

Git para la humanidad

Alexander Martinez Mendez

Grupo Halley

@mxrtinez

Presentación adaptada de:


Alice Bartlett. Senior Developer, Financial Times @alicebartlett

Presentación: <https://speakerdeck.com/alicebartlett/git-for-humans?slide=95>

Video: <https://www.youtube.com/watch?v=eWxxfttcMts>

Git

¿Qué es Git?

A man with short hair, wearing a dark t-shirt, is speaking at a podium. He is gesturing with his hands while holding a small object in his right hand. The background is dark and out of focus.

“Git es una aplicación que se ejecuta en el ordenador, como un navegador web o un procesador de textos.”

Tom Stuart

<http://codon.com/>

¿Qué hace Git?

Git te ayuda a gestionar el trabajo realizado en tus proyectos.

**GIT ES POCO
AMIGABLE**


```
o-techdocs — bash — 172x45
fsevent_watch ... bash ... bash bash bash node bash +
ft-origami o-footer o-techdocs strathausen-dracula-a6a5fa7
fticons o-forms o-typography test
google-amp o-ft-icons-blog-post o-video top_u_r_l_hits_20160205_150147.csv
headshot-images o-grid origami-build-service
logo-images o-header origami-build-tools
n-light-signup o-header-readme-draft origami-image-service
20:29:08-alice.bartlett~/Code$ git checkout o-techdocs/
fatal: Not a git repository (or any of the parent directories): .git
20:29:14-alice.bartlett~/Code$ cd o-techdocs/
20:29:18-alice.bartlett~/Code/o-techdocs (fix-code-color-contrast)$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
20:29:29-alice.bartlett~/Code/o-techdocs (master)$ git pull origin master
remote: Counting objects: 8, done.
remote: Total 8 (delta 4), reused 4 (delta 4), pack-reused 4
Unpacking objects: 100% (8/8), done.
From github.com:Financial-Times/o-techdocs
 * branch      master      -> FETCH_HEAD
   8e805e9..55e0b1e master    -> origin/master
Updating 8e805e9..55e0b1e
Fast-forward
 circle.yml | 4 ++--
 origami.json | 1 +
 2 files changed, 3 insertions(+), 2 deletions(-)
20:29:39-alice.bartlett~/Code/o-techdocs (master)$ git branch
  add-pally
  fix-code-color-contrast
 * master
  remove-benton
  removeBentonSans
20:29:54-alice.bartlett~/Code/o-techdocs (master)$ git branch -d add-pally
Deleted branch add-pally (was 6a139f6).
20:30:04-alice.bartlett~/Code/o-techdocs (master)$ git branch -d fix-code-color-contrast
Deleted branch fix-code-color-contrast (was 87fe768).
20:30:19-alice.bartlett~/Code/o-techdocs (master)$ git branch -d remove-benton
Deleted branch remove-benton (was 2e3cd0a).
20:30:29-alice.bartlett~/Code/o-techdocs (master)$ git branch -d removeBentonSans
Deleted branch removeBentonSans (was 8cfaa98).
20:30:39-alice.bartlett~/Code/o-techdocs (master)$ git branch
 * master
20:30:42-alice.bartlett~/Code/o-techdocs (master)$ git checkout -b add-services-header
Switched to a new branch 'add-services-header'
20:30:52-alice.bartlett~/Code/o-techdocs (add-services-header)$ atom .
20:30:58-alice.bartlett~/Code/o-techdocs (add-services-header)$ atom .
20:33:00-alice.bartlett~/Code/o-techdocs (add-services-header)$
```


GIT



**SIN EMBARGO, GIT ES
BASTANTE SIMPLE**

**Hay otras aplicaciones que
puedes utilizar para usar
Git.**

**¿POR QUÉ ESTÁS
AQUÍ?**

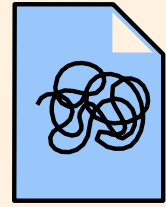
1. Asunto 1
2. Asunto 2
3. Asunto 3
4. Asunto 4
5. Asunto 5

Asunto 1:

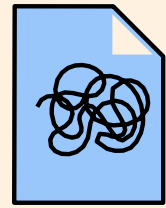
**GIT TE PERMITE CONTAR
LA HISTORIA DE TU
PROYECTO**

Git se utiliza para tomar **imágenes** de todos los archivos de una carpeta. Esta carpeta se llama **repositorio** o **repo**.

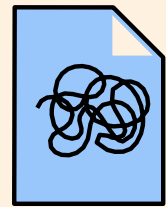
Cuando se desea tomar una imagen de un archivo o archivos, se crea un **commit**



