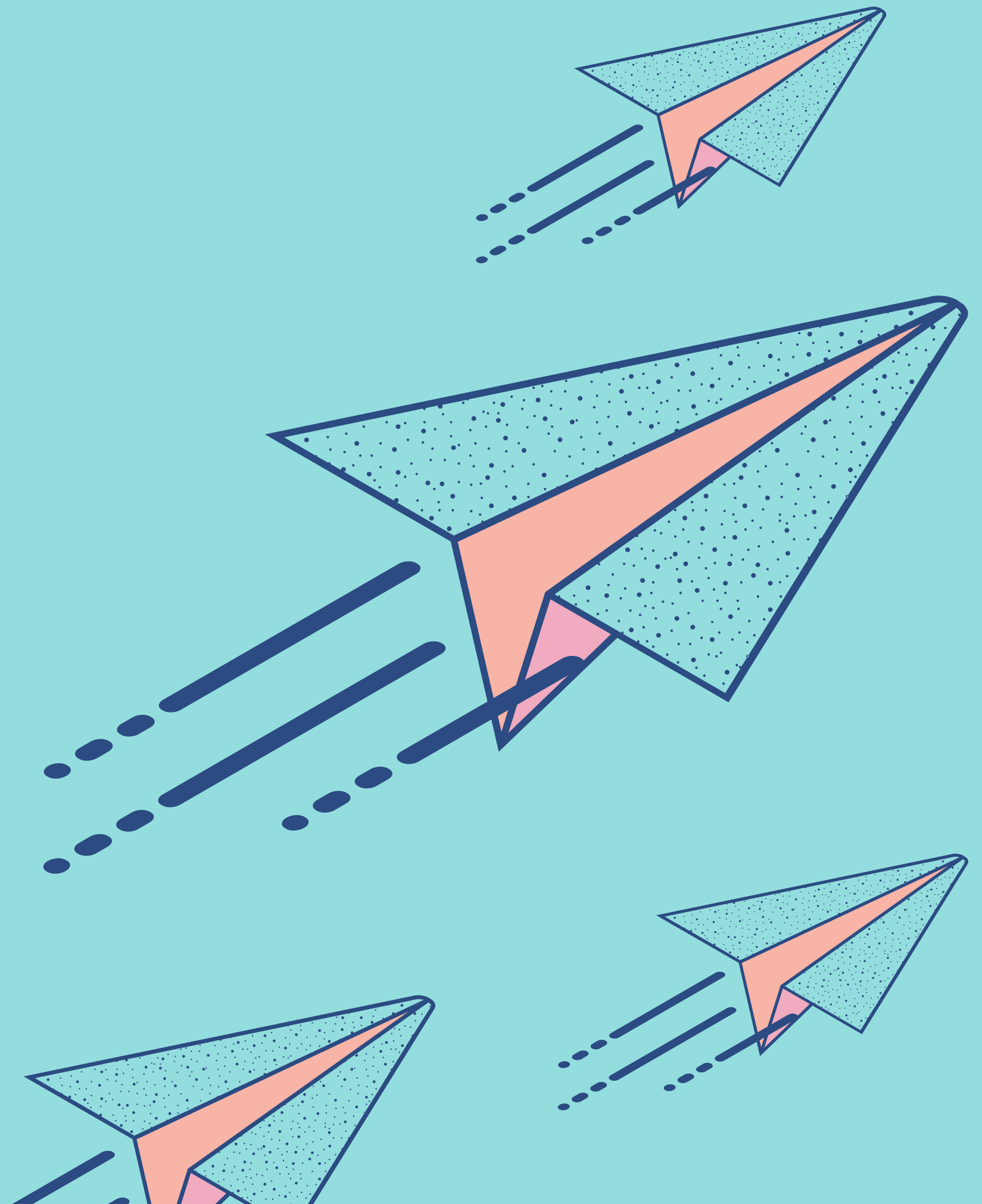




Ramas, merge y conflictos



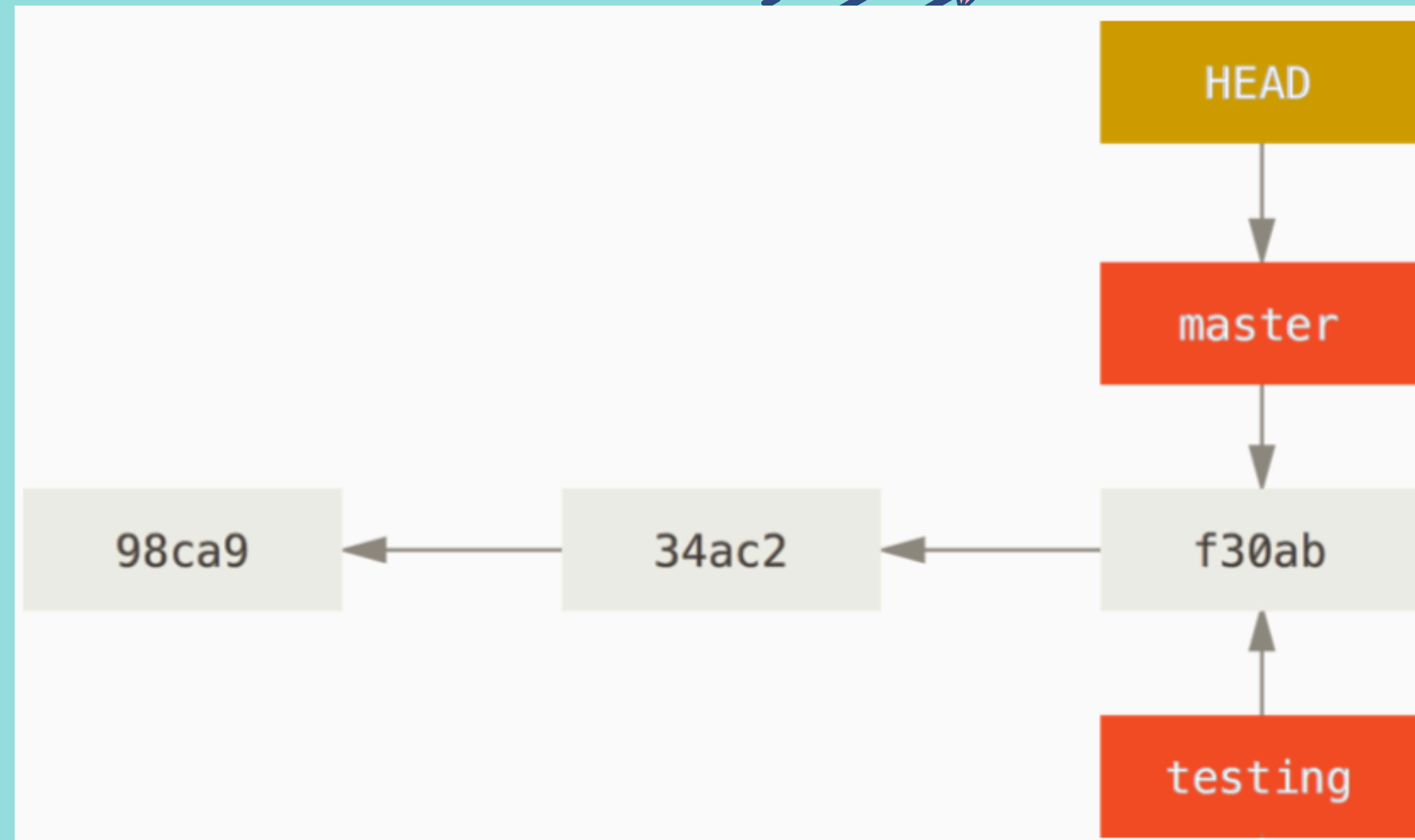
¿Qué es una
rama?



