

Introducción a GIT

con diferencia el Sistema de Control de Versiones más usado en la actualidad

Aplicado al proyecto de ejemplo de la UT.1

¿Qué es git?

Hoy en día, **Git** es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo

Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia.



Descárgalo de aquí: <https://git-scm.com/>

¿Qué es Github?

Github es un sitio que permite alojar repositorios de **Git**.



GitHub

Hay muchas otras opciones para alojar tus repositorios de Git

Algunas de las más conocidas son:



GitLab



Qué deberías tener instalado antes de empezar con este tutorial



En este tutorial también se usa **Visual Studio Code**. Instálalo si aún no lo has hecho.

Instala el software de Git para el Sistema Operativo que estés usando. En nuestro caso Windows

¿Qué haremos en este tutorial?

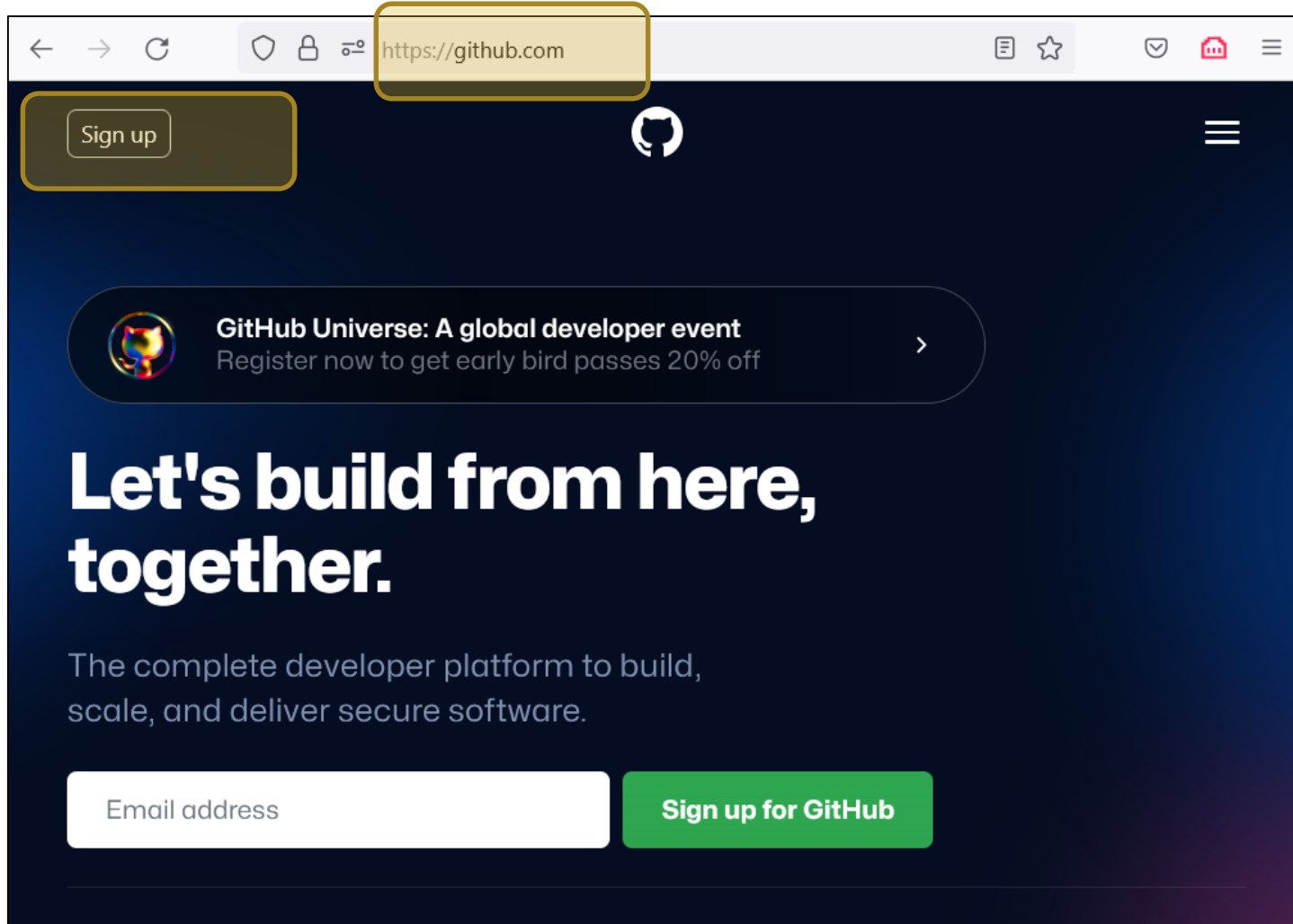
Simplemente
empezaremos a usar
Git con un proyecto
de ejemplo que
subiremos a **Github**



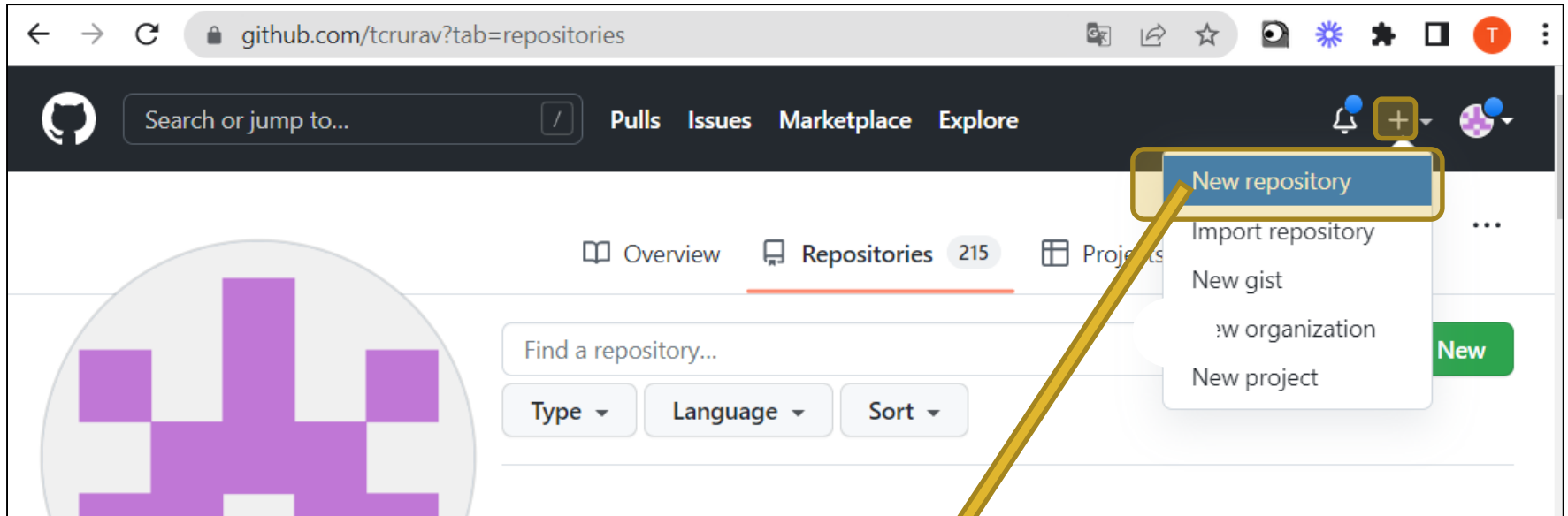
El ejemplo es el que
hemos trabajado en
clase en esta U.T.
Haremos énfasis en
explicar los problemas
que te puedes encontrar



