



El futuro digital
es de todos

MinTIC

```
te( "name" );  
"type" );
```

```
if ( type == "sprite" )  
  
std::string item_name = item->Attribute( "name" );  
std::string spritename = item->Attribute( "spritename" );  
float x = boost::lexical_cast<float>( item->Attribute( "x" ) );  
float y = boost::lexical_cast<float>( item->Attribute( "y" ) );  
float offset = boost::lexical_cast<float>( item->Attribute( "offset" ) );  
  
SpriteDescList::iterator sp = sprite_descs.begin();  
for( ; sp != sprite_descs.end(); ++sp )  
    if ( sp->name_ == spritename )  
        break;
```

Ciclo 3:

Desarrollo de Software



Misión
TIC2022

VERSIÓN 1.0

Unidad de educación
continua y permanente
Facultad de Ingeniería



Unidad Camilo Torres
Calle 44 # 45-67
Bloque B5 piso 1



(57) + 316 5000
uec_ibog@unaleduco

GIT y Bitbucket

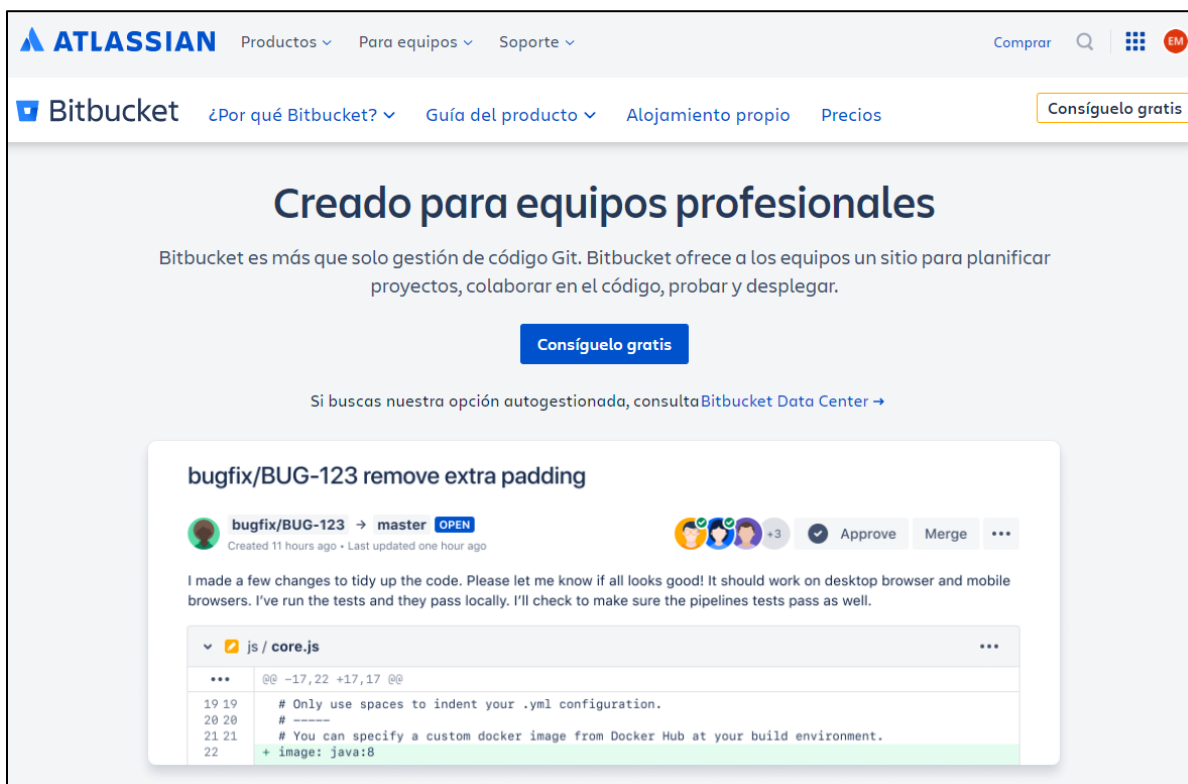
Actividad Práctica

En sesiones anteriores se explicaron los temas relacionados con SCRUM, Jira y los Sistemas de Control de Versiones. En esta actividad se busca afianzar los conceptos de GIT y relacionarlos e integrarlos con las metodologías ágiles. Para ello, se propone el desarrollo de un ejercicio para aprender a utilizar Bitbucket junto con su integración con Jira, y para realizar una práctica con los conceptos de GIT vistos. El ejercicio consiste en la creación de un repositorio en Bitbucket, su integración con Jira, y el control local del repositorio remoto con GIT.

Vinculación de Bitbucket con la cuenta de Atlassian

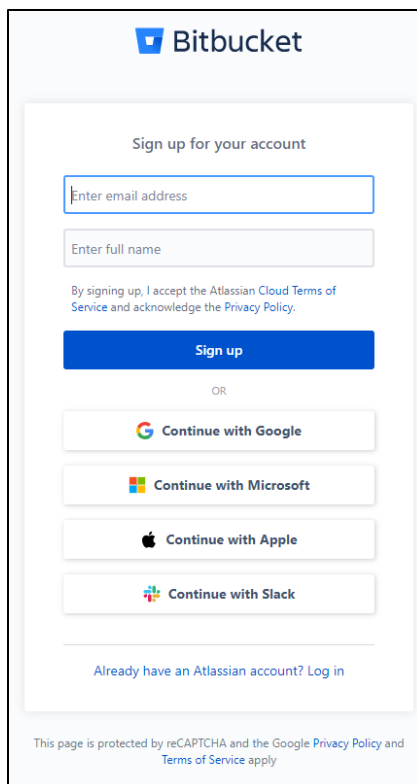
Bitbucket es la plataforma de hosting de repositorios de Atlassian, por ello, es necesario disponer de una cuenta de Atlassian.

Para vincular una cuenta de Atlassian con Bitbucket se debe acceder a la página oficial de Bitbucket: <https://www.atlassian.com/es/software/bitbucket> y elegir la opción *consíguelo gratis*:



The screenshot shows the Bitbucket website interface. At the top, there's the Atlassian logo and navigation links. Below that, the Bitbucket logo is followed by links like '¿Por qué Bitbucket?', 'Guía del producto', 'Alojamiento propio', and 'Precios'. A prominent 'Consíguelo gratis' button is visible. The main heading reads 'Creado para equipos profesionales'. Below this, a description states that Bitbucket offers more than just Git code management, including project planning, collaboration, testing, and deployment. A 'Consíguelo gratis' button is repeated. A link to 'Bitbucket Data Center' is also present. The lower section displays a sample pull request titled 'bugfix/BUG-123 remove extra padding'. It shows the pull request details, including the source branch 'bugfix/BUG-123' and the target branch 'master'. The pull request description mentions changes to tidy up the code. Below the description, a code diff is shown for the file 'js / core.js', highlighting changes to the Dockerfile configuration.

Esto redirecciona a otra página donde se solicita el registro. Allí se indica que ya se tiene una cuenta de Atlassian y se inicia sesión en la cuenta de Atlassian:



Bitbucket

Sign up for your account

Enter email address

Enter full name

By signing up, I accept the Atlassian Cloud Terms of Service and acknowledge the Privacy Policy.

Sign up

OR

Continue with Google

Continue with Microsoft


Continue with Apple

Continue with Slack

Already have an Atlassian account? [Log in](#)

This page is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply

Una vez se ha iniciado sesión, se solicita ingresar un nombre de usuario. Se ingresa y se selecciona *continue*:



Almost done

Create a username for Bitbucket Cloud


Email
ricava8916@gocasin.com

Username *

Continue

[Return to log in](#)

Allí se encuentran cuatro preguntas que se saltan con el botón *Skip*, para así terminar de vincular la cuenta con Bitbucket.



Before we drop you in Bitbucket, help us tailor your experience

What describes your experience with source control?

How many people do you think you'll work with on Bitbucket?

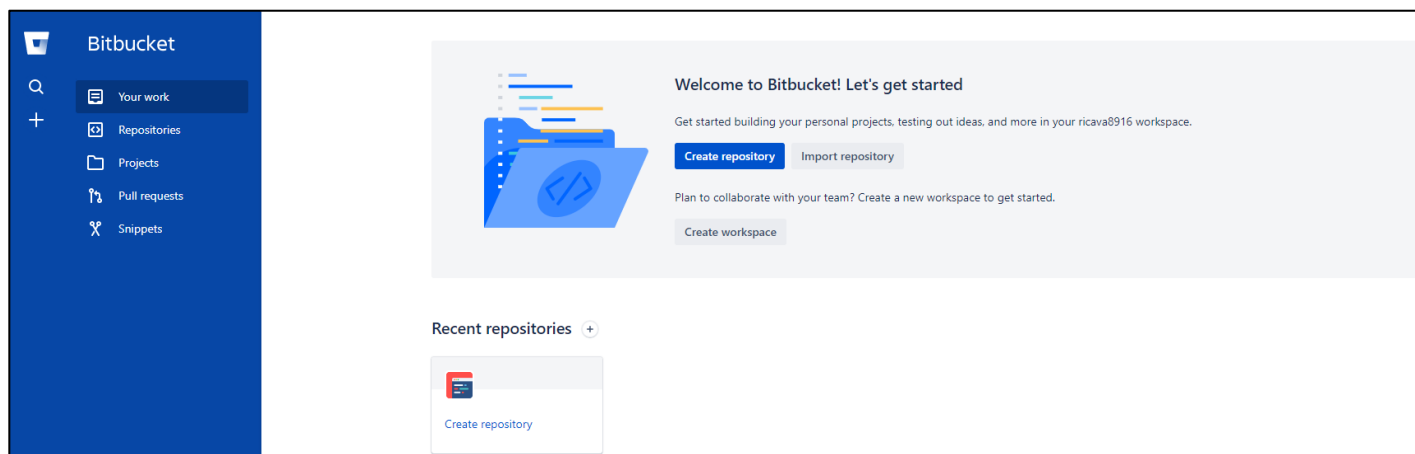
What best describes your role?

What best describes how you plan to use Bitbucket?

[Skip](#) [Submit](#)

Creación de un repositorio en Bitbucket

En la página principal de Bitbucket: <https://bitbucket.org/dashboard/overview>, en la sección *Your Work*, se presiona el botón **+** que se encuentra junto al texto *Recent repositories*:



Se redirige a un formulario para la creación de un repositorio. Allí se ingresa el nombre del proyecto y el nombre del repositorio, y se pulsa el botón *Create repository* para crear el repositorio:

Create a new repository

[Import repository](#)

Workspace

Emiliano Martinez

Project name*

Repository name*

Access level

☒ Private repository

Uncheck to make this repository public. Public repositories typically contain open-source code and can be viewed by anyone.

Include a README?

Yes, with a tutorial (for beginne...)

Default branch name

e.g., 'main'

Include .gitignore?

Yes (recommended)

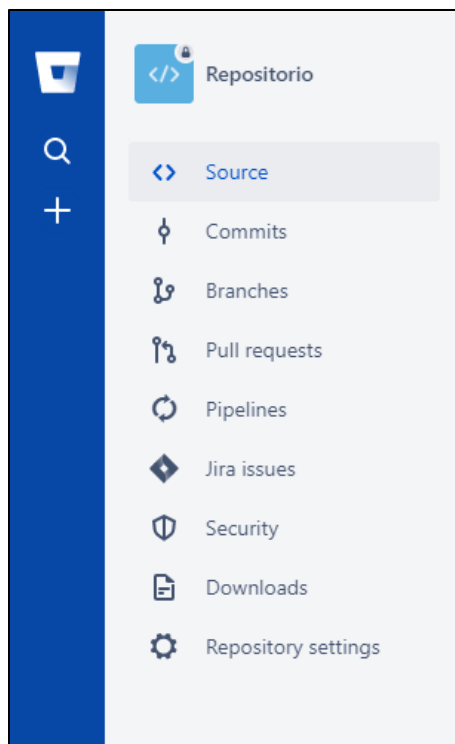
[Advanced settings](#)

Create repository

Cancel

Interfaz del repositorio

Una vez se ha creado el repositorio, se puede ver la interfaz con la cual se realizará su administración:



No se utilizarán algunas de las secciones que se muestran, sin embargo, aquellas que se utilizarán se explican a continuación:

Source: en esta sección se puede interactuar con el repositorio y sus archivos por medio de acciones como explorar los archivos, visualizar las ramas, clonar el repositorio, o buscar, crear y editar archivos en cualquier rama.

Repositorio

Source

Commits

Branches

Pull requests

Pipelines

Jira issues

Security

Downloads

Repository settings

Emiliano Martinez / Repo

Repositorio

Clone

master

Files

Filter files

/

Name	Size	Last commit	Message
.gitignore	624 B	20 minutes ago	Initial commit
README.md	2.56 KB	20 minutes ago	Initial commit

README.md

Edit a file, create a new file, and clone from Bitbucket in under 2 minutes

When you're done, you can delete the content in this README and update the file with details for others getting started with your repository.

We recommend that you open this README in another tab as you perform the tasks below. You can watch our video for a full demo of all the steps in this tutorial. Open the video in a new tab to avoid leaving Bitbucket.

Commits: en esta sección se pueden visualizar los commits y su información, y se pueden filtrar de acuerdo con la rama en la que fueron creados.

Repositorio

Source

Commits

Branches

Pull requests

Pipelines

Jira issues

Security

Downloads

Repository settings

Emiliano Martinez / Repo / Repositorio

Commits

Clone

Search commits

All branches

Author	Commit	Message	Date
Emiliano Martinez	711d48e	Initial commit	24 minutes ago

Branches: en esta sección se pueden ver, crear, comparar y eliminar las ramas del repositorio.

Repositorio

Source

Commits

Branches

Pull requests

Pipelines

Jira issues

Security

Downloads

Repository settings

Emiliano Martinez / Repo / Repositorio

Branches

Create branch

Search branches


Active branches

Branch type

Branch	Behind	Ahead	Updated	Pull request	Builds	Actions
master MAIN DEVELOPMENT			25 minutes ago			

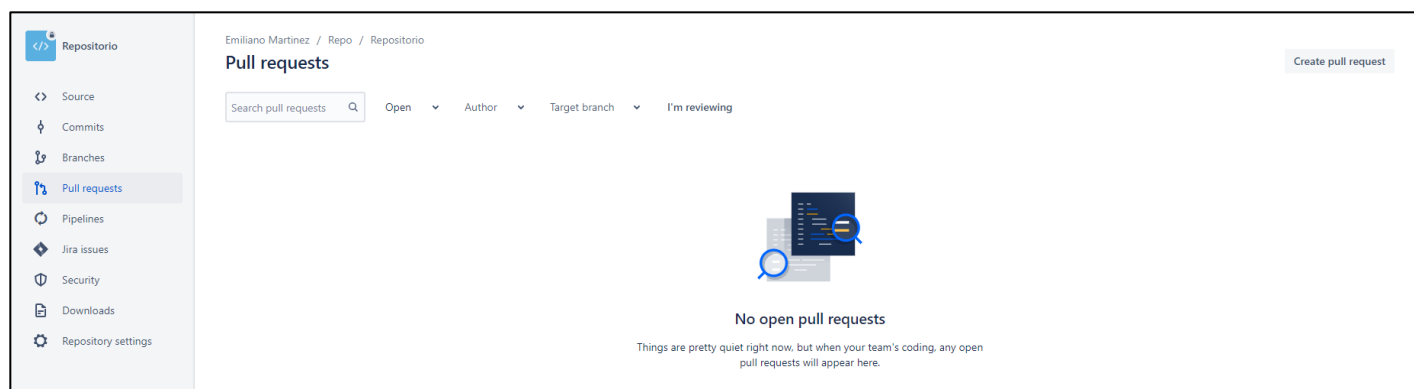
Pull requests: en esta sección aparecerán las *pull request* que se realicen para hacer *merge* entre dos ramas.

Facultad de
INGENIERÍA
Sede Bogotá



UNIVERSIDAD
NACIONAL
DE COLOMBIA

pág 6 de 26

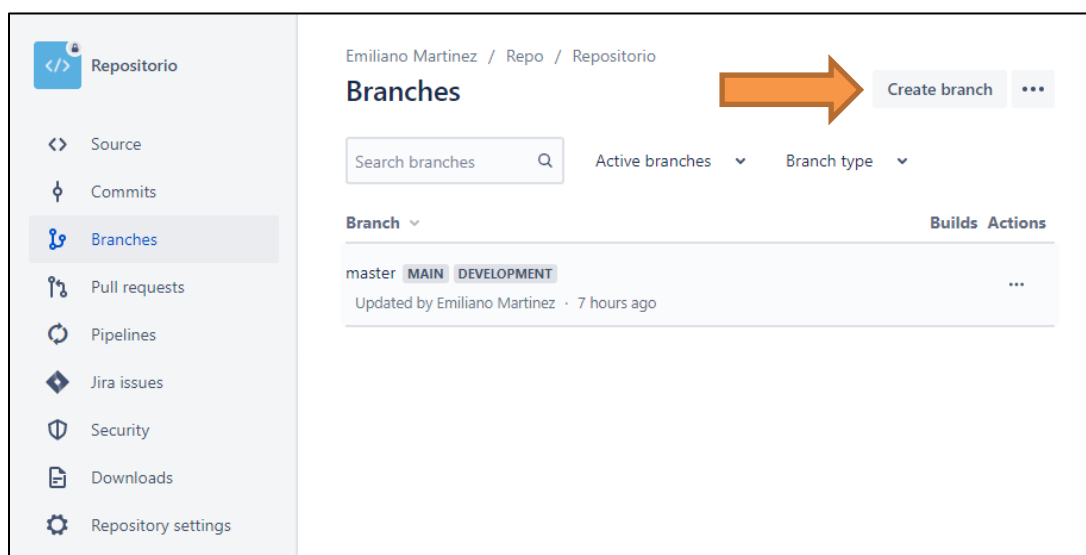


Se recomienda la exploración de la interfaz por parte del estudiante para que se familiarice con cada sección.

Uso del repositorio

Para administrar un repositorio y sus ramas se puede utilizar GIT o la interfaz de la plataforma de hosting en la que se encuentra el repositorio, en este caso, la interfaz de Bitbucket. A continuación, se realiza un ejercicio en el cual se manipula el repositorio por medio de la interfaz de Bitbucket para que el estudiante tenga como referencia.

Para empezar, es necesario crear una nueva rama en el repositorio. Para ello, en la sección [Branches](#) se pulsa el botón [Create branch](#), el cual abre un modal en el cual se indica el nombre y la rama base de la nueva rama, y se pulsa el botón [Create](#). En este ejercicio se van a nombrar las ramas de tal forma que el nombre indique el avance del trabajo de la rama. Ya que esta es la primera rama del repositorio donde se hará el avance inicial se llamará [Avance_0](#).



Create branch

Type 1

Other

From branch

master

Branch name

Avance_0

Create Cancel

Una vez se crea la rama, se añade un archivo a la nueva rama. Para ello, en la sección [Source](#) se cambia a la nueva rama.

Emiliano Martinez / Repo

Repositorio

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to your repository](#).

master

Filter branches

Branches Tags

master

[Avance_0](#)

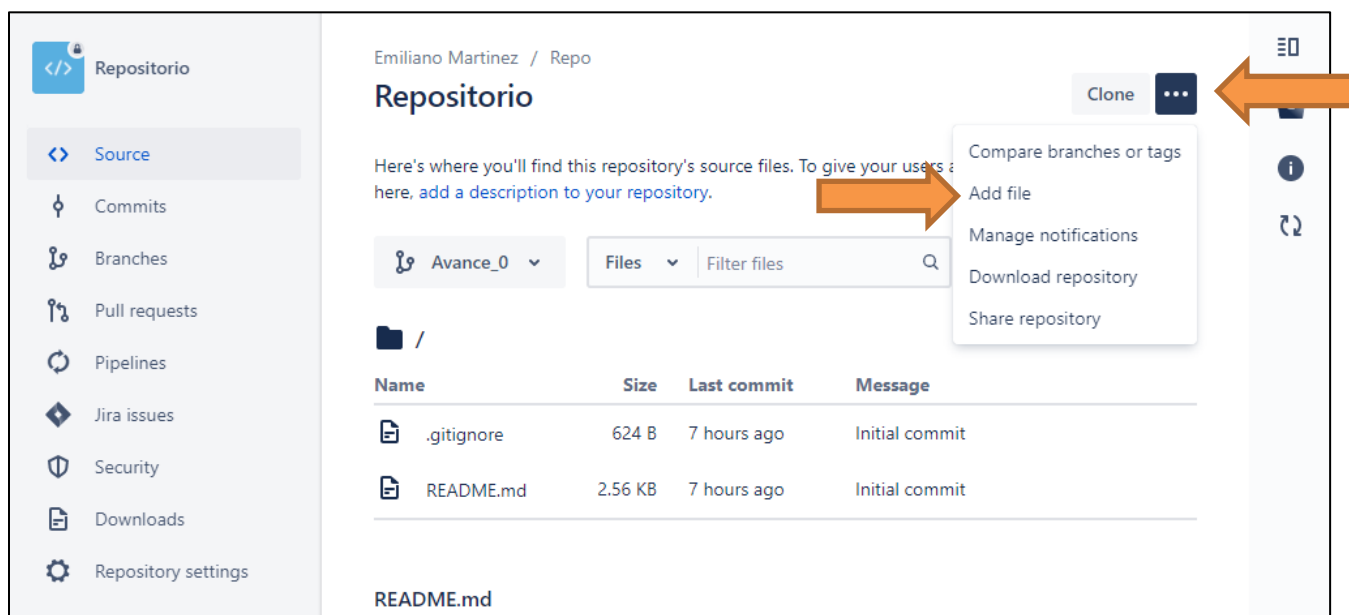
README.md

Edit a file, create a new file, and clone from Bitbucket in under 2 minutes

When you're done, you can delete the content in this README and update the file with details for others getting started with your repository.

We recommend that you open this README in another tab as you perform the tasks below. You can [watch our video](#) for a full demo of all the steps in this tutorial. Open the video in a new tab to avoid leaving Bitbucket

Posteriormente, en los tres puntos que se encuentran en la parte superior derecha, se selecciona [add file](#).