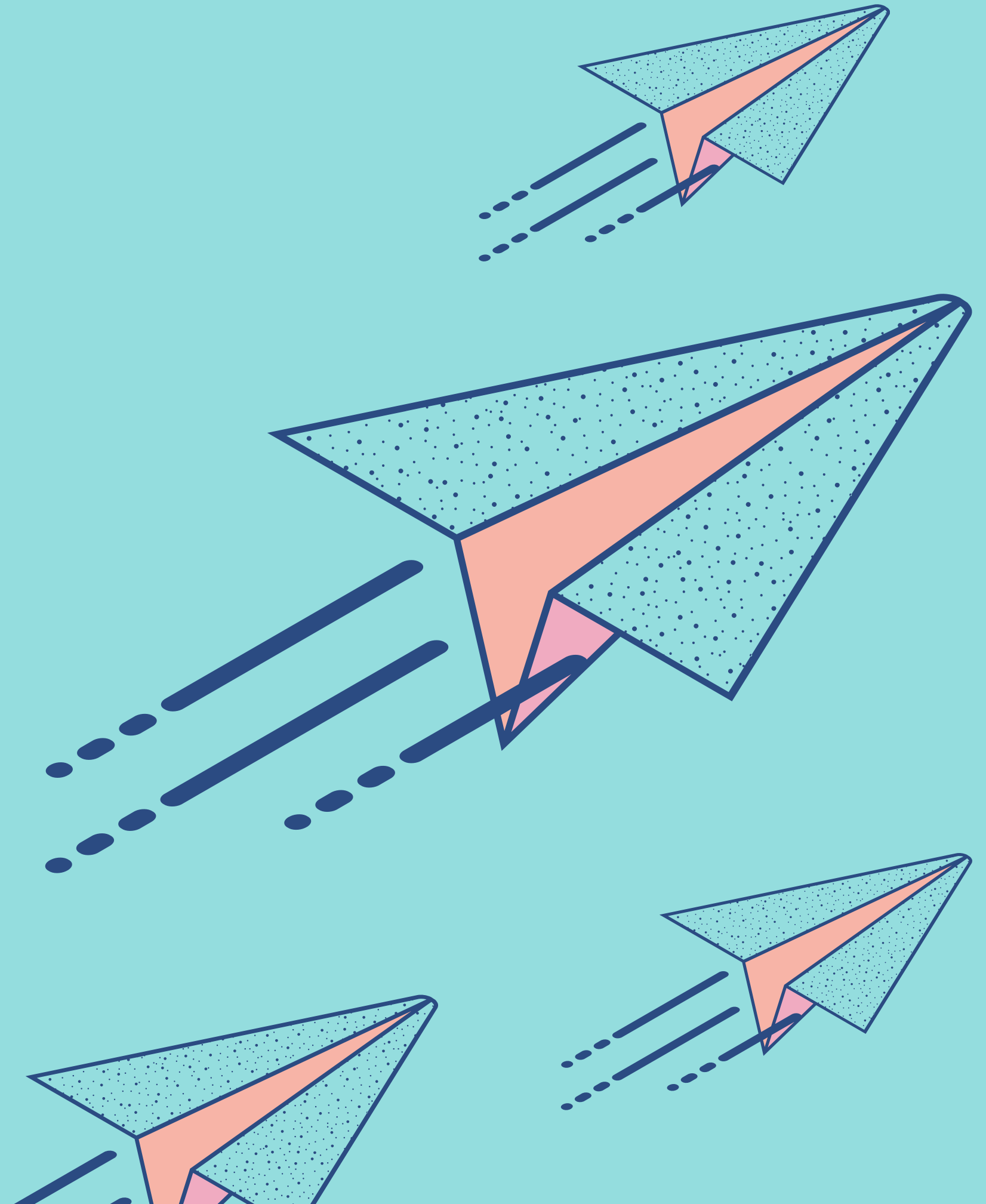
¿Qué es una rama?

Son una instantánea (snapshot) de la división del estado del código.

A nivel técnico, es un nuevo apuntador hacia una de las confirmaciones.

