



Universidad Nacional de Rosario
Facultad de Ciencias Económicas y Estadística

Programación 1

Carreras: Lic. en Estadística y Lic. en Ciencia de Datos

Tipo de Materia: Obligatoria

Ciclo: Formación Técnica

Año: 1°

Escuela: Estadística

Duración: Cuatrimestral

Carga Horaria: 64 horas

Profesor Titular: Marcos Miguel Prunello

FUNDAMENTACIÓN

El desarrollo de la Estadística moderna y el avance de la Ciencia de Datos ha generado una creciente necesidad de dominar herramientas computacionales para el procesamiento eficiente de datos. En este contexto, la programación es una competencia fundamental para cualquier profesional en estas disciplinas, ya que permite automatizar procesos, implementar modelos estadísticos y desarrollar soluciones personalizadas para el análisis de datos en distintos ámbitos de aplicación.

La asignatura Programación 1 introduce a los estudiantes en el pensamiento algorítmico y en la resolución estructurada de problemas computacionales. Su propósito es brindar una base en programación que facilite el desarrollo de habilidades fundamentales para el posterior trabajo con datos. Se focaliza en el lenguaje R, ampliamente utilizado en la comunidad estadística y de ciencia de datos, debido a su flexibilidad, riqueza de librerías y potencia para el análisis y visualización de datos.

Dado que se dicta en el primer cuatrimestre del primer año, esta asignatura cumple un rol clave en la formación inicial de los estudiantes. Se ubica dentro del área de Fundamentos Computacionales y pertenece al Ciclo de Formación Técnica, proporcionando los conocimientos necesarios para afrontar cursos más avanzados como Programación 2. Además, sienta las bases informáticas para asignaturas posteriores relacionadas con la gestión, análisis y modelado de datos.

La asignatura no sólo enfatiza los aspectos técnicos de la programación, sino que también promueve buenas prácticas de desarrollo, como la organización eficiente del código, el uso de la terminal y la creación de funciones modulares para mejorar la reutilización del código. Estas competencias resultan esenciales para que los futuros profesionales puedan enfrentarse con éxito a los desafíos del manejo y análisis de datos en diversos contextos, desde la industria hasta la investigación académica.

OBJETIVOS

Que los estudiantes logren:

- analizar problemas computacionales y formular soluciones algorítmicas;
- evaluar salidas computacionales y depurar código de programación con errores;
- manejar expresiones simbólicas para formalizar operaciones lógicas que puedan ser evaluadas por la computadora;
- identificar distintos tipos de representaciones de datos y las estructuras abstractas que permiten su manipulación;
- aplicar distintas estructuras de control de flujo de código para administrar el orden en el que el mismo es evaluado;
- emplear el principio de la descomposición algorítmica para resolver problemas grandes y complejos al dividirlos en partes de menor extensión, a través del uso de funciones;
- integrar buenas prácticas de programación, como el uso de guías de estilo y la documentación del código; y
- usar la terminal para ejecutar programas.

CONTENIDOS

Unidad 1. Introducción a la programación con R

Problemas y algoritmos. Lenguajes de programación. Errores y depuración. Lenguaje R y su ecosistema. Interfaz RStudio. Ambiente y objetos. Tipo de datos elementales. Operadores aritméticos, relacionales y lógicos. Buenas prácticas en la escritura de código. Organización del trabajo en proyectos.

Unidad 2. Estructuras de control

Secuencias y flujo de ejecución. Condicionales simples, dobles y anidadas. Bucles iterativos (“para”, “mientras que”, “hasta que”).

Unidad 3. Descomposición algorítmica

Principios de descomposición y refinamiento sucesivo. Creación de funciones en R. Paso de parámetros y ámbito de variables. Transparencia referencial y documentación de funciones.

Unidad 4. Uso de la terminal

Introducción a la terminal. Comandos básicos para la gestión de archivos. Ejecución de programas desde la terminal. Elaboración de programas para la interacción con el usuario desde la terminal.

Unidad 5. Estructuras de datos

Vectores, matrices y listas en R. Indexación y manipulación de estructuras. Operaciones vectorizadas.

Unidad 6. Desarrollo de librerías de software

Integración de conceptos mediante la creación de paquetes en R.

CARGA HORARIA

La asignatura cuenta con una carga horaria de 64 horas, divididas en 2 clases semanales de 2 horas de duración, una teórica y otra práctica.

METODOLOGÍA DE ENSEÑANZA

Las clases teóricas tienen el objetivo de presentar los contenidos conceptuales de cada unidad, mediante la exposición interactiva del tema a abordar. Se utilizan presentaciones y demostraciones de escritura de código en vivo. Las clases prácticas se dictan en el Laboratorio de Computación de la Escuela de Estadística. En las mismas, los estudiantes son guiados por los docentes para resolver una serie de actividades prácticas que permiten aplicar los conceptos desarrollados. La resolución de estos problemas requiere elaborar una estrategia algorítmica y su correspondiente implementación en el lenguaje de programación estadística R. Se incentiva el trabajo colaborativo entre pares dentro del aula, bajo la modalidad de taller. Finalmente, se complementan las actividades prácticas con ejercitación propuesta para ser resuelta fuera de clase y se publican sus soluciones, favoreciendo el desarrollo de un sistema de autoevaluación continua.

El contenido teórico, los ejemplos de aplicación, las prácticas, sus soluciones y todo el material didáctico empleado forma parte de un libro digital de referencia para el curso, desarrollado por los docentes y publicado de manera abierta en internet [en la página web de la asignatura](#).

ACTIVIDADES

Ejercitación

Los contenidos conceptuales de cada unidad están acompañados por una selección de ejercicios prácticos para facilitar su comprensión y ejemplificar su aplicación. Algunos de estos ejercicios son abordados en conjunto entre los docentes y los estudiantes en el contexto de la clase, mientras que otros se dejan propuestos para la resolución autónoma de los estudiantes. Cada ejercicio describe un problema cuya resolución requiere el desarrollo de un programa en el lenguaje R. Los ejercicios van aumentando su complejidad en el transcurso del curso, iniciando por tareas sencillas que permitan a los estudiantes familiarizarse con las reglas de escritura de código, hasta llegar a problemas más elaborados donde se debe hacer foco en el desarrollo de un algoritmo creativo para llegar a la solución.

Trabajo práctico y tarea individual

Durante la cursada los estudiantes deben resolver un trabajo práctico de elaboración grupal para abordar un problema de mayor complejidad que requiera de debate entre pares y de tiempo de trabajo fuera del aula. Por otro lado, tal como se indica más adelante, los estudiantes que aspiran a la promoción de la asignatura deben completar una tarea de programación de carácter individual en el tramo final del cuatrimestre. Ambas actividades tienen el objetivo de integrar los contenidos de las distintas unidades y de poner en práctica la creación de programas y el seguimiento de convenciones de estilo.

Actividades autoevaluativas

A lo largo del cursado se ponen a disposición [en la página web de la asignatura](#) actividades autoevaluativas no obligatorias del tipo de opciones múltiples, con soluciones disponibles de forma inmediata, para que los estudiantes puedan realizar una evaluación personal de su avance.

Foros y clases de consulta

En el aula virtual de la asignatura dentro de la plataforma Comunidades 3 se presentan múltiples foros de consultas, que permiten abordar la resolución de dudas de forma colaborativa sobre contenidos conceptuales, ejercicios prácticos y demás aspectos relacionados con la dinámica del cursado. Asimismo se ofrecen clases de consulta.

SISTEMA DE EVALUACIÓN

La evaluación de la asignatura se realiza a través de un examen parcial individual y un trabajo práctico grupal. Ambas actividades son evaluadas con una calificación entre 0 y 10 y se ponderan para obtener una nota de cursada (60% examen parcial, 40% trabajo práctico).

