

Digital  
House >  
Coding School

...

**GIT**



git

## Places

- Recent
- Home
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos
- Trash

## Devices

- T1106333W0B
- Iphone Daro
- Computer

## Network

- Browse Network
- Connect to Server



Curriculum.docx



Curriculum2.docx



Curriculum2 Inglés.  
docx



Curriculum2-  
Octubre.docx



Curriculum3.docx



foto.jpg



foto\_recorte.png



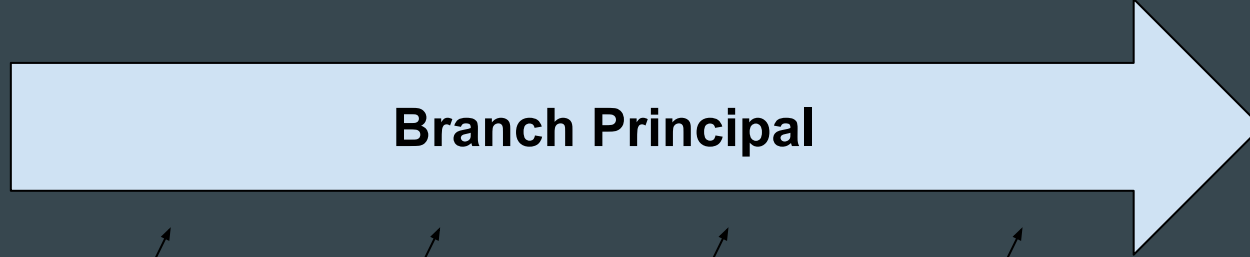
box



git

# Git - Características

- Backups y recuperación de archivos
  - Undo a corto plazo
  - Undo a largo plazo
- Sincronización
  - Seguimiento de cambios
  - Seguimiento de autoría
  - Resolución de conflictos
- Branching and merging
- Forking y Pull Requests
- Espacio de pruebas



**Harina**

**Revisión 1**

**Harina  
Huevos**

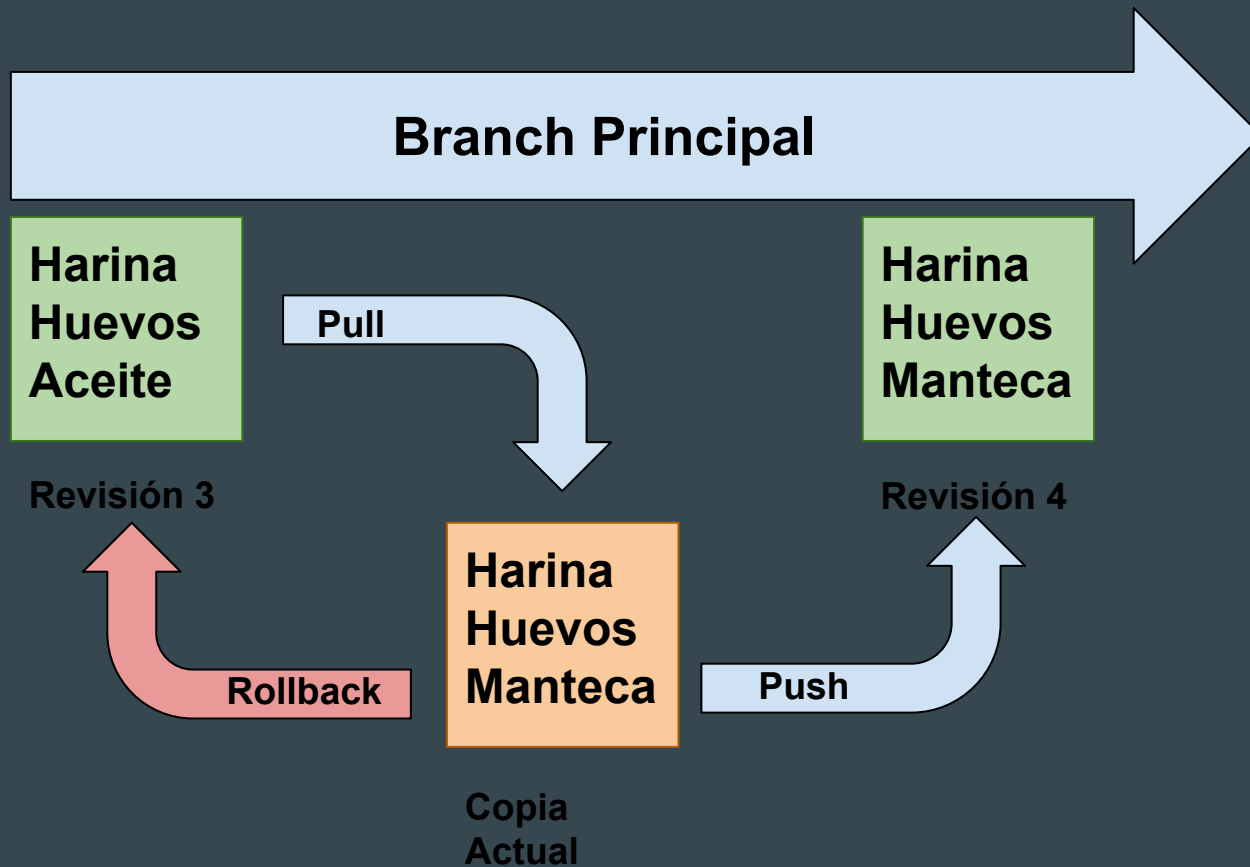
**Revisión 2**

**Harina  
Huevos  
Aceite**

**Revisión 3**

**Harina  
Huevos  
Manteca**

**Revisión 4**





## Branch Principal

**Harina**

+ Huevos

**Harina  
Huevos**

+ Aceite

**Harina  
Huevos  
Aceite**

+ Manteca  
- Aceite

**Harina  
Huevos  
Manteca**

Revisión 1

Revisión 2

Revisión 3

Revisión 4

Revisión 4

Harina  
Huevos  
Aceite

Revisión 6

Harina  
Huevos

Nuevas Funcionalidades

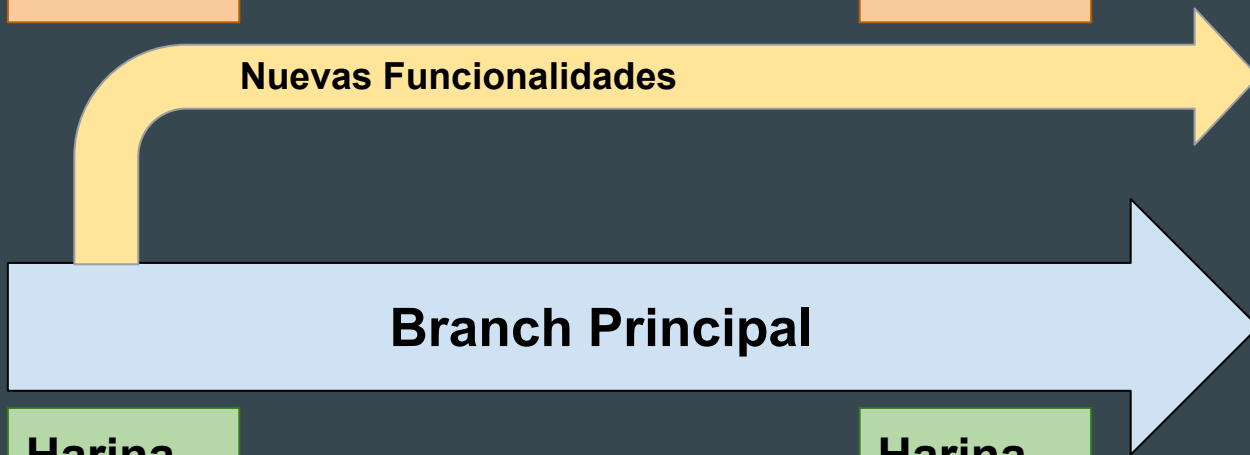
Branch Principal

Harina  
Huevos  
Aceite

Revisión 3

Harina  
Huevos  
Manteca

Revisión 5



Revisión 4

Harina  
Huevos  
Aceite

Revisión 6

Harina  
Huevos  
Café

Nuevas Funcionalidades

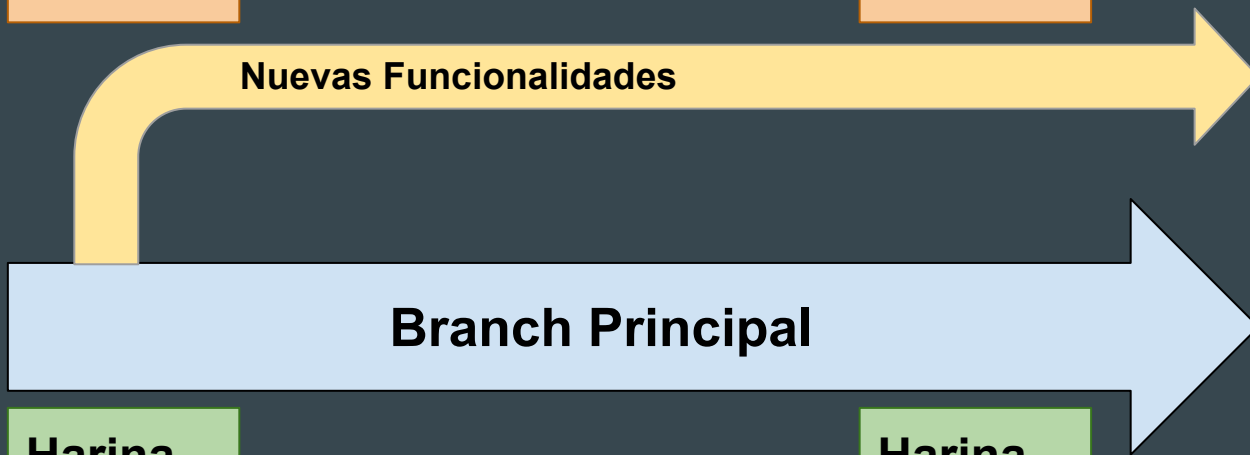
Branch Principal

Harina  
Huevos  
Aceite

Revisión 3

Harina  
Huevos  
Manteca

Revisión 5



Práctica  
(<https://try.github.io/>)  
...

# Vocabulario

...

# Repositorio

*Un repositorio un espacio en donde se guardan archivos para luego ser consultados u obtenidos*

# Servidor y cliente GIT

*Un servidor GIT es un repositorio que permite acceder mediante GIT.*

*Un cliente GIT permite consultar el contenido de un servidor, muchas veces creando un repositorio local.*

# Copia de trabajo actual

*La copia de trabajo actual indica únicamente el contenido visualizado por nuestro cliente (pudiendo estar modificado)*



# Revisión

*Una revisión indica una versión puntual de los archivos de trabajo.*

# Rollback

*Una terminos generales, Rollback significa volver el estado de nuestro repositorio (o un archivo) a una versión anterior existente en el servidor.*

# Forks y Pull Requests

*Un Fork permite clonar un repositorio ajeno y realizar cambios sobre el mismo en un servidor GIT propio.*

*Un Pull Request nos permite pedirle al autor original que integre nuestros cambios en el repositorio original.*

# Branch y Merge

*Una Branch es similar a un Fork pero existente únicamente dentro de nuestro mismo servidor.*

*Al juntar ramas se lo llama Merge*

# Tag

*Un Tag marca un momento historico importante dentro de  
nuestro repositorio.*

# HEAD

*HEAD es una palabra clave dentro de GIT que marca el estado actual del proyecto dentro de nuestra rama.*

# Master, Upstream y Origin

*Origin marca el Fork propio del proyecto.*

*Upstream marca el Fork original del proyecto.*

*Master marca la Branch principal del proyecto*

# Comandos

...



