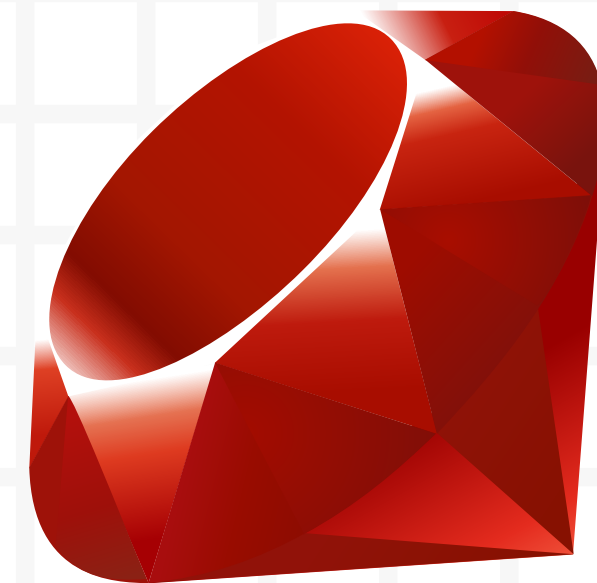



Sofia Rencoret, Cristobal Merino y Nicolas Fernandez

# AYUDANTiA

## MVC



Ruby



# INTRODUCCIÓN

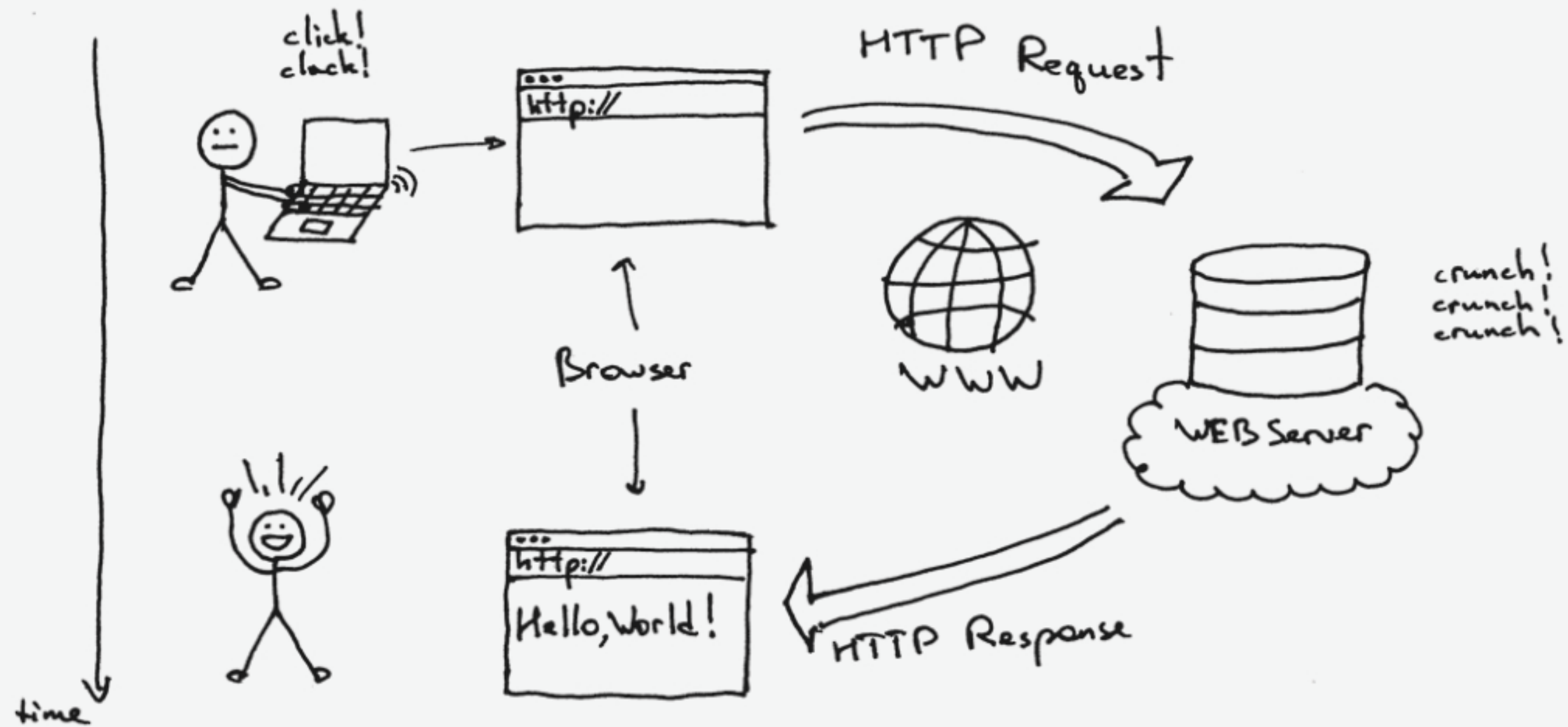
En esta ayudantía veremos la estructura MVC y GitFlow

