

# **Git Advanced**

# **Reimplementando Git**

Leandro Acosta, Mateo Ostorero, Gonzalo Gigena y Joaquin de Francesca

# Estructura de la presentación



1. Git Internals - Repaso
  - a. Comandos de Git (Plumbing y Porcelain)
  - b. Git Objects
  - c. Git Reference
2. Index file
  - a. Cómo funciona
  - b. Nuestra versión simplificada
3. Git cherry pick
  - a. Cómo funciona
  - b. Cuándo se recomienda usarlo
4. Git rebase
  - a. Cómo funciona
  - b. Cuándo se recomienda usarlo
5. Demo práctica



# Git Internals

## Repaso

# Comandos de Git

- Porcelain (orientados a usuario)
  - 50 comandos
  - Con autocompletado en la Cli de Git
- Plumbing (núcleo de Git)
  - > 60 comandos
  - Sin autocompletado



## Porcelain

- `git init`
- `git add`
- `git commit`
- `git checkout`
- `git status`
- `git merge`
- ... y muchos más




# Plumbing

- `git hash-object`
- `git cat-file`
- `git update-index`
- `git ls-tree`
- `git read-tree`
- ... y muchos más



# Equivalente de git add con plumbing commands



```
→ test-add git:(master) x git hash-object -w README.md
3b18e512dba79e4c8300dd08aeb37f8e728b8dad
→ test-add git:(master) x git update-index --add --cacheinfo 100644 3b18e512dba79e4c8300dd08aeb37f8e728b8dad README.md
→ test-add git:(master) x git status


On branch master

No commits yet

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

→ test-add git:(master) x
```

# Equivalente de git commit con plumbing commands



```
→ test-add git:(master) ✕ git write-tree
```

```
43b71c903ff52b9885bd36f3866324ef60e27b9b
```

```
→ test-add git:(master) ✕ git commit-tree 43b71c903ff52b9885bd36f3866324ef60e27b9b -m "First commit"
```

```
92cb30107553af237cd00b84324e241b3d554181
```

```
→ test-add git:(master) ✕ git cat-file -p 92cb30107553af237cd00b84324e241b3d554181
```

```
tree 43b71c903ff52b9885bd36f3866324ef60e27b9b
```

```
author Leandro Acosta <leacosta97@gmail.com> 1666749401 -0300
```

```
committer Leandro Acosta <leacosta97@gmail.com> 1666749401 -0300
```

```
First commit
```

```
→ test-add git:(master) ✕ git update-ref refs/heads/master 92cb30107553af237cd00b84324e241b3d554181
```

```
→ test-add git:(master) █
```



# Spoiler de la implementación de commit



```
1 def commit(message):
2     """
3     Record changes to the repository.
4
5     Create a new commit containing the current contents of the index and the given log
6     message describing the changes. The new commit is a direct child of HEAD.
7
8     https://git-scm.com/docs/git-commit
9     """
10    tree_sha1 = plumbing.write_tree()
11    commit_sha1 = plumbing.commit_tree(tree_sha1, message)
12    current_branch = plumbing.get_current_branch()
13    plumbing.update_ref(current_branch, commit_sha1)
14    return commit_sha1
```

# Git Objects

1. Blob (Binary large object)
2. Tree
3. Commit

5b1d3..

blob	size
<pre>#ifndef REVISION_H #define REVISION_H  #include "parse-options.h"  #define SEEN (1u&lt;&lt;0) #define UNINTERESTING (1u #define TREESAME (1u&lt;&lt;2)</pre>	

c36d4..

tree	size	
blob	5b1d3	README
tree	03e78	lib
tree	cdc8b	test
blob	cba0a	test.rb
blob	911e7	xdiff

ae668..

commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	



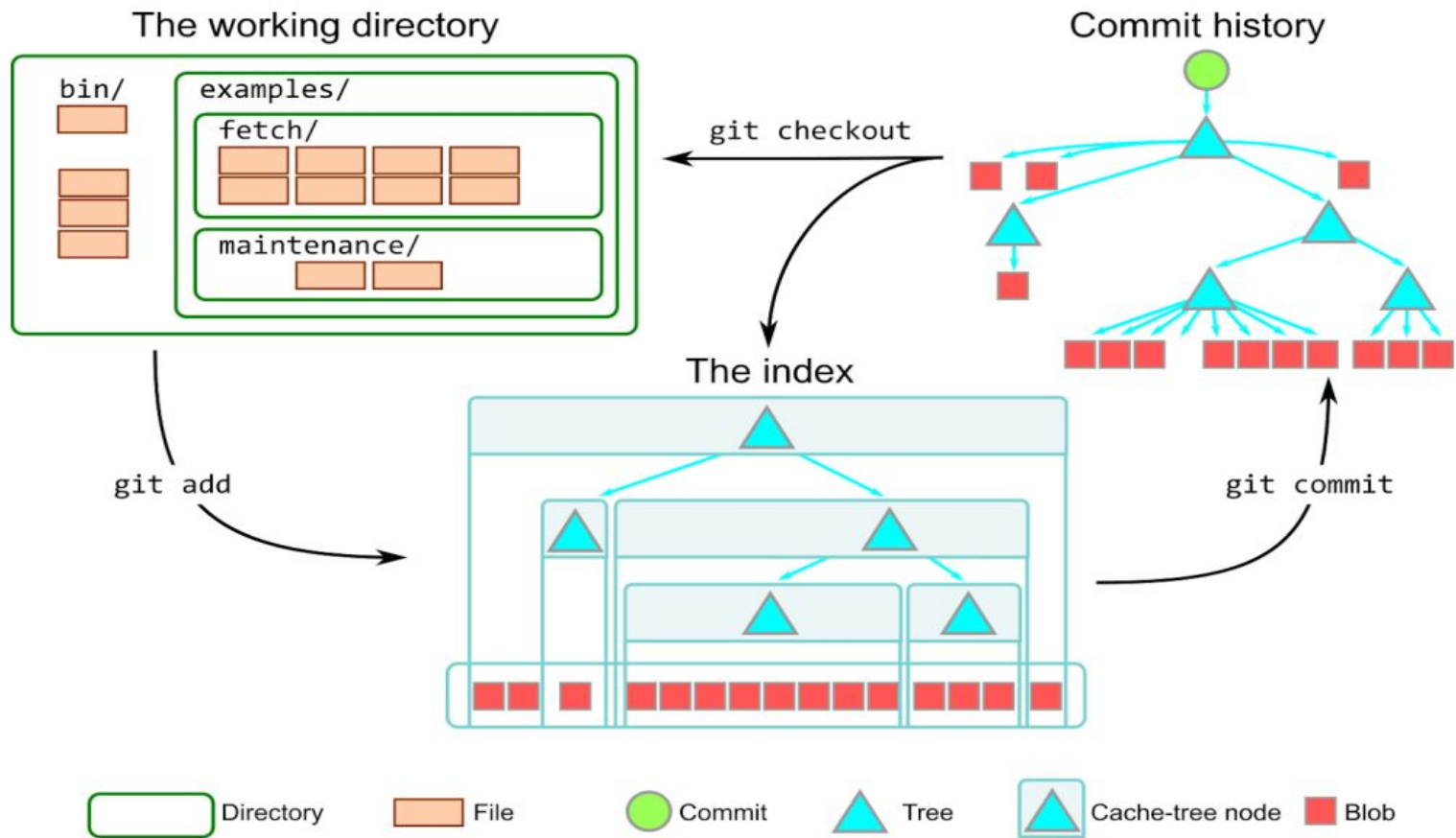
## Comandos implementados

1. init
2. log
3. status
4. add
5. commit
6. hash-object
7. cat-file
8. ls-tree
9. cherry-pick (versión simplificada sin merge)
10. rebase



# Index File

# Index file



[Formato original del index file](#)



