

# Módulo 4. Dashboards personalizables e interactivos para la comunicación de resultados



☰ Unidad 1

☰ Actividades

☰ RShiny: partes fundamentales

☰ Actividades

☰ Referencias

# Unidad 1

---

En los cursos anteriores, hemos destacado la gran cantidad de información de que disponemos como sport scientists [analistas de rendimiento físico] con el desarrollo de nueva tecnología y aparición de nuevos métodos de trabajo derivados de investigaciones científicas. Existe una creciente necesidad de analizar parámetros para tomar las mejores decisiones posibles que afecten positivamente al rendimiento de nuestros jugadores u organización.

También vimos los distintos tipos de comunicación de resultados: desde gráficos sencillos que nos permitan representar directamente y de forma resumida aquello que queremos mostrar hasta informes automatizados que nos permitan repetibilidad en el análisis en distintos momentos de la temporada o para distintos jugadores o equipos. Finalmente, se hizo mención de herramientas de visualización o distribución de resultados que son más interactivas o que permiten seguir una historia/flujo con los datos.

Estas herramientas se diferencian del resto, ya que no responden a esa necesidad inmediata de presentar los resultados de una manera

determinada y tampoco pretenden estandarizar la manera en la que se visualizan. Estas herramientas incorporan múltiples fuentes de información. En el caso de los datos de rendimiento deportivo, pueden abarcar desde datos de rendimiento físico, lesiones e historia médica hasta participación o resultados en competición. No solo disponen de múltiples fuentes de datos, sino que habitualmente también disponen un mayor periodo registrado de estos mismos datos.

Tener a disponibilidad del usuario final una herramienta que permita la exploración de los datos y que interactúe con las distintas fuentes de información permite aportar mayor contexto sobre el elemento que estamos explorando, ver qué variables pueden estar afectando unas a otras y cuál es la tendencia a lo largo del tiempo.

En el ámbito del análisis de datos de rendimiento físico, estas herramientas comúnmente reciben el nombre de AMS (*athlete management/monitoring systems* [gestión de atletas/sistemas de monitoreo]). Estas permiten, en muchos casos, el registro de información directamente en ellas, pero sobre todo permiten disponer de múltiples variables de información sobre el atleta o el equipo. Existen múltiples opciones comerciales según el tipo de datos o tecnología que utilicemos y las particularidades o funcionalidades variarán en función de la empresa proveedora.

Como describen Compton et al. (2019), el objetivo de la elaboración de un sistema de monitorización del atleta es ayudar en los procesos de

toma de decisiones que tienen que ver con la planificación y las modificaciones en el entrenamiento, así como reducir el riesgo de lesión. Destacan, también, que es fundamental conocer cómo deben ser analizados los datos y qué parámetros significativos deben ser analizados para poder extraer y presentar información de manera efectiva.

Otro punto fundamental, aportado por Coutts et al. (2018), es la necesidad de elaborar los *dashboards* [paneles de control] o AMS en función del marco conceptual que se decida, además de estar integrado con la visión de la organización. Con esto, se refieren a determinar la necesidad que queremos responder, de acuerdo con la metodología y filosofía de trabajo, para establecer el diseño y la estructura de la herramienta.

Las principales soluciones que proporciona un AMS son las siguientes.

- Centralizar datos de distintas fuentes en una misma plataforma.
- Fácil acceso a los datos para su comunicación.
- Interactividad para la exploración de resultados.
- Guiar en la toma de decisiones.

- Contextualizar los valores que obtenemos.
- Compartir los resultados de manera eficiente.

Como vemos, para conseguir estos objetivos, no es suficiente con informes automatizados o gráficos sencillos, es por eso por lo que existen múltiples herramientas para la creación de visualizaciones interactivas como Tableau o Power BI. RStudio, de nuevo, se presenta también como un software capaz de desarrollar herramientas como los AMS. Las ventajas de utilizar RStudio en lugar de otras herramientas es su gran capacidad de personalización y creación de *dashboards* o herramientas interactivas adaptadas a las necesidades específicas. Además, podremos integrar fácilmente cualquier tipo de análisis o transformación de datos que realicemos en ellos. No hay que olvidar que se trata de una herramienta libre, por lo que la ausencia de coste económico es un gran punto a favor en muchos de los casos.

