

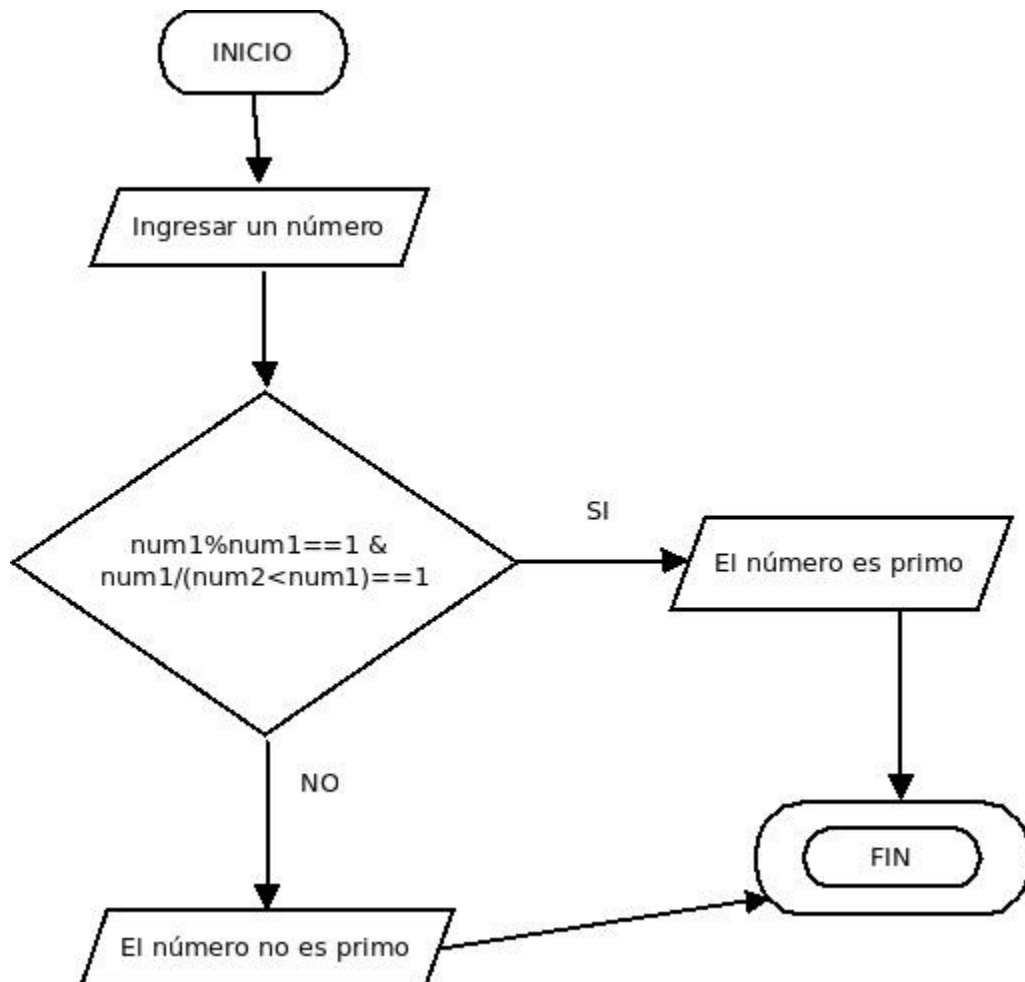
TAREA 3

Determinar si un numero es primo:

INICIO

```
define numero num1;  
define numero num2;  
    imprimir ("Ingrese un numero para su verificación:");  
    num1=leer();  
Hacer desde num2 hasta num1  
    si num1%num1==1 & num1/(num2<num1)==1  
        imprimir ("El número es primo");  
    de lo contrario  
        imprimir ("El número no es primo");
```

FIN



Convertir un valor dado en minutos a horas:

INICIO

```
define numero num;  
define numero res;  
  imprimir ("Ingrese la cantidad a convertir:")  
  num=leer();  
  imprimir ("El tiempo en horas es");  
  res=num/60  
  imprimir (res)
```

FIN



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/2

Google

1.2 · Checking the Status

Good job! As Git just told us, our "octobox" directory now has an empty repository in `/.git/`. The repository is a hidden directory where Git operates.

To save your progress as you go through this tutorial -- and earn a badge when you successfully complete it -- head over to [create a free Code School account](#). We'll wait for you here.

Next up, let's type the git status command to see what the current state of our project is:

`git status`

TryGit—1176x310

Press enter to submit commands

> git init

Initialized empty Git repository in `/.git/`


Success!

\$ |

My Octobox Repository

Advice

Tip:



Code School - Try Git - Mozilla Firefox

Clase #3.pdf Code School - Try Git


https://try.github.io/levels/1/challenges/4

1.4 · Adding Changes

Good, it looks like our Git repository is working properly. Notice how Git says octocat.txt is "untracked"? That means Git sees that octocat.txt is a new file.

To tell Git to start tracking changes made to octocat.txt, we first need to add it to the staging area by using git add.

[git add octocat.txt](#)



```
TryGit—1176x310
Success!
$ git status


# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       octocat.txt
nothing added to commit but untracked files present (use "git add" to track)
Success!
$ |
```

My Octobox Repository

.git

Advice

staged:



Code School - Try Git - Mozilla Firefox


Clase #3.pdf Code School - Try Git

https://try.github.io/levels/1/challenges/5

1.5 · Checking for Changes

Good job! Git is now tracking our octocat.txt file. Let's run git status again to see where we stand:

[git status](#)



```
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   octocat.txt
nothing added to commit but untracked files present (use "git add" to track)
Success!

$ git add octocat.txt

Nice job, you've added octocat.txt to the Staging Area

$ |
```


TryGit—1176x310

My Octobox Repository

.git

Advice

add all:



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/6

Google

11:43


1.6 - Committing

Notice how Git says changes to be committed? The files listed here are in the Staging Area, and they are not in our repository yet. We could add or remove files from the stage before we store them in the repository.

To store our staged changes we run the commit command with a message describing what we've changed. Let's do that now by typing:

```
git commit -m "Add cute octocat story"
```

tryGit



TryGit—1176x310

```
Nice job, you've added octocat.txt to the Staging Area
$ git status

# On branch master
# Initial commit
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   octocat.txt
#
Success!
$ |
```

My Octobox Repository

Advice

Code School - Try Git - Mozilla Firefox

Clase #3.pdf Code School - Try Git


https://try.github.io/levels/1/challenges/6

1.6 · Committing

Notice how Git says changes to be committed? The files listed here are in the Staging Area, and they are not in our repository yet. We could add or remove files from the stage before we store them in the repository.

To store our staged changes we run the commit command with a message describing what we've changed. Let's do that now by typing:

```
git commit -m "Add cute octocat story"
```



```
Nice job, you've added octocat.txt to the Staging Area
$ git status

# On branch master
# Initial commit
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   octocat.txt
#
Success!
$ |
```

TryGit—1176x310

My Octobox Repository

Advice

Code School - Try Git - Mozilla Firefox

Clase #3.pdf Code School - Try Git


https://try.github.io/levels/1/challenges/7

1.7 · Adding All Changes

Great! You also can use wildcards if you want to add many files of the same type. Notice that I've added a bunch of .txt files into your directory below.

I put some in a directory named "octofamily" and some others ended up in the root of our "octobox" directory. Luckily, we can add all the new files using a wildcard with git add. Don't forget the quotes!

```
git add '*.txt'
```



```
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   octocat.txt
#
Success!

$ git commit -m "Add cute octocat story"

[master (root-commit) 20b5ccd] Add cute octocat story
1 file changed, 1 insertion(+)
create mode 100644 octocat.txt

Success!

$ |
```

Try Git—1176x310

My Octobox Repository

Advice

1.8 - Committing All Changes

Okay, you've added all the text files to the staging area. Feel free to run `git status` to see what you're about to commit.

If it looks good, go ahead and run:

```
git commit -m 'Add all the octocat txt files'
```



```
$ git commit -m "Add cute octocat story"

[master (root-commit) 20b5ccd] Add cute octocat story
1 file changed, 1 insertion(+)
create mode 100644 octocat.txt

Success!

$ git add '*.txt'

Success!

$ |
```

My Octobox Repository

.git

Advice

Check all the things!



