

## Modulo 1. Primeros pasos: contextualización

Fuentes:

- [Julia \(programming language\) - Wikipedia](#)
- [Noteworthy Differences from other Languages · The Julia Language](#)

### 1 Descripción general y características del lenguaje

El lenguaje de programación Julia surge como alternativa para los problemas relacionados a la computación científica, tales como la cobertura de operaciones y las expresiones matemáticas, así como los tiempos de ejecución necesarios para grandes volúmenes de datos o algoritmos de alta complejidad. Es una opción viable para considerar la cual que provee según, sus desarrolladores, de un ambiente productivo y eficiente para el prototipado y puesta en marcha de aplicaciones de alto desempeño como por ejemplo trabajos con inteligencia artificial, modelos matemáticos o análisis de ecuaciones complejas.

Se ubica en la categoría de lenguajes de alto nivel, es decir, brinda una gran capa de abstracción frente a un gran conjunto de operaciones con el fin de que pueda ser fácilmente utilizable y reproducible entre sus usuarios. También es un lenguaje dinámico y haciendo referencia a su documentación oficial ello significa *“que no se le puede indicar al compilador sobre el tipo de dato de algún valor”*, en otras palabras, al momento de codificar no se necesita especificar el tipo de dato a utilizar. Esta última característica es posible gracias al proceso de compilación del lenguaje el cual consiste en convertir el código de alto nivel (escrito por un humano) a lenguaje de bajo nivel (lo que se conoce como código de máquina) para que el computador pueda entender las instrucciones.

Otro conjunto de características importantes y que permiten hacerse a una idea de lo robusto que es el lenguaje son:

- Es gratuito y Open source (disponible para todo el mundo con posibilidad de adaptaciones/modificaciones)
- Al tener la computación científica como tópico principal posee librerías que tratan temas como cálculos de punto flotante (por ejemplo, la precisión), Álgebra lineal, Transformadas rápidas de Fourier y ecuaciones diferenciales en diferentes categorías, visualización de datos, cálculos relacionados a la biología y programación molecular como comportamiento de especies, computación de moléculas, entre otras.
- Diseñado para procesos de paralelismo y computación distribuida. De manera general, cada término se refiere a lo siguiente:
  - **Paralelismo:** Se refiere al suceso que una CPU pueda ejecutar varias tareas a la vez para poder resolver uno o más problemas.

- **Computación distribuida:** Se basa en el uso de múltiples computadoras conectadas entre sí mediante una infraestructura de red como lo puede ser Internet con el objetivo de resolver problemas de computación tales como aquellos relacionados con búsqueda de elementos y soluciones, optimizaciones o toma de decisiones.
- Es multiparadigma ya que integra características de la programación imperativa, funcional, y orientada a objetos. Cada uno de ellos puede ser descrito de la siguiente manera:
  - **Imperativa:** Como hacer las cosas especificando los pasos para llegar al resultado.
  - **Funcional:** Como hacer las cosas especificando el comportamiento general del programa mediante el uso de funciones.
  - **Orientado a objetos:** Abstracción de la información que permite modelar un objeto de manera virtual.
- Maneja su propio recolector de basura. Como ya se conoce, los computadores funcionan con memorias que les permiten almacenar la información, específicamente los programas que usamos diariamente están diseñados para usar cierto porcentaje de esta para llevar a cabo sus actividades.

Los programadores detrás de dichas aplicaciones deben de ser cuidadosos con el manejo de la memoria, de otra forma podría resultar en que un programa consuma la mayor parte de esta disponible dejando así a otros programas sin capacidad para ejecutarse. Es por lo anterior que surge el termino de recolector de basura refiriéndose al proceso de manejo de la memoria por parte del lenguaje.

- Es de fácil manejo. Gracias a su característica de lenguaje de alto nivel este presenta operaciones con grandes niveles de expresividad, esto se debe gracias a que se basa en otros lenguajes con algunas características similares como Python, R, Matlab, PERL, LISP, Algol, Dylan, y Fortress.
- Es usado para la programación de sistemas de bajo nivel (sistemas operativos, Software as a Service, procesos de automatización industrial y motores de videojuegos), descripción de hardware de alto nivel (como la manipulación de FPGAs), programación web tanto para el lado del servidor como para el lado del cliente
- Durante la práctica del lenguaje, permite de una manera muy simple la interoperabilidad con otros lenguajes, por ejemplo, mediante la palabra clave `ccall` se logra usar funciones independientes de librerías de C o Fortran, o como sucede con la creación de paquetes que necesiten de funciones de Python donde se necesitaría usar otro paquetes como `pyjulia` y `PythonCall.jl/juliacall`.
- Julia es [rápido](#). Ha sido el cuarto lenguaje después de C, C++ y Fortran en funcionar en las supercomputadoras alcanzando desempeños en escalas de Petaflops. Esto es posible gracias a que está internamente diseñado para cumplir con necesidades de rapidez frente a diferentes situaciones de complejidad, por otra parte, su

característica de **múltiple-despacho** permite generar código adaptativo, es decir, una función puede ser interpretada en diferentes ambientes de ejecución (por ejemplo el uso de tarjetas gráficas para el machine learning).

## 2 Usos de Julia en la industria

Este lenguaje ha sido adoptado por el campo académico con universidades como el MIT, Stanford y la universidad de Berkley. En el ambiente gubernamental por agentes como la NASA o la FAA (Administración de aviación federal) y en entornos de industrias tecnológicas como Amazon, IBM o JPMorgan.

Algunos proyectos que involucran a Julia han sido:

### 1) Computación científica

- La alianza del modelamiento del clima (CliMA) usó Julia para implementar su nuevo modelo global del clima para proveer de nueva información sobre los cambios y retos en los cambios climáticos ([JuliaHub, 2019](#)).
- La NASA realizó varios proyectos, entre ellos:
  - Dinámica de separación de naves espaciales.
  - Analizar conjuntos de datos sobre exoplanetas alrededor de TRAPPIST-1, específicamente datos como masa, radio y densidad en altas precisiones fraccionadas para luego ser comparadas con la tierra y poder ser clasificadas.
  - Analizar la información sobre microondas cósmicas del Big Bang ([JuliaCon, 2022](#))
- El Instituto Nacional de Brasil para la Búsqueda Espacial (INPE por sus siglas en inglés) planea misiones espaciales y simula satélites in Julia por medio de su herramienta y paquete de Julia SatelliteToolBox.jl .
- El CERN (Organización Europea para la Búsqueda Nuclear en Francés) usa Julia en el gran colisionador de hadrones (LHCb experiment) para el análisis de resultados durante su ejecución y preparación de ambientes de desarrollo (herramientas como [UnRoot.jl](#), [lgcmake.jl](#)).
- La herramienta [Celeste.jl](#) caracteriza estrellas y galaxias en imágenes astronómicas

### 2) Otros usos notables

- Creación de modelos económicos por parte del banco de Reserve Federal de los Estados Unidos en 2015. Sus más recientes versiones incluyen los efectos del COVID-19 para el 2021.
- La empresa BlackRock analiza series de tiempo.
- Cálculo de riesgos de seguros y otros indicadores por parte de la empresa de seguros británica Aviva.
- Verificación de software