RShiny es la librería dentro de RStudio que permitirá crear aplicaciones web o dashboards interactivos. Permite la personalización de cada una de las partes y la integración del análisis de datos y su visualización.

Ya vimos algún ejemplo en módulos anteriores sobre estas visualizaciones, pero mostramos dos ejemplos más de posibles diseños

de *dashboards* que puedan ser aplicados al contexto del *sport scientist* [ciencias del deporte].

En esta primera imagen, vemos un dashboard creado para el análisis de una sesión de entrenamiento, en este caso, para poder visualizar las demandas físicas a partir de los datos obtenidos mediante dispositivos GPS. Como vemos en los paneles de la izquierda, hay ciertos elementos que invitan a la interacción del usuario de la aplicación o el dashboard. Este es su elemento fundamental. El usuario puede seleccionar, entre otras, la sesión a analizar, los jugadores de cierta posición o variables específicas.

¿Cuáles de las siguientes afirmaciones son ventajas de utilizar RStudio en lugar de otras herramientas?

---

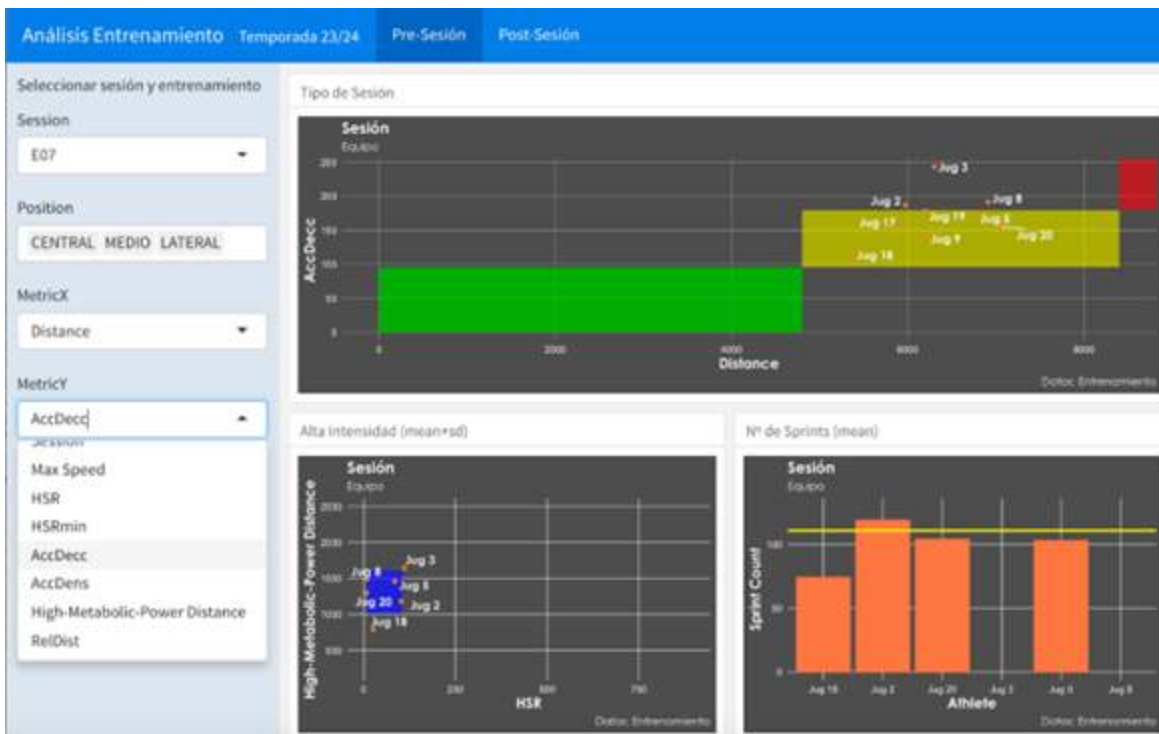
- Permite una gran capacidad de personalización y creación de dashboards.
- Facilita la integración de cualquier tipo de análisis o transformación de datos.
- Es una herramienta de pago con opciones avanzadas.

