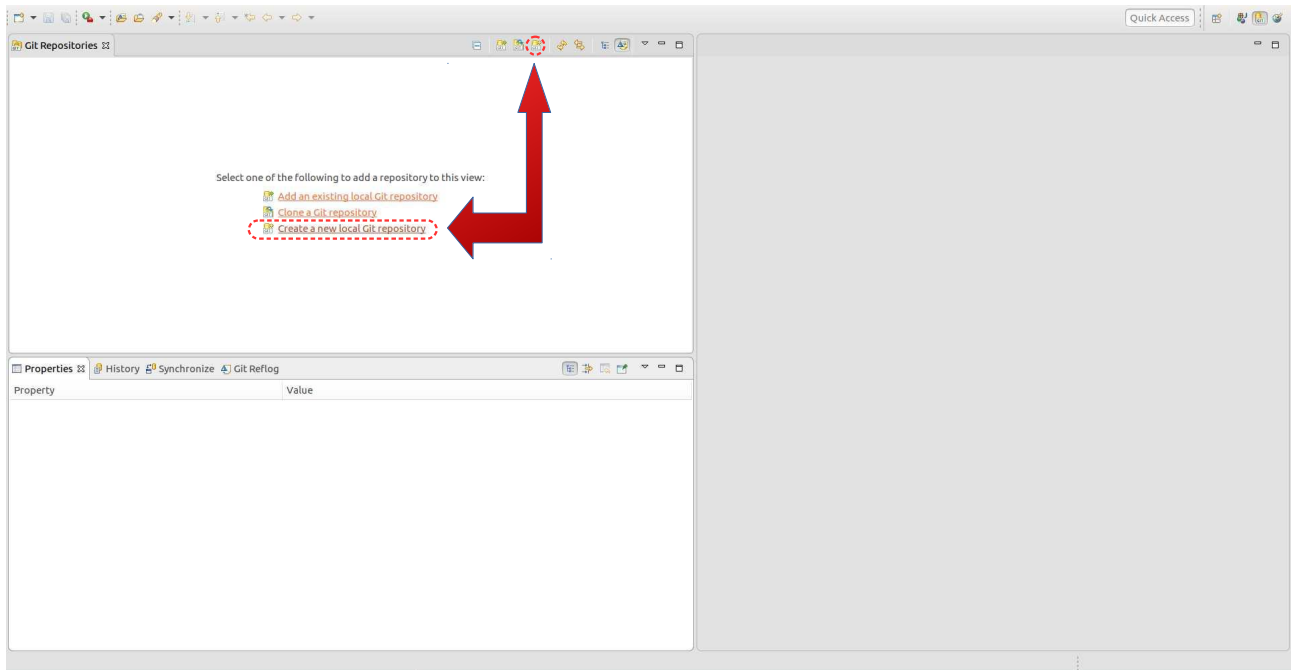


Cómo utilizar GitHub con Eclipse

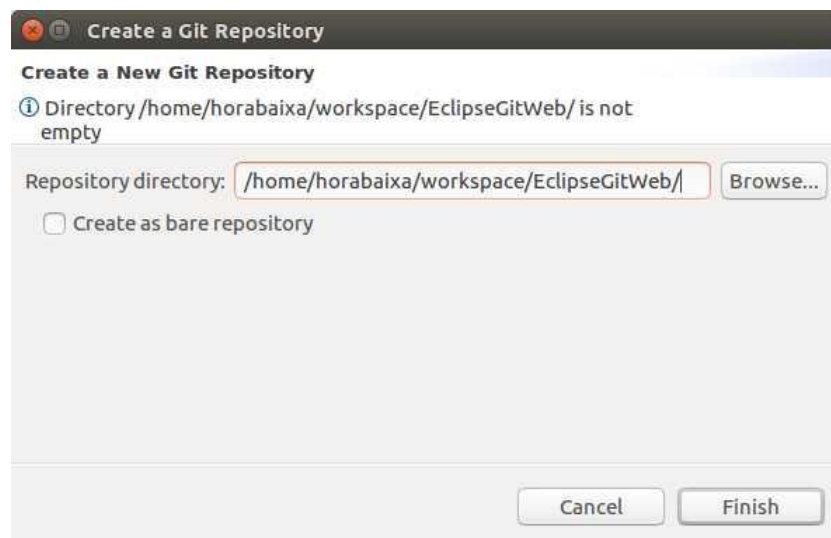
Eclipse Neon Rev: 2.4.6.2

(Es posible que las imágenes sean ligeramente diferentes a su versión)

