

5. Uso de Git en Netbeans

El entorno de desarrollo integrado o IDE Netbeans versión 8.2 tiene incluida una versión de Git. Para verificar que Git esté instalado se ejecuta Netbeans y se revisa la opción “Tools” -> “Options” → “Team”:

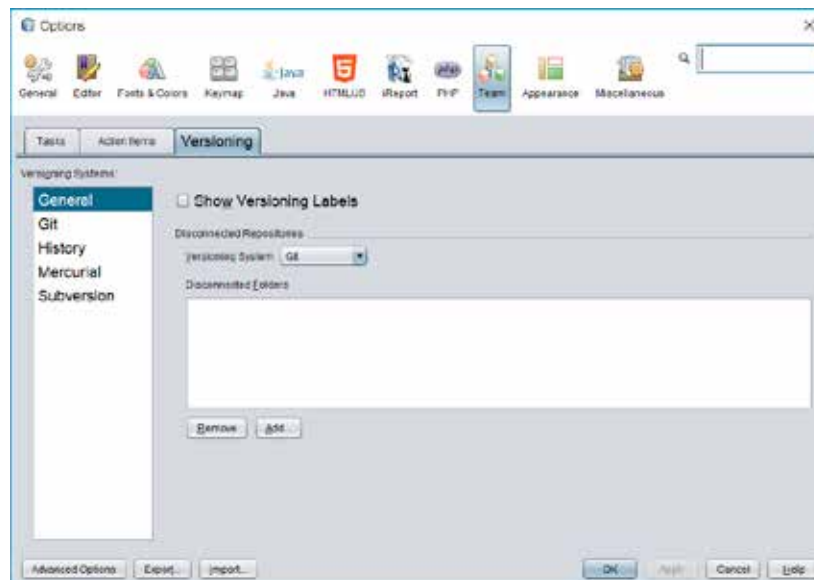


Figura 5.1. Verificación de la instalación de Git dentro de Netbeans.

Se puede observar que en la pestaña “Versioning” aparece Git dentro de los sistemas de versionamiento. Al dar clic sobre la opción Git del menú lateral aparece lo siguiente:

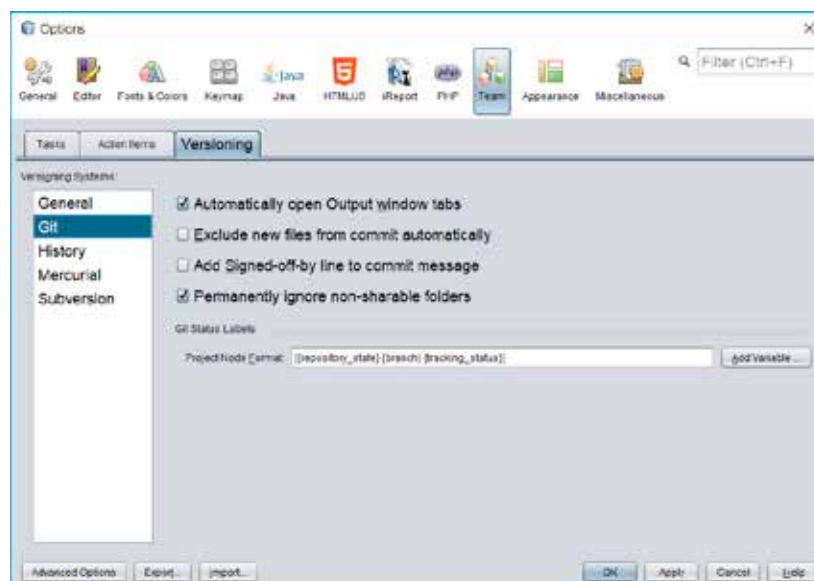


Figura 5.2. Verificación de la instalación de Git dentro de Netbeans.

5.1. Creación de una aplicación de pruebas

Para conocer la herramienta Git incluida en Netbeans se requiere crear una aplicación de pruebas. Para este ejercicio se creará una aplicación de tipo HTML5 con la ayuda de la mencionada herramienta. La aplicación se crea usando la opción “File” → “New project”. Una vez seleccionada aparece lo siguiente en pantalla:

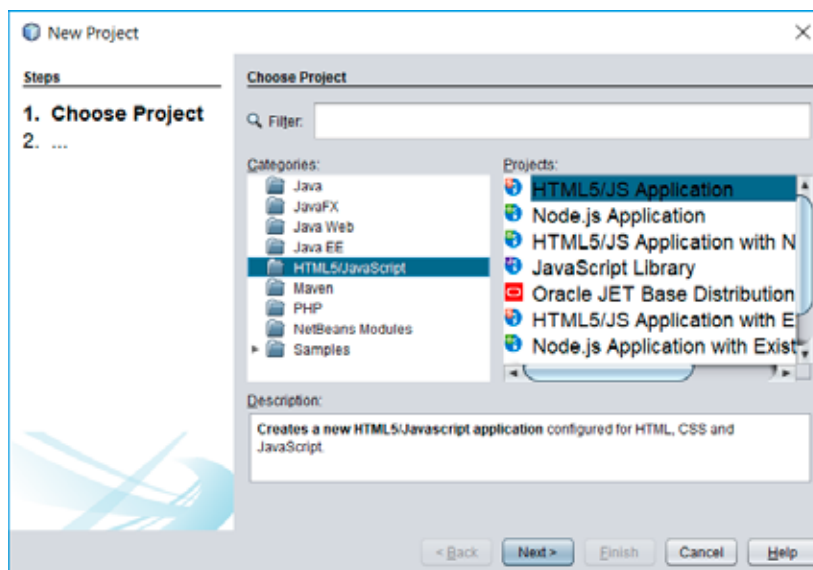


Figura 5.3. Creación de un proyecto HTML5 en Netbeans.

Se debe oprimir el botón “Next” y aparece lo siguiente en pantalla:

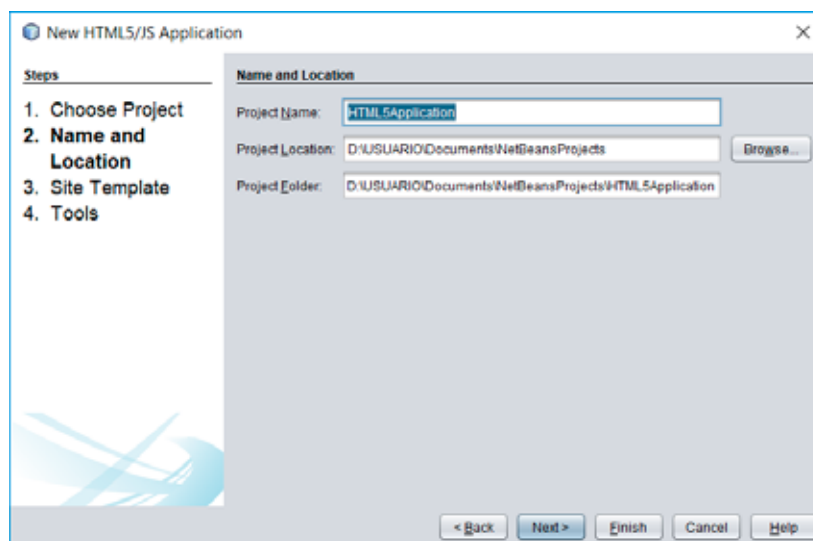


Figura 5.4. Creación de un proyecto HTML5 en Netbeans parte 2.

