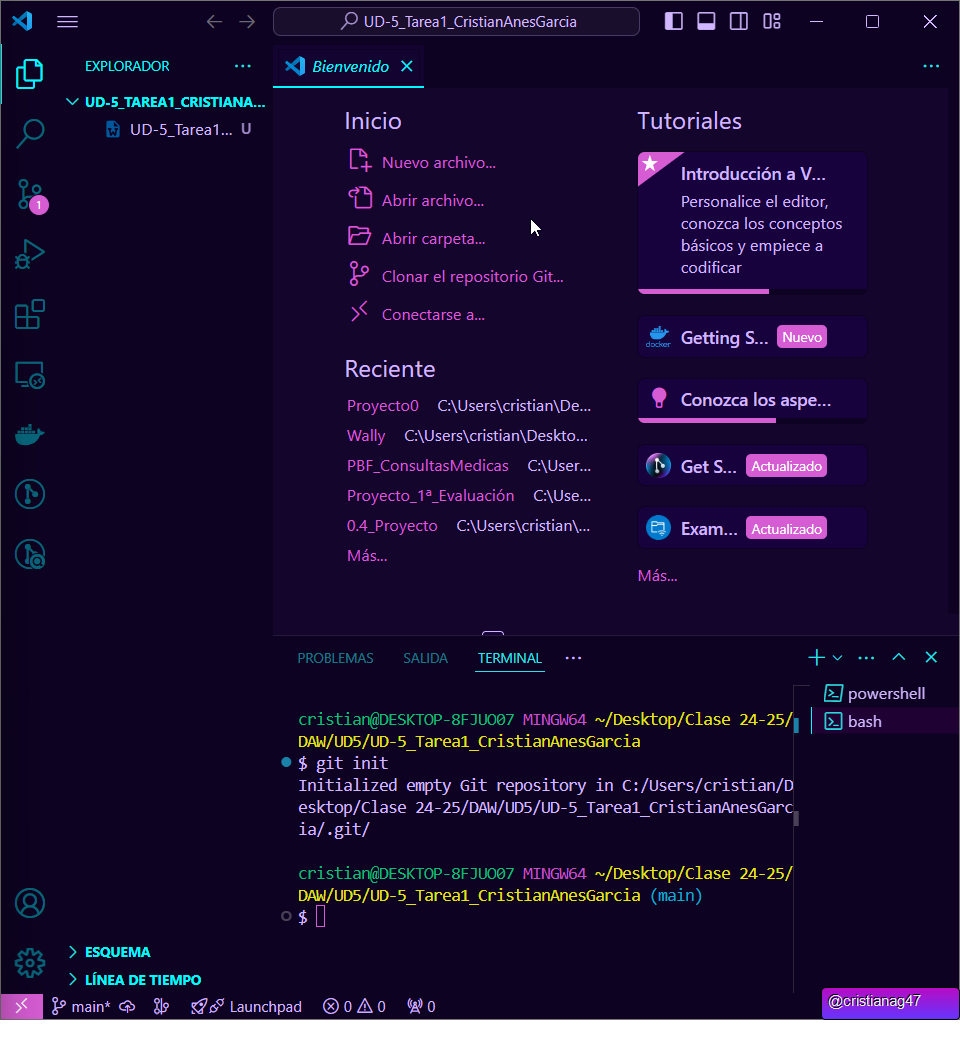
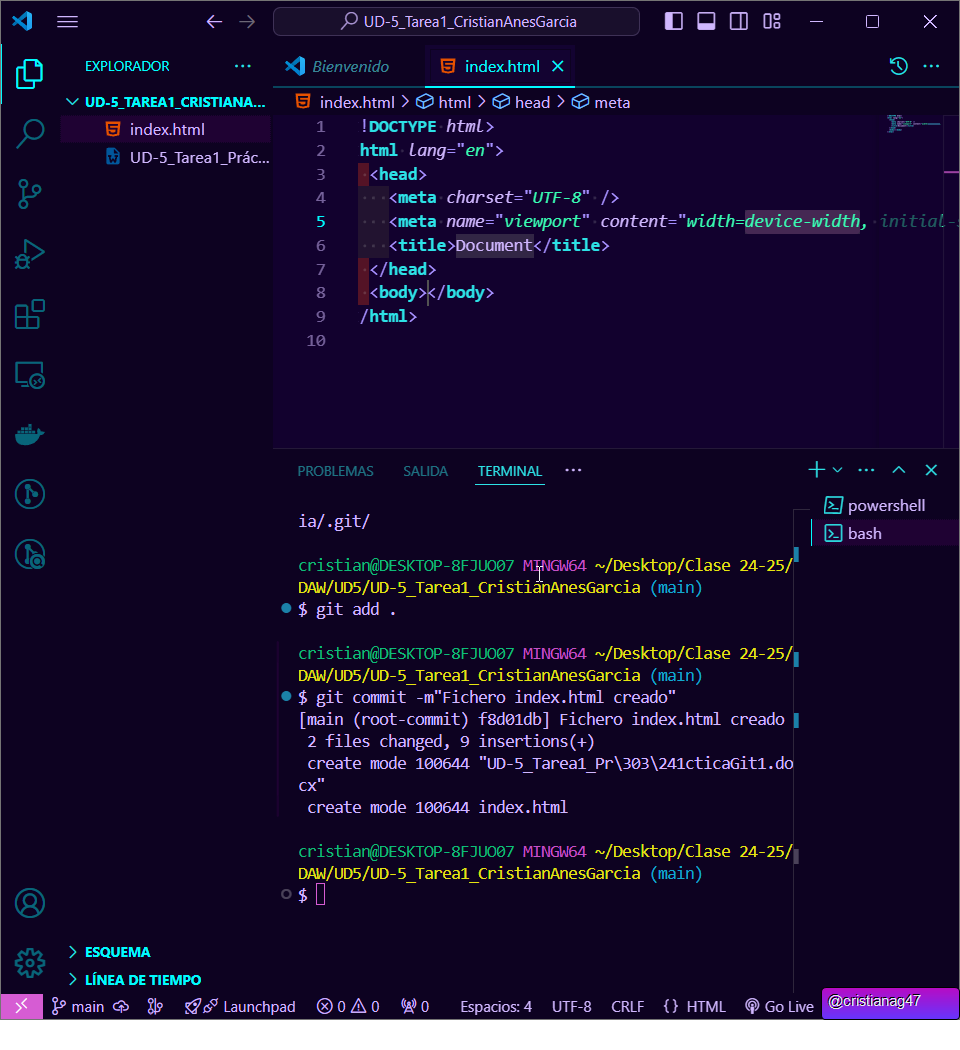
|  |  |  |
| --- | --- | --- |
|  | **DESPLIEGUE APLICACIONES WEB 2ºDAW** | **Curso 2024-2025** |
| **UD-5: Sistemas de Control de Versiones GIT** | |
| **Tarea1: Aprendiendo a usar GIT** | |
| **Nombre:** | **Grupo:** |

Para esta práctica serán necesaria una máquina con Windows, que tenga instalado el programa Visual Studio Code (<https://code.visualstudio.com/>), y el sistema GIT (<https://git-scm.com>). Los