# "Explorando Julia: Aprende a Programar como un Científico"

# Equipo de Trabajo:

- Gobea, David
- Gonzalez, Maria
- Hernandez, Manuel
  - Solett, Jose

#### Introducción:

¡Bienvenidos a la aventura de aprender a programar con Julia! Julia es un lenguaje de programación especial que usamos para dar instrucciones a las computadoras y hacer que realicen tareas. Es como un lenguaje secreto que puedes aprender para crear cosas increíbles, como juegos, dibujos y hasta resolver problemas de matemáticas.

En esta guía, vamos a explorar Julia de una forma fácil y divertida. Empezaremos con lo más básico, como los números y palabras que usamos para que Julia entienda lo que queremos hacer. Luego, aprenderemos a guardar información en cajas llamadas variables y a darles valores que usaremos más tarde.

A medida que avancemos, veremos cómo Julia nos permite hacer preguntas y tomar decisiones para que el programa haga diferentes cosas según lo que necesitemos. También aprenderemos a repetir tareas usando ciclos, a usar herramientas especiales llamadas funciones, y a importar otras cajas de herramientas para que podamos hacer aún más cosas.

Esta guía es para aquellos que quieren empezar a programar desde cero y conocer lo básico de Julia. No importa si nunca has programado antes; aquí aprenderemos paso a paso. Al final, sabrás lo suficiente como para empezar a crear tus propios programas. ¡Vamos a aprender y a divertirnos con Julia!

## Requerimientos e instalación:

Para crear programas divertidos requerimos de previamente instalar Julia; para lo que necesitaremos una computadora con 4GB de RAM y al menos 300MB de espacio en el disco duro. No importara que sistema operativo utilices ya que julia es multiplataforma. Los pasos para instalar julia son los siguientes:

- 1. Dirígete a <a href="https://julialang.org/downloads/">https://julialang.org/downloads/</a>.
- Selecciona la versión de Julia que funciona para tu computadora, ya sea Windows, macOS o Linux.
- 3. Haz clic en el botón para descargar el archivo.
- 4. Abre el archivo que descargaste y sigue las instrucciones. Normalmente, solo tienes que hacer clic en "Siguiente" unas cuantas veces hasta que termine.
- Si usas Windows, selecciona la opción Agregar Julia al PATH para poder usar Julia desde cualquier lugar.
- Aunque puedes usar Julia directamente, hay un programa llamado VS Code que hace que programar sea más fácil.
- Descarga VS Code de <a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a> y luego, dentro de VS Code, instala la extensión de Julia para empezar a programar.

Ya podemos empezar con nuestra tan ansiada aventura, sigamos aprendiendo...

## 1. Tipos de Datos:

Son las diferentes formas en las que Julia guarda la información: números, palabras, y respuestas de sí o no. Entender los tipos de datos nos ayuda a saber qué tipo de información podemos guardar y cómo usarla en nuestro programa.

- Números Enteros (Int): Son como los números completos, sin decimales, como 7 o -3.
- Números Decimales (Float): Son números con punto decimal, como 2.5 o
   -8.2.
- Texto (String): Son palabras o frases que escribimos entre comillas, como "Hola, amigos".
- Booleanos (Bool): Son respuestas de sí o no, true o false.

#### 2. Variables:

Las variables son como cajitas con nombres que guardan información para que la usemos luego. Nos permiten almacenar información que queremos usar más adelante o que cambia mientras el programa corre. Cada caja tiene un nombre y dentro le podemos poner un valor. Para guardar un valor en una variable usamos el símbolo "=", por ejemplo, guardar el nombre de un amigo como: nombre = "Luis".

#### 3. Constantes:

Parecidas a las variables, pero su valor no cambia una vez que lo defines. Nos ayudan a guardar información que sabemos que no debe cambiar nunca. A pesar de que en julia no existen las constantes se toma por convención a las variables que tienen su nombre en mayúsculas, **por ejemplo**, **PI = 3.14** para los cálculos.

## 4. Estructuras de Datos:

Son formas de organizar varios datos en un solo lugar. En Julia, las más comunes son listas, tuplas y diccionarios. Ayudan a guardar más de un valor a la vez, como si tuvieras varias cajitas dentro de una caja grande. Algunas de estas estructuras son:

- Listas (Arrays): Son como cajas grandes que guardan varias cosas, en un orden específico, por ejemplo, frutas = ["manzana", "naranja", "plátano"].
- Tuplas (Tuple): Parecidas a las listas, pero no se pueden cambiar, por ejemplo, coordenadas = (4, 7).
- Diccionarios (Dict): Guardan información en pares de "clave" y "valor".
   Como una libreta de contactos, donde el nombre es la clave y el número de teléfono es el valor, por ejemplo, direcciones = Dict("casa" => "123 Calle Falsa", "escuela" => "456 Avenida Verdadera").