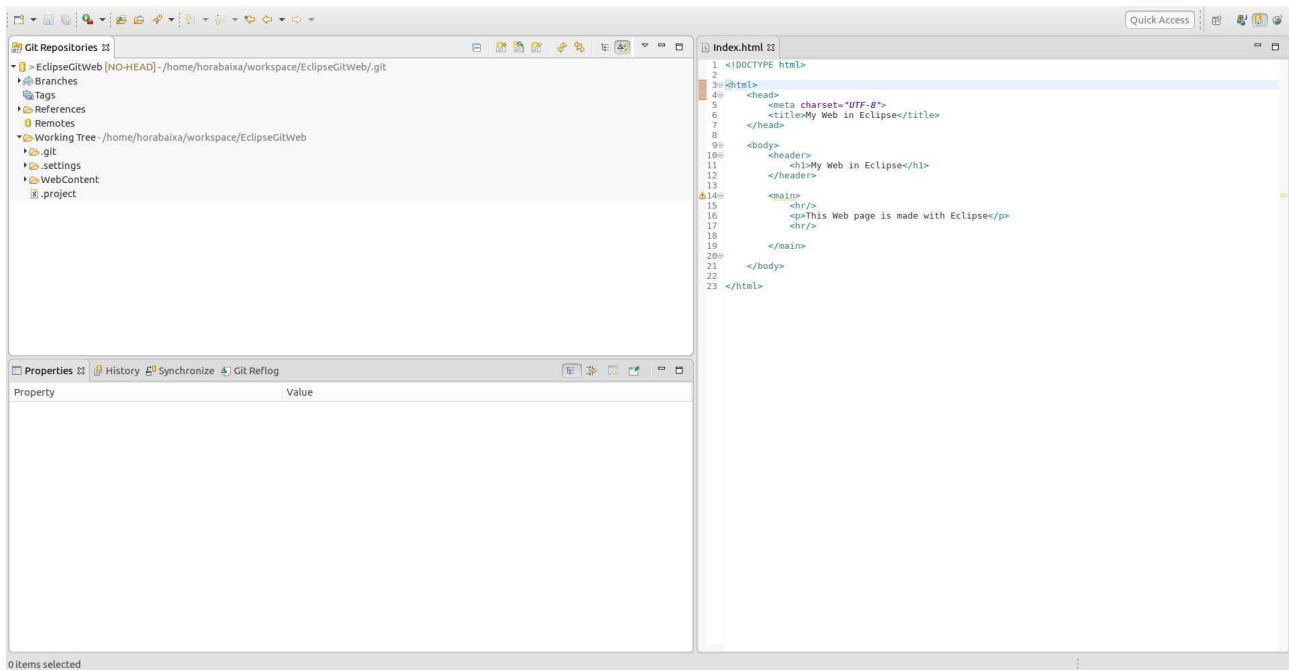
Empezamos creando un repositorio local desde Eclipse Neon desde el editor GiT



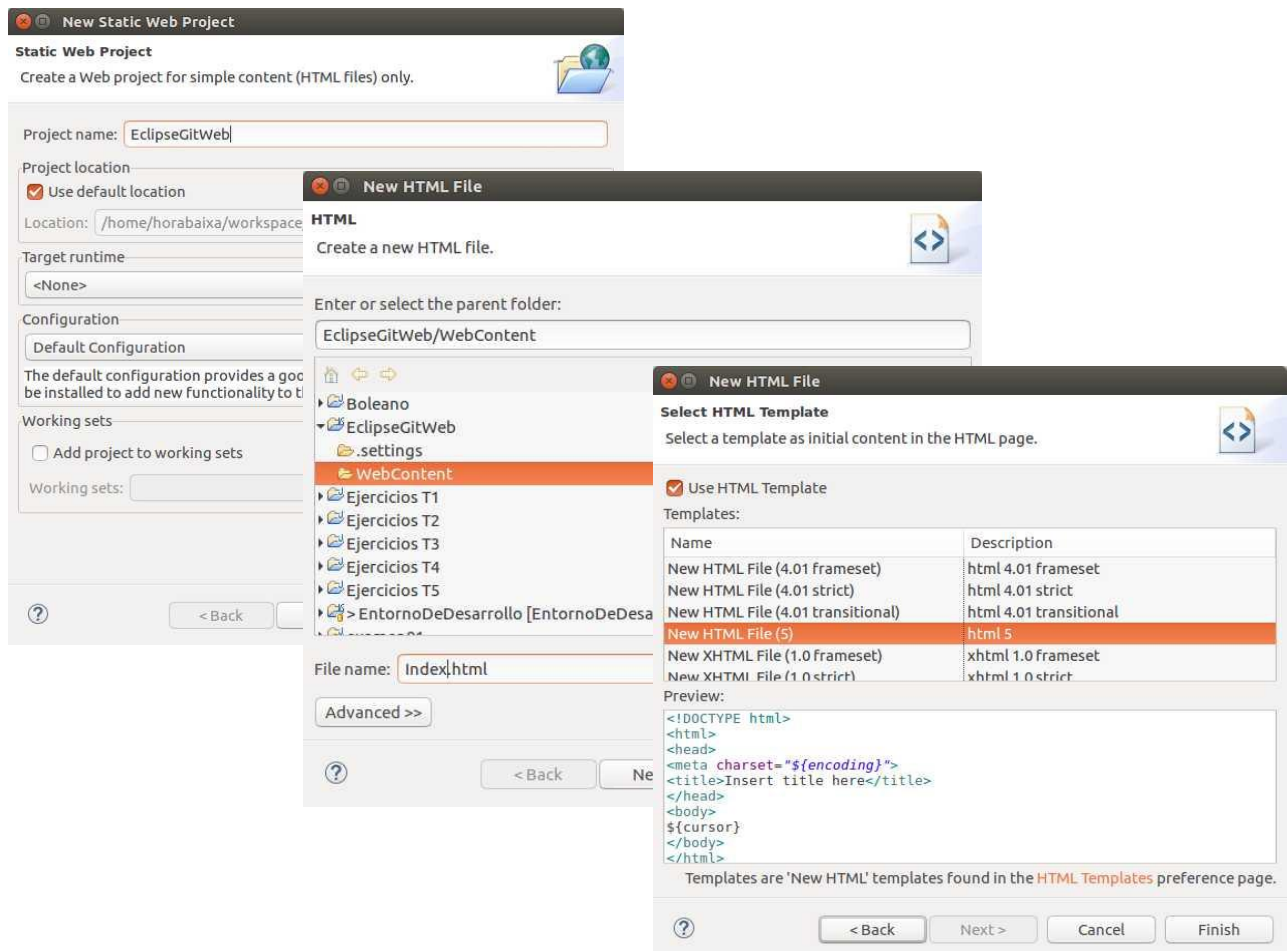
Elegimos un nombre y ruta para nuestro repositorio (Hay que acordarse del nombre para luego evitar problemas con el GitHub “WEB”).