pasos a seguir son los indicados a continuación. Se deben incluir pantallazos (tras realizar cada paso) de los resultados obtenidos en cada uno de ellos, incluyendo una breve descripción de cada pantallazo.

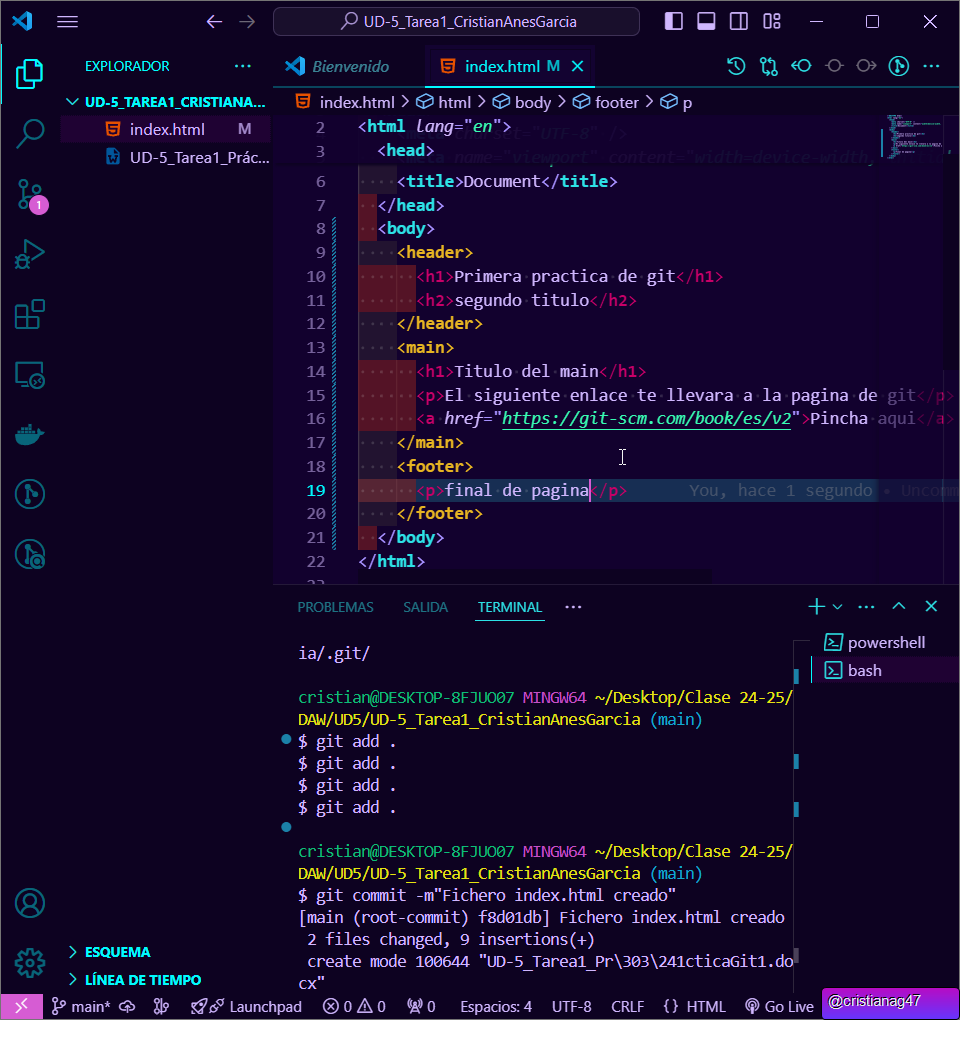
* Crear una carpeta llamada “Tema 5” (o similar) dentro del directorio de trabajo del alumno. Dentro dicha carpeta, crear otra llamada UD-5\_Tarea1\_NombreAlumno. A continuación, abrir dicho directorio con el VSCode, y