## Nuestra versión simplificada del Index File

☰ index    ✕

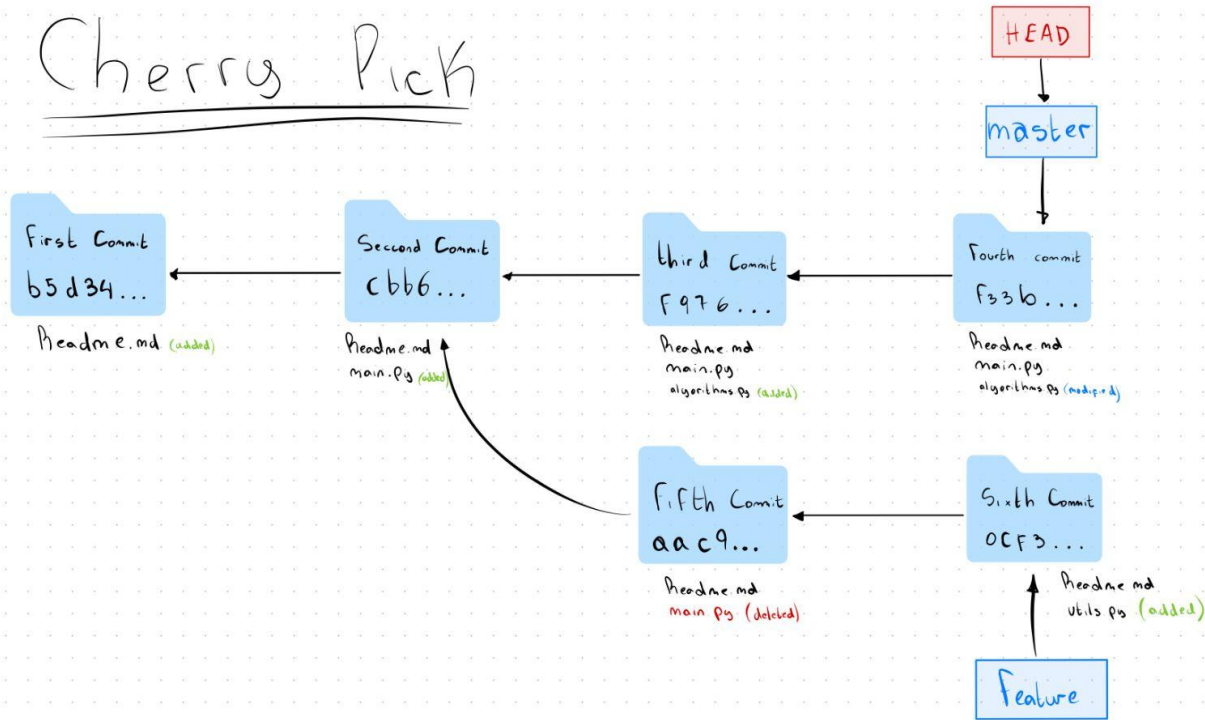
Users > leandroacosta > Facultad > calp > tests > .calp > ☰ index

- 1    d00491fd7e5bb6fa28c517a0bb32b8b506539d4d 1.txt
- 2    0cfbf08886fca9a91cb753ec8734c84fcbe52c9f A/2.txt
- 3    7ed6ff82de6bcc2a78243fc9c54d3ef5ac14da69 A/5.txt
- 4    1e8b314962144c26d5e0e50fd29d2ca327864913 B/C/6.txt
- 5    7f8f011eb73d6043d2e6db9d2c101195ae2801f2 B/7.txt
- 6



