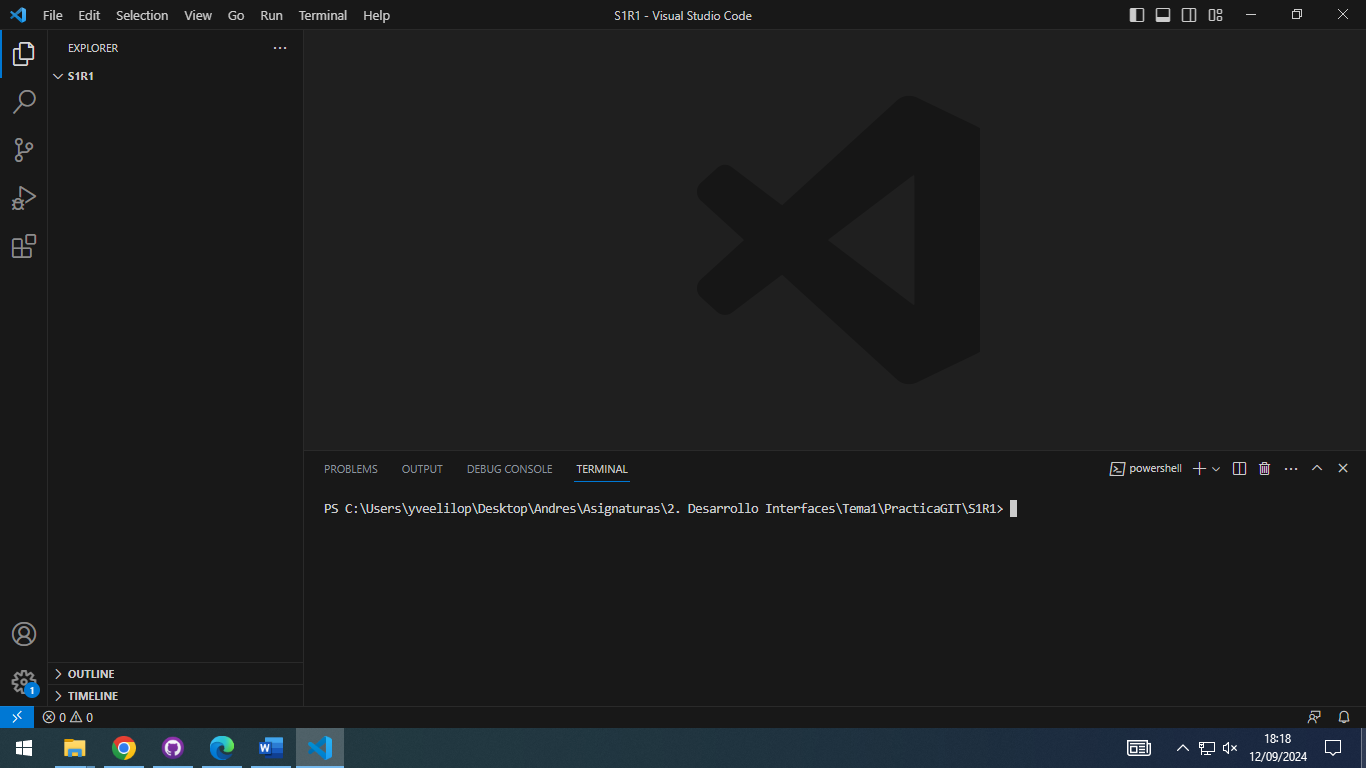
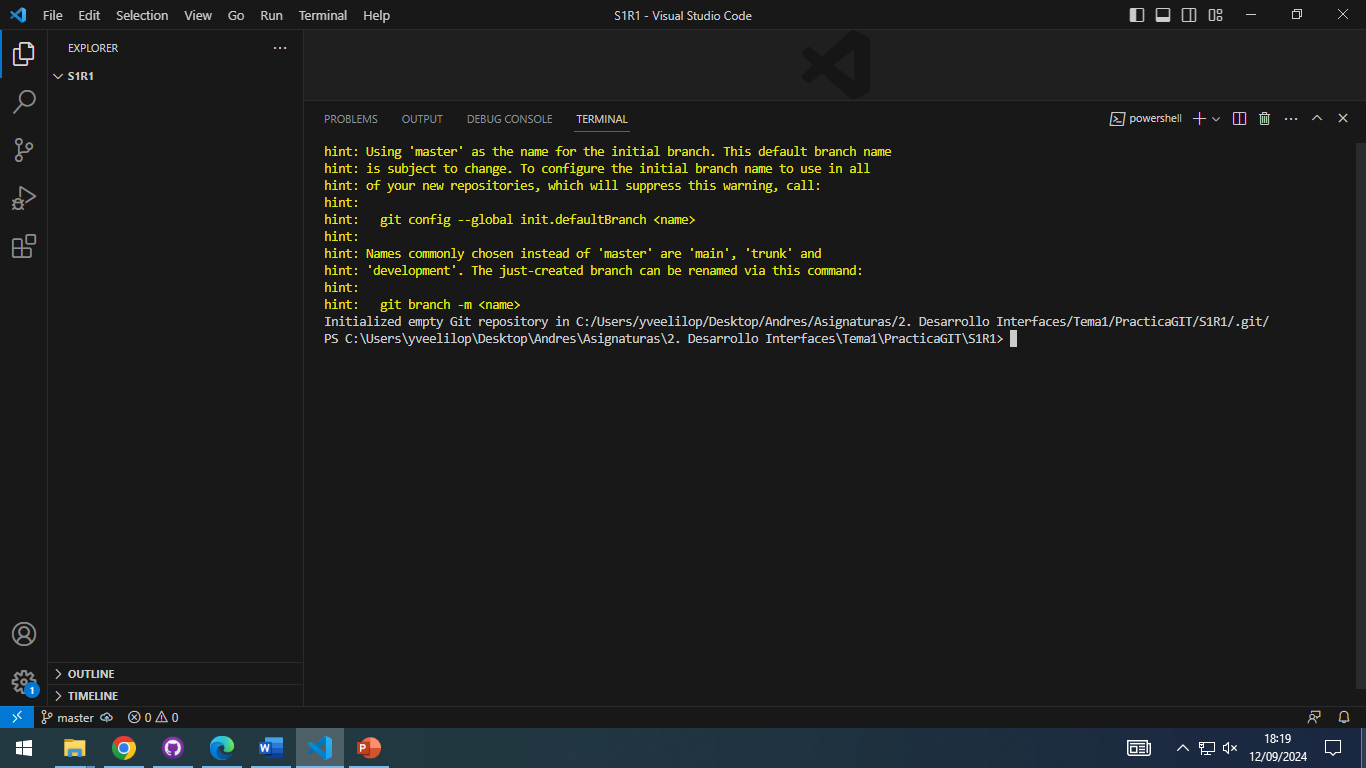
**1. FUNDAMENTOS DE GIT I**

