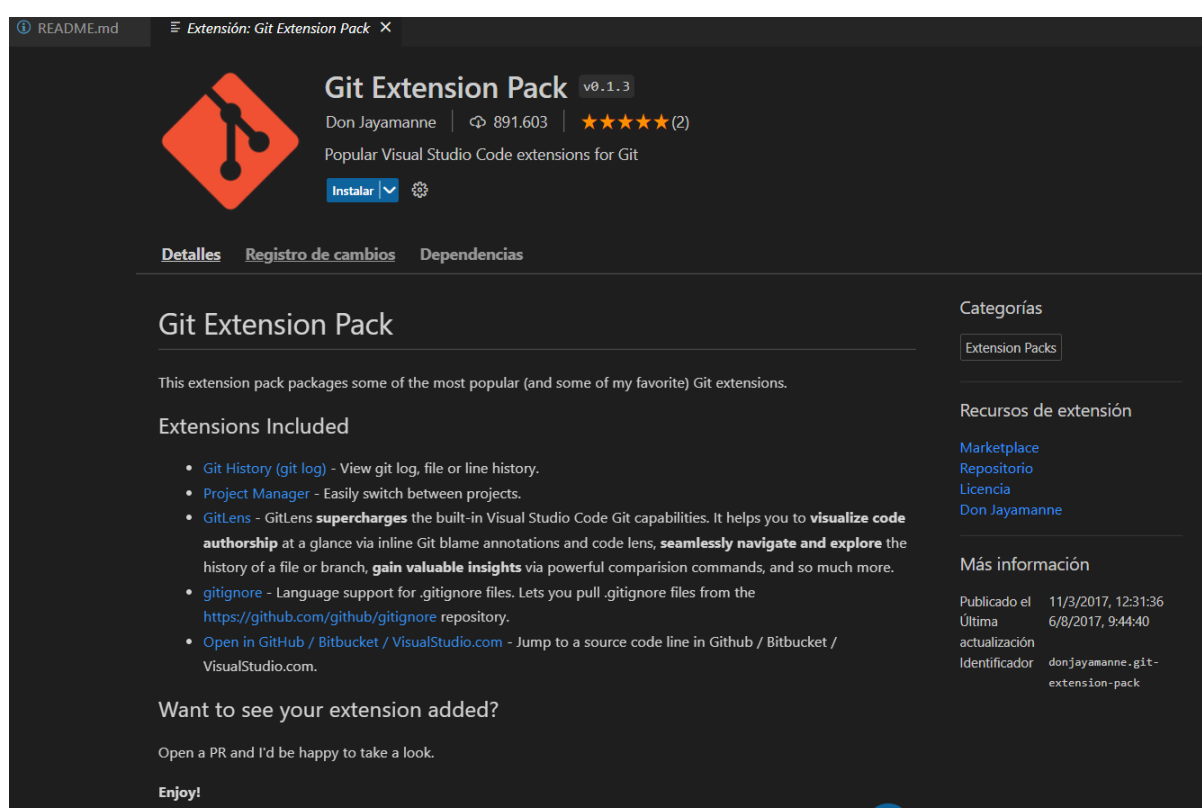


## 1. ¿Necesito instalar extensiones para trabajar con Git en Code? ¿Cuáles?

Actualmente no es necesario instalar extensiones para trabajar con Git, ya que como hemos comprobado en la guía introductoria podemos modificar nuestros archivos y subirlos aquellos que estén en local a través de Git, o tener nuestros archivos en nuestro directorio de trabajo, aún siendo creados en remoto, y de esta forma poder modificar nuestro código en el IDE de trabajo, en nuestro caso Code.

Además investigando en la [documentación oficial de Visual Studio Code](#) podemos ver cómo explica que Git tiene compatibilidad con Code ya que “Code tiene administración de control de código fuente integrada”, aunque sí es cierto que existen extensiones para mejorar la línea de trabajo cómo “Git Extension Pack”.



