**Logotipo, nombre de la empresa

Descripción generada automáticamenteIcono

Descripción generada automáticamente**Tecnológico Nacional De México

Instituto Tecnológico de Oaxaca.

Dpto. Sistemas Computacionales

7SA – Programación Web

Cuadro sinóptico sobre los estándares de la IEEE 802

Nombre:

Mendoza Ramírez Cristian Eduardo

Docente: Luis Eduardo Osorio Hernández

31 de Marzo de 2025.

CONTENIDO

INTRODUCCIÓN…………………………………………………….………….3

USO DE PLY EN EL ANÁLISIS SINTACTICO……………………………….4

INTEGRACIÓN DE PLY EN NUESTRA APLICACIÓN……………………...5

CONCLUSIÓN…………………………………………………………………...6

**INTRODUCCIÓN**

Git es un sistema de control de versiones distribuido que permite gestionar y registrar los cambios en el código de un proyecto de software. Gracias a Git, los desarrolladores pueden trabajar en equipo sin perder el historial de modificaciones, facilitando la colaboración y la recuperación de versiones anteriores del código si es necesario.

GitHub, por otro lado, es una plataforma en línea que utiliza Git para alojar proyectos y mejorar el trabajo en equipo. Con GitHub, varios desarrolladores pueden contribuir a un mismo proyecto, hacer revisiones de código y gestionar tareas de manera organizada.

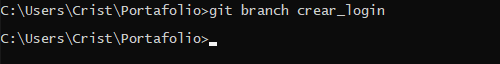
Las ramas (branches) en Git permiten dividir el desarrollo en diferentes líneas de trabajo sin afectar la versión principal del proyecto. Esto ayuda a implementar nuevas funciones, corregir errores o preparar actualizaciones sin comprometer el código estable.

Un commit es un registro de cambios en el código. Cada commit tiene un mensaje que describe lo que se modificó, facilitando el seguimiento del desarrollo. Los commits son esenciales porque permiten:

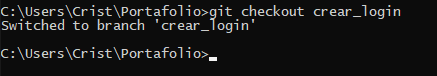
* Guardar avances de forma ordenada.
* Documentar el progreso del proyecto.
* Restaurar versiones anteriores en caso de errores.
* Coordinar mejor el trabajo en equipo.

**CREACIÓN DE RAMAS**

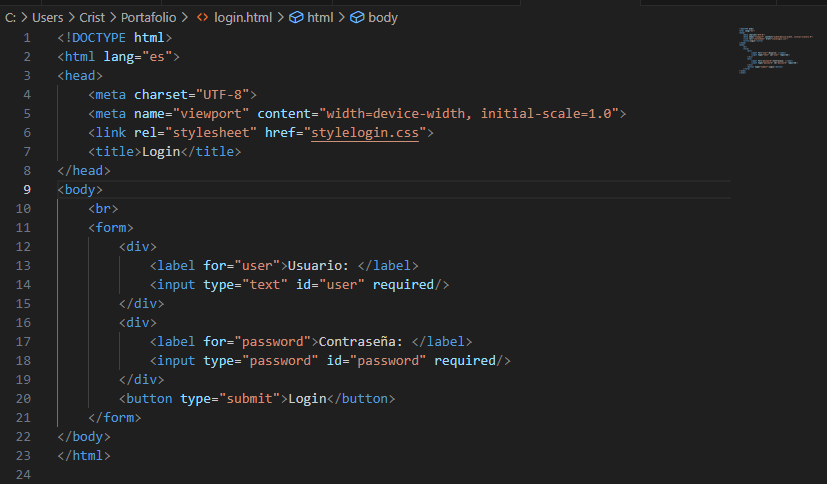
Para comenzar a crear los distintos archivos HTML, generaremos una rama separada para cada archivo. Para ello, utilizamos el siguiente comando:



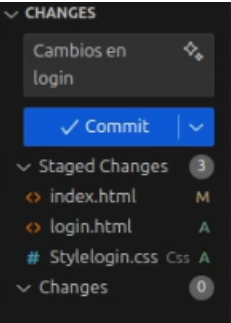
Para cambiar a la rama que acabamos de crear escribimos el siguiente comando:

