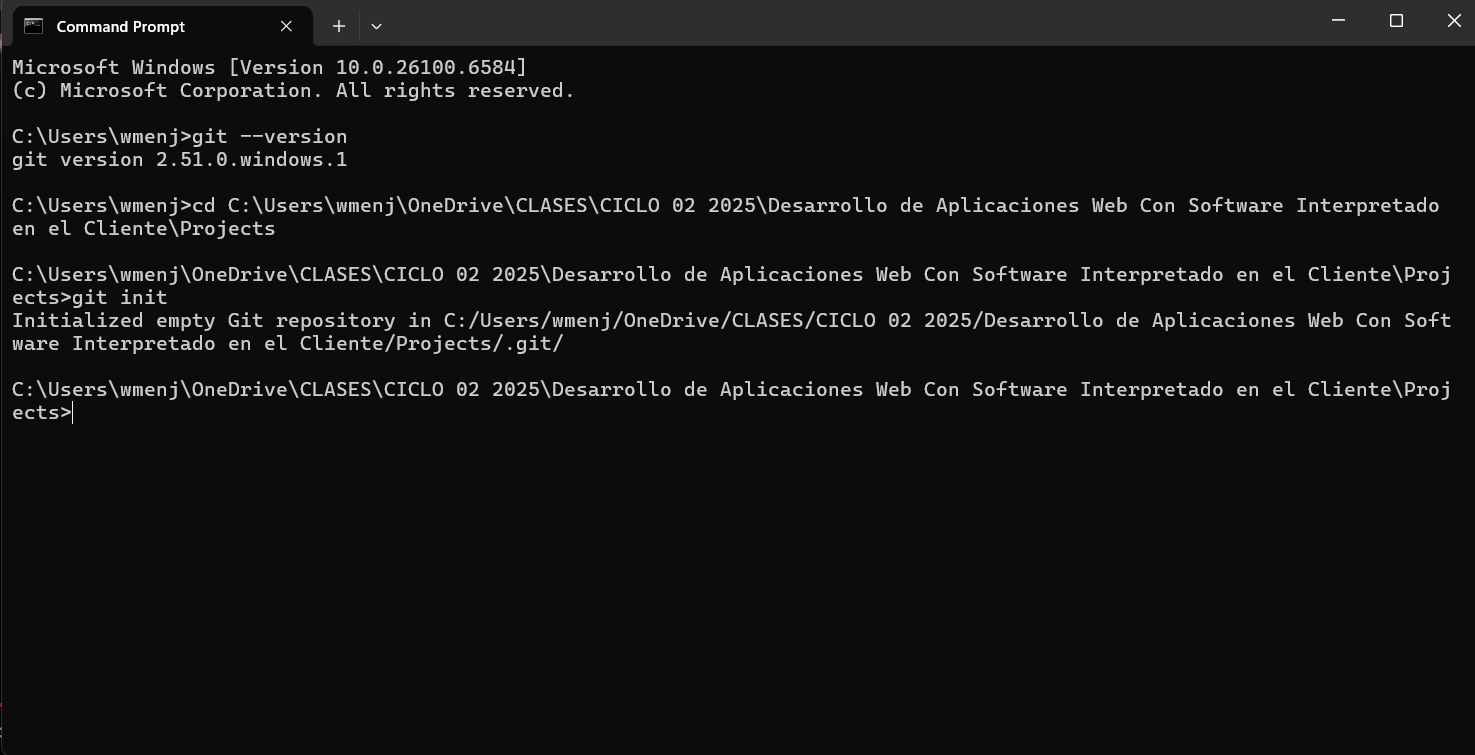
Pasos para el versionamiento con GIT.