por último inicializar el repositorio git en dicho dicho directorio desde la línea de comandos (Comando git init)



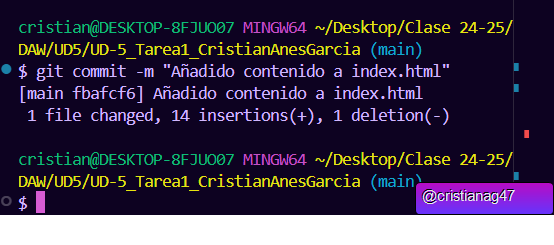
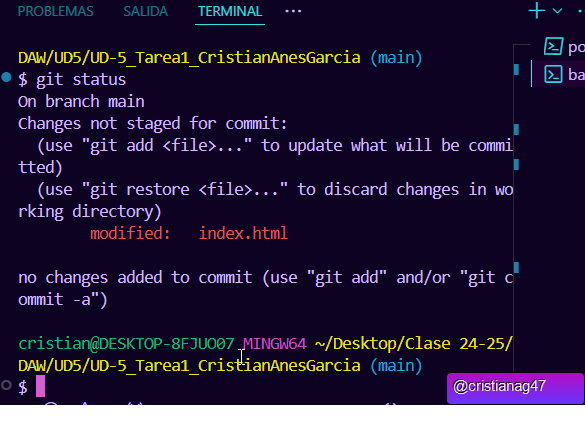
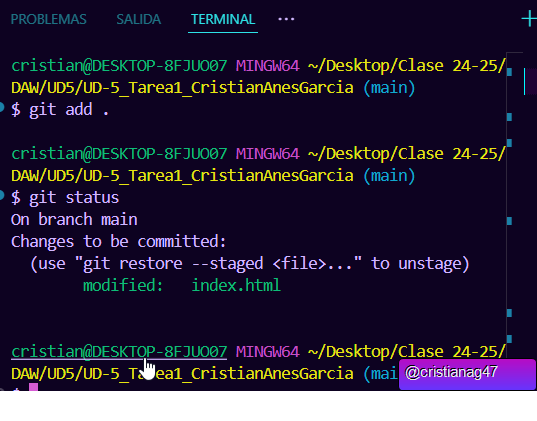
* Desde VSCode, crear el archivo index.html, e incluir el código básico de una estructura HTML. A continuación, guardar el archivo, añadirlo a la zona de staging y hacer un commit de los cambios con la descripción “Fichero index.html creado”.



