

**Formulario de aprobación de curso de
posgrado/educación permanente**

Asignatura: Computación Científica en Julia

(Si el nombre contiene siglas deberán ser aclaradas)

Modalidad:

(posgrado, educación permanente o ambas)

Posgrado

Educación permanente

Profesor de la asignatura ¹:

Dr. Ing. Marcelo Forets, Asistente, Departamento de Matemática y Aplicaciones, Centro Universitario Regional del Este, Universidad de la República

Dr. Daniel Freire Caporale, Asistente, Instituto de Física, Facultad de Ciencias, Universidad de la República.

(título, nombre, grado o cargo, instituto o institución)

Profesor Responsable Local ¹: Dr. Agustín Laguarda, Asistente, Instituto de Física, Facultad de Ingeniería, Universidad de la República

(título, nombre, grado, instituto)

Otros docentes de la Facultad:

(título, nombre, grado, instituto)

Docentes fuera de Facultad:

(título, nombre, cargo, institución, país)

¹ Adjuntar CV si el curso se dicta por primera vez. Se adjuntan CV de los profesores.

(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

[Si es curso de posgrado]

Programa(s) de posgrado: Estudiantes de posgrado en Ingeniería Física, Ingeniería de la Energía, Ingeniería Eléctrica e Ingeniería Mecánica.

Instituto o unidad: Instituto de Física

Departamento o área: -

Horas Presenciales: 38

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

Nº de Créditos: 8

[Exclusivamente para curso de posgrado]

(de acuerdo a la definición de la UdelaR, un crédito equivale a 15 horas de dedicación del estudiante según se detalla en el ítem Metodología de enseñanza)

Público objetivo: El curso está dirigido a estudiantes de posgrado y/o profesionales egresados de carreras Ingeniería Mecánica, Eléctrica, Civil, Hidráulica o Computación. Egresados de la Licenciatura en Física. Egresados de la Licenciatura en Ciencias de la Atmósfera. Estudiantes de grado avanzados que acrediten los conocimientos básicos requeridos.

Cupos: Máximo 20

Objetivos: Presentar los conceptos básicos de la computación científica moderna mediante el lenguaje de programación Julia y su ecosistema de código abierto. Brindar nociones y patrones de diseño que faciliten el uso de herramientas computacionales para la generación de códigos mantenibles, eficientes y reproducibles.

Conocimientos previos exigidos: Nociones básicas de Métodos Numéricos. Nociones básicas de programación.

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

Metodología de enseñanza: Exposiciones teórico-prácticas y trabajo en clase en la aplicación de los conceptos presentados, utilizando canales de comunicación en línea (por ejemplo Zoom, Zulip) y poniendo a disposición materiales de clase en Eva y/o gitlab-fing (como ser Jupyter notebooks, entre otros). Las clases serán principalmente en formato virtual, que se complementarán con clases presenciales para atender consultas y para la defensa del proyecto.

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

Detalle de horas:

- Horas de clase (teórico): 22
- Horas de clase (práctico): 10
- Horas de clase (laboratorio): 0
- Horas de consulta: 4
- Horas de evaluación: 2
 - o Subtotal de horas presenciales: 38
- Horas de estudio: 22
- Horas de resolución de ejercicios/prácticos: 15
- Horas proyecto final/monografía: 45
 - o Total de horas de dedicación del estudiante: 120

Forma de evaluación: La evaluación consta de un trabajo final del curso enfocado al desarrollo de 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. Finalmente, el proyecto completo, acompañado del código fuente y la documentación correspondiente, se defenderá en una exposición oral abierta a otros estudiantes del curso con preguntas de los docentes.

[Indique la forma de evaluación para estudiantes de posgrado, si corresponde]

[Indique la forma de evaluación para estudiantes de educación permanente, si corresponde]

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. Herramienta *git* para control de revisión. Conceptos de desarrollo *open source*. Patrones de diseño de *software* en la computación científica.

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*). Interfaces. Macros. Metaprogramación y ejemplos de lenguajes específicos de dominio (DSL).

3. **Patrones de diseño:** Generación de paquetes. Patrones para reutilización de código. Generación de tests. Documentación de código. Integración continua. Patrones para código de alto desempeño (*performance*). Análisis de desempeño (*profiling*). Depuración de código. Patrones de robustez. Programación genérica. Programación paralela. Anti-patrones.

4. **Ecosistema de Julia para computación científica:** Conceptos avanzados de gráficos y generación de gráficos para publicaciones. Paquetes para trabajar con datos, como 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

Datos del curso

Fecha de inicio y finalización: Inicio tentativo 1 de agosto de 2022, finalización de clases tentativa 17 de octubre de 2022.

Horario y Salón: A definir.

Arancel: No corresponde

[Si la modalidad no corresponde indique "no corresponde". Si el curso contempla otorgar becas, indíquelo]

Arancel para estudiantes inscriptos en la modalidad posgrado: No corresponde

Arancel para estudiantes inscriptos en la modalidad educación permanente: UI 1500 (mil quinientas unidades indexadas).