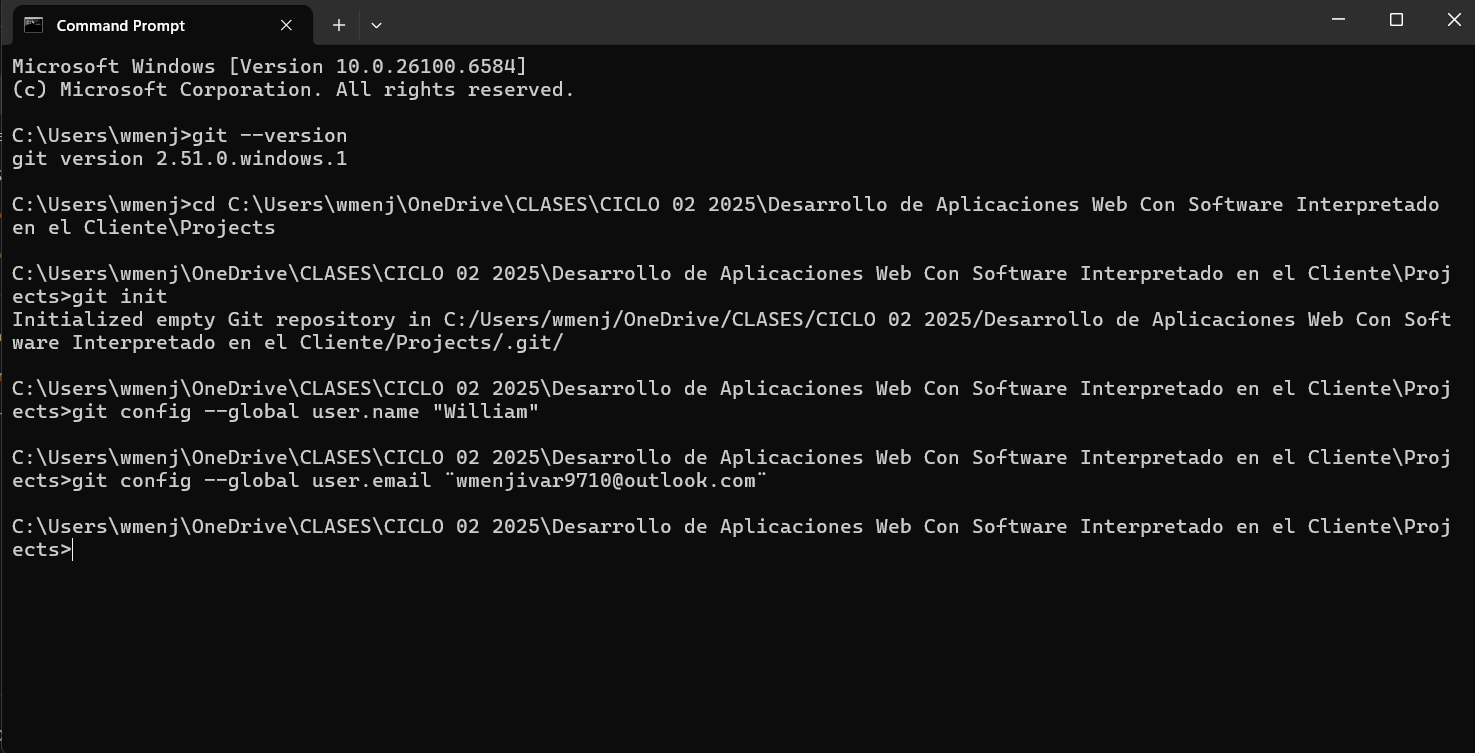
1. Abrimos el CMD desde la barra search de la barra de tareas y para comprobar que Git esta instalado se ejecuta el siguiente comando: git –version.