## 5. Operadores:

- Operadores Aritméticos: Son símbolos que usamos para hacer cuentas, como suma "+", resta "-", multiplicación "\*", división "/" y potencia "^".
   Nos permiten hacer cálculos y trabajar con números. Por ejemplo, sumar dos números, suma = 4 + 3.
- Operadores de Comparación: Estos operadores se utilizan para hacer comparaciones entre números, letras, y otros tipos de datos que Julia entiende. Estos son:
  - "==": Compara si dos valores son iguales. Si lo son, el resultado es verdadero; si no, es falso, por ejemplo, 3 == 3 da verdadero.
  - "!=": Verifica si dos valores son diferentes. Si son diferentes, el resultado es verdadero; si no, es falso, por ejemplo, 5 != 3 da verdadero.
  - ">": Comprueba si un valor es mayor que otro, por ejemplo, 7 > 5
    da verdadero.
  - "<": Comprueba si un valor es menor que otro, por ejemplo, 4 < 6 da verdadero.
  - ">=": Verifica si un valor es mayor o igual a otro, por ejemplo, 8
     >= 8 da verdadero.
  - "<=": Verifica si un valor es menor o igual a otro, por ejemplo, 2</li>
     3 da verdadero.
- Operadores de Asignación: Son símbolos que ayudan a guardar el resultado de operaciones en una variable, como "=" o "+=". Estos son:
  - o "=": Asigna un valor a una variable.

- o "+=", "-=", "\*=", "/=": Operadores que modifican el valor de una variable a la vez.
- Operadores Lógicos: Son símbolos como "&&" (y) y "||" (o), que usamos para comparar valores y tomar decisiones. Nos ayudan a verificar varias condiciones a la vez y decidir si algo es verdadero o falso. Estos son:
  - Y (&&): Verifica si ambas condiciones son verdaderas, por ejemplo, mayor\_edad = (edad >= 18) && (edad < 65).</li>
  - O (||): Verifica si al menos una condición es verdadera, por ejemplo, fuera\_rango = (numero < 5) || (numero > 10).
  - No (!): Cambia un valor true a false y viceversa, por ejemplo, mentira = !true.

## 6. Entrada y Salida de Datos:

La entrada es cuando el programa pregunta algo y el usuario responde; la salida es cuando el programa muestra información.

- Entrada de datos: Cuando el programa pregunta algo y tú escribes la respuesta. En Julia, usamos readline() para leer algo que escribimos.
- Salida de datos: Es cuando el programa muestra información en la pantalla, usando println().

Por ejemplo, print("¿Cuál es tu color favorito?: "), nos mostrara en pantalla el mensaje indicado y luego podemos leer el color con color = readline().

#### 7. Condicionales:

Los condicionales son decisiones que el programa toma según una pregunta de "¿Es verdadero o falso?". Permiten que el programa haga cosas diferentes según la situación. Usamos if, elseif y else para decidir qué hacer. Los podemos ver como si es verdadero (if) se hará esto; sino verfica si esto es verdad, de ser así haz esto (elseif), de no ser nada haz esto (else). Por ejemplo:

**edad = 12** 

if edad < 13

```
println("Eres un niño.")
elseif edad < 18
  println("Eres un adolescente.")
else
  println("Eres un adulto.")
end</pre>
```

## 8. Ciclos:

Son instrucciones que hacen que el programa repita una tarea varias veces. Nos ayudan a hacer cosas repetitivas de forma rápida y sencilla.

• For: Se usa para repetir algo un número específico de veces. Por ejemplo:

```
for i in 1:5

println("Esto es divertido!")

end
```

 While: Se usa para repetir algo mientras una condición sea verdadera. Por ejemplo:

```
respuesta = "si"

while respuesta == "si"

println("Repito una y otra vez")

println("Quiere repetir?: ")

respuesta = readline()
```

end

## 9. Funciones:

Son bloques de código que hacen una tarea específica. Se les da un nombre y se pueden usar cada vez que sea necesario. Ayudan a organizar el código y a no repetir cosas, haciendo el programa más fácil de leer y de usar. **Por ejemplo:** 

```
function saludar(nombre)

println("Hola " + nombre + " !")

end

saludar("Victoria")
```

## 10. Importar funciones y librerías:

Julia tiene paquetes y librerías especiales que agregan habilidades nuevas. Los paquetes son como cajas de herramientas. Para usar un paquete, lo traemos al programa usando "using". Por ejemplo, using Random para traer funciones que nos permiten crear números al azar, como rand().

## Profundiza y Refuerza lo Aprendido:

En los siguientes enlaces podrás conseguir dos divertidos ejemplos donde podrás revisar todo lo aprendido en esta documentación:

- Calculadora de años humanos de mascota: https://github.com/devzelix/Documentacion-Julia/blob/main/Julia/src/calculadora agnos.jl, es un programa donde podremos calcular la edad de tu mascota en años humanos, cabe destacar que acá utilizamos dos nuevos conocimientos como son: "," se utiliza para concatenar los textos con cualquier otro tipo de dato; y la función parse(), que solicita dos parámetros, el primero será el tipo de dato al que queremos convertir el dato que pasemos como el segundo parámetro.
- Calculadora de raíces de una ecuación cuadrática:
   https://github.com/devzelix/Documentacion Julia/blob/main/Julia/src/calculadora\_raices.jl, es un programa donde
   podremos calcular las raíces reales de una ecuación cuadrática, cabe
   destacar que aquí también utilizamos un nuevo conocimiento como la función
   sqrt(), que solicita un único parámetro y nos retornara la raíz cuadrada del
   número que le pasemos.

Para profundizar más acerca de julia puedes ir directamente a la documentación de oficial, pero toma en cuenta que no será tan fácil de entender, ya que es para un nivel más avanzado y convertirte en un experto, para disfrutar de aventuras más grandes: https://docs.julialang.org/en/v1/.