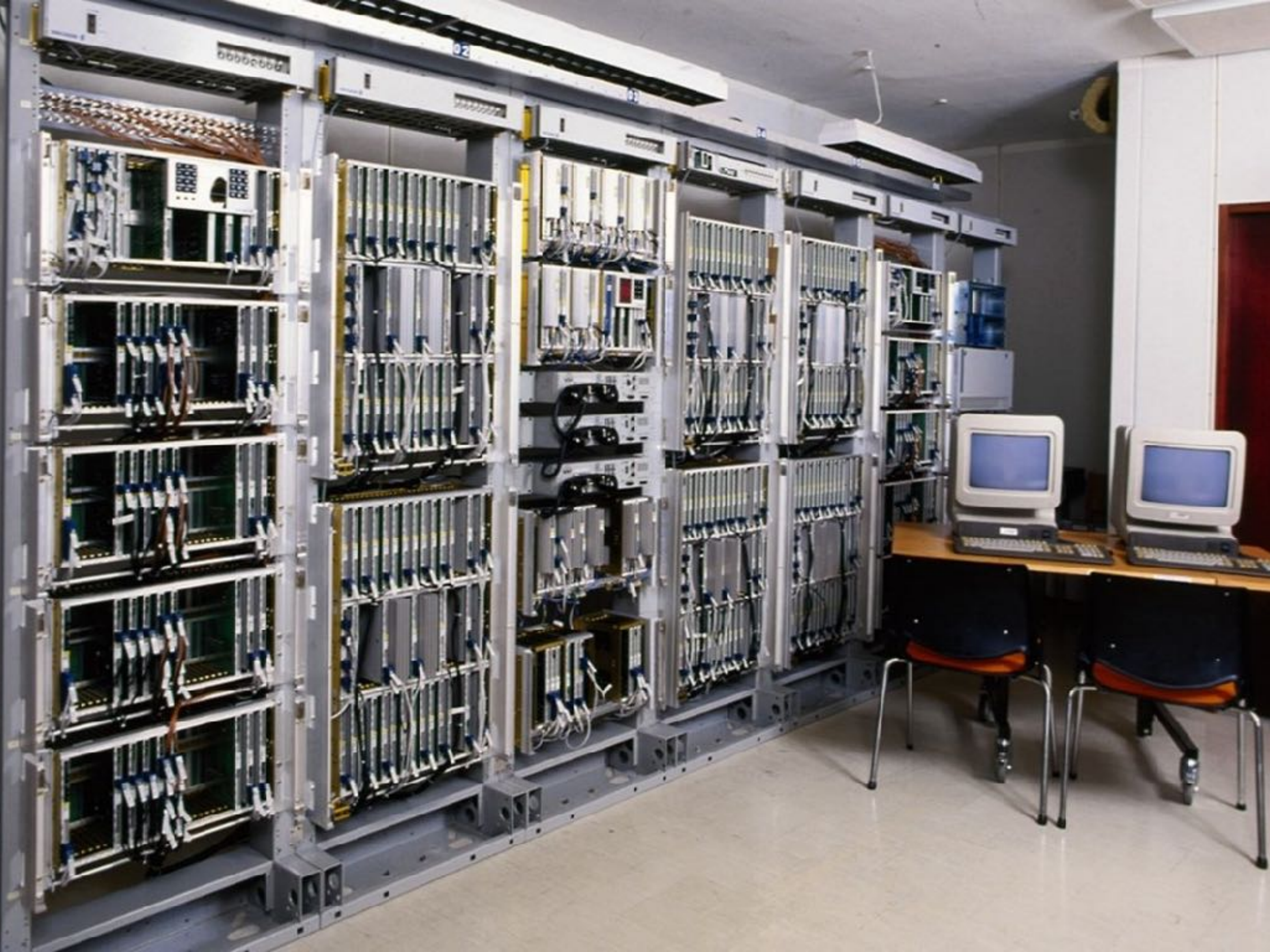
Elixir Meetup

La Plata

Elixir, Phoenix y los desafíos de la nueva web



```elixir
defmodule Math do
  # this is a comment
  def sum(a, b) do
    a + b
  end
  def duplicate(a) do
    2 * a
  end
end

Math.sum(1, 2) #=> 3
Math.duplicate(3) #=> 6
```

Los objetos en Ruby acoplan:

* representación de datos
* lógica de programa
* organización de código
* manejo de estado
* polimorfismo (por herencia)

```ruby
# Ruby
"HOLA".downcase
#=> "hola"
```

```elixir
# Elixir
String.downcase("HOLA")
#=> "hola"
```

Ruby

```ruby
"HOLA".equal?("HOLA")
```

```ruby
# Ruby

a = "HOLA"
b = "HOLA"
```

```ruby
a.object_id #=> 70227497266080
b.object_id #=> 70227497236260
```

```elixir
# Elixir

"HOLA" == "HOLA"
#=> true
```

```elixir
# Elixir

[1, 2, 3] == [1, 2, 3]
#=> true
```

```ruby
# Ruby

a = "HOLA"
a.downcase! #=> "nil"
a #=> "hola"
```

```elixir
# Elixir

a = "HOLA"
String.downcase(a) #=> "hola"
a #=> "HOLA"
```

```
Value |> f() |> g() |> h()
```

```
spawn(fn -> 1 + 2 end)
#=> #PID<0.111.0>
```

GenStage

Task

Agent

GenServer

Supervisor


TM

```ruby
class CreateArticles < ActiveRecord::Migration[6.0]
  def change
    create_table :articles do |t|
      t.string :name
      t.text :description
      t.timestamps
    end
  end
end
```

```elixir
defmodule MyWeb.Repo.Migrations.CreateArticle do
  use Ecto.Migration

  def change do
    create table(:articles) do
      add :title, :string
      add :description, :string
      timestamps
    end
  end
end
```

```ruby
resources :articles do
  resources :comments, only: [:comment]
end

post "login", to: "sessions#login"

root to: "pages#index"
```

```ruby
resources "/articles", ArticleController do
  resources "/comments", CommentController, only: [:show]
end

post "/login", SessionController, :login

get "/", PageController, :index
```

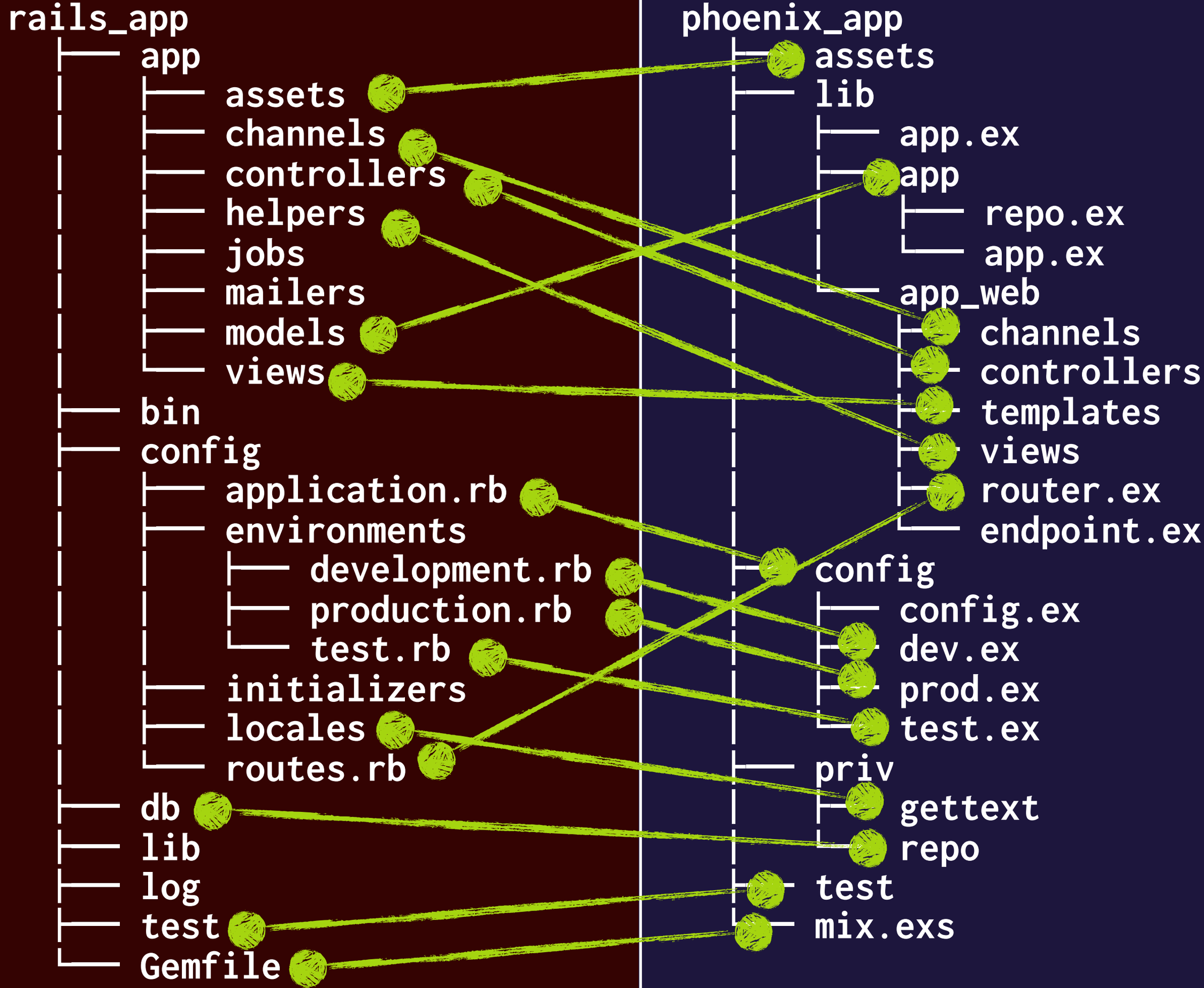
```ruby
class ArticlesController < ApplicationController
  def show
    @article = Article.find(params[:id])
    # render "show.html"
  end
end
```

```elixir
defmodule MyWeb.ArticleController do
  use MyWeb.Web, :controller

  def show(conn, %{"id" => id}) do
    article = Blog.get_article!(id)
    render(conn, "show.html", article: article)
  end
end
```

```
<h1><%= @article.title %></h1>
<p><%= @article.description %></p>

<%= for comment <- @article.comments do %>
  <p class="comment">
    <%= comment.content %>
  </p>
<% end %>

<%= link "Editar",
    to: article_path(@conn, :edit, @article) %>

<%= link "Volver", to: article_path(@conn, :index) %>
```

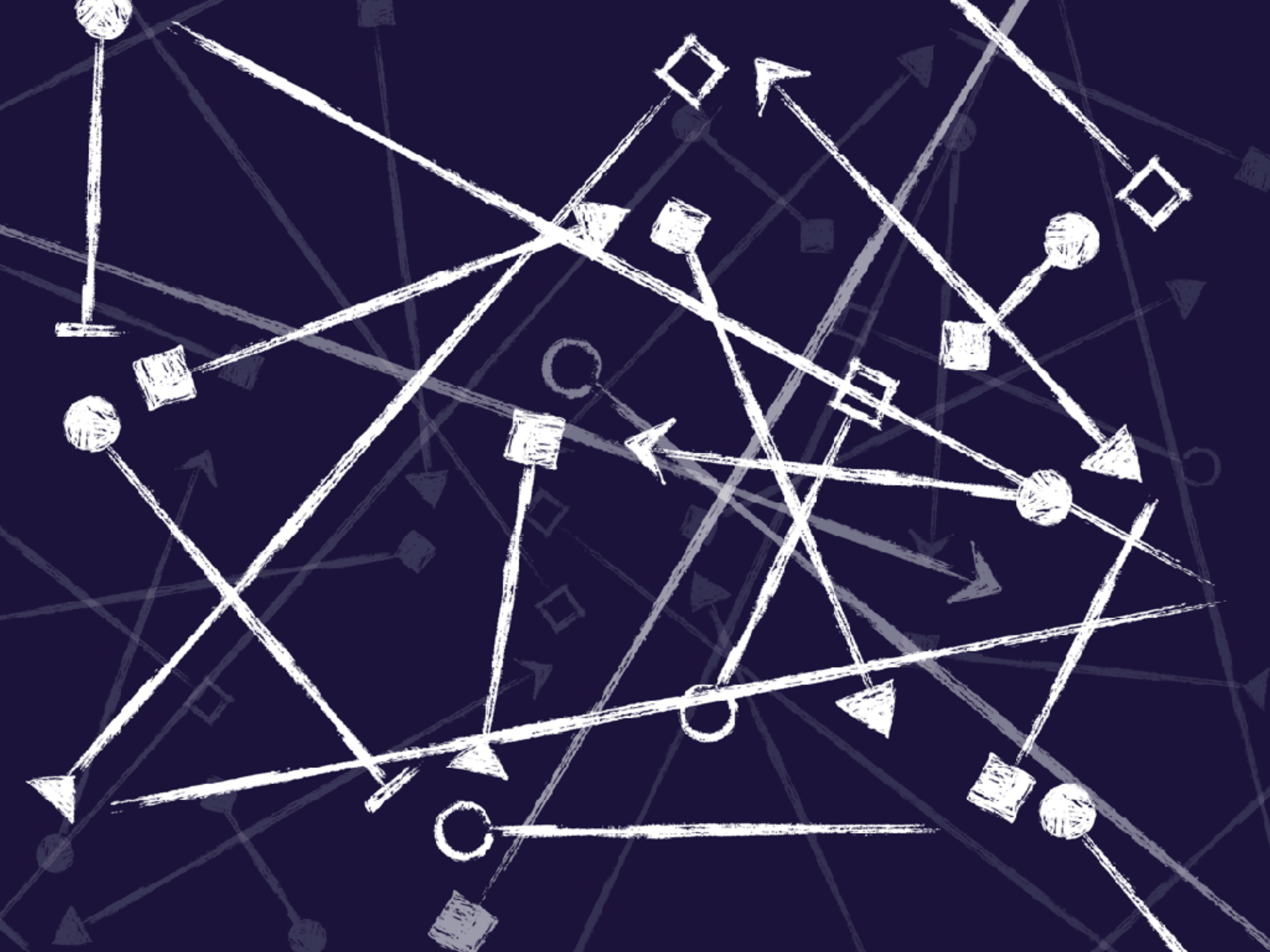
```
rails_app
├── app
│   ├── assets
│   ├── channels
│   ├── controllers
│   ├── helpers
│   ├── jobs
│   ├── mailers
│   ├── models
│   └── views
├── bin
├── config
│   ├── application.rb
│   ├── environments
│   │   ├── development.rb
│   │   ├── production.rb
│   │   └── test.rb
│   ├── initializers
│   ├── locales
│   └── routes.rb
├── db
├── lib
├── log
├── test
└── Gemfile
```

```
phoenix_app
├── assets
├── lib
│   ├── app.ex
│   ├── app
│   │   ├── repo.ex
│   │   └── app.ex
│   └── app_web
│       ├── channels
│       ├── controllers
│       ├── templates
│       ├── views
│       ├── router.ex
│       └── endpoint.ex
├── config
│   ├── config.ex
│   ├── dev.ex
│   ├── prod.ex
│   └── test.ex
├── priv
│   ├── gettext
│   └── repo
├── test
└── mix.exs
```


rails_app
app
assets
channels
controllers
helpers
jobs
mailers
models
views
bin
config
application.rb
environments
development.rb
production.rb
test.rb
initializers
locales
routes.rb
db
lib
log
test
Gemfile
phoenix_app
assets
lib
app.ex
app
repo.ex
app.ex
app_web
channels
controllers
templates
views
router.ex
endpoint.ex
config
config.ex
dev.ex
prod.ex
test.ex
priv
gettext
repo
test
mix.exs

Channels

Sent 200 in 517µs



Gracias

github.com/nicanor