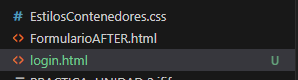


Ahora crearemos nuestro archivo login.html en visual studio

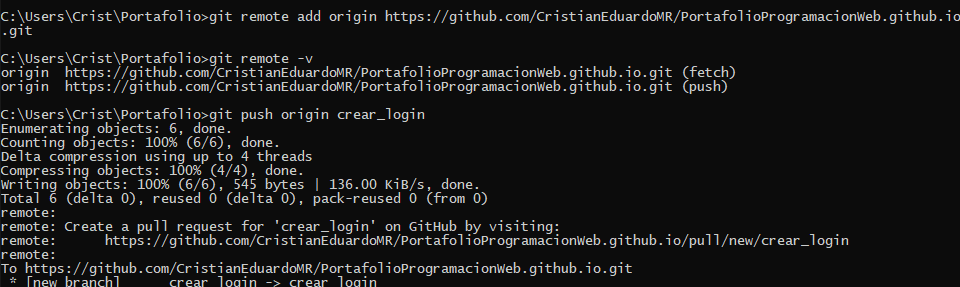


Una vez que tengamos listo el código, vamos a hacer un commit en Visual Studio. Primero, vamos a la pestaña **Changes**, donde veremos los cambios hechos en el proyecto. En **Staged Changes**, seleccionamos los cambios que queremos guardar en el repositorio. Luego, en **Commit Message**, escribimos una breve descripción de los cambios.

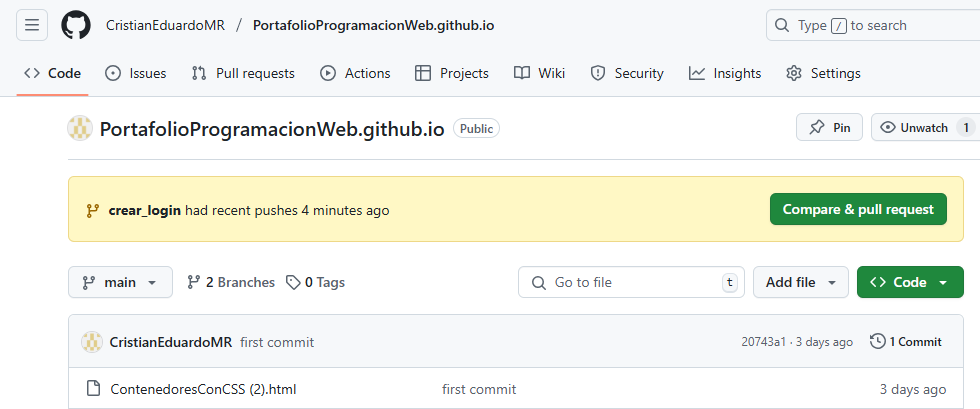




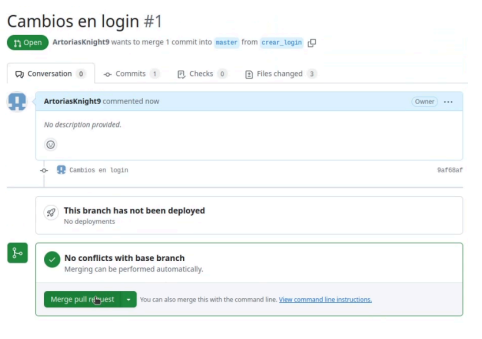
Después de hacer el commit, introduciremos los siguientes comandos para hacer un push a nuestro repositorio:



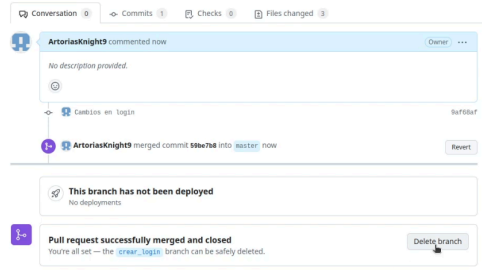
Esto se reflejará en nuestro repositorio como una **pull request** (solicitud para que los cambios realizados en una rama se fusionen con otra rama del repositorio).



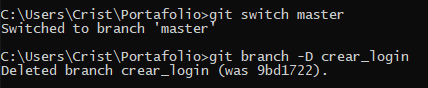
Después de crear la pull request, presionamos en Merge pull request para fusionar los cambios de la rama en la rama principal (o la rama de destino).



Después de fusionar la rama, podemos **borrar la rama** que creamos directamente en GitHub, ya que los cambios ya han sido incorporados a la rama principal (o la rama de destino).



Después de borrar la rama en GitHub, cambiaremos a la rama principal y eliminaremos la rama local que creamos utilizando el siguiente comando:



Por último haremos un pul con el comando: git pull

