

CURSO-TALLER

Introducción a Julia



Eficiencia de alto nivel
Un acercamiento al lenguaje de programación Julia





Introduction

Expositores y contactos



Óscar Anuar Alvarado Morán

Soy egresado de la licenciatura en **física** y reciente egresado de la licenciatura en **ciencia de datos**

Me entusiasmé con Julia por la **facilidad** de sintaxis que ofrece a los no programadores y por todo el **poder** que ofrece una vez que te adentras más en la programación.

Me interesa la idea de **agregar gente** a este culto y **colaborar** con la comunidad científica teniendo a Julia en común.





Mario Horacio Garrido Czacki

Soy reciente egresado de la licenciatura en **ciencia de datos** y estoy por concluir **ingeniería en computación**.

Me entusiasmé con Julia por la promesa central de **simplicidad de sintaxis** y **alto desempeño**. Poder generar código eficiente mediante una sintaxis clara y concisa permite centrarse en el algoritmo y no la implementación.

Me interesa utilizar Julia para desarrollar proyectos de ciencia de datos y ponerlos en producción con el **mismo código**.





Oscar Alejandro Esquivel Flores

Soy investigador Asociado C en el departamento ISCA-IIMAS de la UNAM y mi trabajo se orienta al cómputo **científico**, cómputo **paralelo**, y el desarrollo de algoritmos **iterativos**.

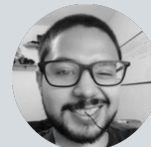
Me entusiasmé con Julia por la comunidad que labora el **desarrollo**, **difusión** y **enseñanza** del lenguaje y por la mezcla ideal de **eficiencia** y **facilidad** de Julia.

Me interesa fomentar el uso de Julia en nuestra comunidad **académica** y utilizarlo como una herramienta para hacer **investigación**.





mario.garrido@iimas.unam.mx



OscarAlvarado@ciencias.unam.mx



oscar.esquivel@iimas.unam.mx



taller.juliaiimas@gmail.com



Julia-IIMAS v1.0



Julia programming language

Código eficiente y expresivo



“If we had one language
that was simultaneously
easy to use and fast, you
could just unleash
innovation across the
world.”

Viral Shah, CEO of Julia Computing and co-creator of
the Julia programming language





Cómputo Científico



Desde su concepción Julia fue diseñado para ofrecer **alto rendimiento**. Los programas de Julia se compilan en código nativo eficiente para múltiples plataformas a través de **LLVM**

Aprendizaje Automatizado



El paquete **MLJ.jl** proporciona una interfaz unificada para algoritmos de aprendizaje automático comunes, que incluyen modelos lineales generalizados, árboles de decisión y agrupación en clústeres. **Flux.jl** y **Knet.jl** son potentes paquetes para **Deep Learning**.



Ciencia de datos



El ecosistema de datos de Julia proporciona **DataFrames.jl** para manejo de conjuntos de datos. **CSV.jl** es un paquete rápido para leer archivos csv. Los cálculos sobre flujo de datos se pueden realizar con **OnlineStats.jl**.

Cómputo paralelo



Julia está diseñado para el paralelismo y proporciona primitivas integradas para la computación paralela en todos los niveles: **paralelismo a nivel de instrucción, multi-procesamiento, cómputo con GPUs y computación distribuida.**



Julians

Creadores y comunidad Juliana



Alan Edelman



Profesor del MIT, co-creador de Julia en el laboratorio de ciencias de la computación e inteligencia artificial del MIT (CSAIL).

“The release of 1.0 says that Julia is now ready to change the technical world by combining the high-level productivity and ease of use of Python and R with the lightning-fast speed of C++.”

Stefan Karpinsky



Co-creador de Julia y co-fundador de Julia computing, Inc.

“This would become Julia, a new programming language that aims to change the face of data science and scientific and numerical programming”



Viral B. Shah



Co-creador de Julia y CEO de Julia computing, Inc.

“Simplicity meets speed. We make Julia easy to use, easy to deploy and easy to scale”

Jeff Bezanson



Co-creador de Julia y co-fundador de Julia computing, Inc.

“An equilibrium point in the design space can be found by combining selection and specialization into a single appropriately designed dynamic type-based dispatch mechanism”



Tim Besard



Ingeniero de Software en Julia Computing, Inc. Su trabajo se enfoca en las tecnologías de compiladores y es responsable del desarrollo del “backend” **CUDA-GPU** para Julia. PhD en ciencia e ingeniería de la computación, por la universidad de Ghent, Bélgica.