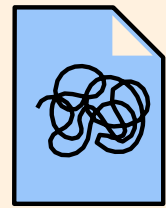
logo.svg



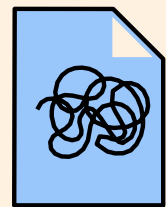
logo-2.svg



logo-3-monica-feedback.svg



logo-3-FINAL.svg

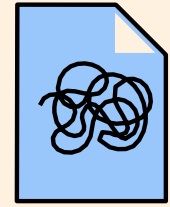


logo-3-FINAL-1.svg

Guardando copias

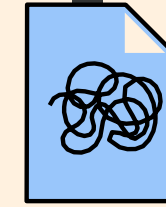
Haciendo commits

Guardando copias



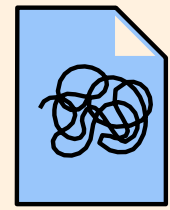
logo.svg

Haciendo commits

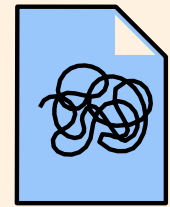


logo.svg

Guardando copias



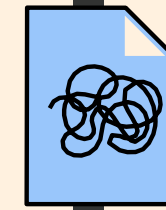
logo.svg



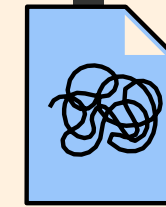
logo-2.svg



Haciendo commits

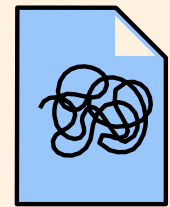


logo.svg

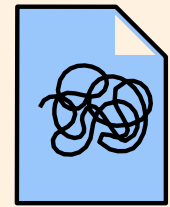


logo.svg

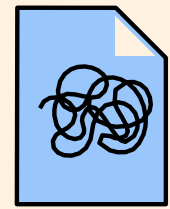
Guardando copias



logo.svg



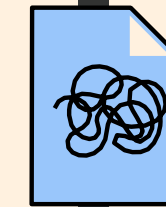
logo-2.svg



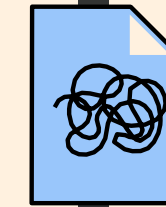
logo-3-monica-feedback.svg



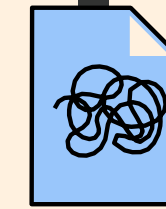
Haciendo commits



logo.svg

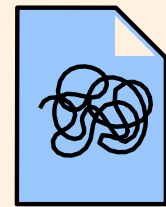


logo.svg

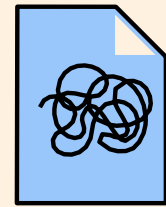


logo.svg

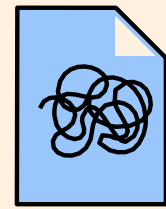
Guardando copias



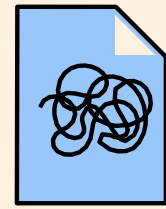
logo.svg



logo-2.svg

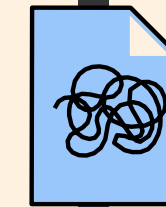


logo-3-monica-feedback.svg

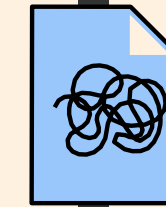


logo-3-FINAL.svg

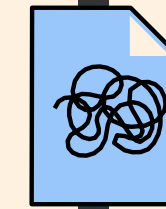
Haciendo commits



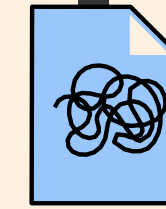
logo.svg



logo.svg



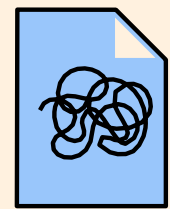
logo.svg



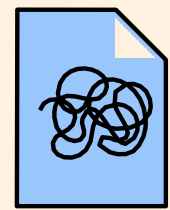
logo.svg



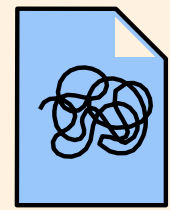
Guardando copias



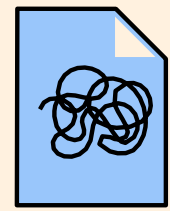
logo.svg



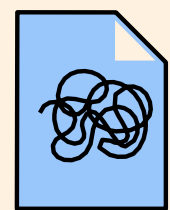
logo-2.svg



logo-3-monica-feedback.svg

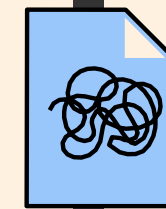


logo-3-FINAL.svg

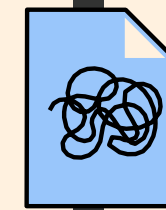


logo-3-FINAL-1.svg

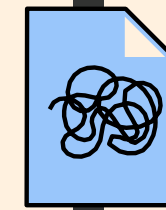
Haciendo commits



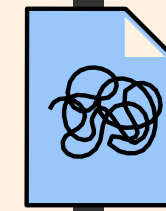
logo.svg



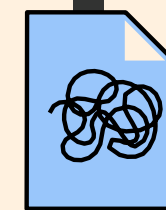
logo.svg



logo.svg



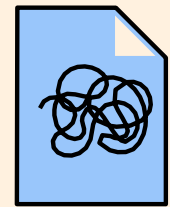
logo.svg



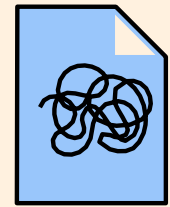
logo.svg



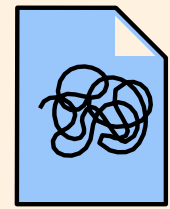
Guardando copias



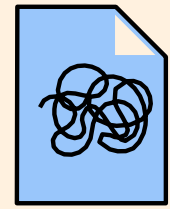
logo.svg



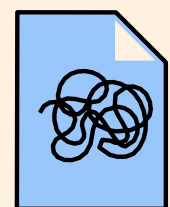
logo-2.svg



logo-3-monica-feedback.svg

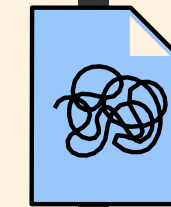


logo-3-FINAL.svg

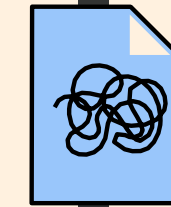


logo-3-FINAL-1.svg

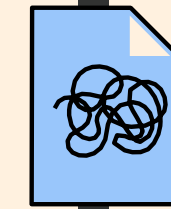
Haciendo commits



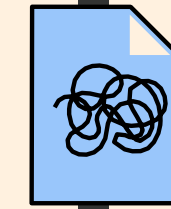
logo.svg



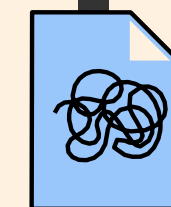
logo.svg



logo.svg



logo.svg



logo.svg

Al hacer **commit** a un archivo o archivos, se guarda cierta información junto con los cambios realizados en el archivo

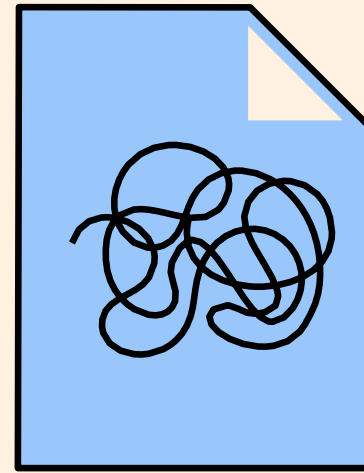
1. ¿Quién?
2. ¿Cuándo?

Puedes añadir más
información sobre los
cambios que has
realizado en un **commit
message**

Un buen mensaje para un commit:

Actualizar el estilo de los enlaces

La investigación de usuarios ha demostrado que muchas personas no detectaban los enlaces en el texto. Este commit actualiza el estilo de los enlaces al nuevo estilo subrayado, que funciona mejor.



