Universidad de Navarra

Lenguajes de programación (MC2)

Guía docente 2024-25

PRESENTACIÓN

Programar consiste en comunicarse con el ordenador para lograr que lleve a cabo una serie de tareas. En esta comunicación es fundamental dominar un ***lenguaje*** que pueda entender tanto una persona como el ordenador. El vocabulario y la gramática requieren de una ***sintaxis*** especial que se trata de aprender. Hay lenguajes especialmente valiosos para el uso científico; en esta asignatura se aprenderá **Matlab** y **Python**. En realidad, muchos conceptos de programación son comunes a todos los lenguajes; programar es un arte que se puede adquirir y cultivar. El científico que se libera de la condición de usuario para adquirir la de programador (aun a nivel elemental) multiplica su eficiencia a la hora de extraer información de los datos: tal es el objetivo de esta asignatura.

- **Titulación**: Máster en Métodos Computacionales en Ciencias
- **Carácter**: Obligatorio. *Módulo 2. Materia 2.2: Algoritmos y modelos en ciencias.*
- **ECTS** *: 5.*
- **Profesorado**: Profesor responsable: Angel Garcimartín Montero
- **Idioma**: *Se imparte en español (bibliografía en inglés).*
- **Aula 1** (Edificio Los Castaños).
- **Horario**: Primer semestre. Consultar el Calendario del Máster

RESULTADOS DE APRENDIZAJE (Competencias)

	Materia 2.2: Algoritmos y modelos en ciencias
	Lenguajes de programación
	COMPETENCIAS BÁSICAS
CB6	Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo o aplicación de ideas, a menudo en un contexto de investigación.
CB7	Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.

CB10	Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.
	COMPETENCIAS GENERALES
CG1	Emplear la ciencia de datos y la tecnología de la computación, a través de procedimientos de alto rendimiento, para ampliar el horizonte científico abriendo nuevas posibilidades.
CG2	Conocer los límites de los métodos estadísticos en el análisis de datos experimentales y desarrollar capacidad crítica para evaluar las implicaciones de los resultados obtenidos.
CG3	Conocer los principales problemas que se presentan en la adquisición y tratamiento de datos experimentales y cómo darles respuesta.
CG4	Comunicar tanto de manera oral como escrita un tema o datos de investigación en el área de las ciencias experimentales.
CG5	Poseer capacidad crítica, tanto en la lectura de literatura científica, como en la interpretación de los resultados experimentales.
	COMPETENCIAS ESPECÍFICAS
CE1	Conocer los lenguajes de programación estructurada más habituales en el ámbito científico, en lo que se refiere a la sintaxis, el control de flujo y el entorno de programación.

CE2	Elaborar un pseudocódigo para implementar los algoritmos más habituales de procesamiento de datos científicos, y a partir de él escribir un programa en un lenguaje informático de alto nivel.
CE3	Tomar conciencia de la necesidad de que los datos científicos se almacenen, procesen, filtren y representen de una manera formal y bien documentada para que resulten útiles y los experimentos puedan reproducirse.
CE4	Utilizar las técnicas de procesamiento y tratamiento de datos por ordenador más útiles para la investigación científica.
CE5	Aplicar los métodos computacionales de procesamiento de datos a un problema científico particular de la disciplina de interés para el estudiante.
CE6	Diseñar un experimento científico para que sea rico en información, recogiendo gran cantidad de datos de manera estructurada que faciliten su procesamiento posterior.
CE7	Integrar en el análisis científico datos obtenidos de fuentes heterogéneas.
CE8	Adquirir datos (bien en el laboratorio, o bien mediante minería on-line), organizarlos, filtrarlos, procesarlos, representarlos y refinarlos.
CE9	Extraer información de los datos con técnicas computacionales siguiendo un método científico.
CE10	Presentar los datos experimentales y la información científica de manera que se comuniquen de manera eficiente y fidedigna.

PROGRAMA

Universidad de Navarra

El curso se divide en dos partes: la primera en Matlab, y la segunda en Python (esta última de mayor duración). Al final de cada parte se hará un examen de ese lenguaje de programación.