Condición de los estudiantes al finalizar el cursado

- Libre: obtienen la condición de “libre” aquellos estudiantes con nota de cursada inferior a 6 o alguna nota individual menor a 4.
- Regular: obtienen la condición de “regular” aquellos estudiantes con nota de cursada mayor o igual a 6, siempre que no tengan ninguna nota individual menor a 4.
- Promovido: obtienen la condición de “promovido” aquellos estudiantes que con nota de cursada mayor o igual a 8, que no tengan ninguna nota individual menor a 6 y que aprueben una tarea de realización opcional e individual al final del cuatrimestre. La nota de cursada se computa como nota final de la asignatura.

Examen parcial

El examen parcial es de resolución individual y presencial, sin uso de computadora, a carpeta abierta. Incluye preguntas conceptuales y ejercicios prácticos relacionados a la escritura e interpretación de código en R y a la solución algorítmica de problemas de corta extensión, incluyendo preguntas abiertas; opciones múltiples; verdadero o falso; completar, corregir o interpretar código o capturas de pantalla; etc. Como criterio de evaluación se prioriza la integración y relación entre conceptos, la identificación de las herramientas prácticas aplicables a la resolución de problemas cortos y la obtención de respuestas correctas. Quienes obtienen una nota inferior a 6 pueden acceder a un examen recuperatorio de iguales características.

Trabajo práctico

El trabajo práctico es de entrega grupal y de realización fuera de clase. Permite abordar problemas de programación que integran los contenidos de las primeras unidades temáticas y cuya resolución requiere un mayor tiempo de debate y trabajo entre los integrantes de los equipos. Se evalúa teniendo en cuenta los siguientes componentes: diseño del algoritmo (si la solución es correcta, cumple el objetivo, excluye casos particulares o es general, etc.), la ejecución del programa (corre correctamente, produce errores, funciona sólo para algunos casos, etc.) y estilo (claridad, prolijidad, formato, uso apropiado de comentarios, sangrías, etc.). Los estudiantes disponen de al menos tres semanas para su resolución y entrega, período durante el cual pueden recurrir a la colaboración de los docentes para alcanzar los objetivos propuestos por el trabajo.

Tarea individual

La tarea individual debe ser completada opcionalmente por los estudiantes que estén en condiciones de alcanzar la promoción. Consiste en una tarea de programación breve a completar en el tramo final del cuatrimestre, que permite abordar de forma integral todos los contenidos de la asignatura mediante el desarrollo de una librería de software. Los criterios de evaluación coinciden con aquellos establecidos para el trabajo práctico.

Exámenes finales

El examen final regular se realiza en el Laboratorio de Computación de la Escuela de Estadística con una prueba de programación en R, mientras que el examen final libre incluye además una instancia escrita de características semejantes al examen parcial del cursado. Según corresponda, se tienen en cuenta como criterios de evaluación la obtención de respuestas correctas, el diseño algorítmico, la ejecución del programa y la adhesión a buenas prácticas en escritura de código. La nota final de la

asignatura para los estudiantes regulares se calcula como la ponderación entre la calificación obtenida en el examen final y la nota de cursada (60% y 40%, respectivamente).

BIBLIOGRAFÍA

Unidad 1

Grolemund, G. (2014). Hands-On Programming with R. O'Reilly. Capítulos 2 y 3. Disponible online: <https://rstudio-education.github.io/hopr/>.

Paradis, E. (2005). R para Principiantes. Universit'e Montpellier II. Capítulos 2 y 3. Disponible online: https://cran.r-project.org/doc/contrib/rdebuts_es.pdf.

Santana, S.; Mateos Farfán, E. (2014). El arte de programar en R: un lenguaje para la estadística. Instituto Mexicano de Tecnología del Agua, UNESCO. Capítulo 1. Disponible online: https://cran.r-project.org/doc/contrib/Santana_El_arte_de_programar_en_R.pdf.

Wickham, H.; Cetinkaya-Rundel, M.; Grolemund, G. (2023). R para Ciencia de Datos. O'Reilly. 2da Ed. Capítulos 1, 4, 6 y 8. Disponible online: <https://es.r4ds.hadley.nz/>.

Unidad 2

Grolemund, G. (2014). Hands-On Programming with R. O'Reilly. Capítulos 9 y 11 . Disponible online: <https://rstudio-education.github.io/hopr/>.

Paradis, E. (2005). R para Principiantes. Universit'e Montpellier II. Capítulos 6. Disponible online: https://cran.r-project.org/doc/contrib/rdebuts_es.pdf.

Santana, S.; Mateos Farfán, E. (2014). El arte de programar en R: un lenguaje para la estadística. Instituto Mexicano de Tecnología del Agua, UNESCO. Capítulo 4. Disponible online: https://cran.r-project.org/doc/contrib/Santana_El_arte_de_programar_en_R.pdf.

Wickham, H. (2019). Advanced R. Chapman and Hall/CRC. 2nd Ed. Capítulo 5. Disponible online: <https://adv-r.hadley.nz/>.

Wickham, H.; Cetinkaya-Rundel, M.; Grolemund, G. (2023). R para Ciencia de Datos. O'Reilly. 2da Ed. Capítulo 21. Disponible online: <https://es.r4ds.hadley.nz/>.

Unidad 3

Grolemund, G. (2014). Hands-On Programming with R. O'Reilly. Capítulo 2 . Disponible online: <https://rstudio-education.github.io/hopr/>.

Santana, S.; Mateos Farfán, E. (2014). El arte de programar en R: un lenguaje para la estadística. Instituto Mexicano de Tecnología del Agua, UNESCO. Capítulo 5. Disponible online: https://cran.r-project.org/doc/contrib/Santana_El_arte_de_programar_en_R.pdf.

Wickham, H. (2019). Advanced R. Chapman and Hall/CRC. 2nd Ed. Capítulos 6, 7 y 8. Disponible online: <https://adv-r.hadley.nz/>.

Wickham, H.; Cetinkaya-Rundel, M.; Grolemund, G. (2023). R para Ciencia de Datos. O'Reilly. 2da Ed. Capítulo 19. Disponible online: <https://es.r4ds.hadley.nz/>.

Unidad 4

Moeller, J. (2013). The Windows Command Line Beginner's Guide. Azure Flame Media. 2nd Ed. Capítulos 1, 4 y 7.

Santana, S.; Mateos Farfán, E. (2014). El arte de programar en R: un lenguaje para la estadística. Instituto Mexicano de Tecnología del Agua, UNESCO. Capítulo 10. Disponible online: https://cran.r-project.org/doc/contrib/Santana_El_arte_de_programar_en_R.pdf.

Shotts, W. (2024). The Linux Command Line: A Complete Introduction. No Starch Press. 6th Internet Edition. Capítulos 1 a 4. Disponible online: <https://sourceforge.net/projects/linuxcommand/files/TLCL/24.11/TLCL-24.11.pdf/download>

Unidad 5

Grolemund, G. (2014). Hands-On Programming with R. O'Reilly. Capítulos 5, 6 y 7. Disponible online: <https://rstudio-education.github.io/hopr/>.

Paradis, E. (2005). R para Principiantes. Université Montpellier II. Capítulo 3. Disponible online: https://cran.r-project.org/doc/contrib/rdebuts_es.pdf.

Santana, S.; Mateos Farfán, E. (2014). El arte de programar en R: un lenguaje para la estadística. Instituto Mexicano de Tecnología del Agua, UNESCO. Capítulos 2 y 3. Disponible online: https://cran.r-project.org/doc/contrib/Santana_El_arte_de_programar_en_R.pdf.

Wickham, H. (2019). Advanced R. Chapman and Hall/CRC. 2nd Ed. Capítulos 3 y 4. Disponible online: <https://adv-r.hadley.nz/>.

Wickham, H.; Cetinkaya-Rundel, M.; Grolemund, G. (2023). R para Ciencia de Datos. O'Reilly. 2da Ed. Capítulo 20. Disponible online: <https://es.r4ds.hadley.nz/>.

Unidad 6

Peng, R.; Kross, S.; Anderson, B. (2020). Mastering Software Development in R. Leanpub. Capítulo 3. Disponible online: <https://bookdown.org/rdpeng/RProgDA/>.

Wickham, H.; Bryan, J. (2023). R Packages. O'Reilly. 2nd Ed. Capítulos 1 a 5. Disponible online: <https://r-pkgs.org/>.