

**ENVIOS DE TU NEGOCIO A  
TODO MÉXICO CON ELIXIR?**

**NICHOLAS RAPHAEL FARIAS JIMENEZ**

**@OOHWHAT**

eiya!

# ADOPTION

- DIFFERENT ON EACH ORGANIZATION

# ADOPTION

- > DIFFERENT ON EACH ORGANIZATION
  - > BE THE SALESPERSON

# ADOPTION

- > DIFFERENT ON EACH ORGANIZATION
  - > BE THE SALESPERSON
- > THE FIRST PROTOTYPE IS IMPORTANT



## Iniciar Sesión

[¿Olvidaste tu contraseña?](#)

Iniciar Sesión

# FIRST PROTOTYPE

- TANGIBLE BUSINESS VALUE
- CENTRAL ENOUGH TO BE SEEN AND BIG ENOUGH TO MATTER
  - SMALL ENOUGH TO PERMIT FAILURE

# WHY ELIXIR

## > PERFORMANCE



# WHY ELIXIR

- > PERFORMANCE
- > TALENT AQUISITION

# WHY ELIXIR

- > PERFORMANCE
- > TALENT AQUISITION
- > LANGUAGE FEATURES

# PATTERN MATCHING

```
{a, b, c} = {:hello, "world", %{'hello': 'world'}}
```

```
a #=> :hello
```

```
b #=> "world"
```

```
c #=> %{'hello': 'world'}
```

# PATTERN MATCHING

```
%Delivery{position: [lat, lng], cost: cost}  
= Delivery.get('ASD24X')
```

```
lat #=> 19.434446
```

```
lng #=> -99.180885
```

```
cost #=> 4900
```

# GUARDS

```
def notify(%Delivery{weight: weight}) when weight >= 10 do
  IO.puts("Its heavy :/ ")
end
```

```
def notify(%Delivery{weight: weight}) do
  IO.puts("Less than 10kg ")
end
```

```
notify(Delivery.get('ASD24X'))
```

```
#=> "Less than 10kg
```

```
#=> "Its heavy :/ "
```

# WHY ELIXIR – SIMPLIFY BY ENCAPSULATION

```
def checknums(min, max):  
    for num in range(min, max):  
        if num % 2 == 0:  
            print('Es multiplo de 2')  
        elif num % 3 == 0:  
            print('Es multiplo de 3')  
        elif num % 5 == 0:  
            print('Es multiplo de 5')  
        else:  
            print(num)
```

# WHY ELIXIR – SIMPLIFY BY ENCAPSULATION

```
defmodule CheckNumbers do
  def enjoy(min, max), do: Enum.each(min..max, &check/1)

  def check(num) when rem(num, 2) == 0, do: IO.puts("multiplo de 2")
  def check(num) when rem(num, 3) == 0, do: IO.puts("multiplo de 3")
  def check(num) when rem(num, 5) == 0, do: IO.puts("multiplo de 5")
  def check(num), do: IO.puts(":D")
end
```

# WHY ELIXIR – COMPLEX TO SIMPLE

```
if request.method == 'POST':  
    ...  
    if Validator.is_valid_csv(filtered_csvs):  
        ...  
        if batch_status:  
            ...  
            if deliveries_status:  
                ...  
                if geolocation_status:  
                    ...
```



# WHY ELIXIR – COMPLEX TO SIMPLE

```
with {:ok, user} <- Validator.validate(csv),  
     {:ok, batch_id} <- Batch.create_batch(data),  
     {:ok, delivery_id} <- Delivery.create(batch_id, info)  
     {:ok, map} <- Map.add(delivery_id, location) do  
    ...  
else  
  {:error, error} ->  
    ...  
end
```

# WHY ELIXIR IN EIYA 2.0

- > NO REWRITES
- > MAKES SIMPLE EASY
- > IT MAKES SENSE

# TOOLS

# TOOLING Y LA COMUNIDAD DE ELIXIR

WEB FRAMEWORK

---

PHOENIX

PACKAGE MANAGER

---

HEX

DEPLOYMENT

EDELIVER

DISTILLERY

---

CRONTAB

POISON

EX\_DOC

CREDO

---

QUANTUM

TIMEX

BAMBOO

ROLLBAX

# CODE CONSISTENCY IS A TEAM EFFORT

## BAD HABITS DIE HARD

- > CODING STANDARDS
  - > FORMAT
  - > CREDO
- > DOCUMENTATION

# FORMATTER

\$ mix format

```
defmodule HelloWorld do
  def greet( first, last ) do
    name = first<>" "<>last
    IO.puts "Hello #{ name }"
  end
end
```

# FORMATTER

mix format

```
defmodule HelloWorld do
  def greet(first, last) do
    name = first <> " " <> last
    IO.puts("Hello #{ name }")
  end
end
```

# CREDO

CREDO IS A STATIC CODE ANALYSIS TOOL FOR THE ELIXIR LANGUAGE WITH A FOCUS ON **TEACHING**  
AND **CODE CONSISTENCY**.



# CREDO

```
defp deps do
  [
    {:credo, "~> 1.0.0", only: [:dev, :test], runtime: false}
  ]
end
```

```
$ mix deps.get
```

```
$ mix credo
```

# CREDO

**\$ MIX CREDO LIB/PHOENIX/CHANNEL.EX:26**

```
| Phoenix.Channel
|
|   ...
|   [R] Category: refactor
|   ↗ Priority: medium
|
|   ...
|   CODE IN QUESTION
|
|   ...
|   WHY IT MATTERS
|
|   ...
```

# CREDO

- > CONSISTENCY
- > READABILITY
- > REFACTORING OPPORTUNITIES
  - > SOFTWARE DESIGN
  - > WARNINGS

# DOCUMENTATION

```
defmodule MyApp.Hello do
  @moduledoc """
  This is the Hello module.
  """

  @doc """
  This is the world function.
  """

  def world(name) do
    IO.puts "hello #{name}"
  end
end
```

# WHAT WE GOT

- > PUSH ON THE RIGHT PATH
  - > AWESOME TOOLS
  - > EASIER GROWTH

**MUCHAS GRACIAS!**

# Q & A

EIYA – [HTTPS://EIYA.MX/](https://eiya.mx/) – [HTTPS://TWITTER.COM/EIYAMX](https://twitter.com/eiyamx)

NICO – [HTTPS://GITHUB.COM/RALPHMARKS](https://github.com/RalphMarks) – [HTTPS://TWITTER.COM/OOHWHAT](https://twitter.com/ooHwHAT)