Crea una cuenta de Github en <https://github.com>

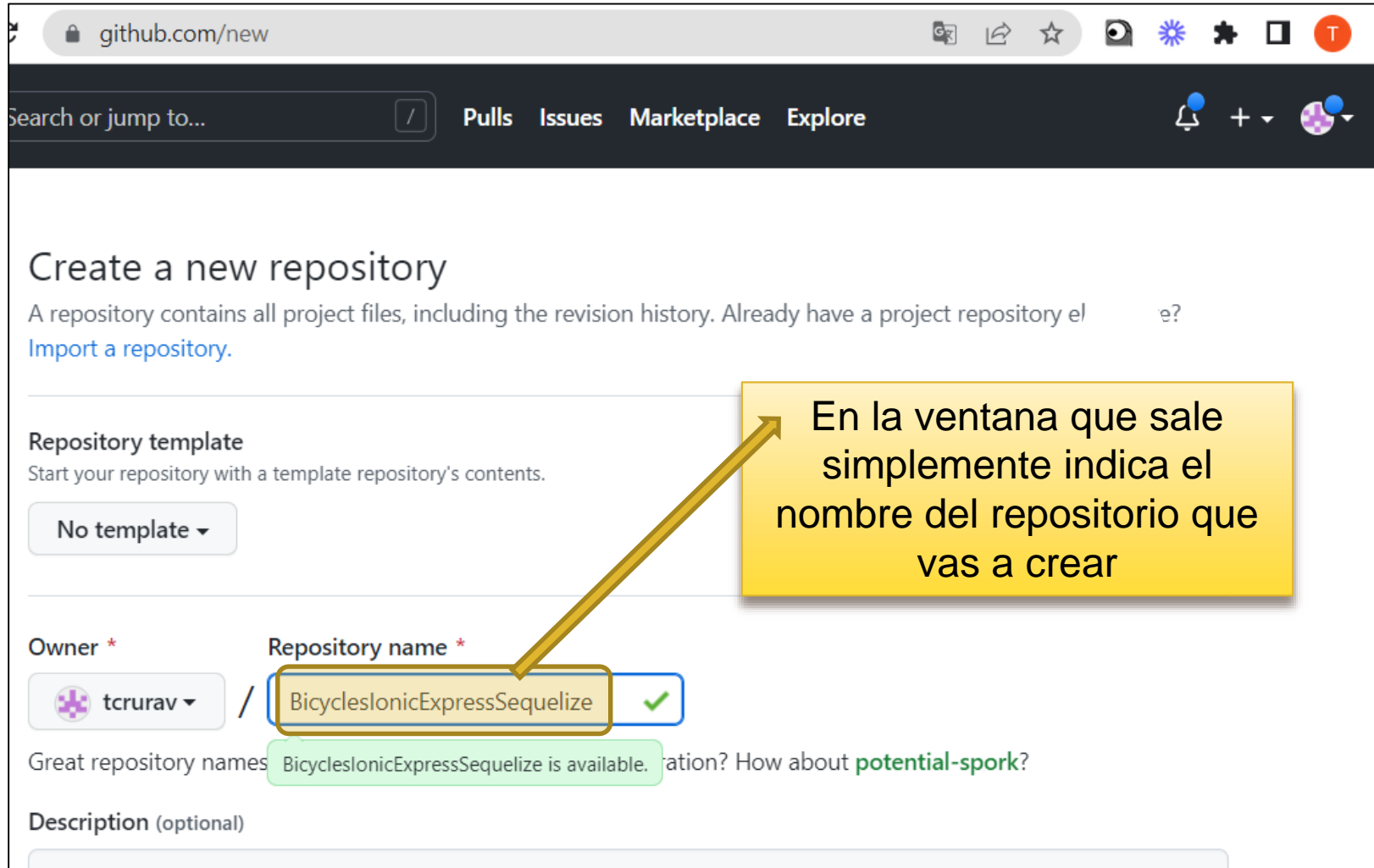


Crea un nuevo repositorio remoto con Github



Una vez creada la cuenta y autenticado en **Github** haz clic en el **símbolo +** y luego crea un nuevo repositorio

Crea un nuevo repositorio remoto con Github



The screenshot shows the GitHub 'Create a new repository' page. The browser address bar shows 'github.com/new'. The page has a dark header with navigation links: 'Search or jump to...', 'Pulls', 'Issues', 'Marketplace', and 'Explore'. Below the header, the main content area is titled 'Create a new repository'. It includes a sub-header 'Repository template' with a dropdown menu set to 'No template'. The 'Owner' field is set to 'tcrurav'. The 'Repository name' field is highlighted with a yellow box and contains the text 'BicycleslonicExpressSequelize', which is also highlighted by a yellow callout box. A green checkmark is visible next to the repository name. Below the name field, a green message states 'BicycleslonicExpressSequelize is available.' and suggests 'potential-spork?' as an alternative. The 'Description (optional)' field is at the bottom.

github.com/new

Search or jump to... Pulls Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository el e?

[Import a repository.](#)

Repository template
Start your repository with a template repository's contents.

No template ▾

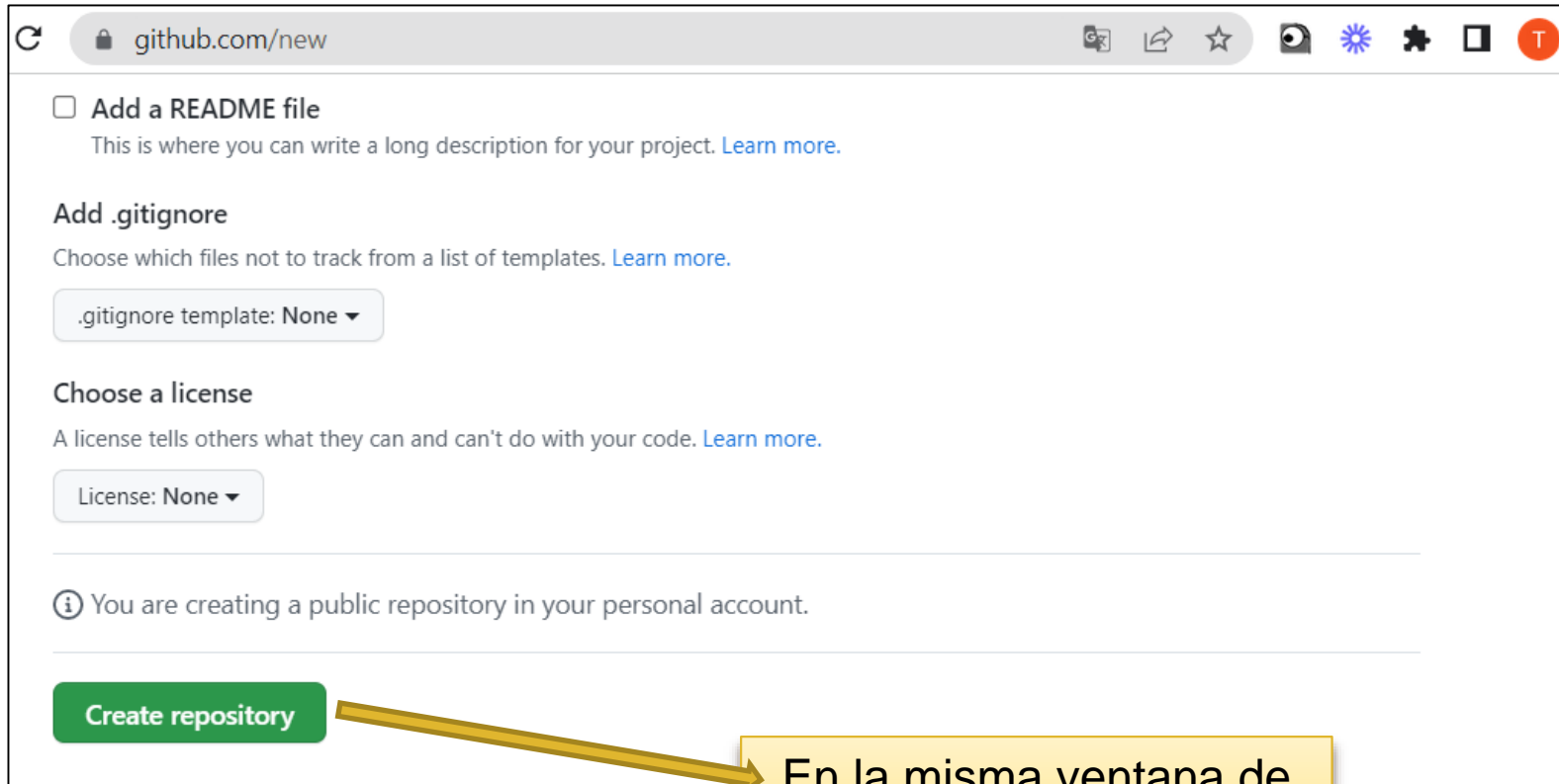
Owner * tcrurav ▾ / **Repository name *** BicycleslonicExpressSequelize ✓

Great repository names BicycleslonicExpressSequelize is available. ation? How about **potential-spork?**

Description (optional)

En la ventana que sale simplemente indica el nombre del repositorio que vas a crear

Crea un nuevo repositorio remoto con Github



github.com/new

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

Create repository

En la misma ventana de la página anterior abajo del todo haz clic en el botón para crear el repositorio

Crea un nuevo repositorio remoto con Github

The screenshot shows the GitHub interface for a newly created repository. The browser address bar shows 'github.com/tcrurav/BicyclesIonicExpressSequelize'. The repository name 'tcrurav / BicyclesIonicExpressSequelize' is displayed with a 'Public' badge. Below the repository name are buttons for 'Pin', 'Unwatch' (with a count of 1), and 'Fork'. A navigation bar includes links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', and 'Security'. A light blue box contains a 'Quick setup' section with options for 'Set up in Desktop', 'HTTPS', and 'SSH', along with the repository's URL. Below this, a section titled '...or create a new repository on the command line' provides a list of terminal commands to initialize a new repository and create a first commit.

github.com/tcrurav/BicyclesIonicExpressSequelize

Search or jump to... / Pulls Issues Marketplace Explore

tcrurav / BicyclesIonicExpressSequelize Public

Pin Unwatch 1 Fork

<> Code Issues Pull requests Actions Projects Wiki Security

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/tcrurav/BicyclesIonicExpressSequelize>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [.gitignore](#).