The screenshot shows the Visual Studio Code extension marketplace interface. At the top, there's a header with a README link and the extension name 'Git Extension Pack' by Don Jayamanne, version v0.1.3, with 891,603 installations and a 5-star rating. Below this, there's a description: 'Popular Visual Studio Code extensions for Git'. The main content area is titled 'Git Extension Pack' and includes a description: 'This extension pack packages some of the most popular (and some of my favorite) Git extensions.' It lists several extensions included: Git History (git log), Project Manager, GitLens, gitignore, and Open in GitHub / Bitbucket / VisualStudio.com. On the right side, there's a sidebar with 'Categorías' (Extension Packs), 'Recursos de extensión' (Marketplace, Repositorio, Licencia, Don Jayamanne), and 'Más información' (Publicado el, Última actualización, Identificador).

README.md Extension: Git Extension Pack X

**Git Extension Pack** v0.1.3  
Don Jayamanne | 891.603 | ★★★★★ (2)  
Popular Visual Studio Code extensions for Git  
[Instalar](#) ⌵ ⚙

[Detalles](#) [Registro de cambios](#) [Dependencias](#)

### Git Extension Pack

This extension pack packages some of the most popular (and some of my favorite) Git extensions.

#### Extensions Included

- [Git History \(git log\)](#) - View git log, file or line history.
- [Project Manager](#) - Easily switch between projects.
- [GitLens](#) - GitLens **supercharges** the built-in Visual Studio Code Git capabilities. It helps you to **visualize code authorship** at a glance via inline Git blame annotations and code lens, **seamlessly navigate and explore** the history of a file or branch, **gain valuable insights** via powerful comparison commands, and so much more.
- [gitignore](#) - Language support for .gitignore files. Lets you pull .gitignore files from the <https://github.com/github/gitignore> repository.
- [Open in GitHub / Bitbucket / VisualStudio.com](#) - Jump to a source code line in Github / Bitbucket / VisualStudio.com.

Want to see your extension added?

Open a PR and I'd be happy to take a look.

Enjoy!

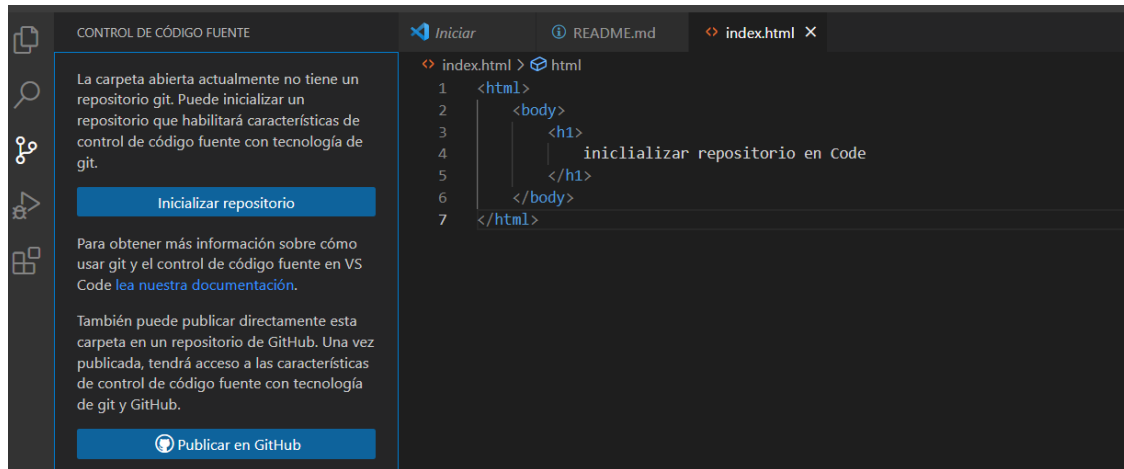
**Categorías**  
[Extension Packs](#)

