## Datos básicos de la asignatura

| | |
|---|---|
| **Titulación:** | Grado en Ingeniería Informática-Ingeniería del Software |
| **Año plan de estudio:** | 2010 |
| **Curso implantación:** | 2010-11 |
| **Centro responsable:** | E.T.S. Ingeniería Informática |
| **Nombre asignatura:** | Fundamentos de Programación |
| **Código asigantura:** | 2050001 |
| **Tipología:** | TRONCAL / FORMACIÓN BÁSICA |
| **Curso:** | 1 |
| **Periodo impartición:** | Anual |
| **Créditos ECTS:** | 12 |
| **Horas totales:** | 300 |
| **Área/s:** | Lenguajes y Sistema Informáticos |
| **Departamento/s:** | Lenguajes y Sistemas Informáticos |

## Coordinador de la asignatura

REINA QUINTERO, ANTONIA MARIA

## Profesorado (puede sufrir modificaciones a lo largo del curso por necesidades organizativas del Departamento)

**Profesorado de grupo principal**

CARRANZA GARCIA, MANUEL
RIQUELME SANTOS, JOSE CRISTOBAL

**Profesorado de otros grupos**

CRUZ MATA, FERMIN
JIMENEZ AGUIRRE, PATRICIA
SANCHEZ LOPEZ, JOSE ENRIQUE
VEGA MARQUEZ, BELEN

## Objetivos y resultados del aprendizaje

OBJETIVOS:

Aprender una metodología para el diseño, implementación y documentación de programas.

Apreciar el papel central que juega la abstracción en la tarea de programar.

Conocer estructuras de datos, algoritmos y esquemas de uso general.

Introducir el paradigma (y aprender un lenguaje) de programación imperativa.

Introducir el paradigma (y aprender un lenguaje) de programación orientada a objetos.

COMPETENCIAS:

Competencias específicas:

E03, E04, E05

Competencias genéricas:

Esta asignatura no tiene asignadas competencias genéricas en la memoria de verificación.

## Contenidos o bloques temáticos

Parte I: Python

1. Introducción a Python. Expresiones y tipos básicos

2. Control de flujo y abstracción funcional

3. Estructuras de datos

4. Entrada/salida

Parte II: Java

1. Introducción al lenguaje Java

2. Diseño de tipos

3. Colecciones

4. Tratamientos secuenciales

## Relación detallada y ordenación temporal de los contenidos

Parte I: Python (1er cuatrimestre)

1. Introducción a Python (10 horas)

2. Expresiones, tipos predefinidos y entrada/salida (10 horas)

3. Instrucciones condicionales y bucles (10 horas)

4. Funciones (10 horas)

5. Secuencias, listas y tuplas (10 horas)

6. Diccionarios y conjuntos (10 horas)

Parte II: Java (2º cuatrimestre)

1. Introducción al lenguaje Java (18 horas)

2. Diseño de tipos (14 horas)

3. Colecciones (14 horas)

4. Tratamientos secuenciales (14 horas)

## Actividades formativas y horas lectivas

| Actividad | Horas |
|---|---|
| A Clases Teóricas | 72 |
| E Prácticas de Laboratorio | 24 |
| G Prácticas de Informática | 24 |

## Idioma de impartición del grupo

ESPAÑOL

## Sistemas y criterios de evaluación y calificación

Tal y como establece el artículo 6 de la normativa de la Universidad de Sevilla que regula la evaluación y calificación de las asignaturas, la evaluación de las competencias, conocimientos y capacidades adquiridas por los estudiantes podrán basarse en actividades de evaluación continua, exámenes parciales y/o exámenes finales. La asistencia a clases teóricas podrá puntuar de manera positiva en la calificación final. Además se podrán contemplar requisitos específicos, que deberán ser definidos en los proyectos docentes anuales, en relación a la realización de exámenes, a la realización de cualquier otro tipo de pruebas, a la obligatoriedad en la realización de trabajos, a la obligatoriedad a la asistencia a clases prácticas, a proyectos y a clases prácticas de laboratorio, así como a la participación en seminarios.