...or create a new repository on the command line

```
echo "# BicyclesIonicExpressSequelize" >> README.md
git init
git add README.md
git commit -m "first commit"
```

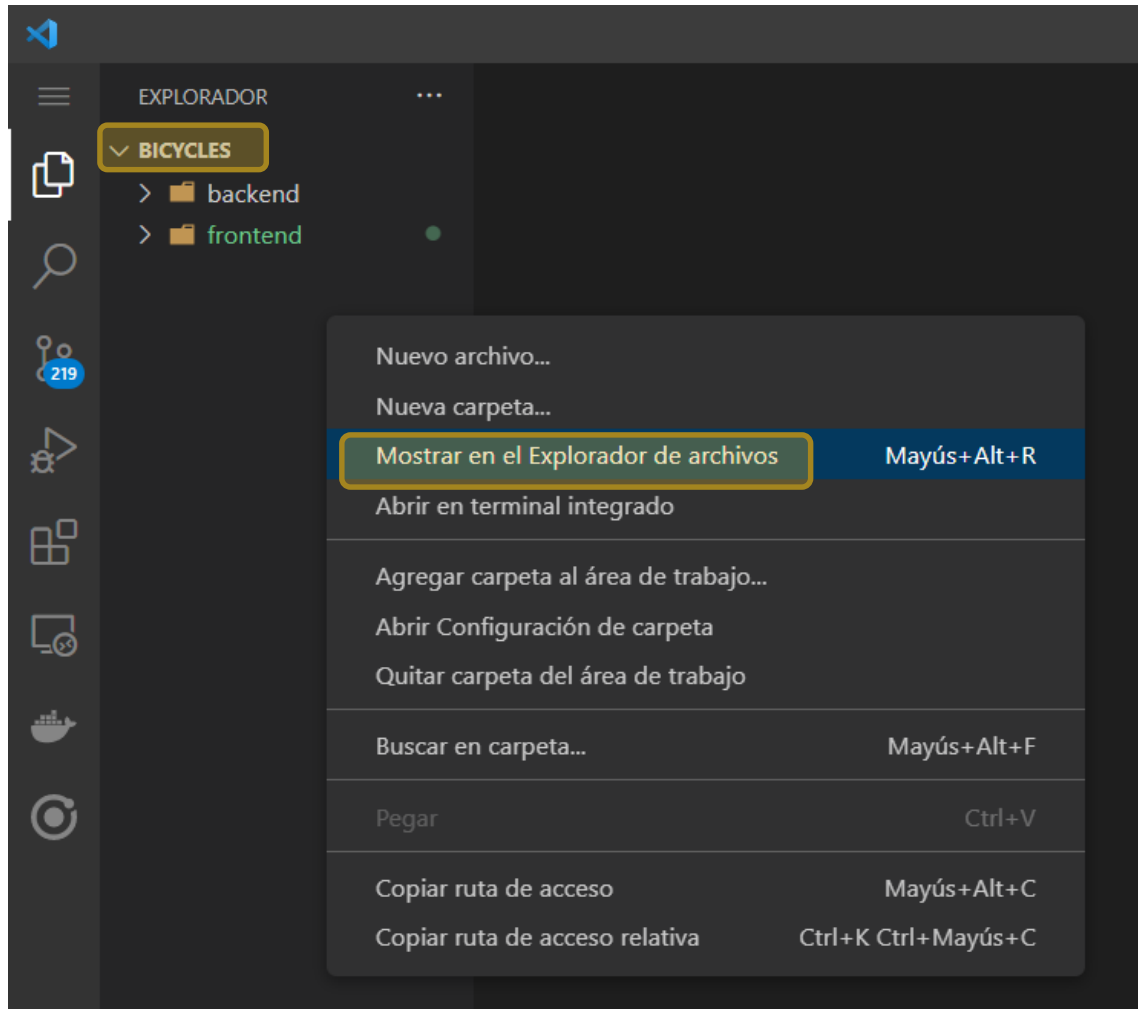
Aquí se ve como el repositorio se ha creado pero aún está vacío porque aún no has subido nada a él. También se ven algunas instrucciones de cómo hacerlo.

En los siguientes pasos vamos a hacerlo

Ya hemos creado nuestro
repositorio remoto en Github

**Ahora vamos a empezar a
trabajar localmente**

Trabajo previo antes de subir nuestro primer commit a Github



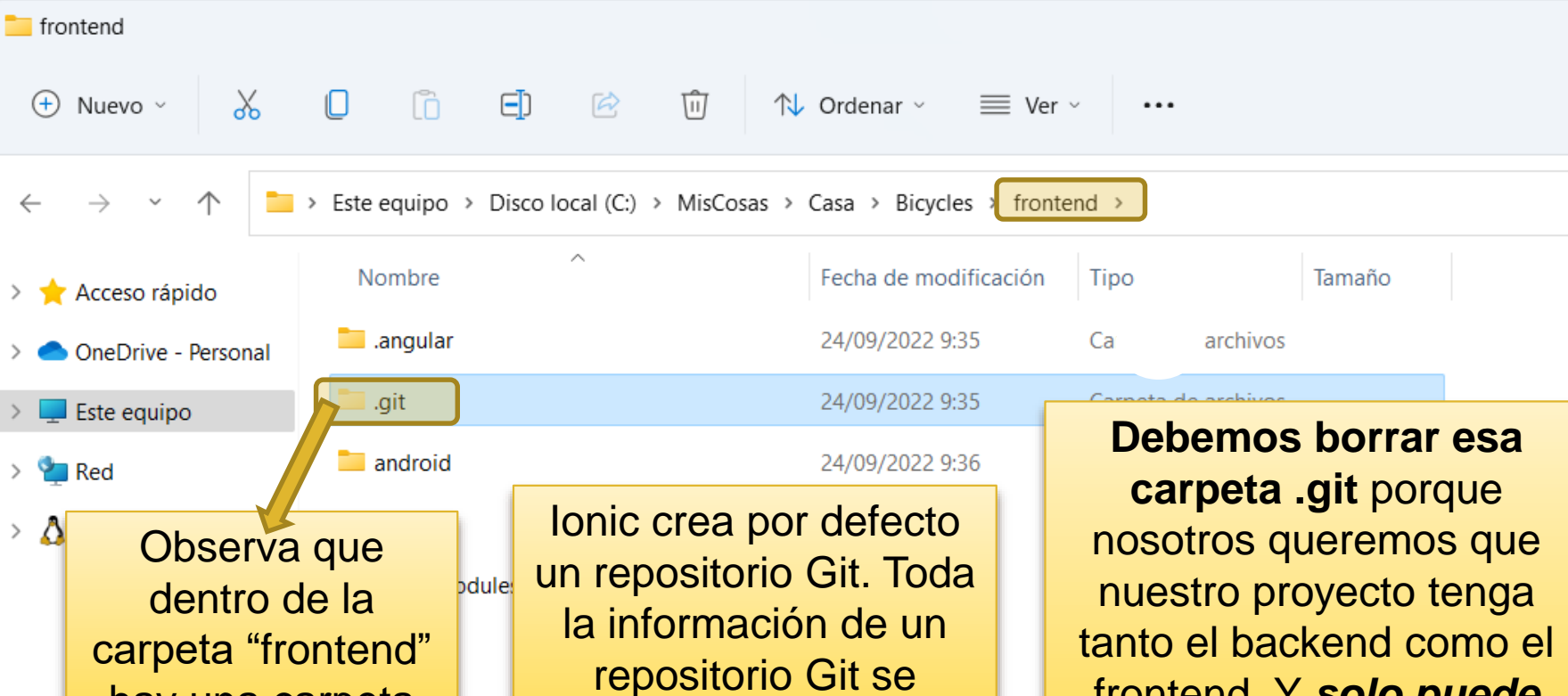
El proyecto que vamos a subir a Github en este tutorial es el que hemos estado trabajando. Este proyecto contiene dos carpetas: backend y frontend.

El pantallazo muestra como se ve el proyecto en Visual Studio Code

Con el botón derecho sobre el proyecto puedes ir al explorador de archivos donde se encuentra el código.

Trabajo previo

Borra la carpeta “frontend/.git”



frontend

Nuevo ▾

Ordenar ▾ Ver ▾

← → ▾ ↑

Este equipo > Disco local (C:) > MisCosas > Casa > Bicycles > frontend >

	Nombre	Fecha de modificación	Tipo	Tamaño
> ★ Acceso rápido				
> OneDrive - Personal				
> Este equipo				
> Red				
>				
	.angular	24/09/2022 9:35	Ca	archivos
	.git	24/09/2022 9:35	Carpeta de archivos	
	android	24/09/2022 9:36		

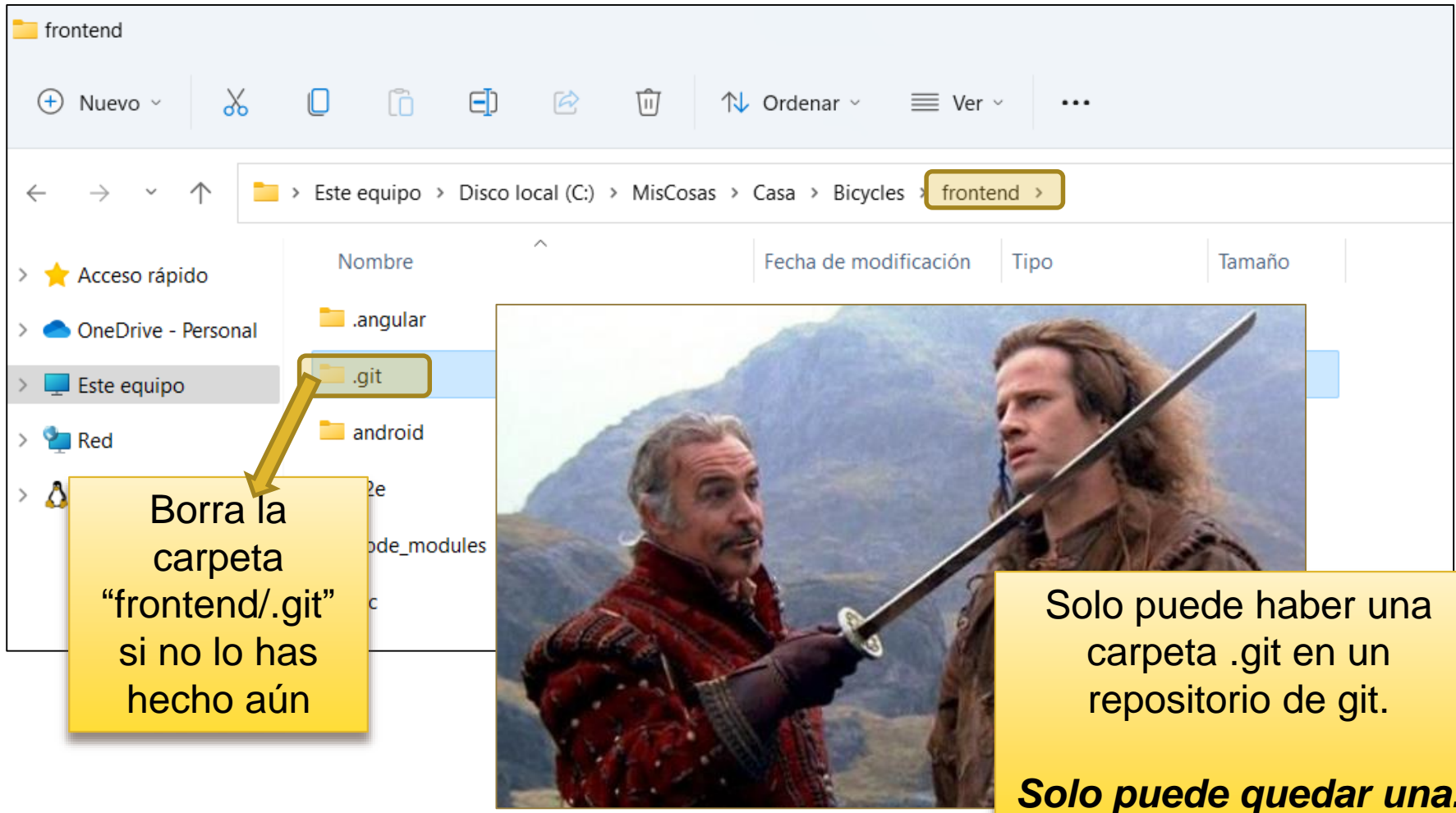
Observa que dentro de la carpeta “frontend” hay una carpeta “.git”

Ionic crea por defecto un repositorio Git. Toda la información de un repositorio Git se encuentra en dicha carpeta .git

Debemos borrar esa carpeta .git porque nosotros queremos que nuestro proyecto tenga tanto el backend como el frontend. Y **solo puede haber una carpeta .git** en el directorio raíz de nuestro proyecto.

Trabajo previo

Solo puede haber una carpeta .git en un repositorio de Git



frontend

Nuevo ▾

Ordenar ▾ Ver ▾

← → ▾ ↑

Este equipo > Disco local (C:) > MisCosas > Casa > Bicycles > frontend >

	Nombre	Fecha de modificación	Tipo	Tamaño
> ★ Acceso rápido				
> OneDrive - Personal				
> Este equipo				
> Red				
>				
	.angular			
	.git			
	android			
	code_modules			

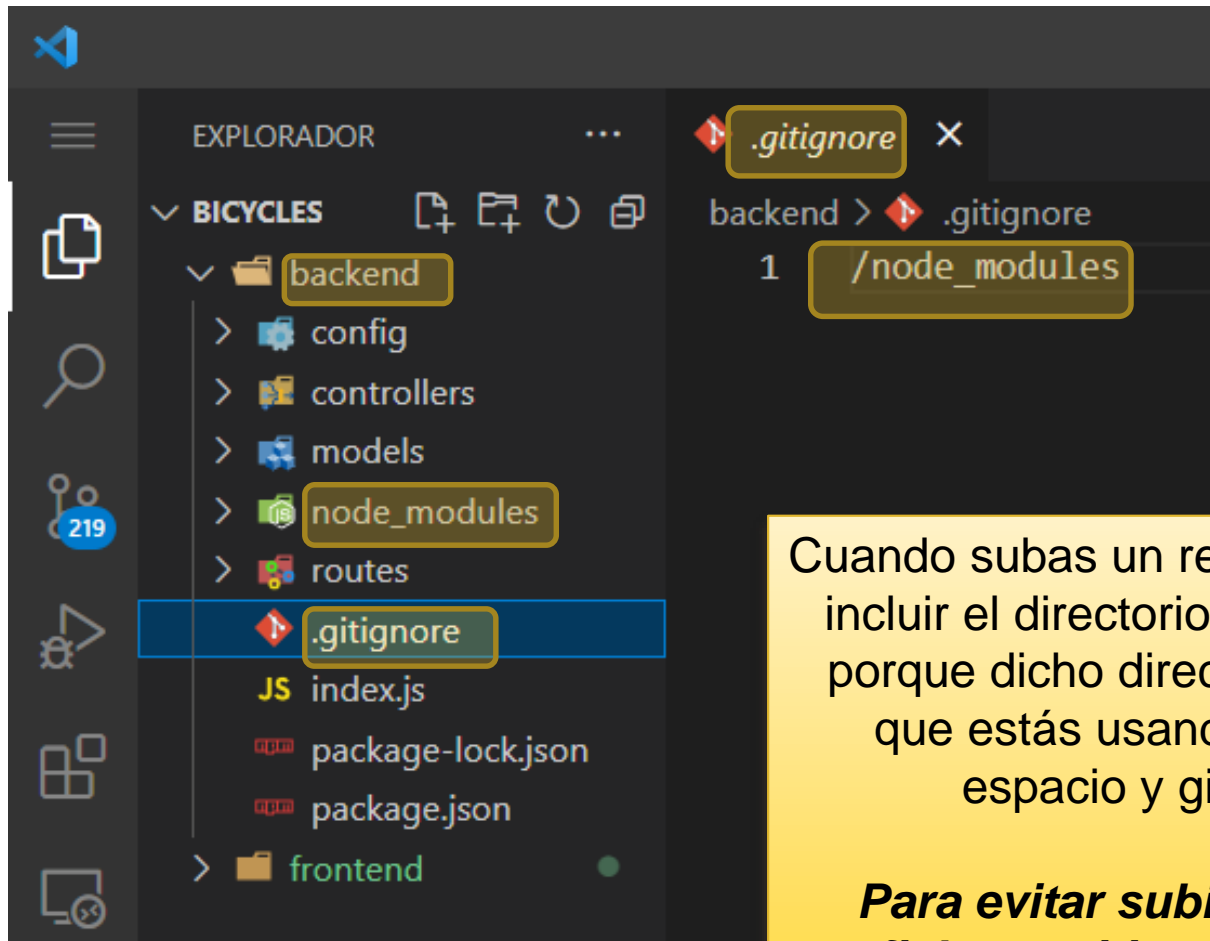
Borra la carpeta "frontend/.git" si no lo has hecho aún

Solo puede haber una carpeta .git en un repositorio de git.

