz

INDICE

Ejercicio 1.....3

Ejercicio 2.....9

Este tipo de actividades se realizarán en tu servidor Linux ya sea Ubuntu o Debian y Windows, habrá dos tipos de actividades, una relacionada con **phpDocumentor** y otra con **GitHub**.

Ejercicio 1

Este ejercicio relacionado con generación de documentación consistirá en los siguientes apartados :

- **Actividad 1.1**

Instala la herramienta phpdocumentor en tu servidor Linux y comenta los aspectos más importantes de tal herramienta, así como las etiquetas principales que se usan.

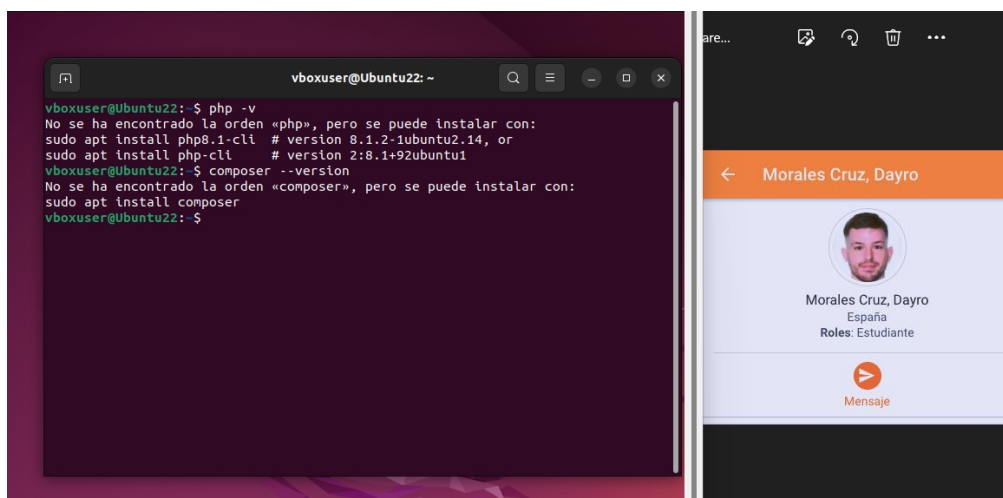
PHPDocumentor es una herramienta de documentación para PHP. Permite generar documentación de código fuente de PHP en varios formatos, incluyendo HTML, PDF y etc. Algunas de las etiquetas más utilizadas en PHPDocumentor son:

- **@param:** Se utiliza para documentar los parámetros de una función o método.
- **@return:** Se utiliza para documentar el valor de retorno de una función o método.
- **@var:** Se utiliza para documentar la variable de una clase.
- **@throws:** Se utiliza para indicar las excepciones que puede lanzar un método o función.

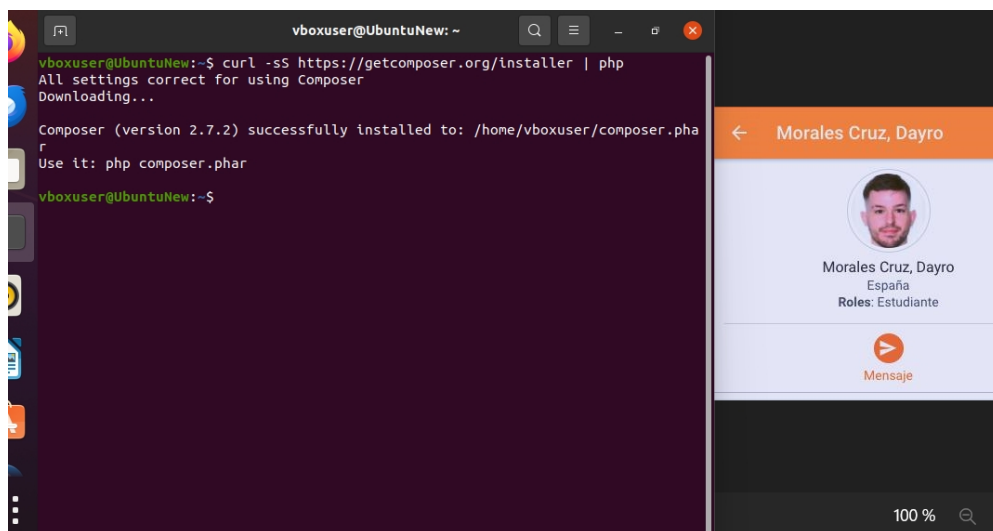
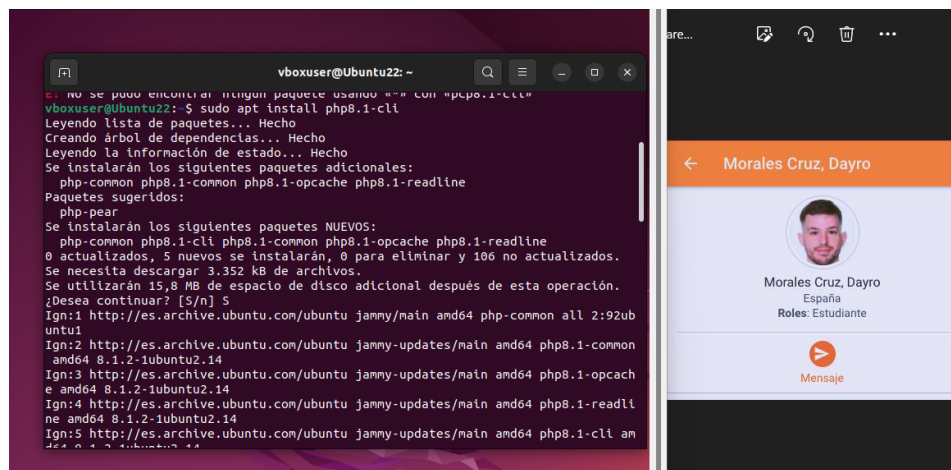
Primero instalaremos nuestro pdpdocumentor, pero antes comprobaremos si tenemos instalado php y composer:

php -v

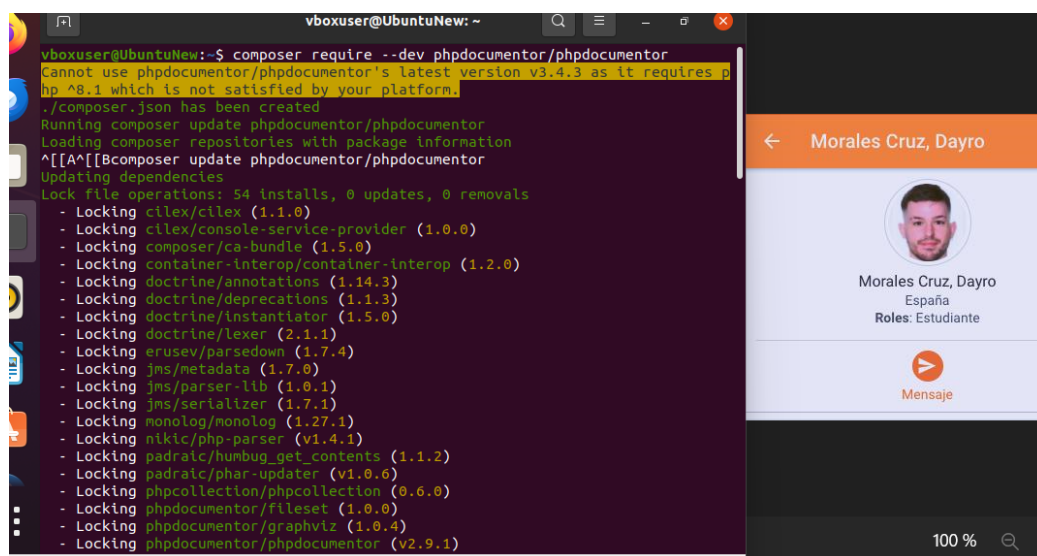
composer --version

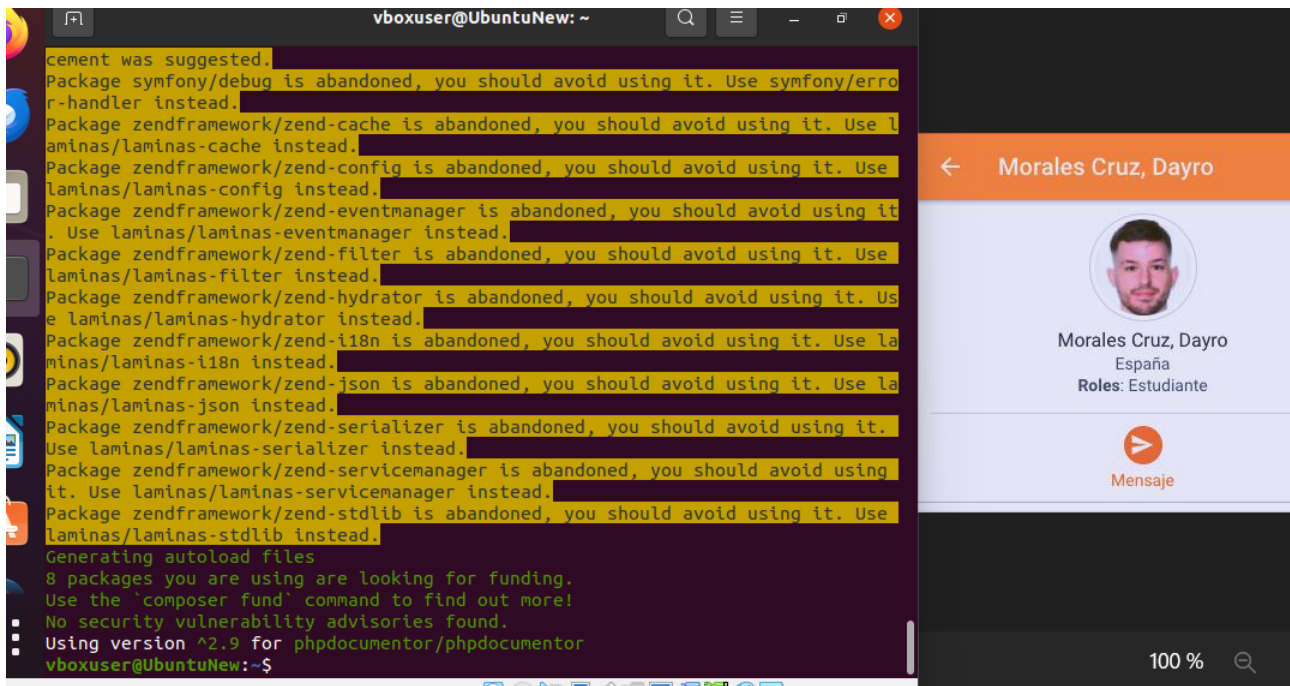


En nuestro caso debemos instalar ambos. Procedemos a instalarlos



Una vez instalado procedemos a instalar **phpdocumentor**





• Actividad 1.2

En esta actividad debes crear en tu servidor un script PHP con el nombre practica-XXXXXX.php, donde XXXXXX será tu apellido. A continuación escribe dentro de este script bloques de código y DocBlocks para que luego se pueda generar la documentación correspondiente. El script debe contener al menos dos funciones documentadas, indicando mediante las etiquetas vistas en la unidad los siguientes elementos:

- Parámetros de entrada de la función.
- Parámetros de devuelve la función.
- Autor y versión del script.
- Una anotación que solo sea visible en la documentación para desarrolladores.

Nota: Hay que adjuntar el fichero practica-XXXXXX.php que realices en el directorio .zip.

Mediante un editor de textos, escribimos un script simple en php

```
<?php
/**
 * Este es un script de práctica para PHPDocumentor.
 *
 * @author Tu Nombre
 * @version 1.0
 */

/**
```

```

* Esta función suma dos números.
*
* @param int $a Primer número a sumar.
* @param int $b Segundo número a sumar.
* @return int La suma de los dos números.
*/
function suma($a, $b) {
    return $a + $b;
}

/**
* Esta función resta dos números.
*
* @param int $a El minuendo.
* @param int $b El sustraendo.
* @return int La resta de los dos números.
* @internal Solo para uso interno.
*/
function resta($a, $b) {
    return $a - $b;
}
?>

```

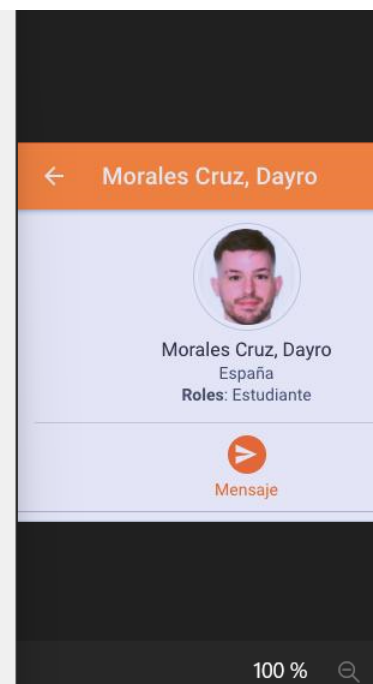
The screenshot shows a code editor window titled 'practica-Morales.php' with a file icon, a dropdown menu, and a 'Guardar' (Save) button. The code is as follows:

```

1 ?php
2 /**
3  * Esto es un script para PHPDocumentor
4  *
5  * @author Dayro Morales Cruz
6  * @version 1.0
7  */
8
9  /**
10 * Esta función suma 2 números
11 *
12 * @param int $a primer número a sumar
13 * @param int $b segundo numero a sumar
14 * @return int La suma de ambos números
15 */
16 function suma($a, $b) {
17     return $a + $b;
18 }
19
20 /**
21 * Esta funcion restará dos números
22 *
23 * @param int $a será el minuendo
24 * @param int $b será el sustraendo
25 * @return int La resta entre ambos números
26 * @internal solo para uso interno
27 */
28 function resta($a, $b) {

```

The editor's status bar at the bottom shows 'PHP', 'Anchura del tabulador: 8', 'Ln 1, Col 1', and 'INS'.



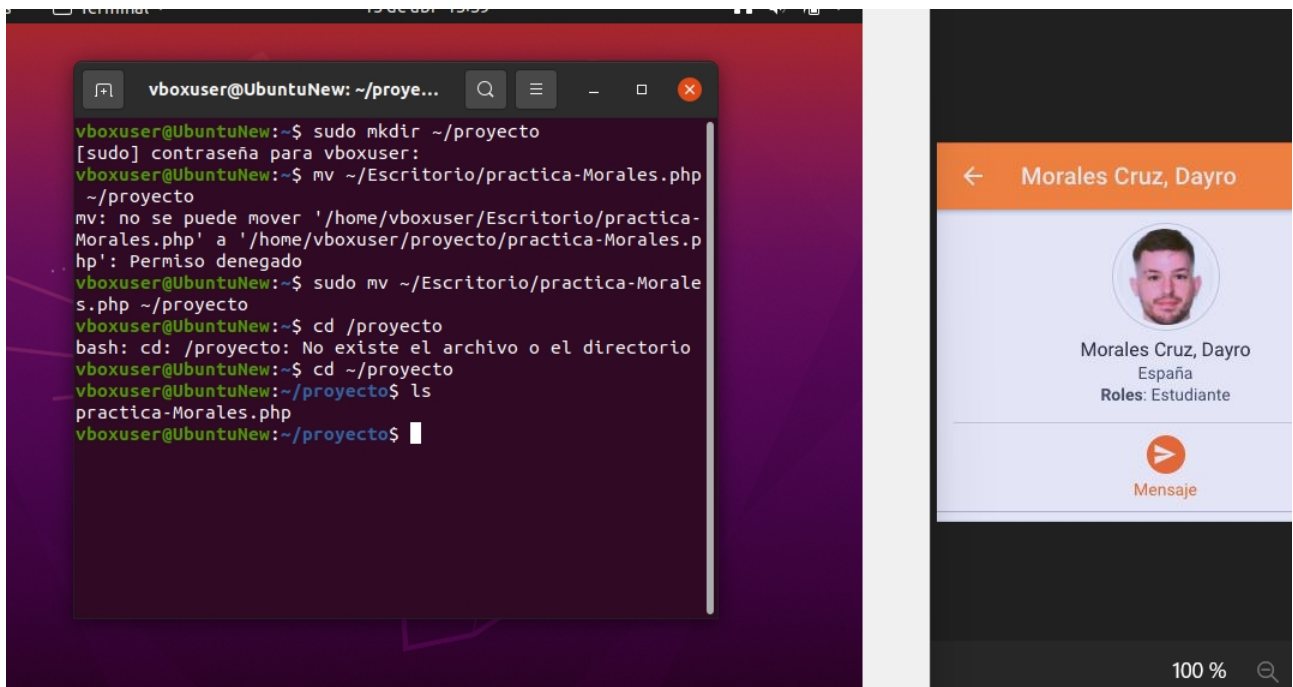
Lo guardamos y renombramos con el nombre que nos requiere la tarea **practica-Morales.php**

Crearemos una carpeta llamada **proyecto** donde guardaremos nuestro proyecto para luego poder ejecutar phpdocumentor desde allí

sudo mkdir ~/proyecto

sudo mv ~/Escritorio/practica-morales-php ~/proyecto

usaremos **cd ~/proyecto** para ingresar en la carpeta y con **ls** comprobamos que se movió correctamente

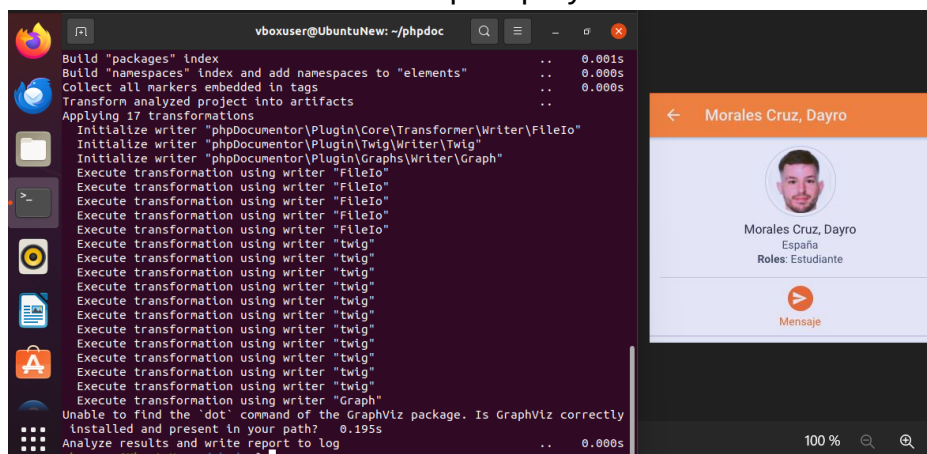


Ahora ejecutaremos nuestro documento con phpdocumentor.

Para ejecutar el archivo con el programa reciente usaremos el archivo phpdoc ubicado en la siguiente ruta

/home/vboxuser/vendor/bin/phpdoc run -d /home/vboxuser/proyecto -t /home/vboxuser/proyecto

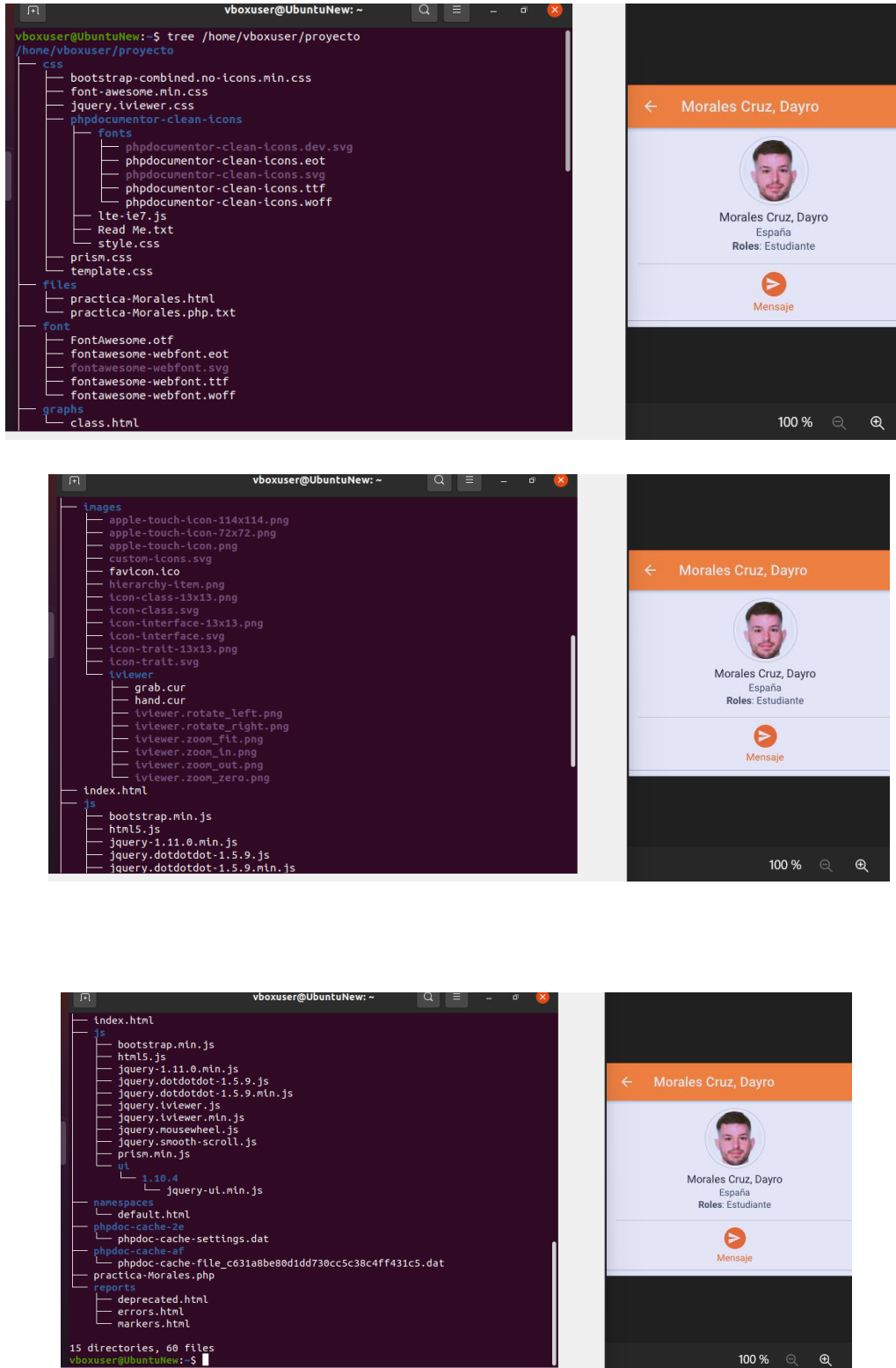
Eso hará el cree el documento en la carpeta proyecto



