



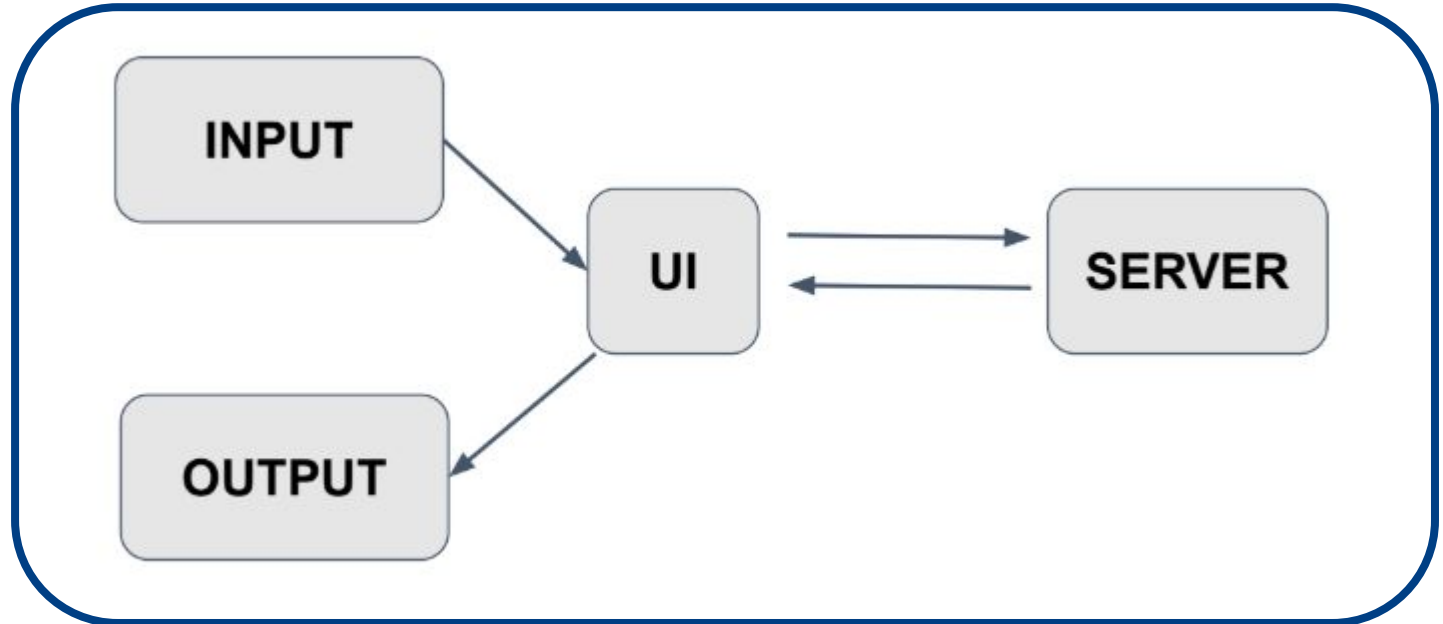
---

# Functions & Shiny modules

Mastering Shiny

---

# Interacción Shiny app





---

# 18. Functions

Mastering Shiny

---

---

## ¿Por qué usar funciones?

- Cuando tenés **bloques repetido** en **ui o server**,
  - Mejora la **legibilidad**.
  - Permite mantener tu **app escalable** a medida que crece.
-

---

# Escribir funciones para organizar el código:

- En la UI, permite reducir repeticiones. Si tenés componentes similares en varias partes, agruparlos en una función mejora la claridad y facilita cambios globales.
  - En el server, mover reactivos complejos a funciones independientes (aunque se usen una sola vez) facilita la depuración, ya que podés probarlos fuera del contexto reactivo.
  - Además, usar funciones permite dividir tu app en varios archivos, en lugar de tener todo en un único `app.R`, lo que hace el desarrollo más manejable.
-

```
# Función para crear sliders
```

```
sliderPercentInput <- function(id, label) {  
  sliderInput(id, label, min = 0, max = 1, value = 0.5, step = 0.01)  
}
```

**UI**

```
ui <- fluidPage(  
  sliderPercentInput("input1", "Primero"),  
  sliderPercentInput("input2", "Segundo"),  
  sliderPercentInput("input3", "Tercero")  
)
```

```
server <- function(input, output, session) {}
```

```
shinyapp(ui, server)
```

```
generate_data <- function(n, mean = 0, sd = 1) {  
  rnorm(n, mean, sd)  
}
```

```
hist_data <- function(input) {  
  reactive({  
    generate_data(input$n, input$mean, input$sd)  
  })  
}
```

```
ui <- fluidPage(plotOutput("hist"))
```

```
server <- function(input, output, session) {  
  data <- hist_data(input)  
  
  output$hist <- renderPlot({  
    hist(data(), col = "skyblue", border = "white")  
  })  
}
```