Emiliano Martinez / Repo

Repositorio

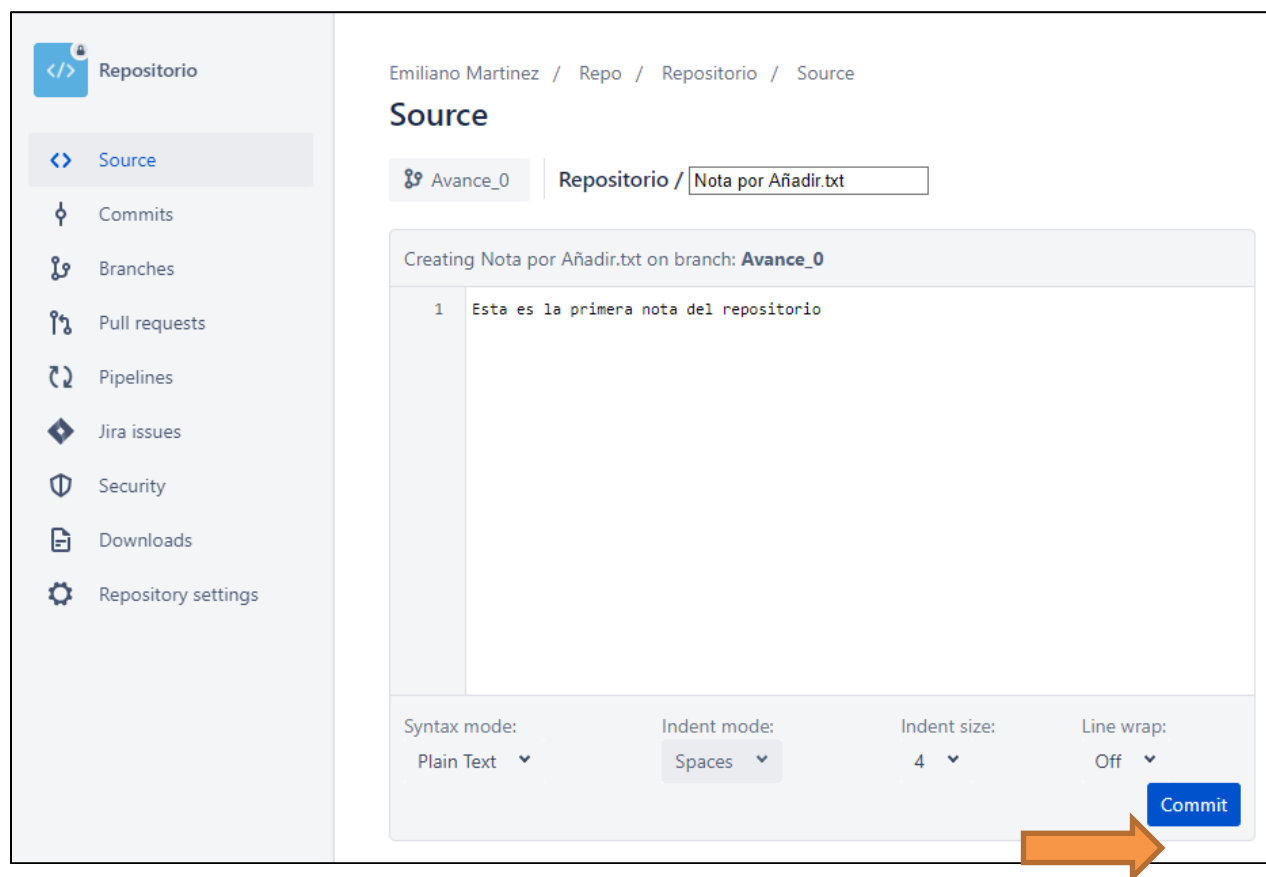
Here's where you'll find this repository's source files. To give your users a better experience here, [add a description to your repository](#).

Avance_0 Files Filter files

Name	Size	Last commit	Message
.gitignore	624 B	7 hours ago	Initial commit
README.md	2.56 KB	7 hours ago	Initial commit

README.md

Allí se ingresa el nombre del archivo y el contenido del archivo, y se presiona el botón **Commit**.



Emiliano Martinez / Repo / Repositorio / Source

Source

Avance_0 Repositorio / Nota por Añadir.txt

Creating Nota por Añadir.txt on branch: Avance_0

```
1 Esta es la primera nota del repositorio
```

Syntax mode: Plain Text Indent mode: Spaces Indent size: 4 Line wrap: Off

Commit

En el modal emergente se personaliza el mensaje, y se presiona el botón [Commit](#).


Commit changes

Commit message

Nota creada online con Bitbucket

☐ Create a pull request for this change

Commit



Una vez se ha creado el commit, se realiza el merge entre las dos ramas del repositorio. Para ello, se vuelve a la sección [Branches](#) y se pulsa los tres botones que se encuentran a la derecha del nombre de la nueva rama.

Repositorio

Source
Commits
Branches
Pull requests
Pipelines
Jira issues
Security
Downloads
Repository settings

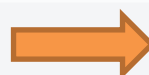
Emiliano Martinez / Repo / Repositorio

Branches

Create branch ...

Search branches
Active branches
Branch type

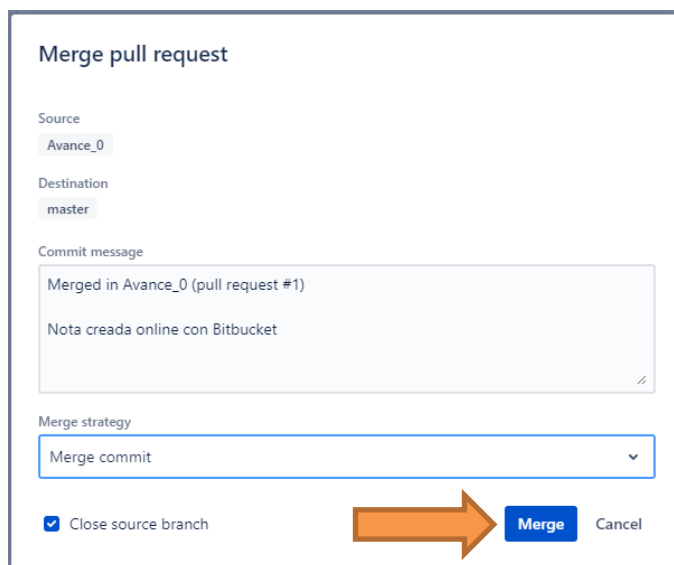
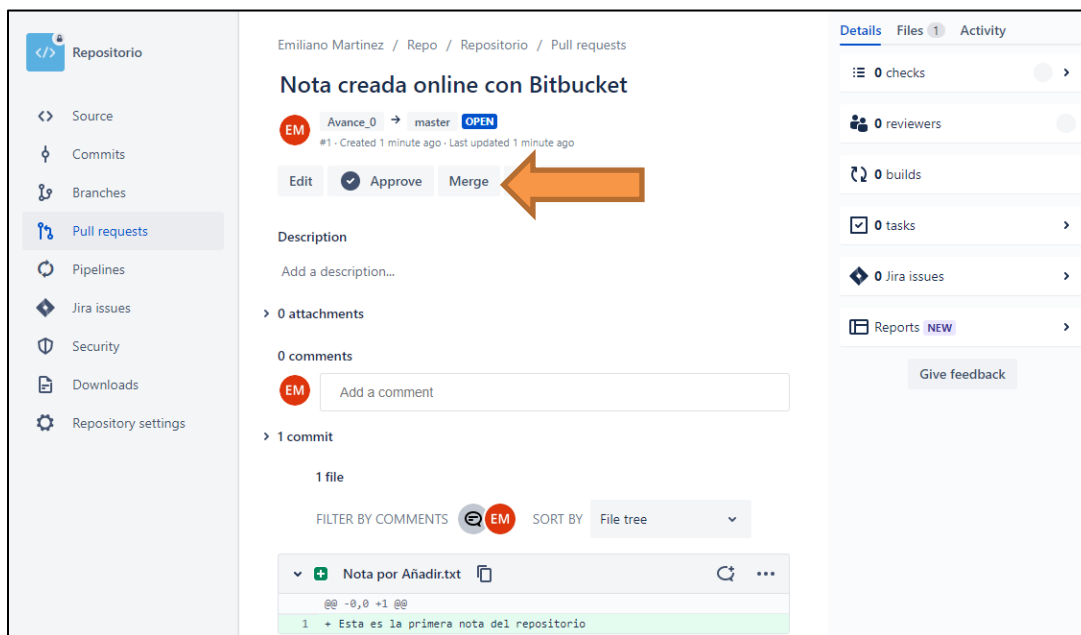
Branch	Builds	Actions
master MAIN DEVELOPMENT Updated by Emiliano Martinez · 8 hours ago		...
Avance_0 Updated by Emiliano Martinez · 3 minutes ago		...



