

**Diploma de
Matemática
ANEP UdelaR**

Comité Académico

Formulario de Aprobación Curso de Posgrado

Asignatura: Computación Científica en Julia

Núcleo:

Profesor de la asignatura: Dr. Ing. Marcelo Forets (Gr. 2)
(título, nombre, grado o cargo, Instituto o Institución)

Instituto o Unidad: Departamento de Matemática y Aplicaciones, CURE, Universidad de la República

Departamento o Área:

Profesor de la asignatura: Melina Colombo (Gr. 1)
(título, nombre, grado o cargo, Instituto o Institución)

Instituto o Unidad: Departamento de Matemática y Aplicaciones, CURE, Universidad de la República

Departamento o Área:

Fecha de inicio y finalización: primer semestre 2022 según calendario CURE

Horario y Salón:

Horas Presenciales: 68.5

(se deberán discriminar las mismas en el ítem Metodología de enseñanza)

Nº de Créditos: 8

(Un crédito equivale a 15 horas de dedicación del estudiante según se detalla en el ítem metodología de la enseñanza)

Público objetivo y Cupos: Egresados de formación docente, profesores de educación secundaria. Estudiantes y egresados de carreras universitarias de Ciencias e Ingeniería.

Cupo máximo 20 estudiantes.

(si corresponde, se indicará el número de plazas, mínimo y máximo y los criterios de selección. Si no existe indicación particular para el cupo máximo, el criterio general será el orden de inscripción en el Depto. de Postgrado, hasta completar el cupo asignado)

Objetivos: Presentar los conceptos básicos de la computación científica moderna mediante el lenguaje de programación Julia y su ecosistema de código abierto. Introducir herramientas para la generación de códigos mantenibles, eficientes y reproducibles en investigación y enseñanza..

Conocimientos previos exigidos: Conocimientos de programación.

Conocimientos previos recomendados: Dominio de un lenguaje de programación de alto nivel (ejemplos: GNU-Octave, MATLAB, Python, R, Julia). Nociones de métodos numéricos.

Metodología de enseñanza: Exposiciones teórico-prácticas y trabajo en clase en la aplicación de los conceptos presentados. Las clases serán principalmente en formato virtual. La defensa del proyecto será presencial. Para realizar el curso es necesario disponer de una computadora con conexión a internet.

(comprende una descripción de las horas dedicadas por el estudiante a la asignatura y su distribución en horas presenciales -de clase práctica, teórico, laboratorio, consulta, etc.- y no presenciales de trabajo personal del estudiante)

- Horas clase (teórico): 45
- Horas clase (práctico): 23
- Horas clase (laboratorio): 0
- Horas consulta: 2
- Horas evaluación: 2
- Subtotal horas presenciales: 72
- Horas estudio: 15
- Horas resolución ejercicios/prácticos: 20
- Horas proyecto final/monografía: 13
- Total de horas de dedicación del estudiante: 120

Forma de evaluación: La evaluación consta de (i) entregables a modo de evaluación continua y (ii) un trabajo final. El proyecto consistirá en desarrollar un paquete de Julia para la resolución de un problema de computación científica de interés para el/la estudiante. El estudiante defenderá un anteproyecto en un plazo límite establecido al inicio del curso. El proyecto completo, acompañado del código fuente y la documentación correspondiente, se defenderá en una exposición oral al final del curso.

Temario:

1. **Herramientas básicas.** Alcance de la computación científica. Revisión de lenguajes de alto nivel vs. lenguajes de programación de sistemas. Revisión de aritmética de punto flotante. Conceptos básicos de Julia. Entorno de desarrollo.
 2. **Fundamentos de Julia.** Tipos de datos básicos. Variables. Arreglos. Módulos. Gestor de paquetes. Estructuras de datos. Gráficos en 2D y 3D. Lectura y escritura de archivos. Tipos concretos y abstractos. Tipos paramétricos. Despacho múltiple (multiple dispatch). Diseño de funciones. Evaluación en el sitio (in-place). Evaluación perezosa (lazy). Herramienta git para control de revisión.
 3. **Algoritmos.** Métodos directos para resolución de sistemas lineales. Métodos iterativos. Integración numérica. Matrices estructuradas. Análisis de errores y estabilidad de algoritmos numéricos. Factorización SVD y QR. Matrices esparsas. Recursión. Algoritmos en grafos.
 4. **Ecosistema de Julia para la computación científica.** Conceptos de desarrollo open source. Conceptos avanzados de gráficos y generación de gráficos para publicaciones. Paquetes para trabajar con data frames y aprendizaje estadístico. Paquetes para resolver sistemas dinámicos, e.g. ecuaciones
-

diferenciales ordinarias y ecuaciones en derivadas parciales. Paquetes para modelado y resolución de problemas de optimización matemática. *Nota: los contenidos de esta parte son indicativos y se contempla adaptarlos a los intereses de los/las estudiantes.*

Bibliografía:

- *Hands-On Design Patterns and Best Practices with Julia: Proven solutions to common problems in software design for Julia 1. x.* Kwong, T. Packt Publishing Ltd, 2020. ISBN: 9781838648817
 - *Fundamentals of Numerical Computation (Vol. 154).* Driscoll, T. A., & Braun, R. J. SIAM 2017. ISBN: 9781611975079
 - *Algorithms for Optimization.* Kochenderfer, M and Wheeler, T. A. 2019. ISBN: 9780262039420
 - *Pro git.* Chacon, S. and Straub, B. Springer Nature, 2014. ISBN: 9781430218333
 - *Principles of scientific computing.* Bindel, D. and Goodman, J. New York University, 2009.
 - *Statistics with Julia.* Nazarathy, Yoni, and Hayden Klok. 2021. 2365-5682
 - *Numerical Linear Algebra with Julia.* Darve E, Wootters M. SIAM; 2021. Darve E, Wootters M. ISBN: 9781611976540
-