Se debe especificar un nombre de proyecto en el campo “Project Name”. Para este ejercicio el nombre será: “adsi_git”. Luego se oprime el botón “Next”. Aparece lo siguiente en pantalla:

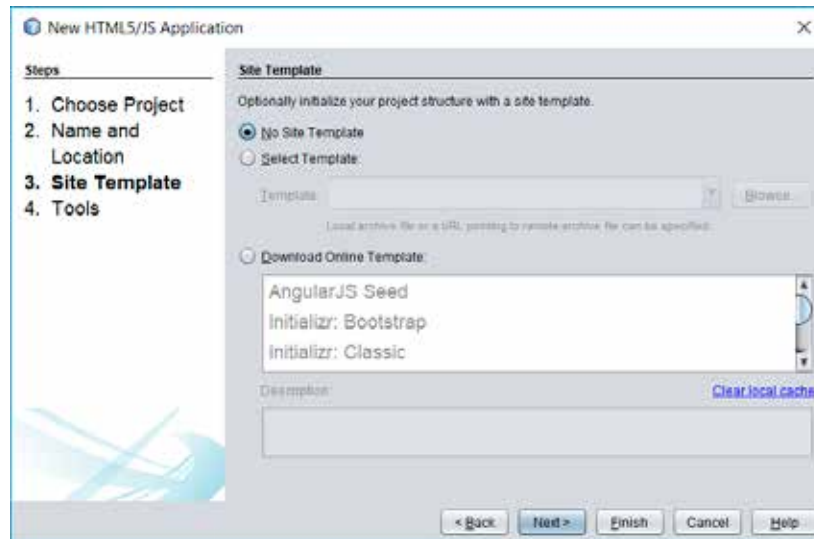


Figura 5.4. Creación de un proyecto HTML5 en Netbeans parte 3.

En esta pantalla no se selecciona ninguna opción y se oprime el botón “Next”. Aparece lo siguiente en pantalla:

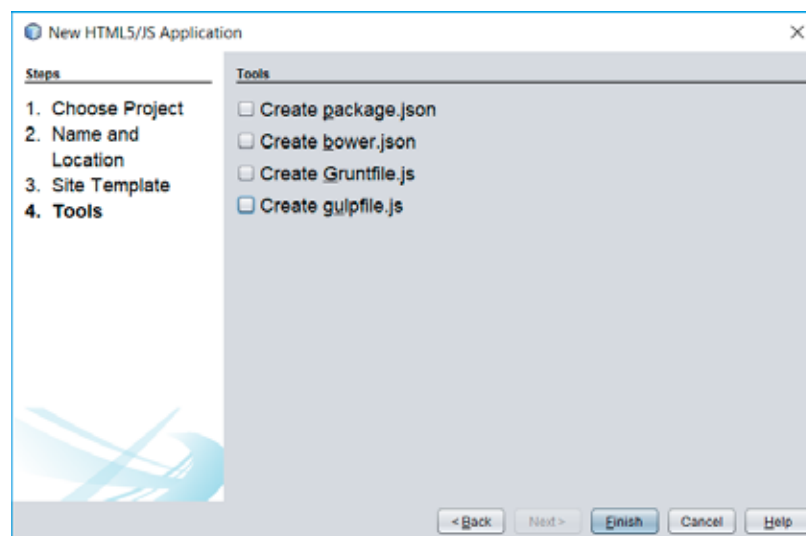


Figura 5.5. Creación de un proyecto HTML5 en Netbeans parte 4.

En esta pantalla se quitan las opciones seleccionadas por defecto. Por último se oprime el botón “Finish”. Aparece lo siguiente en pantalla:

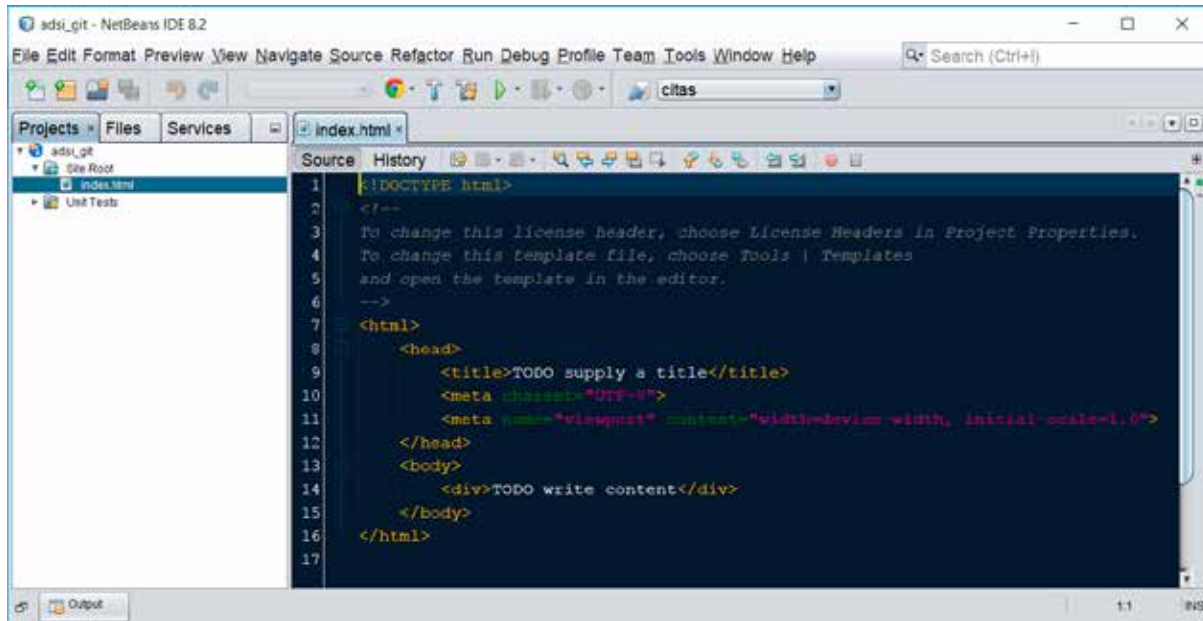


Figura 5.6. Creación de un proyecto HTML5 en Netbeans parte 5.

Al ejecutar el proyecto por la opción “Run” u oprimiendo la tecla F6 aparece lo siguiente en pantalla:

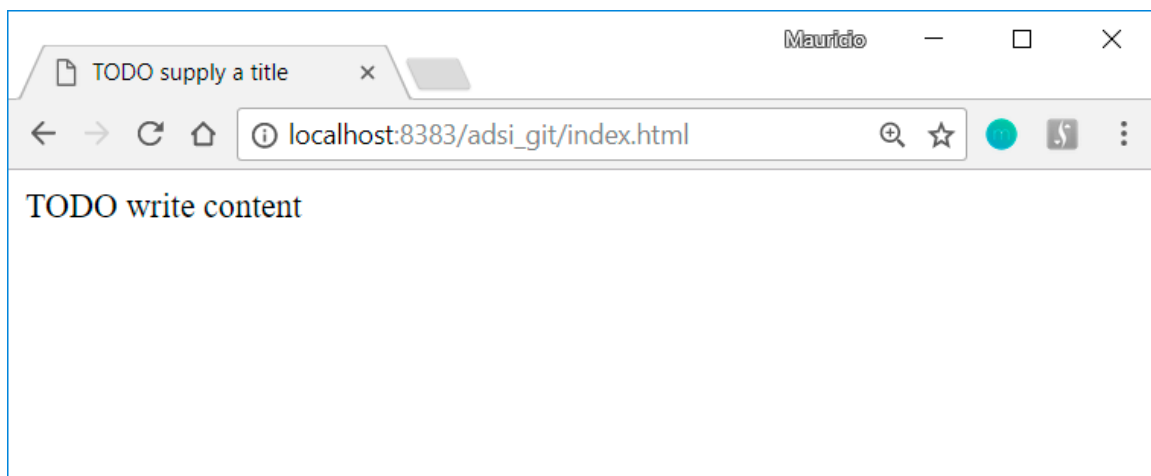


Figura 5.7. Creación de un proyecto HTML5 en Netbeans parte 6.

5.2. Creación del repositorio

Una vez creado el proyecto se inicia el repositorio con la opción del menú: “Team” → “Git” → “Initialize repository” como se muestra a continuación:

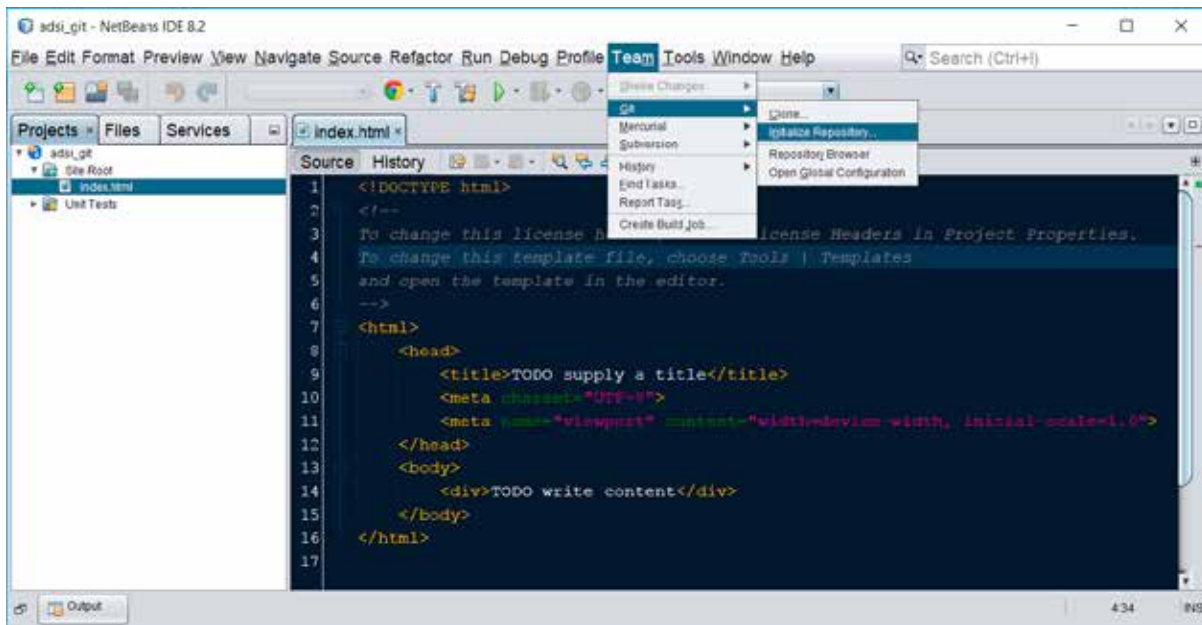


Figura 5.8. Iniciación del repositorio.

Una vez seleccionada la opción aparece lo siguiente en pantalla:

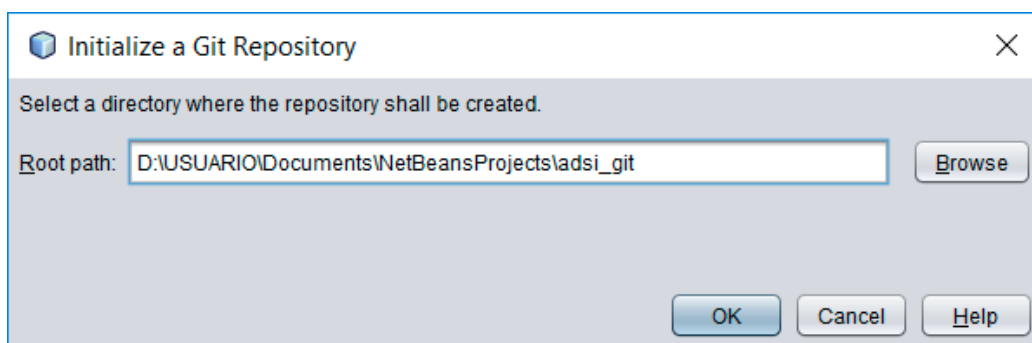


Figura 5.9. Iniciación del repositorio parte 2.

En este punto Netbeans indica el nombre del repositorio que por defecto es el mismo nombre del proyecto. Se debe oprimir el botón “Ok” para continuar. Aparecerá lo siguiente en la consola “Output”:

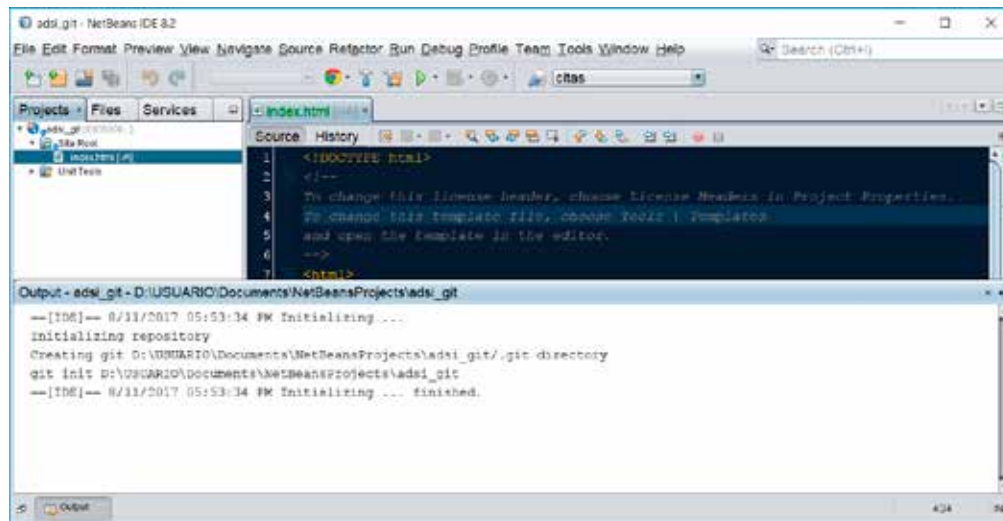


Figura 5.10. Iniciación del repositorio parte 3.

La ventana “Output” muestra los mensajes. En ellos se puede ver que el comando aplicado fue “git init” y el nombre del directorio solicitado. Por último, muestra que finalizó sin mostrar ningún error.

Se puede observar también que la pestaña que muestra el nombre del archivo “index.html” se encuentra de color verde. Lo anterior indica que el archivo está siendo controlado por Git.

5.3. Primera confirmación o commit

Se va a tomar este código como la versión uno del proyecto. Para indicar esto a Git se deben agregar los archivos con el comando “git add” que en esta versión integrada de Git se encuentra en “Team” → “Add” como se muestra a continuación:

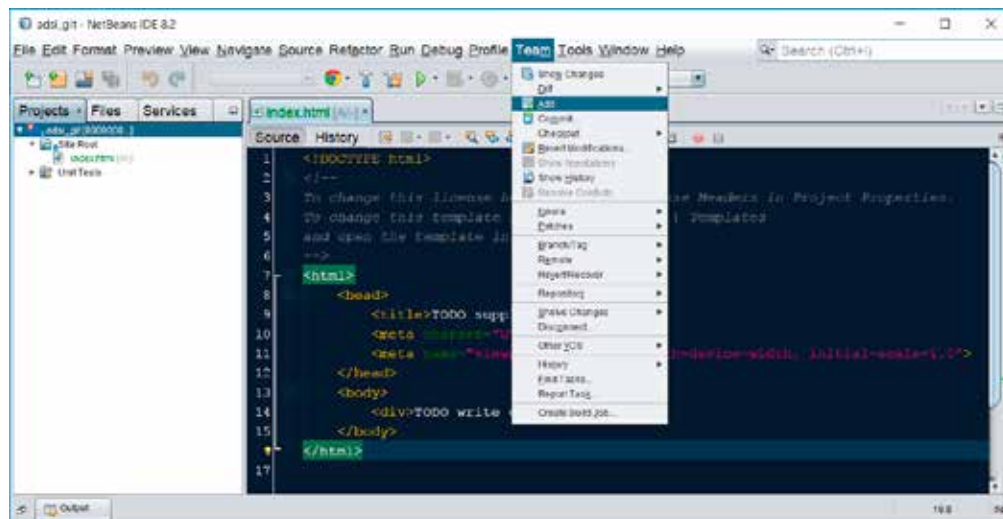


Figura 5.11. Adición de archivos al repositorio.

Una vez ejecutada la opción se puede observar en la ventana “output” los comandos Git usados por netbeans.

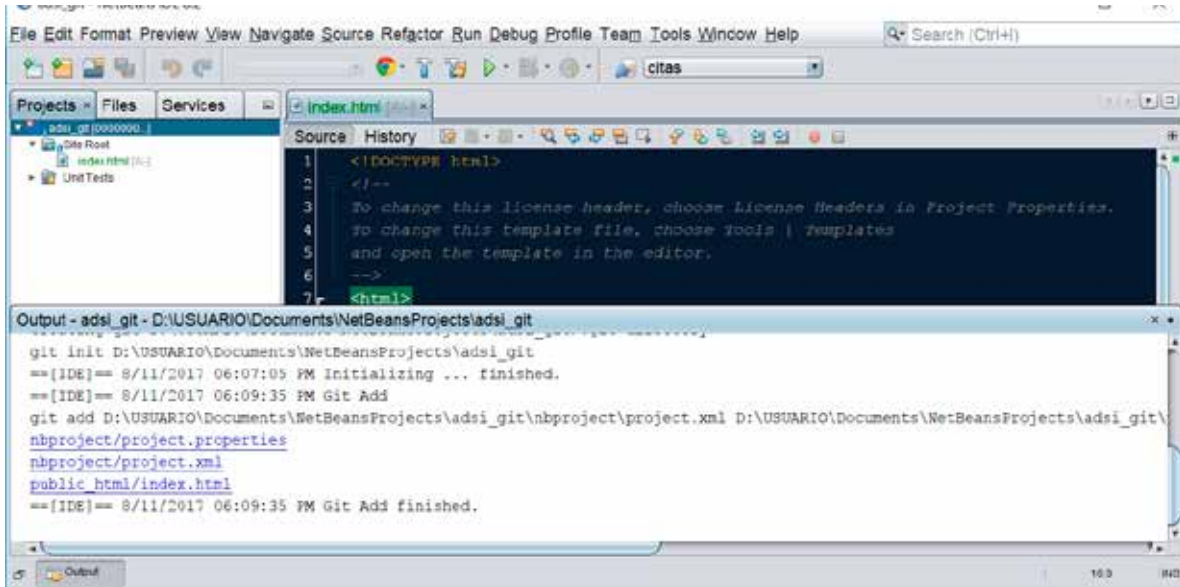


Figura 5.12 Adición de archivos al repositorio parte 2.

Se puede observar que el comando Git utilizado por Netbeans fue: “git add “. Este comando tomó como parámetros además del archivo “Index.html” otros usados internamente por Netbeans.

Luego se procede a confirmar los cambios. Para lo anterior se usa la opción del menú: “Team” → “Commit ...” aparece lo siguiente en pantalla:

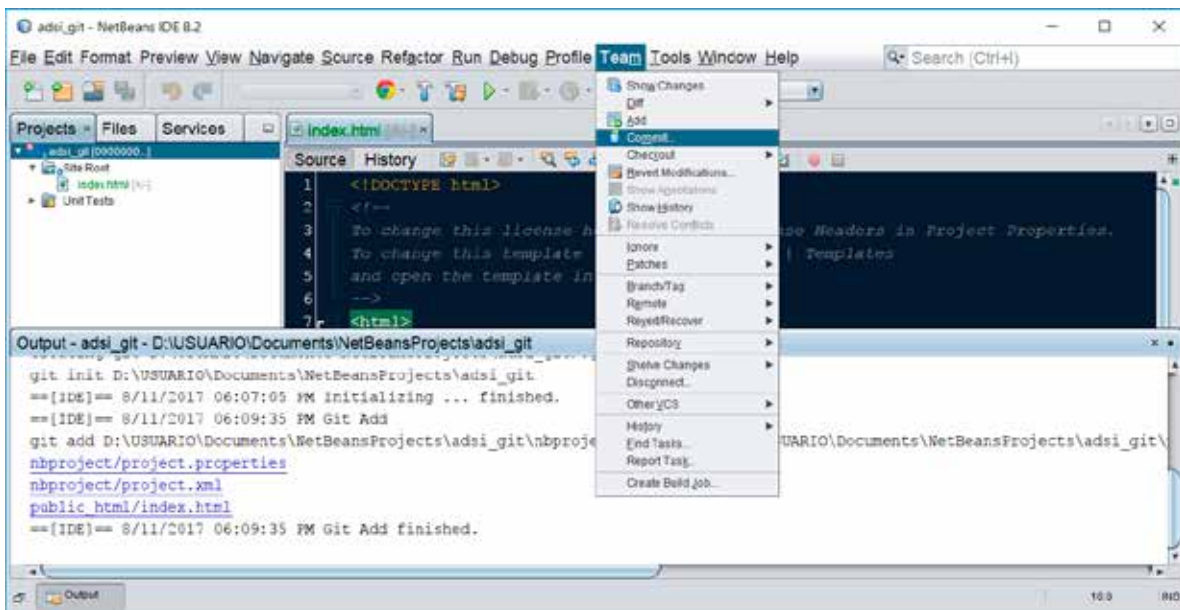


Figura 5.13 Primera confirmación de cambios o commit del proyecto.

Una vez seleccionada la opción aparece lo siguiente:

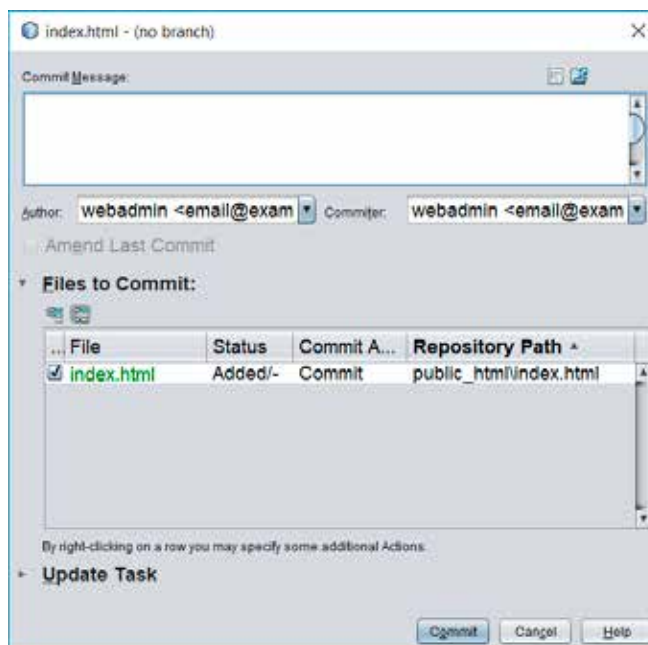


Figura 5.14 Primera confirmación de cambios o commit del proyecto parte 2.

En el espacio para el “Commit message” se ingresa el nombre de la primera versión que para este ejemplo se usará “versión_1.0” así:

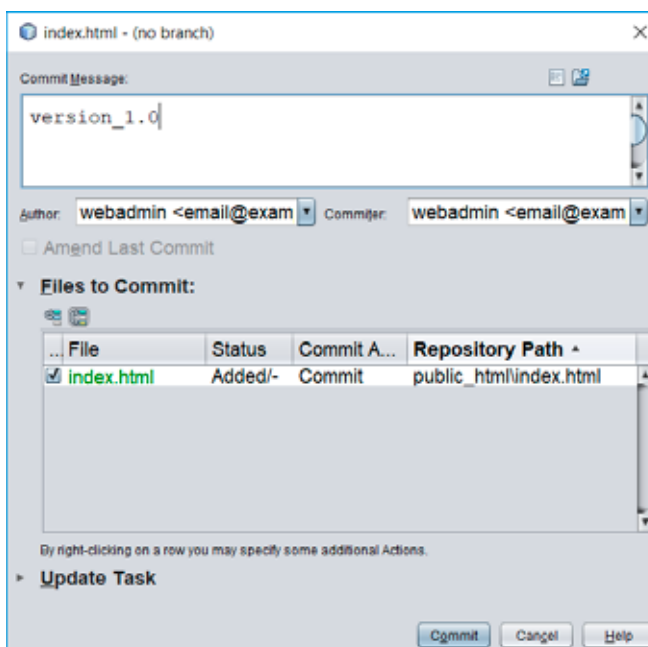
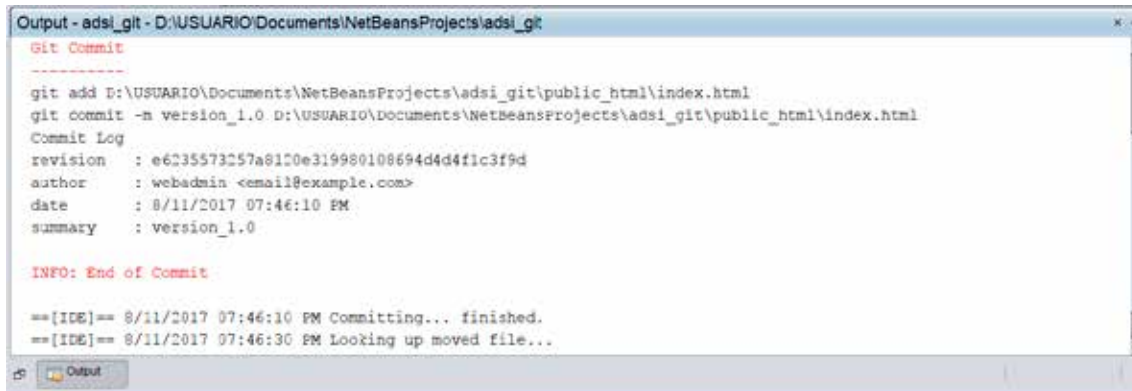


Figura 5.15. Primera confirmación de cambios o commit del proyecto parte 3.