**Recursos de extensión**  
[Marketplace](#)  
[Repositorio](#)  
[Licencia](#)  
[Don Jayamanne](#)

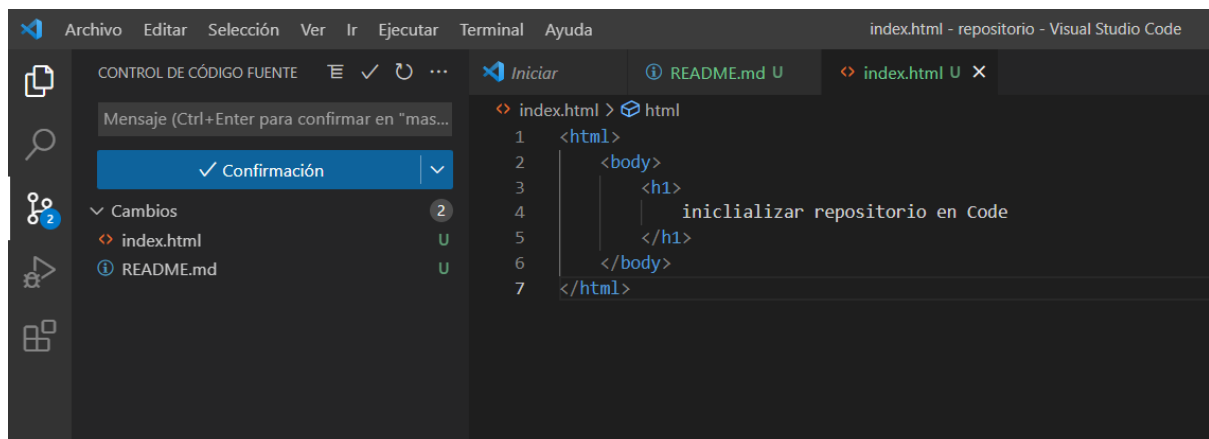
**Más información**  
Publicado el 11/3/2017, 12:31:36  
Última actualización 6/8/2017, 9:44:40  
Identificador donjayamanne.git-extension-pack

## 2. ¿Cómo inicializo un repositorio en Code?

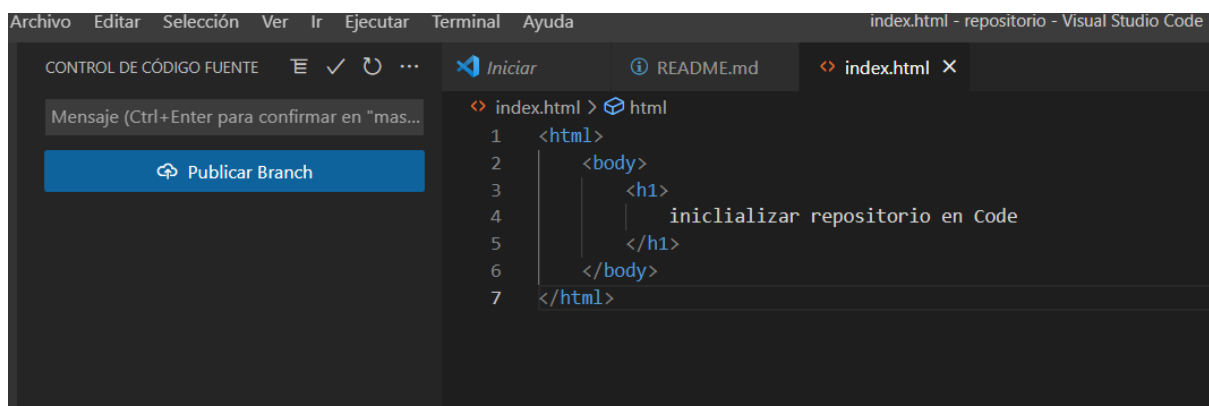
Para ver cómo inicializamos un repositorio en Code, vamos a crear un nuevo directorio, con nuevos archivos.



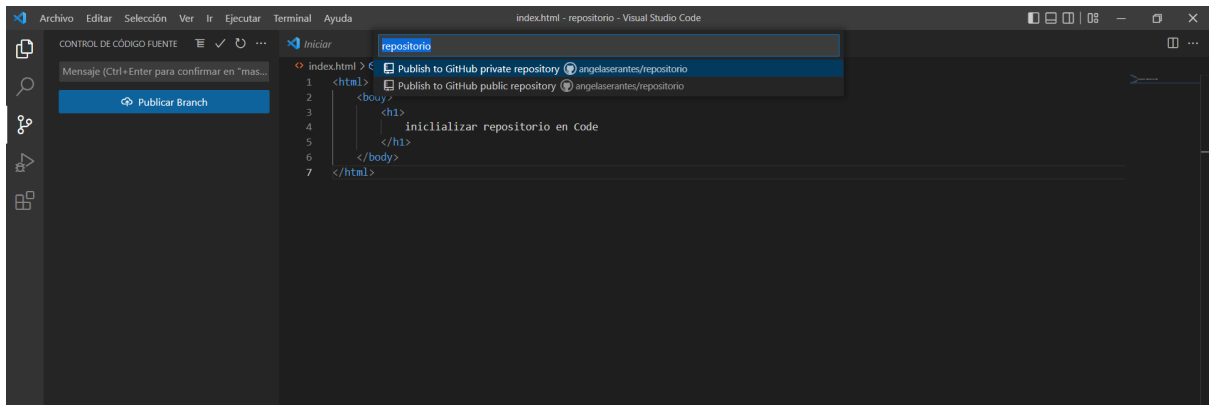
En este caso clicamos en inicializar repositorio.



