Guía docente
330053 - I - Informática

Última modificación: 25/04/2024

Unidad responsable: Escuela Politécnica Superior de Ingeniería de Manresa
Unidad que imparte: 750 - EMIT - Departamento de Ingeniería Minera, Industrial y TIC.

Titulación:
GRADO EN INGENIERÍA ELÉCTRICA (Plan 2009). (Asignatura obligatoria).
GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA (Plan 2009). (Asignatura obligatoria).
GRADO EN INGENIERÍA MECÁNICA (Plan 2009). (Asignatura obligatoria).
GRADO EN INGENIERÍA QUÍMICA (Plan 2009). (Asignatura obligatoria).
GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA (Plan 2016). (Asignatura obligatoria).
GRADO EN INGENIERÍA MECÁNICA (Plan 2016). (Asignatura obligatoria).
GRADO EN INGENIERÍA QUÍMICA (Plan 2016). (Asignatura obligatoria).
GRADO EN INGENIERÍA DE RECURSOS MINERALES Y SU RECICLAJE (Plan 2021). (Asignatura obligatoria).

Curso: 2024 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: Vives Pons, Jordi

Otros:
Llusà Serra, Aleix
Arumi Casanovas, Arnau
Masip Riera, Jordi

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:
1. Conocimientos fundamentales sobre la utilización y la programación de los ordenadores.
2. Sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Transversales:
3. COMUNICACIÓN EFICAZ ORAL Y ESCRITA - Nivel 1: Planificar la comunicación oral, responder de manera adecuada a las cuestiones formuladas y redactar textos de nivel básico con corrección ortográfica y gramatical.
4. TRABAJO EN EQUIPO - Nivel 1: Participar en el trabajo en equipo y colaborar, una vez identificados los objetivos y las responsabilidades colectivas e individuales, y decidir conjuntamente la estrategia que se debe seguir.
5. USO SOLVENTE DE LOS RECURSOS DE INFORMACIÓN - Nivel 1: Identificar las propias necesidades de información y utilizar las colecciones, los espacios y los servicios disponibles para diseñar y ejecutar búsquedas simples adecuadas al ámbito temático.
6. APRENDIZAJE AUTÓNOMO - Nivel 1: Llevar a cabo tareas encomendadas en el tiempo previsto, trabajando con las fuentes de información indicadas, de acuerdo con las pautas marcadas por el profesorado.

METODOLOGÍAS DOCENTES

La asignatura se estructura en dos clases de dos horas por semana. De estas cuatro horas presenciales semanales una se dedica a presentar los principales contenidos de forma expositiva, la segunda en la resolución de problemas bajo demanda del estudiantado y las dos restantes a resolver problemas prácticos en el laboratorio informático. Al estudiante se le indican semanalmente las tareas de estudio y solución de problemas que debe hacer. Estas tareas se aconseja hacerlas, al menos parcialmente, trabajando en equipo. Periódicament se evalúa el progreso de cada estudiante
individualmente. La asignatura también incorpora un proyecto de desarrollo de software de un tamaño medio que hay que trabajar en equipo.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Tras superar esta asignatura el estudiante debe:
1. Saber aplicar los conceptos fundamentales de programación de computadores.
2. Demostrar destreza en la utilización de las técnicas y las herramientas básicas de la programación.
3. Tener capacidad para resolver problemas mediante el desarrollo de programas de complejidad pequeña y media.
4. Tener capacidad de abstracción en la utilización de modelos para la resolución de problemas reales.
5. Saber planificar la comunicación oral, responder de manera adecuada las cuestiones formuladas y redactar textos de nivel básico con corrección ortográfica y gramatical.
6. Saber identificar las propias necesidades de información y utilizar las colecciones, los espacios y los servicios disponibles para diseñar y ejecutar búsquedas simples adecuadas al ámbito temático.
7. Poder llevar a cabo las tareas encomendadas en el tiempo previsto, trabajando con las fuentes de información indicadas, de acuerdo con las pautas marcadas por el profesorado.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00
Horas grupo grande	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

Tema 1. Introducción a la programación

Descripción:
Es necesario que el estudiante entre por la vía directa al mundo de la programación. Por lo tanto hay que introducir éste en los conceptos más básicos, sin profundizar excesivamente, por lo que tenga un primer contacto con la programación. No se busca una comprensión sólida sino más bien que, desde la experimentación, adquiera una serie de herramientas que luego le permitan avanzar más deprisa.
Palabras clave: Computador, programa, algoritmo, error de programación (bug), lenguaje de programación, portabilidad, intérprete, shell, script, debugació, errores sintaxis, errores ejecución, errores semánticos, valor, variable, tipo, asignación, entrada, salida, lectura, escritura, sentencia, palabra reservada, expresión, operador, operando, precedencia, evaluación, composición de sentencias, función, cabecera, cuerpo, grita, parámetros, valor de retorno, localidad.

