

**Proyecto/Guía docente de la asignatura**

<b>Asignatura</b>	PARADIGMAS DE PROGRAMACIÓN		
<b>Materia</b>	ENTORNO SOFTWARE		
<b>Módulo</b>	COMUNES A LA INFORMÁTICA		
<b>Titulación</b>	GRADO EN INGENIERÍA INFORMÁTICA		
<b>Plan</b>	545	<b>Código</b>	46909
<b>Periodo de impartición</b>	2º CUATRIMESTRE	<b>Tipo/Carácter</b>	Complementos de Informática
<b>Nivel/Ciclo</b>	GRADO	<b>Curso</b>	1º
<b>Créditos ECTS</b>	6 ECTS		
<b>Lengua en que se imparte</b>	CASTELLANO		
<b>Profesor/es responsable/s</b>	César Vaca Rodríguez		
<b>Datos de contacto (E-mail, teléfono...)</b>	TELÉFONO: 983 423000 ext. 5620 E-MAIL: <a href="mailto:cvaca@infor.uva.es">cvaca@infor.uva.es</a>		
<b>Horario de tutorías</b>	Véase <a href="http://www.uva.es">www.uva.es</a> → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías		
<b>Departamento</b>	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, discurrendo 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
G02	Conocimientos básicos de la profesión
G03	Capacidad de análisis y síntesis
G04	Capacidad de organizar y planificar
G05	Comunicación oral y escrita en la lengua propia
G08	Habilidades de gestión de la información
G09	Resolución de problemas
G10	Toma de decisiones
G11	Capacidad crítica y autocrítica
G12	Trabajo en equipo
G14	Responsabilidad y compromiso ético
G15	Liderazgo
G16	Capacidad de aplicar los conocimientos en la práctica
G18	Capacidad de aprender
G19	Capacidad de adaptarse a nuevas situaciones
G20	Capacidad de generar nuevas ideas
G21	Habilidad para trabajar de forma autónoma

### 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.
CI14	Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

## 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.
CI14.1	Razonar sobre las características de los distintos paradigmas de programación: imperativo, declarativo, funcional y lógico.
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.



#### 4. Contenidos y/o 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.
CI14.1	Razonar sobre las características de los distintos paradigmas de programación: imperativo, declarativo, funcional y lógico.
CI8.3	Conocer y saber utilizar lenguajes de scripting.

##### 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	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• Estudio de casos en aula</li><li>• Resolución de problemas</li></ul>
Clase práctica	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• 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.</li></ul>
Seminarios	<ul style="list-style-type: none"><li>• Casos prácticos guiados por el profesor</li></ul>

#### 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 este 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.

#### g1. Bibliografía básica

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

#### g2. Bibliografía complementaria

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

#### g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)

Consultar Aula Virtual de la asignatura

#### h. Recursos necesarios

Material de apoyo en la web:

[Transparencias] <http://www.infor.uva.es/~cvaca/asigs/paradigmas.html>

#### i. 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
CI8.5	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	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• Estudio de casos en aula</li><li>• Resolución de problemas</li></ul>
Clase práctica	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• 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.</li></ul>
Seminarios	<ul style="list-style-type: none"><li>• Casos prácticos guiados por el profesor</li></ul>

#### e. Plan de trabajo

Para este 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.

#### g1. Bibliografía básica

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

#### g2. Bibliografía complementaria

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

#### g3. Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)

Consultar Aula Virtual de la asignatura

#### h. Recursos necesarios

Material de apoyo en la web:

[Transparencias] <http://www.infor.uva.es/~cvaca/asigs/paradigmas.html>

#### i. 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.
CI14.1	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
<b>Clase de teoría</b>	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• Estudio de casos en aula</li><li>• Resolución de problemas</li></ul>
<b>Clase práctica</b>	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• 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.</li></ul>
<b>Seminarios</b>	<ul style="list-style-type: none"><li>• Casos prácticos guiados por el profesor</li></ul>



### e. Plan de trabajo

---

Para este 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.

### g1. Bibliografía básica

---

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

### g2. Bibliografía complementaria

---

[Bird] Bird, R. "Introduction to Functional Programming using Haskell", (2º ed.), Prentice-Hall, 1998.

[Clocksin] 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.

### g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)

---

Consultar Aula Virtual de la asignatura

### h. 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>

### i. Temporalización

---

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
2.0	5 semanas

**5. Métodos docentes y principios metodológicos**

Actividad	Metodología
<b>Clase de teoría</b>	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• Estudio de casos en aula</li><li>• Resolución de problemas</li></ul>
<b>Clase práctica</b>	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• 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.</li></ul>
<b>Seminarios</b>	<ul style="list-style-type: none"><li>• Estudio e implementación de casos prácticos</li></ul>

**6. Tabla de dedicación del estudiante a la asignatura**

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA <sup>(1)</sup>	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	4		
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>
TOTAL presencial + no presencial			<b>150</b>

(1) Actividad presencial a distancia es cuando un grupo sigue una videoconferencia de forma síncrona a la clase impartida por el profesor para otro grupo presente en el aula.

## 7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen teórico bloque 1 (T1)	20 %	Aproximadamente 7ª Semana. Recuperable en el examen extraordinario.
Entrega primera práctica (P1)	15 %	Aproximadamente 9ª Semana. No recuperable.
Examen teórico bloque 2 (T2)	20 %	Aproximadamente 11ª Semana. Recuperable en el examen extraordinario
Entrega segunda práctica (P2)	15 %	Aproximadamente 15ª Semana. No recuperable.
Examen teórico bloque 3 (T3)	30 %	En la fecha del examen ordinario. Recuperable en el examen extraordinario

### Procedimientos y Sistemas de Evaluación

Los exámenes teóricos consistirán en preguntas tipo test, preguntas de respuesta corta y la resolución de problemas sencillos.

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:**
  - Se realizará mediante evaluación continua mediante las pruebas mencionadas en el apartado anterior. No se establece nota mínima en las pruebas para su contabilización.
  - La nota obtenida consistirá en la suma ponderada de las calificaciones obtenidas en los exámenes teóricos parciales y las prácticas.
  - Se considerará aprobado si  $T1+P1+T2+P2+T3 \geq 5.0$
- **Convocatoria extraordinaria:**
  - Consistirá en un único examen en el que se podrá reevaluar cualquier combinación de los 3 bloques teóricos de la asignatura, a elección del alumno.
  - La parte práctica no se reevaluará.
  - Se considerará aprobado si  $T1+P1+T2+P2+T3 \geq 5.0$

## 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>