- Es una herramienta libre, sin coste económico.
- Se adapta a necesidades específicas a través de herramientas interactivas.

SUBMIT

Como hemos repetido durante el curso, tenemos a nuestra disposición una gran cantidad de variables y métricas después de cada sesión o entrenamiento. En función de nuestros objetivos, momento de la temporada o estado del jugador, estaremos más interesados en conocer los resultados de una variable u otra. El *dashboard* permite esa interacción y guía en la toma de decisiones, determinando dónde se encuentra el jugador respecto al resto del equipo.

## Figura 1. RStudio



Fuente: captura de pantalla de RStudio (Allaire, 2011). Captura de pantalla de App de RStudio de elaboración propia.

En este segundo ejemplo, vemos el mismo formato de *dashboard* pero con un objetivo distinto. Esta aplicación pretende ayudar en la toma de decisiones antes del entrenamiento. Utilizando los datos históricos del equipo y relaciones estadísticas entre duración, espacio y densidad de jugadores, podemos estimar la carga de cada una de las posiciones en las distintas métricas de rendimiento que seleccionemos. De esta manera, podemos interactuar con el *dashboard*, seleccionando tareas y duraciones para establecer un plan antes del entrenamiento. El mismo principio lo podemos aplicar a sesiones de readaptación o variables de carga interna.

## Figura 2. Tabla en RStudio

Estimación Demandas Entrenamiento Entrenamientos Tabla

Selección de Tareas y Duraciones

Tarea1  
4x4+2\_32x30

Duración  
13

Tarea2  
7x7\_50x40

Duración  
13.5

Tarea3  
10x10\_60x65

Duración  
15

Tarea4

Distancia

Position Name	tarea	dist	HSR	SP	HMLe	Acc	Dec	Id
CENTRAL	4x4+2_32x30	1293.765	83.70768	1.538772	34.96872	23.57009	24.50281	Tarea1
DELANTERO	4x4+2_32x30	1282.069	77.01904	3.155175	42.73877	31.22839	29.14458	Tarea1
EXTREMO	4x4+2_32x30	1418.467	138.15757	11.358945	48.73157	37.02446	32.04921	Tarea1
LATERAL	4x4+2_32x30	1311.919	97.79842	3.900825	44.25534	35.12211	29.58037	Tarea1
MEDIA PUNTA	4x4+2_32x30	1324.935	73.01000	3.062540	38.46853	29.49882	29.20793	Tarea1
MEDIO	4x4+2_32x30	1377.162	103.68255	1.064865	44.91278	31.55302	33.32216	Tarea1
CENTRO								
CENTRAL	7x7_50x40	1529.191	44.44700	0.000000	30.66289	21.04991	19.72630	Tarea2
DELANTERO	7x7_50x40	1721.299	77.47448	0.000000	41.12929	31.43824	22.71417	Tarea2
EXTREMO	7x7_50x40	1993.910	200.62214	32.378251	60.07669	39.26099	34.72123	Tarea2
LATERAL	7x7_50x40	1721.201	186.98169	33.496796	45.74771	33.43666	32.12843	Tarea2
MEDIA PUNTA	7x7_50x40	1673.685	25.21287	0.000000	36.02827	29.33929	19.97470	Tarea2
MEDIO	7x7_50x40	1823.571	81.58862	5.266310	41.80884	26.85531	26.41103	Tarea2
CENTRO								
CENTRAL	10x10_60x65	2681.667	114.12459	8.246749	27.15031	29.71375	31.58847	Tarea3
DELANTERO	10x10_60x65	2628.850	172.20936	20.717385	31.25805	35.02709	31.71850	Tarea3
EXTREMO	10x10_60x65	2762.913	296.52306	38.215459	53.79457	43.20400	46.41205	Tarea3
LATERAL	10x10_60x65	2728.403	248.80196	37.620481	41.68245	39.88912	41.70034	Tarea3

Fuente: captura de pantalla de RStudio (Allaire, 2011). Captura de pantalla de App de RStudio de elaboración propia.

CONTINUAR

## Actividades

---

¿Cuál de las siguientes afirmaciones sobre los sistemas de gestión o monitoreo de atletas (AMS) es correcta?