PROGRAMA TEÓRICO

1. **Hardware y software**. Sistema operativo. Lenguajes y entornos de programación.
2. **Control de flujo**. Pseudocódigo. Condiciones, ramificaciones y bucles.
3. **Matlab**
    1. Sintaxis. Tipos de variable. Vectores y matrices. Programas: scripts y funciones.
    2. IDE. Ayuda y documentación. *Toolboxes*.
    3. Gráficos.
    4. Entrada y salida. Lectura y escritura de ficheros de datos.
4. **Python**
    1. Tipos de variable. Operadores y expresiones. Sintaxis. Control de flujo. Funciones.
    2. IDEs: Jupyter Notebook, Spyder.
    3. Entrada y salida.
    4. Módulos: *numpy*, *matplotlib*.

PROGRAMA PRÁCTICO

Aparte de ejercicios sobre la aplicación de los conceptos teóricos, se estudiarán casos prácticos de interés científico. Para el curso 2024/25, se propondrán -entre otros- los siguientes:

- Filtro de suavizado: la media móvil.
- Procesos estocásticos: simulación de una cola.
- Grafos (representación gráfica de relaciones).
- La mapa logísica.
- Análisis de supervivencia.
- *Clustering*: *k-means.*
- Análisis del sonido. Transformada de Fourier.
- *Data wrangling*.

El programa práctico contiene casos adicionales que se pueden realizar de manera opcional.

Es necesario acudir a las clases con ordenador portátil.

ACTIVIDADES FORMATIVAS

- **Clases teóricas**: 12 horas. Cubrirán el programa teórico.
- **Clases prácticas**: 18 horas. Se programarán detenidamente algunos casos prácticos.
- **Seminarios**: 17 horas: explicaciones sobre problemas y casos prácticos y presentaciones en público de trabajos personales; implementación de otros métodos y algoritmos de interés.
- **Tutorías**: 3 horas, *ad lib*.

Todas estas actividades son presenciales. La ausencia por motivos razonables debe justificarse.

Además, se requieren unas 50 horas de **trabajo personal** (el tiempo necesario para asimilar la materia puede variar de una persona a otra).

Universidad de Navarra

EVALUACIÓN

CONVOCATORIA ORDINARIA

La nota incluirá los siguientes apartados:

1. **Examen final** de Matlab (20 %) y **Examen final** de Python (40 %). Esta pruebas serán de carácter práctico y consistirán en la elaboración de un programa.
2. **Notas de clase:** 10%. En este apartado entran los siguientes ítems:
    1. **exposición pública de casos prácticos resueltos**
    2. **trabajos y tareas para realizar en casa**
    3. **intervenciones orales y respuestas a preguntas cortas a petición del profesor**
3. Resolución de **casos prácticos:** 10%. Se recogerán durante el curso, y se calificará tanto el cuaderno (diagrama de flujo, etc.) como el programa elaborado.

Nótese que los porcentajes suman el 80%: si solo se resuelven los casos prácticos obligatorios, la nota máxima que se puede obtener es Notable (8). Para sacar más de Notable, es necesario presentar también los casos prácticos voluntarios, de acuerdo con el profesor; en ese caso los porcentajes de los apartados (2) y (3) suben hasta el 20% cada uno de ellos.

Las calificaciones se otorgarán con el siguiente baremo (se indica la correspondencia numérica en una escala de 0 a 10): **A** = excelente = 9 ; **B** = muy bien = 7.5 ; **C** = bien = 6 ; **D** = aceptable = 5 ; **E** = deficiente = 3. Las calificaciones se pueden modular con **+** y **-**.

Durante el curso se asignarán trabajos, problemas o casos prácticos para entregar en los días fijados por el profesor. Salvo causa justificada **la asistencia a clase es obligatoria**, pues se puede solicitar en cualquier momento al alumno, sin aviso previo, entregar y exponer públicamente el trabajo realizado. La falta de asistencia no justificada puede conllevar una penalización en la calificación de la asignatura.