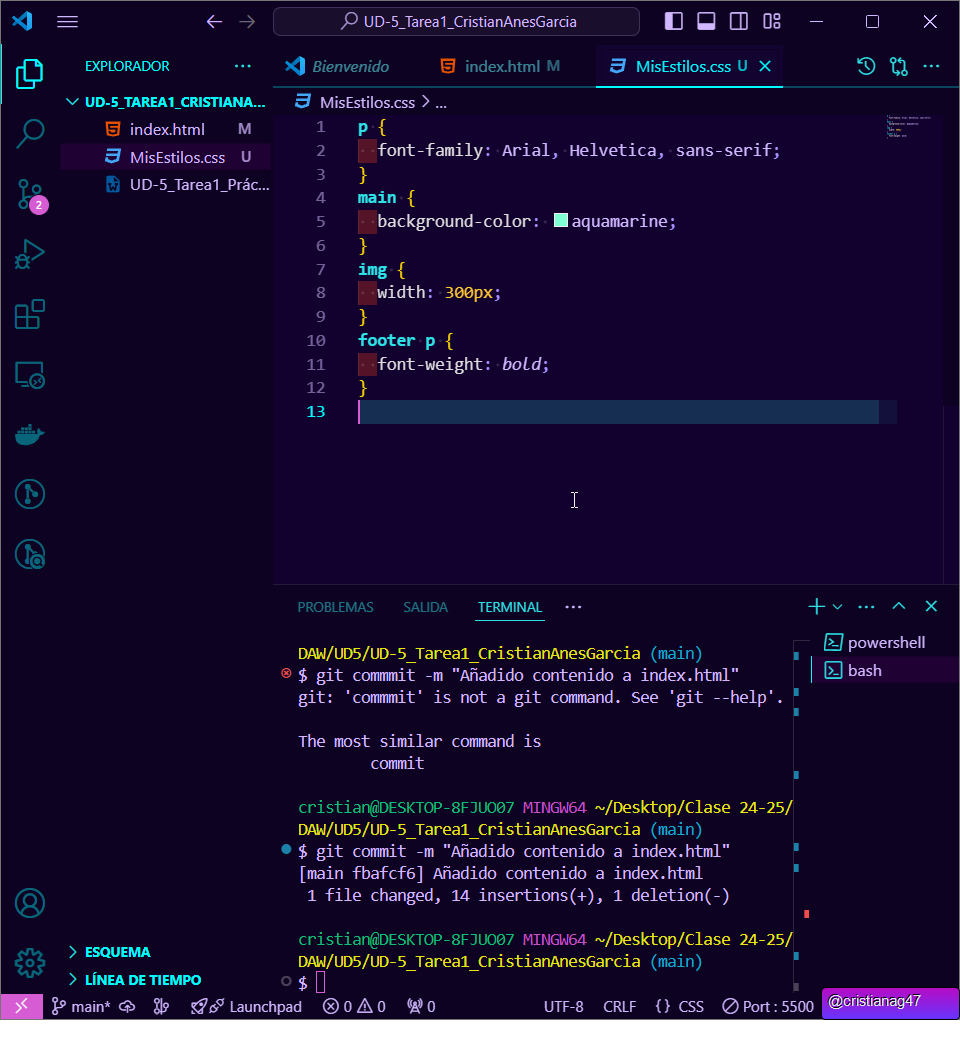
* Añadir en el <body> del fichero index.html los siguientes elementos (los textos a incluir en cada elemento son a elección libre del alumno).
  + Una sección de tipo <header>, con Un título de tipo <h1> y otro de tipo <h2>.
  + Una sección <main>, con un título de tipo <h1>, un párrafo <p> y un enlace a la siguiente página: <https://git-scm.com/book/es/v2>
