

Curso Elixir

Sección 1

¿Que es elixir?

Elixir es un lenguaje funcional moderno construido sobre Erlang VM. Es totalmente compatible con Erlang, pero presenta una sintaxis más estándar y muchas más funciones.

Características de la plataforma

- Escalabilidad: Todo el código de Elixir se ejecuta dentro de hilos de ejecución ligeros (llamados procesos) que están aislados e intercambian información a través de mensajes

```
current_process = self()

# Spawn an Elixir process (not an operating system one!)
spawn_link(fn ->
  send(current_process, {:msg, "hello world"})
end)

# Block until the message is received
receive do
  {:msg, contents} -> IO.puts(contents)
end
```

- Tolerancia a fallos:

Es inevitable tener fallos en el software que se ejecuta en producción, aún más cuanto este interactúa con redes, sistemas de archivos y otros recursos de terceros. Para hacer frente a las fallas, Elixir proporciona supervisores que describen cómo reiniciar partes de su sistema cuando las cosas salen mal, volviendo a un estado inicial conocido que garantiza que funcione.

Características del lenguaje

- Programación funcional:

La programación funcional promueve un estilo de codificación que ayuda a los desarrolladores a escribir código que sea breve, conciso y fácil de mantener. Por ejemplo, el patrón de coincidencia permite a los desarrolladores desestructurar fácilmente los datos y acceder a sus contenidos.

Todas las funciones en elixir regresan un valor.

```
data = %{numero: 5, letra: "cinco"}
%{letra: valor_letra} = data
valor_letra
```

- Extensibilidad:

Elixir ha sido diseñado para ser extensible, permitiendo a los desarrolladores extender naturalmente el lenguaje a dominios particulares, con el fin de aumentar su productividad.

```
ExUnit.start()
defmodule MathTest do
  use ExUnit.Case, async: true
  test "can add two numbers" do
    assert 1 + 1 == 2
  end
end
```

Esas características se crean utilizando macros de Elixir, lo que permite agregar nuevas construcciones como si fueran parte del lenguaje en sí.

Características de herramientas

- Un ecosistema en crecimiento

Elixir viene con un gran conjunto de herramientas para facilitar el desarrollo. Mix es una herramienta de compilación que permite crear fácilmente proyectos, administrar tareas, ejecutar pruebas y más:

```
$ mix new my_app
$ cd my_app
$ mix test
```

- Desarrollo interactivo:

Herramientas como iex son capaces de aprovechar muchos aspectos del lenguaje y la plataforma para proporcionar herramientas de depuración, autocompletado, recarga de código y documentación bien formateada:

```
$ iex
iex> h String.trim
iex> i "Hello, World"
iex> break! String.trim/1
iex> recompile
```

- Compatible con Erlang:

Elixir se ejecuta en Erlang VM, lo que brinda a los desarrolladores acceso completo al ecosistema de Erlang, utilizado por compañías como Heroku, WhatsApp, Klarna y muchas más para crear aplicaciones distribuidas y tolerantes a fallas. Un programador de Elixir puede invocar cualquier función de Erlang sin costo de tiempo de ejecución:

```
iex> :crypto.hash(:md5, "Using crypto from Erlang OTP")
```

Actividad 1

Ejecutar los comandos vistos en esta sección.