Actividades vinculadas:
Todas las que constan.

Dedicación: 12h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 8h

Tema 2. Condicionales e interacciones

Descripción:
Este tema extiende el anterior siguiendo la misma línea: un paseo rápido, práctico e intuitivo para los elementos básicos. La intención no es adquirir un conocimiento profundo sino una perspectiva superficial. En este caso, los elementos básicos que se estudian son las construcciones alternativa y iterativa principalmente.
Palabras clave: Módulo, tipos y valores booleanos, expresiones booleanas, operadores booleanos, sentencia alternativa (o condicional), bloque, condicionales encadenados y anidados, retorno de una función, lectura del teclado, conversión de tipo (cast), None, composición de funciones en expresiones, funciones booleanas, funciones como valores (con suavidad!), sentencia iterativa, iteraciones infinitas, variables locales.

Actividades vinculadas:
Todas las que constan.

Dedicación: 12h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 8h

Tema 3. Documentación y test de programas

Descripción:
Este tema es una especie de relajamiento después de los dos temas anteriores que sólo introduce el concepto de unido test conjuntamente con las herramientas de doctests y nose. El objetivo es doble. Por un lado hay que consolidar los conocimientos adquiridos en los temas anteriores y por el otro hay que adquirir los conocimientos sobre unido testing vía doctest que permitan incorporar esta herramienta con normalidad a partir de este tema.
Palabras clave: Unit test, doctest, nose, test driven development.

Actividades vinculadas:
Todas las que constan.

Dedicación: 6h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 4h

Tema 4. Strings

Descripción:
En los tres primeros temas se han expuesto de manera rápida pero aplicable los elementos más básicos de un lenguaje de programación. Estos tres temas juntos conforman el primer bloque temático del curso. El bloque que se inicia con este tema añade encima de los anteriores los elementos más básicos de almacenamiento de información. Después de este segundo bloque, la potencia de los elementos disponibles es muy considerable. Para iniciar el bloque, este tema se dedica a los strings. Estos, aparte del su interés inherente, son los representantes de dos conceptos importantes en Python: las secuencias y los tipos inmutables.
Palabras clave: String, tipo estructural (en contraposición a simple), índice, operación de acceso, rebanada (slice), recorrido (traversal), operador de pertenencia, inmutabilidad, parámetros opcionales de una función, valores por omisión (default), módulo string, operador de formato de strings.

Actividades vinculadas:
Todas las que constan.

Dedicación: 12h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 8h

Tema 5. Listas

Descripción:
Continuando con el bloque sobre los elementos más básicos de almacenamiento de información, este tema se dedica a las listas. Las listas son la estructura de datos leneal por excelencia y Python son un tipo de datos nativo. Como complemento, se usan las listas para introducir el concepto fundamental de tipo montable.
Palabras clave: Lista y tipo lista. Lista de listas. Listas homogéneas y heterogéneas. operaciones de acceso a los elementos de una lista. Operaciones sobre listas: longitud, pertenencia, concatenación, repetición. Rebanadas o intervalos en listas. El constructor "range". Mutabilidad: el caso de las listas. Borrado en listas. Objetos, valores y aliasing. Clonación. El iterador de listas. mutabilidad y parámetros: el caso de las listas. Funciones puras y efecto lateral. Matrices. Relación entre tipo y strings.

Actividades vinculadas:
Todas las que constan.

Dedicación: 12h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 8h

Tema 6. Esquemas de tratamiento secuencial

Descripción:
Contrariamente a los temas vistos hasta ahora, este tema es de carácter metodológico. Después de haber introducido los principales elementos del lenguaje Python, en este tema se aborda la problemática del diseño de iteraciones. El enfoque adoptado es el de los esquemas de programación: un mecanismo sencillo y potente en manos de programadores entrenants.
Palabras clave: Esquema de programación. Secuencia. Esquema de recorrido. Esquema de búsqueda. Buscar con booleano. Iterador for. Sentencia break. Sentencia else (aplicada a iteraciones).