  + Una sección tipo <footer> con un párrafo <p>.



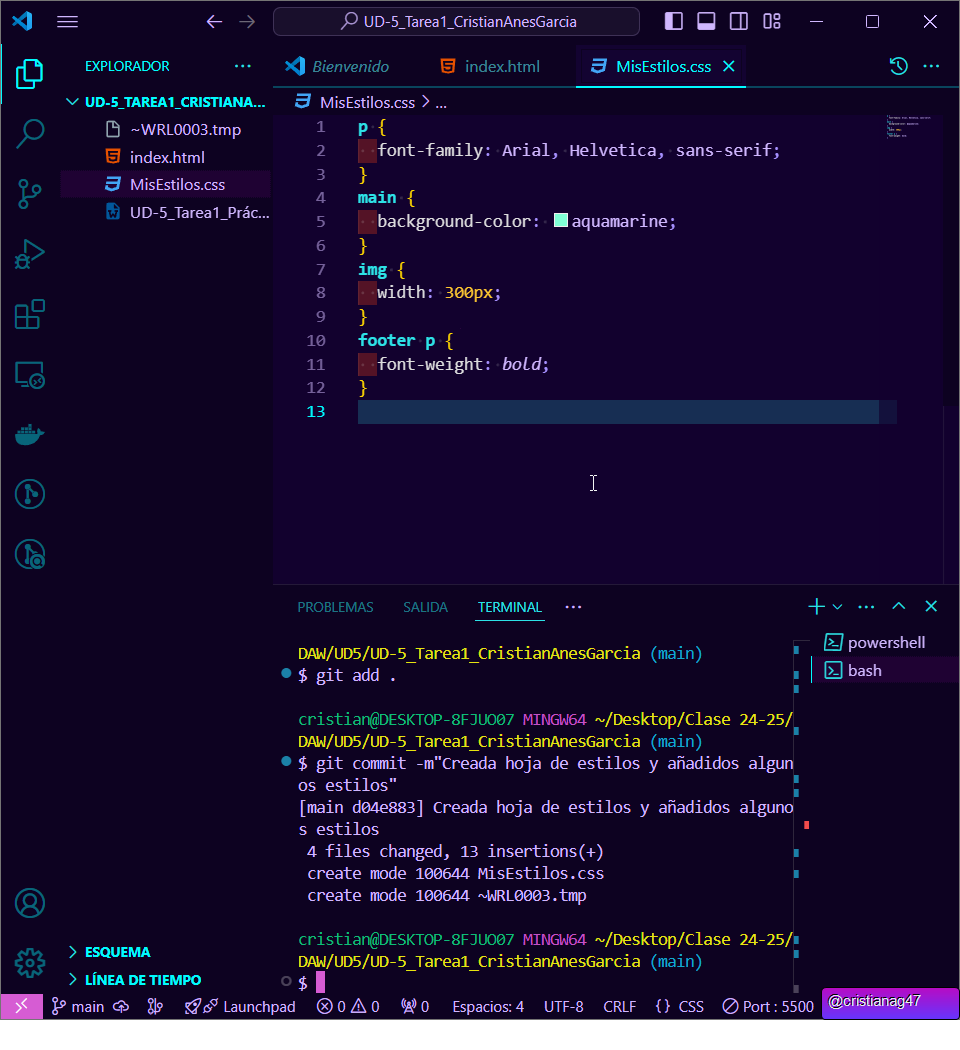
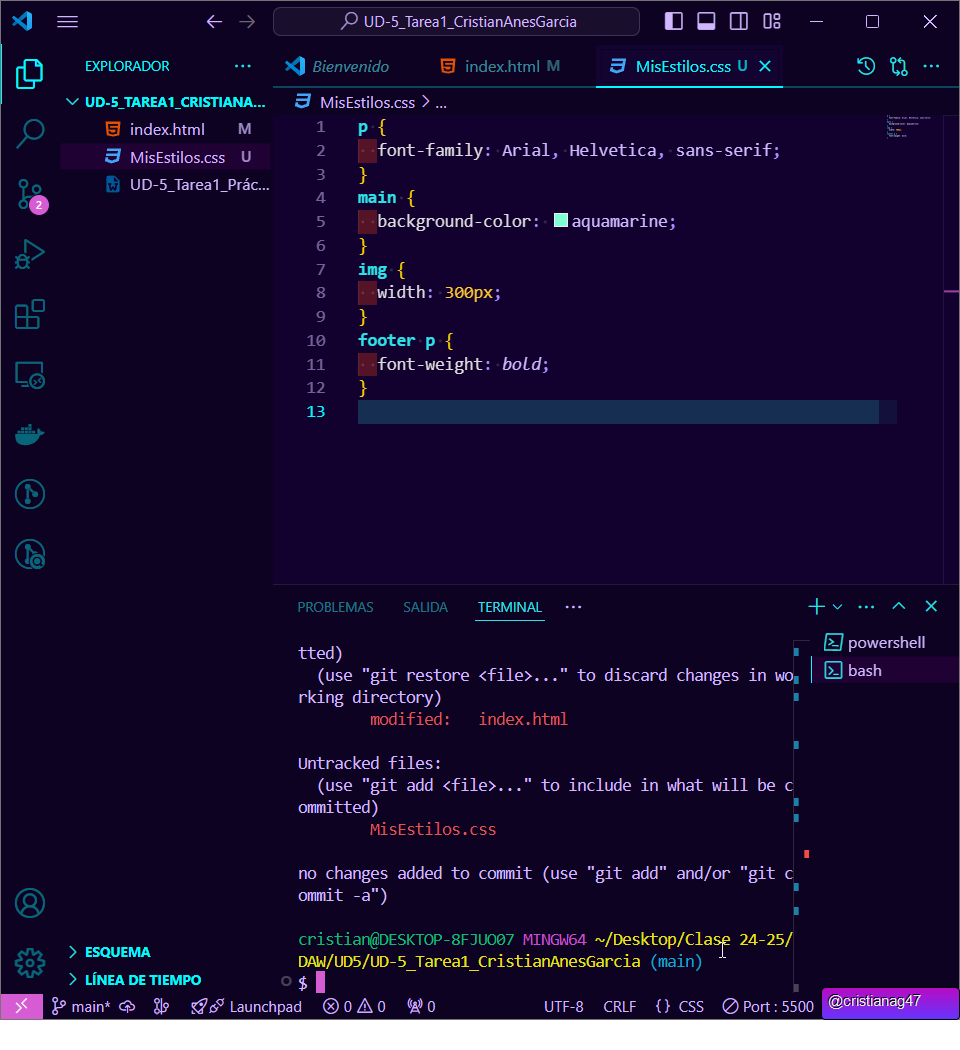
* Usar el comando git status para comprobar que se detectan los nuevos cambios. Añadir el archivo index.html a la zona de staging y a continuación confirmar los cambios, con la descripción “Añadido contenido a index.html”.



