

# Is Haskell Useless?

The road to effectful programming

¿Es Haskell inútil?

El camino hacia la programación con efectos

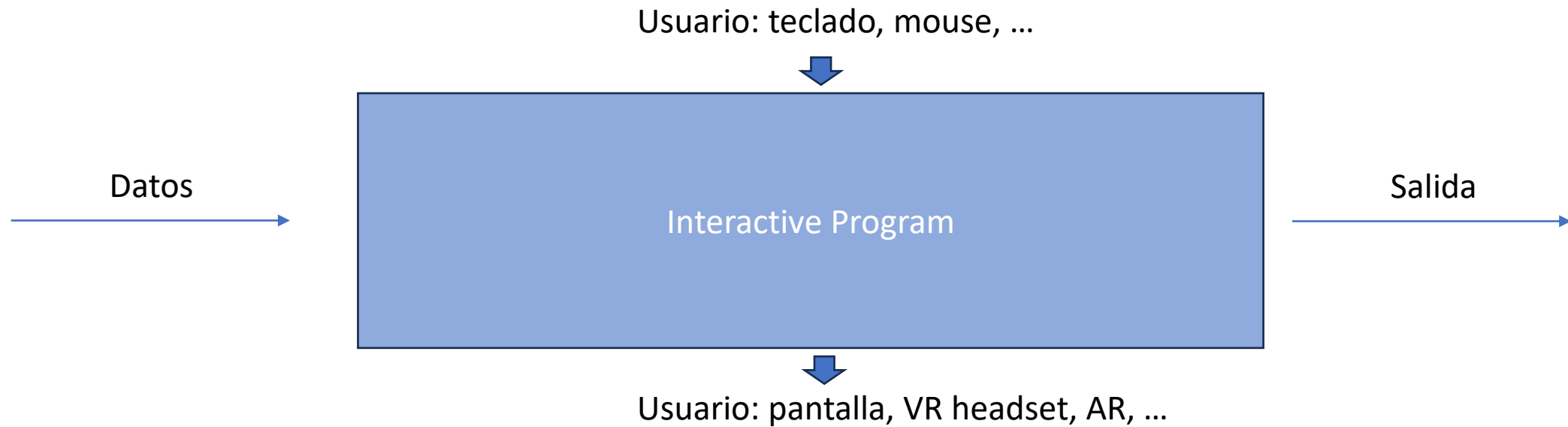
# Procesamiento en “Batch”