@maleadt

Huda Nassar



Es posdoctorante en el laboratorio Aghaeepour en la Universidad de Stanford. Es una usuaria entusiasta de Julia y autora del paquete de algoritmos gráficos **MatrixNetworks.jl**.

@nassarhuda



Matt Bauman



Investigador científico senior en Julia Computing, Inc. en Chicago, donde emplea su tiempo trabajando en arreglos y **broadcasting**. Ha contribuido al desarrollo del lenguaje Julia y al diseño de múltiples paquetes.

@maleadt

Jane Herriman



Directora de **difusión** y diversidad en Julia Computing y estudiante de doctorado en el tecnológico de California, Caltech. Es **instructora** y entusias activa de Julia.

@JaneHerriman



Valentin Churavy



Trabaja en el CSAIL-MIT en aspectos de **cómputo heterogéneo** con Julia, contribuye a las investigaciones del grupo Julia Lab en aspectos teóricos y numéricos.

@vchuravy

Christopher Rackauckas



Es instructor de Matemáticas en el MIT, director de modelado y simulación en Julia Computing y creador/desarrollador principal de **JuliaSim**. Director de Investigación Científica en **Pumas-AI** y creador/desarrollador líder de **Pumas**. Desarrollador principal de la organización de software de código abierto **SciML**.

@ChrisRackauckas



David Sanders



Profesor titular en el departamento de física de la **FC-UNAM**. Lleva a cabo una estancia sabática en el Stata Center en el MIT. Entusiasta de Julia, participa activamente en la difusión y enseñanza del lenguaje. Ha desarrollado los paquetes **validatednumerics.jl** e **intervalconstraintprogramming.jl**

@DavidPSanders



Resources

Comunidad, aprendizaje, contactos

Comunidad



<https://www.youtube.com/user/JuliaLanguage>



<https://github.com/JuliaLang>



<https://discourse.julialang.org/>



[@JuliaLanguage](#)

Tutoriales



Introducción a Julia



Cómputo paralelo
Julia Academy



Introducción a Julia
Optimización de código
Data Science



Aprendizaje automatizado
Ecuaciones Diferenciales
Modelos y simulación



Introducción a Julia
Despacho múltiple
Metaprogramación
Optimización por intervalos
Computational thinking MIT



Programación GPUs



Productos

Julia
computing

<https://juliacomputing.com>

JuliaCon



<https://juliacon.org/>



Schedule

Planificación del curso y contenidos

Lun 02/08/2020

Mar 03/08/2020



Introducción

- Presentación del curso
- Gente y comunidad Juliana
- Microbenchmarks

Elementos básicos

- Ambientes de desarrollo
- Instalación y pkg
- Primeras líneas de código

Introducción al Despacho múltiple



Despacho múltiple

- Estructuras
- Métodos
- Estructuras definidas por el usuario
- Estructuras parametrizadas

Aceleración de código

- Recetas de aceleración de código



Visualización

- Backends
- Gr, Plotly
- Statsplot
- Visualizaciones dinámicas

Mie 04/08/2020



Aprendizaje Automatizado

Manejo de datos

A. supervisado y A. no supervisado

A. profundo

Aplicaciones generales

Álgebra lineal

Optimización

ODE's

Jue 05/08/2020



Introducción al paralelismo

Corutinas

Tareas

Canales

Multiprocesamiento

Cómputo distribuido

Vie 06/08/2020



GPUs

CUDA-C

CUDA.jl

Caso práctico

Optimización de una cadena
de producción usando PSO

Paralelización con GPUs





Greatfulness

Reconocimientos y agradecimientos



Universidad Nacional Autónoma de México

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Coordinación de Posgrado en Ciencia e Ingeniería de la Computación
Coordinación de la licenciatura en Ciencia de Datos



Dirección General de Asuntos de Personal Académico

Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica

Agradecemos el apoyo al proyecto PAPIIT-IA104720



**“ ‘People say that Bangalore
(Bengaluru) only has IT coolies
and no one does anything
meaningful’
Shah said, pointing out that
nothing could be further from
the truth”**

“Not just “IT coolies”: An Indian computer scientist helped build
an alternative programming language in Bengaluruon coolies IT”

By Maria Thomas
QUARTZ, INDIA