Solo puede quedar una.

Trabajo previo

Crea el fichero backend/.gitignore para no subir la carpeta backend/node_modules



Cuando subas un repositorio a Github no debes incluir el directorio **backend/node_modules** porque dicho directorio contiene las librerías que estás usando, éstas ocupan mucho espacio y github tiene un límite.

Para evitar subir dicha carpeta crea un fichero .gitignore e indica dentro que quieres ignorar dicha carpeta.

Crea el repositorio Git localmente (Por fin se acabo el trabajo previo)

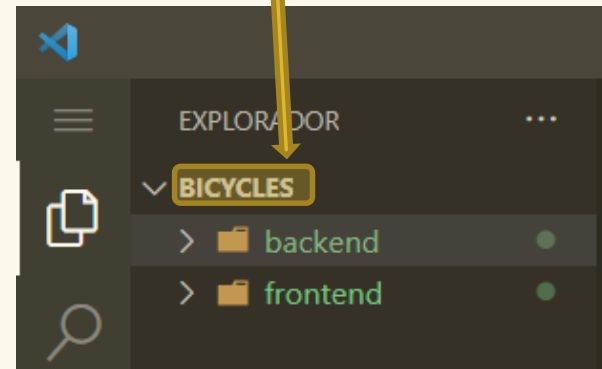
```
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles  
$ git init
```

Ha llegado el momento de crear tu repositorio de git en la carpeta raíz de tu proyecto.

Usa el comando: "git init"

Esto te creará un directorio .git con toda la información de tu repositorio local

Recuerda que debes crear el repositorio en el directorio raíz de tu proyecto que contiene las carpetas backend y frontend



Se ha creado la carpeta .git en la raíz de tu repositorio Git

Antes de seguir asegúrate que en la raíz de tu proyecto hay una carpeta .git, y que en ninguna otra subcarpeta de tu proyecto hay otra carpeta .git

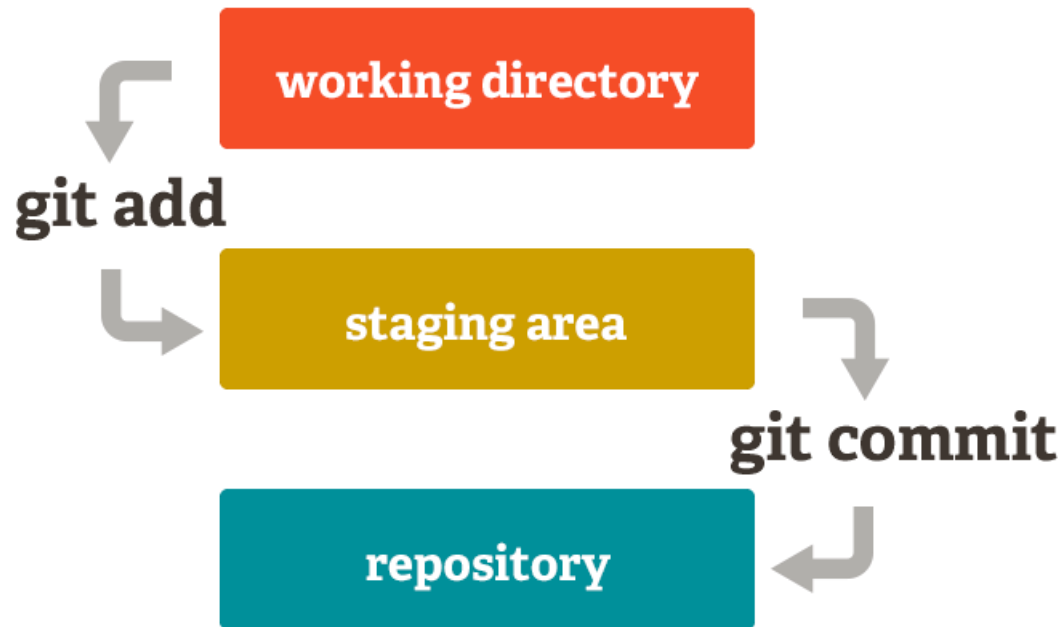


Solo puede haber una carpeta .git en un repositorio de git.

Solo puede quedar una.

Ciclo de vida

Un poco de teoría antes de seguir



El **directorio de trabajo** es en nuestro caso la carpeta Bicycles que contiene las carpetas backend y frontend

La **staging area** es el área en el que iremos añadiendo (con **git add**) los archivos con cambios que se añadirán a nuestro repositorio en el próximo commit que hagamos (con **git commit**)

El **repositorio** es el lugar que contiene toda la información de versiones que hemos ido creando

The image depicts a stage setting. Two heavy, red, vertically-pleated curtains are pulled back to reveal a light gray background. The curtains are tied back with red fabric loops. The floor in the foreground is made of polished wooden planks with a visible grain pattern. The overall scene is clean and minimalist, focusing on the text presented in the center.

“Stage” en inglés es **Escenario**.

En el **“stage”** se colocan los cambios que queremos que formen parte del próximo cambio a confirmar (**commit**).

Añade al stage todo el contenido de la carpeta de tu proyecto

```
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)  
$ git add .
```

Desde la raíz de tu proyecto ejecuta este comando para añadir todo el proyecto al “stage”

No te olvides del punto. Es “git add .”

Observa que ahora te aparece que se ha creado la rama “master”

git commit

Crea un commit con el contenido de lo que hay en el “stage”

```
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)  
$ git commit -m "first commit"
```

Lo que anteriormente
añadiste al “**stage**” pasa
ahora al **repositorio local**.

Con la opción **-m** indicas un
asunto que describe en qué
consiste el cambio realizado

Observa que
estás en la rama
“master”

git commit

(Atención la primera vez)

```
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)
$ git commit -m "first commit"
Author identity unknown

*** Please tell me who you are.

Run

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'tibur@DESKTOP-02362TM.(none)')

tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)
$ git config --global user.email "tiburcio.cruz@gmail.com"

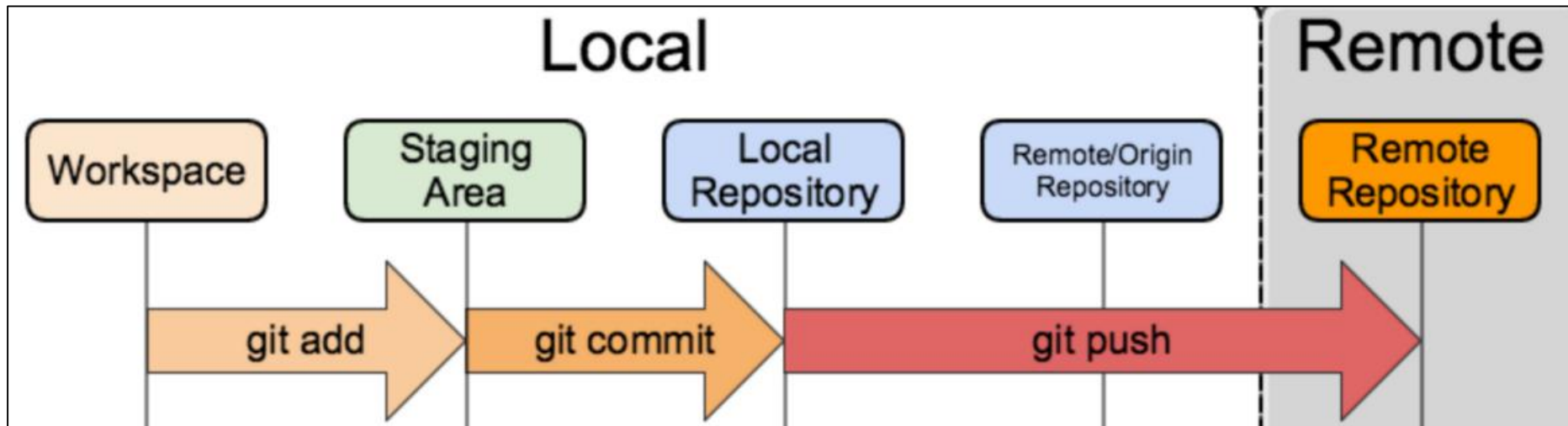
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)
$ git config --global user.name "tcrurav"
```

La primera vez que haces un commit hay que indicar quién hace el commit. Por eso te sale este mensaje

Con **git config** indicamos el email y el nombre de la persona que realiza el commit.

git push

(subir los últimos commits al repositorio remoto)



El “**Workspace**”, el “**stage**” y el “**repositorio local**” se encuentra localmente.