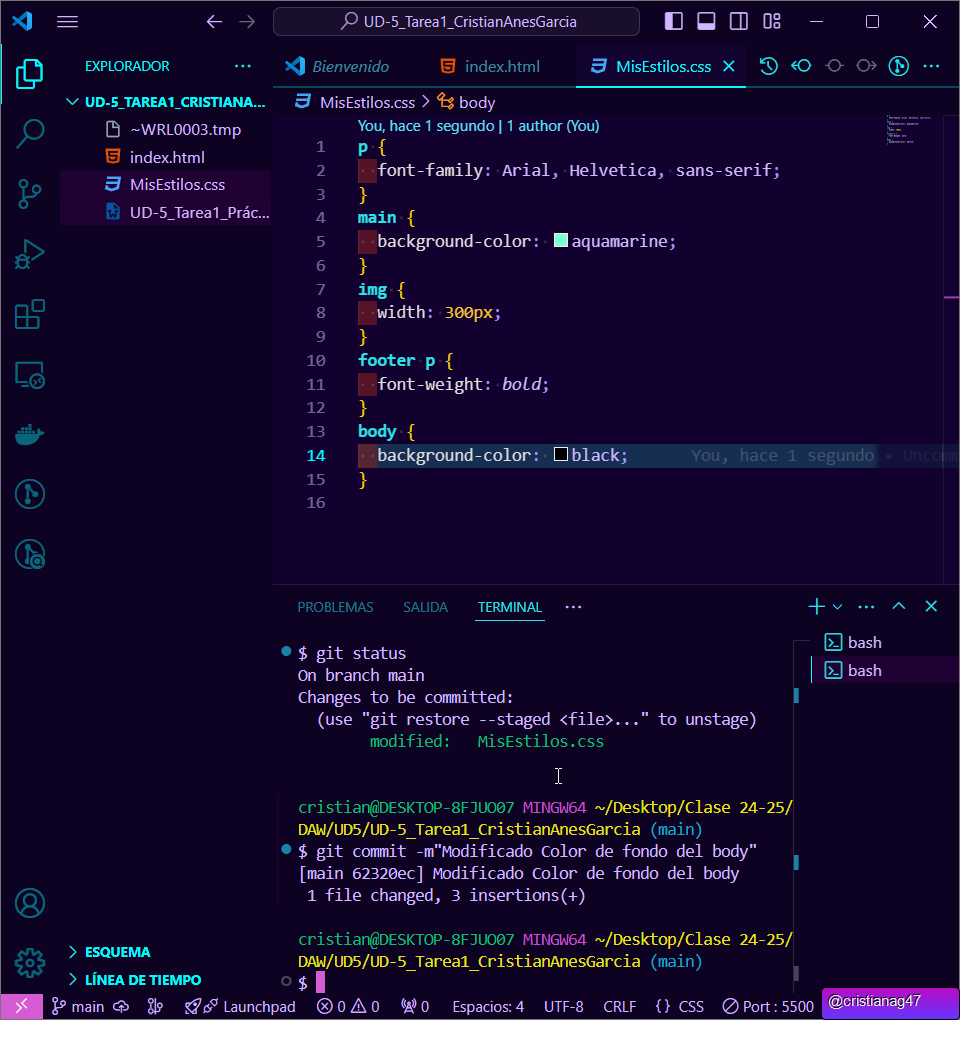
* Desde VSCode, crear el archivo “MisEstilos.css”, y añadir las reglas de estilo necesarias para hacer lo siguiente:
  + Que los párrafos tengan la fuente Arial.
  + Que el main tenga color de fondo aquamarine.
  + Que las imágenes tengan un ancho de 300px.
  + Que los párrafos del footer estén en negrita.



* Modificar el archivo index.html para que importe la hoja de estilos creados. Usar el comando git status para comprobar que se detectan los nuevos cambios en ambos ficheros. Añadir ambos ficheros a la zona de staging y a continuación confirmar los cambios, con la descripción “Creada hoja de estilos y añadidos algunos estilos”.



* Modificar la hoja de estilos añadiendo una regla para poner el fondo del <body> de color negro. Añadir el archivo MisEstilos.css a la zona de staging y confirmar los cambios con la descripción “Modificado Color de fondo del body”.