CONVOCATORIA EXTRAORDINARIA

- Se repetirán los exámenes finales de Matlab y de Python, que contarán con los mismos porcentajes de la convocatoria ordinaria. Las notas de los casos prácticos, notas de clase, y trabajos prácticos (apartados 2 y 3 de la convocatoria ordinaria) se guardan para la convocatoria extraordinaria.

HORARIOS DE ATENCIÓN

Ángel Garcimartín Montero (angel@unav.es)

- Despacho O-240 (Departamento de Física y Matemática Aplicada, Edificio Los Castaños).
- Horario de tutoría: Se establecerá un horario de atención de consultas al comienzo de curso de acuerdo con los estudiantes. También se atenderán consultas previa petición de hora por correo electrónico en otros momentos en los que el profesor esté disponible.

BIBLIOGRAFÍA

Muchos recursos no están en forma de libro, sino en herramientas y contenidos digitales que serán presentados durante el curso: por ejemplo, la ayuda en línea de las interfaces de programación.

Universidad de Navarra

Las obras disponibles en versión electrónica se pueden encontrar en la sección **Material docente** de esta asignatura, en la carpeta *Colección de libros (en formato PDF)*.

Bibliografía para iniciarse en Matlab:

- Al instalar Matlab, se puede acceder y descargar toda la documentación en PDF. Es muy recomendable el manual "***Getting Startd with Matlab***" -o *"Matlab Primer"*, según las versiones- (una de ellas está en la carpeta *libros en formato pdf*). También son útiles "*Data processing*", y "*Graphics*". Los manuales y la documentación en pdf varían mucho según la versión.

Bibliografía complementaria, Matlab:

- W. GANDER, "**Learning Matlab"**. Localízalo en la Biblioteca [recurso electrónico]
- S. ATTAWAY, "**Matlab: a practical introduction to programming and problem solving**". Localízalo en la Biblioteca [recurso electrónico]

Bibliografía para iniciarse en Python:

- S. R. DOTY, "**Python basics**". Documento gratuito (el pdf está en la carpeta *libros en formato pdf*) . Una introducción rápida, en 20 páginas.
- C. H. SWAROOP, "**A byte of Python**". Libro gratuito, disponible online - Localízalo en la Biblioteca y en epub, pdf. También se puede comprar una edición impresa (el pdf está en la carpeta *libros en formato pdf*).
- R. GONZÁLEZ DUQUE, "**Python para todos**". Libro gratuito (el pdf está en la carpeta *libros en formato pdf*). Libro bastante completo sobre Python, pero solo sobre el lenguaje; no trata apenas de los paquetes científicos.

***Bibliografía recomendada, Python*:**

- J. VANDERPLAS, "**Python Data Science Handbook**", excelente manual que explica en profundidad los paquetes Numpy, Pandas, Matplotlib y una introducción Machine Learning. Versión *online* (en el repositorio del autor).
- VARIOS AUTORES, "**NumPy User Guide**". Libro gratuito (el pdf está en la carpeta *libros en formato pdf*) . Contiene dos capítulos de introducción, uno básico y otro intermedio, muy recomendables.

Bibliografía complementaria, Python:

- E. BRESSERT, "**SciPy and NumPy**". [Agotado]. [Hay ejemplares electrónicos]. Muy cortito, es una introducción a los módulos científicos.
- VARIOS AUTORES, "**Scipy Lecture Notes**". Libro gratuito (el pdf está en la carpeta *libros en formato pdf*) . Trata sobre las librerías científicas en Python. Interesante para consultar procedimientos específicos.
- C. HILL, " **Learning scientific programming with Python**". Disponible en versión electrónica en la biblioteca: Localízalo en la Biblioteca

Sitios web:

- Centro de documentación de Matlab
- Centro de documentación de Python
- NumPy (computación científica en Python)
- matplotlib (visualización gráfica en Python)
- SciPy (librerías científicas de Python)