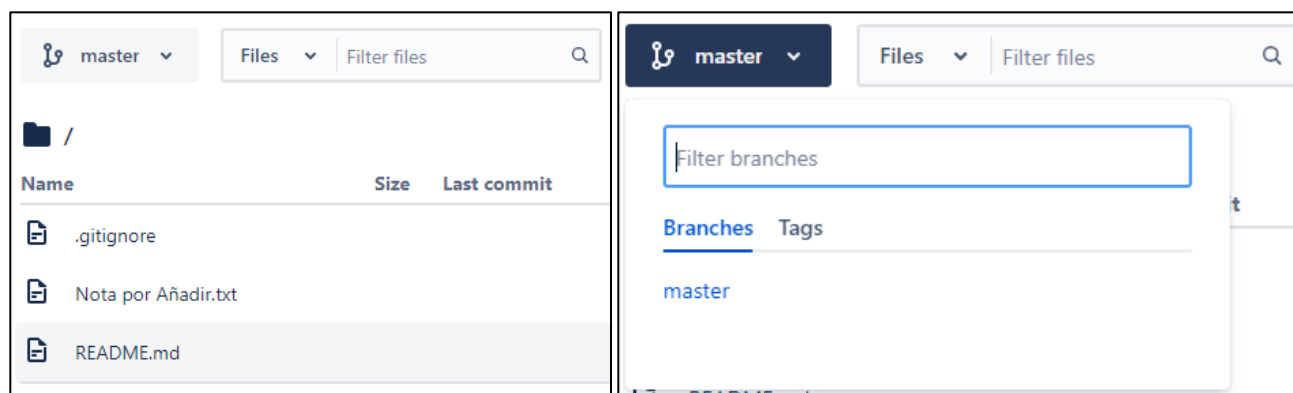
En el menú emergente, se selecciona [Create pull request](#).

En la nueva ventana se muestra que se hará un merge entre la rama *Avance_0* y la rama *Master* y que los cambios se guardarán en la rama *Master*. Esto se puede cambiar, si se desea. Ahora, se marca la casilla que indica que se borrará la rama *Avance_0* cuando se haga el merge, y se presiona el botón *Create pull request*.

Finalmente, se presiona el botón **Merge** en las dos ventanas siguientes.



Si se dirige a la sección Source, se puede ver que el archivo añadido a la rama **Avance_0** ahora se encuentra en la rama **Master**, y que la rama **Avance_0** fue eliminada.



Instalación de GIT

De acuerdo con el sistema operativo que se tenga, GIT se instala y utiliza de una u otra forma. Por lo tanto, se deben seguir los pasos de instalación indicados para Windows, Linux o MacOS según se requiera.

Windows:

1. Se descarga el instalador de GIT para Windows más reciente desde la página oficial de GIT <https://git-scm.com/downloads>
2. Cuando se haya iniciado correctamente el instalador, se debe ver la pantalla del asistente de configuración de GIT. Se seleccionan las opciones Next (Siguiente) y Finish (Finalizar) para completar la instalación. Las opciones predeterminadas son las más lógicas en la mayoría de los casos.
3. Una vez instalado, para comprobar que la instalación se hizo correctamente, se presiona clic derecho sobre el escritorio y se asegura que en el menú contextual se encuentra la opción *GIT bash here*. Esta es la consola en la que se utilizan los comandos de GIT.

Linux (Debian/Ubuntu)

1. Desde el shell de Linux, se instala GIT mediante apt-get con los comandos: *sudo apt-get update*, y *sudo apt-get install GIT*.
2. Una vez instalado, se utiliza en el shell el comando *GIT --version* para comprobar que la instalación se hizo correctamente. A partir de ahora, todos los comandos de GIT se utilizarán en la consola de Linux.

MacOS:

1. Se descarga el instalador de GIT para Mac más reciente https://sourceforge.net/projects/GIT-osx-installer/files/GIT-2.23.0-intel-universal-mavericks.dmg/download?use_mirror=autoselect.
2. Se siguen las instrucciones indicadas por el instalador.

3. Cuando el instalador finaliza, se abre una terminal y se escribe el comando `GIT --version` para comprobar que la instalación se hizo correctamente. A partir de ahora, todos los comandos de GIT se utilizarán en una consola del sistema operativo.

Uso de GIT

Una vez se tiene GIT, se puede utilizar como se necesite. Sin embargo, es importante realizar una configuración inicial para indicarle a GIT el usuario que hace los commits en los repositorios. Para ello, se utilizan los comandos `GIT config --global user.email "email"`, y `GIT config --global user.name "username"`, cambiando las palabras `email` y `username` por la dirección del correo electrónico de la cuenta de Atlassian, y el nombre de usuario creado en la vinculación de la cuenta con Bitbucket, respectivamente.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (master)
$ git config --global user.name "ricava8916"
```

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (master)
$ git config --global user.email "ricava8916@gocasin.com"
```

A continuación, se explican los conceptos y comandos más importantes de GIT, y se dará un breve ejemplo de su uso.

Repositorio local:

Un repositorio es local cuando está almacenado en un computador personal, y puede ser modificado sin necesidad de tener conexión a internet. Hay 2 formas de tener un repositorio local:

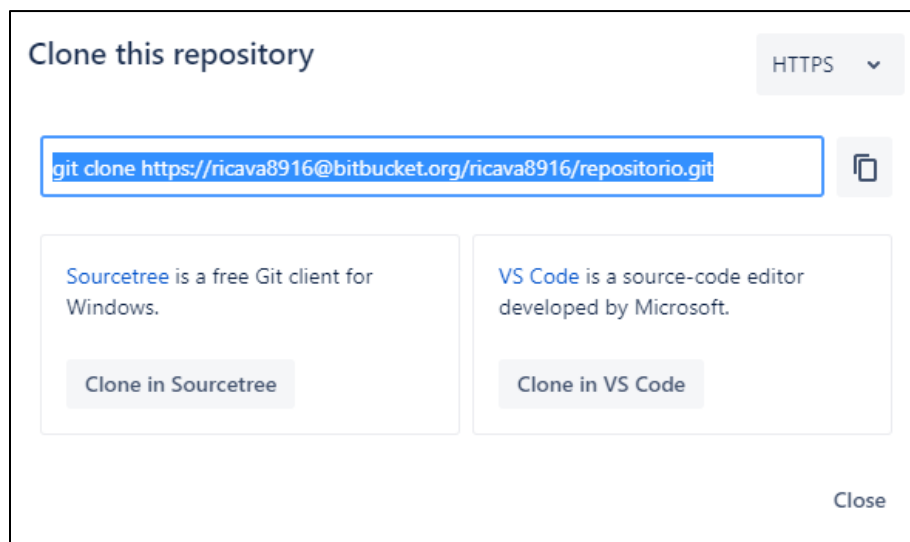
1. **Iniciar un repositorio local:** para ello se abre una consola en cualquier carpeta del computador donde se desee crear el repositorio y se utiliza el comando `GIT init`. Este se encarga de configurar el repositorio y crear los archivos necesarios para la creación del repositorio.