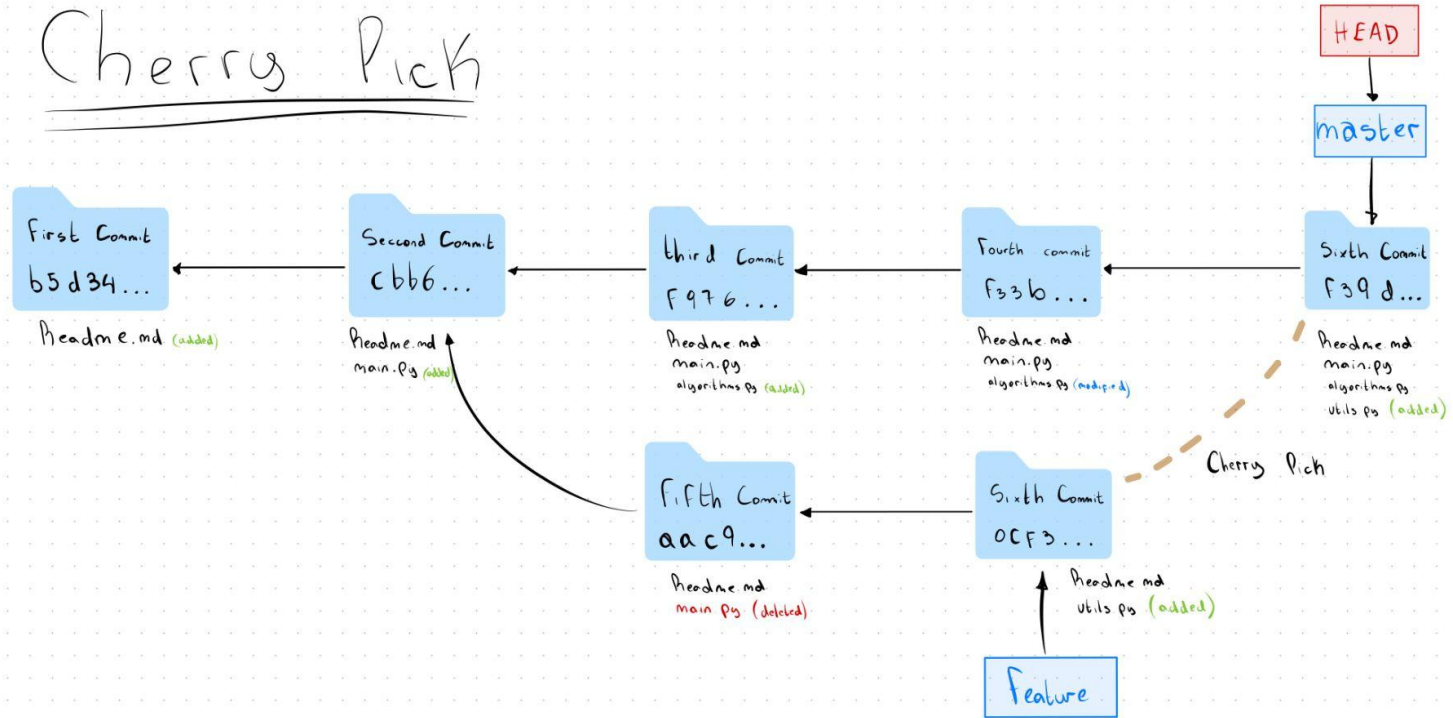
# Cherry Pick

# Cherry Pick





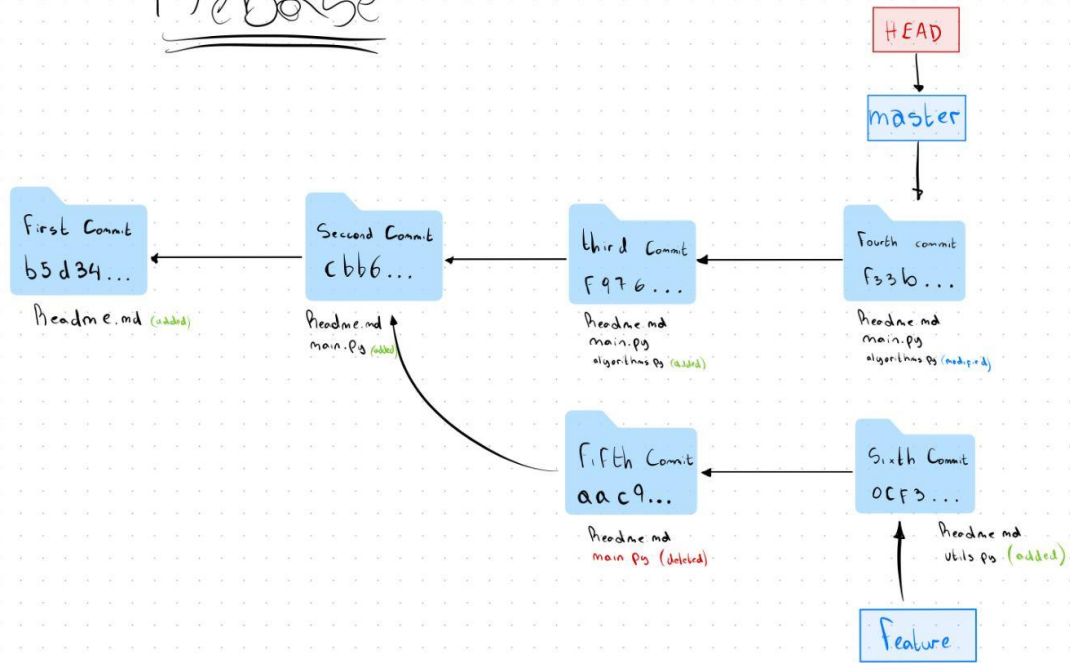
# Cherry Pick



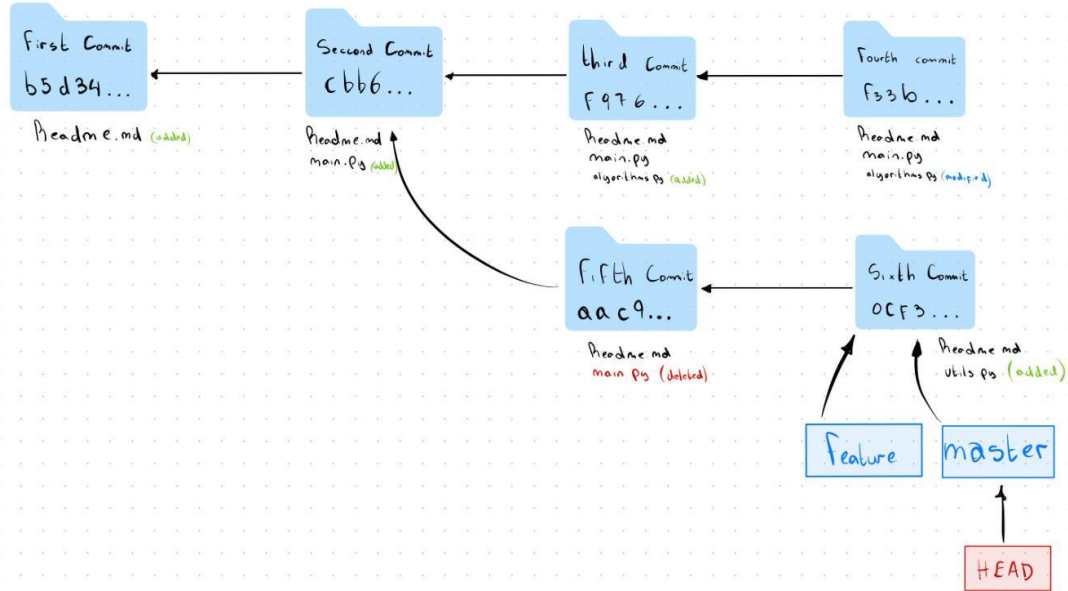


# Rebase

# Rebase

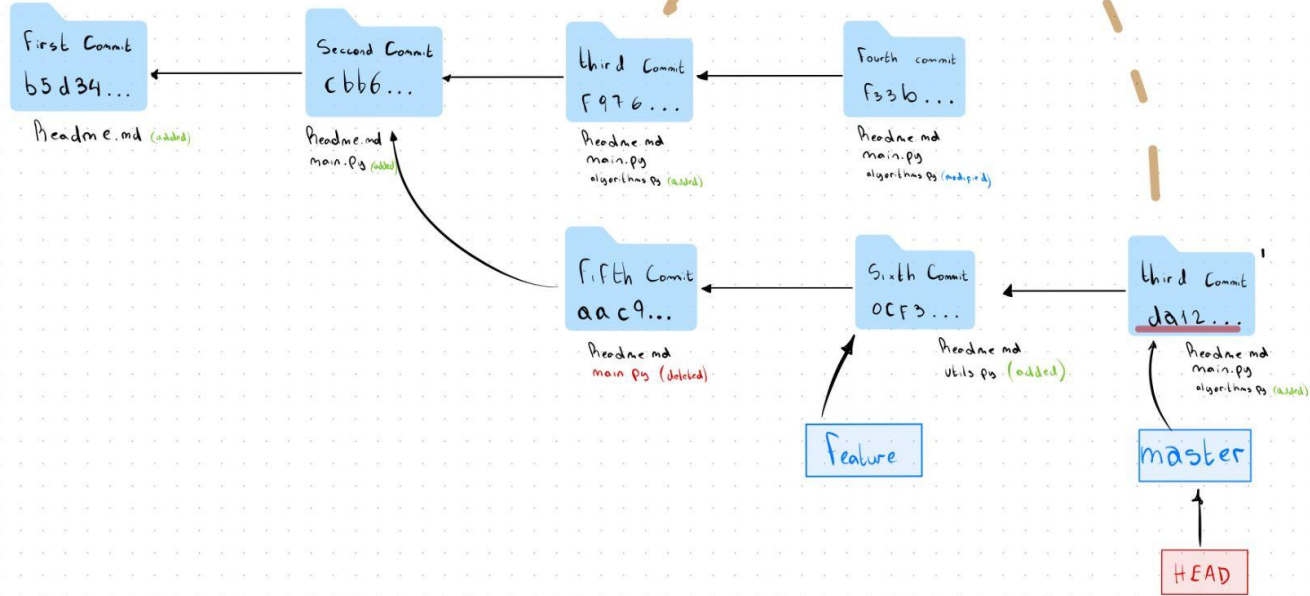