**shinyapp(ui, server)**

---

## 18. Functions

Extraer **funciones** puede parecer lento al principio, pero con práctica se vuelve una herramienta fundamental. Eso sí, estas **funciones sólo manejan UI o server**, no ambos. Para eso están los **módulos**

---



---

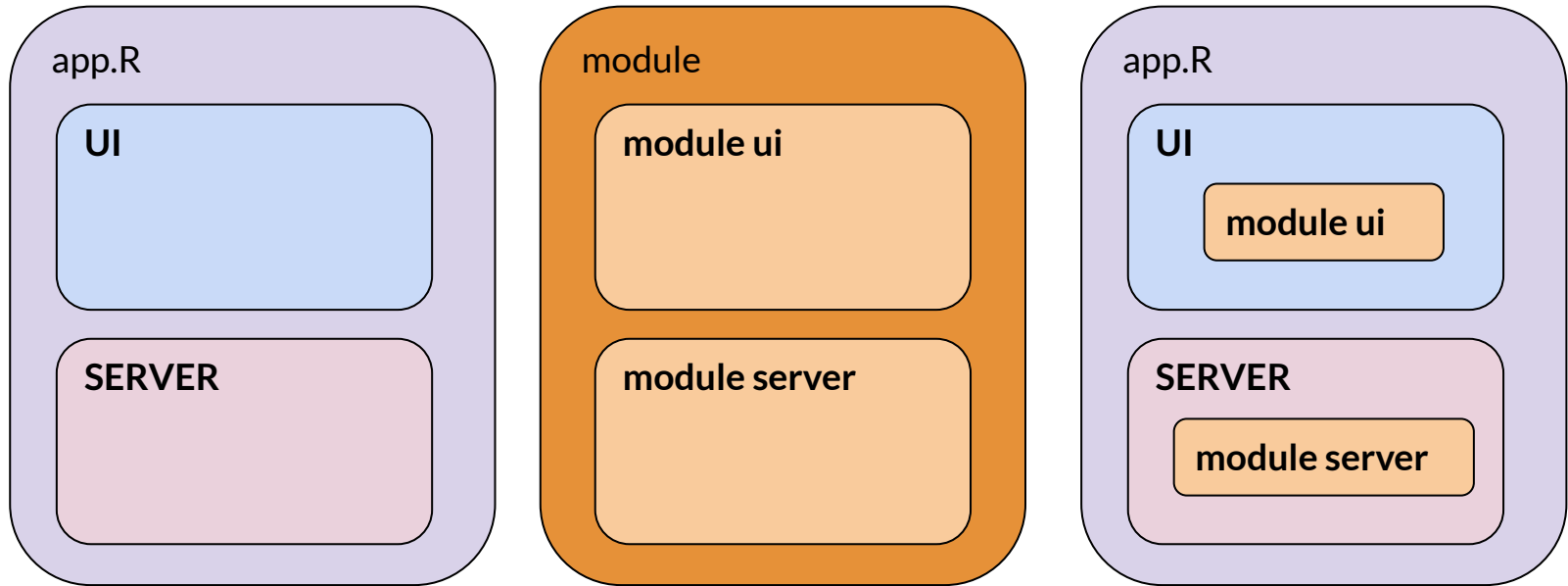
# 19. Modules

Mastering Shiny

---

---

Los módulos permiten dividir la app en partes **autónomas y reutilizables**.



---

Característica	Funciones	Módulos
¿Qué encapsulan?	Solo UI o solo server	UI y server juntos
¿Para qué sirven?	Reutilizar fragmentos simples	Reutilizar componentes completos e independientes
¿Reutilizables múltiples veces?	Sí, pero pueden requerir ajustes	Sí, están diseñados para eso con namespacing
¿Usan `NS()` y `moduleServer()`?	No	Sí
¿Uso en apps grandes?	Para organizar lógicas o fragmentos comunes	Para estructurar la app en partes independientes
¿Fácil de probar por separado?	Sí, especialmente si no dependen de la app	Sí, pero requieren estructura propia
Ejemplo típico	Un gráfico con parámetros comunes	Un selector + gráfico + tabla como una unidad reutilizable

---

---

### ¿Cuándo usar funciones?

- Cuando tenés muchas repeticiones de código **simples**
- Para separar lógicas que no dependen de input, output, session
- Para mejorar la legibilidad de tu código

### ¿Cuándo usar módulos?

- Cuando una parte de la app **tiene UI + lógica propia**
  - Cuando querés usar **varias veces** el mismo componente en distintas partes
  - Para dividir tu app en partes independientes y escalables
-

- 
- [Effective use of Shiny modules](#)
  - [Shiny Modules in 100 Seconds](#)
  - [Avoid code duplication with Shiny modules](#)
  - [Structure your app: introduction to Shiny modules](#)
  - [GitHub - jonas-hag/structure\\_your\\_app: Course material for "Structure your app: introduction to Shiny modules"](#)
  - [Using Shiny modules](#)

**Recursos interesantes**

---



---

---

Gracias ✨ ✨ ✨ ✨



<https://github.com/SoyAndrea/shiny-modules>

Mastering Shiny

---