**1. Crea una carpeta denominada S1R1. Realiza las siguientes acciones en ella.**

*a) Crear un repositorio GIT.*





*b) Crea un fichero denominado* ***libros.txt.***

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

*c) Haz un primer ‘commit’.*

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

*d) Añade dos libros al archivo* ***libros.txt.***

*e) Haz un segundo ‘commit’.*

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

*f) Crea un fichero denominado* ***peliculas.txt.*** *Añade tres títulos de películas a dicho archivo.*

*g) Haz una captura de pantalla del comando git status.*

Captura de pantalla de computadora

Descripción generada automáticamente

*h) Crea un fichero denominado* ***comidas.txt.*** *Añade tres nombres de comidas a dicho archivo.*

Captura de pantalla de computadora

Descripción generada automáticamente

*i) Haz un tercer ‘commit’ que incluya los archivos* ***peliculas.txt*** *y* ***comidas.txt.***

Captura de pantalla de computadora

Descripción generada automáticamente

*j) Elimina el archivo* ***comidas.txt*** *desde el navegador de archivos.*

*k) Añade dos películas más al archivo* ***peliculas.txt.***

*l) Haz una captura de pantalla que muestre los cambios en el directorio de trabajo.*

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

*m) Añade los cambios al área de preparación.*

*n) Haz una captura de pantalla del comando git status. Debe indicar que se ha borrado el archivo* ***comidas.txt*** *y que se ha modificado el archivo* ***peliculas.txt.***

Captura de pantalla de computadora

Descripción generada automáticamente

*ñ) Haz un cuarto ‘commit’.*

Captura de pantalla de computadora

Descripción generada automáticamente

*o) Crea un archivo denominado* ***datos.bak.*** *Añade tres títulos de libros a dicho archivo.*

Captura de pantalla de computadora

Descripción generada automáticamente

*p) Crea una subcarpeta denominada output. Crea un archivo denominado salida.txt en su interior. Escribe tu nombre y apellidos en dicho archivo.*

*q) Haz una captura de pantalla del comando git status. Deben aparecer los archivos* ***datos.bak*** *y* ***output/salida.txt*** *como archivos nuevos (color rojo).*

*r) Crea un archivo .gitignore para que los ficheros con extensión* ***.bak*** *y el contenido de la carpeta output/ no se incluyan en el repositorio.*

*s) Haz una nueva captura de pantalla del comando git status. Ahora no deben aparecer los archivos* ***datos.bak*** *y* ***output/salida.txt*** *como archivos nuevos, sino que en su lugar debe aparecer únicamente el archivo* ***.gitignore.***