**¿Qué debemos hacer si queremos ver,  
actualizar o añadir el contenido de  
nuestra página?**

Necesitamos trabajar con lógica adicional



# Protocollo HTTP



# MVC

Es un patrón de arquitectura del programa para poder separar el código en distintas responsabilidades (Models, Views, Controllers)

Bases de datos



**MODEL**

UPDATES

**VIEW**

MANIPULATES

**CONTROLLER**

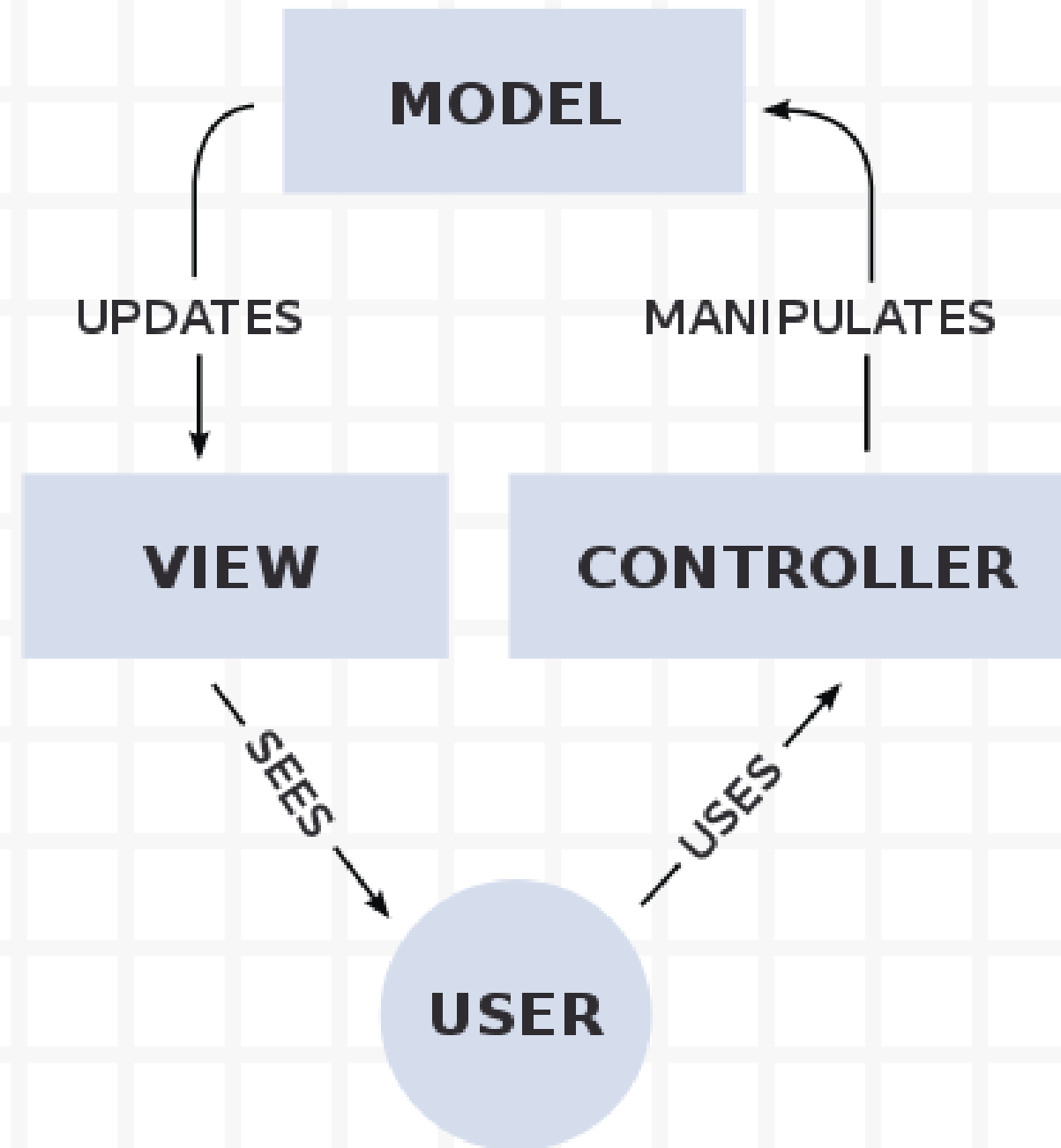
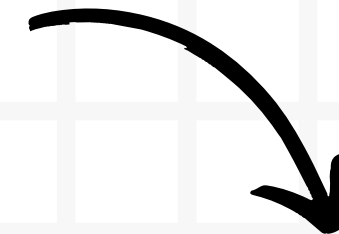
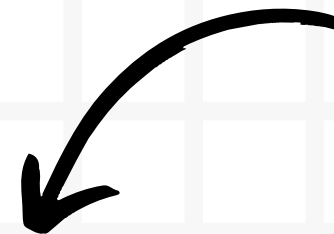
SEES

**USER**

USES

Frontend: lo que  
ve el usuario

Backend: lógica detrás  
de cada request



# ¿Dónde los encuentro en mi proyecto?

Creamos el proyecto

```
rails new Ayudantia-2024 --database=postgresql
```

Y la base de datos

```
rails db:create
```

▼ AYUDANTIA-2024

▼ app

> assets

> channels

> controllers ←

> helpers

> javascript

> jobs

> mailers

> models ←

> views ←

> bin

# MODELS

Son clases de Ruby que se usan para representar datos. Pueden interactuar con la base de datos de la aplicación a través de Active Record.

Se crean con:

```
/Ayudantia-2024$ rails generate model nombreModelo atributo1 atributo2
```

Por ejemplo:

```
c0bal@CocoPC:/mnt/c/Users/cobal/Desktop/PUC/2024-1/Ayudantia-2024$ rails generate model Alumno nombre:string major:string
invoke  active_record
create  db/migrate/20240419020421_create_alumnos.rb
create  app/models/alumno.rb
invoke  test_unit
create  test/models/alumno_test.rb
create  test/fixtures/alumnos.yml
```



# ACTUALIZAR LA BD

Se nos crea la siguiente migración:

```
20240419020421_create_alumnos.rb U X
db > migrate > 20240419020421_create_alumnos.rb
1  class CreateAlumnos < ActiveRecord::Migration[7.1]
2    def change
3      create_table :alumnos do |t|
4        t.string :nombre
5        t.string :major
6
7        t.timestamps
8      end
9    end
10  end
```

```
c0bal@CocoPC:/mnt/c/Users/cobal/Desktop/PUC/2024-1/Ayudantia-2024$ rails db:migrate
== 20240419020421 CreateAlumnos: migrating =====
-- create_table(:alumnos)
   -> 0.0103s
== 20240419020421 CreateAlumnos: migrated (0.0103s) =====
```

# QUE HACEMOS EN MODELS

Agregamos la referencia en caso de no haberlo hecho antes.

```
c0bal@CocoPC:/mnt/c/Users/cobal/Desktop/PUC/2024-1/Ayudantia-2024$ rails generate migration AddCursoIdToAlumnos curso:references
  invoke  active_record
  create  db/migrate/20240419025350_add_curso_id_to_alumnos.rb
c0bal@CocoPC:/mnt/c/Users/cobal/Desktop/PUC/2024-1/Ayudantia-2024$ rails db:migrate
== 20240419025350 AddCursoIdToAlumnos: migrating =====
-- add_reference(:alumnos, :curso, {:null=>false, :foreign_key=>true})
   -> 0.0112s
== 20240419025350 AddCursoIdToAlumnos: migrated (0.0112s) =====
```

```
alumno.rb U X
app > models > alumno.rb
1  class Alumno < ApplicationRecord
2    # Asociaciones
3    belongs_to :curso
4
5    # Validaciones
6    validates :nombre, presence: true
7    validates :major, presence: true
8  end
9
```

```
curso.rb U X
app > models > curso.rb
1  class Curso < ApplicationRecord
2    # Asociaciones
3    has_many :alumnos
4
5    # Validaciones
6    validates :nombre, presence: true
7    validates :sigla, presence: true
8  end
9
```

# CONTROLLERS

Es el código necesario para obtener los datos. Es un enlace entre los Views y los Models.

Nos permite actualizar o revisar los datos de ciertos objetos. Y también enviarlos a una view para actualizar nuestra página WEB

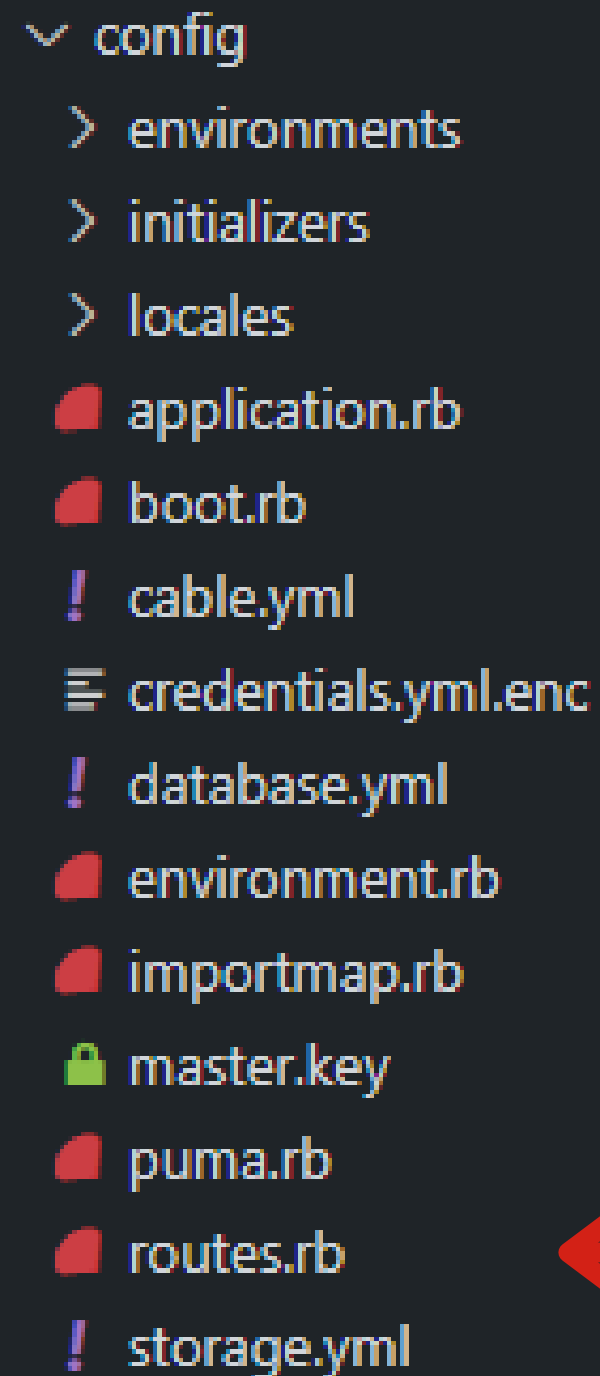
Creamos el controlador

```
/Ayudantia-2024$ rails generate controller Cursos
```

```
app > controllers > cursos_controller.rb
1  class CursosController < ApplicationController
2
3      # GET /cursos
4      def index
5          @cursos = Curso.all
6      end
7
8      # GET /cursos/1
9      def show
10         @curso = Curso.find(params[:id])
11     end
12
13     # GET /cursos/new
14     def new
15         @curso = Curso.new
16     end
17
18     # GET /cursos/1/edit
19     def edit
20         @curso = Curso.find(params[:id])
21     end
22
23     # POST /cursos
24     def create
25         @curso = Curso.new(curso_params)
26
27         if @curso.save
28             redirect_to @curso, notice: "El curso fue creado correctamente."
29         end
30     end
end
```

# ROUTES

El archivo **routes.rb** en una aplicación Rails es donde defines las rutas para tu aplicación web. Las rutas determinan cómo se mapean las URL a las acciones de los controladores en tu aplicación.



```
config
├── environments
├── initializers
├── locales
├── application.rb
├── boot.rb
├── cable.yml
├── credentials.yml.enc
├── database.yml
├── environment.rb
├── importmap.rb
├── master.key
├── puma.rb
├── routes.rb
└── storage.yml
```

routes.rb U X

config > routes.rb

```
1 Rails.application.routes.draw do
2   resources :cursos
3   resources :alumnos
4   root "cursos#index"
5 end
```

# ViEWS

Acá vemos la estética de la página usando HTML. Pueden recibir datos de los modelos.

**HTML**



**CSS**



# HTML

Código para poder mostrar los diferentes objetos de la página por medio de diversos comandos.

El diseño puede ser creado con CSS y/o bootstrap.

```
<h1> Título </h1>  
<h2> Título un poco más pequeño que h1 </h2>  
<p> Párrafo <p>  
<h1 class="titulo-modificado"> Título </h1> #archivo css que modifica h1.
```

```
.titulo-modificado {  
  color: ■ red;  
}
```



**Título**

**Título un poco más pequeño que h1**

Párrafo

**Título**

# HTML.ERB

HTML en esteroides que nos permitirá “inyectarle” código Ruby a los archivos HTML

```
<p> <strong>Nombre:</strong>
  <%= curso.nombre %>
</p>
<p> Sigla:
  <% curso.sigla %>
</p>
```

**Nombre:** Ingenieria de Software

Sigla:



No imprime la sigla

```
<p> <strong>Nombre:</strong>
  <%= curso.nombre %>
</p>
<p> Sigla:
  <%= curso.sigla %>
</p>
```

**Nombre:** Ingenieria de Software

Sigla: IIC2143



Imprime la sigla

# HTML.ERB

<> index.html.erb U X

app > views > cursos > <> index.html.erb

```
1 <p style="color: green"><%= notice %></p>
2
3 <h1>Cursos</h1>
4
5 <div id="cursos">
6   <% @cursos.each do |curso| %>
7     <%= render curso %>
8     <p>
9       <%= link_to "Show del curso", curso %>
10    </p>
11    <% end %>
12 </div>
13
14 <%= link_to "Nuevo curso", new_curso_path %>
```

<> \_curso.html.erb U X

app > views > cursos > <> \_curso.html.erb

```
1 <div>
2
3   <p> <strong>Nombre:</strong>
4     <%= curso.nombre %>
5   </p>
6   <p> Sigla:
7     <%= curso.sigla %>
8   </p>
9
10 </div>
```