## Batch Processing



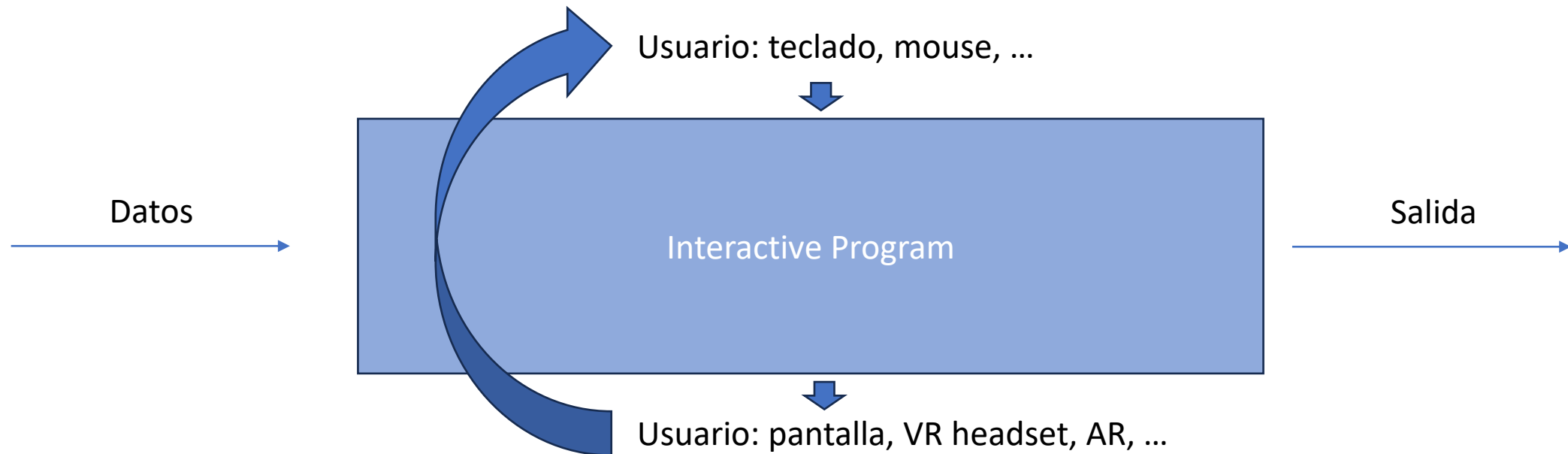
# Procesamiento Interactivo

## Interactive Processing



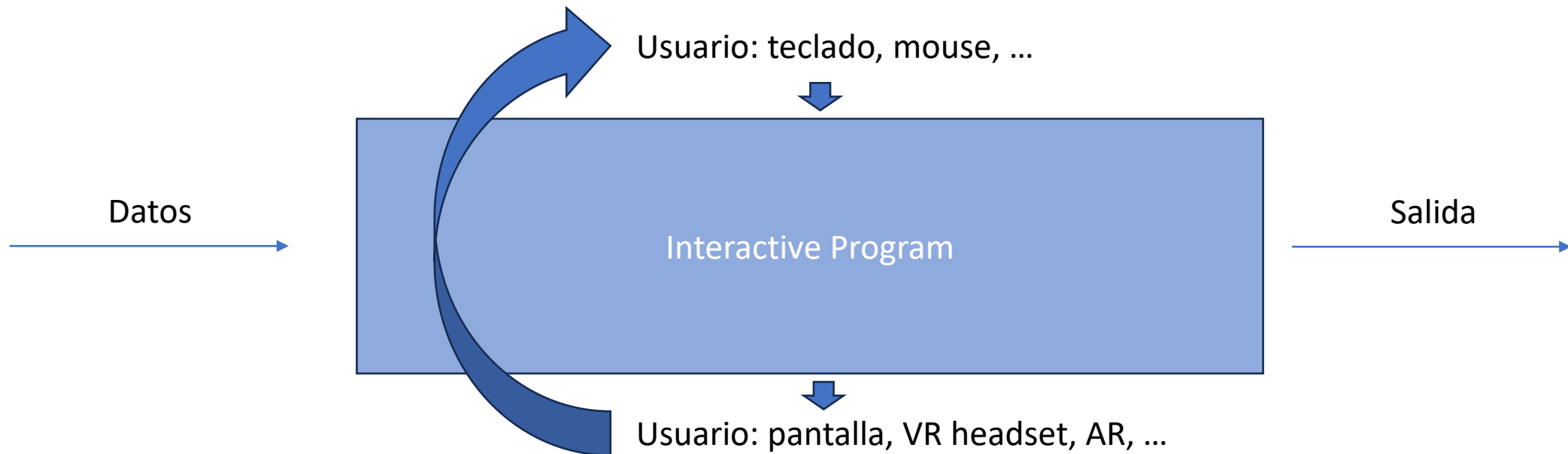
# Procesamiento Interactivo

## Interactive Processing



# Procesamiento Interactivo

## Interactive Processing



Haskell is Useless!

<https://www.youtube.com/watch?v=iSmkqocn0oQ>

# Modelo de Tipos Conceptual

type IO = Mundo -> Mundo

# Modelo de Tipos Conceptual

type IO = Mundo -> Mundo

type IO a = Mundo -> (a, Mundo)

# Modelo de Tipos Conceptual

type IO = Mundo -> Mundo

type IO a = Mundo -> (a, Mundo)

Realización del modelo conceptual:

data IO a = ... magic ...



# Acciones Básicas de IO

`getChar :: () -> IO Char`

`getChar = ... magic ...`

`putChar :: Char -> IO ()`

`putChar c = ... magic ...`

`putStr :: [Char] -> IO ()`

`putStr = ... not magic ... (ver lámina siguiente)`

# putStr

```
putStr :: String -> IO ()
```

```
putStr [] = return ()
```

```
putStr (c, cs) = do
```

```
    putChar c
```

```
    putStr cs
```

```
putStrLn :: String -> IO ()
```

```
putStrLn cs = do
```

```
    putStr cs
```

```
    putChar '\n'
```

# return

return :: a -> IO a  
return v = ... **magic** ...



[\(3009\) Olivia Newton John - Magic – YouTube](https://www.youtube.com/watch?v=sl5bqHP0-KA)

<https://www.youtube.com/watch?v=sl5bqHP0-KA>

“return provee el camino para convertir **expresiones puras**, que no tienen efectos colaterales, en **acciones impuras** con efectos colaterales” [Graham Hutton]

Intenten cambiar **return ()** por **()** en putStr y vean qué pasa ...

Prepare to Enter the Monad: Nov 3 – Nov 7