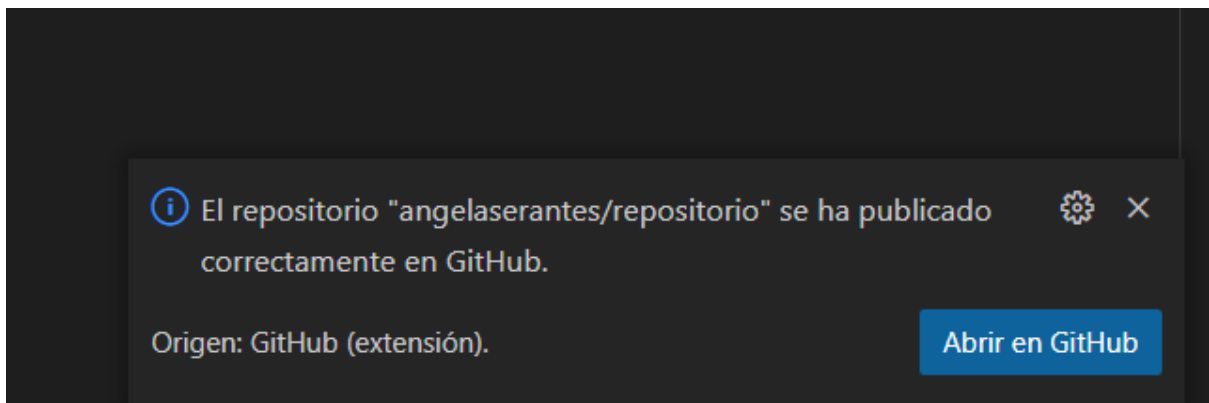
Una vez hecho esto confirmamos los cambios escribiendo un mensaje. Y publicamos la rama:



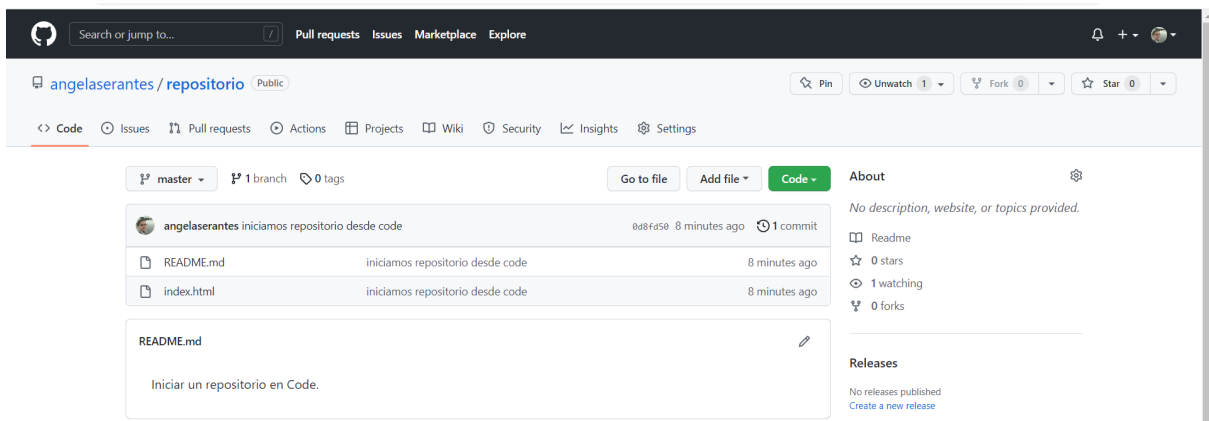
A continuación creamos un repositorio público:



Y el propio IDE, nos informa que el repositorio se ha subido correctamente:

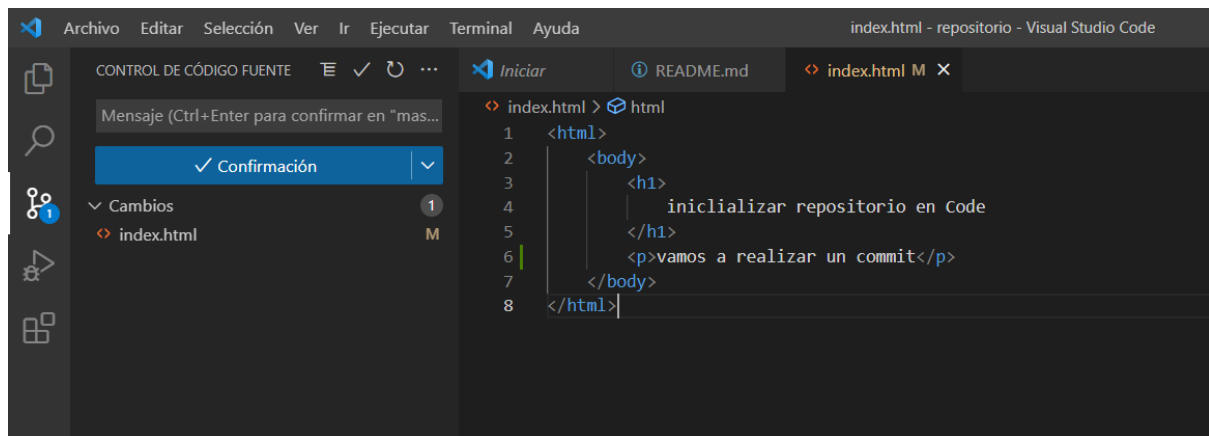


Y ya podemos visualizar nuestro repositorio en el navegador:



### 3. ¿Cómo se hace un commit en Code?

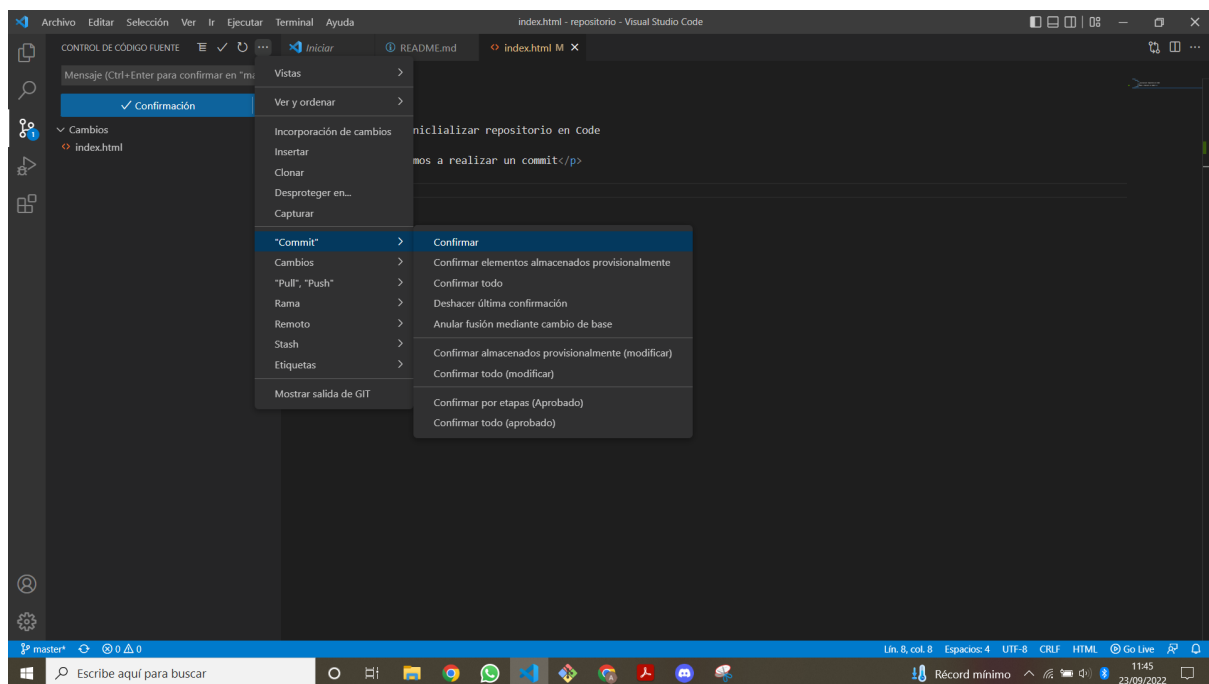
Para explicar cómo se hace un commit vamos a utilizar el repositorio anterior, en primer lugar cambiaremos parte del código.