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/9

Google


1.9 • History

So we've made a few commits. Now let's browse them to see what we changed.

Fortunately for us, there's git log. Think of Git's log as a journal that remembers all the changes we've committed so far, in the order we committed them. Try running it now:

[git log](#)

TryGit



TryGit—1176x310


```
Success!  
$ git commit -m 'Add all the octocat txt files'  
  
[master 3852b4d] Add all the octocat txt files  
4 files changed, 4 insertions(+)  
create mode 100644 blue_octocat.txt  
create mode 100644 octofamily/baby_octocat.txt  
create mode 100644 octofamily/momma_octocat.txt  
create mode 100644 red_octocat.txt  
  
Success!  
$ |
```

My Octobox Repository

.git

Advice

More useful logs:



Code School - Try Git - Mozilla Firefox

Clase #3.pdf Code School - Try Git

https://try.github.io/levels/1/challenges/10


1.10 · Remote Repositories

Great job! We've gone ahead and created a new empty GitHub repository for you to use with Try Git at https://github.com/try-git/try_git.git. To push our local *repo* to the GitHub server we'll need to add a remote repository.

This command takes a *remote name* and a *repository URL*, which in your case is https://github.com/try-git/try_git.git.

Go ahead and run `git remote add` with the options below:

```
git remote add origin https://github.com/try-git/try_git.git
```



```
$ git log

commit 3852b4db1634463d0bb4d267edb7b3f9cd02ace1
Author: Try Git <try_git@github.com>
Date: Sat Oct 10 08:30:00 2020 -0500

    Add all the octocat txt files

commit b652edfd888cd3d5e7fcb857d0dabc5a0fcb5e28
Author: Try Git <try_git@github.com>
Date: Sat Oct 10 08:30:00 2020 -0500

    Added cute octocat story

Success!

$ |
```

TryGit—1176x310

My Octobox Repository Advice

Code School - Try Git - Mozilla Firefox

Clase #3.pdf Code School - Try Git


https://try.github.io/levels/1/challenges/11

1.11 · Pushing Remotely

The push command tells Git where to put our commits when we're ready, and boy we're ready. So let's push our local changes to our **origin** repo (on GitHub).

The name of our remote is origin and the default local branch name is master. The -u tells Git to remember the parameters, so that next time we can simply run git push and Git will know what to do. Go ahead and push it!

```
git push -u origin master
```



```
TryGit—1176x310
Added cute octocat story
Success!
$ git remote add origin https://github.com/try_git.git
Did not use the correct URL
$ git remote add origin https://github.com/try-git/try_git.git
Success!
$ |
```

My Octobox Repository

Advice

1.12 - Pulling Remotely

Let's pretend some time has passed. We've invited other people to our github project who have pulled your changes, made their own commits, and pushed them.

We can check for changes on our GitHub repository and pull down any new changes by running:

```
git pull origin master
```



```
TryGit—1176x310

Did not use the correct URL
$ git remote add origin https://github.com/try-git/try_git.git

Success!
$ git push -u origin master

Branch master set up to track remote branch master from origin.
Success!
$ |
```

My Octobox Repository

[.git](#)

Advice

git stash:



1.13 · Differences

Uh oh, looks like there have been some additions and changes to the octocat family. Let's take a look at what is different from our last commit by using the git diff command.

In this case we want the diff of our most recent commit, which we can refer to using the HEAD pointer.

→ `git diff HEAD`



```
Branch master set up to track remote branch master from origin.
```

```
Success!
```

```
$ git pull origin master
```

```
Updating 3852b4d..3e70b0f
Fast-forward
 yellow_octocat.txt | 1
 1 file changed, 1 insertion(+)
 create mode 100644 yellow_octocat.txt
```

```
Success!
```

```
$ |
```

My Octobox Repository

Advice

HEAD



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/14

Google

11:48


1.14 · Staged Differences

Another great use for diff is looking at changes within files that have already been staged. Remember, staged files are files we have told git that are ready to be committed.