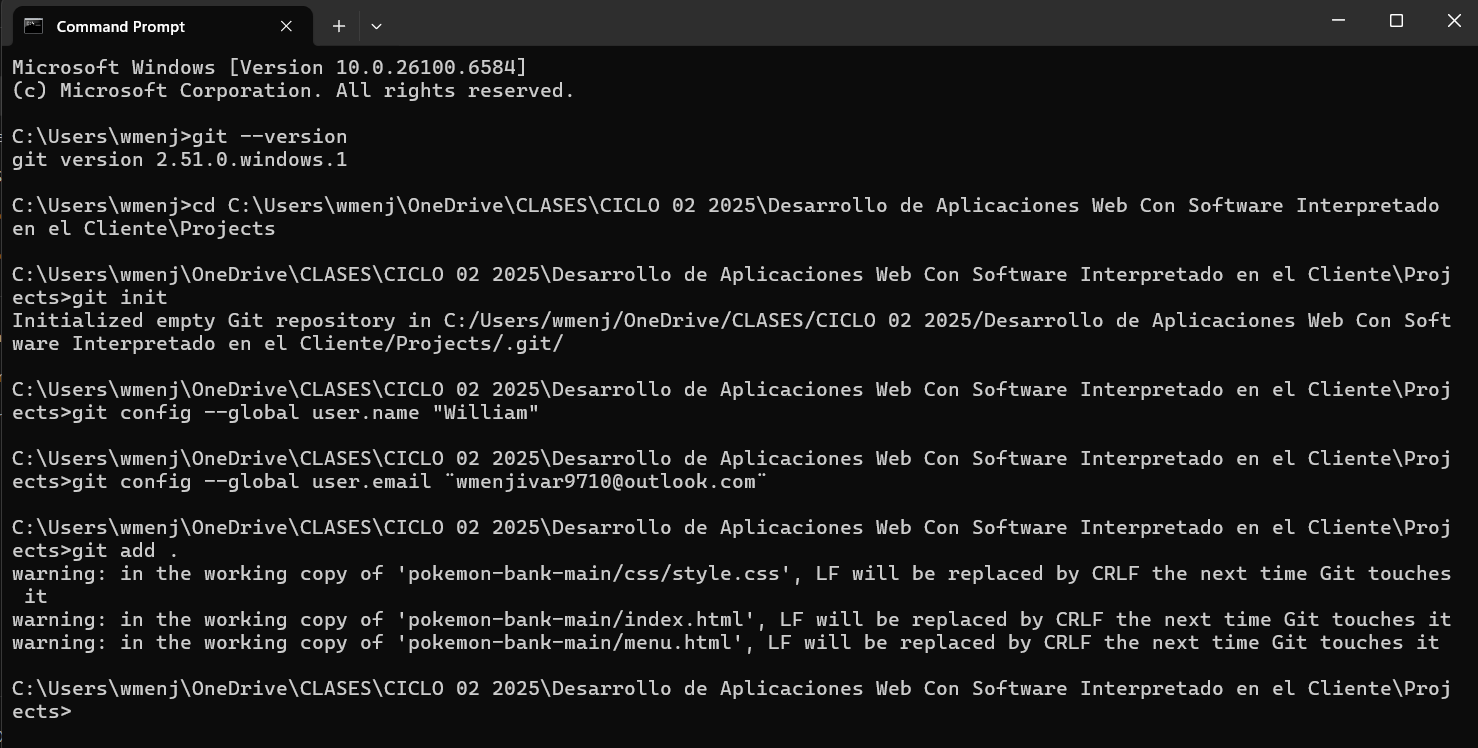
1. Luego se procede a ir al directorio donde esta el código e inicializar el repositorio con git init.