Actividad 1.3

Después de revisar la documentación, especialmente el apartado 1.2, genera la documentación del script PHP que hemos creado en la actividad 1.2. A continuación muestra el árbol de carpetas y archivos que genera como salida.

Ahora mostramos el árbol de carpetas y archivos que generó como salida con el comando **tree**



Ejercicio 2

Este ejercicio está relacionado con **Github**, se puede realizar en Windows, que consistirá en los siguientes apartados :

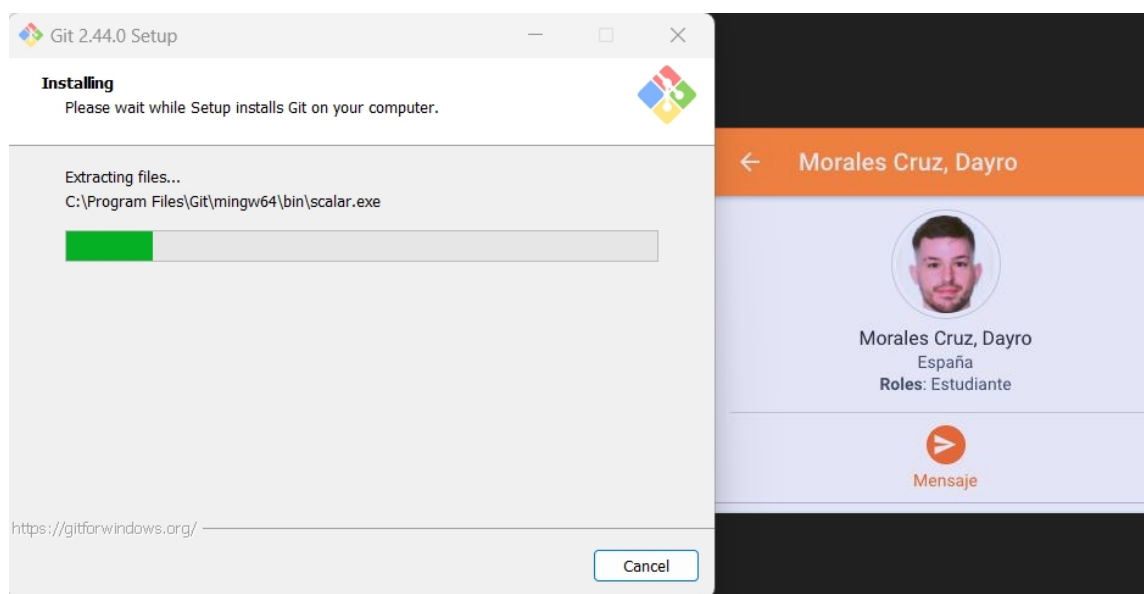
- **Actividad 2.1**

1. Elabora un pequeño tutorial donde será necesario instalar git en Windows y configurarlo en tu computadora, donde se detallarán las explicaciones teóricas (¿Qué es Git? ¿Para qué sirve? ¿Última versión? etc..)y las capturas de pantallas que sean necesarias.

Primero elaboraremos un pequeño tutorial y responderemos a unas preguntas básicas:

1. **¿Qué es Git?** Git es un sistema de control de versiones distribuido, lo que significa que cada desarrollador tiene una copia completa del historial de cambios del proyecto en su computadora local. Esto permite a los desarrolladores trabajar de manera eficiente incluso cuando están desconectados de la red principal del proyecto.
2. **¿Para qué sirve?** Git se utiliza para rastrear los cambios en el código fuente de un proyecto a lo largo del tiempo. Permite a los desarrolladores colaborar en un proyecto sin pisarse los pies, ya que pueden trabajar en diferentes partes del código simultáneamente y luego combinar sus cambios de manera segura.
3. **¿Última versión?** La última versión de Git puede variar dependiendo de cuándo leas esto, pero puedes verificar la última versión en la página oficial de Git.
4. **Instalación de Git en Windows:** Puedes descargar el instalador de Git para Windows desde la página oficial de Git. Una vez descargado, simplemente ejecuta el instalador y sigue las instrucciones en pantalla.

Primero descargaremos el programa desde la [página oficial de Git](https://gitforwindows.org/) e iniciamos el proceso de instalación. Avanzamos marcando las opciones que nos recomiendan.



- **Actividad 2.2**

Realiza la siguiente configuración:

- Configura tu nombre, apellidos y cuenta de correo
- Muestra la versión instalada.
- Indica cuál es tu directorio de trabajo.

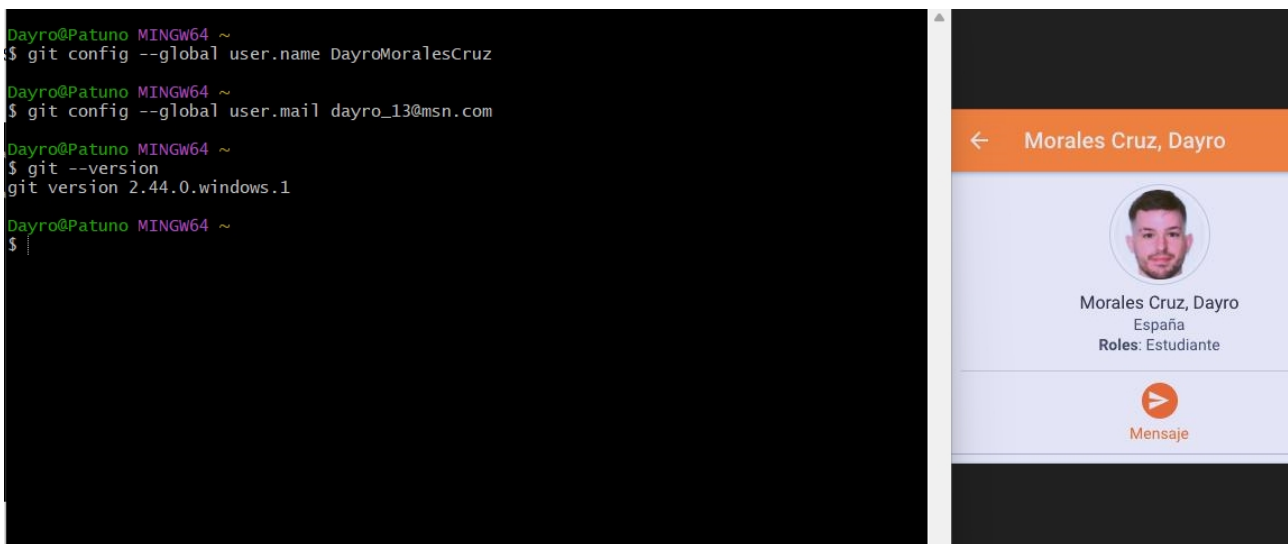
Una vez instalamos abrimos nuestro programa **Git Bash** ingresamos el comando **git config --global user.name Dayro**

Configuramos nuestra cuenta de correo

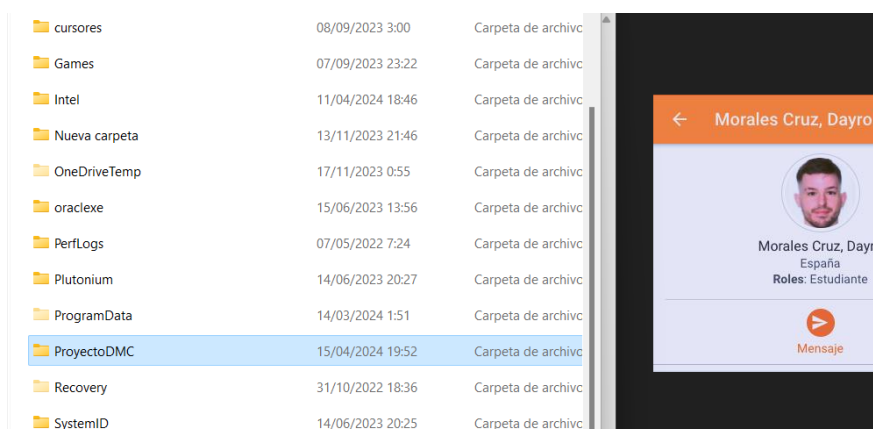
git config --global user.mail dayro_13@msn.com

Mostramos la versión instalada de **Git**

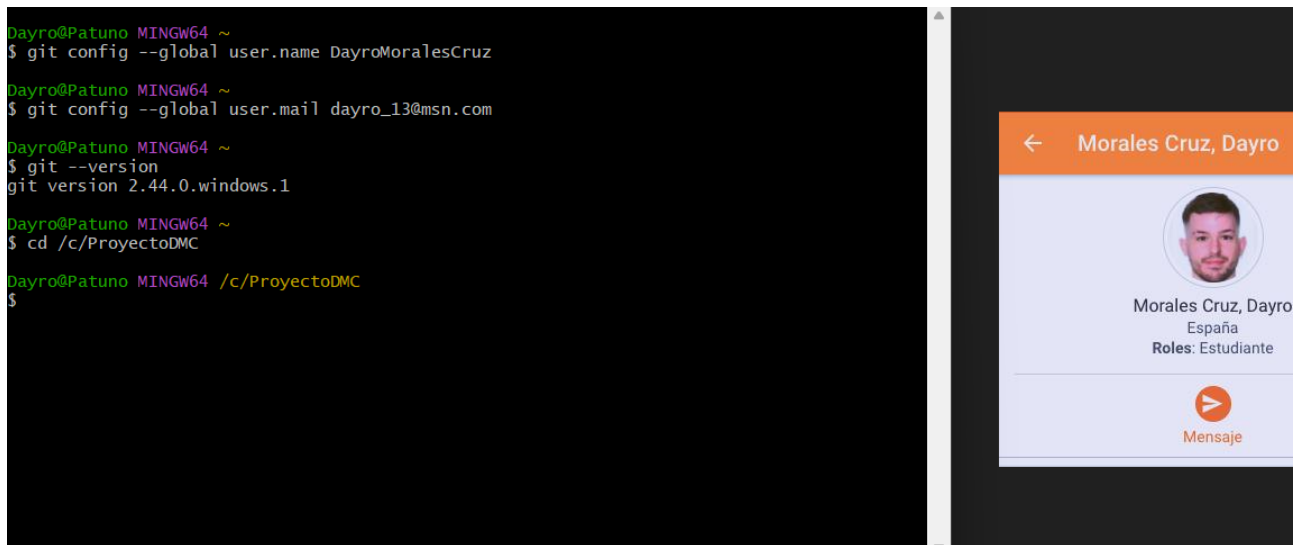
git --version



A continuación indicamos cual es nuestro directorio de trabajo navegando hasta el. En nuestro caso será la carpeta proyectoDMC ubicada en la unidad de disco :C



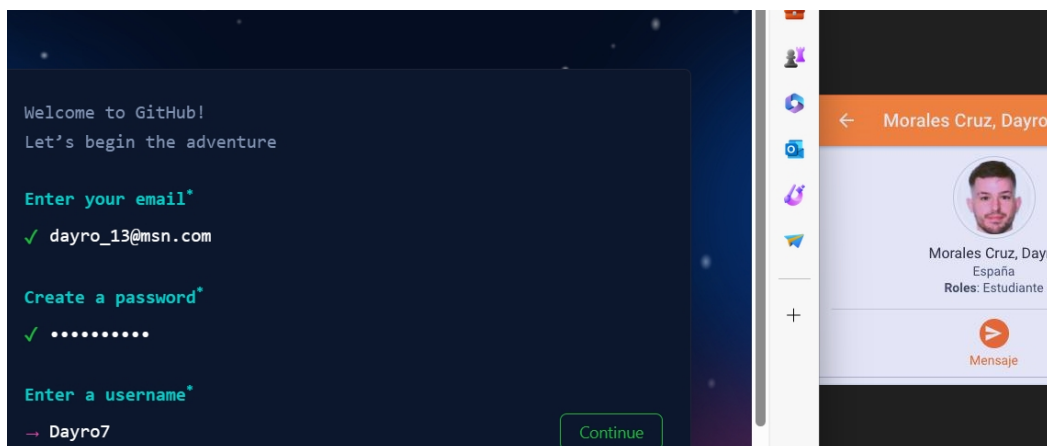
Por último mostramos la ruta a nuestra carpeta de trabajo



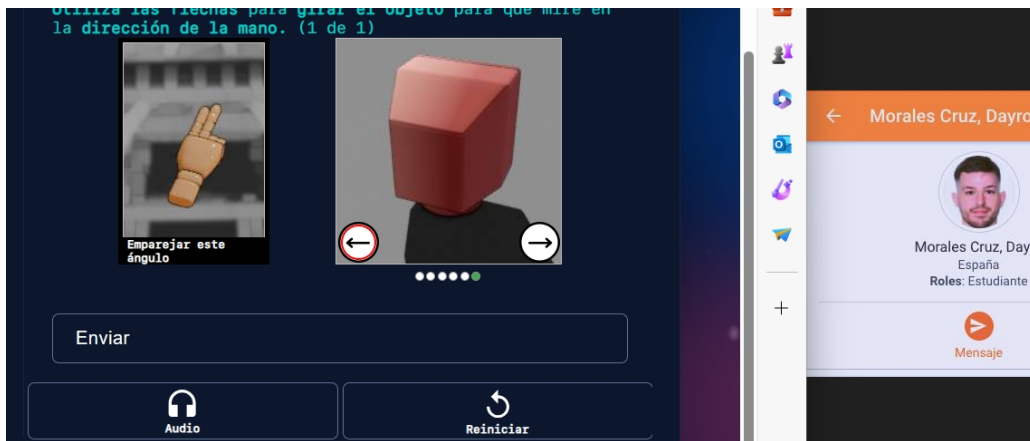
- **Actividad 2.3**

En este apartado que se escogerá el fichero que hemos realizado en el apartado 1.2 y demostrar como funciona git en tal proyecto, por ejemplo subiendo a git todo el proyecto completo y posteriormente es necesario modificar algo para que se detecten los cambios. Es necesario realizarlo con NetBeans. Hay que darse de alta en la url <https://github.com/> y crear un repositorio llamado **distanciadaw2324** para llevar los ficheros del proyecto php. Demostrarlo con capturas de pantalla.

