

## Function Captures

En Gleam se pueden escribir lambdas del tipo:

```
let add_one_v1 = fn(x) { add(1, x) }
```

Como:

```
let add_one_v1 = add(1, _)
```

Esta sintaxis es válida solamente cuando la función anónima tiene un solo argumento. Esta funcionalidad ayuda mucho al momento de hacer currying.

## Pipelines

En Gleam existe el concepto de **pipe** como en bash. Se ve de la siguiente forma:

```
// Without the pipe operator
io.debug(string.drop_left(string.drop_right("Hello, Joe!", 1), 7))
```

```
// With the pipe operator
"Hello, Mike!"
  |> string.drop_right(1)
  |> string.drop_left(7)
  |> io.debug
```

## ¿Cómo funciona?

Si se tiene la siguiente pipeline:

```
a
  |> b(1,2)
```

Entonces Gleam primero intenta `b(a,1,2)` y si esto es inválido entonces intenta `b(1,2)(a)`. Recuerda que puedes usar Function Captures para determinar el lugar en el que quieres insertar el parámetro:

```
"1"
  |> string.append("2")
  |> string.append("3", _)
  |> io.debug
```

```
// outputs: "312"
```