# Git init

Crear un repositorio GIT vacío.

```
dario@lucina:~$ cd TESTDH/test
dario@lucina:~/TESTDH/test$ git init
Initialized empty Git repository in /home/dario/TESTDH/test/.git/
dario@lucina:~/TESTDH/test$
```

# Git clone

Clonar un repositorio en un directorio nuevo

```
dario@lucina:~/TESTDH/test/minoru$ git clone https://github.com/esneider/minoru.git
Cloning into 'minoru'...
remote: Counting objects: 950, done.
remote: Total 950 (delta 0), reused 0 (delta 0), pack-reused 950
Receiving objects: 100% (950/950), 117.83 MiB | 2.92 MiB/s, done.
Resolving deltas: 100% (296/296), done.
Checking connectivity... done.
dario@lucina:~/TESTDH/test/minoru$
```

# Git status

Muestra el estado de nuestra copia local

```
dario@lucina:~/TESTDH/test-git$ git status
On branch master

Initial commit

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

        README.md

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

# Git add / Git rm

Agregar o quitar archivos de nuestro próximo commit

```
dario@lucina:~/TESTDH/test-git$ git status
On branch master

Initial commit

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

    README.md

nothing added to commit but untracked files present (use "git add" to track)
dario@lucina:~/TESTDH/test-git$ git add README.md
dario@lucina:~/TESTDH/test-git$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   README.md

dario@lucina:~/TESTDH/test-git$
```

# Git commit

Guardar cambios en el repositorio local

```
dario@lucina:~/TESTDH/test-git$ git commit -m "Se agrega el README"
[master (root-commit) d37be51] Se agrega el README
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
dario@lucina:~/TESTDH/test-git$
```

- -a agrega todos los archivos modificados al commit.
- -m permite poner un mensaje desde la línea de comandos.

# Git pull / Git push

Actualizar referencias en el repositorio remoto

```
dario@lucina:~/TESTDH/test-git$ git push origin master
Username for 'https://github.com': dariosus
Password for 'https://dariosus@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 219 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/dariosus/test-git.git
 * [new branch]      master -> master
dario@lucina:~/TESTDH/test-git$ git pull
Already up-to-date.
dario@lucina:~/TESTDH/test-git$ █
```

- Se utiliza “git [push|pull] FORK BRANCH
- “git push -u FORK BRANCH” nos permite actualizar ramas nuevas.

# Git diff

Permite ver las modificaciones en un archivo

```
dario@lucina:~/TESTDH/test-git$ git diff README.md
diff --git a/README.md b/README.md
index e69de29..09bd404 100644
--- a/README.md
+++ b/README.md
@@ -0,0 +1,4 @@
+Línea de prueba
+Línea número 2
+
+dario@lucina:~/TESTDH/test-git$
```

# Git checkout [1]

Permite crear y moverse entre distintas ramas

```
dario@lucina:~/TESTDH/test-git$ git checkout -b prueba
M      README.md
Switched to a new branch 'prueba'
dario@lucina:~/TESTDH/test-git$
```

- La opción -b permite crear una rama nueva



# Git checkout [2]

Permite hacer un rollback sobre un archivo

```
dario@lucina:~/TESTDH/test-git$ git checkout README.md
dario@lucina:~/TESTDH/test-git$ git status
On branch prueba
nothing to commit, working directory clean
dario@lucina:~/TESTDH/test-git$
```

# Git checkout [3]

Permite hacer un rollback en el repositorio

```
dario@lucina:~/TESTDH/test-git$ git checkout d37be51a437bc8d775909f36bbd418bfc4fd2b7a
Note: checking out 'd37be51a437bc8d775909f36bbd418bfc4fd2b7a'.
```

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 performing another checkout.

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

```
git checkout -b new_branch_name
```

```
HEAD is now at d37be51... Se agrega el README
```

# Git branch

Permite ver las ramas existentes

```
dario@lucina:~/TESTDH/test-git$ git branch
  master
* prueba
dario@lucina:~/TESTDH/test-git$
```

# Git merge

Permite unir ramas

```
dario@lucina:~/TESTDH/test-git$ git merge prueba
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
dario@lucina:~/TESTDH/test-git$
```