El comando “**git push**” se usa para subir al repositorio remoto los commits que aún no se han subido.

git remote

¿A dónde subimos los últimos commit?

```
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)  
$ git remote add origin https://github.com/tcrurav/BicyclesIonicExpressSequelize.git
```

origin es el nombre que le hemos dado al repositorio remoto cuya URL es la que se indica a continuación

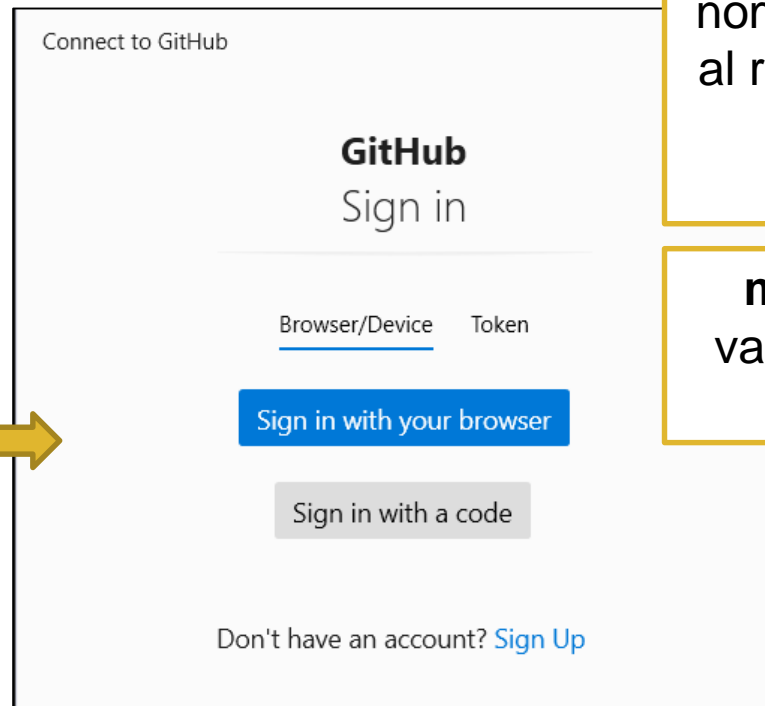
Antes de poder hacer el “**git push**” para subir los últimos commit al repositorio remoto hay que indicar dónde se encuentra dicho repositorio remoto. Eso se consigue con un “**git remote**”

git push

(para subir al repositorio remoto los últimos commit realizados localmente)

```
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)  
$ git push origin master
```

Con **git push** subimos al repositorio remoto los últimos commit realizados localmente. La primera vez te pedirá las credenciales de Github.



origin es simplemente el nombre que le hemos dado al repositorio remoto con el comando **git remote** anteriormente

master es la rama que vamos subir al repositorio remoto

Administrador de credenciales (Windows)

En Windows se guardan las credenciales en el Administrador de Credenciales

**Panel de Control →
Administrador de
Credenciales**

Puede ser que necesites borrar dichas credenciales si estás usando el PC del centro y no quieres que alguien use tus credenciales.


The screenshot shows the Windows Credential Manager window. The title bar reads 'Administrador de credenciales'. The breadcrumb navigation path is 'Panel de control > Todos los elementos de Panel de control > Administrador de credenciales'. The main heading is 'Administrar credenciales' with a subtitle 'Vea y elimine su información de inicio de sesión guardada para sitios web, redes y aplicaciones conectadas.' There are two main sections: 'Credenciales web' and 'Credenciales de Windows'. Below these are links for 'Copia de seguridad de credenciales' and 'Restaurar credenciales'. The 'Credenciales de Windows' section shows 'No hay credenciales de Windows.' with a link to 'Agregar una credencial de Windows'. The 'Credenciales basadas en certificados' section shows 'No hay certificados.' with a link to 'Agregar una credencial basada en certificado'. The 'Credenciales genéricas' section shows a list of credentials, including 'vscodevscode.github-authentication/github.auth' with a date of '25/05/2022' and 'git:https://github.com' with a date of '22/09/2022'.


Administrador de credenciales

Panel de control > Todos los elementos de Panel de control > Administrador de credenciales

Administrar credenciales

Vea y elimine su información de inicio de sesión guardada para sitios web, redes y aplicaciones conectadas.

 **Credenciales web**

 **Credenciales de Windows**

[Copia de seguridad de credenciales](#) [Restaurar credenciales](#)

Credenciales de Windows [Agregar una credencial de Windows](#)

No hay credenciales de Windows.

Credenciales basadas en certificados [Agregar una credencial basada en certificado](#)

No hay certificados.

Credenciales genéricas [Agregar una credencial genérica](#)

vscodevscode.github-authentication/github.auth	Fecha de modificación: 25/05/2022
git:https://github.com	Fecha de modificación: 22/09/2022

Se ha subido tu último commit a tu repositorio remoto. ¡Llujuuuu!

github.com/tcrurav/BicyclesIonicExpressSequelize

Search or jump to... / Pulls Issues Marketplace Explore

tcrurav / BicyclesIonicExpressSequelize Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights

master Go to file Add file Code About

tcrurav first commit	1 minute ago	2
backend first commit	17 minutes ago	
frontend first commit	1 minute ago	

Help people interested in this repository understand your project by adding a README. Add a README

No description provided.

0 stars 1 watching 0 forks

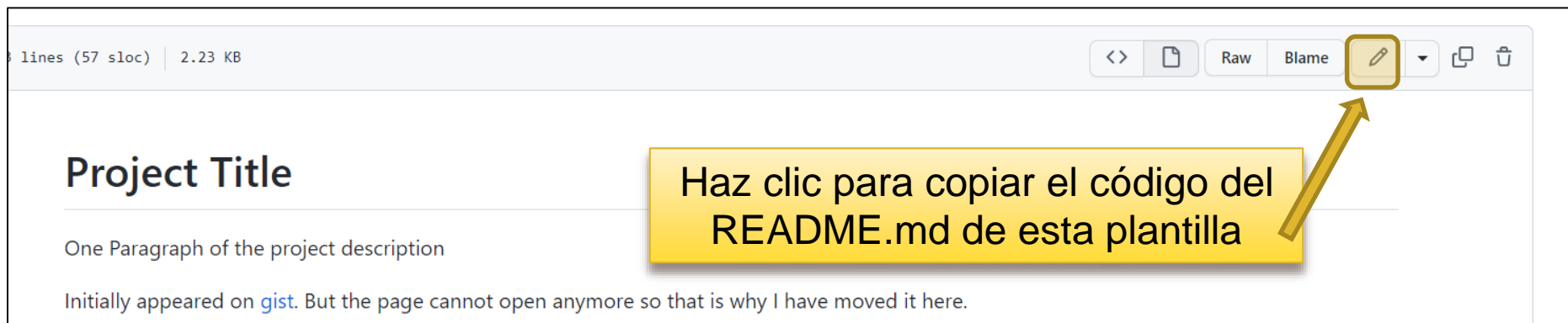
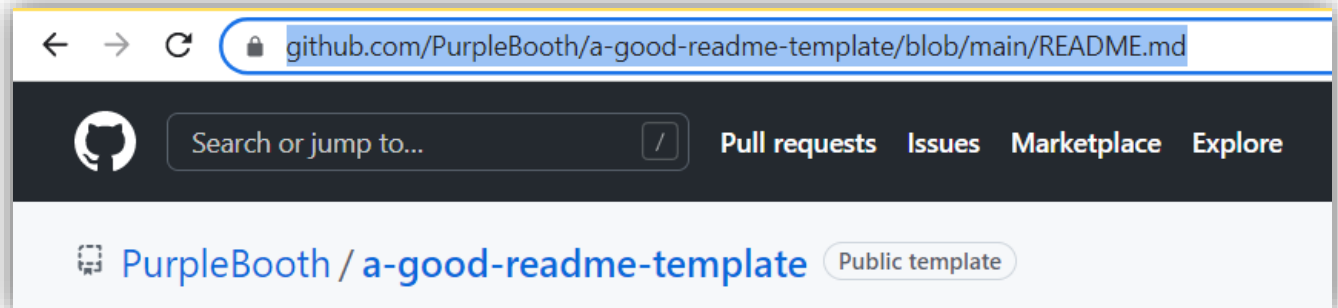
Releases

No releases published

Si haces **F5** en tu navegador verás que el repositorio remoto contiene ahora tu último commit.