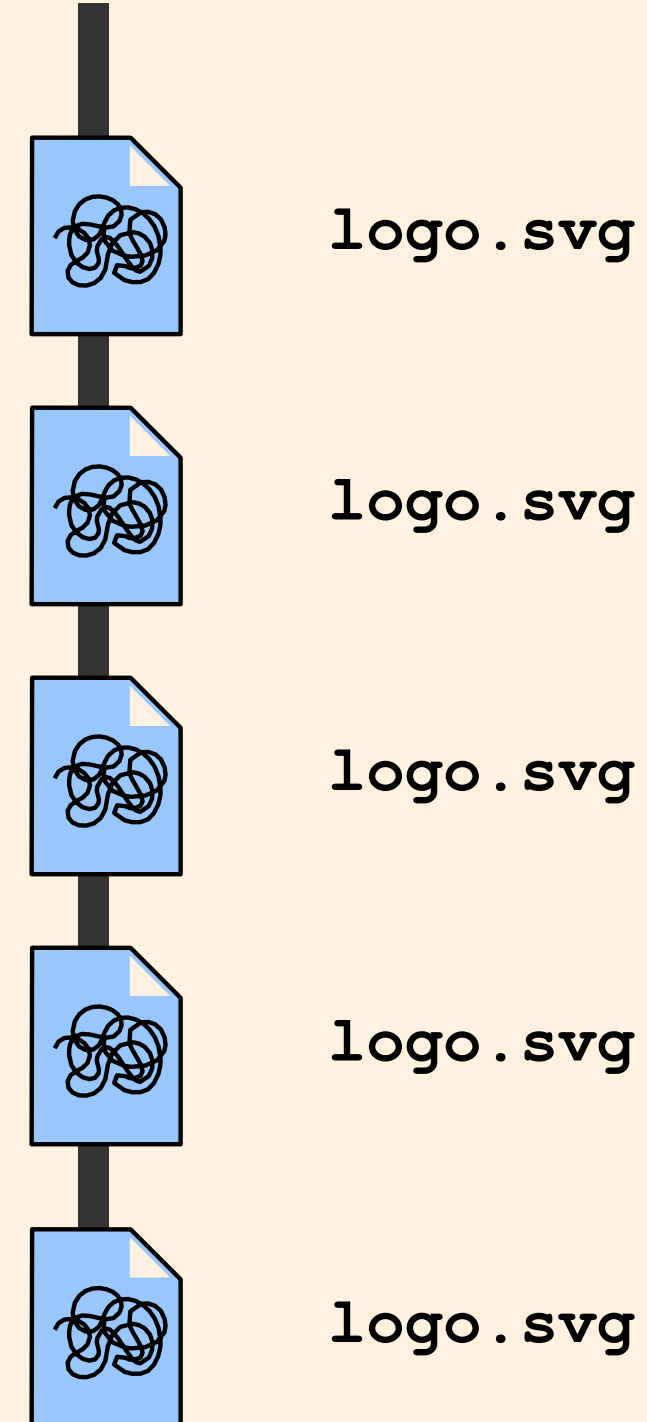
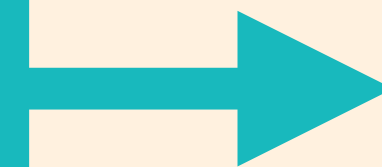
logo-3-FINAL-1.svg

Alice Bartlett
10:34am March 11th
2016

Actualizar el estilo de los enlaces

La investigación de usuarios ha demostrado que muchas personas no detectaban los enlaces en el texto. Este commit actualiza el estilo de los enlaces al nuevo estilo subrayado, que funciona mejor.

Haciendo commits

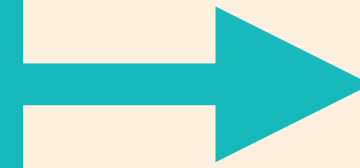




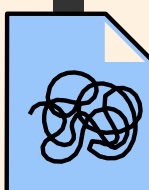

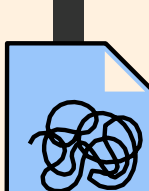
Haciendo commits

Alice Bartlett
12:43pm May 5th 2016

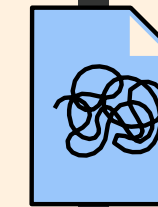
Agregar nuevos colores

Nuevos colores para la campaña electoral estadounidense

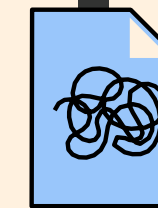


-  logo.svg
-  logo.svg
-  logo.svg
-  logo.svg
-  logo.svg

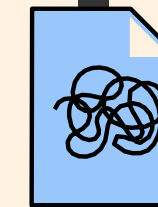
Haciendo commits



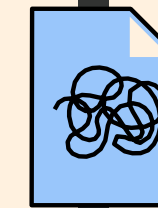
logo.svg



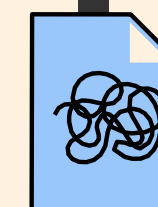
logo.svg



logo.svg



logo.svg



logo.svg

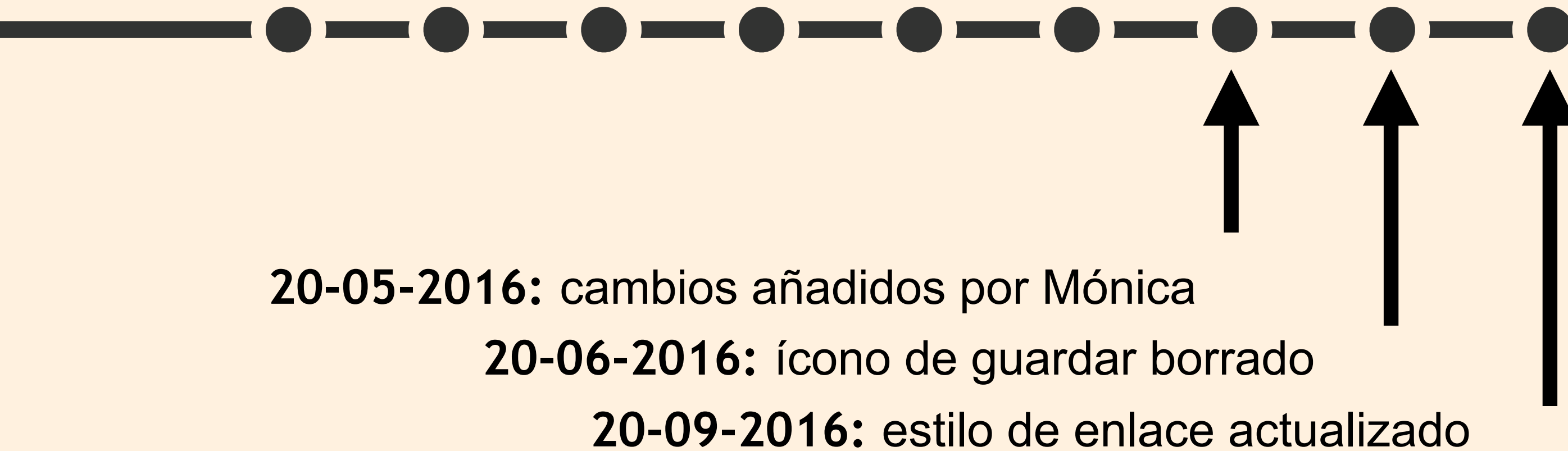
Alice Bartlett

12:43pm May 8th 2016

Corregir Naranja

El naranja que utilizamos no supera las pruebas de contraste de accesibilidad de la AAA, así que lo mejoré para que contraste correctamente.