---

- Permiten registrar información directamente en ellos.
- Solo funcionan con un tipo específico de datos.
- Ofrecen múltiples variables de información sobre atletas o equipos.
- Todas las herramientas AMS tienen las mismas funcionalidades.
- Las funcionalidades de los AMS pueden variar según la empresa proveedora.

SUBMIT

CONTINUAR

## RShiny: partes fundamentales

---

Todos los procesos, análisis o uso de funciones en RStudio siguen el mismo patrón, debemos conocer cuáles son los nombres de las funciones que utilizamos, cuál es su estructura y sus argumentos. Para ser eficientes en su uso, debemos repetir el proceso múltiples veces, conocer las posibles causas de los errores que obtenemos y familiarizarnos con los resultados. RStudio se trata de una herramienta de cierta dificultad de aprendizaje en sus inicios debido a su complejidad, y RShiny, como parte de RStudio, tendrá esas mismas particularidades. Es una funcionalidad avanzada, por lo que deberemos conocer sus principios básicos y estructura para utilizarlo como punto de partida para aplicaciones de mayor envergadura y proyectos más avanzados. Tenemos múltiples recursos disponibles para profundizar en la herramienta y ajustarla a nuestras necesidades como profesionales.

Existen dos componentes principales de cada aplicación Shiny (Wickham, 2021).

- UI: es la interfaz de usuario. En esta parte, deberemos realizar las modificaciones pertinentes para definir el aspecto de la aplicación, desde la distribución de los gráficos o *inputs* [entradas] de selección hasta los colores o páginas de la aplicación.
- Server: este elemento define cómo trabaja la aplicación, es decir, qué código o datos están siendo procesados y de qué manera se visualizan.

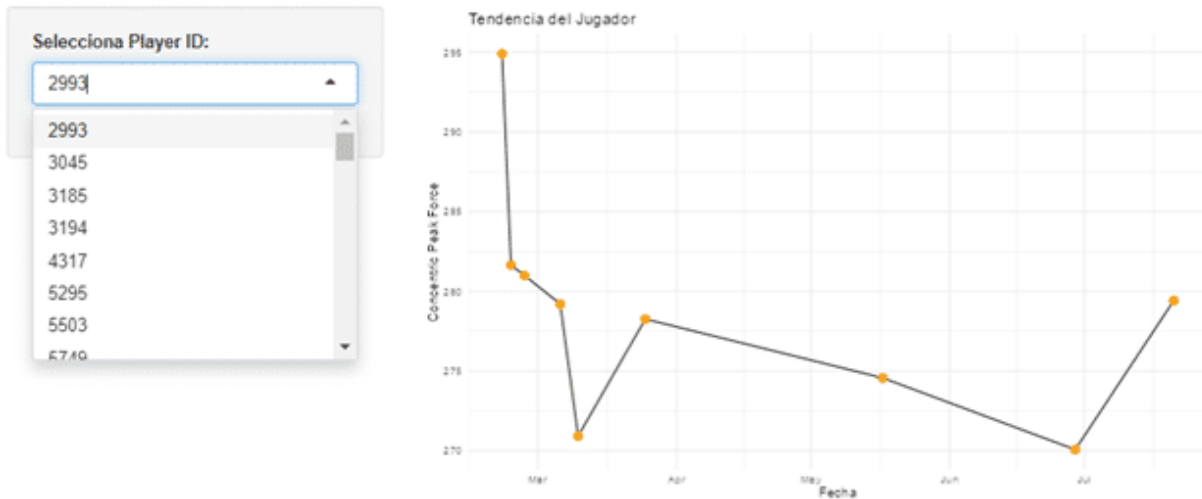
Necesitaremos definir cada una de estas partes con su estructura correspondiente para poder utilizar la aplicación. Aunque veamos ejemplos durante el material de video, vamos a explicar la lógica de esta aplicación de ejemplo para comprender qué está sucediendo en cada una de las partes.

Queremos representar la tendencia de los valores de un jugador en una métrica analizada mediante plataforma de fuerza. La aplicación tiene el siguiente aspecto.

A la izquierda, ubicamos un elemento donde podemos filtrar por el identificador del jugador, cuando se seleccione el ID se actualizará automáticamente el resultado del gráfico.

