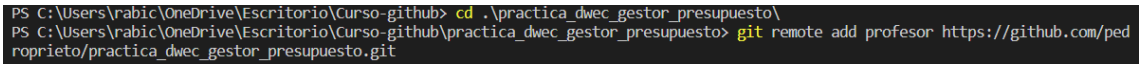
Preparación a las prácticas

Preparación

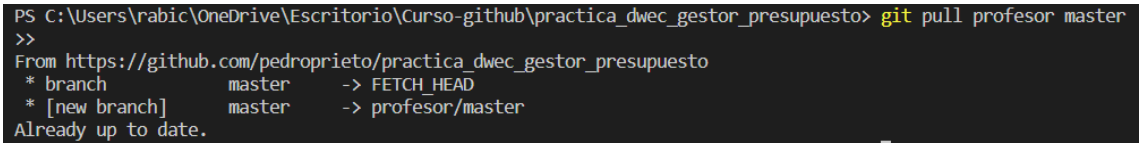
1. Instalar los requisitos de software indicados
2. Abrir un terminal
3. Situarse en la carpeta del repositorio personal de la práctica
4. **Añadir el remoto secundario** a tu repositorio personal. Para ello hay que ejecutar el comando:

   ```
   git remote add profesor
   https://github.com/pedroprieto/practica_dwec_gestor_presupuesto.git
   ```

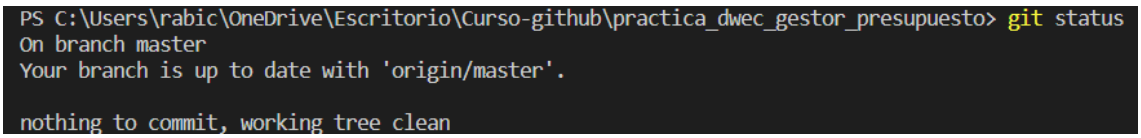
   
   ```
   PS C:\Users\rabic\OneDrive\Escritorio\Curso-github> cd .\practica_dwec_gestor_presupuesto\
   PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git remote add profesor https://github.com/ped
   roprieto/practica_dwec_gestor_presupuesto.git
   ```

5. **Incorporar a tu repositorio personal los cambios** realizados por el profesor correspondientes a los archivos de esta práctica. Para ello hay que ejecutar:

   ```
   git pull profesor master
   ```

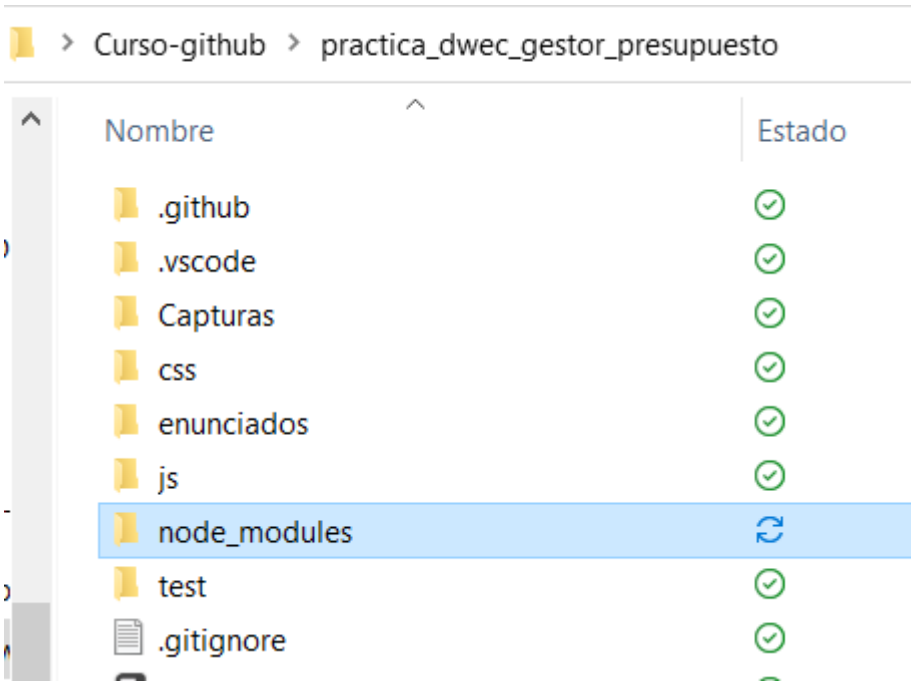
   
   ```
   PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git pull profesor master
   >>
   From https://github.com/pedroprieto/practica_dwec_gestor_presupuesto
    * branch            master     -> FETCH_HEAD
    * [new branch]      master     -> profesor/master
   Already up to date.
   ```

6. Este comando **descarga** los cambios que ha realizado el **profesor** en el **repositorio base** y los **integra** en tu repositorio personal. Tras realizar este paso, seguramente git **abra el editor configurado por defecto** para que introduzcas un **mensaje para crear un nuevo commit** que integre tus cambios y los cambios del profesor. Debes introducir el texto y guardar los cambios.