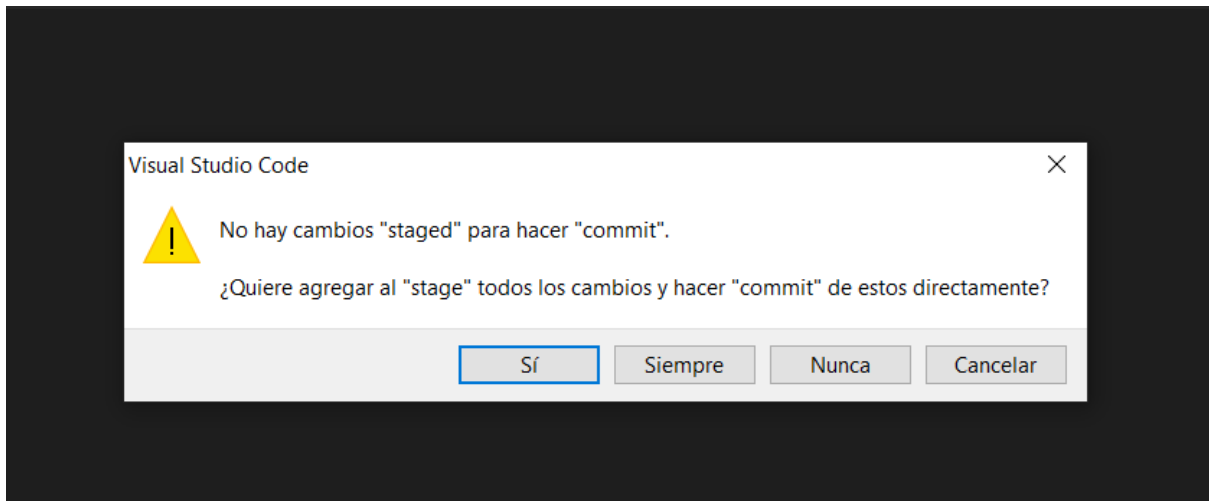
Cómo podemos observar el propio IDE nos indica en color verde que en la línea 6 hay un cambio y nos indica a mano izquierda que hay un cambio en el *index.html*. Tenemos varias opciones para hacer un commit.

En primer lugar tenemos la opción de desplegar el botón de al lado de confirmación y nos proporciona las opciones de: confirmación, hacer commit e insertar o hacer commit y sincronizar.

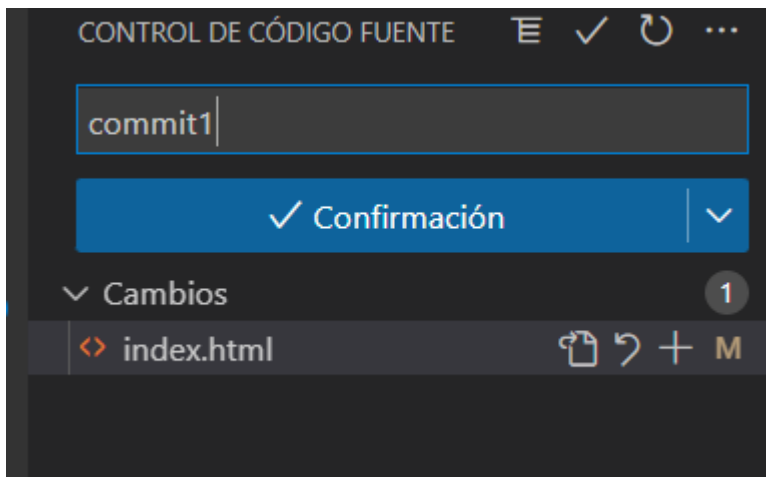
En segundo lugar podemos clicar en los tres puntos al lado de código fuente y podemos seleccionar “commit” y clicar en confirmar.