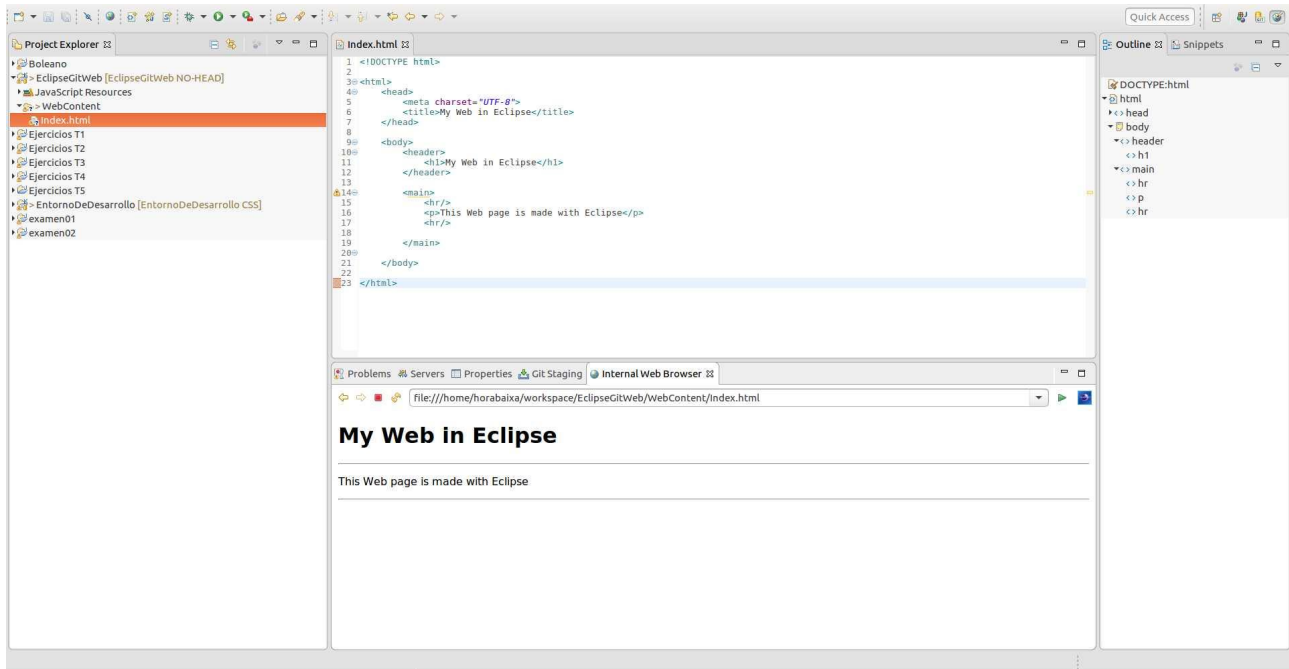
Una vez creado el directorio con su respectivo nombre nos quedara algo como esto:



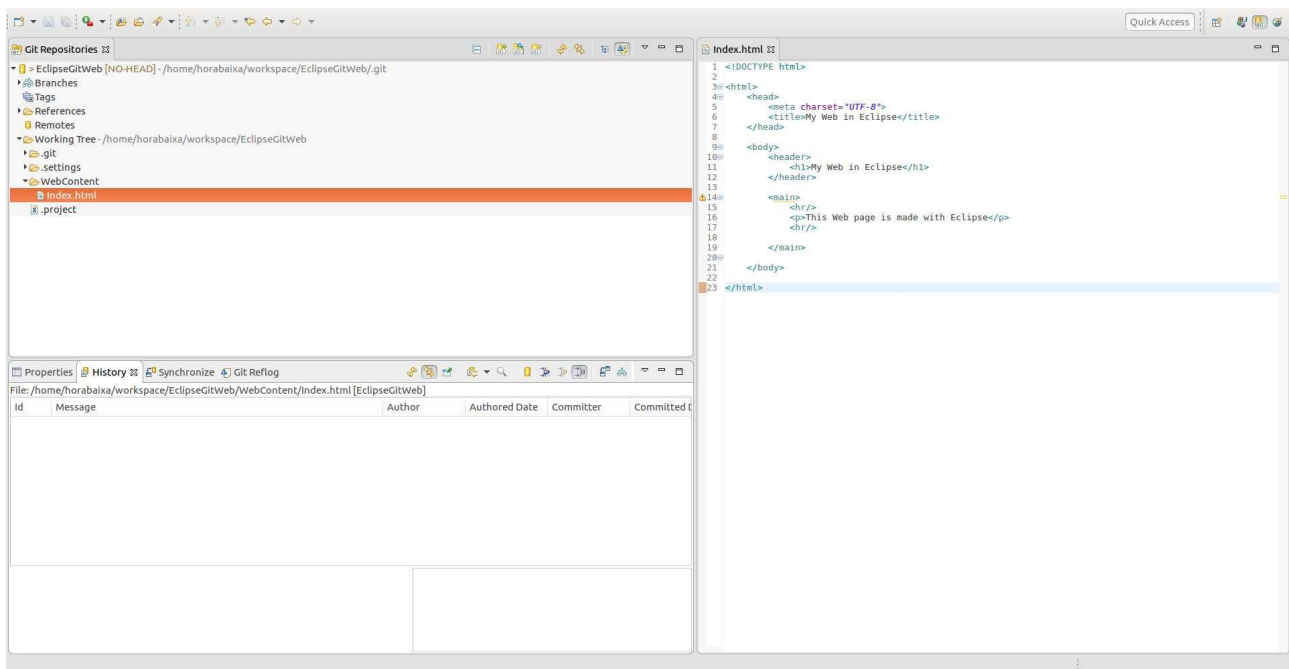
Ahora nos vamos al modo WEB de Eclipse y creamos un nuevo proyecto en la ruta de Git local que hemos creado antes.