## Figura 3. Tendencia del jugador en RStudio

### Tendencia Jugador



Fuente: captura de pantalla de RStudio (Allaire, 2011). Captura de pantalla de App de RStudio de elaboración propia.

Veamos el código que corresponde a cada una de las partes

En primer lugar, importamos las librerías con las que queremos trabajar y la base de datos utilizada. En este caso, se trata de datos que ya hemos utilizado con anterioridad, hay columnas que indican la fecha, el identificador del jugador y cada una de las variables de análisis.

## Figura 4. Base de datos

```
library(tidyverse)
library(shiny)
library(lubridate)

datos_salto <- read.csv("forceplates.csv", sep=",")

ui <- fluidPage(
  titlePanel("Tendencia Jugador"),
  sidebarLayout(
    sidebarPanel(
      selectInput("player", "Selecciona Player ID:", choices = unique(datos_salto$player_id))
    ),
    mainPanel(
      plotOutput("trendPlot")
    )
  )
)
```

Fuente: elaboración propia.

---

El segundo apartado es definir la interfaz de usuario.

- fluidPage: esta función permitirá que las dimensiones de los gráficos y elementos se ajusten a la dimensión de la ventana en la que los estemos visualizando.
- titlePanel: título que queremos que se muestre en la aplicación.
- sidebarLayout-sidebarPanel: permite definir los elementos que queremos que se muestren a la izquierda de la aplicación para realizar distintos filtros. A continuacin, encontraremos otros ejemplos

de elementos que podemos seleccionar para las aplicaciones en función de las necesidades.

- selectInput: se ha decidido que vamos a utilizar un elemento que permita seleccionar un jugador específico dentro de la lista de todos los indicadores.
  - «Player»: será el indicador que utilizaremos en la parte de «server».
  - «Selecciona Player ID»: es el texto que queremos mostrar encima del elemento de selección.
  - «Choices»: son todas las posibles opciones que queremos que el usuario tenga (en este caso, todos los indicadores de los jugadores obtenidos utilizando la función unique() que ya vimos en cursos anteriores).
- mainPanel: qué queremos que se muestre en el panel de visualización principal, en este caso un gráfico (plotOutput) al cual decidimos asignar el nombre de «trendPlot».

## **Figura 5. Elemento «server»**

```

server <- function(input, output) {
  output$trendPlot <- renderPlot({

    player_data <- datos_salto %>%
      filter(player_id == input$player)

    ggplot(player_data, aes(x = ymd(test_date), y = concentric_peak_force)) +
      geom_line(group="player") +
      geom_point(size=3,color="orange") +
      labs(title = paste("Tendencia del Jugador"),
           x = "Fecha",
           y = "Concentric Peak Force")+
      theme_minimal()
  })
}

shinyApp(ui = ui, server = server)

```

Fuente: elaboración propia.

---

Para definir el elemento «server», necesitamos crear una función que siga la estructura mostrada. En este caso, contiene «input» (la selección del jugador que queremos) y «output» (el gráfico que queremos mostrar).

- `output$trendPlot <- renderPlot`: tiene el mismo nombre que hemos decidido asignar en la `iu`; si tuviéramos más de un gráfico a mostrar, el nombre indicaría que visualización estamos modificando.
- `player_data`: realizamos modificaciones sobre nuestra base de datos para filtrar solo el jugador deseado, para ello, utilizamos `input$player` y de nuevo utilizamos el mismo nombre «player» que hemos decidido en la `iu`.