1. Se procede a configurar la identidad:



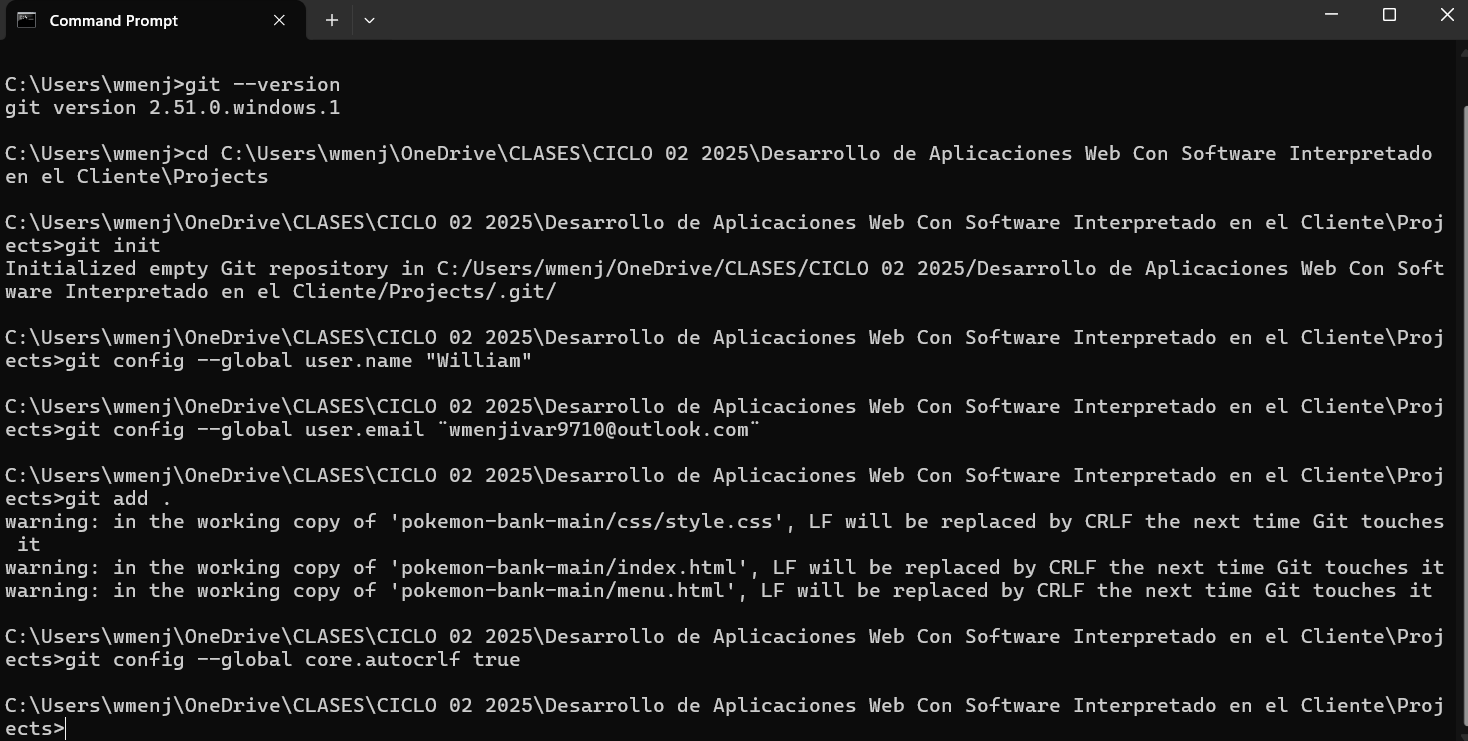
1. Procedemos a añadir la carpeta con los archivos al repositorio con el comando git add .



\*\*El warning que aparece no se debe a ningún error en el proceso sino a que Git detectó que los archivos tienen saltos de línea tipo LF (usados en Linux/macOS), pero como estamos trabajando en Windows, Git los convertirá a CRLF cuando estos se editen o se haga checkout.