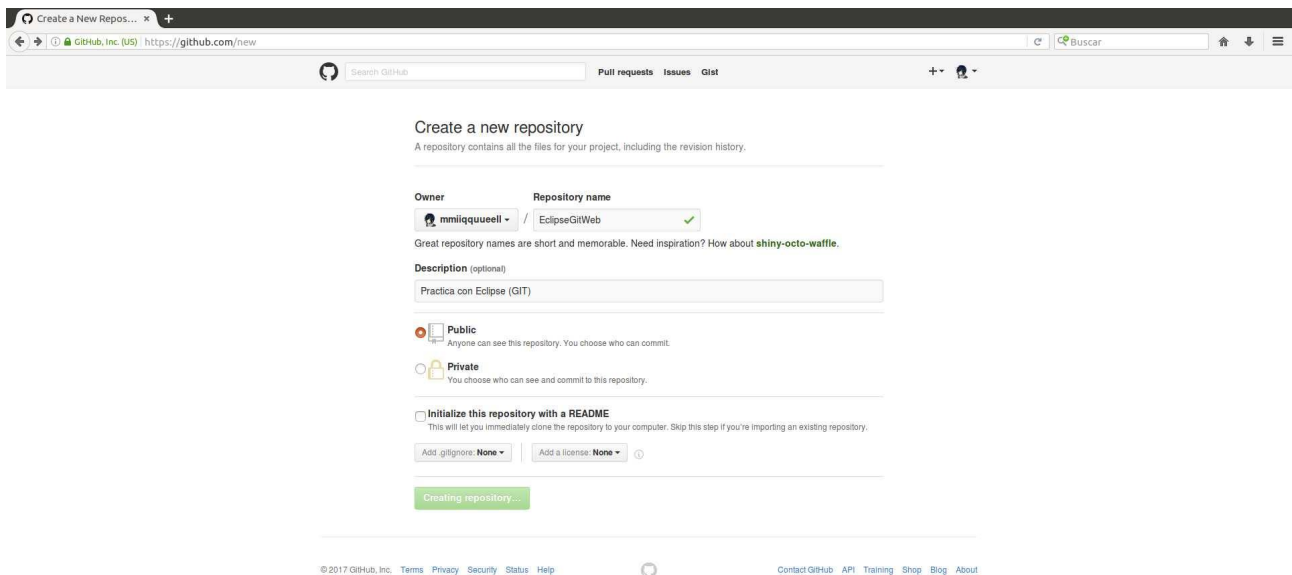
Creamos nuestro documento HTML (para este ejemplo hemos puesto un par de etiquetas)



Ahora nos vamos al modo GIT

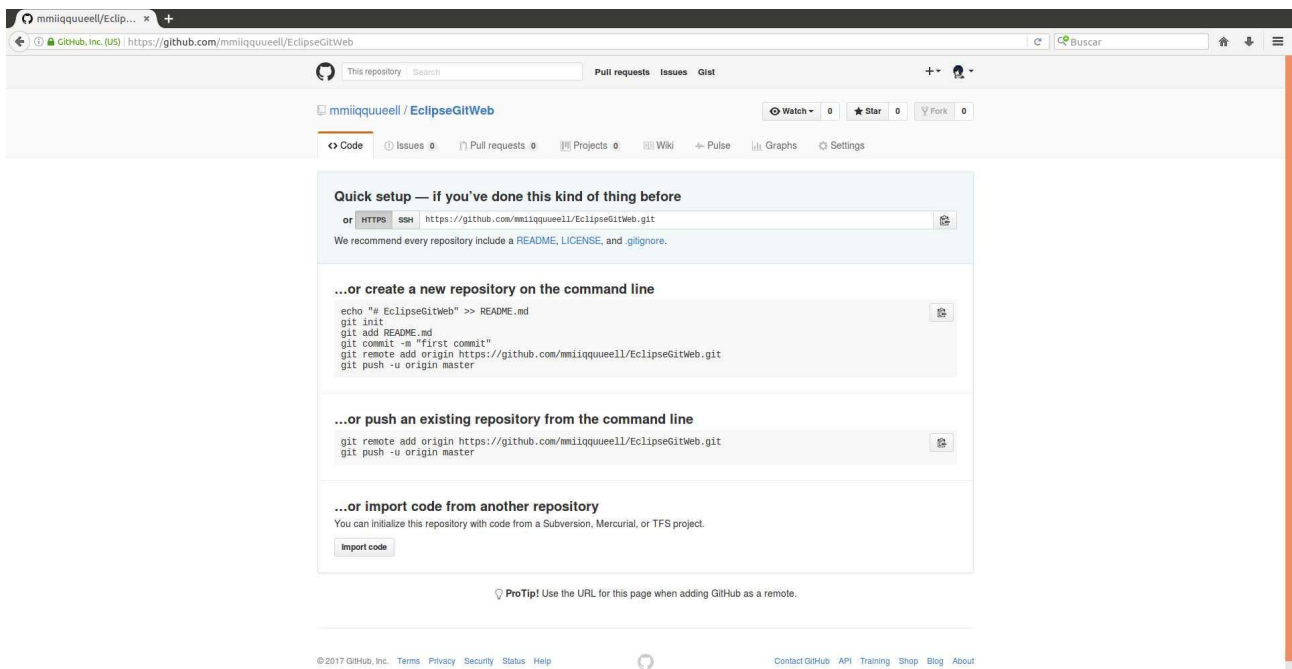


Nos vamos al navegador web y entramos a GitHub con nuestra cuenta y creamos un repositorio (como se indico anteriormente utilizamos el mismo nombre que el proyecto WEB)