Git almacena todo el historial de tu proyecto



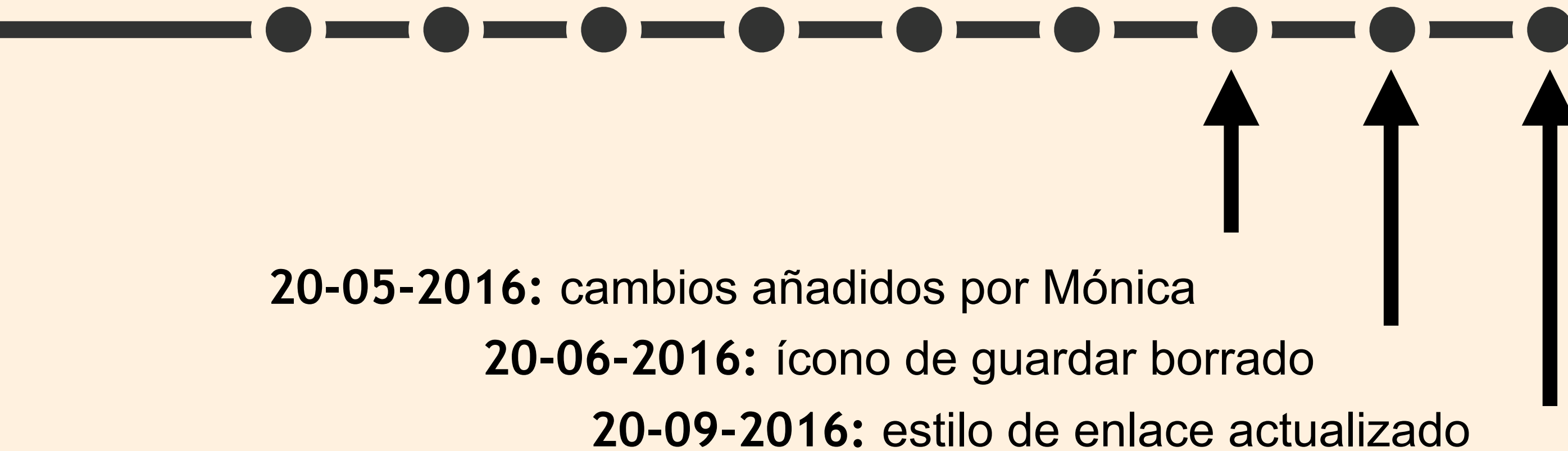
repositorio - La carpeta de tu proyecto
commit - Guardar una imagen

Asunto 2:

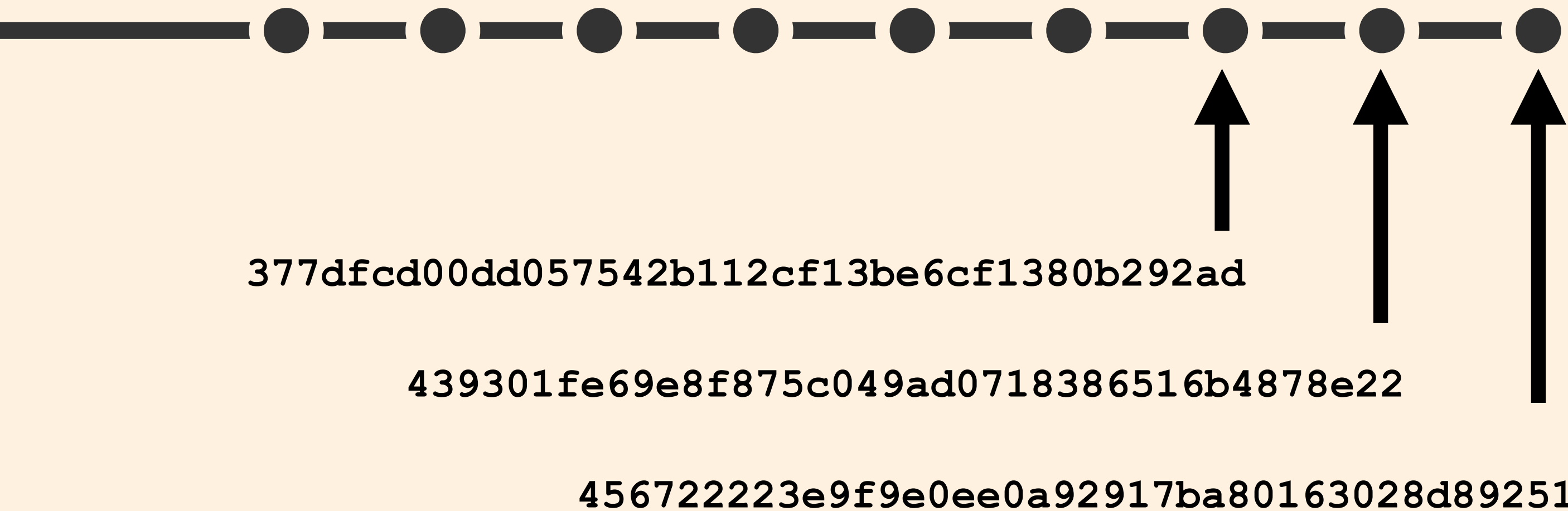
GIT TE PERMITE VIAJAR EN EL TIEMPO

**Una vez que hayas
guardado algunas
imágenes, Git te permite
moverte por ellas**

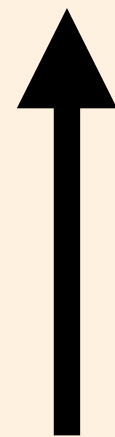
Git almacena todo el historial de tu proyecto



Cada uno de estos commits tiene un id llamado **hash**



**Puedo decirle a Git qué commit quiero
comprobar utilizando el hash del commit**



20-05-2016: ícono de guardar borrado

d5b87865bc2cd9d38ba8284c2eaa0d0241d800bb

Obtener los archivos de
un commit en el pasado
se conoce como hacer
un **check out**

**Puedo decirle a Git qué commit
quiero comprobar utilizando el hash
del commit**



20-05-2016: ícono de guardar borrado

d5b87865bc2cd9d38ba8284c2eaa0d0241d800bb

**Puedo decirle a Git qué commit quiero
comprobar utilizando el hash del
commit**



20-05-2016: ícono de guardar borrado

d5b87865bc2cd9d38ba8284c2eaa0d0241d800bb

Mis otros commits siguen existiendo, pero cuando miro en mi repo, es como si nunca hubieran ocurrido



20-05-2016: ícono de guardar borrado

d5b87865bc2cd9d38ba8284c2eaa0d0241d800bb

hash - un identificador generado por ordenador

checkout - viajar en el tiempo a un commit específico

Asunto 3:

GIT TE AYUDA A EXPERIMENTAR

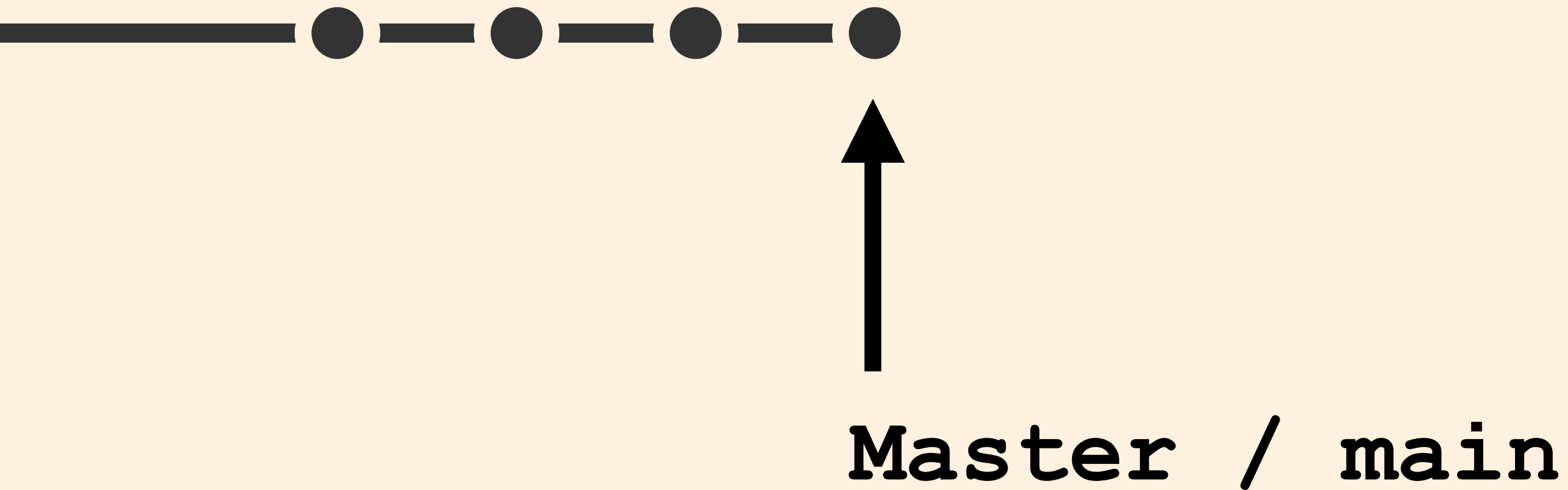
**Hasta ahora, todo ha sido
muy lineal y ordenado.**

**Así no es como funcionan los
proyectos, a veces quieres
hacer experimentos
fácilmente descartables**

**La forma de hacerlo en
Git es con **branches**
(**ramas**)**

Un **branch** es una etiqueta
móvil adjunta a un commit

El nombre de rama por defecto en Git es
master/main



@mxrtinez

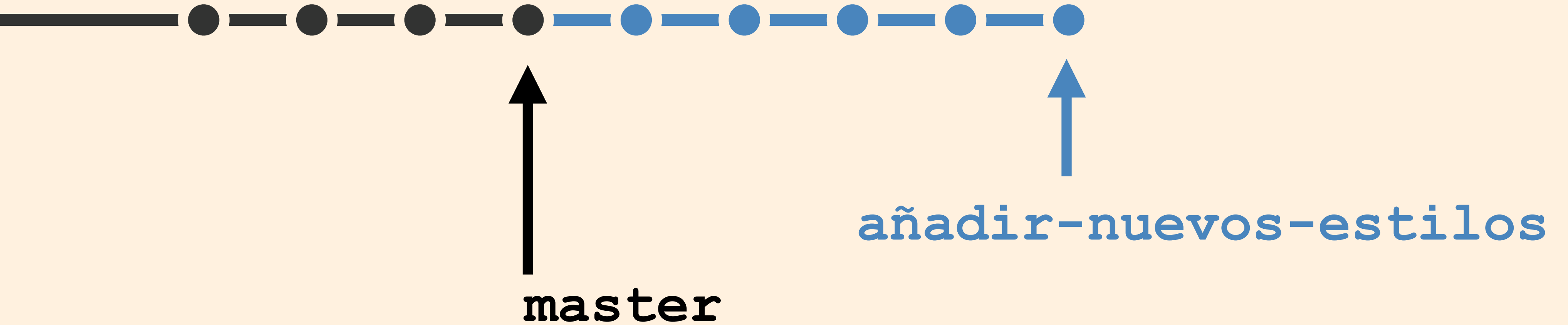
También puedes añadir tus propias ramas

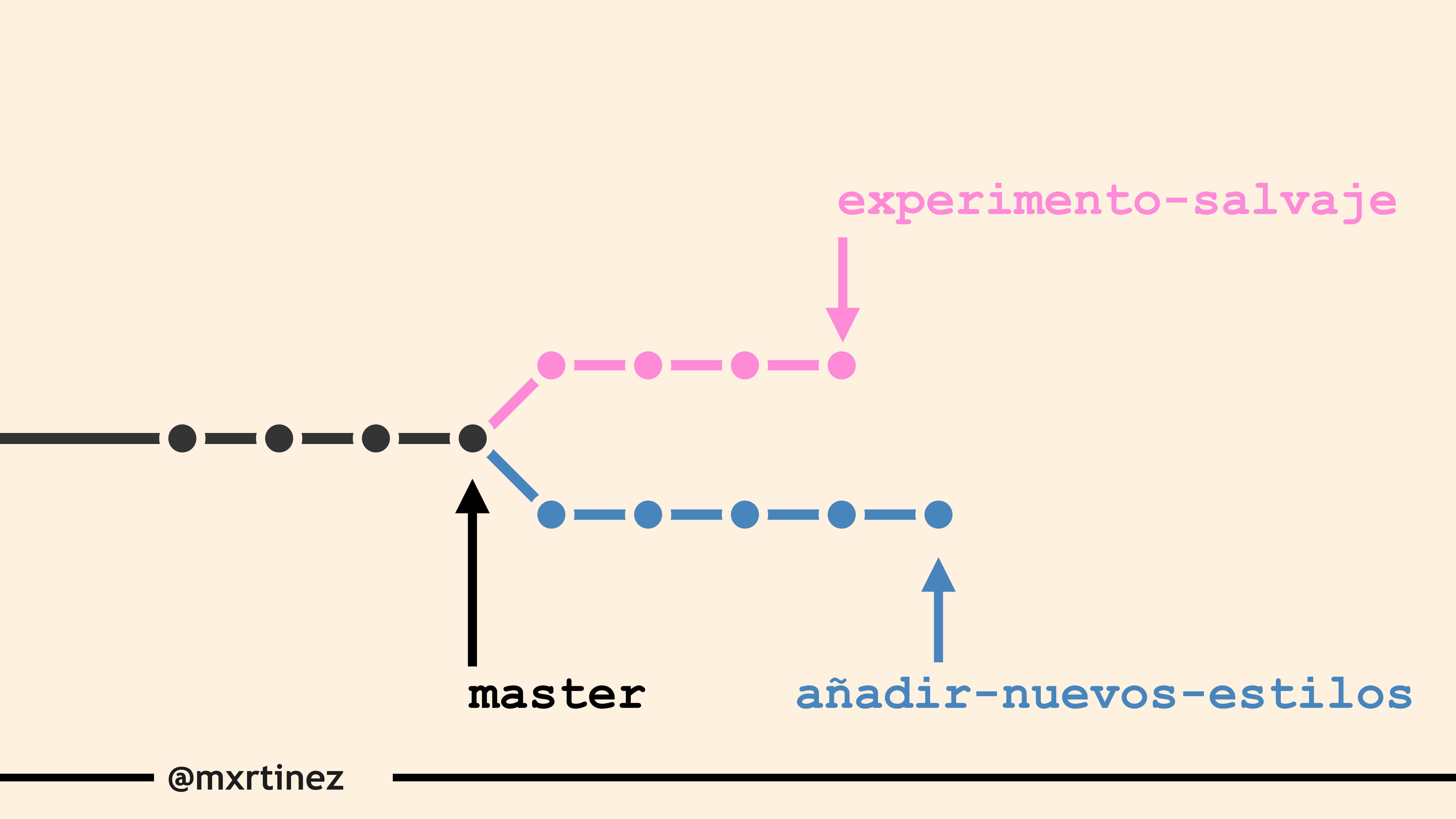


`Añadir-nuevos-estilos`

`master`

El desarrollo suele realizarse en una rama





experimento-salvaje

master

añadir-nuevos-estilos

@mxrtinez

Las ramas son útiles para probar cosas, ya que son muy fáciles de desechar si decides que no te gustan los cambios.