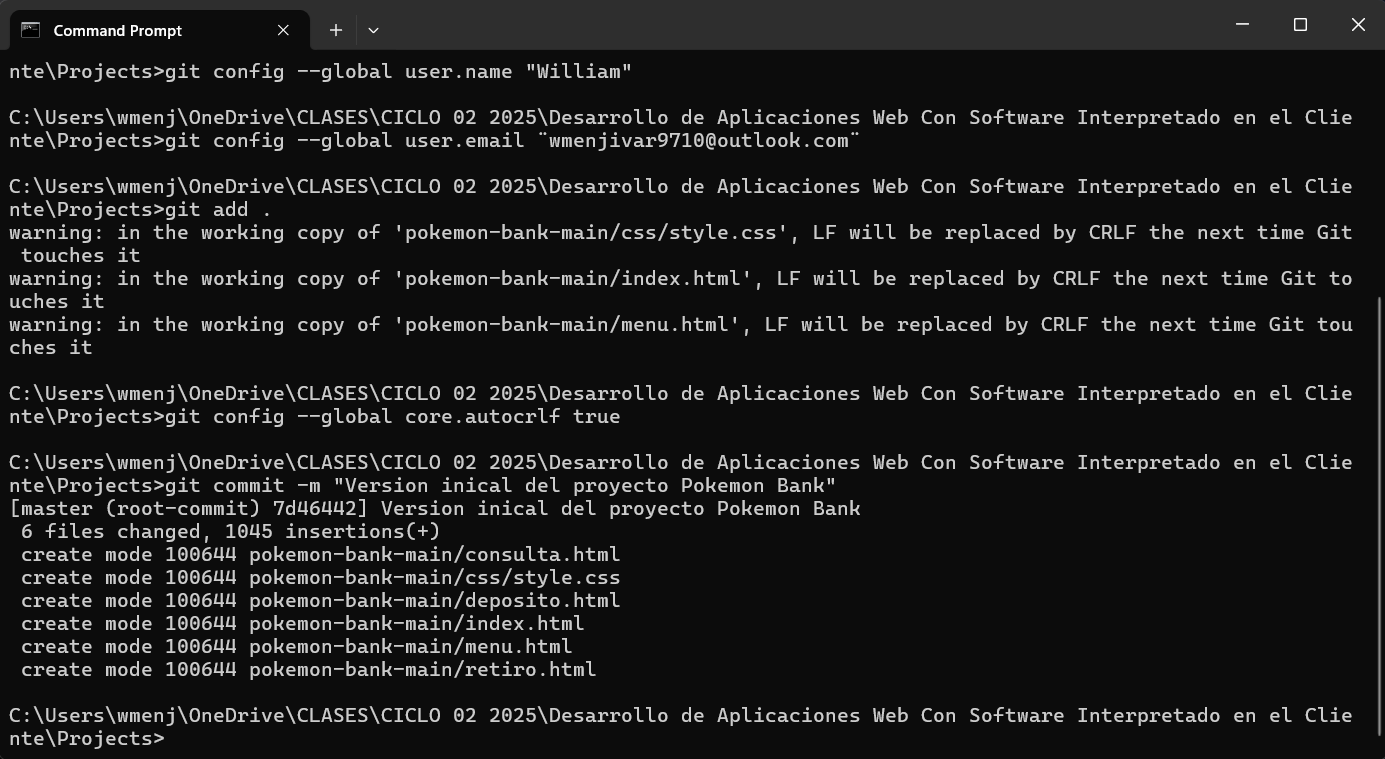
Esto puede solucionarse usando otro comando:

git config --global core.autocrlf true. Como hice en este caso.