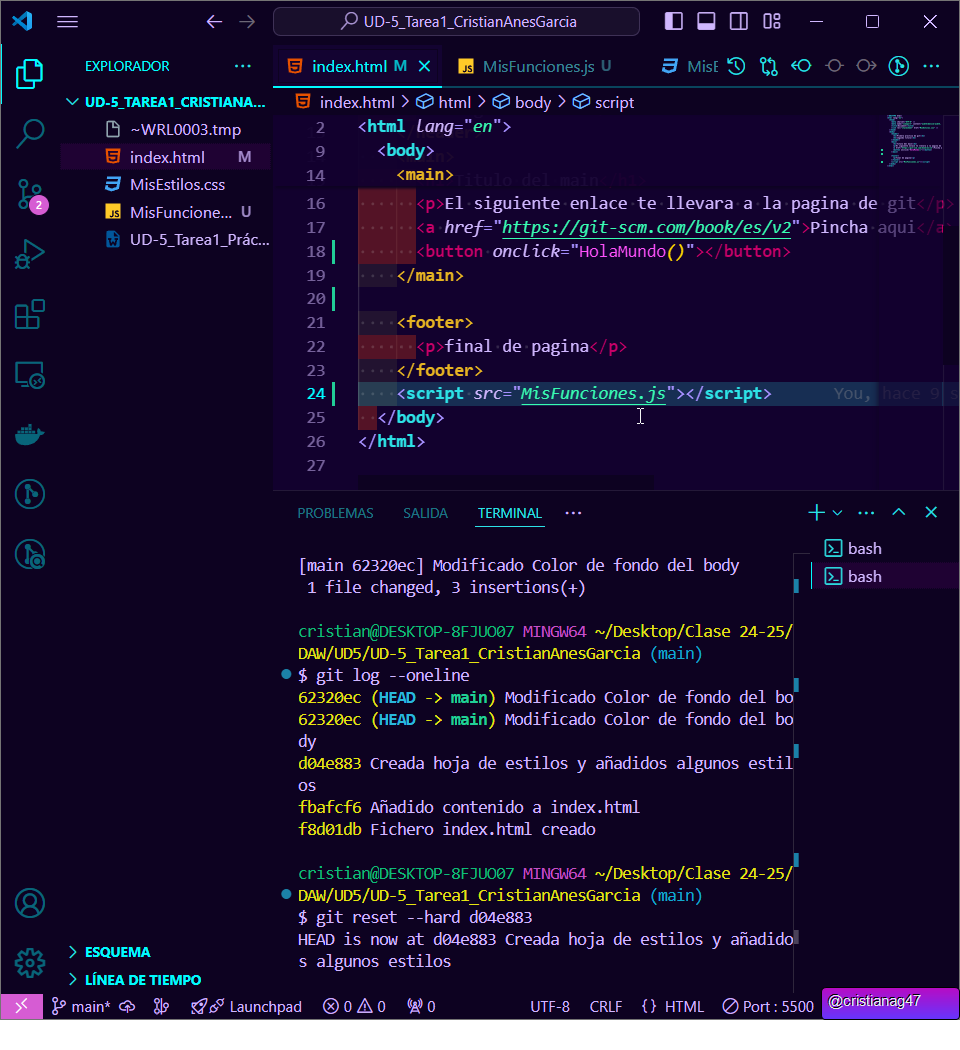
* Abrir con un navegador el archivo index.html, para ver que el cambio de estilos no ha sido muy apropiado.