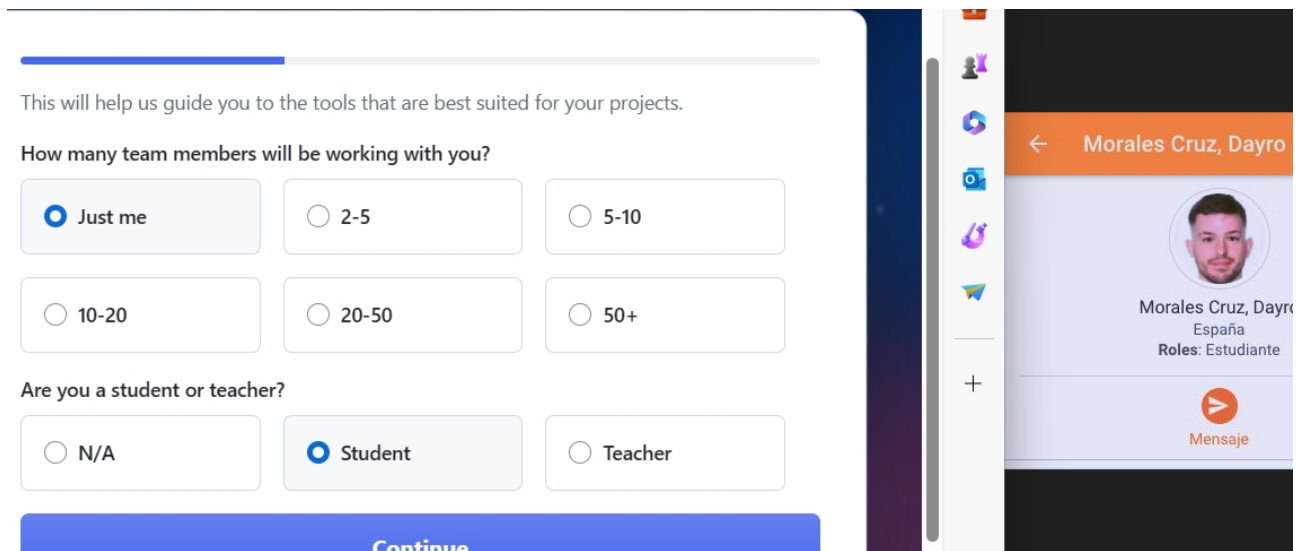
Primero nos dirigimos a la pagina oficial de [github](https://github.com/) y nos damos de alta



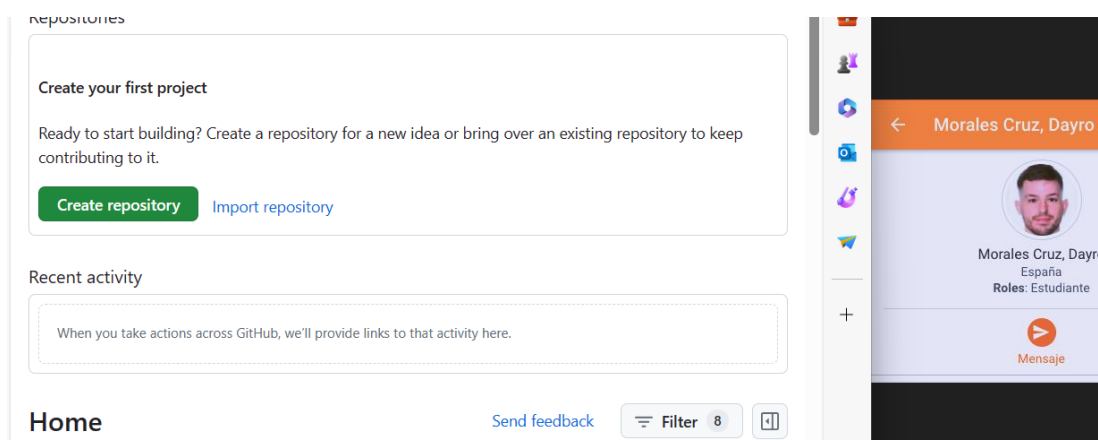
Verificamos la cuenta mediante catcha



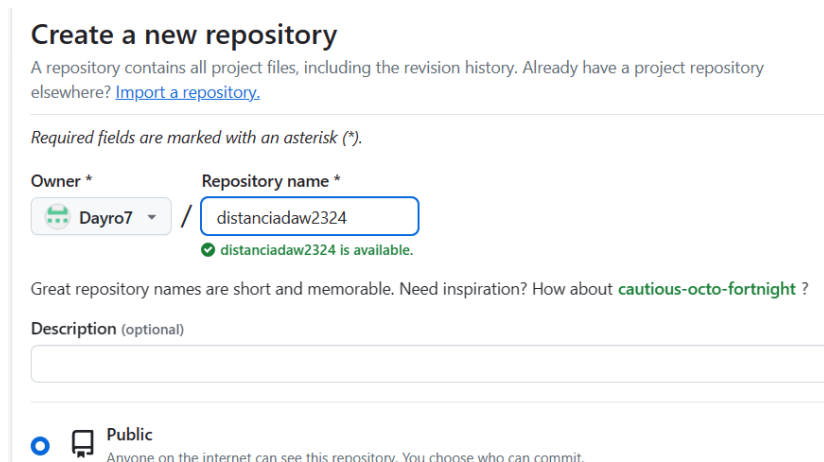
A continuación seleccionamos que somos nosotros quien usaremos la cuenta y que somos estudiantes.



Ahora crearemos un repositorio dándole a **Create repository**



Y lo llamamos como requiere la tarea ***distanciadaw2324***



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 *

Dayro7 / distanciadaw2324

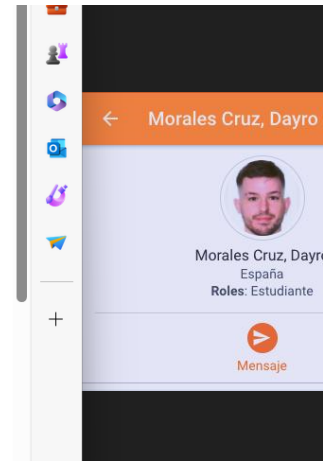
✓ distanciadaw2324 is available.

Great repository names are short and memorable. Need inspiration? How about **cautious-octo-fortnight** ?