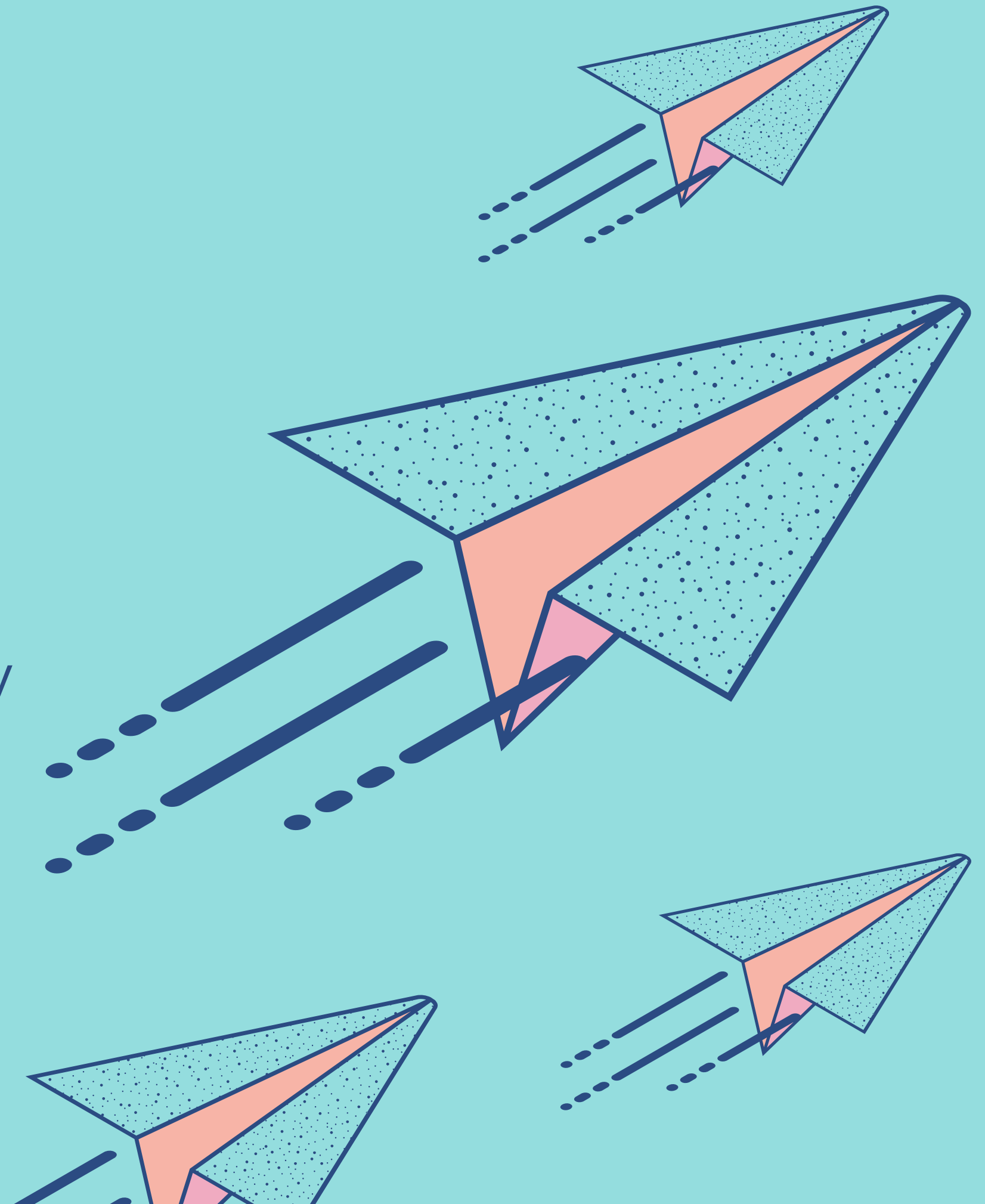


¿Para qué sirven
las ramas?



¿Para qué sirven las ramas?

Permiten realizar un desarrollo no lineal y colaborativo.



Creando nuestra primera rama

El comando `git branch` nos permite crear, listar, eliminar y renombrar ramas. Al crear una rama, para movernos a ella, tendremos que usar otro comando: `git switch`.

```
# creamos la rama mi-primera-rama
$ git branch mi-primera-rama

# cambiamos a la rama mi-primera-rama
$ git switch mi-primera-rama
Switched to branch 'mi-primera-rama'
```

Si quieres hacer los dos pasos a la vez, puedes usar el comando `git switch -c mi-primera-rama`. Esto creará la rama y te llevará a ella con un sólo comando. Ten en cuenta que, si el nombre de la rama ya existe, recibirás un error:

```
# creamos la rama mi-primera-rama
$ git switch -c mi-primera-rama

# intentamos crear una rama con el mismo nombre
$ git switch -c mi-primera-rama
fatal: A branch named 'mi-primera-rama' already exists.
```

Para poder crear una rama debes, al menos, tener un commit en el repositorio. Si ejecutas `git init` y lo primero que intentas es crear una rama con `git branch` verás que recibes un error. Tiene sentido. ¿Cómo se podría crear una rama si no existe ningún tronco del que sostenerla?

Fusionar ramas

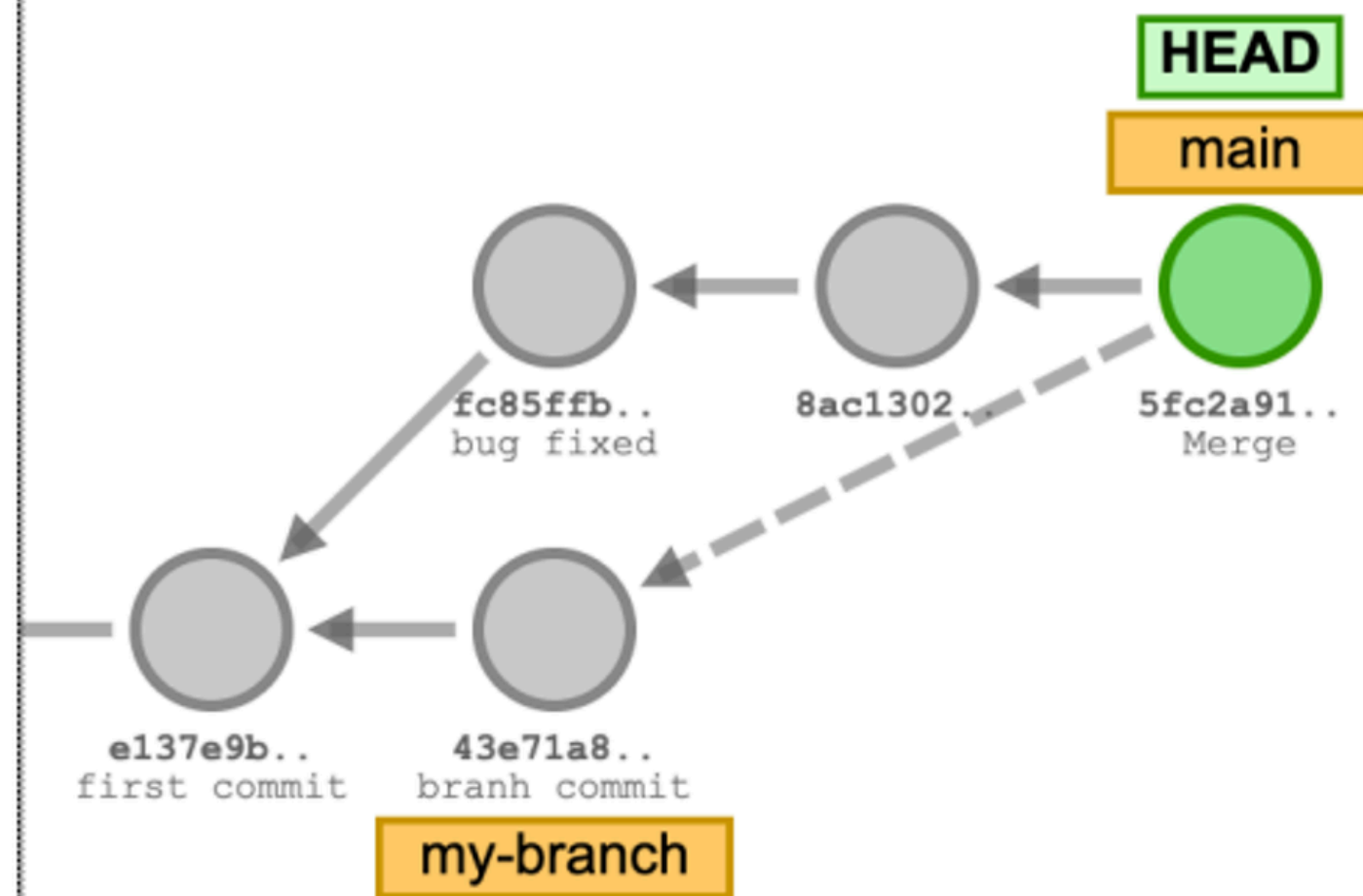
Las bifurcaciones de código que hemos creado en forma de ramas tendrán dos destinos: acabar en el olvido para no terminar en ningún lado o ser fusionada en otra rama.