Let's use git add to stage octofamily/octodog.txt, which I just added to the family for you.

```
git add octofamily/octodog.txt
```

TryGit



TryGit—1176x310

```
create mode 100644 yellow_octocat.txt
Success!

$ git diff HEAD

diff --git a/octocat.txt b/octocat.txt
index 7d8d908..e725ef6 100644
--- a/octocat.txt
+++ b/octocat.txt
@@ -1,1 @@
-A Tale of Two Octocats
+The Tale of Two Octocats and an Octodog
Success!


$ |
```

My Octobox Repository

.git

Advice

Commit Etiquette:



Code School - Try Git - Mozilla Firefox

Clase #3.pdf

Code School - Try Git

https://try.github.io/levels/1/challenges/15

Google

11:49

1.15 · Staged Differences (cont'd)

Good, now go ahead and run git diff with the --staged option to see the changes you just staged. You should see that octodog.txt was created.

[git diff --staged](#)

TryGit—1176x310

```
Success!
$ git add octofamily/octodog.txt


fatal: pathspec 'octofamily/octodog.txt' did not match any files
Did not add octofamily/octodog.txt
$ git add octofamily/octodog.txt

Success!
$ |
```

My Octobox Repository

Advice

Commit Etiquette:



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/16

Google


1.16 · Resetting the Stage

So now that octodog is part of the family, octocat is all depressed. Since we love octocat more than octodog, we'll turn his frown around by removing octodog.txt.

You can unstage files by using the git reset command. Go ahead and remove octofamily/octodog.txt.

[git reset octofamily/octodog.txt](#)

TryGit



TryGit—1176x310


```
Success!  
$ git diff --staged  
  
diff --git a/octofamily/octodog.txt b/octofamily/octodog.txt  
new file mode 100644  
index 0000000..cfbc74a  
--- /dev/null  
+++ b/octofamily/octodog.txt  
@@ -0,0 +1 @@  
[removed]  
Success!  
$ |
```

My Octobox Repository

.git

Advice

Commit Etiquette:



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/17

Google


1.17 · Undo

git reset did a great job of unstaging octodog.txt, but you'll notice that he's still there. He's just not staged anymore. It would be great if we could go back to how things were before octodog came around and ruined the party.

Files can be changed back to how they were at the last commit by using the command: `git checkout -- <target>`. Go ahead and get rid of all the changes since the last commit for octocat.txt

```
git checkout -- octocat.txt
```

TryGit—1176x310

```
diff --git a/octofamily/octodog.txt b/octofamily/octodog.txt
new file mode 100644
index 0000000..cfbc74a
--- /dev/null
+++ b/octofamily/octodog.txt
@@ -0,0 +1 @@
+
Success!
$ git reset octofamily/octodog.txt
Success!
$
```

My Octobox Repository

Advice

Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/18

Google

1.18 · Branching Out

When developers are working on a feature or bug they'll often create a copy (aka. branch) of their code they can make separate commits to. Then when they're done they can merge this branch back into their main master branch.

We want to remove all these pesky octocats, so let's create a branch called `clean_up`, where we'll do all the work:

```
git branch clean_up
```

TryGit—1176x310

```
Success!
$ git reset octofamily/octodog.txt

Success!
$ git checkout -- octocat.txt

Success!
$ |
```

My Octobox Repository

Advice

Branching



Code School - Try Git - Mozilla Firefox

Clase #3.pdf Code School - Try Git


https://try.github.io/levels/1/challenges/19

1.19 · Switching Branches

Great! Now if you type `git branch` you'll see two local branches: a main branch named `master` and your new branch named `clean_up`.

You can switch branches using the `git checkout <branch>` command. Try it now to switch to the `clean_up` branch:

```
git checkout clean_up
```



```
TryGit-1176x310
Success!
$ git checkout -- octocat.txt


Success!
$ git branch clean_up

Success!
$ |
```

My Octobox Repository

.git

Advice
All at Once



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/20

Google


1.20 · Removing All The Things

Ok, so you're in the clean_up branch. You can finally remove all those pesky octocats by using the git rm command which will not only remove the actual files from disk, but will also stage the removal of the files for us.

You're going to want to use a wildcard again to get all the octocats in one sweep, go ahead and run:

```
git rm '*.txt'
```

TryGit



TryGit—1176x310


```
Success!  
$ git branch clean_up  
  
Success!  
$ git checkout clean_up  
  
Switched to branch 'clean_up'  
Success!  
$ |
```

My Octobox Repository

.git

Advice

Remove all the things!