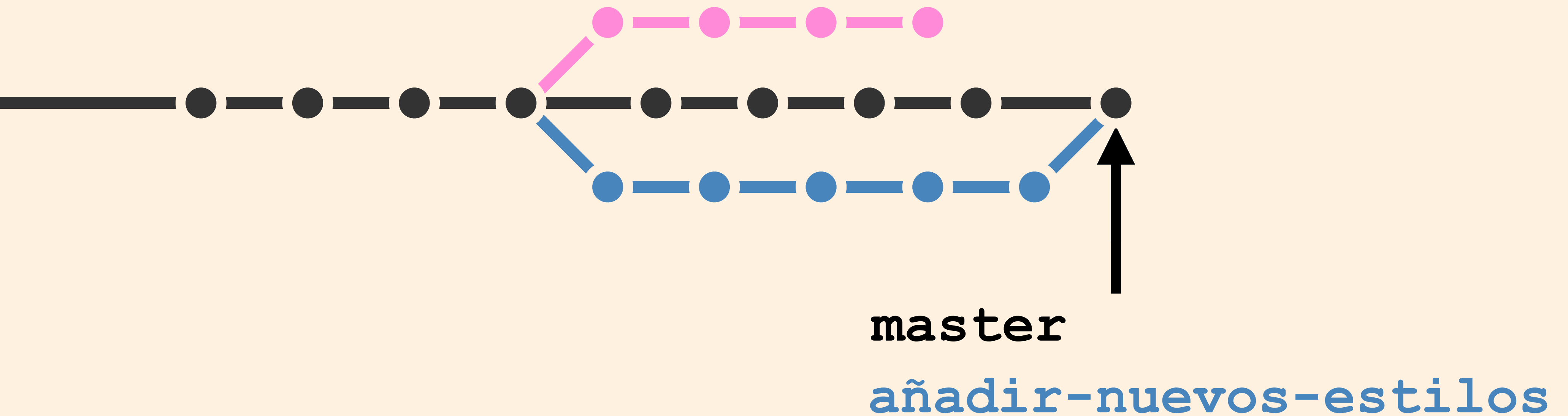
La mayor parte del trabajo de desarrollo se realiza en ramas, la rama master se considera especial

Es habitual que la rama master sea la versión del código o los archivos que están activos en el sitio.

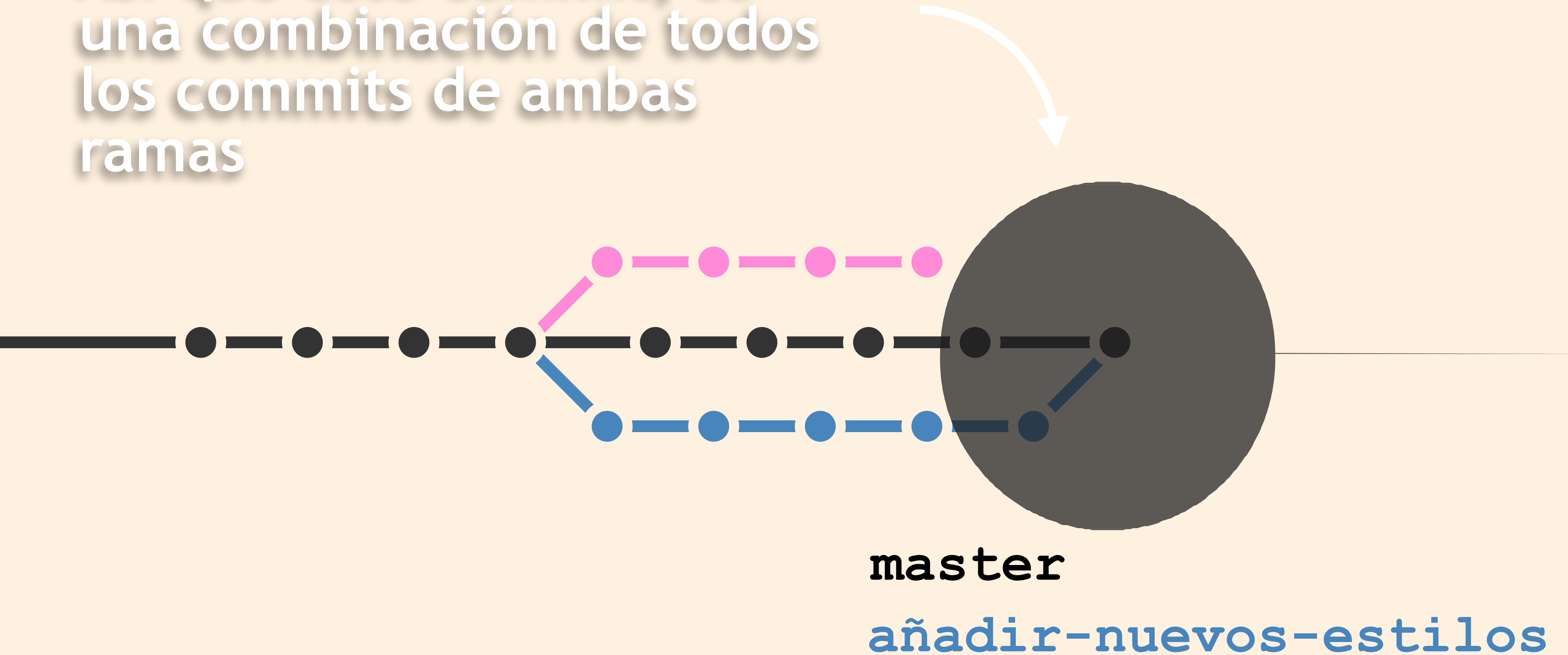
**Mientras que otras ramas
pueden contener trabajos
en curso**

**Una vez que estés
satisfecho con algún
trabajo, necesitas una
forma de volver a ponerlo
en master**

Para pasar los cambios de una rama a otra,
hay que hacer un **merge**



Así que este commit, es una combinación de todos los commits de ambas ramas



branch - una etiqueta móvil que señala a un
commit

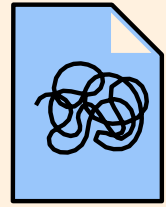
merge - la combinación de dos o más ramas

Asunto 4:

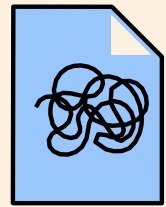
**GIT TE AYUDA A HACER
COPIAS DE SEGURIDAD DE
TU TRABAJO**

**Todo el mundo sabe que
hay que hacer copias de
seguridad periódicas.**

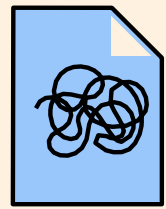
**Idealmente a un lugar
geográficamente distinto
de tu ordenador**



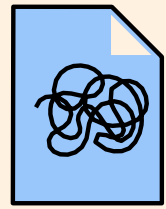
logo.svg



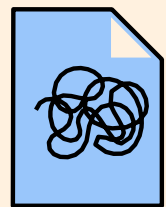
logo-2.svg



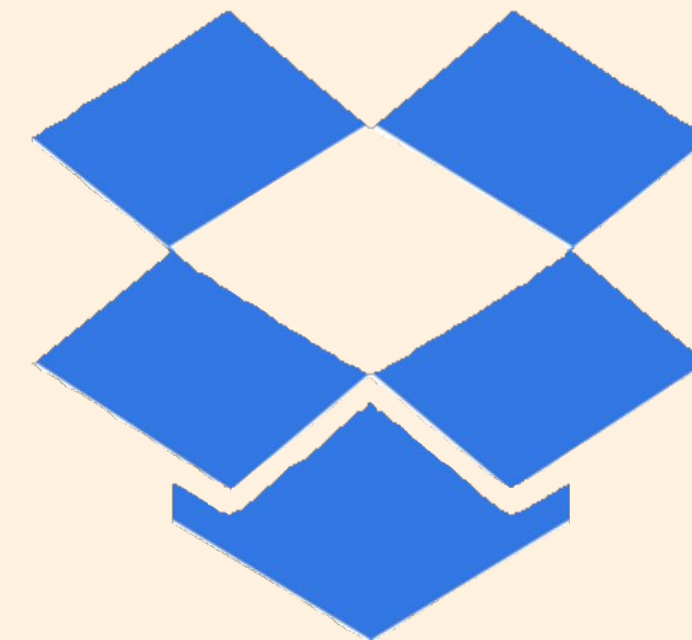
logo-3-monica-feedback.svg



logo-3-FINAL.svg

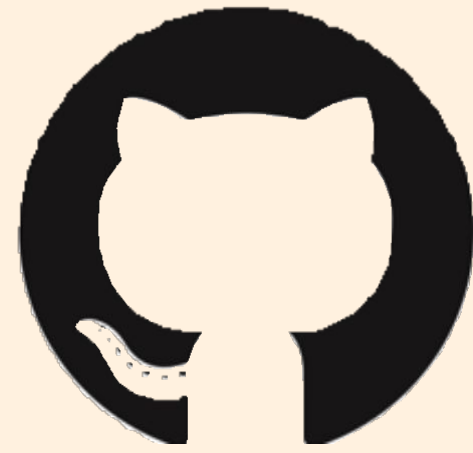


logo-3-FINAL-1.svg



- **Más seguro**
- **Acceso desde distintos lugares**
- **Acceso compartido**

En Git este lugar se
denomina **remote**



Unos muy populares son
GitHub / GitLab

G-Lab 

Para trabajar sobre algún
remote debes hacer un
clone



Remote



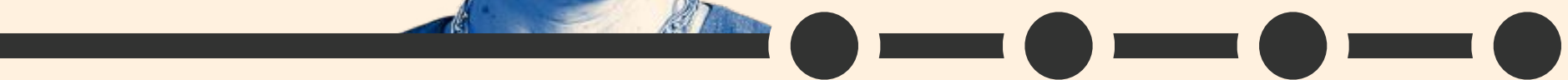
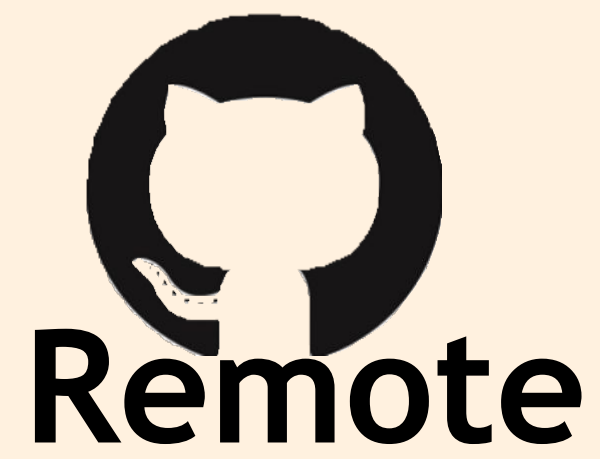
@mxrtinez