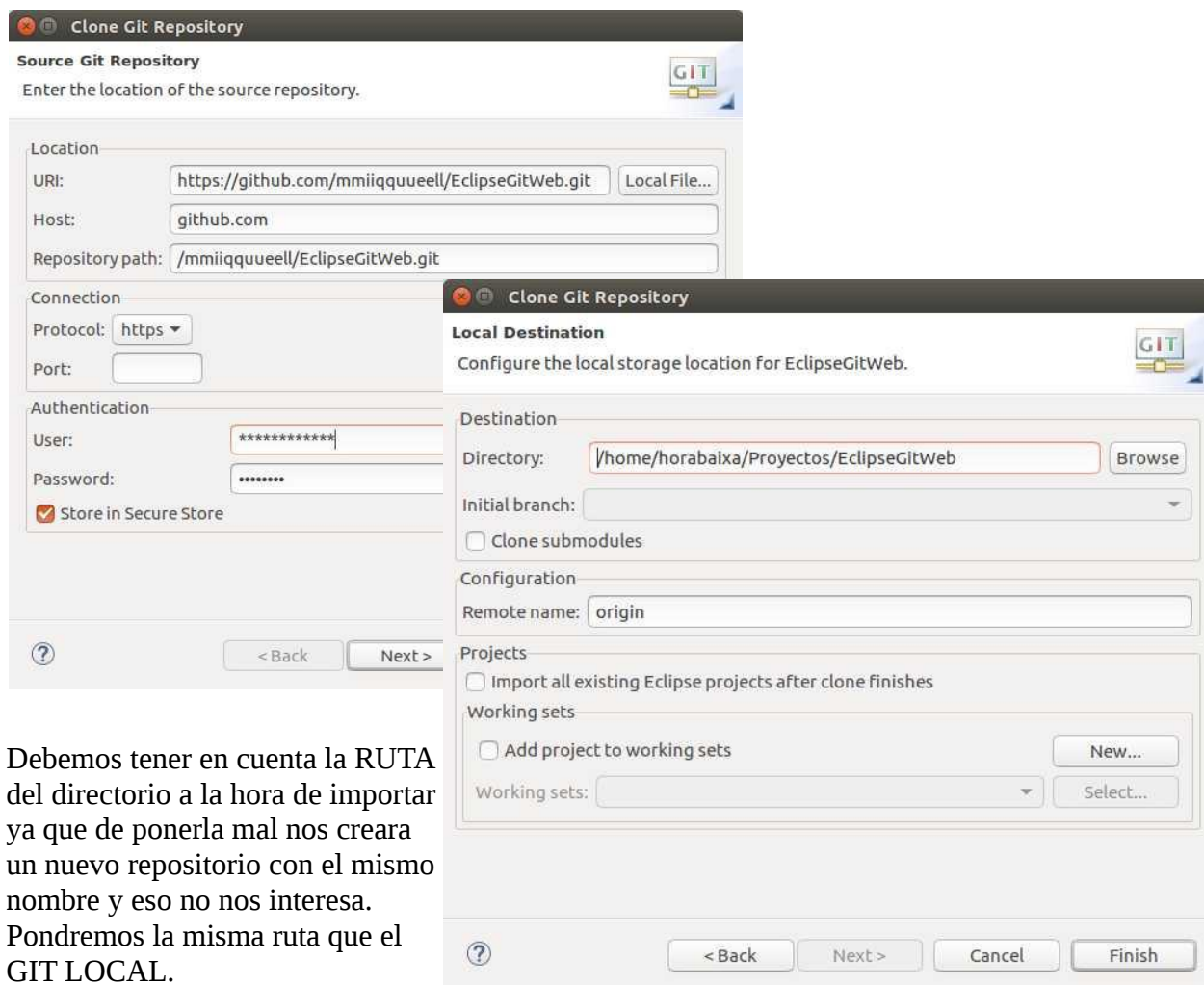
The screenshot shows the GitHub 'Create a new repository' page. The browser address bar shows 'https://github.com/new'. The page title is 'Create a new repository'. Below the title, it says 'A repository contains all the files for your project, including the revision history.' The form includes fields for 'Owner' (mmiqueell) and 'Repository name' (EclipseGitWeb). There is a 'Description (optional)' field with the text 'Practica con Eclipse (GIT)'. The 'Visibility' section has 'Public' selected. There is a checkbox for 'Initialize this repository with a README'. At the bottom, there are buttons for 'Add .gitignore: None' and 'Add a license: None'. A green button at the bottom says 'Creating repository...'. The footer shows '© 2017 GitHub, Inc.' and various links like 'Terms', 'Privacy', 'Security', 'Status', 'Help', 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', 'About'.