Cuando hablamos de fusión nos referimos a que los cambios que hemos realizado en la rama se integran en otra rama, de forma que el código que habíamos generado en la nueva rama se asimila en otra.



Fusionando ramas

Empleamos el comando *git merge* para incorporar los cambios de una rama a la rama en la que nos encontramos en ese momento.



Modificando el mensaje del commit

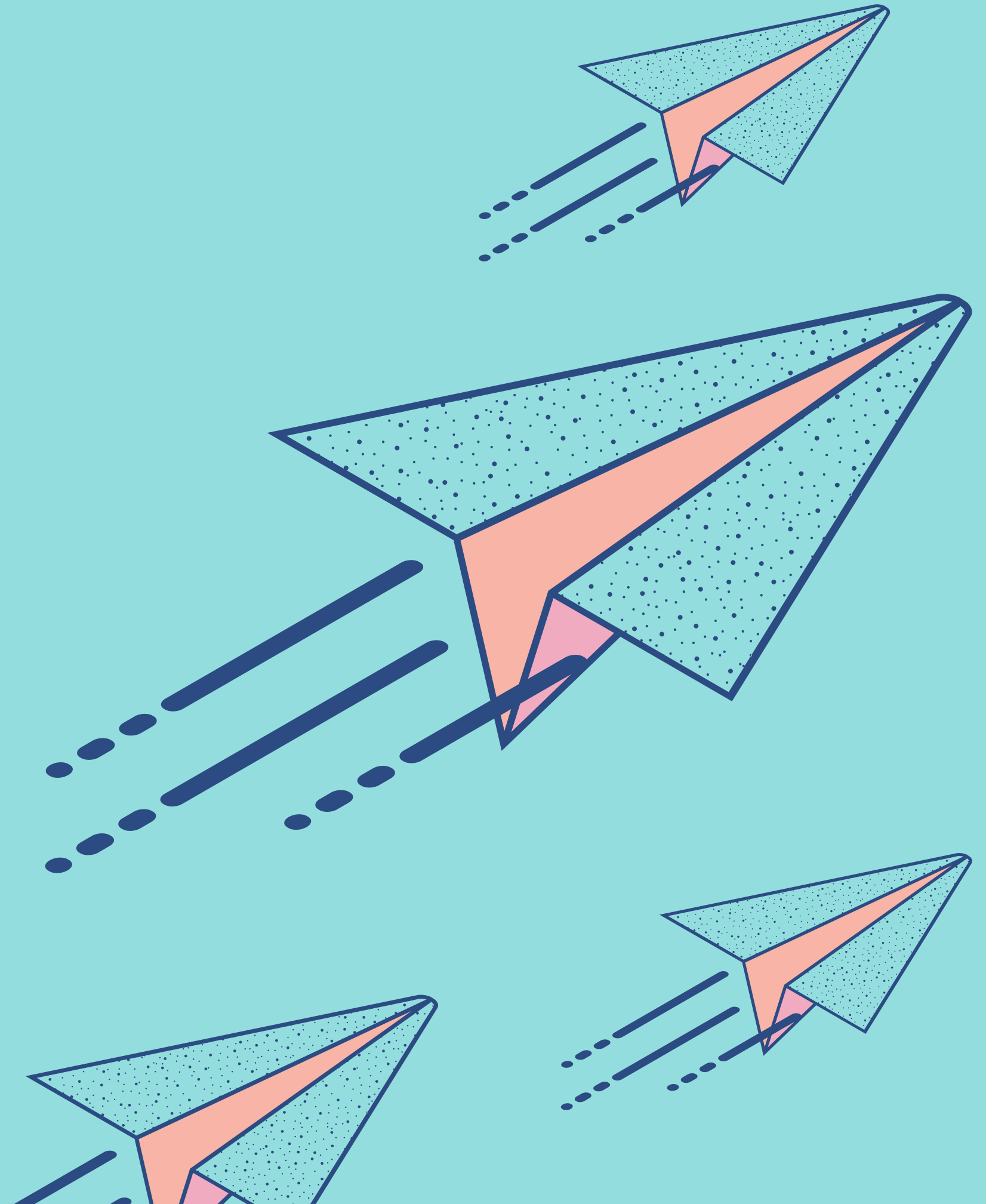
```
# Abre el editor antes de hacer el commit  
$ git merge --edit
```

```
# Evita que haga commit automáticamente  
$ git merge --no-commit
```