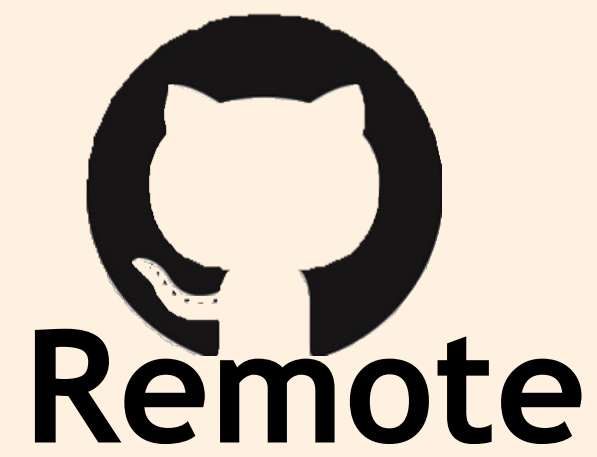
Remote



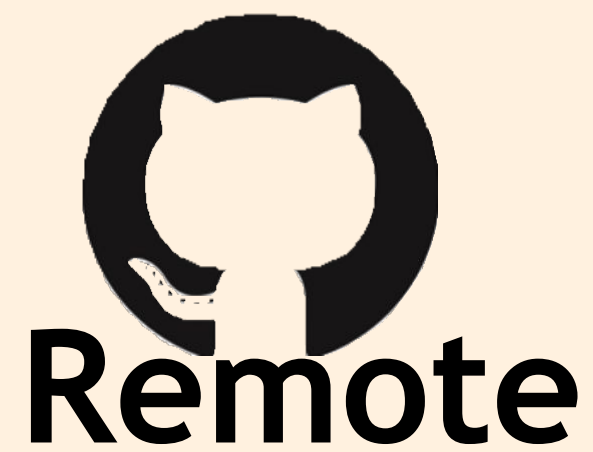
@mxrtinez



@mxrtinez



Ahora todo el mundo tiene el repositorio en su ordenador

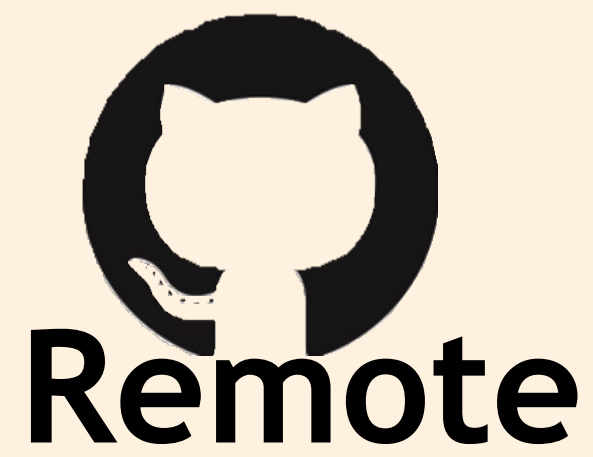


Lucy Kellaway
10:34am November 4th
2016
Arreglar el tintado roto de los iconos

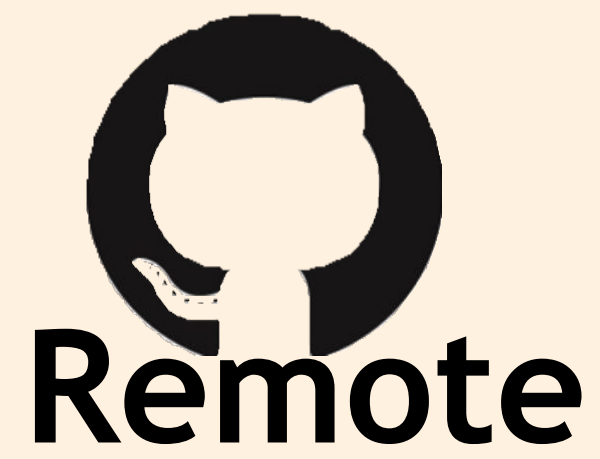
El tintado de iconos distinguía entre mayúsculas y minúsculas, por lo que `#FFF` funcionaba pero `#fff` no. Este commit elimina este error.



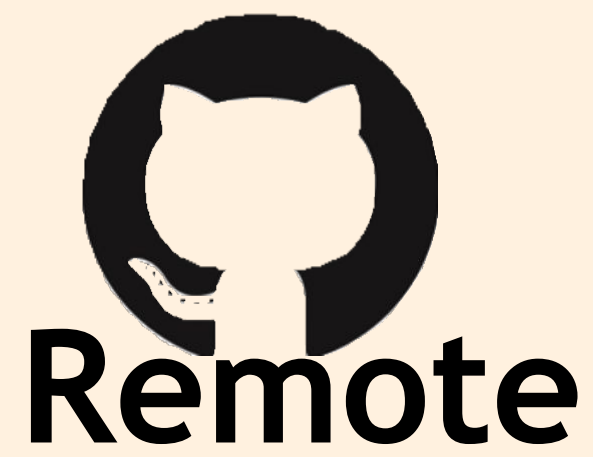
@mxrtinez



Lucy puede
enviar sus
cambios al
remote



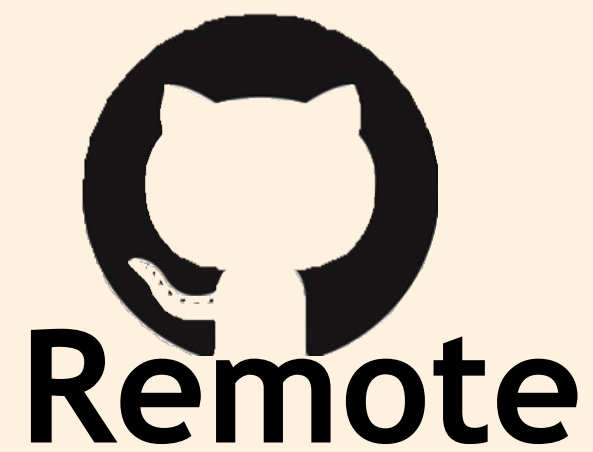
@mxrtinez



Esto se conoce
como **push**



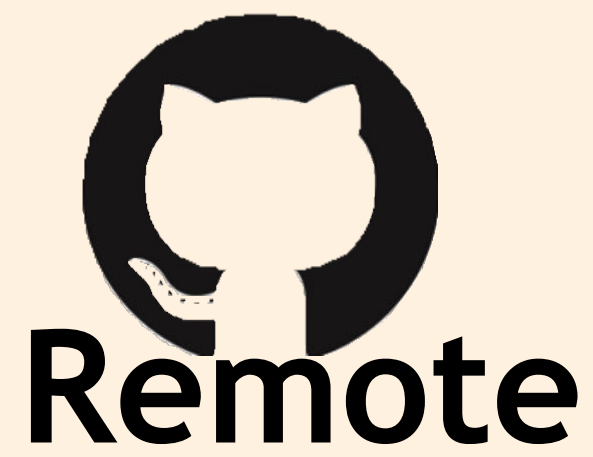
@mxrtinez



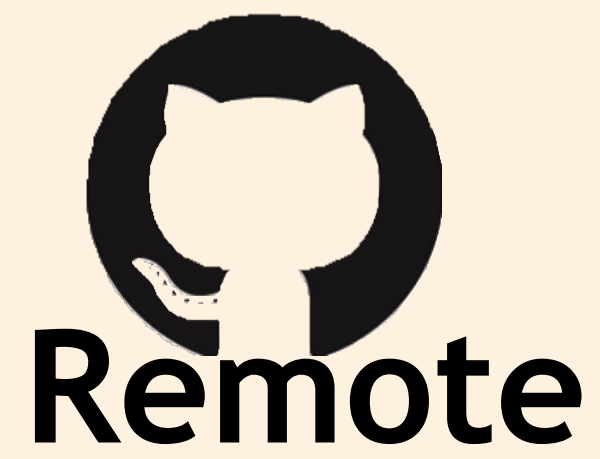
Ahora *Martin* está desactualizado



@mxrtinez



Para obtener estos cambios, Martin tendrá que hacer un **pull**



@mxrtinez

remote - un ordenador con un repositorio

clone - obtener el repositorio del remoto por primera vez

pull - obtener nuevos commits en el repositorio desde el remoto

push - envía tus nuevos commits al remoto

Asunto 5:

GIT TE AYUDA A COLABORAR

**Committing te ayuda a contar
a los demás la historia de tu
proyecto**

**Los remotes permiten que
otras personas accedan a
tu proyecto**

**Un merge ayuda a combinar
tu trabajo con el de otras
personas**

Git permite que mucha gente trabaje en el mismo proyecto, por eso la gente sufre su terrible UX.

Términos Git que hemos tratado

- repository** la carpeta de tu proyecto
- commit** una imagen de tu repo
- hash** un id de un commit
- checkout** viaje en el tiempo a un commit específico
- branch** una etiqueta móvil que señala un commit
- merge** combinar dos ramas
- remote** un ordenador con el repositorio
- clone** obtener el repositorio desde un remoto por primera vez
- push** enviar commits a un remote
- pull** obtener commits desde un remote

1. **Cuenta la historia de tu proyecto**
2. **Viajar en el tiempo**
3. **Experimentar con cambios**
4. **Hacer copias de seguridad**
5. **Colaborar en proyectos**

Gracias

Alexander Martinez Mendez

@mxrtinez

Es hora de usar git

1.