Programación

Curso Académico: (2020 / 2021) **Fecha de revisión: 09-07-2020**

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

OBJETIVOS

Competencias generales del título cubiertas:
- CGB4: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- CGB5: Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.

Competencias comunes a la rama de informática del título cubiertas:
- CECRI6: Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- CECRI7: Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.

Competencias específicas de la asignatura:
- Conocimiento de los conceptos básicos de la programación imperativa desde un enfoque orientado a objetos (PO a,k).
- Conocimientos básicos de la sintaxis de un lenguaje de programación orientado a objetos (PO k).
- Capacidad para descomponer un problema real según una metodología orientada a objetos para su posterior codificación en un programa (PO a,c,e,k).
- Conocimiento de estructuras sencillas para el manejo de información (PO a).
- Capacidad para comprender documentación técnica y reutilizar código desarrollado por terceras partes (PO a,k).
- Conocimientos básicos del concepto de complejidad computacional (PO a,k)

Competencias generales:
- Capacidad para la autoorganización y planificación del trabajo individual y del proceso de aprendizaje (PO k).
- Capacidad para el trabajo en grupo (PO d).
- Comunicación oral y escrita (PO g).
- Capacidad de análisis y síntesis (PO a).
- Capacidad para la toma de decisiones (PO e).
- Motivación por la calidad (PO k).

DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

1. Introducción a la asignatura y a la programación
   a. Elementos de un programa: datos y algoritmos
   b. Arquitectura básica de un ordenador
   c. Codificación de la información en memoria
   d. Breve introducción histórica a la programación: desde el código binario hasta la programación orientada a componentes
   e. Compilación vs. interpretación de programas
   f. Paradigmas de programación: imperativa, lógica y funcional
2. Diagramas de flujo

3. Introducción al lenguaje Python
   a. El intérprete Python
   b. Tipos de datos básicos
   c. Creación de programas en Python
   d. Variables y constantes
   e. Entrada, salida y números aleatorios
   f. Operadores
   g. Conversiones entre tipos
   h. Comentarios
   i. Depuración de programas: tipos de errores
4. Control de flujo: condicionales y bucles
   a. Condicionales
   b. Ámbito de variables y bloques de código
   c. Bucles
5. Estructuras de datos sencillas
   a. Listas y tuplas
   b. Diccionarios
6. Funciones
   a. Descomposición
   b. Reutilización de código
   c. Ocultación de la implementación
   d. Paso por valor y referencia
   e. Introducción a la recursividad
7. Introducción a la Programación Orientada a Objetos
8. Algoritmos sobre listas
   a. Introducción a la complejidad computacional
   b. Búsqueda (lineal y binaria)
   c. Ordenación (métodos directos: burbuja, inserción directa y selección directa)

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

Actividades formativas

1. Clases magistrales (1 crédito ECTS) (PO a,k)

- 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. Trabajo individual en clase y en casa (3 créditos ECTS) (PO a,c,e,g,k)

- Orientado al aprendizaje autónomo de las competencias específicas de la asignatura, asegurando una base de conocimientos mínima que poder utilizar en el trabajo en grupo. Se seguirá un enfoque de aprendizaje basado en problemas (PBL).
- 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 individual y del proceso de aprendizaje.

3. Prácticas en grupo (2 créditos ECTS) (PO a,c,d,e,g,k)

- Completa las actividades anteriores fijando las competencias específicas y generales y haciendo hincapié en las competencias generales siguientes: Capacidad para el trabajo en grupo y Comunicación oral y escrita. Se seguirá también un enfoque basado en PBL.

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 y programas de ejemplo. Al final de la clase se podrán plantear preguntas cortas o tipo test sobre la materia explicada en esa sesión. Estas preguntas se pondrán posteriormente a disposición de los alumnos en forma de test de Aula Global para que las respondan las veces que consideren necesario.

- 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 (ver cronograma). La práctica final y los ejercicios semanales se desarrollarán en Python.

- Trabajo autónomo del alumno: durante la semana los alumnos deberán terminar los ejercicios propuestos en las clases de prácticas y entregarlos 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 anima 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 (PO a,c,e,g,k) (competencias CGB4 y CECRI6).
- Presentaciones orales de los ejercicios en clase de prácticas (nota individual): 0,5 puntos (competencias CGB4 y CECRI6)
- Ejercicios semanales: 0,5 puntos (será necesario entregar al menos el 80% de ellos) (PO a,c,d,e,g,k) (competencias CGB4, CGB5, CECRI6 y CECRI7)
- Práctica final (nota por parejas):2,5 puntos (PO a,c,d,e,g,k) (competencias CGB4, CGB5, CECRI6 y CECRI7)
- Examen teórico-práctico final (nota individual): 5 puntos (competencias CGB4 y CECRI6) (PO a,c,e,g,k)

Las fechas de entrega de la práctica final y de los exámenes parciales se encuentran en el cronograma de la asignatura.

Para aprobar la asignatura es necesario obtener 5 puntos calculados de la siguiente manera, según la convocatoria:
En la convocatoria ordinaria, la nota será el máximo entre:
1) 0,60 por la nota del examen final calculada sobre 10 puntos
y
2) notas de evaluación continua + nota del examen final (sobre 5)

Para aplicar la fórmula del punto 2) la nota del examen final deberá ser al menos de un 2 sobre 5 (un 4 sobre 10). En caso de que se detecte copia en cualquiera de las prácticas, todos los implicados 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 (sobre 5)

De igual manera que antes, para aplicar 2) la nota del examen extraordinario deberá ser al menos de un 2 sobre 5 (un 4 sobre 10)

No se propondrán nuevas prácticas para Junio ni se podrán entregar las ya realizadas 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/