Este comando no se va a utilizar en esta práctica, sin embargo, es importante conocer su existencia pues es el comando más básico de GIT.

2. **Clonar un repositorio remoto:** los equipos de desarrollo de software normalmente utilizan un repositorio almacenado en alguna plataforma de hosting. Por lo tanto, para poder hacer cambios sobre el repositorio es necesario descargarlo o clonarlo.

La descarga se puede realizar desde la página oficial de la plataforma de hosting que utilice el equipo de trabajo, sin embargo, es más rápido clonar el repositorio. Para ello, se utiliza el comando `GIT clone url`, cambiando la palabra `url` por el enlace del repositorio que se va a descargar.

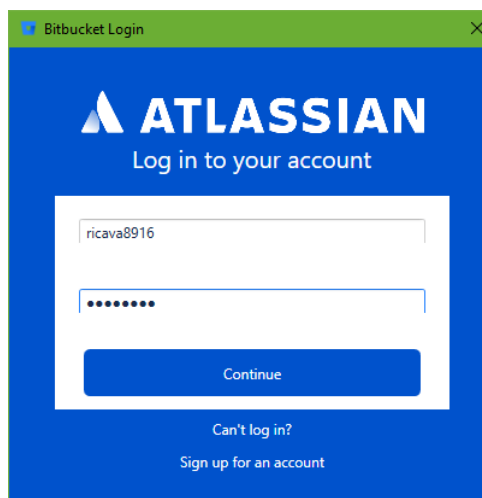
Ejercicio: ya que se va a editar localmente el repositorio creado anteriormente en Bitbucket, es necesario clonarlo y, por ende, tener el enlace del repositorio. Para conseguir este enlace, se dirige a la sección `Source` del repositorio en Bitbucket, y se selecciona `Clone` en la parte superior derecha de la página. Una vez allí, se copia el comando que se muestra.



Con el comando copiado se abre una consola en la carpeta en la que se almacenará el repositorio, y se pega.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto
$ git clone https://ricava8916@bitbucket.org/ricava8916/repositorio.git
Cloning into 'repositorio'...
```

Una vez se ingresa el comando, se solicita el ingreso del usuario y contraseña de la cuenta de Atlassian. Dependiendo de la consola en la que se utiliza GIT, esta solicitud se hace en la misma consola o en una ventana emergente.



Cuando se ingresan las credenciales se finaliza la descarga del repositorio y se pueden hacer los cambios que se indican más adelante en la guía.

Ramas del repositorio:

Entre los cambios más comunes que se hacen a un repositorio se encuentran la creación y destrucción de ramas, y el cambio entre estas.

1. **Crear ramas:** para crear ramas en el repositorio local se utiliza el comando `GIT branch name`, cambiando la palabra name por el nombre de la rama nueva. A la hora de elegir un nombre se debe tener en cuenta que este no puede tener espacios y que debe ser único, es decir, que en un repositorio no pueden existir dos ramas con el mismo nombre.

Ejercicio: debido a que más adelante se modificará el repositorio es necesario crear nuevas ramas, para ello se abre una ventana de comandos dentro de la carpeta que se descargó al clonar el repositorio, y se utilizan los comandos `GIT branch Avance_1` y `GIT branch Avance_2`. Estos nombres se eligieron teniendo en cuenta la nomenclatura de ramas definida anteriormente.

```
julia@DESKTOP-AK48UIJ MINGW64 /d/Proyecto/repositorio (master)
$ git branch Avance_1

julia@DESKTOP-AK48UIJ MINGW64 /d/Proyecto/repositorio (master)
$ git branch Avance_2
```

2. **Cambiar de rama:** para cambiar la rama en la que se ubica el repositorio local se utiliza el comando `GIT checkout name`, cambiando la palabra name por el nombre de la rama a la que se cambiará.

Ejercicio: una vez creadas las ramas en el repositorio, es necesario moverse a alguna de ellas para poder modificar sus archivos. Ya que el cambio anterior se hizo en la rama `Avance_0` tiene sentido que el próximo cambio se haga en la rama `Avance_1`. Por ello, se cambia de rama utilizando el comando `GIT checkout Avance_1`.

```
julia@DESKTOP-AK48UIJ MINGW64 /d/Proyecto/repositorio (master)
$ git checkout Avance_1
Switched to branch 'Avance_1'

julia@DESKTOP-AK48UIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$
```

Si el sistema operativo del computador es Windows y se utiliza GIT bash, se puede ver el nombre de la rama en la que GIT se encuentra en la línea de comandos. Por otro lado, si el computador tiene un sistema operativo diferente, o si se está utilizando una consola diferente a GIT bash en Windows, es posible utilizar el comando `GIT branch` para visualizar las ramas existentes y saber en cual se encuentra GIT.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git branch
* Avance_1
  Avance_2
  master
```

3. **Borrar ramas:** para borrar una rama del repositorio local se utiliza el comando `GIT branch -d name`, cambiando la palabra name por el nombre de la rama a borrar. Se debe tener cuidado con este comando pues al eliminar una rama se eliminan todos los cambios que se hayan realizado sobre esta. Normalmente se elimina una rama luego de hacer un merge con otra rama para mantener los cambios.

Ejercicio: ya que toda modificación se hará en la rama Avance_1, no es necesario tener la rama Avance_2. Por ello se borra esta rama con el comando `GIT branch -d Avance_2`. Si más adelante se necesita esta rama, se puede volver a crear.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git branch -d Avance_2
Deleted branch Avance_2 (was 711d48e).
```

Manejar cambios del repositorio remoto:

Para subir al repositorio remoto los cambios que han sido realizados en el repositorio local, es necesario utilizar los comandos de Staging, junto con los comandos commit, push y pull.

1. **Comandos de Staging:** Los comandos para manejar los cambios que se realizan en el repositorio local son:
 - **Status:** para conocer el estado de los cambios se utiliza el comando `GIT status`.
 - **Add:** para añadir los cambios realizados al próximo commit se utiliza el comando `GIT add "file"`, cambiando file por el nombre del archivo que se quiere añadir al commit. Si se quieren añadir todos los cambios realizados al commit, file se reemplaza por un punto (.).
 - **Remove:** si se desea descartar algún archivo de los añadidos al siguiente commit se utiliza el comando `GIT rm "file" --cached`, cambiando file por el nombre del archivo que se quiere remover.

Ejercicio: ahora que GIT está ubicado en la rama Avance_1 del repositorio local, se harán los cambios adecuados. En este caso, se crearán 2 archivos nuevos en la carpeta principal. En caso de ser necesario, es posible crear cualquier tipo de archivo, sin embargo, para simplificar el ejercicio se crean 2 notas de texto con los nombres Nota adicional y Nota por eliminar. Así, el contenido de la carpeta del repositorio es el siguiente:



This PC > Local Disk (D:) > Proyecto > repositorio >			
Name	Date modified	Type	Size
.git	6/23/2021 3:50 PM	File folder	
.gitignore	6/23/2021 2:45 PM	Text Document	1 KB
Nota adicional.txt	6/23/2021 4:33 PM	Text Document	0 KB
Nota por añadir.txt	6/23/2021 4:29 PM	Text Document	0 KB
Nota por eliminar.txt	6/23/2021 4:30 PM	Text Document	0 KB
README.md	6/23/2021 2:45 PM	Documento MD	3 KB

Para subir estos cambios al repositorio remoto es necesario hacer un commit, y para ello, se le debe añadir los archivos que han cambiado. De esta forma, se utiliza el comando `GIT add`, el cual se encarga de añadir todos los archivos cambiados al próximo commit.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git add .
```