Tan solo debemos copiar la URL del repositorio acabado en *.GIT y añadirlo al proyecto de Eclipse



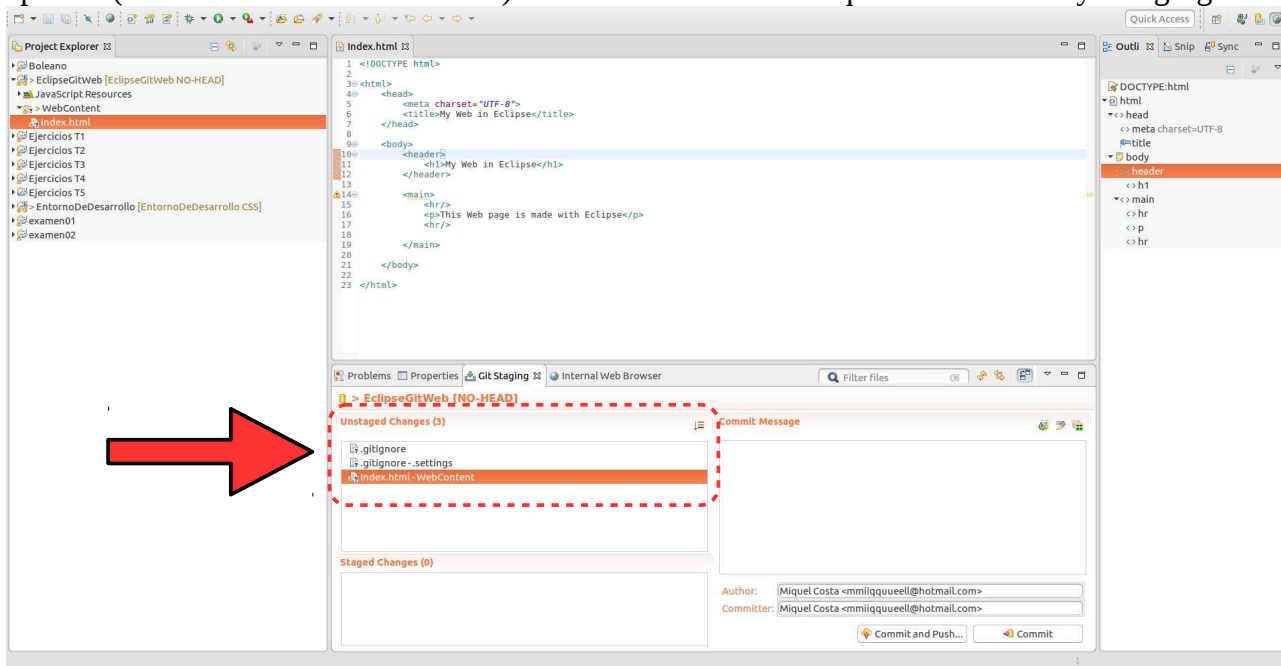
The screenshot shows the GitHub repository page for 'mmiqueell/EclipseGitWeb'. The browser address bar shows 'https://github.com/mmiqueell/EclipseGitWeb'. The page title is 'mmiqueell / EclipseGitWeb'. Below the title, there are buttons for 'Watch', 'Star', and 'Fork'. The 'Code' tab is selected, showing the 'Quick setup' section. This section includes the repository URL 'https://github.com/mmiqueell/EclipseGitWeb.git' and instructions for cloning the repository using the command line. The instructions are: 'echo "# EclipseGitWeb" >> README.md', 'git init', 'git add README.md', 'git commit -m "first commit"', 'git remote add origin https://github.com/mmiqueell/EclipseGitWeb.git', and 'git push -u origin master'. There is also a section for '...or push an existing repository from the command line' with the same commands. At the bottom, there is a 'ProTip!' section with the text 'Use the URL for this page when adding GitHub as a remote.'

De vuelta al Eclipse y con la URL en el portapapeles del sistema vamos a añadir un repositorio existente (Eclipse pone “CLONE GIT” cuando en realidad estamos añadiendo uno que ya existe).

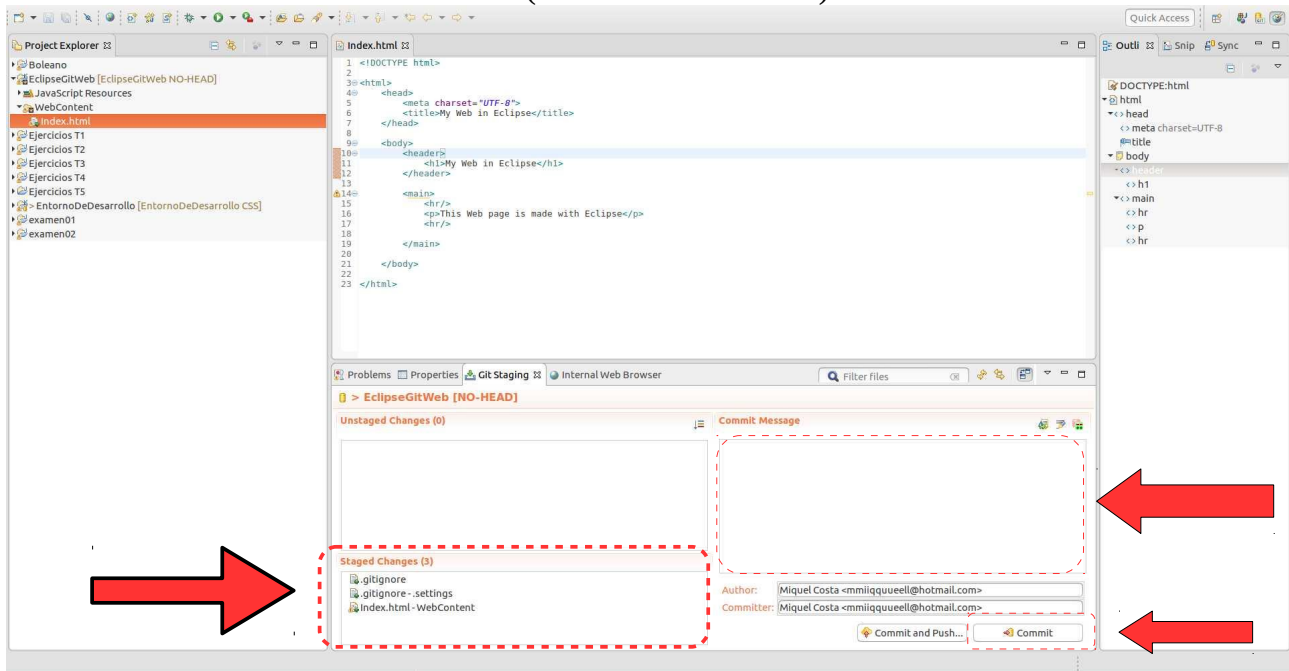


Debemos tener en cuenta la RUTA del directorio a la hora de importar ya que de ponerla mal nos creara un nuevo repositorio con el mismo nombre y eso no nos interesa. Pondremos la misma ruta que el GIT LOCAL.