# HTML.ERB

Podemos crear condiciones...

```
<% if condition1 %>
  <%= link_to "Acción 1", action1_path, class: "action-btn" %>
<% elsif condition2 %>
  <%= link_to "Acción 2", action2_path, class: "action-btn" %>
<% else %>
  <%= link_to "Acción por defecto", default_action_path, class: "action-btn" %>
<% end %>
```

# HTML.ERB + CSS

Agregamos esta línea para importar la plantilla de estilos

```
<%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
```

```
app > views > cursos > <> show.html.erb
```

```
1 <div class="container">
2   <h1>Curso</h1>
3
4   <div id="curso">
5     <%= render @curso %>
6   </div>
7
8   <div class="actions">
9     <%= link_to "Editar curso", edit_curso_path(@curso), class: "edit-btn" %>
10    <%= link_to "Volver a cursos", cursos_path, class: "back-btn" %>
11    <%= button_to "Eliminar curso", @curso, method: :delete, class: "destroy-btn" %>
12  </div>
13
14  <% if notice.present? %>
15    <p class="notice"><%= notice %></p>
16  <% end %>
17 </div>
```

## Curso

**Nombre:** Ingenieria de Software

**Sigla:** IIC2143

Editar curso

Volver a cursos

Eliminar curso



# GiTFLOW

Es un flujo de trabajo para mantener un orden cuando subimos avances al repositorio y así que distintas personas puedan modificar el código.

# BRANCHES

Rama principal de su repositorio: Main  
Pueden crear las que ustedes quieran.  
Se recomienda tener una por cada feature.

Comandos clave:

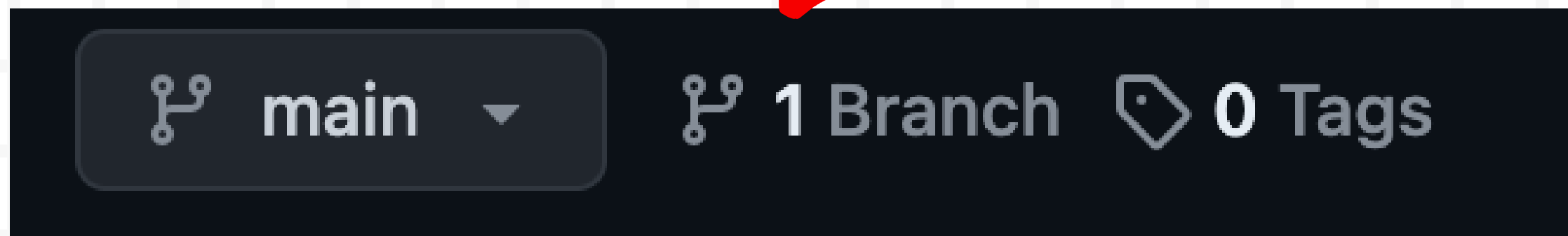
- `git checkout <nombre_de_la_rama>`
- `git branch`

```
● sofiaencoret@sofias-mbp 2023-2-grupo-46 % git branch
  comentario
  development
* main
  render_usuario
○ sofiaencoret@sofias-mbp 2023-2-grupo-46 %
```

```
sofiaencoret@sofias-mbp 2023-2-grupo-46 % git checkout development

Switched to branch 'development'
Your branch is behind 'origin/development' by 10 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

# BRANCHES



Branches New branch

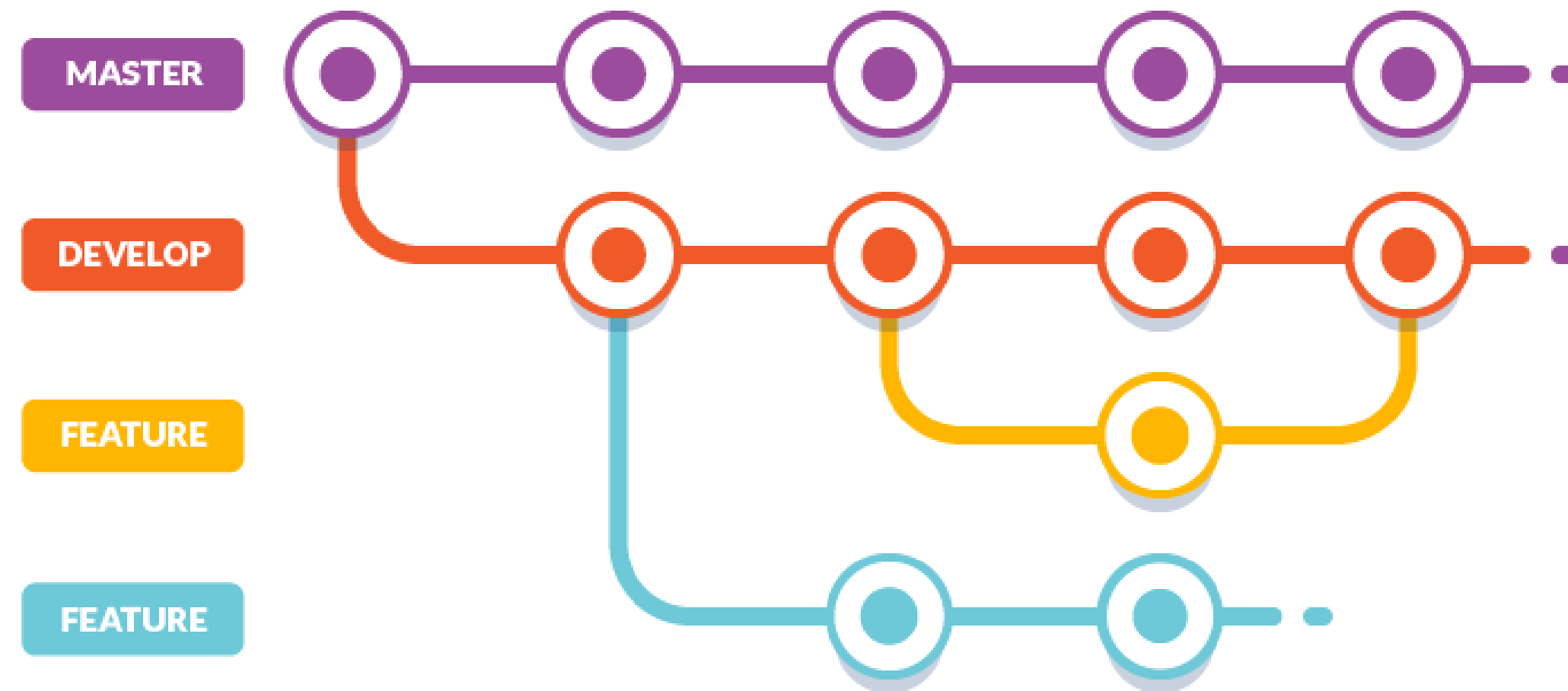
Overview Yours Active Stale All

Search branches...


Default

Branch	Updated	Check status	Behind Ahead	Pull request
main			Default	

# BRANCHES



# MERGE

 **development** had recent pushes 18 seconds ago

Compare & pull request



Add a title

para-ayudantia


Add a description


Write

Preview

H B I ≡ <> 🔗 | ☰ ☷ ⚙ | 📎 @ 💬 ↩ 🗑

Add your description here...

 Markdown is supported

 Paste, drop, or click to add files

Create pull request





Srencoret commented 1 minute ago



No description provided.



para-ayudantia

4e417e5

Add more commits by pushing to the **development** branch on [IIC2143/2023-2-grupo-46](#).



**This branch has no conflicts with the base branch**

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



Add a comment

Write

Preview



Add your comment here...

Markdown is supported

Paste, drop, or click to add files



Close pull request

Comment



Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).



**Pull request successfully merged and closed**

You're all set—the **development** branch can be safely deleted.

Delete branch





**A Programar**