1. Procedemos a ejecutar el commit para guardar el proyecto con una pequeña descripción:

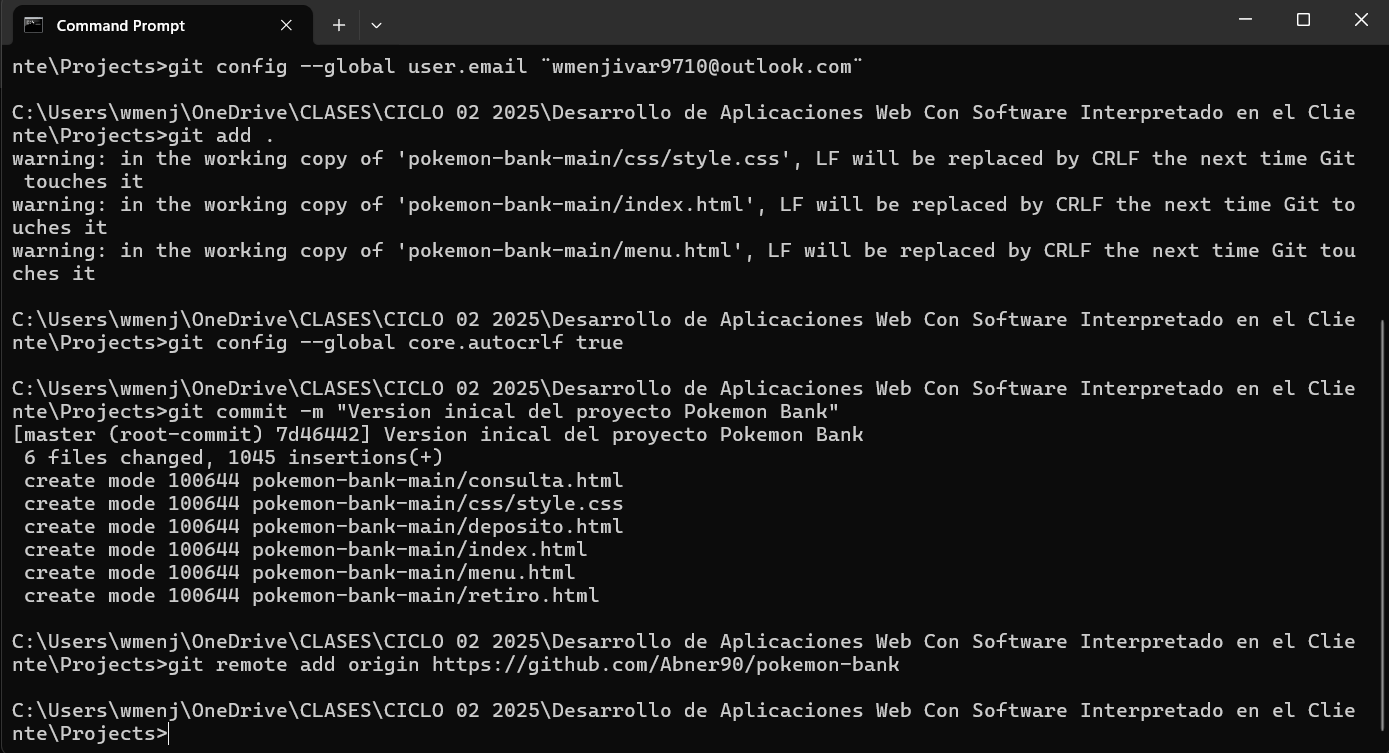
git commit -m "Versión inicial del proyecto Pokémon Bank"



Como se puede apreciar el commit fue exitoso.

1. Finalmente se procede a enlazar el repositorio de Github con Git, para poder hacer el push hacia el repositorio. El comando a utilizar en este paso es el siguiente:

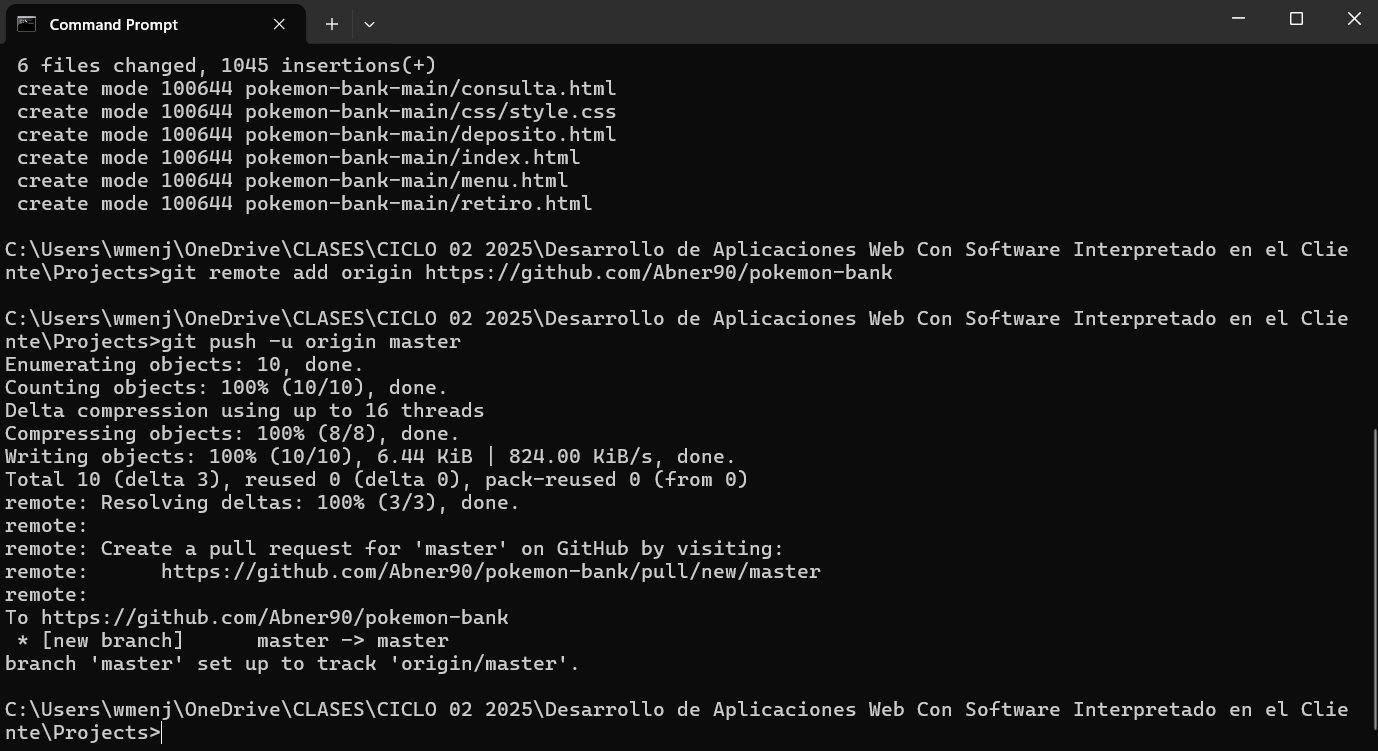
git remote add origin <https://github.com/Abner90/pokemon-bank>