Por último, se oprime el botón “Commit”. Aparece lo siguiente en la ventana “Output” del editor:



```

Git Commit
-----
git add D:\USUARIO\Documents\NetBeansProjects\ads_i_git\public_html\index.html
git commit -m version_1.0 D:\USUARIO\Documents\NetBeansProjects\ads_i_git\public_html\index.html
Commit Log
revision : e6235573257a8120e319980108694d4d4f1c3f9d
author   : webadmin <email@example.com>
date     : 8/11/2017 07:46:10 PM
summary  : version_1.0

INFO: End of Commit

==[IDE]== 8/11/2017 07:46:10 PM Committing... finished.
==[IDE]== 8/11/2017 07:46:30 PM Looking up moved file...
  
```

Figura 5.16. Primera confirmación de cambios o commit del proyecto parte 4.

Se puede observar que Netbeans usó internamente el comando “git commit” y el resultado se muestra en la ventana “Output”. Además, se muestra el hash, la fecha y hora, el usuario y el autor.

Se puede observar que una vez confirmados los cambios el nombre de la pestaña del archivo “index.html” cambia del color verde al negro.

Se pueden ver los cambios que ha tenido el proyecto hasta el momento usando la opción “Team” → “History” → “Show history” así:

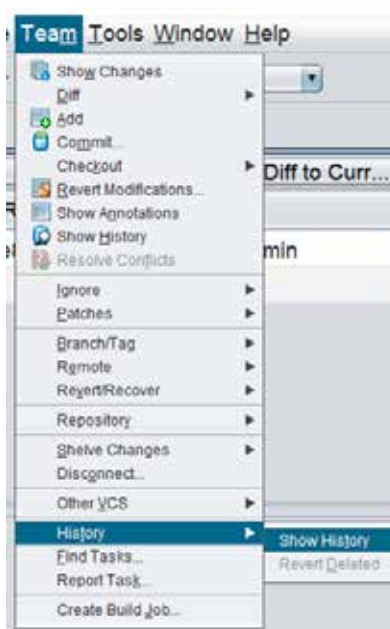


Figura 5.17.1. Primera confirmación de cambios o commit del proyecto parte 5.

Una vez seleccionada la opción aparece lo siguiente en pantalla:

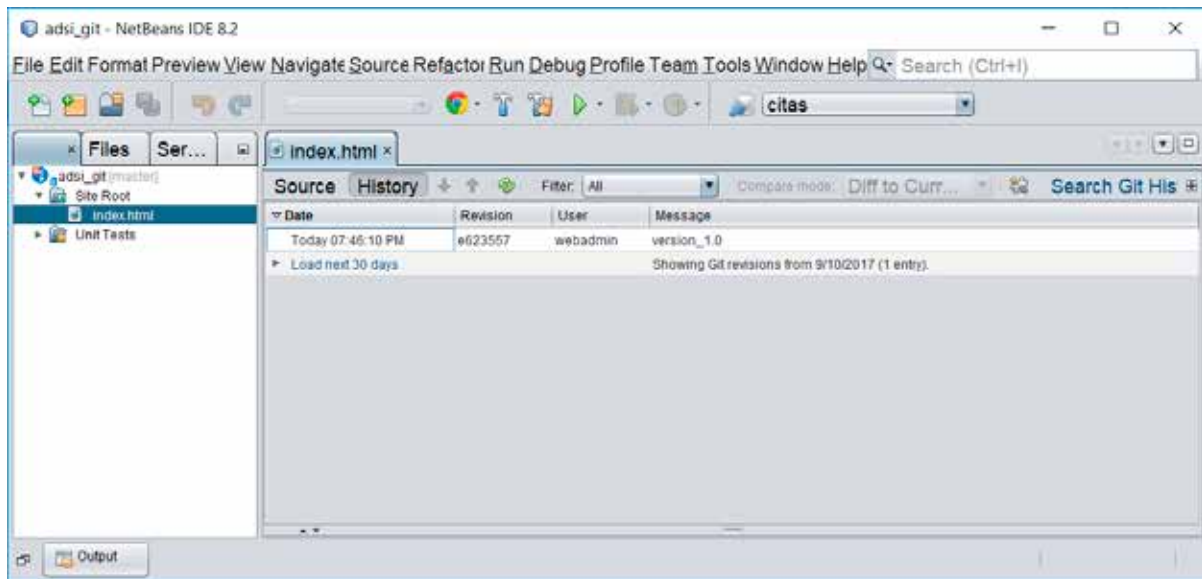


Figura 5.17.2. Primera confirmación de cambios o commit del proyecto parte 6.

Se puede observar que se relaciona el único cambio realizado cuyo mensaje es “version_1.0”. También se puede observar que el historial es ahora otra pestaña del editor. Para regresar al código sólo hay que oprimir el nombre de la pestaña “Source”. Una vez realizada lo anterior aparece lo siguiente:

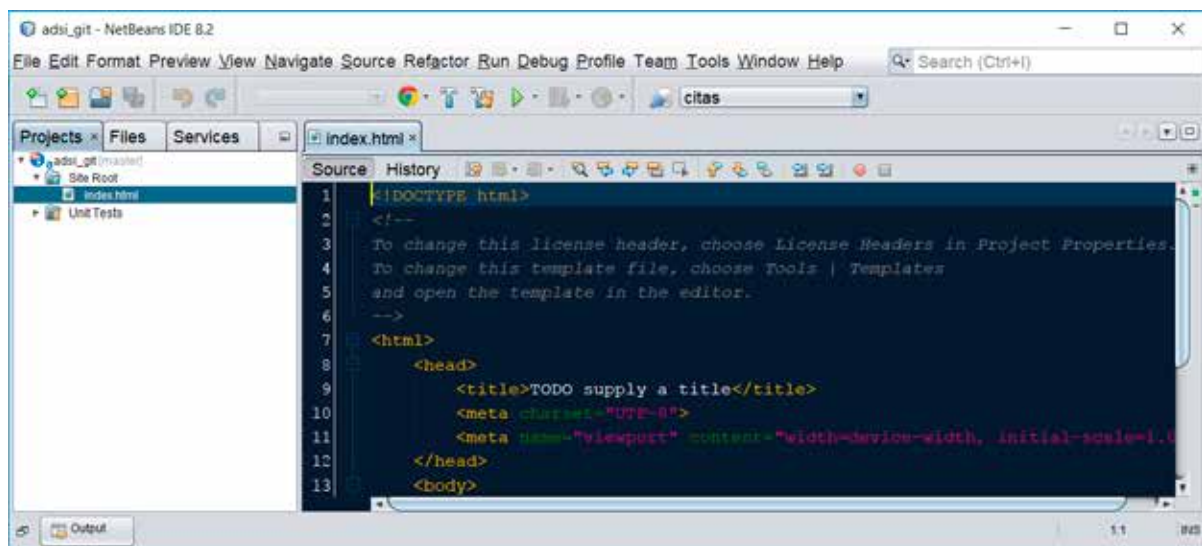


Figura 5.17.3. Primera confirmación de cambios o commit del proyecto parte 7.

5.4. Aplicación de nuevos cambios al proyecto

En el numeral anterior se realizó la confirmación de la primera versión del código fuente. A medida que un proyecto avanza se van introduciendo cambios los cuales se irán agregando usando la opción commit.