*t) Haz un último ‘commit’ para incluir el archivo* ***.gitignore*** *en el repositorio.*

*u) Haz una captura de pantalla que muestre el histórico de cambios del repositorio.*

**2. FUNDAMENTOS DE GIT II**

**1. Crea una carpeta denominada S2R1. Realiza las siguientes acciones en ella.**

*a) Crea un repositorio Git.*

Captura de pantalla de computadora

Descripción generada automáticamente

*b) Crea un fichero denominado* ***actores.txt***

*c) Haz un primer ‘commit’*

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

*d) Crea una rama denominada test.*

Captura de pantalla de computadora

Descripción generada automáticamente

*e) Cambia a la rama “test”*

Captura de pantalla de computadora

Descripción generada automáticamente

*f ) En la rama test crea un fichero denominado* ***actrices.txt.*** *Añade tres nombres de actrices y realiza un ‘commit’ en dicha rama.*

Captura de pantalla de computadora

Descripción generada automáticamente

*g) Haz una captura de pantalla del resultado del comando git log --graph --all*

*h) Cambia a la rama master.*

*i) Incorpora los cambios de la rama test a la rama master. Haz una captura de pantalla de los comandos que has utilizado y de su resultado.*

*j) Crea una segunda rama denominada test2. La rama test2 apunta al mismo ‘commit’ que la rama master en este momento.*

*k) En la rama master, añade una actriz al fichero* ***actrices.txt*** *y haz un ‘commit’.*

*l) Cambia a la rama test2.*

*m) En la rama test2, añade una actriz al fichero* ***actrices.txt*** *y haz otro ‘commit’.*

*n) Haz una captura de pantalla del resultado del comando git log --graph --all. Debe haber dos caminos distintos: uno para la rama master y otro para la rama test2.*

*ñ) Cambia a la rama máster*

*o) Incorpora los cambios de la rama test2 a la rama master. ¿Se produce un conflicto? De ser así realiza una captura del comando git status.*

*p) Resuelve el conflicto incorporando los dos nombres de actrices.*

*q) Haz una captura de pantalla del resultado del comando git log --graph --all. Observa que se ha creado un nuevo ‘commit’ que integra los dos caminos anteriores.*

**2. Crea una carpeta denominada S2R2-remoto. Inicializa un repositorio Git en su interior mediante el comando git init --bare. Esta carpeta se utilizará como repositorio remoto.**

**3. Clona el repositorio S2R2-remoto en una carpeta denominada S2R2. Adjunta captura de pantalla del resultado del comando de clonado. A continuación, realiza las siguientes acciones en el repositorio S2R2.**

*a) Crea un archivo denominado* ***directores.txt.*** *Añade el nombre de tres directores de cine.*

*b) Haz un commit.*

*c) Realiza un push al repositorio remoto. Adjunta captura de pantalla del resultado.*

*d) Crea una rama denominada version1.*

*e) Cambia a la rama version1.*

*f) En la rama version1 añade el nombre de dos directores de cine más al archivo directores.txt y haz un commit de los cambios.*

*g) Realiza un push de la rama al repositorio remoto de manera que quede asociada a la rama remota del mismo nombre. Adjunta captura de pantalla del resultado.*

**4. Clona el repositorio S2R2-remoto en una segunda carpeta denominada S2R3. Realiza las siguientes acciones sobre ella.**

*a) Muestra en la consola el contenido del fichero* ***directores.txt*** *y el resultado del comando git status. Debe mostrar tres directores.*

*b) Cambia a la rama versión1. Muestra el resultado del comando. Comprueba que se crea una rama local versión1 con el contenido de la rama remota origin/versón1 y enlazada con ella. Al clonar el repositorio la rama no existía (solo se clona la rama principal, master), pero al cambiar a una rama que existe en el remoto se produce su creación local y enlazado con su correspondiente remota.*

*c) Muestra el contenido del fichero* ***directores.txt*** *por la pantalla. Comprueba que se muestran los 5 nombres de directores esperados. Adjunta captura de pantalla.*

*d) Cambia a la rama master.*

*e) Incorpora los cambios de la rama versión1 a la rama master.*

*f) Sube la rama master actualizada al servidor. Adjunta captura de pantalla del resultado del comando.*

**5. Vuelve de nuevo a la carpeta S2R2 y realiza las siguientes acciones.**

*a) Obtén los cambios que hay en el repositorio remoto sin fusionarlos en la rama local. Adjunta captura de pantalla del resultado del comando utilizado.*

*b) Actualiza la rama master local con el contenido de la rama master del repositorio remoto. Adjunta captura de pantalla del resultado del comando utilizado.*

*c) Comprueba que aparecen los 5 nombres de directores esperados.*