Sin embargo, el archivo *Nota por eliminar* no se va a subir al repositorio remoto, pues contiene información que no tiene que ver con el proyecto. Por esta razón, se utiliza el comando `GIT rm 'Nota por eliminar.txt' --cached` para eliminar la nota de texto del próximo commit. Es importante recordar que el nombre del archivo siempre debe incluir la extensión del archivo, en este caso *.txt*.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git rm 'Nota por eliminar.txt' --cached
rm 'Nota por eliminar.txt'
```

Para conocer el estado del commit, y los cambios que han sido añadidos a este se utiliza el comando `GIT status`. En este se puede ver que se ha añadido el archivo *Nota adicional* mientras que el archivo *Nota por eliminar* no está en el commit.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git status
On branch Avance_1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Nota adicional.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Nota por eliminar.txt
```

2. **Commit:** para realizar el commit con los cambios añadidos con los comandos de Staging se utiliza el comando `GIT commit -m "msg"`, cambiando la palabra *msg* con el mensaje que se desea añadir al commit. Este mensaje es obligatorio para hacer el commit, y se suele utilizar para indicar el cambio que se ha realizado en este.

Ejercicio: una vez se han incluido al commit los archivos por subir al repositorio remoto, se registran los cambios creando un commit con el comando `GIT commit -m "Creación de notas de texto"`, describiendo el cambio en el mensaje.

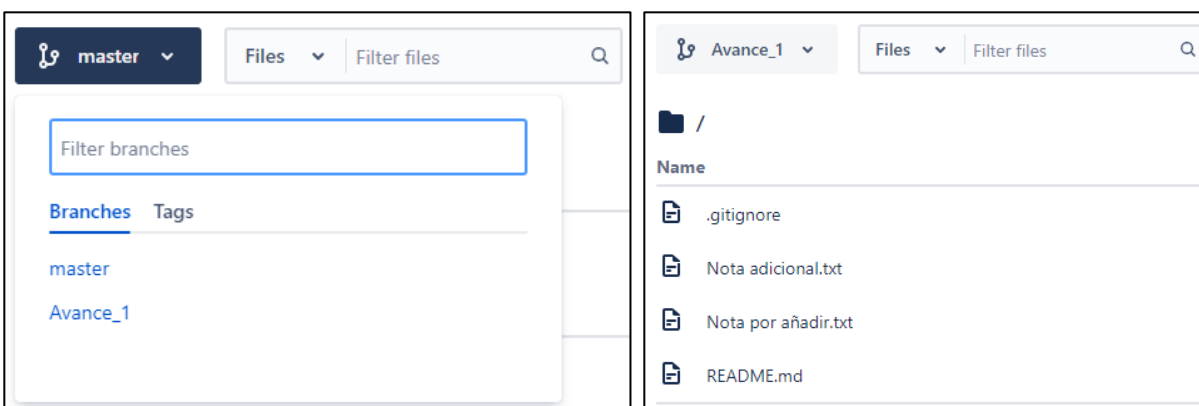
```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git commit -m "Creación de notas de texto"
[Avance_1 a4dc8bd] Creación de notas de texto
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Nota adicional.txt
```

3. **Push:** para subir los cambios realizados en el repositorio local al repositorio remoto se utiliza el comando `GIT push origin name`, cambiando name por el nombre de la rama donde se encuentran los cambios realizados en el repositorio local.

Ejercicio: ahora que se han hecho los cambios requeridos y que se ha realizado al menos un commit, se pueden subir los cambios al repositorio remoto. Para subir los cambios a la rama `Avance_1` se utiliza el comando `GIT push origin Avance_1`. De acuerdo con la configuración de GIT, es posible que la primera vez que se haga un pull se solicite el ingreso de la contraseña de la cuenta de Atlassian.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git push origin Avance_1
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create pull request for Avance_1:
remote:   https://bitbucket.org/ricava8916/repositorio/pull-requests/new?source=Avance_1&t=1
remote:
To https://bitbucket.org/ricava8916/repositorio.git
 * [new branch]      Avance_1 -> Avance_1
```

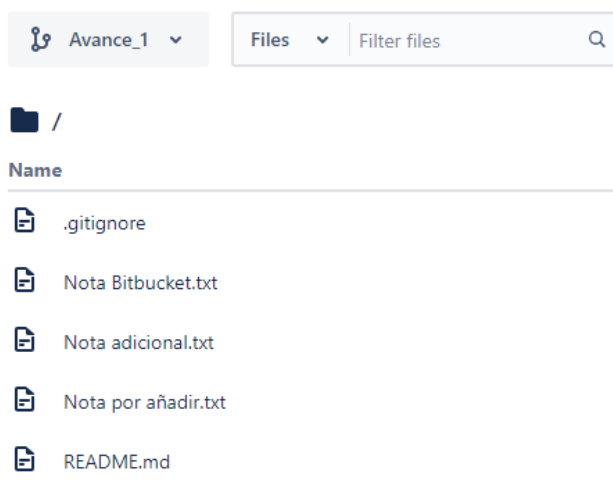
Una vez se ha hecho el push, en la sección [Source](#) del repositorio de Bitbucket se podrá ver la creación de la nueva rama, y las notas de texto dentro de la misma.



Es importante recordar que las notas se crearon en la rama [Avance_1](#), por lo tanto, si no se realiza el cambio de rama en Bitbucket no se podrán ver los cambios.

4. **Pull:** para descargar en el repositorio local los cambios nuevos que hayan sido subidos a una rama del repositorio remoto se utiliza el comando `GIT pull origin name`, cambiando [name](#) por el nombre de la rama cuyos cambios se quieren descargar.

Ejercicio: para hacer un pull es necesario hacer algún cambio en el repositorio remoto, por ello, se crea un nuevo archivo en la rama [Avance_1](#) utilizando la interfaz gráfica de Bitbucket, siguiendo las mismas instrucciones indicadas anteriormente. Este archivo se llama [Nota Bitbucket](#).



Una vez ha sido creado el archivo en el repositorio remoto, se utiliza en la consola del repositorio local el comando `GIT pull origin Avance_1` para descargar los cambios realizados en la rama [Avance_1](#).



```
julia@DESKTOP-AK48UIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git pull origin Avance_1
From https://bitbucket.org/ricava8916/repositorio
 * branch            Avance_1    -> FETCH_HEAD
Updating d730ce1..fb9a2c1
Fast-forward
 Nota Bitbucket.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Nota Bitbucket.txt
```

Cuando se descargan los cambios, al repositorio local se añade el archivo creado en el repositorio remoto.

This PC > Local Disk (D:) > Proyecto > repositorio			
Name	Date modified	Type	Size
.git	6/23/2021 7:06 PM	File folder	
.gitignore	6/23/2021 2:45 PM	Text Document	1 KB
Nota adicional.txt	6/23/2021 4:33 PM	Text Document	0 KB
Nota Bitbucket.txt	6/23/2021 7:06 PM	Text Document	1 KB
Nota por añadir.txt	6/23/2021 4:29 PM	Text Document	0 KB
Nota por eliminar.txt	6/23/2021 4:30 PM	Text Document	0 KB
README.md	6/23/2021 2:45 PM	Documento MD	3 KB

Moverse entre commits:

Una de las ventajas que proveen los repositorios es el poder cambiar entre commits sin problemas. Para hacer esto se utilizan los comandos log, head y reset.

1. **Log:** para consultar los commits existentes y conocer su valor SHA se utiliza el comando *GIT log*.

Ejercicio: para conocer el valor SHA de los commits que se han hecho, se utiliza el comando GIT log. En la información se puede ver la información básica de cada commit: autor, fecha, mensaje y valor SHA.



