Proyecto docente de la asignatura

Asignatura	PARADIGMAS DE PROGRAMACIÓN		
Materia	ENTORNO SOFTWARE		
Módulo	COMUNES A LA INFORMÁTICA		
Titulación	GRADO EN INGENIERÍA INFORMÁTICA		
Plan	545	**Código**	46909
Periodo de impartición	2º CUATRIMESTRE	**Tipo/Carácter**	Complementos de Informática
Nivel/Ciclo	GRADO	**Curso**	1º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	César Vaca Rodríguez		
Datos de contacto (E-mail, teléfono…)	TELÉFONO: 983 423000 ext. 5620 E-MAIL: cvaca@infor.uva.es		
Horario de tutorías	Véase www.inf.uva.es >> Alumno >> Apoyo >> Tutorías		
Departamento	INFORMÁTICA (ATC, CCIA y LSI)		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

El objetivo principal de esta asignatura es, en su parte teórica, el proporcionar al alumno una visión general de los distintos paradigmas (imperativo, funcional, lógico) y técnicas (orientación a objetos, orientación a eventos, genericidad) de programación existentes. La parte práctica se basa en el aprendizaje del lenguaje Python, como ejemplo característico de lenguaje de scripting, discurriendo en paralelo la enseñanza teórica de las distintas técnicas y paradigmas de programación con su correspondiente implementación en Python.

1.2 Relación con otras materias

La asignatura introduce técnicas y conceptos que se ampliarán en otras asignaturas, particularmente **Interacción Persona-Computadora** (Orientación a Eventos), **Estructuras de Datos y Algoritmos** (TADs, Tipado Algebraico), **Análisis de Algoritmos, Lenguajes de Programación** y **Programación Orientada a Objetos**.

1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno posea conocimientos básicos de programación, en particular haber cursado y conseguido las habilidades y destrezas establecidas en la guía docente de la asignatura de **Fundamentos de Programación**. También es recomendable disponer de un nivel de inglés que permita al estudiante leer bibliografía de consulta.

2. Competencias

2.1 Generales

Código	Descripción
CG1	Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas
CG2	Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria
CG3	Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.
CG5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería de software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria.
CG6	Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria.
CG10	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria.

2.2 Específicas

Código	Descripción
CI8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

3. Objetivos

Código	Descripción
CI8.1	Comprender los distintos modelos de computación y paradigmas de programación.
CI8.2	Conocer la estructura de los lenguajes de programación y las diversas familias de lenguajes.
CI8.3	Conocer y saber utilizar lenguajes de scripting.
CI8.4	Conocer y ser capaz de interpretar las estructuras de los lenguajes de programación orientados a objeto y el contenido semántico de sus construcciones.
CI8.5	Razonar sobre las características de los distintos paradigmas de programación: imperativo, declarativo, funcional y lógico.

4. Bloques temáticos

Bloque 1: Paradigma Imperativo

Carga de trabajo en créditos ECTS: 2.0

a. Contextualización y justificación

En este bloque temático se proporcionará una visión general de las distintas técnicas y paradigmas de programación, los modelos de cómputo en que se basan y los distintos lenguajes de programación y su evolución histórica. También se examinará en detalle el paradigma imperativo y los paradigmas de programación estructurada, procedimental y modular, estrechamente asociados a él. Por último se contempla con cierto detalle las técnicas de tratamiento de excepciones existentes.

b. Objetivos de aprendizaje

Código	Descripción
CI8.1	Comprender los distintos modelos de computación y paradigmas de programación.
CI8.2	Conocer la estructura de los lenguajes de programación y las diversas familias de lenguajes.
CI8.3	Conocer y saber utilizar lenguajes de scripting.
CI8.5	Razonar sobre las características de los distintos paradigmas de programación: imperativo, declarativo, funcional y lógico.

c. Contenidos

PARTE TEÓRICA

Tema 1. Introducción

- Conceptos fundamentales
- Modelos de cómputo
- Lenguajes de programación
- Familias de lenguajes y evolución histórica

Tema 2. Paradigma Imperativo

- Programación estructurada, procedimental y modular
- Tratamiento de excepciones
- Continuations, Closures, Corutinas
- Sistemas de tipado
- Datos estructurados y referencias
- Fortaleza y seguridad en sistemas de tipado

PARTE PRÁCTICA

Tema 1. Introducción

Tema 2. Elementos básicos del lenguaje

Tema 3. Funciones

Tema 4. Datos Estructurados

d. Métodos docentes

Actividad	Metodología
Clase de teoría	• Clase magistral participativa • Estudio de casos en aula • Resolución de problemas
Clase práctica	• Clase magistral participativa • Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	• Casos prácticos guiados por el profesor

e. Plan de trabajo

Para éste bloque se estiman 20 horas presenciales distribuidas en 8 horas teóricas, 8 prácticas, 2 seminarios y 2 para efectuar las evaluaciones. El tiempo de dedicación no presencial del alumno medio es unas 30 horas.