## Metodología de enseñanza-aprendizaje

Clases teóricas

Las clases de teoría son aquellas en las que el profesor expone los principales conceptos teóricos, correspondientes a los temas del programa. Se imparten en un aula de teoría con la pizarra como medio didáctico fundamental y con el apoyo del ordenador para mostrar esquemas algorítmicos complejos o código fuente. La labor del alumno en estas clases consistirá básicamente en trasladar a sus apuntes las principales ideas que el profesor transmita y preguntar las dudas que le puedan surgir.

Las clases de problemas se centran, sobre todo, en la resolución de ejercicios y ejemplos, una vez expuestos los conceptos sobre los cuales estos están basados. Los ejercicios consisten en la resolución de algoritmos mediante la aplicación de esquemas de creciente dificultad a lo largo del curso; dichos esquemas se describen en el lenguaje de programación que se utilizará posteriormente en las clases de laboratorio. La labor del alumno en estas clases consistirá básicamente en participar en la resolución de los ejercicios y problemas propuestos por el profesor.

Una vez en su casa, el alumno debería repasar la clase, comprendiendo los conceptos teóricos y repasando los ejercicios, por si hubiera alguna duda o alternativa en su solución. Estas dudas podrán ser planteadas al profesor en la siguiente clase o más tranquilamente en su horario de tutorías.

Cada tema del curso tendrá asociado uno o varios boletines de problemas. Algunos problemas de estos boletines se resolverán en clase de problemas, otros se resolverán en el laboratorio y el resto deberá realizarlos el alumno por su propia cuenta.

Prácticas de Laboratorio

Las clases de prácticas se realizan en un aula de laboratorio que dispone de ordenadores con el software necesario para implementar los programas. El grupo de clase se divide en dos o tres subgrupos de prácticas y cada subgrupo recibe docencia de un profesor.

El profesor comenzará la clase repasando los conceptos que se trabajarán en la misma. Seguidamente planteará y resolverá varios ejercicios del boletín de problemas con la participación de los alumnos.

## Horarios del grupo del proyecto docente

https://www.informatica.us.es/index.php/horarios

## Calendario de exámenes

https://www.informatica.us.es/index.php/calendario-de-examenes

## Tribunales específicos de evaluación y apelación

Presidente: VICENTE CARRILLO MONTERO
Vocal: IRENE BARBA RODRIGUEZ
Secretario: JUAN MANUEL CORDERO VALLE
Suplente 1: MARIA DEL MAR MARTINEZ BALLESTEROS
Suplente 2: CARMELO DEL VALLE SEVILLANO
Suplente 3: MANUEL RESINAS ARIAS DE REYNA

## Sistemas y criterios de evaluación y calificación del grupo

**Sistemas de evaluación**

Tal y como establece el artículo 6 de la normativa de la Universidad de Sevilla que regula la evaluación y calificación de las asignaturas, la evaluación de las competencias, conocimientos y capacidades adquiridas por los estudiantes podrán basarse en actividades de evaluación continua, exámenes parciales y/o exámenes finales. La asistencia a clases teóricas podrá puntuar de manera positiva en la calificación final. Además se podrán contemplar requisitos específicos, que deberán ser definidos en los proyectos docentes anuales, en relación a la realización de exámenes, a la realización de cualquier otro tipo de pruebas, a la obligatoriedad en la realización de trabajos, a la obligatoriedad a la asistencia a clases prácticas, a proyectos y a clases prácticas de laboratorio, así como a la participación en seminarios.

**Criterio de calificación**
PRIMERA CONVOCATORIA

==========

En la primera convocatoria existen dos sistemas de evaluación: la evaluación continua y la evaluación ordinaria.