```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git log
commit e2234bd7e035406186143a6535f808f7b0f0837d (HEAD -> Avance_1, origin/Avance_1)
Author: Emiliano Martinez <ricava8916@gocasin.com>
Date: Thu Jun 24 02:09:13 2021 +0000

    Nota Bitbucket.txt created online with Bitbucket

commit a4dc8bd139f421da9e4817e53216fff8f426a221
Author: ricava8916 <ricava8916@gocasin.com>
Date: Wed Jun 23 21:04:05 2021 -0500

    Creación de notas de texto

commit f790a7810e6983a235a369dd2cabda8822a40ac4 (origin/master, origin/HEAD, master)
Merge: 711d48e 91ef37a
Author: Emiliano Martinez <ricava8916@gocasin.com>
Date: Thu Jun 24 01:33:37 2021 +0000

    Merged in Avance_0 (pull request #1)

    Nota creada online con Bitbucket

commit 91ef37aa7e33a979e52e657a906f1f9390e96dc4
Author: Emiliano Martinez <ricava8916@gocasin.com>
Date: Thu Jun 24 01:18:27 2021 +0000

    Nota creada online con Bitbucket

commit 711d48eb07f5ccfe1b09539d599af3f9931b26cf
Author: Emiliano Martinez <ricava8916@gocasin.com>
Date: Wed Jun 23 17:43:43 2021 +0000

    Initial commit
```

2. **Checkout:** para moverse entre commits se utiliza el comando `GIT checkout name`, cambiando la palabra name por el valor SHA del commit al cual se desea mover. Este comando sirve tanto para cambiar de ramas como para movernos entre commits, lo único que hay que tener presente es el valor por el que se cambia la palabra name.

Ejercicio: con la información adquirida con el log se puede mover el HEAD por los commits existentes. En este caso, se moverá al penúltimo commit de la rama *Avance_1*, cuyo mensaje es *Creación de notas de texto*. Para ello se utiliza el comando `GIT checkout a4dc8bd139f421da9e4817e53216fff8f426a221`, con el valor SHA de dicho commit.



```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git checkout a4dc8bd139f421da9e4817e53216fff8f426a221
Note: switching to 'a4dc8bd139f421da9e4817e53216fff8f426a221'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at a4dc8bd Creaci
```

Es posible comprobar que se ha movido al commit con el comando *GIT status* o comprobando los archivos del repositorio, ya que el archivo Nota Bitbucket no debería existir, pues este se añadió después de realizar el commit al que se cambió.

Name	Date modified	Type	Size
.git	6/24/2021 4:23 PM	File folder	
.gitignore	6/23/2021 8:58 PM	Text Document	1 KB
Nota adicional.txt	6/23/2021 7:45 PM	Text Document	0 KB
Nota por Añadir.txt	6/23/2021 8:58 PM	Text Document	1 KB
Nota por eliminar.txt	6/23/2021 7:47 PM	Text Document	0 KB
README.md	6/23/2021 8:58 PM	Documento MD	3 KB

Para mover el HEAD al main se utiliza el comando *GIT checkout Avance_1*, pues se vuelve a la rama donde se estaba anteriormente.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio ((a4dc8bd...))
$ git checkout Avance_1
Previous HEAD position was a4dc8bd Creaci
Switched to branch 'Avance_1'
```


3. **Reset:** para mover el main al commit anterior, se utiliza el comando `GIT reset --hard`.

Ejercicio: si en lugar de mover el HEAD al commit anterior se necesita mover tanto el HEAD como el main a dicho commit se utiliza el comando `reset`. En este caso, ya no se explorará el último commit como en el ejercicio anterior, sino que se moverá el HEAD y el main a dicho commit, para lo cual se utiliza el comando `GIT reset --hard`.

```
julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git reset --hard
HEAD is now at a4dc8bd Creaci|n de notas de texto

julia@DESKTOP-AK4BUIJ MINGW64 /d/Proyecto/repositorio (Avance_1)
$ git status
On branch Avance_1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Nota Bitbucket.txt
    Nota por eliminar.txt

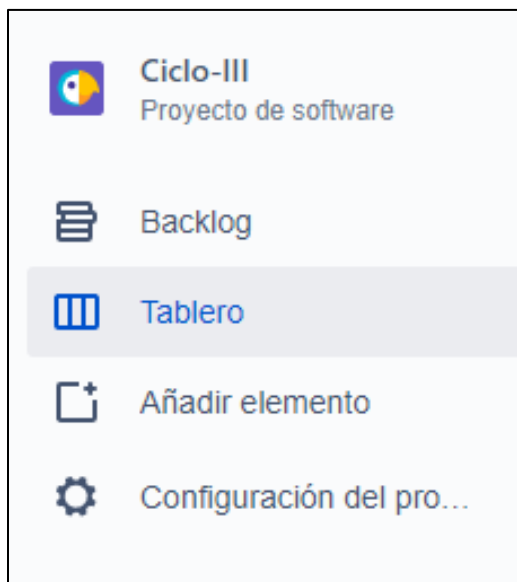
nothing added to commit but untracked files present (use "git add" to track)
```

Al hacerlo, si bien no se elimina la nota de texto *Nota Bitbucket*, si se considera como un cambio que no ha sido incluido al commit, pues cuando se hizo este commit dicho archivo no existía.

Integración de Bitbucket con Jira

Una gran ventaja que se obtiene al utilizar Jira es la integración que tiene con repositorios de Bitbucket y GitHub. La cual nos permite tener un acceso rápido al repositorio desde el proyecto de Jira asociado, y crear y asociar ramas a partir de los Issues de cada sprint.

Para realizar la integración, se ingresa al proyecto de Jira al cual se va a vincular el repositorio y se selecciona la opción [Añadir elemento](#) en la barra lateral izquierda



Luego, se selecciona el botón [Añadir](#) que se encuentra al lado de la sección [Repositorio](#).



Se ingresa el enlace del repositorio y el nombre del repositorio que se desee, y se presiona el botón *Establecer enlace*.

Conectar un repositorio

Vincular un repositorio mostrará información acerca de tus ramas, confirmaciones y solicitudes de extracción en las incidencias de Jira.

Enlace al repositorio

p. ej., <https://bitbucket.org/org/projectrepository>

Nombrar

p. ej., repositorio del proyecto

Establecer enlace Cancelar

Posteriormente se presiona el botón *Integrar con Bitbucket*.

Integrar JIRA con Bitbucket



Integrar con Bitbucket Editar repositorio

Finalmente, se otorga el acceso al workspace adecuado. A partir de ahora, cuando se ingrese al proyecto de Jira se podrá ver un nuevo elemento en la barra lateral izquierda con el nombre que se indicó anteriormente.

Utilizando las ventajas de integración entre Jira y Bitbucket

Las 3 ventajas que se obtienen al realizar la integración son las siguientes:

- 1) Acceso rápido al repositorio: una vez se ha integrado Bitbucket a Jira, en la barra lateral izquierda se encuentra un elemento que al dar clic sobre él redirige al repositorio asociado al proyecto.
- 2) Creación de ramas a partir de Issues: otra funcionalidad que proporciona la integración es la capacidad de creación rápida de ramas en el repositorio de acuerdo con los *Issues* del proyecto. Para ello, se va a la ventana de edición de un *Issue* y selecciona *crear rama* en el apartado rama.
- 3) Issues en Bitbucket: ahora, en la pestaña Jira Issues del repositorio de Bitbucket se pueden visualizar los *Issues* del proyecto de Jira y toda su información, y crear ramas a partir de ellos.