Aunque puede ser útil en algunos casos muy concretos... lo cierto es que lo normal es simplemente dejar que Git haga el commit automáticamente.

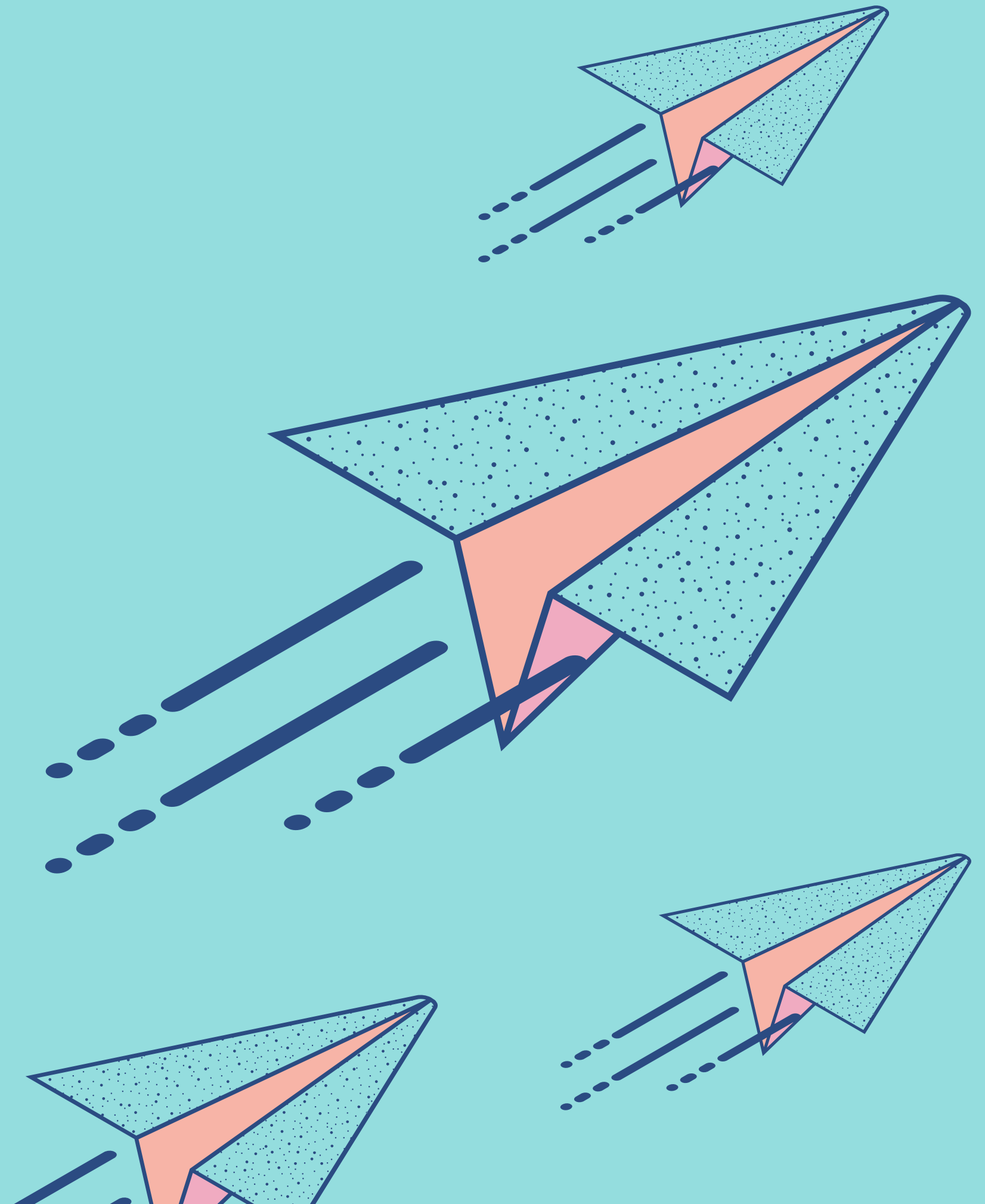
Al ejecutar el comando `git merge`, se crea un nuevo commit que incluye todos los cambios de la rama de origen a la rama en la que nos encontramos ahora

Eliminar ramas ¿por qué?



Eliminar ramas ¿por qué?

Porque es una buena práctica, además que las ramas tiene un propósito único y corto de periodo.



Eliminando ramas

Después de fusionar una rama en otra rama (normalmente, la rama principal) es posible que quieras eliminarla para no dejarla suelta. Para ello puedes usar el comando `git branch` con el parámetro `--delete` o, de forma corta, `-d`.

```
# borramos la rama llamada "mi-primera-rama"
$ git branch --delete mi-primera-rama
Deleted branch mi-primera-rama (was 7c60765).
```

Si la rama ya ha sido fusionada previamente, entonces todo habrá ido correctamente y nos habrá devuelto un mensaje similar al que tienes arriba... Sin embargo, si la rama no la habías fusionado (*merge*) previamente, entonces te devolverá un error:

```
error: The branch 'mi-primera-rama' is not fully merged.
If you are sure you want to delete it, run 'git branch -D mi-primera-rama'.
```

Creo que el mensaje nos ha devuelto un *spoiler*, directamente. Y es que en el caso que quieras borrar una rama que no ha sido fusionada previamente, deberás usar el parámetro `-D`. Este parámetro le indica a Git que queremos borrar la rama sin importar si ha sido fusionada o no.

```
# borramos la rama llamada "mi-primera-rama"
$ git branch -D mi-primera-rama
Deleted branch mi-primera-rama (was 7c60765).
```

¿Por qué hace esta comprobación Git? Es sencillo. Borrar una rama que no ha sido fusionada... puede hacerte perder muchas horas de trabajo. Asegúrate de que estás seguro de querer borrarla antes de hacerlo.

Comandos

Para crear ramas, visualizar ramas, borrar ramas, realizar merges y otros relacionados.