Crea un README.md

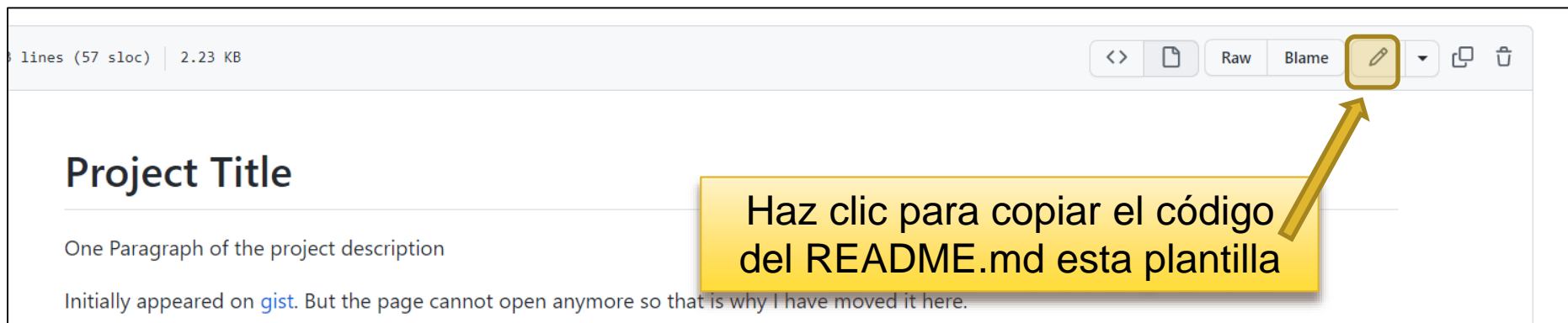
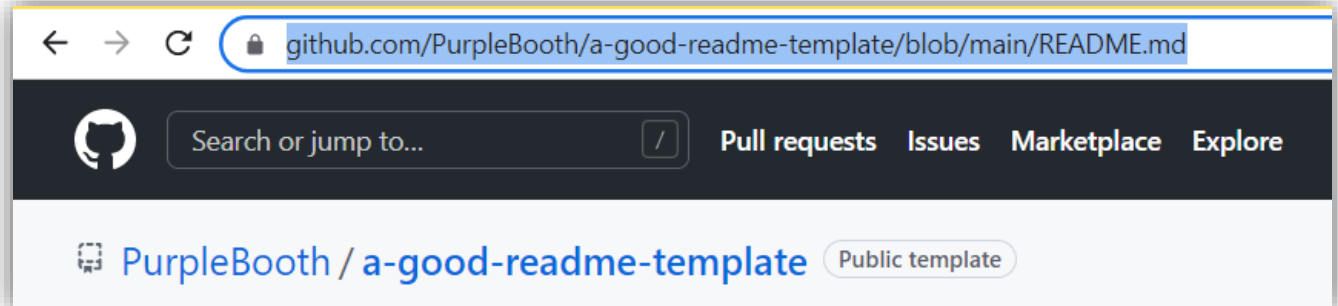
En este repositorio
puedes encontrar una
buena plantilla para
crear tu README.md



Haz clic para copiar el código del
README.md de esta plantilla

Crea un README.md

En este repositorio
puedes encontrar una
buena plantilla para
crear tu README.md



Haz clic para copiar el código
del README.md esta plantilla

README.md

Como puedes ver los archivos README.md son archivos con un tipo de lenguaje de marcas

← → ↻ raw.githubusercontent.com/PurpleBooth/a-go

Project Title

One Paragraph of the project description

Initially appeared on
[gist](https://gist.github.com/PurpleBooth/109311bb0361f32d87a2). But the page

Getting Started

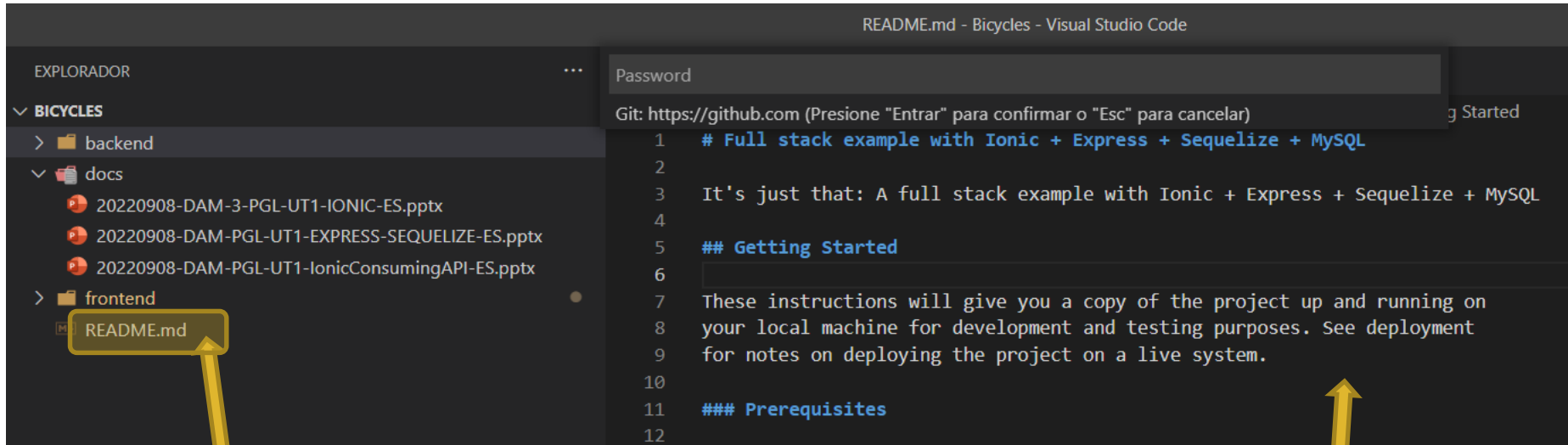
These instructions will give you a copy of the project up and running on your local machine for development and testing purposes. See deployment for notes on deploying the project on a live system.

Prerequisites

Requirements for the software and other tools to build, test and push

- [Example 1](https://www.example.com)
- [Example 2](https://www.example.com)

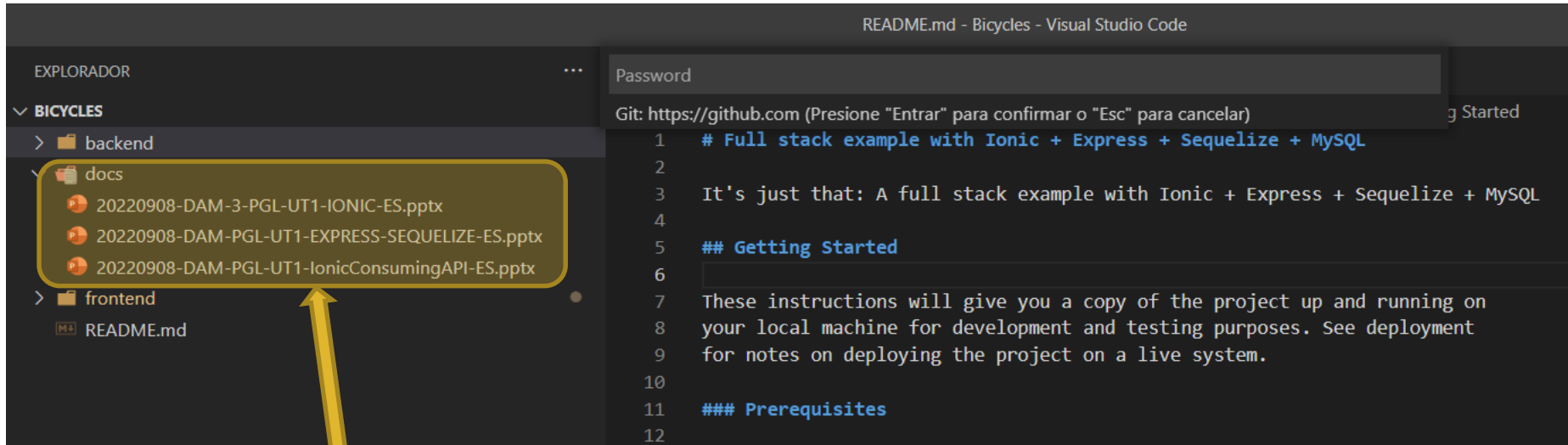
README.md



Añade a tu proyecto el
archivo README.md
en el directorio raíz.