f. Evaluación

La evaluación teórica de éste bloque se efectuará mediante una prueba escrita de 2 horas de duración y se realizará en la séptima semana o en la anterior o posterior.
La evaluación práctica consistirá en la entrega y defensa de un trabajo práctico (2 horas de duración) y se realizará en la novena semana o en la anterior o posterior.

g. Bibliografía básica

[Tucker] Tucker, A., Noonan, R., "Lenguajes de Programación. Principios y Paradigmas", Mc Graw-Hill, 1998.

h. Bibliografía complementaria

[Llamas] Llamas, C. "Introducción a la Informática. Modelos de Cómputo", Thomson, 2004.

i. Recursos necesarios

Material de apoyo en la web:
[Transparencias] http://www.infor.uva.es/~cvaca/asigs/paradigmas.html

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
2.0	5 semanas

Bloque 2: Orientación a Objetos y Eventos

Carga de trabajo en créditos ECTS: 2.0

a. Contextualización y justificación

En este bloque temático se proporcionará una introducción a las técnicas de programación orientada a objetos, programación orientada a eventos y genericidad. Se intentará proporcionar una visión general que resalte las interacciones existentes entre estas técnicas.

b. Objetivos de aprendizaje

Código	Descripción
CI8.4	Conocer y ser capaz de interpretar las estructuras de los lenguajes de programación orientados a objeto y el contenido semántico de sus construcciones.

c. Contenidos

PARTE TEÓRICA

Tema 3. Orientación a Objetos

- Conceptos fundamentales
- Estructura estática: Clases, Encapsulamiento, Herencia
- Estructura dinámica: Objetos, tipos de métodos
- Polimorfismo, Ligadura dinámica
- Representación y gestión de memoria en O.O.

Tema 4: Orientación a Eventos

- Arquitectura de un sistema orientado a eventos
- Interfaces Gráficas de Usuario
- Paso de mensajes y manejadores de eventos
- Técnicas: Callbacks, Orientación a Objetos

Tema 5. Genericidad

- Objetivos. Técnicas principales.
- Genericidad y O.O.: Clases parametrizadas

PARTE PRÁCTICA

Tema 5. Orientación a Objetos

Tema 6. Programación en GUI

d. Métodos docentes

Actividad	Metodología
Clase de teoría	• Clase magistral participativa • Estudio de casos en aula • Resolución de problemas
Clase práctica	• Clase magistral participativa • Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	• Casos prácticos guiados por el profesor

e. Plan de trabajo

Para éste bloque se estiman 20 horas presenciales distribuidas en 8 horas teóricas, 8 horas prácticas, 2 seminarios y 2 para efectuar evaluaciones. El tiempo de dedicación no presencial del alumno medio es unas 30 horas.

f. Evaluación

La evaluación teórica de éste bloque se efectuará mediante una prueba escrita de 2 horas de duración y se realizará en la onceava semana o en la anterior o posterior.
La evaluación práctica consistirá en la entrega y defensa de un trabajo práctico (2 horas de duración) y se realizará en la última semana o en la penúltima.

g. Bibliografía básica

[Tucker] Tucker, A., Noonan, R., "Lenguajes de Programación. Principios y Paradigmas", Mc Graw-Hill, 1998.

h. Bibliografía complementaria

[Meyer] Meyer, B. "Object-Oriented Software Construction" (2º ed.), Prentice-Hall, 1997.

i. Recursos necesarios

Material de apoyo en la web:
[Transparencias] http://www.infor.uva.es/~cvaca/asigs/paradigmas.html

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
2.0	5 semanas

Bloque 3: Paradigma Funcional

Carga de trabajo en créditos ECTS: 2.0

a. Contextualización y justificación

En este bloque temático se estudiará el paradigma funcional tomando como base el lenguaje de programación Haskell. Se estudiará el tipado algebraico como una alternativa a las técnicas de orientación a objetos y genericidad. Por último se presentará al alumno el esquema map/filter/reduce de resolución de problemas.
Por último se presentará al alumno el paradigma lógico y sus características tomando como base el lenguaje Prolog.

b. Objetivos de aprendizaje

Código	Descripción
CI8.1	Comprender los distintos modelos de computación y paradigmas de programación.
CI8.2	Conocer la estructura de los lenguajes de programación y las diversas familias de lenguajes.
CI8.5	Razonar sobre las características de los distintos paradigmas de programación: imperativo, declarativo, funcional y lógico.