git branch

git branch <nombre rama>

git branch -a

git branch -d <nombre rama>

git merge <nombre rama>

git switch <nombre rama>

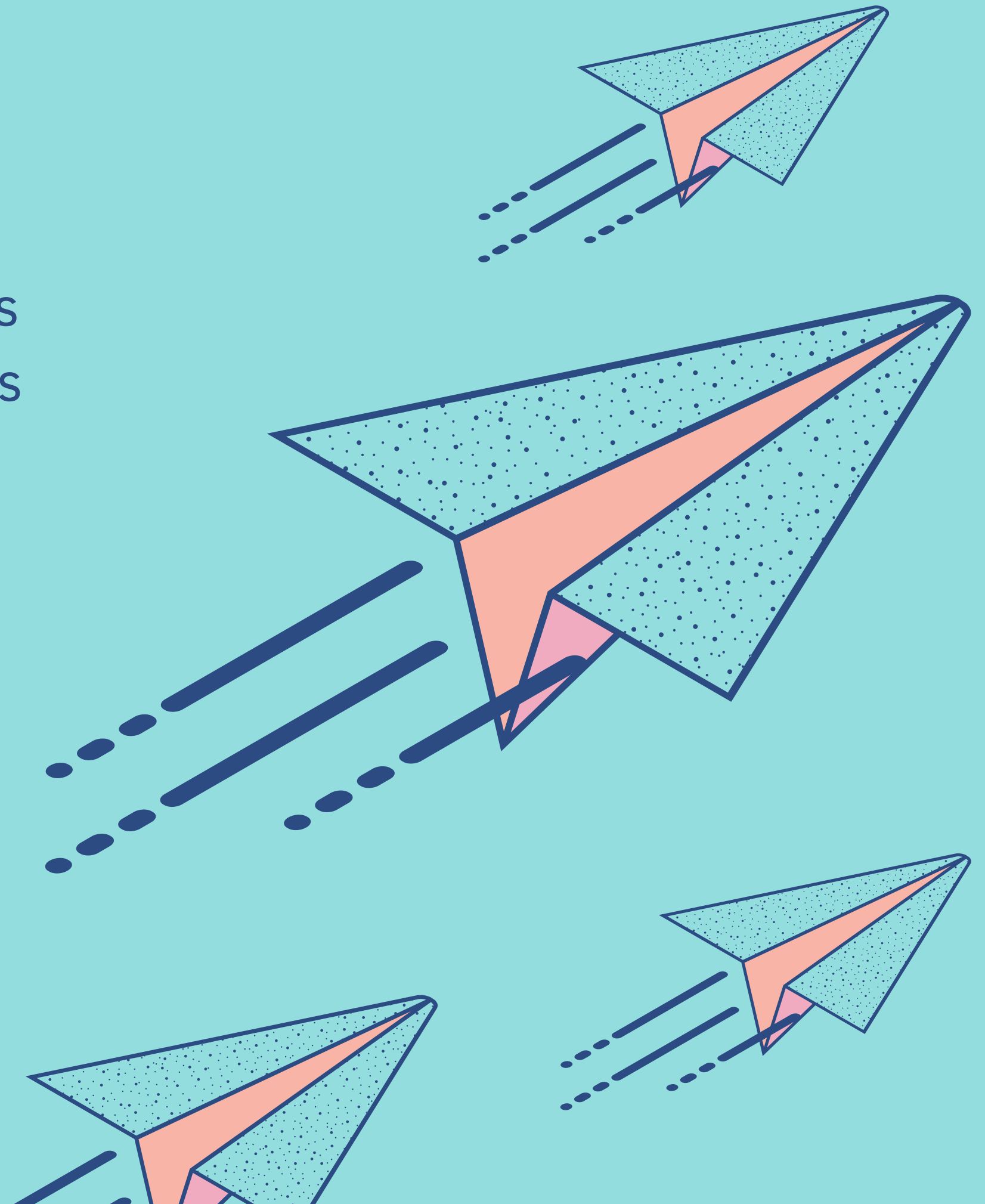
git merge <nombre rama> --no-ff

git checkout <nombre rama>

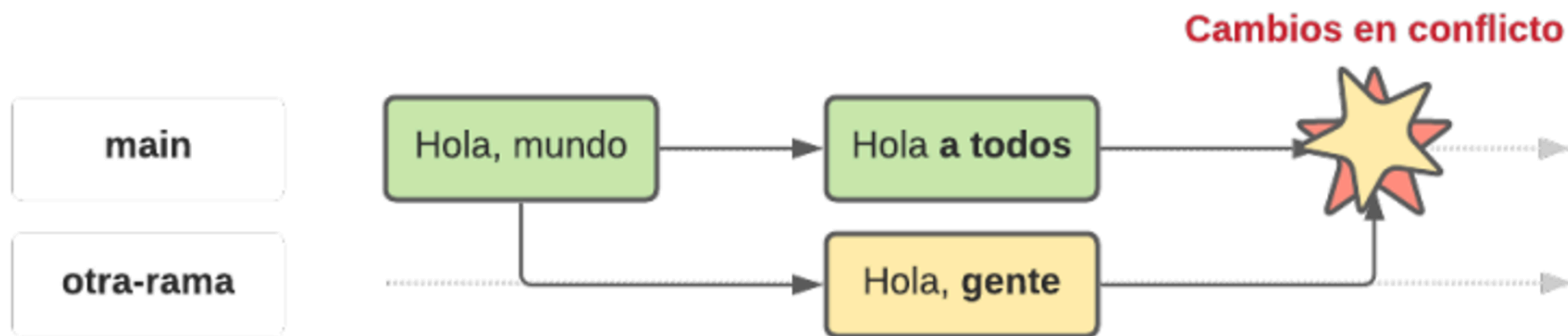
Conflictos en Git

¿Qué pasa si al querer fusionar dos ramas, la de destino ha realizado cambios en las mismas líneas de un fichero que los que queremos fusionar?

Tendríamos conflictos.



Un conflicto es una situación en la que **Git no es capaz de determinar qué cambio es el que tiene que prevalecer** una vez ocurra la fusión y, por lo tanto, requiere que el usuario lo resuelva.



Dada la naturaleza de sistema distribuido, es normal que a veces ocurran conflictos al intentar fusionar dos ramas en Git. ¿Cómo iba a saber Git qué cambio es más importante que otro?

Resolviendo conflictos

Al resolver, deberemos decidir entre:

- Nos quedamos con los cambios de la **rama main**.
- Nos quedamos con los cambios que vienen de la **rama changes**.
- Modificamos los cambios para hacer una fusión personalizada.

```
$ git diff
++<<<<<<< HEAD
+<p>This is our initial content</p>
+<p>New content to the file</p>
++=====
+ <p>Completely different content</p>
++>>>>>>> changes
```

Arriba tenemos el contenido que ya existía.

=====

Debajo tenemos el contenido de la rama que queremos incorporar.