   
   ```
   PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> git status
   On branch master
   Your branch is up to date with 'origin/master'.

   nothing to commit, working tree clean
   ```

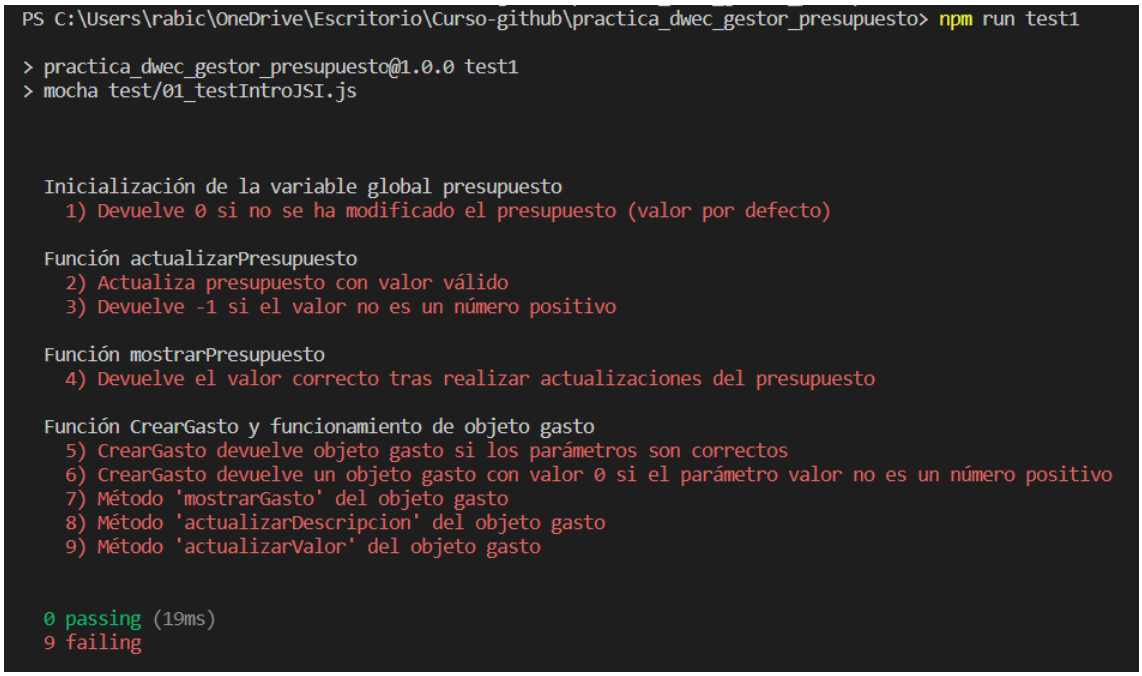
7. En principio no deben producirse **conflictos**. En caso de que se produzcan (por ejemplo, porque has editado el fichero `.gitignore` y yo también porque lo exigía la práctica), **resuélvelos y notifícamelo a través de un Issue**.

8. Ejecuta el comando `git push` para subir los cambios a tu repositorio personal (el remoto principal) en GitHub y que queden guardados ahí también.
9. Ejecutar el comando `npm install`. Este comando instalará todas las librerías de Node necesarias para ejecutar los tests. Dichas librerías se guardarán en una carpeta con nombre `node_modules` dentro del repositorio. Nótese que dicha carpeta está excluida del repositorio en el archivo `.gitignore`.




```
PS C:\Users\rabic\OneDrive\Escritorio\Curso-github\practica_dwec_gestor_presupuesto> npm install
d.
npm WARN deprecated har-validator@5.1.5: this library is no longer supported

added 224 packages, and audited 225 packages in 2m

36 packages are looking for funding
  run `npm fund` for details

6 vulnerabilities (2 moderate, 1 high, 3 critical)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.15.0 -> 8.19.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.19.2
npm notice Run npm install -g npm@8.19.2 to update!
npm notice
```

10. Ejecutar el comando `npm run test1` para lanzar los **tests de esta práctica**. Este comando podrá ejecutarse tantas veces como sea necesario. Por pantalla se mostrarán los tests que se pasan y los que no, de manera que se tendrá información sobre las acciones que hay que realizar. Los tests también se ejecutarán automáticamente en GitHub Actions al subir los cambios al repositorio y al realizar la pull request.



11. Si se está utilizando VS Code se puede lanzar la ejecución de los tests desde el propio editor y utilizar **breakpoints** para interrumpir la ejecución y **depurar** el programa. Las configuraciones de lanzamiento creadas para ello están en el archivo `.vscode/launch.json`.
12. Opcionalmente (recomendable), ejecutar el comando `npm test` para lanzar **todos los tests presentes en el repositorio**. Se deberá comprobar que se pasan los tests de las prácticas anteriores a la que se esté realizando. Lógicamente, si el repositorio incluye los tests de prácticas **posteriores** a la que se esté realizando, dichos tests no se pasarán (ya que el trabajo está todavía por hacer). Este último caso puede darse si la persona no está realizando la práctica propuesta en la semana actual (va con "retraso", por así decirlo). En **GitHub Actions** se ejecutarán todos los tests en tareas independientes: así se podrá comprobar si el test de la práctica que se está realizando se ha pasado.