c. Contenidos

PARTE TEÓRICA

Tema 6. Paradigma funcional

- Objetivos, conceptos y abstracciones fundamentales
- Funciones como elementos de primer orden
- Concordancia de patrones y Evaluación diferida
- Tipado algebraico y genericidad
- Proceso de Listas: Map/Filter/Folder (Reduce)

PARTE PRÁCTICA

Tema 7. Elementos de programación funcional

d. Métodos docentes

Actividad	Metodología
Clase de teoría	• Clase magistral participativa • Estudio de casos en aula • Resolución de problemas
Clase práctica	• Clase magistral participativa • Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	• Casos prácticos guiados por el profesor

e. Plan de trabajo

Para éste bloque se estiman 20 horas presenciales distribuidas en 10 horas teóricas, 8 horas prácticas y 2 de seminarios. El tiempo de dedicación no presencial del alumno medio es unas 30 horas.

f. Evaluación

La evaluación teórica de éste bloque se efectuará en el examen final de la asignatura.

g. Bibliografía básica

[Tucker] Tucker, A., Noonan, R., "Lenguajes de Programación. Principios y Paradigmas", Mc Graw-Hill, 1998.

h. Bibliografía complementaria

[Bird] Bird, R. "Introduction to Functional Programming using Haskell", (2º ed.), Prentice-Hall, 1998.
[Cloksin] Clocksin, W.F., Mellish, C.S., "Programming in Prolog", Springer-Verlag, 1994.
[Arenas] Arenas, A., "Lógica Formal para Informáticos", Ed. Díaz de Santos, Madrid, 1996.

i. Recursos necesarios

Material de apoyo en la web:
[Transparencias] http://www.infor.uva.es/~cvaca/asigs/paradigmas.html
[Haskell] http://www.lcc.uma.es/~blas/pfHaskell/gentle
[SICP] http://mitpress.mit.edu/sicp/full-text/book/book.html

j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
2.0	5 semanas

5. Métodos docentes y principios metodológicos

Actividad	Metodología
Clase de teoría	• Clase magistral participativa • Estudio de casos en aula • Resolución de problemas
Clase práctica	• Clase magistral participativa • Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	• Casos prácticos guiados por el profesor

6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	28	Estudio y trabajo autónomo individual	70
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	20
Laboratorios (L)	22		
Prácticas externas, clínicas o de campo			
Seminarios (S)	6		
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	4		
Total presencial	**60**	**Total no presencial**	**90**

7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen teórico bloque 1 (T1 ∈ [0,2])	20 %	Aproximadamente 7ª Semana. Recuperable en los exámenes ordinario y extraordinario.
Entrega primera práctica (P1 ∈ [0,1.5])	15 %	Aproximadamente 9ª Semana. No recuperable.
Examen teórico bloque 2 (T2 ∈ [0,2])	20 %	Aproximadamente 11ª Semana. Recuperable en los examen extraordinario
Entrega segunda práctica (P2 ∈ [0,1.5])	15 %	Aproximadamente 15ª Semana. No recuperable.
Examen final (T3 ∈ [0,3] ó T3+T1 ó T3+T2 ó T3+T2+T1)	30-70 %	En el examen final se evaluará la parte teórica correspondiente al tercer bloque de la asignatura, pero existirán apartados opcionales para sustituir la calificación obtenida en los exámenes teóricos parciales No se establece nota mínima en las pruebas anteriores para presentarse al examen final.

Procedimientos y Sistemas de Evaluación

Los exámenes teóricos (incluido el final) consistirán en preguntas tipo test y preguntas de respuesta corta. En los exámenes finales se incluirán apartados opcionales para la reevaluación de la calificación obtenida en los exámenes teóricos del primer y segundo bloque (y del tercero en el examen final extraordinario).
La parte práctica consistirá en la resolución de problemas mediante la obtención del código Python adecuado. Además de la corrección del código, se valorará la utilización de las técnicas contempladas en la asignatura.

CRITERIOS DE CALIFICACIÓN
• **Convocatoria ordinaria:** o La nota es la suma T1+P1+T2+P2+T3 y se considera aprobado si supera o iguala el valor 5.0 o En el examen final existirán apartados opcionales para sustituir la calificación obtenida en los exámenes teóricos parciales • **Convocatoria extraordinaria:** o Igual que en la convocatoria ordinaria, salvo que si T1+P1+T2+P2+T3 < 5.0 pero se cumple que T1+T2+T3 > 3.5 entonces la nota será 5.0 (aprobado)

8. Consideraciones finales

Las semanas indicadas para las evaluaciones son orientativas, pudiendo sufrir cambios (semana anterior o siguiente) por mejor acomodación del calendario.
En la página web principal de la asignatura se indicará el cronograma de actividades de la asignatura con sus fechas definitivas.
Página web de la asignatura: http://www.infor.uva.es/~cvaca/asigs/paradigmas.html