

COMANDOS GIT BASICS

PEDRO JAHIR HINOJOSA GARCÍA

```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  |   |   |   console.log(value)
                }}
              </ProductConsumer>
            </div>
          </div>
        </div>
      </React.Fragment>
```

QUE ES GIT?

GIT STATUS

GIT ADD .

GIT COMMIT -M “EXPLICACION”

GIT PUSH

```
4 # Prevent database truncation if the migration fails
5 abort("The Rails environment is running in production mode")
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create!(name: "Ruby on Rails")
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |sp|
16     sp.with.test_framework :rspec
17     sp.with.library :rails
18   end
19 end
20
21 # Add additional requires below this line to append them to all features
22 # spec/support/ and its subdirectories.
23 # run as spec files by default.
24 # in _spec.rb will both be required.
25 # run twice. It is recommended to do this in your spec support files.
26 # end with _spec.rb. You can configure the :run option by passing it
27 # another argument to require.
28 # require 'spec_helper'
29
30 # No results found for 'mongoid'
```

>git pull

> start explicacion.txt

Git es un gestor de versiones

Lo que significa que todo lo que hagas en una dirección específica, git lo va a recordar.

Puedes usarlo para regresar a otras versiones

Cuando quieras comparar o has cometido un error.

Es como el historial del buscador pero dentro de tu computadora.

Además tambien es popularmente usado para compartir tu código y sus versiones con la web através de GitHub.

>git help

- GIT INIT PARA CREAR UN NUEVO REPOSITORIO LOCAL.
- GIT ADD PARA MOVER ARCHIVOS AL STAGING AREA.
- GIT COMMIT -M "MENSAJE DEL COMMIT" PARA CREAR UNA INSTANTÁNEA DE TU PROYECTO.
- GIT BRANCH NOMBRE-DE-LA-RAMA PARA CREAR UNA NUEVA RAMA.
- GIT CHECKOUT NOMBRE-DE-LA-RAMA PARA MOVERTE ENTRE RAMAS.
- GIT MERGE NOMBRE-DE-LA-RAMA PARA COMBINAR UNA RAMA EN LA QUE ESTÁS CON OTRA.
- GIT CLONE URL-DEL-REPOSITORIO PARA COPIAR UN REPOSITORIO REMOTO.
- GIT PUSH PARA SUBIR TUS CAMBIOS A UN REPOSITORIO REMOTO.
- GIT PULL PARA OBTENER LA ÚLTIMA VERSIÓN DEL REPOSITORIO REMOTO.

ALGUNOS
COMANDOS
BÁSICOS SON: