

+ Introducción a Elixir sesión 3.

Santiago Posada

SofkaU

#ElDesafíoEsContigo



Pattern matching:

```
iex(4)> person = {"John", 25}
person = {"John", 25}
{"John", 25}
iex(5)> {name, age} = person
{name, age} = person
{"John", 25}
iex(6)> {name, age} = {"John", 25}
{name, age} = {"John", 25}
{"John", 25}
iex(7)> name
name
"John"
iex(8)> age
age
25
```

```
iex(11)> {date, time} = :calendar.local_time()
{date, time} = :calendar.local_time()
{{2019, 4, 29}, {19, 8, 35}}
iex(12)> date
date
{2019, 4, 29}
iex(13)> time
time
{19, 8, 35}
iex(14)> {y, m, d} = date
{y, m, d} = date
{2019, 4, 29}
iex(15)> {h, min, sec} = time
{h, min, sec} = time
{19, 8, 35}
iex(16)> |
```



```
iex(16)> {a, b} = "Won't match"
```

```
{a, b} = "Won't match"
```

```
** (MatchError) no match of right hand side value: "Won't match"  
(stdlib) erl_eval.erl:453: :erl_eval.expr/5  
(iex) lib/iex/evaluator.ex:257: IEx.Evaluator.handle_eval/5  
(iex) lib/iex/evaluator.ex:237: IEx.Evaluator.do_eval/3  
(iex) lib/iex/evaluator.ex:215: IEx.Evaluator.eval/3  
(iex) lib/iex/evaluator.ex:103: IEx.Evaluator.loop/1  
(iex) lib/iex/evaluator.ex:27: IEx.Evaluator.init/4
```



```
iex(17)> 1 = 1
```

```
1 = 1
```

```
1
```

```
iex(18)> rectangle = {:rectangle, 10, 20}
```

```
rectangle = {:rectangle, 10, 20}
```

```
{:rectangle, 10, 20}
```

```
iex(19)> {:rectangle, width, height} = rectangle
```

```
{:rectangle, width, height} = rectangle
```

```
{:rectangle, 10, 20}
```

```
iex(20)> {:ok, result} |
```




```
iex(20)> {:ok, contents} = File.read("example.txt")
{:ok, contents} = File.read("example.txt")
** (MatchError) no match of right hand side value: {:error, :enoent}
(stdlib) erl_eval.erl:453: :erl_eval.expr/5
(iex) lib/iex/evaluator.ex:257: IEx.Evaluator.handle_eval/5
(iex) lib/iex/evaluator.ex:237: IEx.Evaluator.do_eval/3
(iex) lib/iex/evaluator.ex:215: IEx.Evaluator.eval/3
(iex) lib/iex/evaluator.ex:103: IEx.Evaluator.loop/1
(iex) lib/iex/evaluator.ex:27: IEx.Evaluator.init/4
```



```
iex(20)> {:ok, contents} = File.read("example.txt")
{:ok, contents} = File.read("example.txt")
** (MatchError) no match of right hand side value: {:error, :enoent}
(stdlib) erl_eval.erl:453: :erl_eval.expr/5
(iex) lib/iex/evaluator.ex:257: IEx.Evaluator.handle_eval/5
(iex) lib/iex/evaluator.ex:237: IEx.Evaluator.do_eval/3
(iex) lib/iex/evaluator.ex:215: IEx.Evaluator.eval/3
(iex) lib/iex/evaluator.ex:103: IEx.Evaluator.loop/1
(iex) lib/iex/evaluator.ex:27: IEx.Evaluator.init/4
iex(20)> File.read("example.txt")
File.read("example.txt")
{:error, :enoent}
iex(21)> {:error, reason} = File.read("example.txt")
{:error, reason} = File.read("example.txt")
{:error, :enoent}
iex(22)> |
```


Variables anónimas:

```
iex(22)> {_, time} = :calendar.local_time()
{_, time} = :calendar.local_time()
{{2019, 4, 29}, {19, 18, 35}}
iex(23)> time
time
{19, 18, 35}
iex(24)> {_date, time} = :calendar.local_time()
```



```
iex(26)> {_, {hour, _, _}} = :calendar.local_time()  
{_, {hour, _, _}} = :calendar.local_time()  
{{2019, 4, 29}, {19, 21, 14}}
```

```
iex(29)> x = 10
```

```
x = 10
```

```
10
```

```
iex(30)> {^x, _} = {10, "hello"}
```

```
{^x, _} = {10, "hello"}
```

```
{10, "hello"}
```

```
iex(31)> {^x, _} = {11, "hello"}
```

```
{^x, _} = {11, "hello"}
```

```
** (MatchError) no match of right hand side value: {11, "hello"}
```

```
(stdlib) erl_eval.erl:453: :erl_eval.expr/5
```

```
(iex) lib/iex/evaluator.ex:257: IEx.Evaluator.handle_eval/5
```

```
(iex) lib/iex/evaluator.ex:237: IEx.Evaluator.do_eval/3
```

```
(iex) lib/iex/evaluator.ex:215: IEx.Evaluator.eval/3
```

```
(iex) lib/iex/evaluator.ex:103: IEx.Evaluator.loop/1
```

```
(iex) lib/iex/evaluator.ex:27: IEx.Evaluator.init/4
```

```
iex(31)> {x, _} = {11, "hello"}  
{x, _} = {11, "hello"}  
{11, "hello"}
```




```
iex(33)> [f, s, t] = [1, 2, 3]
[f, s, t] = [1, 2, 3]
[1, 2, 3]
iex(34)> f
f
1
iex(35)> s
s
2
iex(36)> t
t
3
```

Cons operator: |

```
iex(38)> [head | tail] = [1, 2, 3, 4, 5]
[head | tail] = [1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
iex(39)> head
head
1
iex(40)> tail
tail
[2, 3, 4, 5]
iex(41)> [head | x_tail] = [1, 2, 3, 4, 5]
```

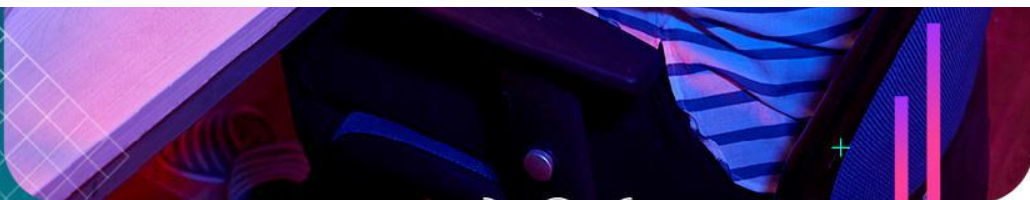
```
iex(44)> [min | _] = Enum.sort([3, 5, 2, 1])  
[min | _] = Enum.sort([3, 5, 2, 1])  
[1, 2, 3, 5]
```



```
iex(46)> %{name: name, age: age} = %{ age: 25, name: "John"}
%{name: name, age: age} = %{ age: 25, name: "John"}
%{age: 25, name: "John"}
iex(47)> age
age
25
iex(48)> name
name
"John"
iex(49)> %{age: age} = %{name: "John", age: 25}
%{age: age} = %{name: "John", age: 25}
%{age: 25, name: "John"}
```



```
iex(50)> %{ages: age} = %{name: "John", age: 25}
%{ages: age} = %{name: "John", age: 25}
** (MatchError) no match of right hand side value: %{age: 25, name:
"John"}
(stdlib) erl_eval.erl:453: :erl_eval.expr/5
(iex) lib/iex/evaluator.ex:257: IEx.Evaluator.handle_eval/5
(iex) lib/iex/evaluator.ex:237: IEx.Evaluator.do_eval/3
(iex) lib/iex/evaluator.ex:215: IEx.Evaluator.eval/3
(iex) lib/iex/evaluator.ex:103: IEx.Evaluator.loop/1
(iex) lib/iex/evaluator.ex:27: IEx.Evaluator.init/4
```



```
iex(54)> [_ , {name, _}, _] = [{"John", 17}, {"Jane", 26}, {"Adam", 30}]
name
"Jane"
```



```
iex(56)> a = (b = 1 + 3)
```

```
a = (b = 1 + 3)
```

```
4
```

```
iex(57)> a
```

```
a
```

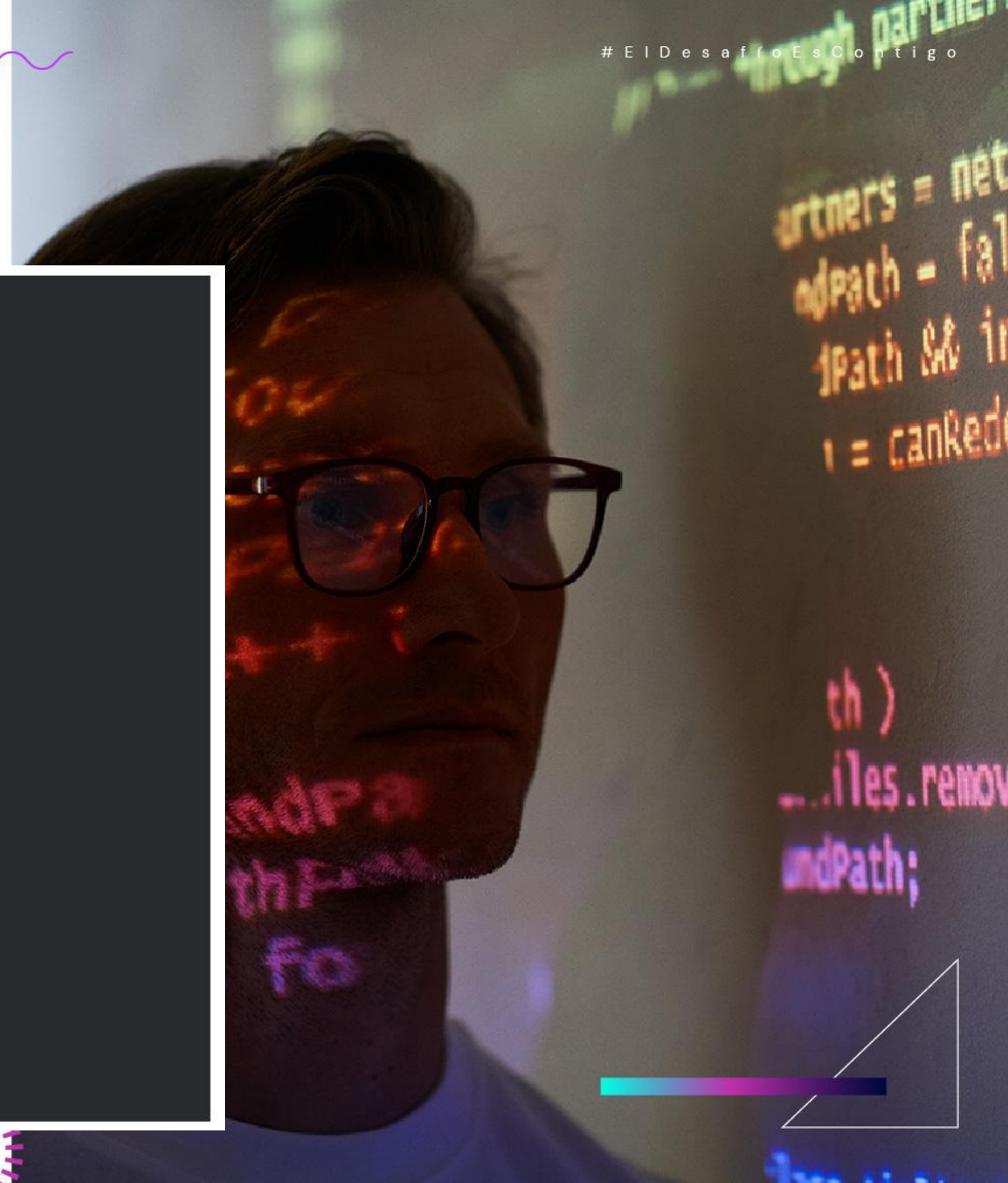
```
4
```

```
iex(58)> b
```

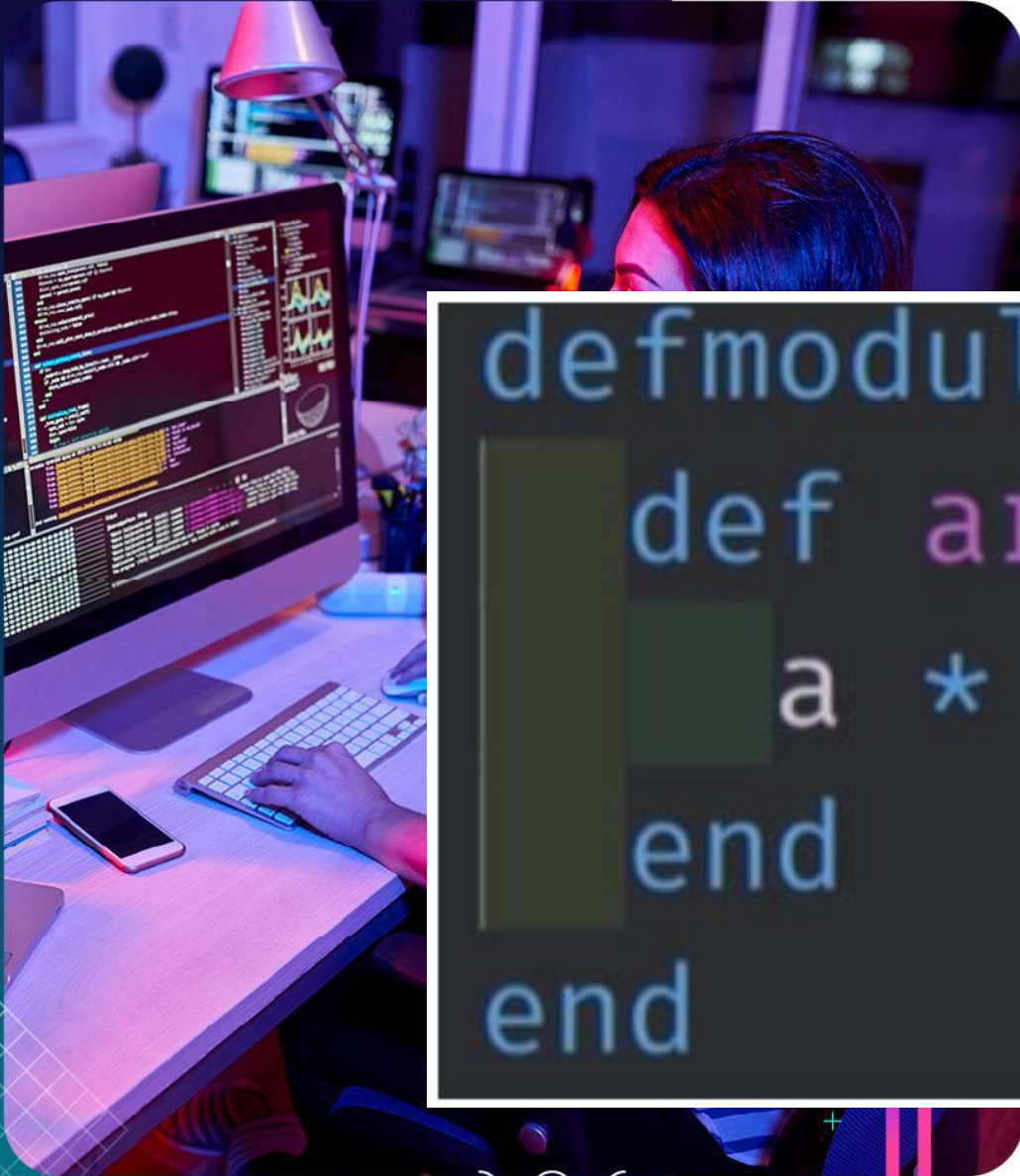
```
b
```

```
4
```


```
iex(59)> |
```



```
iex(59)> date_time = {_, {hour, _, _}} = :calendar.local_time()  
date_time = {_, {hour, _, _}} = :calendar.local_time()  
{2019, 4, 29}, {19, 41, 9}  
iex(60)> date_time  
date_time  
{2019, 4, 29}, {19, 41, 9}
```



```
defmodule Geometry do
  def area({a, b}) do
    a * b
  end
end
```