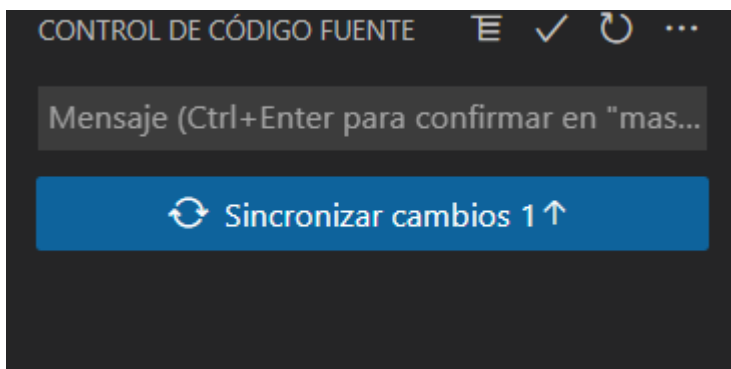
Nos saldrá este aviso y le decimos que sí:



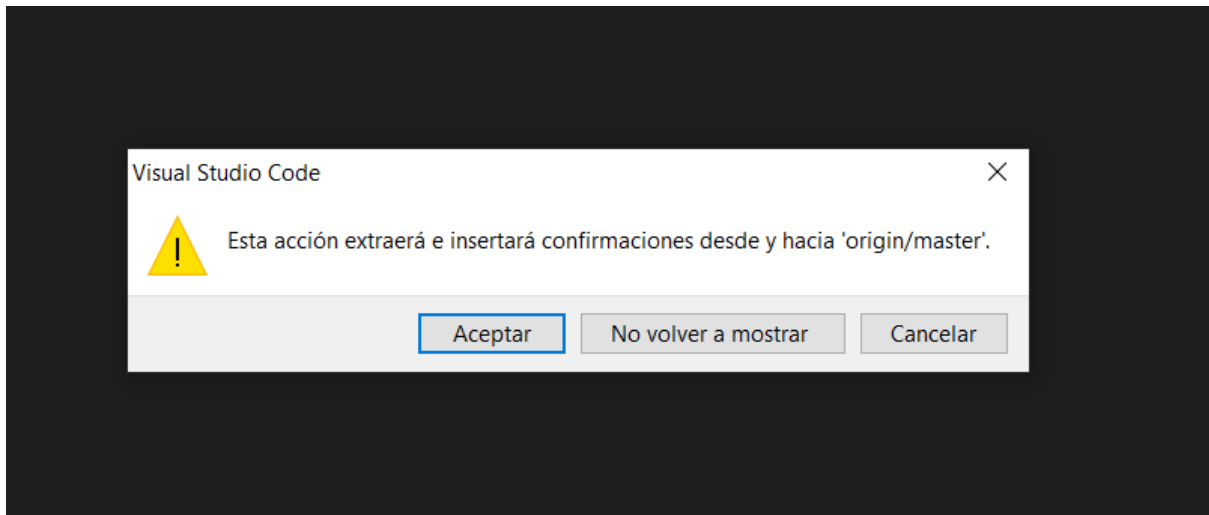
Para poder hacer el commit tendremos que escribir un mensaje