Description (optional)

Public

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



Ahora ingresamos los comandos de manera global

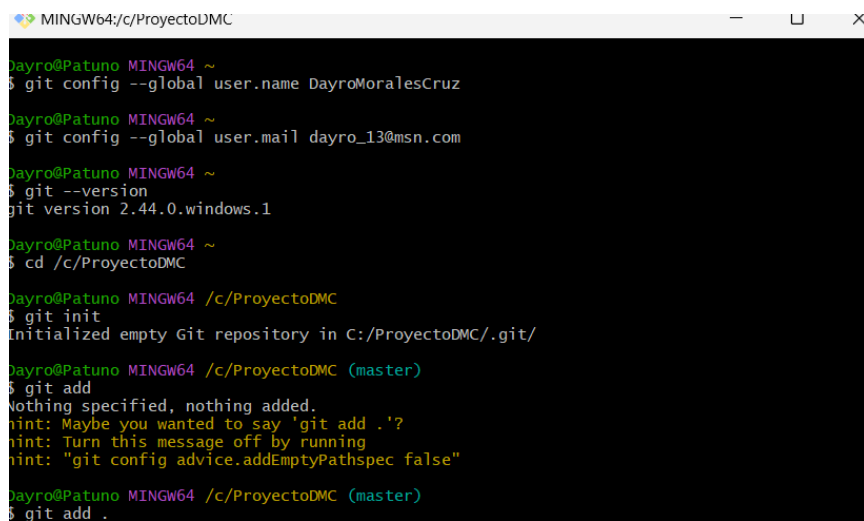
git config --global user.name "Dayro7"

git config --global user.mail "dayro_13@msn.com"

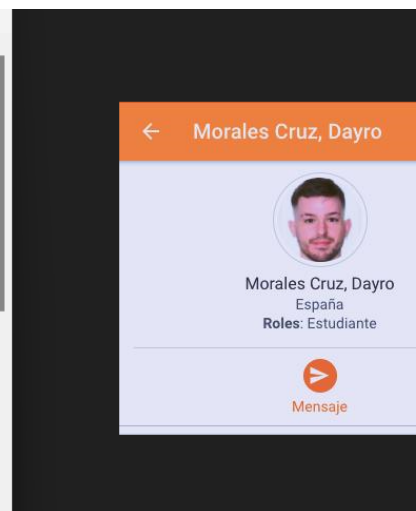
Para pasarlo todo al repositorio

git add .

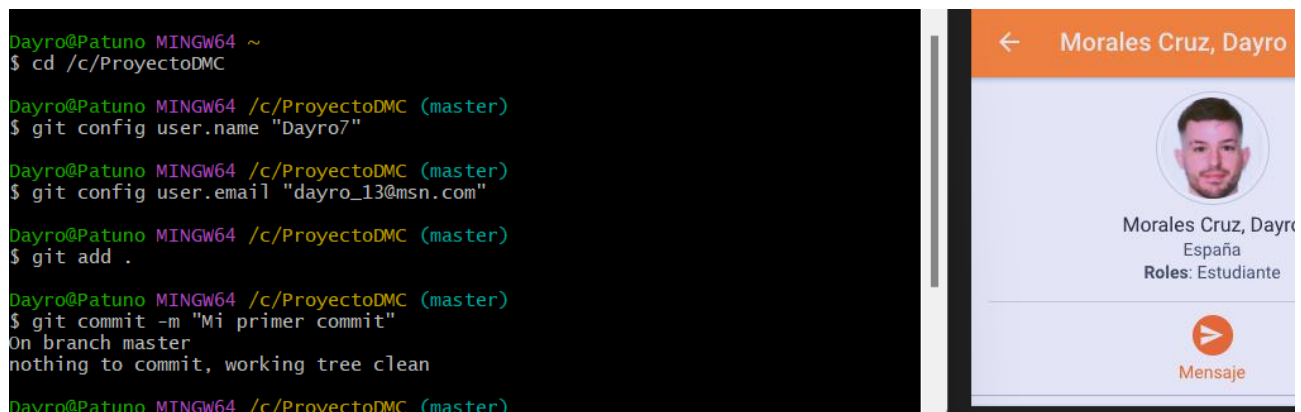
Nos situamos en nuestra carpeta de proyecto y iniciamos con ***git ini***



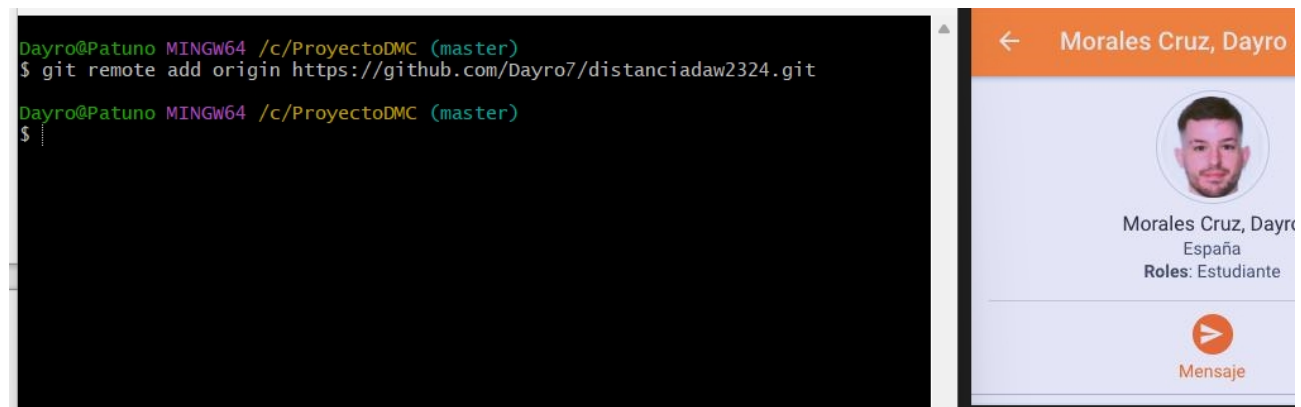
```
MINGW64/c/ProyectoDMC
Dayro@Patuno MINGW64 ~
$ git config --global user.name DayroMoralesCruz
Dayro@Patuno MINGW64 ~
$ git config --global user.mail dayro_13@msn.com
Dayro@Patuno MINGW64 ~
$ git --version
git version 2.44.0.windows.1
Dayro@Patuno MINGW64 ~
$ cd /c/ProyectoDMC
Dayro@Patuno MINGW64 /c/ProyectoDMC
$ git init
Initialized empty Git repository in C:/ProyectoDMC/.git/
Dayro@Patuno MINGW64 /c/ProyectoDMC (master)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'
hint: Turn this message off by running
hint: "git config advice.addEmptyPathsSpec false"
Dayro@Patuno MINGW64 /c/ProyectoDMC (master)
$ git add .
```



Tuve problemas con los nombres y ahora esta corregido, ya realicé el commit y por eso sale el mensaje de que no hay nada que modificar desde mi último commit

