Programación

Curso Académico: (2022 / 2023) **Fecha de revisión: 30-05-2022**

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: GARCIA OLAYA, ANGEL

Tipo: Formación Básica Créditos ECTS : 6.0

Curso : 1 Cuatrimestre : 1

Rama de Conocimiento: Ingeniería y Arquitectura

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

Ninguna

COMPETENCIAS Y RESULTADOS DEL APRENDIZAJE

¿ Reconocer las diferencias entre los paradigmas de programación secuencial imperativa y de programación orientada a objetos así como las diferencias entre lenguajes interpretados y compilados.

¿ Conocer y aplicar las estructuras fundamentales de programación, estructuras de control y algoritmos

¿ Conocer y aplicar las buenas prácticas en programación: descomposición, reutilización de código, ocultación de la implementación, herencia, polimorfismo ¿

DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

1. Introducción a la asignatura y a la programación
   1.1. Elementos de un programa: datos y algoritmos
   1.2. Arquitectura básica de un ordenador
   1.3. Codificación de la información en memoria
   1.4. Breve introducción histórica a la programación
   1.5. Compilación vs. interpretación de programas
   1.6. Paradigmas de programación: imperativa, lógica y funcional
2. Diagramas de flujo
3. Introducción al lenguaje Python
   3.1. El intérprete Python
   3.2. Tipos de datos básicos
   3.3. Creación de programas en Python
   3.4. Variables y constantes
   3.5. Entrada, salida y números aleatorios
   3.6. Operadores
   3.7. Conversiones entre tipos
   3.8. Comentarios
   3.9. Depuración de programas: tipos de errores
4. Control de flujo: condicionales y bucles
   4.1. Condicionales
   4.2. Ámbito de variables y bloques de código
   4.3. Bucles
5. Estructuras de datos sencillas
   5.1. Listas y tuplas
   5.2. Diccionarios
6. Funciones
   6.1. Descomposición
   6.2. Reutilización de código
   6.3. Ocultación de la implementación
   6.4. Paso por valor y referencia
   6.5. Introducción a la recursividad
7. Introducción a la Programación Orientada a Objetos
   7.1. Clases y objetos
   7.2. Métodos
   7.3. Herencia
   7.4. Polimorfismo
8. Algoritmos sobre listas

8.1. Introducción a la complejidad computacional
8.2. Búsqueda
8.3. Ordenación

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

Actividades formativas

1. Clases magistrales (1 ECTS) Orientadas a la enseñanza de las competencias específicas de la asignatura, especialmente las relacionadas con los conocimientos básicos y las técnicas de programación imperativa
2. Clases prácticas (1 ECTS) Orientadas a poner en práctica los conocimientos obtenidos en las clases magistrales
3. Trabajo autónomo (2,5 ECTS) Orientado al aprendizaje individual o en grupo de las competencias específicas de la asignatura. También servirá como base para la adquisición de las competencias genéricas siguientes: Comunicación oral y escrita, Capacidad de análisis y síntesis, Capacidad para la toma de decisiones, Motivación por la calidad y Capacidad para la autoorganización y planificación del trabajo y del proceso de aprendizaje.
4. Tutorías (1 ECTS) Asistencia individualizada (tutorías individuales) o en grupo (tutorías colectivas) a los estudiantes por parte del profesor.
5. Examen final (0,5 ECTS). Desarrollo en un tiempo limitado de un programa orientado a objetos. Permite al estudiante mostrar sus conocimientos de una forma integrada.

METODOLOGÍA

Las clases se dividirán en sesiones magistrales y sesiones de prácticas en laboratorios informáticos. Las clases presenciales se complementarán con el trabajo autónomo del alumno en casa.

- Clases teóricas (grupo magistral): se presentarán los conceptos básicos de programación, utilizando Python como lenguaje para facilitar su explicación. Al final de la clase se podrán plantear preguntas cortas o tipo test sobre la materia explicada en esa sesión.

- Clases prácticas (grupo pequeño): constarán de tres partes. En la primera algunos alumnos, elegidos al azar, explicarán las soluciones a los problemas propuestos la semana anterior. A continuación, el profesor planteará un problema y lo resolverá con la ayuda de los alumnos. Por último, en cada sesión se propondrán una serie de problemas que los alumnos empezarán a resolver durante el resto de la clase de prácticas y que se deben entregar al final de esa semana. Se propondrá una práctica final de la asignatura que los alumnos deberán realizar en grupos de dos y que podrá tener entregas parciales. La práctica final y los ejercicios semanales se desarrollarán en Python.

- Trabajo autónomo del estudiante: durante la semana se deberán terminar los ejercicios propuestos en las clases de prácticas, que se entregarán a través de Aula Global.

Régimen de tutorías

De acuerdo a la normativa de la universidad se establecerá un horario preferente de tutorías presenciales para cada profesor de teoría o práctica. Este horario estará disponible en Aula Global. No obstante se permite a los alumnos a contactar con el profesor mediante correo electrónico para establecer tutorías fuera de esos horarios.
Se utilizará el correo electrónico para resolver dudas puntuales que no requieren de tutoría presencial. Existe también un foro de la asignatura en Aula Global para que los alumnos puedan exponer dudas genéricas sobre la asignatura, que pueden ser contestadas por el profesor o por sus propios compañeros. Durante la práctica final se establecerán tutorías individuales con cada uno de los grupos.

SISTEMA DE EVALUACIÓN

Evaluación combinada y continua a partir de las diferentes actividades realizadas por el alumno de forma individual o en grupo, teniendo en cuenta el esfuerzo realizado por cada alumno en cada una de las actividades formativas descritas anteriormente.
Se realizará una evaluación formativa a través de la realimentación continua, que permita al alumno evaluar qué conoce y qué se espera de él. La evaluación constará de:
- Exámenes parciales y tests en clase (nota individual): 1,5 puntos
- Presentaciones orales de los ejercicios en clase de prácticas (nota individual): 0,5 puntos
- Ejercicios semanales: 0,5 puntos (será necesario entregar al menos el 80% de ellos)
- Práctica final (nota por parejas):2,5 puntos
- Examen teórico-práctico final (nota individual): 5 puntos

Las fechas del examen parcial y de entrega de la práctica final se encuentran en el calendario de evaluación continua.

Para aprobar la asignatura es necesario obtener 5 puntos calculados de la siguiente manera, que varía según la convocatoria:
En la convocatoria ordinaria, la nota será el máximo entre:
1) 60% de la nota obtenida en el examen final, calculada sobre 10 puntos
y
2) notas de evaluación continua + nota del examen final / 2

Para aplicar la fórmula del punto 2) la nota del examen final deberá ser mayor o igual a un 4 sobre 10.
En caso de que se detecte copia en cualquiera de las actividades de la evaluación continua, todos las personas implicadas perderán el derecho a evaluación continua y se les aplicará exclusivamente la fórmula del apartado 1)

En la convocatoria extraordinaria la nota será el máximo entre:
1) la nota del examen extraordinario calculada sobre 10 puntos
y
2) notas de evaluación continua + nota del examen extraordinario / 2

Al igual que en la convocatoria ordinaria, para aplicar 2) la nota del examen extraordinario deberá ser de un 4 sobre 10

No se propondrán nuevas prácticas para la convocatoria extraordinaria ni se podrán realizar las no entregadas en el primer cuatrimestre.

La nota final corresponderá en un 75% a las actividades individuales del alumno y un 25% a las actividades de equipo. Las actividades llevadas a cabo durante el curso, individuales o en grupo, supondrán un 50% de la nota, mientras que el examen final individual constituirá el 50% restante.

Peso porcentual del Examen Final:	50
Peso porcentual del resto de la evaluación:	50

BIBLIOGRAFÍA BÁSICA

- Ana Bell Get Programming Learn to code with Python, Manning publications, 2018

- John S. Conery Explorations in Computing: An Introduction to Computer Science and Python Programming, CRC Press, 2014

RECURSOS ELECTRÓNICOS BÁSICOS

- Guido van Rossum, Barry Warsaw, Nick Coghlan (traducido por Raúl González Duque) . PEP 8 -- Guia de estilo del código Python: http://mundogeek.net/traducciones/guia-estilo-python.htm
- Python Software Foundation . Python for Beginners: https://www.python.org/about/gettingstarted/