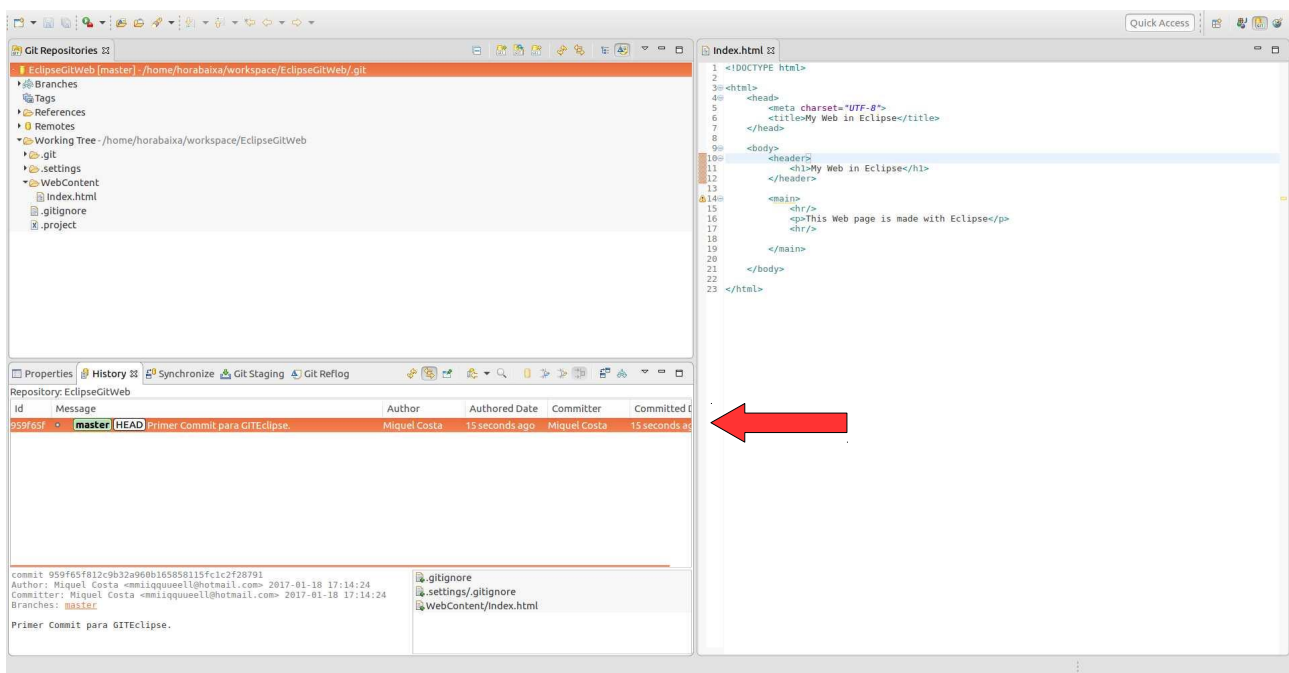
Automáticamente Eclipse detecta los 2 proyectos de GIT y los une en uno solo y ademas ya nos aparece (UNSTAGED CHANGES = 3) entre ellos el index.html que hemos creado y los .gitignore.