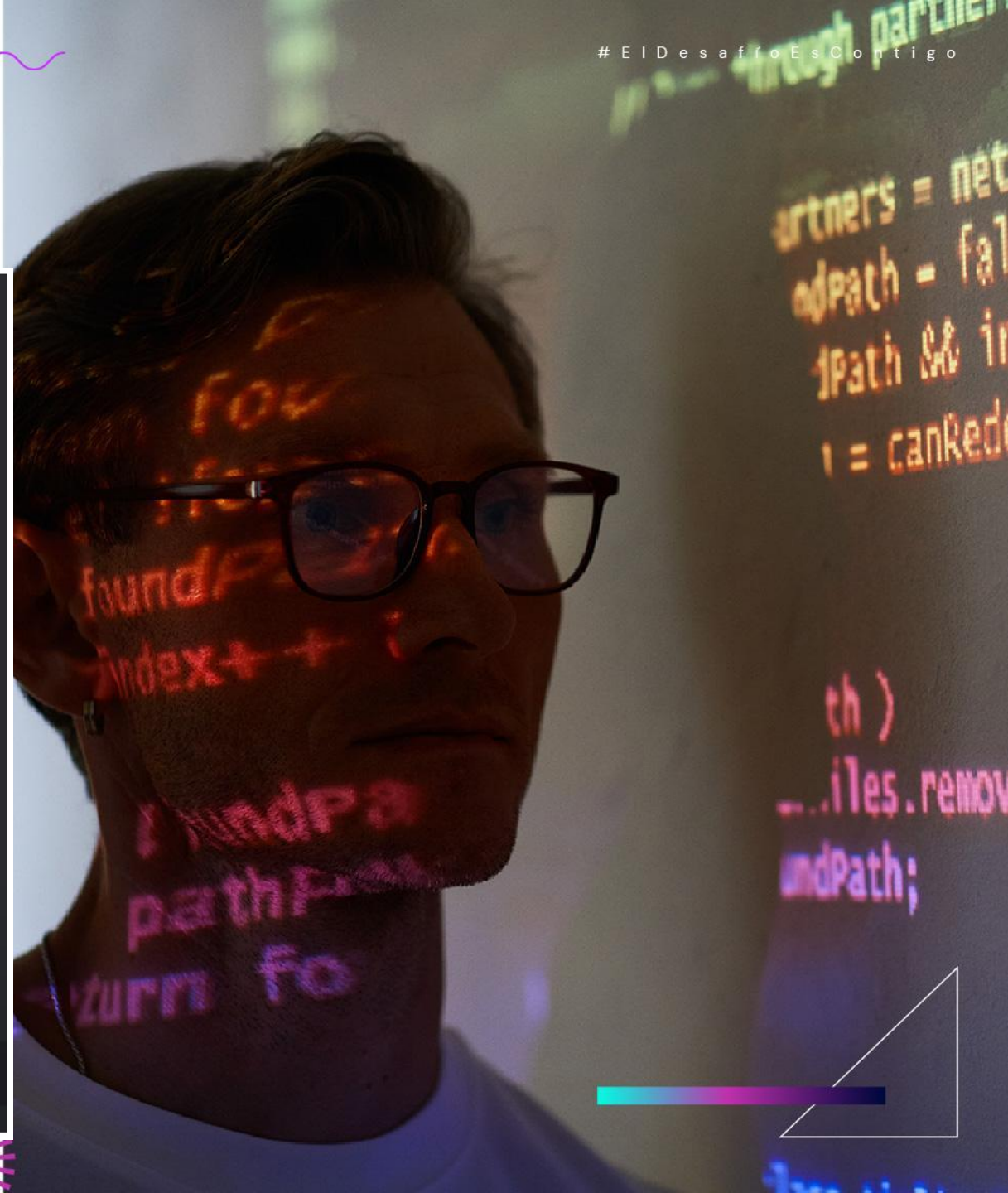
```
iex(64)> Geometry.area({2, 3})  
Geometry.area({2, 3})  
6
```

Sobrecarga de métodos:

```
defmodule Geometry do
  def area({:rectangle, a, b}) do
    a * b
  end

  def area({:square, a}) do
    a * a
  end

  def area({:circle, r}) do
    r * r * 3.14159
  end
end
```



```
iex(66)> Geometry.area({:circle, 2})
```

```
Geometry.area({:circle, 2})
```

```
12.56636
```

```
iex(67)> Geometry.area({:square, 20})
```

```
Geometry.area({:square, 20})
```

```
400
```

```
iex(68)> Geometry.area({:rectangle, 20, 30})
```

```
Geometry.area({:rectangle, 20, 30})
```

```
600
```

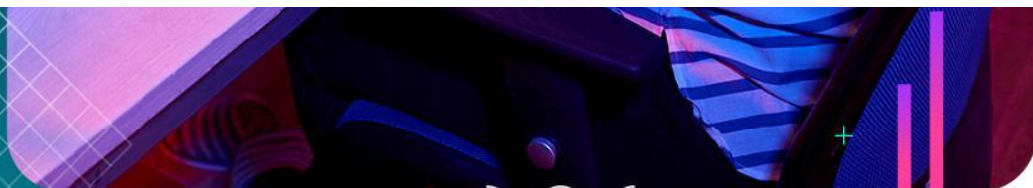



```
iex(69)> Geometry.area({:triangle, 1, 2, 3})
Geometry.area({:triangle, 1, 2, 3})
** (FunctionClauseError) no function clause matching in Geometry.area/1
```

The following arguments were given to Geometry.area/1:

```
# 1
{:triangle, 1, 2, 3}
```

```
main.ex:2: Geometry.area/1
```



Método por defecto:

```
def area(unknown) do  
  {:error, {:unknown_shape, unknown}}  
end
```

```
iex(70)> Geometry.area({:triangle, 1, 2, 3})  
Geometry.area({:triangle, 1, 2, 3})  
{:error, {:unknown_shape, {:triangle, 1, 2, 3}}}
```


< / >



Calle 12 # 30-80 Medellín

Calle 85 # 11 – 53 Int 6 Of. 301 Bogotá



+57 604 266 4547



info@sofka.com.co



www.sofka.com.co



Síguenos



Sofka Technologies



Sofka_Technologies