1. Evaluación continua

La evaluación continua se compone de dos exámenes teóricos y un examen práctico por cada cuatrimestre. Para poder realizar el examen práctico de un cuatrimestre es necesario obtener una media de 4 o más puntos en los exámenes teóricos del cuatrimestre.

La nota de cada cuatrimestre se calcula ponderando en un 20% las notas de los exámenes teóricos y en un 80% la nota del examen práctico:

- Nota del cuatrimestre = 0,2 * nota media de teoría + 0,8 * nota del examen práctico

La nota de la evaluación continua es la media de ambos cuatrimestres (C1 y C2), siempre que se haya obtenido al menos un 4 en cada uno:

- Si C1 >= 4 y C2 >= 4, entonces NOTA_EV_CONT = (C1 + C2) / 2

- Si C1 < 4 o C2 < 4, entonces NOTA_EV_CONT = mínimo (4, (C1 + C2) / 2)

Si NOTA_EV_CONT es igual o superior a 5 puntos, la asignatura estará aprobada.

2. Evaluación ordinaria

Si el alumno no realiza la evaluación continua, o la realiza, pero no aprueba mediante ella, debe presentarse a un examen final, de carácter práctico, que se compone de dos partes, una por cuatrimestre. Para aprobar el examen debe obtenerse al menos un 4 en cada parte y un 5 en la media de ambas.

Si ha obtenido una nota mayor o igual a 4 en uno de los dos cuatrimestres de la evaluación continua, puede presentarse únicamente al otro cuatrimestre en el examen final.

La nota final es la media de ambas partes (F1 y F2), siempre que se haya obtenido al menos un 4 en cada una:

- Si F1 >= 4 y F2 >= 4, entonces NOTA_FINAL = (F1 + F2) / 2

- Si F1 < 4 o F2 < 4, entonces NOTA_FINAL = mínimo (4, (F1 + F2) / 2)

Si NOTA_FINAL es igual o superior a 5 puntos, la asignatura estará aprobada.

SEGUNDA CONVOCATORIA

==========

La evaluación de la segunda convocatoria consiste en un examen práctico compuesto por dos partes, una por cada cuatrimestre. Para aprobar el examen debe obtenerse al menos un 5 en cada parte.

Si el alumno obtuvo una nota mayor o igual que 5 en uno de los dos cuatrimestres en la primera convocatoria, en la segunda convocatoria puede presentarse únicamente a la parte correspondiente al cuatrimestre que no superó (con 5 ó más) en la primera convocatoria.

La nota de la segunda convocatoria es la media de ambas partes (S1 y S2), siempre que se haya obtenido al menos un 5 en cada una:

- Si S1 >= 5 y S2 >= 5, entonces NOTA_2CONV = (S1 + S2) / 2

- Si S1 < 5 o S2 < 5, entonces NOTA_2CONV = mínimo (4, (S1 + S2) / 2)

TERCERA CONVOCATORIA

==========

La evaluación de la tercera convocatoria es análoga a la de la segunda convocatoria, con la diferencia de que no se conservan notas de las partes obtenidas en convocatorias anteriores.

## Bibliografía recomendada

**Información Adicional**

- Lenguaje Python. Python Software Foundation. https://docs.python.org/3/

- Python 3: Los fundamentos del lenguaje (2ª edición). Sébastien Chazallet. Ediciones ENI, 2016. ISBN: 409-00614-2

- Lenguaje Java. ORACLE. http://www.oracle.com/technetwork/java/index.html

- Introducción a la Programación. Miguel Toro. https://sites.google.com/site/lsintroprogramacion/home

- Java Platform, Standard Edition Version 17 API Specification. ORACLE. https://docs.oracle.com/en/java/javase/17/docs/api/

- Objects First with Java: A Practical Introduction Using BlueJ, Global Edition (6ª Ed.). David J.

Barnes. Pearson, 2016. ISBN: 978-1292159041.

- Programación orientada a objetos con Java usando Blue J. David J. Barnes and Michael Kölling. Prentice-Hall, 2013. ISBN: 978-8483227916.