A continuación pega el código de la
plantilla de PurpleBooth y modifícalo
para que tenga sentido para tu proyecto

README.md



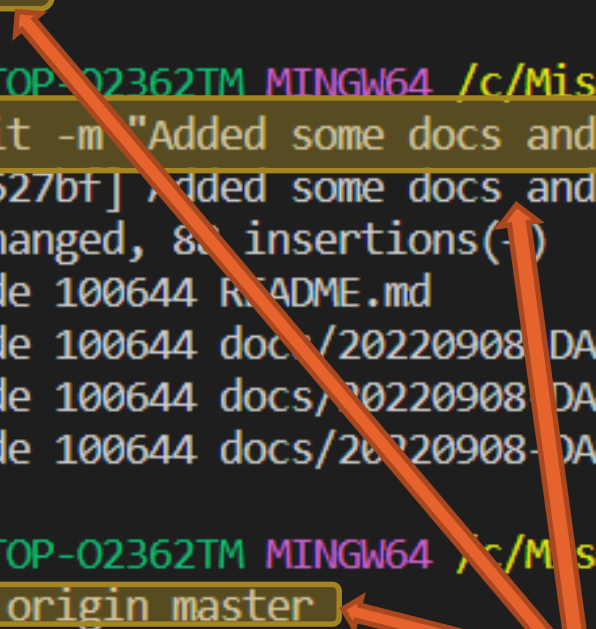
Yo he aprovechado
además para meter los
tutoriales que hemos
estado trabajando en
clase

README.md

```
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)
$ git add .

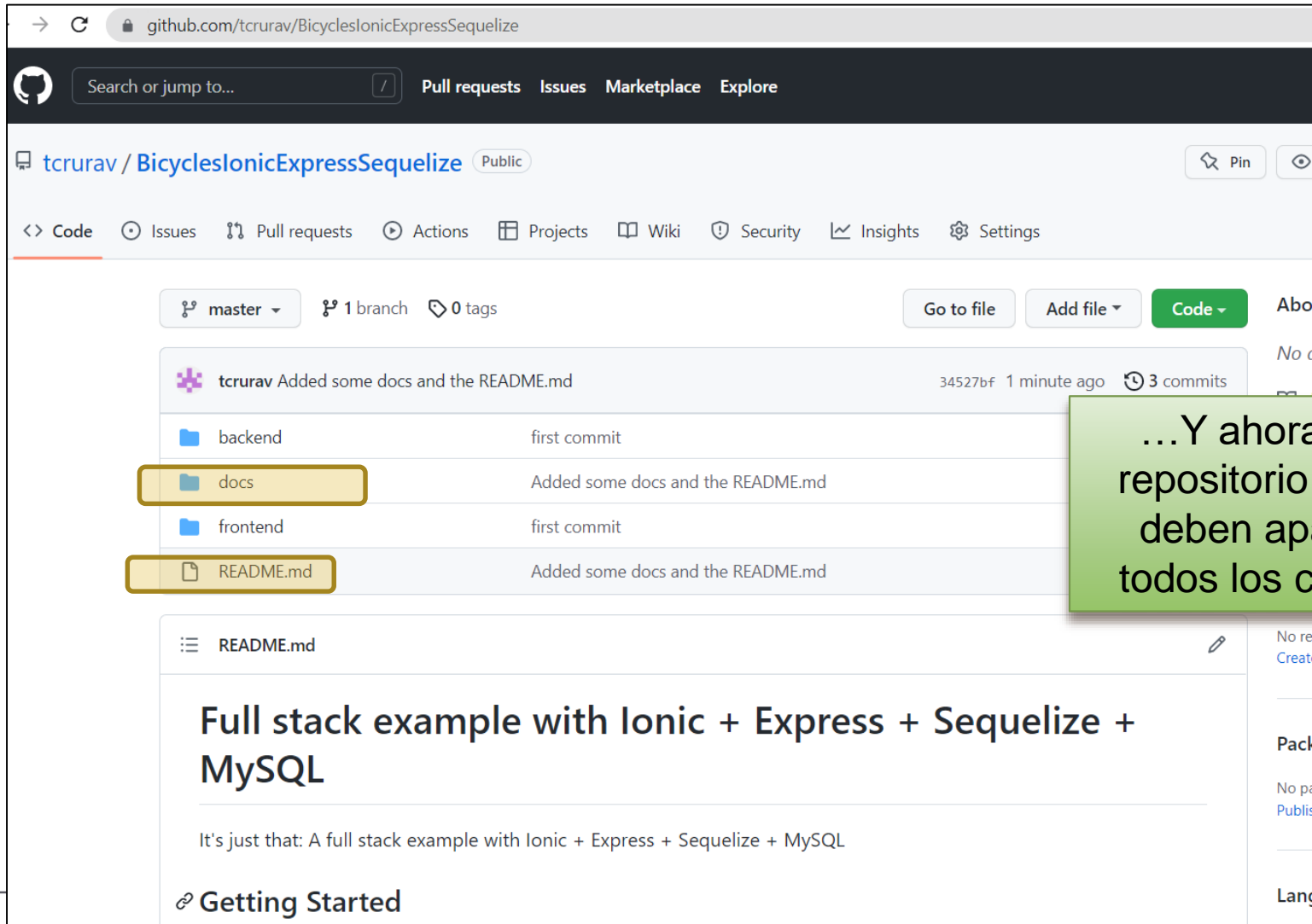
tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)
$ git commit -m "Added some docs and the README.md"
[master 34527bf] Added some docs and the README.md
4 files changed, 80 insertions(-)
create mode 100644 README.md
create mode 100644 docs/20220908-DAM-3-PGL-UT1-IONIC-ES.pptx
create mode 100644 docs/20220908-DAM-PGL-UT1-EXPRESS-SEQUELIZE-ES.pptx
create mode 100644 docs/20220908-DAM-PGL-UT1-IonicConsumingAPI-ES.pptx

tibur@DESKTOP-02362TM MINGW64 /c/MisCosas/Casa/Bicycles (master)
$ git push origin master
```



Si vuelves a hacer un **git add**, **git commit** y **git push** tendrás en el repositorio remoto los nuevos cambios.

Se han subido los cambios. ¡Llujuuuuu!



github.com/tcrurav/BicyclesIonicExpressSequelize

Search or jump to... Pull requests Issues Marketplace Explore

tcrurav / BicyclesIonicExpressSequelize Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code About

tcrurav Added some docs and the README.md 34527bf 1 minute ago 3 commits

backend	first commit
docs	Added some docs and the README.md
frontend	first commit
README.md	Added some docs and the README.md

README.md

Full stack example with Ionic + Express + Sequelize + MySQL

It's just that: A full stack example with Ionic + Express + Sequelize + MySQL

Getting Started

...Y ahora en tu repositorio remoto deben aparecer todos los cambios

Acabamos de comprobar que los cambios se han subido.

Fíjate que para añadir cambios al repositorio local y luego subir estos cambios al repositorio remoto consiste en ejecutar:

```
git add .  
git commit -m “asunto”  
git push origin master
```

Conclusiones

¿Qué hemos aprendido?

- Hemos creado una cuenta en Github.
- Hemos creado un repositorio remoto vacío en Github.
- Hemos creado un repositorio local.
- Hemos subido el repositorio local al repositorio remoto.

Próximos pasos...

- Tienes que aprender a clonar un proyecto. (**git clone**)
 - Tienes que aprender a bajar cambios a un repositorio local desde un repositorio remoto. (**git pull**)
 - Es interesante que aprendas a trabajar con ramas de git.
 - Un buen sistema es el de la siguiente web:
<https://nvie.com/posts/a-successful-git-branching-model/>
-