# Rebase



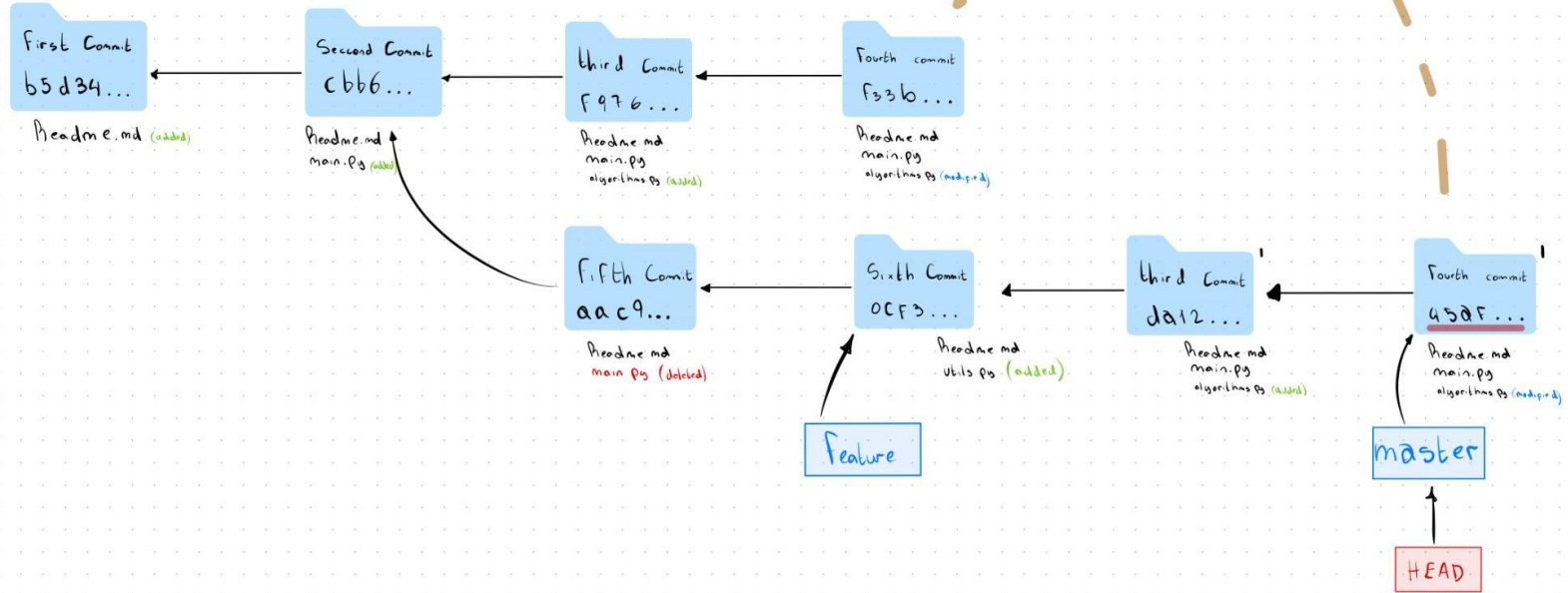
# Rebase

## Cherry Pick

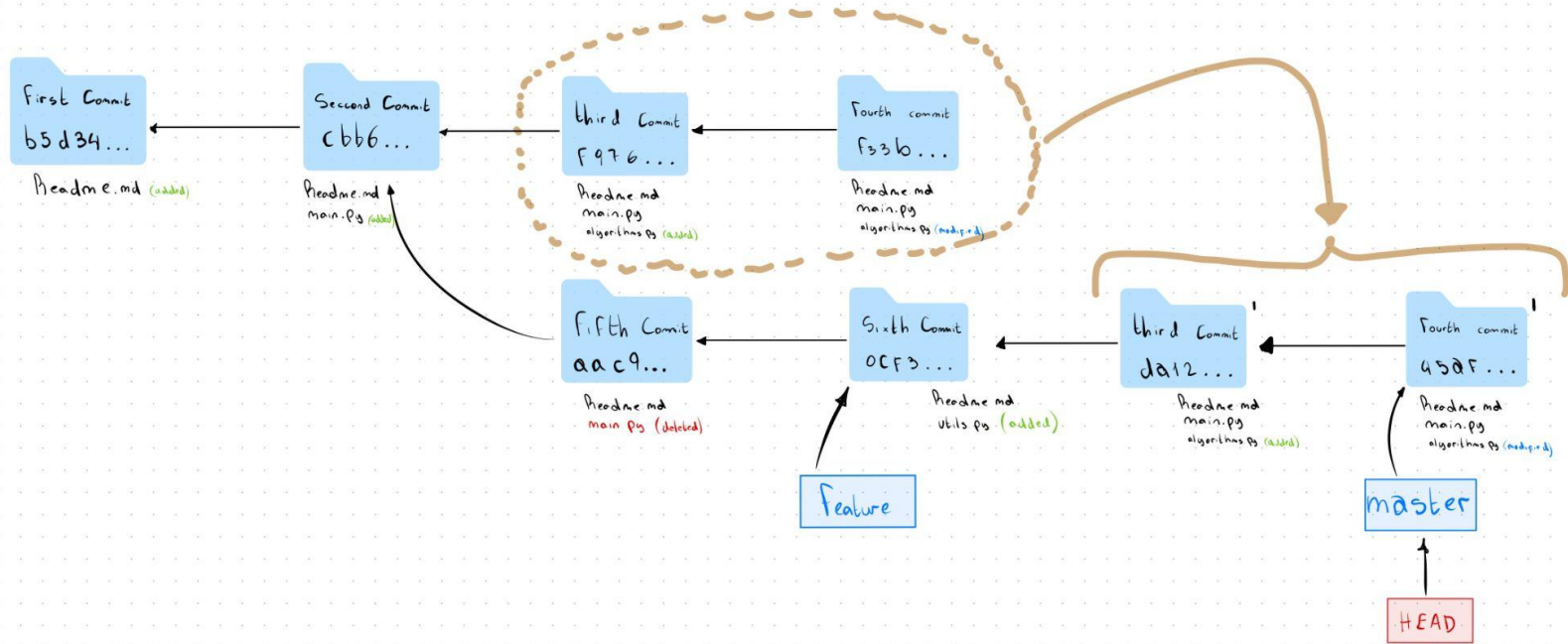