Actividades vinculadas:
Todas las que constan.

Dedicación: 6h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 4h

Tema 7. Módulos y ficheros

Descripción:
Este tema se dedica a dos cuestiones que, si bien están desconexos de la apert central del curso, son de gran importancia. Se trata por un lado los módulos, un mecanismo para mejorar la organización del código y aumentar su reaprovechamiento, y por otro lado los ficheros, el medio básico de almacenamiento de información externa.
Palabras clave: Módulo. Sentencia importe. Ámbito (namespace). Conflicto de identificadores. Sentencia continúe. Atributos y operador de acceso a los tributos. Métodos de un objeto. Métodos de los strings. Métodos de las listas. Archivos. Archivos de texto. Abrir, leer / escribir, cerrar. Fin de archivo. Archivos de
texto. Directorios. El módulo sys. Argv y el paso de parámetros al ejecutable.

Actividades vinculadas:
Todas las que constan.

Dedicación: 12h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 8h

Tema 8. Tuplas

Descripción:
Este es un tema de transición que aporta pocos conocimientos nuevos pero que trabaja los conceptos de mutabilidad / inmutabilidad y las funciones de forma reiterada. Los nuevos conceptos se centran en las estructuras llamadas tuplas (que en el contexto de Python no se deben confundir con los registros) y con las comprehension, una herramienta muy potente para trabajar con listas.
Palabras clave: Tupla (en el sentido Python). Operaciones sobre tuplas. Inmutabilidad de las tuplas. Asignación de tuplas: extensión al caso de las funciones. Comprehension.

Actividades vinculadas:
Todas las que constan.

Dedicación: 6h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 4h

Tema 9. Diccionarios

Descripción:
Este tema cierra la serie de temas dedicados a los tipos de datos estructurales de Python. Los diccionarios son estructuras asociativas que permiten implementar correspondencias clave-valor de forma sencilla y son una herramienta de gran potencia. El conjunto de tipos predefinidos de Python (strings, listas, tuplas y diccionarios) constituye una de sus características más apreciadas.
Palabras clave: Diccionario. Correspondencia. Clave. Valor asociado a una clave. Inserción y borrado de elementos. Constructores de diccionarios. Conjunto de claves y multiconjunto de valores.

Actividades vinculadas:
Todas las que constan.

Dedicación: 20h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 8h
Aprendizaje autónomo: 8h

ACTIVIDADES

ACTIVIDAD 1: CLASE EXPOSITIVA

Descripción:
Son clases presenciales específicamente dedicadas a la comprensión de los contenidos de la asignatura. Son clases con baja participación de los estudiantes.

Material:
Los materiales de apoyo son:
- Referencia principal de la asignatura (libro en formato web).
- Bibliografía básica.
- Colección de problemas de la asignatura.

Dedicación: 12h
Grupo grande/Teoría: 12h

ACTIVIDAD 2: CLASES DE PROBLEMAS

Descripción:
Son clases presenciales específicamente dedicadas a la resolución de problemas. Se hacen en una aula ordinaria y son complementarias de la actividad en el laboratorio. Son clases que requieren la participación de los estudiantes.

Material:
Los materiales de apoyo son:
- Referencia principal de la asignatura (libro en formato web).
- Bibliografía básica.
- Colección de problemas de la asignatura.

Dedicación: 12h
Grupo grande/Teoría: 12h

ACTIVIDAD 3: CLASE DE LABORATORIO

Descripción:
El estudiante tiene como objetivo la solución de pequeños ejercicios que complementan los contenidos y colaboran en la mejor comprensión de los mismos. Los ejercicios se realizan en el laboratorio y comportan la implementación real de programas sobre el computador y su comprobación. La actividad puede suponer la finalización de los ejercicios en tiempo de aprendizaje autónomo.

Material:
Los materiales de apoyo son:
- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.
- Manuales del software utilizado.
- Bibliografía básica.

Entregable:
Periódicamente el estudiante, de manera individual, entrega un pequeño ejercicio que evalúa sus progresos en esta actividad. El ejercicio se realiza dentro del tiempo de la misma actividad. Estos ejercicios computan dentro del epígrafe A en la nota final.

Dedicación: 42h
Grupo pequeño/Laboratorio: 27h
Aprendizaje autónomo: 15h

ACTIVIDAD 4: ESTUDIO DE CONTENIDOS

Descripción:
El estudio de los contenidos es la actividad indivdual o colectiva que conduce a entender y asumir los conocimientos, vocabulario y técnicas que forman parte de los contenidos de la asignatura.