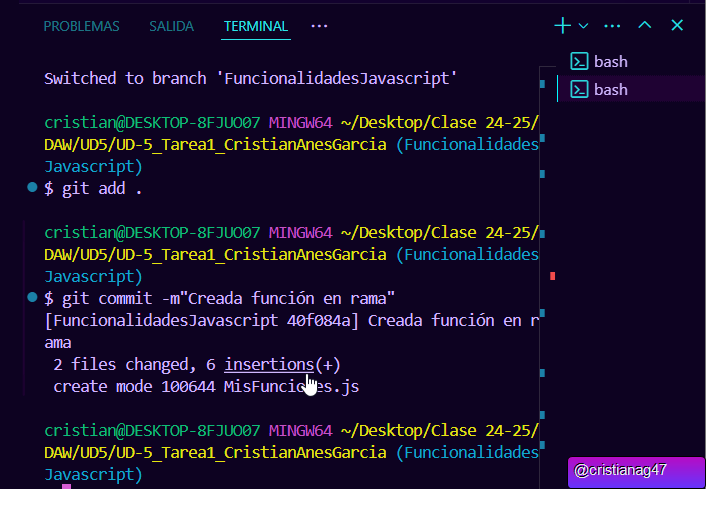
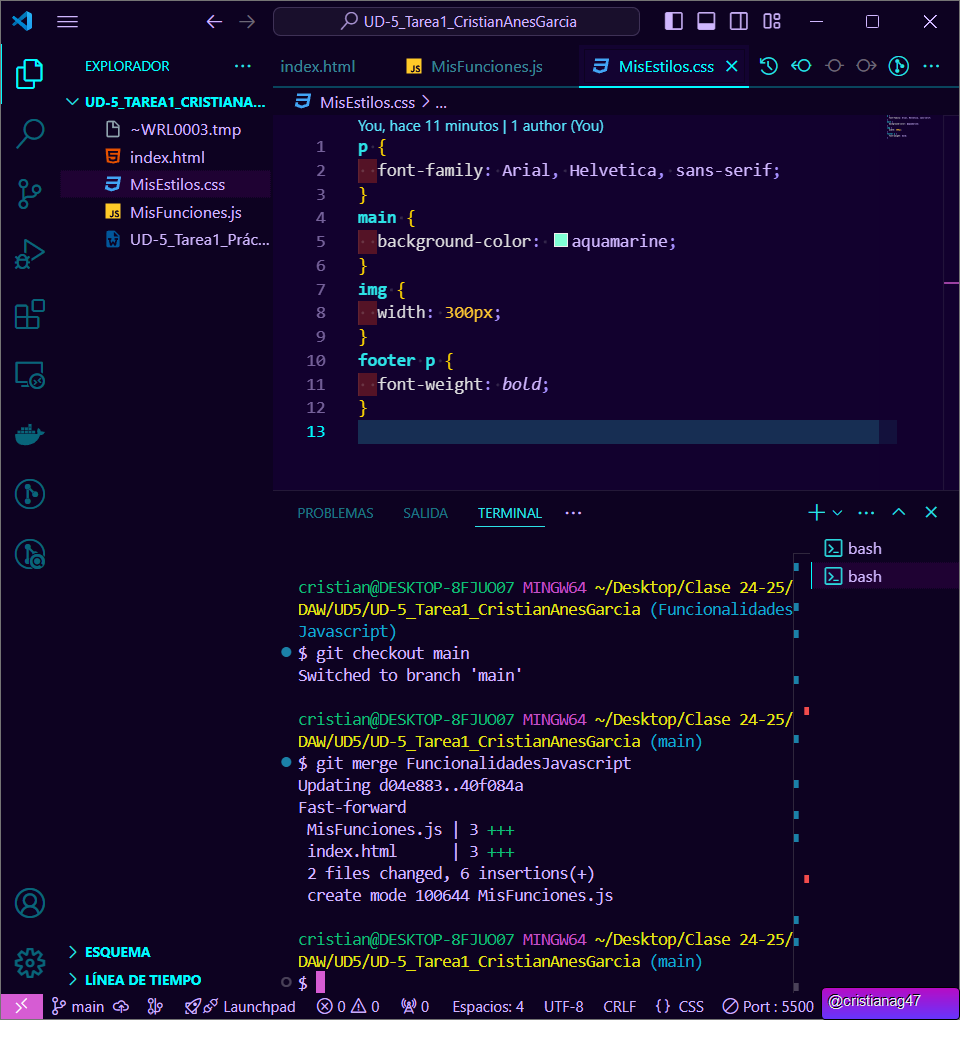
* Retornar el proyecto a una versión anterior usando el comando git reset --hard (buscar en Google como usarlo correctamente). Una vez realizado, hacer git status y volver a abrir index.html en el navegador para comprobar si está todo ok.



* Desde VSCode crear un fichero llamado “MisFunciones.js” (de momento estará vacío) y en el <main> de index.html añadir un botón que al pulsarlo llame a la función “HolaMundo()”.



* Crear una nueva rama llamada FuncionalidadJavascript y colocarnos en ella. A continuación, en el archivo “MisFunciones.js” creamos la función “HolaMundo()” cuyo código simplemente consta de un alert(“Hola Mundo”);. Confirmar los cambios realizados en dicha rama (descripción: Creada función en rama), y posteriormente realizar un merge con la rama master (descripción: Función javascript mergeada).



* Eliminar la rama FuncionalidadesJavascript. A continuación, Crear una nueva rama llamada “EstilosAdicionales”, y realizar los cambios necesarios en el fichero “MisEstilos.css” (tanto en la nueva rama como en la master) para que al intentar el merge de ambas ramas salte un conflicto y debamos decidir qué cambios queremos mantener (debemos quedarnos con los de la rama “EstilosAdicionales”). Los cambios a realizar en los estilos son libres, a elección del alumno. Realizar un commit final con la descripción “Versión final lista para publicar”.