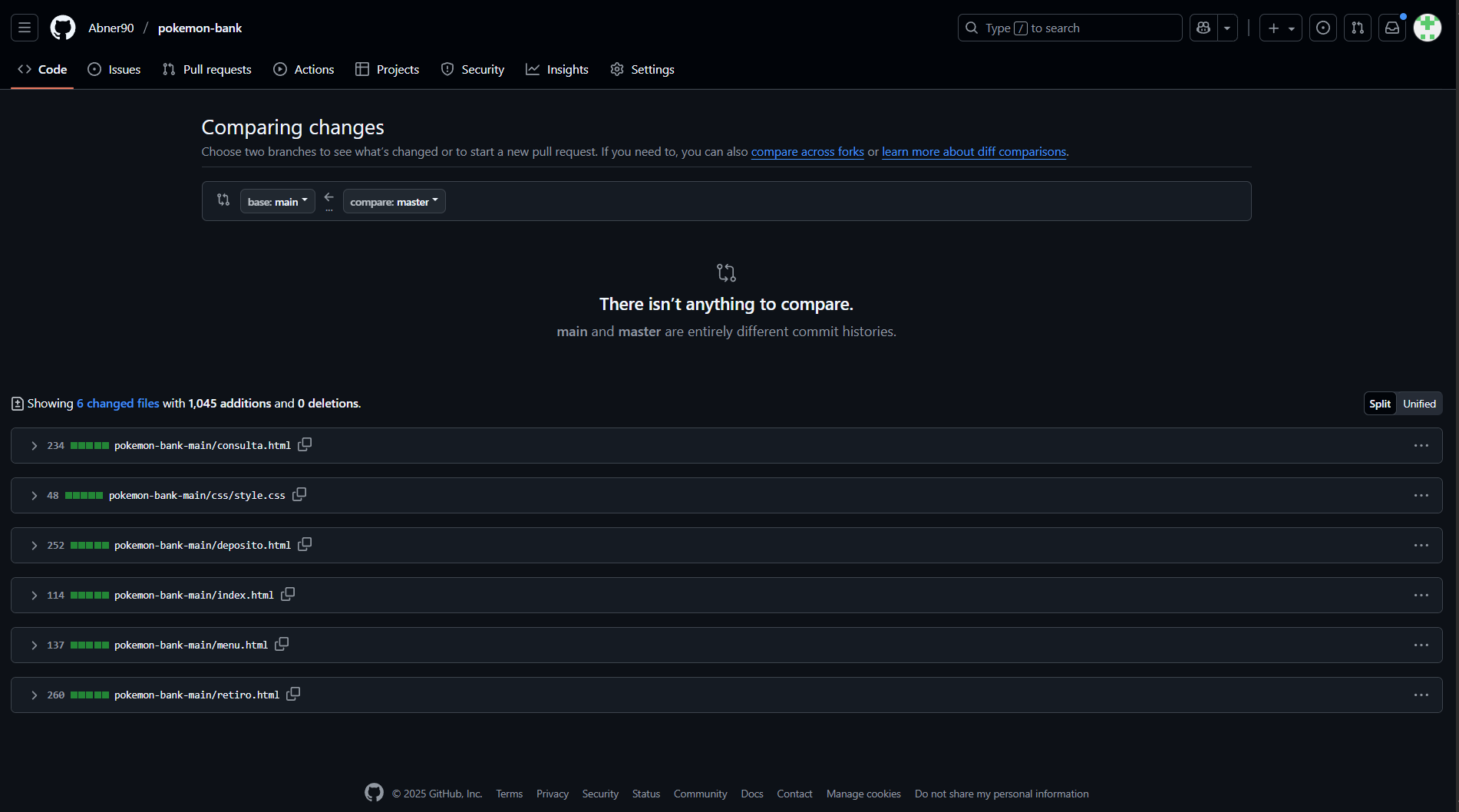
El push se completa con el siguiente comando:

git push -u origin master

el push se completa exitosamente al branch master:



Los cambios y archivos se observan en el repositorio de Github:



Siguiendo estos pasos el proyecto ya esta versionado.

**Versionamiento con Git**

**¿Qué es Git?**  
Git es un sistema de control de versiones distribuido que permite gestionar y registrar los cambios realizados en archivos de código fuente. Fue utilizado en este proyecto para llevar un seguimiento ordenado del desarrollo de la interfaz web del cajero automático Pokémon Bank. Gracias a Git, es posible guardar versiones del proyecto, retroceder a estados anteriores si es necesario, y mantener un historial claro de cada modificación.

**Enlace al repositorio:**  
[Abner90/pokemon-bank: Proyecto DAW901 - Cajero Automatico Pokemon Bank](https://github.com/Abner90/pokemon-bank)

**¿Cómo ayuda Git a mantener versiones y colaborar?**  
Git permite que cada cambio realizado en el proyecto quede documentado mediante *commits*, lo que facilita identificar cuándo y por qué se hicieron modificaciones. Además, al alojar el repositorio en GitHub, se habilita la colaboración entre varios desarrolladores, quienes pueden trabajar en diferentes ramas, proponer mejoras y revisar el código de forma conjunta sin interferir con la versión principal. Esto mejora la organización, la trazabilidad y la eficiencia del trabajo en equipo.

**Librerías JavaScript**

SweetAlert

SweetAlert es una librería JavaScript diseñada para reemplazar las alertas tradicionales del navegador con ventanas modales elegantes y personalizables. Permite mostrar mensajes con íconos, botones interactivos y estilos visuales modernos, mejorando significativamente la experiencia del usuario en aplicaciones web.

Un ejemplo de esta podría ser el siguiente:

<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<script>

Swal.fire('¡Bienvenido!', 'Has iniciado sesión exitosamente.', 'success');

</script>

SweetAlert permite mostrar alertas personalizadas y visualmente atractivas, reemplazando los clásicos de JavaScript el ejemplo de arriba es código fuente de ejemplo.

Chart.js

Chart.js es una librería de visualización de datos que facilita la creación de gráficos dinámicos y responsivos utilizando el elemento de HTML5. Ofrece múltiples tipos de gráficos como barras, líneas, pastel y radar, con una sintaxis sencilla y una apariencia profesional que permite representar información de forma clara y atractiva.

Chart.js permite crear gráficos dinámicos y responsivos usando HTML5 canvas, ideal para visualizar datos de forma clara. Un ejemplo de este seria el siguiente código:

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<canvas id="graficoTransacciones"></canvas>

<script>

const ctx = document.getElementById('graficoTransacciones').getContext('2d');

const chart = new Chart(ctx, {

type: 'bar',

data: {

labels: ['Depósitos', 'Retiros', 'Pagos'],

datasets: [{

label: 'Transacciones',

data: [5, 3, 2],

backgroundColor: ['#4CAF50', '#FF9800', '#2196F3']

}]

}

});

</script>