Material:
Los materiales de apoyo son:
- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.

Dedicación: 20h
Aprendizaje autónomo: 20h

ACTIVIDAD 5: REALIZACIÓN DE EJERCICIOS

Descripción:
Es una actividad que hace el estudiante autónomamente y que consiste en la solución de problemas de programación, generalmente sin ser necesario el apoyo del computador.

Material:
Los materiales de apoyo son:
- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.

Entregable:
La actividad conlleva la entrega durante el curso de algunos problemas que se corrigen convenientemente y forman parte de la evaluación de la asignatura. Estos ejercicios computan dentro del epígrafe A en la nota final.

Dedicación: 25h
Aprendizaje autónomo: 25h

ACTIVIDAD 6: PROYECTO

Descripción:
La asignatura requiere realizar un proyecto de programación de tamaño medio. El proyecto consiste en la implementación y test de un programa cuyo diseño viene dado por el enunciado. Esta actividad se realiza en ruptura y conlleva, además, la escritura de un informe técnico sobre el programa.
Esta actividad tiene naturaleza de síntesis de todos los conocimientos de la asignatura.

Material:
Los materiales de soporte son:
- Servicio de laboratorio informático del CCEPSEM.
- Enunciado y guión del proyecto.
- Ejemplo de informe.
- Apuntes personales y resto de material de apoyo del curso.

Dedicación: 28h
Grupo grande/Teoría: 4h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 20h

ACTIVIDAD 7. EXAMEN

Descripción:
La asignatura contempla un examen final que consiste en un conjunto de ejercicios a resolver individualmente sobre papel sin apoyo de ningún tipo de material y en un tiempo acotado. Esta actividad incluye un tiempo personal de preparación de la prueba.

Dedicación: 12h
Grupo grande/Teoría: 2h
Aprendizaje autónomo: 10h

SISTEMA DE CALIFICACIÓN

La calificación se realiza en base a 3 elementos:

1. La evaluación del trabajo autónomo del estudiante (A). Este componente contiene tanto el progreso hecho en los aspectos teóricos como en los prácticos. Su medida se realiza a base de ejercicios obligatorios entregados durante el curso. Se subdivide en A1: nota correspondiente a la evaluación que se llevará a cabo los ejercicios de laboratorio y A2: nota correspondiente a la media de notas de 3 puntuables que se realizarán en las clases de teoría, junto con la realización por parte de los estudiantes de un ejercicio largo guiado.

2. La evaluación del proyecto (P). Se realiza a partir de una entrega presencial del proyecto de curso que puede comportar una presentación pública y la confección de una memoria.

3. La evaluación final (F). Se hace a través de un examen final que tiene naturaleza global e integra todos los conocimientos y destrezas adquiridos durante el curso.

A partir de estos elementos se calcula la nota final con las siguientes ponderaciones:

Final = 0:35A (0:1A1+0:25A2)+ 0:25P + 0:40F

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Las actividades se realizarán siguiendo los usos y costumbres del trabajo académico y, particularmente, se respetarán las siguientes pautas:
1. Aquellas actividades que sean explícitamente declaradas como individuales, sean de naturaleza presencial o no, se realizarán sin colaboración por parte de otras personas.
2. Las fechas, formatos y demás condiciones de entrega que se fijen serán de obligado cumplimiento.
3. El uso del laboratorio informático se reservará exclusivamente para las actividades académicas y en ningún caso se podrá hacer un uso abusivo.

BIBLIOGRAFÍA

Básica:
- Downey, A. Python for software design: how to think like a computer scientist [en línea]. Cambridge: Cambridge University Press, 2009 [Consulta: 09/11/2020]. Disponible a: http://openbookproject.net/thinkcs/python/english3e/. ISBN 9780521725965.
- Pilgrim, M. Dive into Python [en línea]. New York: Apress, 2004 [Consulta: 06/11/2020]. Disponible a: http://www.diveintopython3.net/. ISBN 1590593561.
- Guzdial, M.; Ericson, B. Introduction to computing & programming in Python: a multimedia approach [en línea]. 2nd ed. Upper Saddle River: Pearson/ Prentice Hall, 2010 [Consulta: 31/05/2022]. Disponible a: https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?docID=5185706. ISBN 9780136060239.