# Rebase

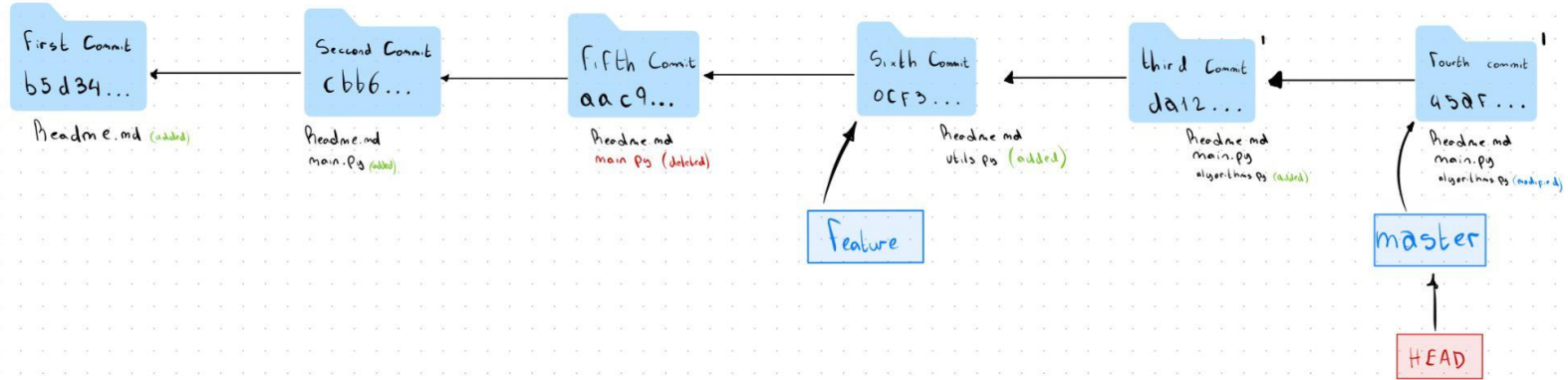
## Cherry Pick



# Rebase



# Rebase







# Nuestra implementación Git