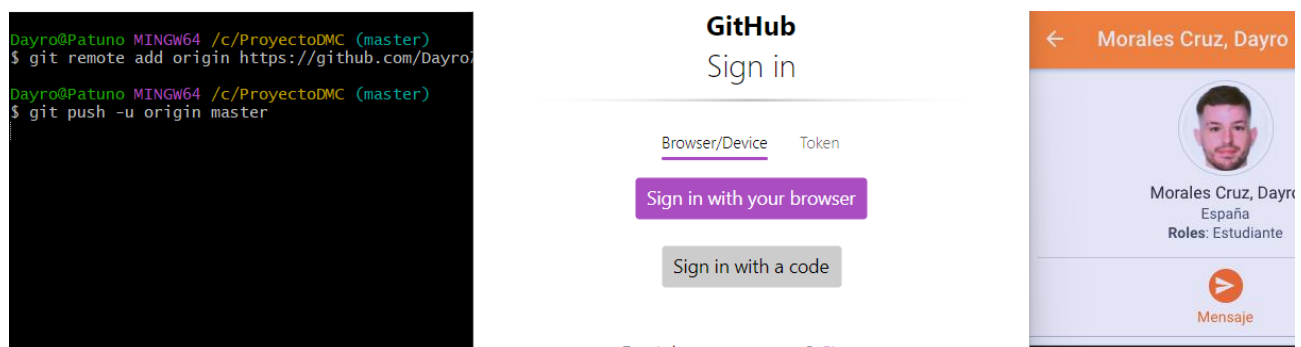


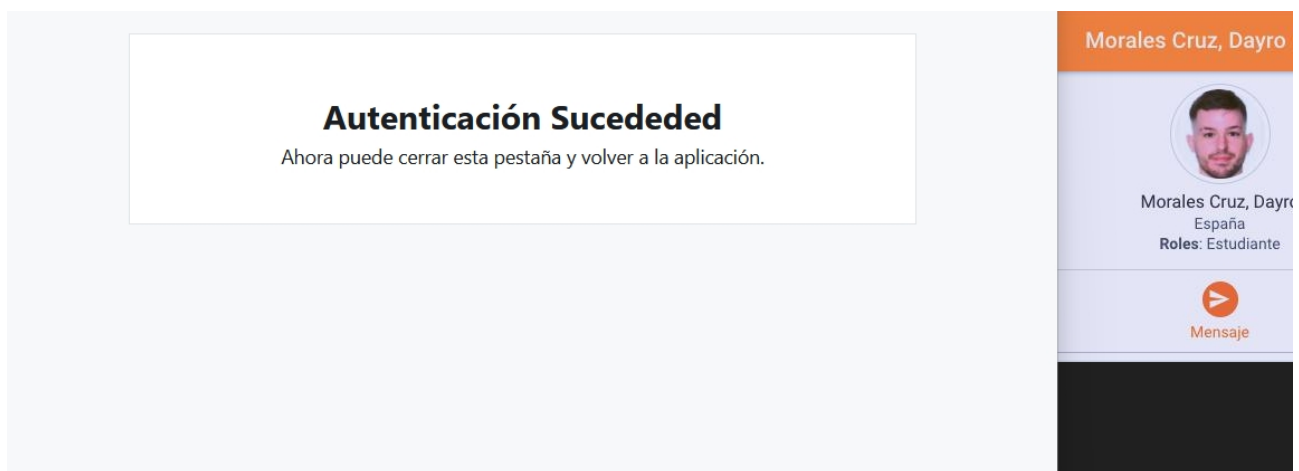
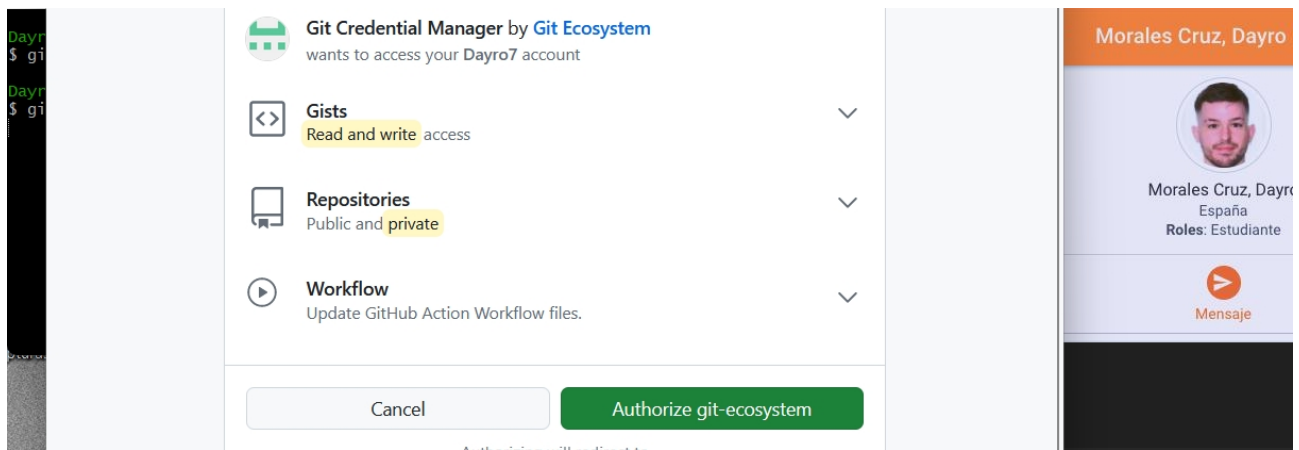
git remote add origin https://github.com/tu_usuario/distanciadaw2324.git Este comando le dice a Git que el repositorio de GitHub que acabas de crear es el “origen” de tu repositorio local



git push -u origin master Este comando sube tus commits al repositorio de GitHub. El -u le dice a Git que recuerde que quieres subir tus cambios a origin (el repositorio de GitHub) en la rama master en el futuro

en la ventana emergente, ingresamos nuestro correo y contraseña de la alta en la pagina de Github





Confirmamos que en el terminal todo se realizó correctamente



Comprobamos que los archivos se subieron correctamente a nuestro repositorio

☆ 0 stars 🍴 0 forks 👁 1 watching 🌿 1 Branch 🏷 0 Tags 🔄 Activity

🌐 Public repository

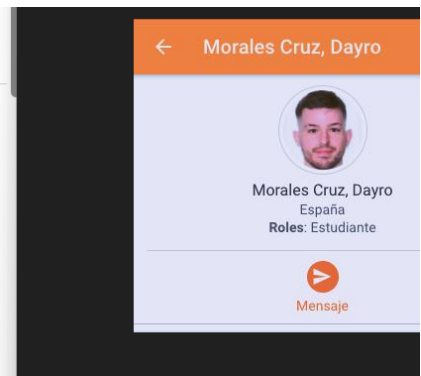
🔗 master 🔗 🔍

Go to file + <> Code

📁 Dayro7 Mi primer commit 15 minutes ago ⌚

📄 practica-Morales.php Mi primer commit 15 minutes ago

📖 README



📁 Dayro7 Mi primer commit da1ac1b · 16 minutes ago ⌚ History

31 líneas (29 loc) · 570 bytes

Código Culpa Crudo 📄 📥 ✎ 🔄

```
1  <?php
2  /**
3   * Este es un script de práctica para PHPDocumentor.
4   *
5   * @author Tu Nombre
6   * @version 1.0
7   */
8
9  /**
10   * Esta función suma dos números.
11   *
12   * @param int $a Primer número a sumar.
13   * @param int $b Segundo número a sumar.
14   * @return int La suma de los dos números.
15   */
16  function suma($a, $b) {
17      return $a + $b;
18  }
19
20  /**
21   * Esta función resta dos números.
22   *
23   * @param int $a El minuendo.
24   * @param int $b El sustraendo.
25   * @return int La resta de los dos números.
26   * @internal Solo para uso interno.
27   */
28  function resta($a, $b) {
29      return $a - $b;
30  }
31  >>
```