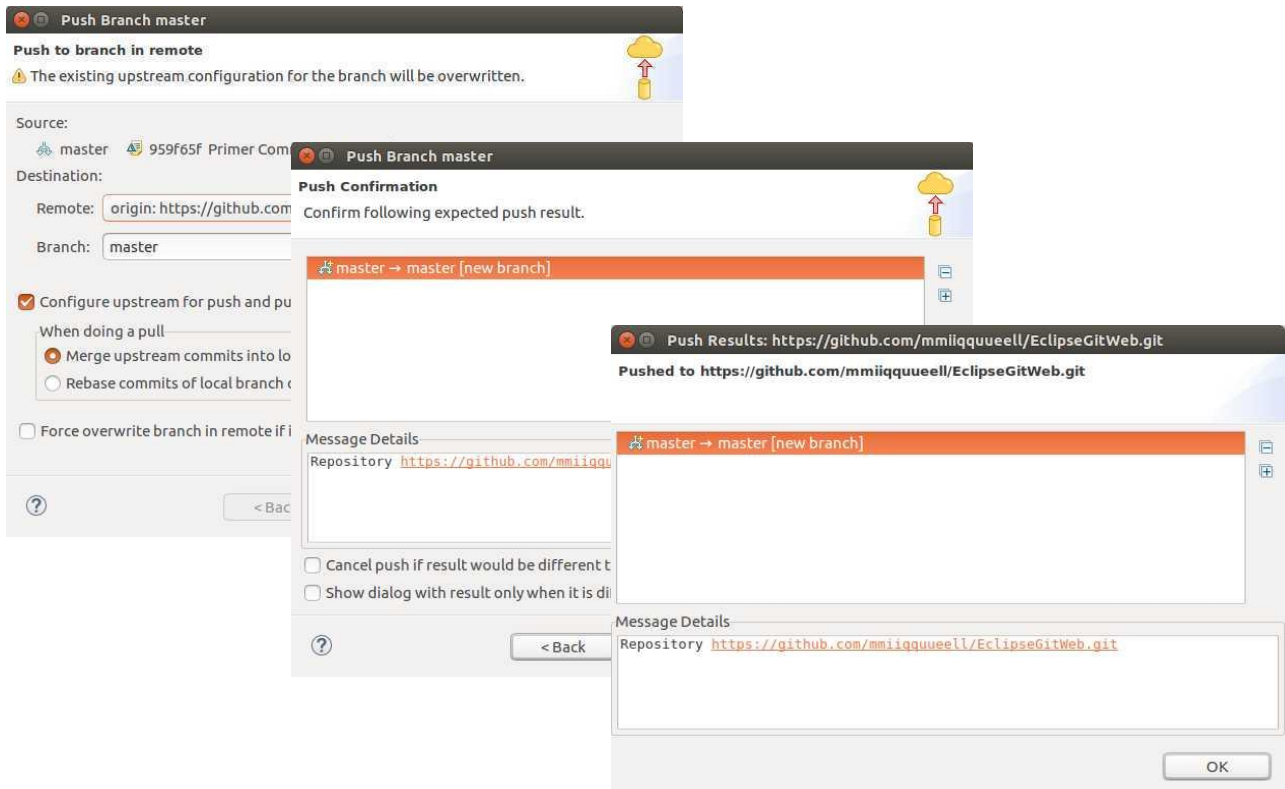
Tan solo debemos añadir los archivos al (STAGED CHANGES) como si fuera un ADD <archivo>



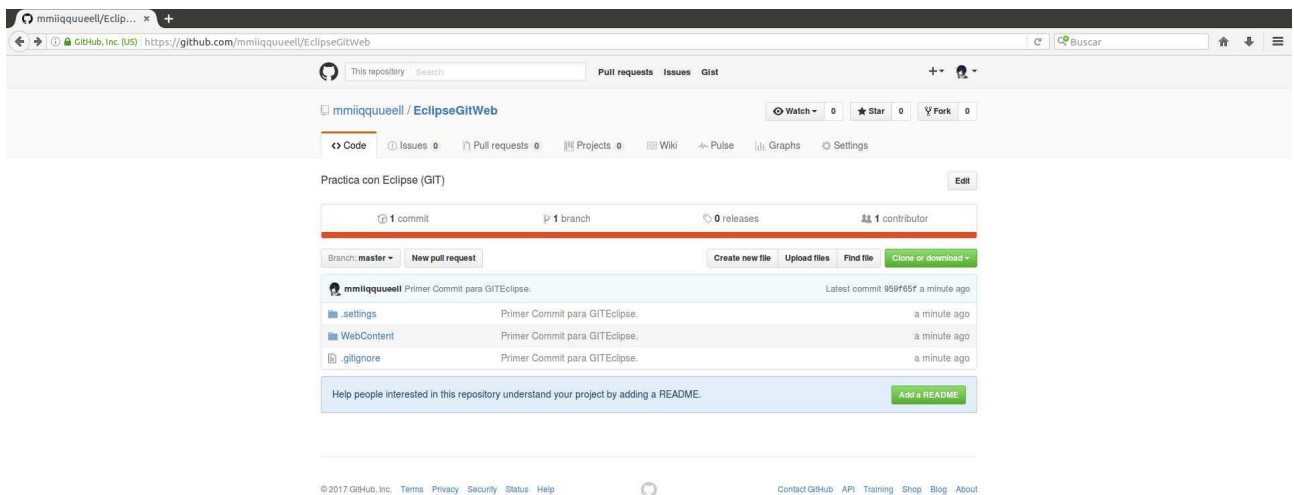
Entonces tras realizar esto podemos hacer un COMMIT y ya tenemos nuestra rama y su commit.



Ahora solo falta hacer un PUSH. (nos pedira usuario y contraseña)

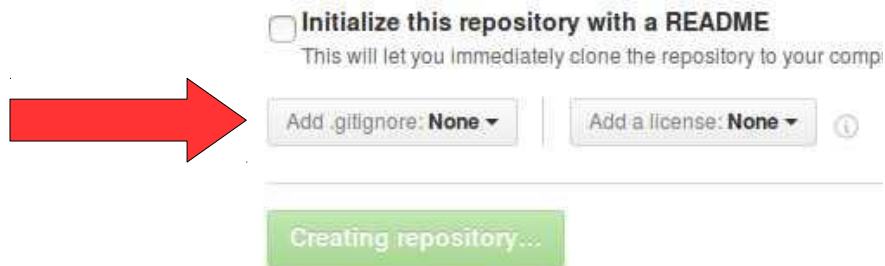


Finalmente ya tenemos realizado el Push y ahora nuestro archivo HTML ya esta en el repositorio.



.GITIGNORE: El archivo .gitIgnore es un archivo que ignora determinados archivos. Debido a que Eclipse genera archivos propios es una buena idea ignorarlos. Hay varias formas de ignorar dichos archivos.

1. Al crear un repertorio en GitHub podemos marcar la opción de crear un .gitIgnore para una aplicación especificada (Lamentablemente Eclipse no está disponible).



2. Buscando por internet una página que contenga los archivos que nos interesa ignorar, en este caso los de Eclipse.
3. Dándole manualmente a “ignore” a los archivos que no nos interesa (los de Eclipse).

Tras eso solo es cuestión de hacer un commit del archivo y automáticamente Eclipse Git ignorará dichos archivos.