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/21

Google


1.21 - Committing Branch Changes

Now that you've removed all the cats you'll need to commit your changes.

Feel free to run git status to check the changes you're about to commit.

```
git commit -m "Remove all the cats"
```

tryGit



TryGit—1176x310

```
Switched to branch 'clean_up'
Success!
$ git rm '*.txt'


rm 'blue octocat.txt'
rm 'octocat.txt'
rm 'octofamily/baby octocat.txt'
rm 'octofamily/momma octocat.txt'
rm 'red octocat.txt'
Success!
$ |
```

My Octobox Repository

.git

Advice

The '-a' option



1.22 · Switching Back to master

Great, you're almost finished with the cat... er the bug fix, you just need to switch back to the master branch so you can copy (or merge) your changes from the clean_up branch back into the master branch.

Go ahead and checkout the master branch:

[git checkout master](#)



```
rm 'red_octocat.txt'
Success!
$ git commit -m "Remove all the cats"

[clean up 63540fe] Remove all the cats
5 files changed, 5 deletions(-)
delete mode 100644 blue_octocat.txt
delete mode 100644 octocat.txt
delete mode 100644 octofamily/baby_octocat.txt
delete mode 100644 octofamily/momma_octocat.txt
delete mode 100644 red_octocat.txt
Success!
$
```

My Octobox Repository

.git

Advice

Pull Requests



Code School - Try Git - Mozilla Firefox

Clase #3.pdf


Code School - Try Git

https://try.github.io/levels/1/challenges/23

Google

1.23 · Preparing to Merge

tryGit



Alrighty, the moment has come when you have to merge your changes from the `clean_up` branch into the master branch. Take a deep breath, it's not that scary.

We're already on the master branch, so we just need to tell Git to merge the `clean_up` branch into it:

[git merge clean_up](#)

TryGit—1176x310

```
[clean up 63540fe] Remove all the cats
5 files changed, 5 deletions(-)
delete mode 100644 blue_octocat.txt
delete mode 100644 octocat.txt
delete mode 100644 octofamily/baby_octocat.txt
delete mode 100644 octofamily/momma_octocat.txt
delete mode 100644 red_octocat.txt

Success!

$ git checkout master

Switched to branch 'master'

Success!


$ |
```

My Octobox Repository

.git

Advice

Merge Conflicts



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/24

Google


1.24 · Keeping Things Clean

Congratulations! You just accomplished your first successful bugfix and merge. All that's left to do is clean up after yourself. Since you're done with the `clean_up` branch you don't need it anymore.

You can use `git branch -d <branch name>` to delete a branch. Go ahead and delete the `clean_up` branch now:

```
git branch -d clean_up
```

TryGit



TryGit—1176x310

```
Updating 3852b4d..ec6888b
Fast-forward
 blue.octocat.txt      | 1
 octocat.txt           | 1
 octofamily/baby.octocat.txt | 1
 octofamily/momma.octocat.txt | 1
 red.octocat.txt       | 1
 5 files changed, 5 deletions(-)
 delete mode 100644 blue.octocat.txt
 delete mode 100644 octocat.txt
 delete mode 100644 octofamily/baby.octocat.txt
 delete mode 100644 octofamily/momma.octocat.txt
 delete mode 100644 red.octocat.txt

Success!


$ |
```

My Octobox Repository

.git

Advice

Force delete



Code School - Try Git - Mozilla Firefox

Clase #3.pdfCode School - Try Git

https://try.github.io/levels/1/challenges/25

Google

1.25 · The Final Push

Great! You now have a little taste of the greatness of Git. You can take a look at the wrap up page for a little more information on Git and GitHub, oh, and of course your badge!

Wrap it all Up

TryGit—1176x310

```
$ git branch -d clean_up  
  
Deleted branch clean_up (was ec6888b).  
Success!  
$ git push  
  
To https://github.com/try-git/try-git.git  
3e70b0f..7dab2c5  master -> master  
Success!  
> |
```

My Octobox Repository

Advice

Learning more about Git