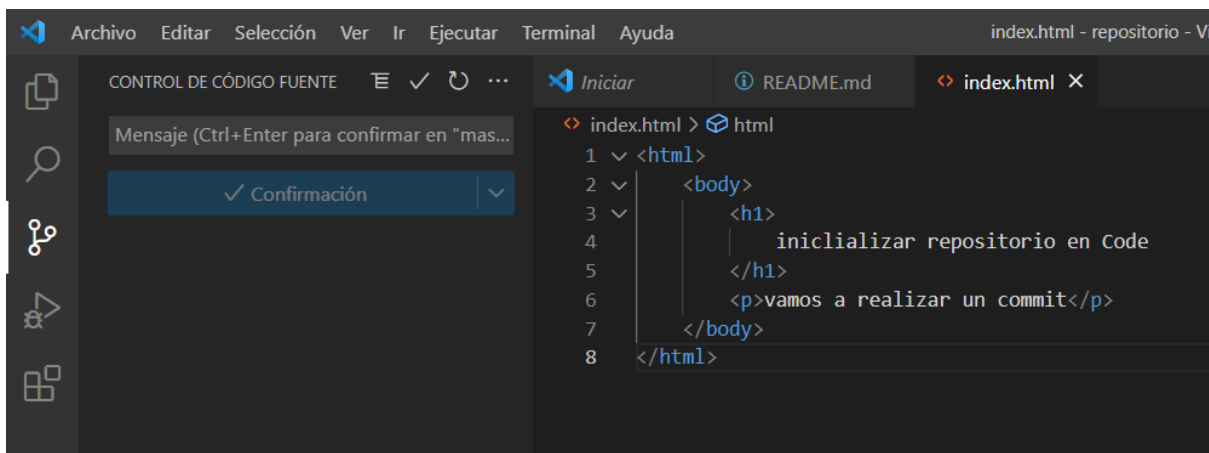
A continuación, sincronizamos los cambios:



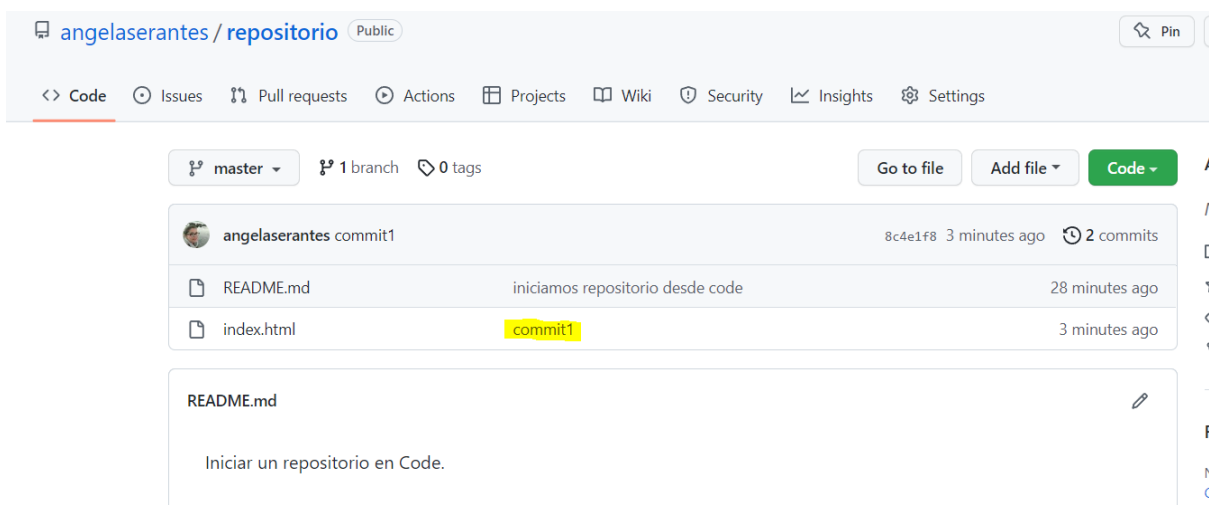
Nuevamente nos sale otra aviso y aceptamos:




En Code no tendríamos más cambios pendientes:




Comprobamos que todo está correcto en Github y observamos que tenemos nuestro primer commit:




 [angelaserantes / repositorio](#) Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

commit1

 master


 angelaserantes committed 4 minutes ago

Showing 1 changed file with 1 addition and 0 deletions.

1 index.html

↑	@@ -3,5 +3,6 @@	
3	3	<h1>
4	4	inicializar repositorio en Code
5	5	</h1>
6	+	<p>vamos a realizar un commit</p>
6	7	</body>
7	8	</html>

0 comments on commit `8c4e1f8`



Write

Preview

Y dentro de el commit tenemos el cambio que hemos realizado en el código.

#### 4. Para crear el repositorio en GitHub, tengo que ir a la página de GitHub?

No es necesario se puede crear el repositorio desde Code por ejemplo como hemos hecho en esta práctica.