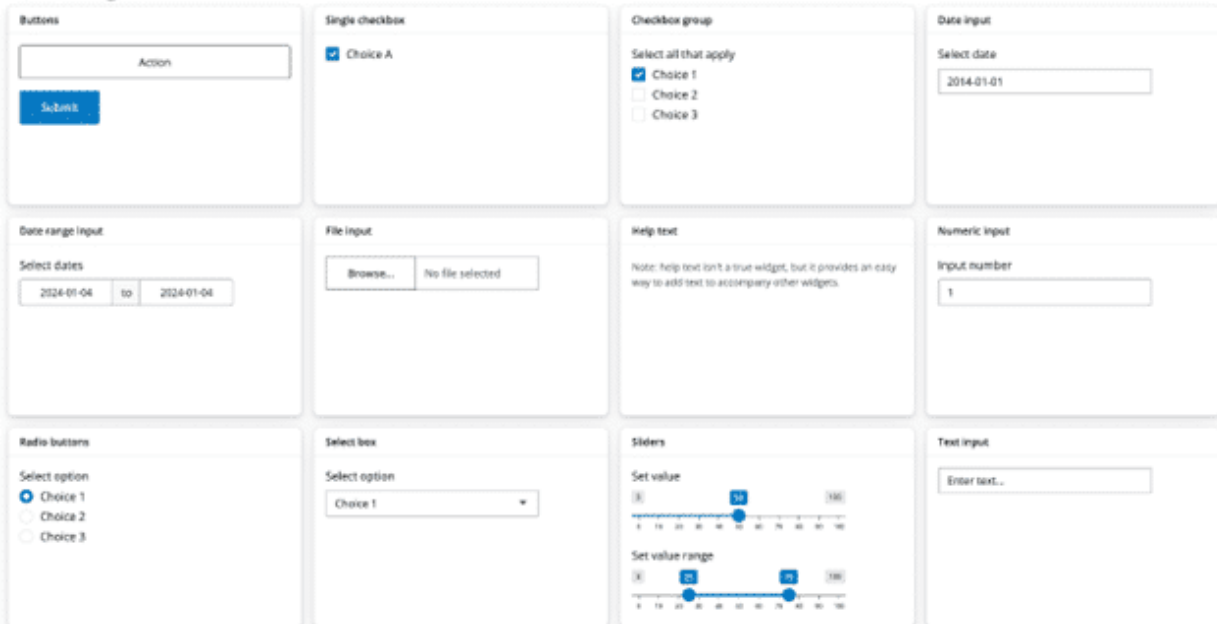
- Gráfico ggplot: de la misma manera que visualizamos en módulos anteriores, se elabora un gráfico para visualizar la información de la manera deseada.

Finalmente, se utiliza la función shinyApp con ambos elementos creados para lanzar la visualización de la app.

Este es un ejemplo muy simple e introductorio que sirve como punto de partida para la creación de herramientas de mayor complejidad. Disponemos de recursos que permiten conocer las posibilidades de esta librería. En la página de <https://shiny.posit.co/> podemos encontrar ejemplos de visualizaciones, así como descripción de tipos de elementos (widgets [módulos interactivos]) que podemos la interfaz de usuario para nuestras visualizaciones.

## **Figura 6. Widgets**

## Basic widgets



Fuente: captura de pantalla de <https://shiny.posit.co/>

Es importante remarcar la premisa de establecer un objetivo a partir del cual construir las aplicaciones, de la misma manera que ocurre con cualquier otra parte del proceso de análisis. Conocer las fortalezas de RStudio y RShiny nos dota de competencias que pueden ser claves para aportar valor en nuestro contexto de rendimiento deportivo.

CONTINUAR

# Actividades

---

Existen dos componentes principales de cada aplicación Shiny (Wickham, 2021). ¿Cuáles son?

---

- UI
- Server
- Player\_data
- RenderPlot
- FluidServer

SUBMIT

CONTINUAR

## Referencias

---

**Allaire, J.** (2011). RStudio (2023.06.1) [herramienta de *software*]. Posit.

**Compton, H., Delaney, J., Duthie, G. y Dascombe, B.** (2019). Developing Athlete Monitoring Systems in Team Sports: Data Analysis and Visualization. *International Journal of Sports Physiology and Performance*, 14 (6), 698–705. <https://doi.org/10.1123/ijsp.2018-0169>

**Coutts, A., Crowcroft, S. y Kempton, T.** (2018). Developing athlete monitoring systems: Theoretical basis and practical applications en M. Kellmann & J. Beckmann (Eds.), *Sport, Recovery and Performance: Interdisciplinary Insights* (pp. 19-32). Routledge.

**Wickham, H.** (2021). *Mastering Shiny*. O'Reilly Media, Inc.

**Posit team.** (2023). RStudio: Integrated Development Environment for R. Posit Software, PBC, Boston, MA. URL: <http://www.posit.co/>.

CONTINUAR