Se introducirá un cambio al archivo “index.html” que consistirá en la adición de una línea de código así:

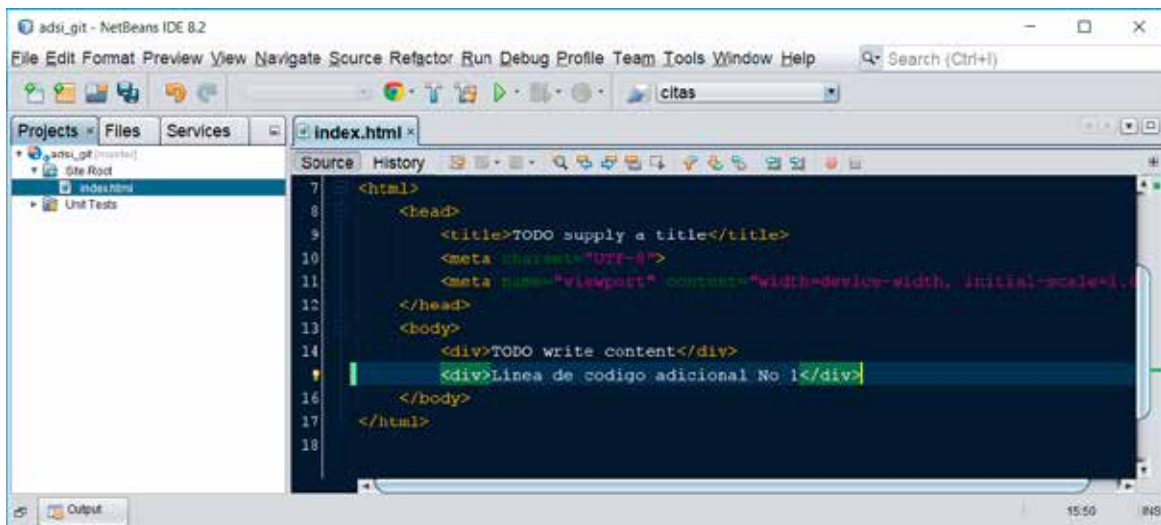


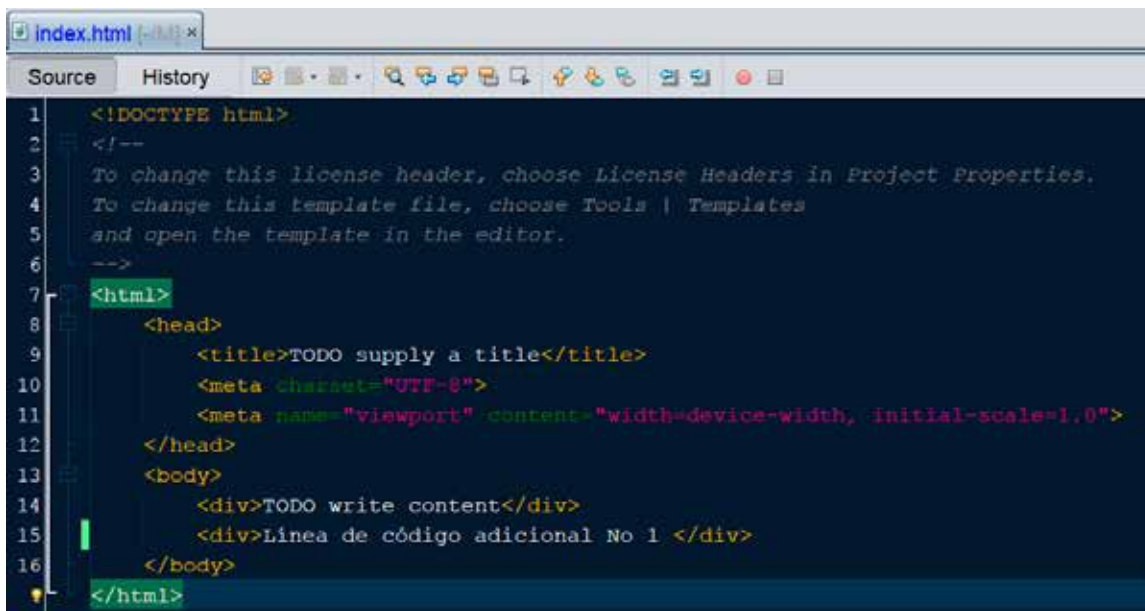
Figura 5.18. Adición de una línea de código al archivo index.html.

Luego se procede a agregar los cambios con la opción “Team” → “Add”. Aparece lo siguiente en la consola:

```
==[IDE]== 8/11/2017 06:29:54 PM Searching - D:\USUARIO\Documents\NetBeansProjects\adsi_git... finished.
==[IDE]== 8/11/2017 06:52:22 PM Git Add
==[IDE]== 8/11/2017 06:52:22 PM Git Add finished.
```



En este punto se puede observar que el color de la pestaña ha cambiado a azul y tiene un indicador “-M” que indica que el archivo ha sido modificado.



```
1 <!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project Properties.
4 To change this template file, choose Tools | Templates
5 and open the template in the editor.
6 -->
7 <html>
8 <head>
9 <title>TODO supply a title</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
13 <body>
14 <div>TODO write content</div>
15 <div>Línea de código adicional No 1 </div>
16 </body>
17 </html>
```

Figura 5.19 Adición de una línea de código al archivo index.html parte 2.

Por último se genera una nueva confirmación que se llamará: “version_1.1” así:

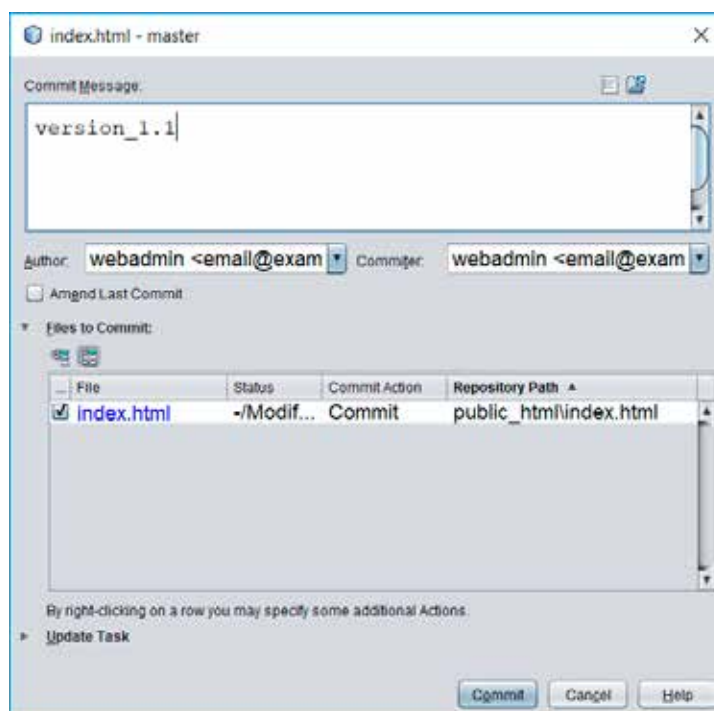


Figura 5.20. Adición de una línea de código al archivo index.html parte 3.

Una vez realizada se oprime el botón “Commit” aparece lo siguiente en la consola:



```

Output - adsi_git - D:\USUARIO\Documents\NetBeansProjects\adsi_git
Git Commit
git add D:\USUARIO\Documents\NetBeansProjects\adsi_git\public_html\index.html
git commit -m version_1.1 D:\USUARIO\Documents\NetBeansProjects\adsi_git\public_html\index.html
Commit Log
revision : ad5172c539f5977b779f170634b5fdb501c4581
author : webadmin <email@example.com>
date : 8/11/2017 07:51:47 PM
summary : version_1.1
INFO: End of Commit
  
```

Figura 5.21.1. Adición de una línea de código al archivo index.html parte 4.

En este punto se puede revisar el historial de cambios dando clic sobre la pestaña “History”. Aparece lo siguiente en pantalla:



Date	Revision	User	Message
Today 07:51:47 PM	ad5172c	webadmin	version_1.1
Today 07:46:10 PM	e623557	webadmin	version_1.0
Today 07:45:39 PM	Local		

Showing Git revisions from 9/09/2017 (2 entries).

Figura 5.21.2. Adición de una línea de código al archivo index.html parte 5.

5.5. Revisión de los cambios aplicados

Para ubicar el proyecto en una versión específica se utiliza la opción “Checkout revision” del menú “Checkout” de la opción “Team” como se muestra a continuación:

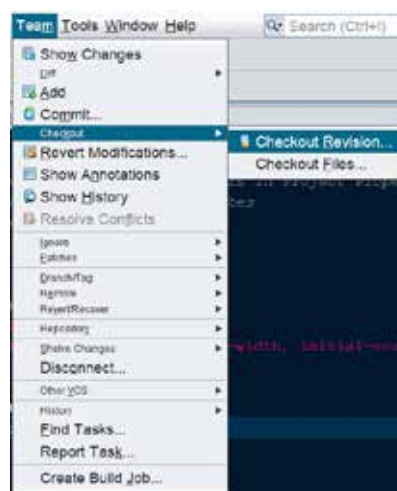


Figura 5.22. Revisión de cambios con el comando checkout.

Al seleccionar la opción aparece lo siguiente en pantalla:

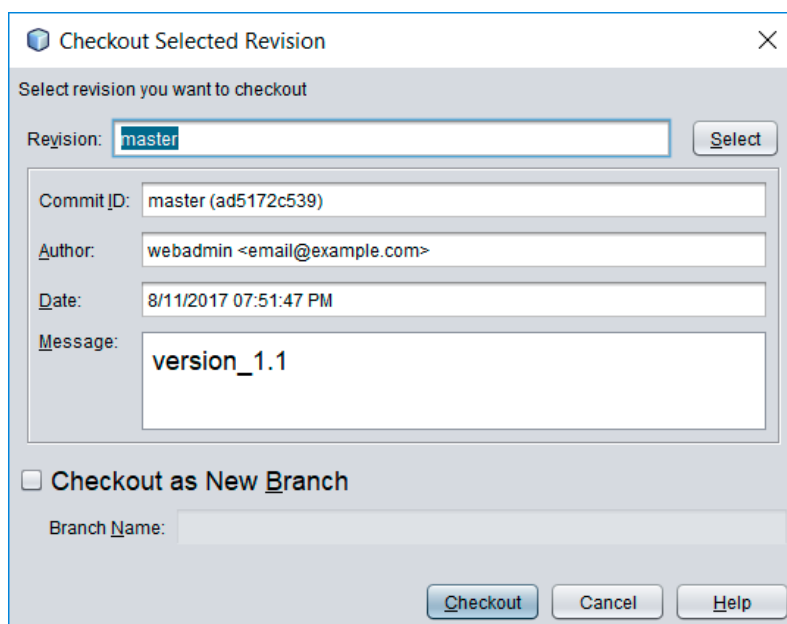


Figura 5.23. Revisión de cambios con el comando checkout parte 2.

La pantalla indica que está ubicado sobre la última confirmación o “master”. También pide que se seleccione el punto de recuperación o commit al cual se quiere revisar. Para realizar esta última selección se oprime el botón “Select”. Aparece lo siguiente en pantalla:

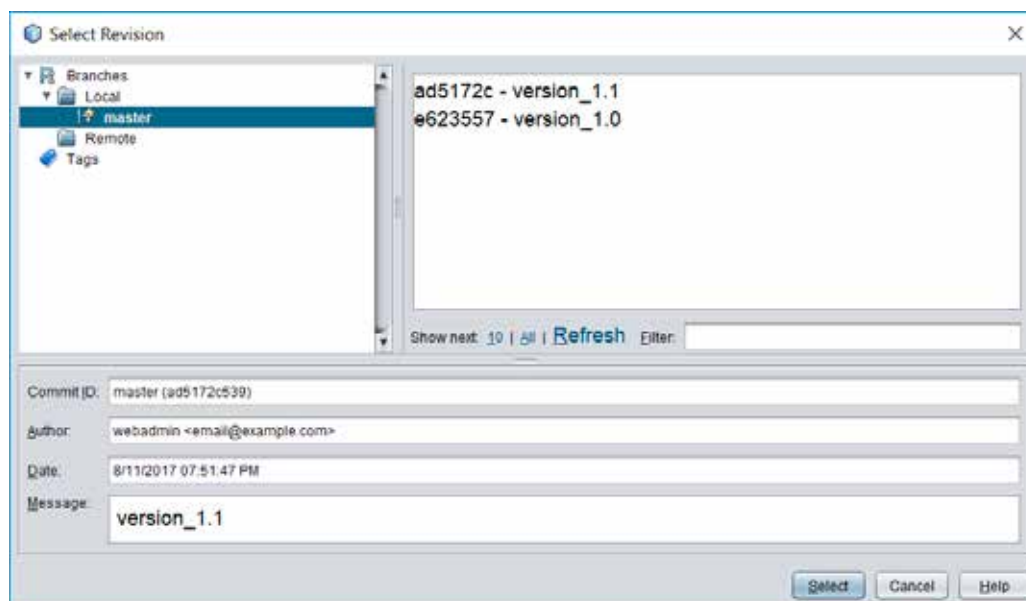


Figura 5.24. Revisión de cambios con el comando checkout parte 3.

Se puede observar que existen dos puntos de restauración o commits. Se solicitará que se regrese el proyecto al commit llamado “version_1.0” que fue el commit inicial. Se selecciona este último dando clic sobre el nombre así:

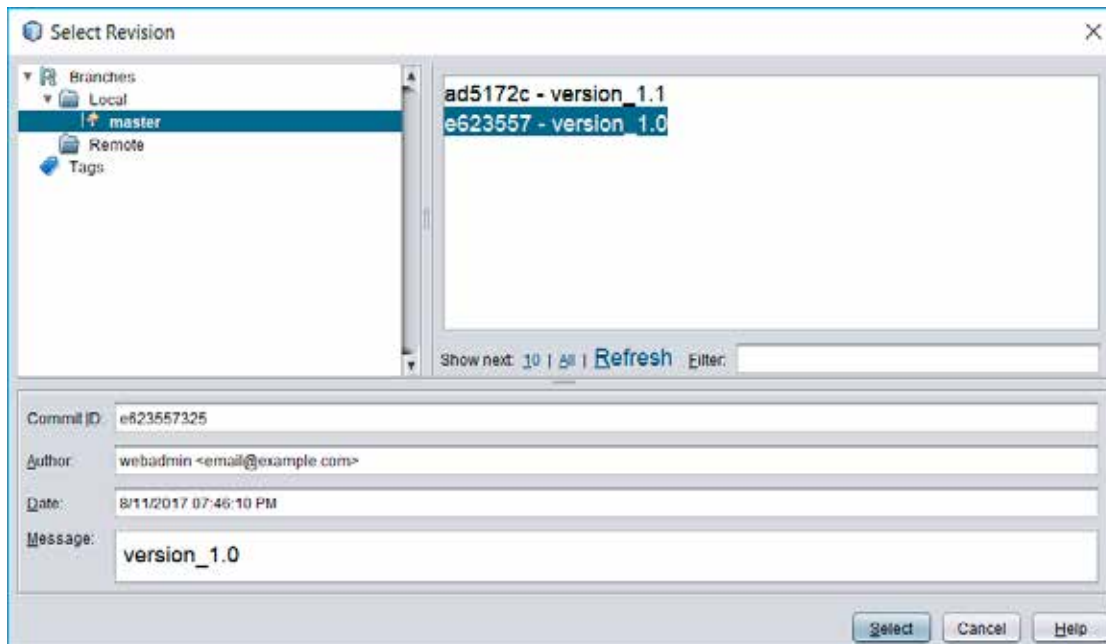


Figura 5.25. Revisión de cambios con el comando checkout parte 4.

Al dar clic el sistema actualiza los campos “Commit ID”, “Author”, “Date” y “Message”. Por último, se da clic sobre el botón “Select”. Aparece lo siguiente en pantalla:

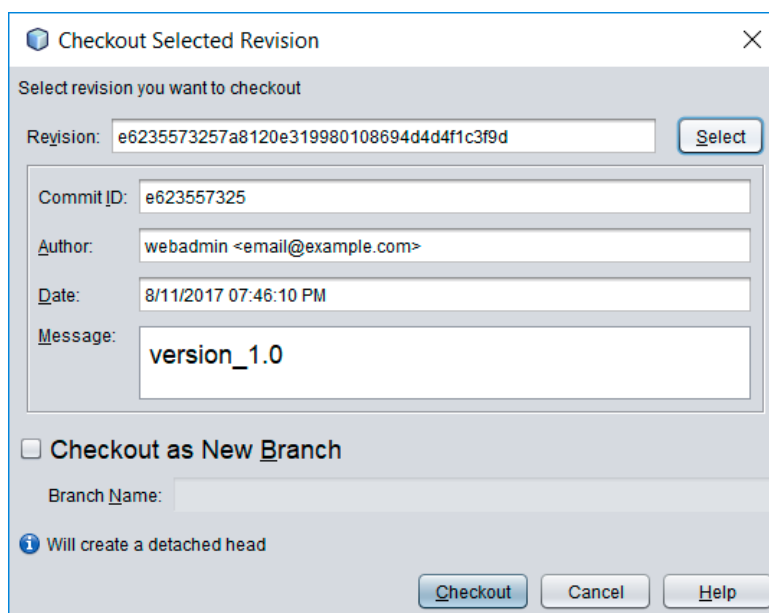


Figura 5.26. Revisión de cambios con el comando checkout parte 5.

La ventana es una última confirmación de los cambios. Para que la aplicación se ubique en la versión 1.0 se da clic sobre el botón “Checkout”. Una vez oprimido aparece lo siguiente en pantalla:

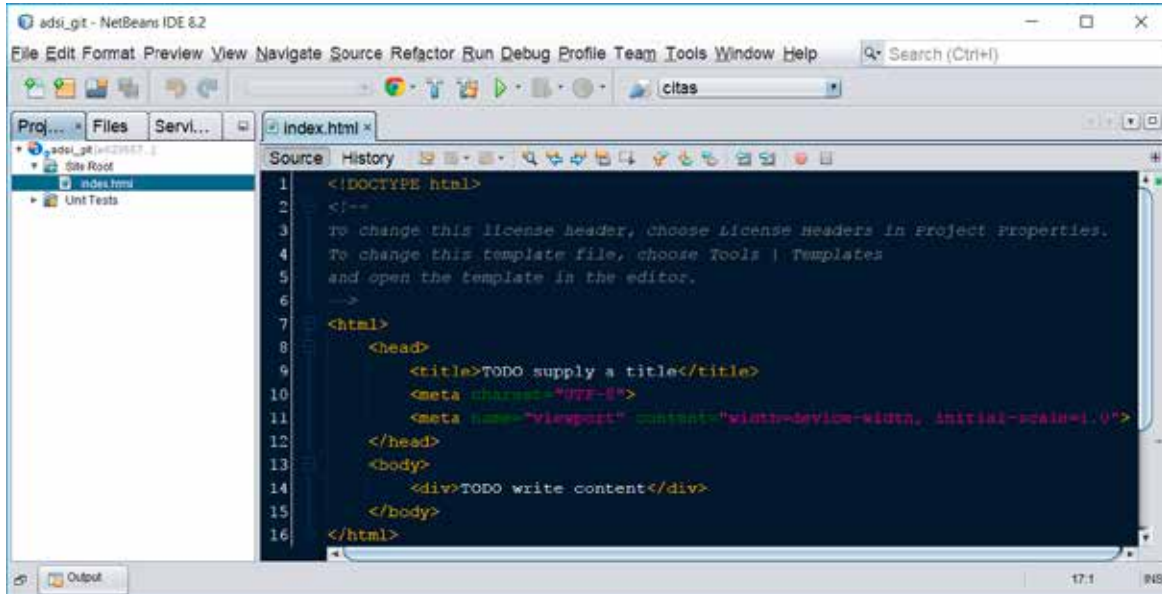


Figura 5.27. Revisión de cambios con el comando checkout parte 6.

Se puede observar que el archivo “index.html” volvió a su estado inicial. En este punto se puede revisar el proyecto como se encontraba en la versión 1.0. Una vez finalizada esta revisión se procede a regresar a la versión actual del mismo.

Para lo anterior se selecciona nuevamente la opción “Team” → “Checkout” → “Checkout revisions ...”. Luego al seleccionar el punto de restauración o commit se escoge “Master” como se muestra a continuación:

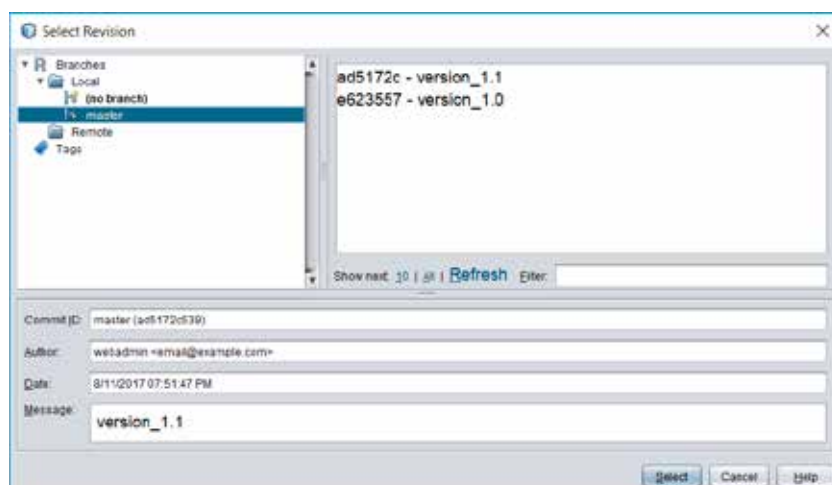


Figura 5.28. Revisión de cambios con el comando checkout parte 7.

Por último, se oprime el botón “Select”. Posteriormente se oprime el botón “Checkout” y el proyecto regresa a su estado actual como se muestra a continuación:

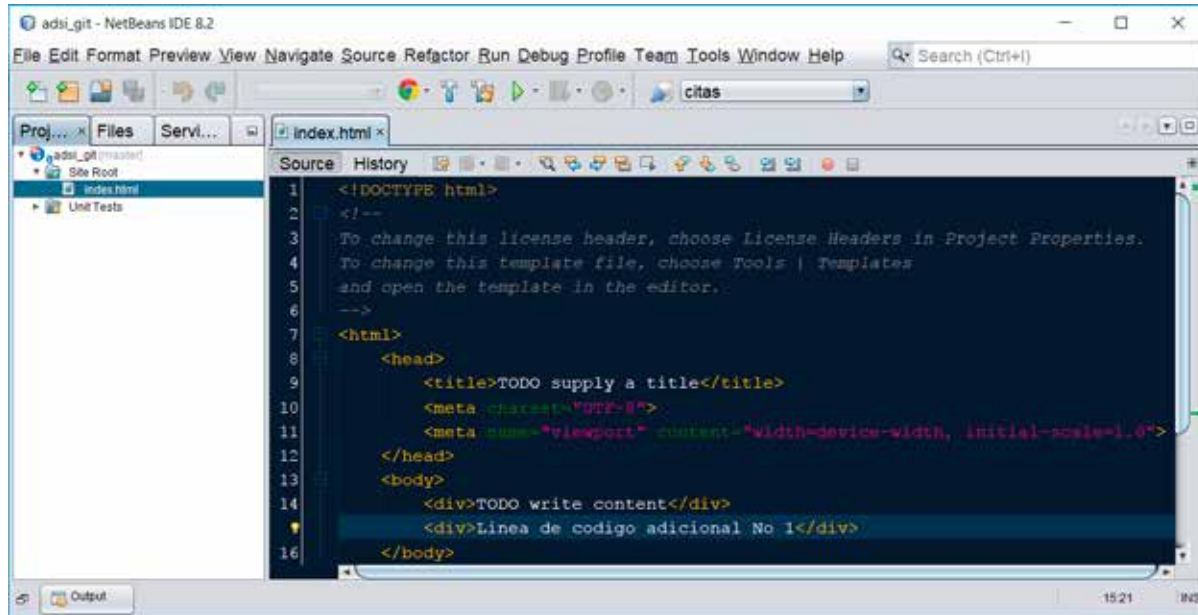


Figura 5.29. Revisión de cambios con el comando checkout parte 8.