

**40963 - TECNOLOGÍAS DE
PROGRAMACIÓN**

CENTRO: 180 - Escuela de Ingeniería Informática

TITULACIÓN: 4008 - Grado en Ingeniería Informática

ASIGNATURA: 40963 - TECNOLOGÍAS DE PROGRAMACIÓN

CÓDIGO UNESCO: **TIPO:** Obligatoria **CURSO:** 2 **SEMESTRE:** 1º semestre

CRÉDITOS ECTS: 6 **Especificar créditos de cada lengua:** **ESPAÑOL:** 6 **INGLÉS:**

SUMMARY

Programming Technologies is a course of the Programming subject, attached to the Software Development Engineering module, common to the branch of Computer Science Engineering. It is not, however, the student's first contact with programming.

This first contact occurs in the courses indicated as prerequisites. These courses provide the student with training in the imperative programming approach, using a general-purpose, imperative, structured, object-oriented, and weakly typed programming language, and also offer an introduction to a strongly typed language.

Programming Technologies delves into the subject using the object-oriented programming model as its axis, using a language that illustrates the main elements of this paradigm: classes, objects, inheritance, and polymorphism, and that also incorporates event-driven programming for GUI and concurrent programming.

REQUISITOS PREVIOS

- Fundamentos de Programación I
- Fundamentos de Programación II

Plan de Enseñanza (Plan de trabajo del profesorado)

Contribución de la asignatura al perfil profesional:

Tecnologías de Programación es una asignatura de la materia Programación, adscrita al módulo de Ingeniería de Desarrollo de Software, común a la rama de la Ingeniería Informática. No es, sin embargo, el primer contacto del estudiante con la programación.

Este primer contacto se produce en las asignaturas de primero indicadas como requisitos previos. Estas asignaturas proporcionan al alumno formación en el modelo de programación imperativa, utilizando un lenguaje de programación de propósito general, imperativo, estructurado, orientado a objetos y débilmente tipado. Además, ofrecen una introducción a un lenguaje fuertemente tipado.

Tecnologías de Programación profundiza en la materia usando como eje el modelo de programación orientado a objetos, empleando para ello un lenguaje que ilustra de forma sencilla los principales elementos de este paradigma: clases, objetos, herencia y polimorfismo y que también incorpora programación guiada por eventos y concurrencia.

Competencias que tiene asignadas:

CB1, CB5
G3, G4, G8, G9
N2, N3
CI1, CI8, CI11, CI14, CI17

Objetivos:

Que el alumno adquiera destreza en:

Ob1) Aplicar el paradigma de programación orientada a objetos en el desarrollo de programas.

Ob2) Utilizar técnicas avanzadas de programación orientada a objetos como uso y desarrollo de contenedores.

Ob3) Aplicar en el desarrollo de programas el modelo de programación guiada por eventos de forma adecuada y construir interfaces gráficas de usuario utilizando principios básicos de diseño.

Ob4) Diseñar y desarrollar aplicaciones concurrentes y distribuidas.

Contenidos:

1. Conceptos avanzados de Programación Orientada a Objetos
1.1. Revisión de conceptos fundamentales de Programación Orientada a Objetos
1.2. Conceptos avanzados de modularidad, herencia y reusabilidad
1.3. Genericidad; contenedores como ejemplo de aplicación
Bibliografía: [1] [2] [3]
Competencias: CB1, CB5, G3, G4, G8, G9, N2, N3, CI1, CI8
Resultados de aprendizaje: RA1, RA2

2. Fundamentos de la Programación Guiada por Eventos e Interfaz Gráfica
2.1. Introducción, gestores de eventos y clases anidadas
2.2. Interfaz gráfica de usuarios
Bibliografía: [1] [2] [3]
Competencias: CB1, CB5, G3, G4, G8, G9, N2, N3, CI1, CI8, CI17
Resultados de aprendizaje: RA3

3. Programación Concurrente y Distribuida.
3.1. Introducción al paradigma de la programación concurrente.
3.2. Introducción a la programación distribuida.
Bibliografía: [1][2][4]
Competencias: CB1, CB5, G3, G4, G8, G9, N2, N3, CI1, CI8, CI11, CI14
Resultados de aprendizaje: RA4

Metodología:

Se aplicará, preferentemente, una metodología mixta de tipo blended learning, combinando el trabajo presencial, en aula y laboratorio, con el trabajo en línea.

Todos los materiales de estudio y las actividades para realizar por el estudiante estarán disponibles en línea, distinguiéndose entre actividades asíncronas, que el estudiante realizará a su propio ritmo, dentro de un periodo temporal prefijado, y actividades síncronas, que deberá realizar en un

momento temporal concreto, bajo supervisión del equipo docente y, preferentemente, de manera presencial en un laboratorio docente.

El trabajo en línea se complementará con clases en aula y laboratorio, en las que se reforzarán los contenidos principales, se resolverán dudas, se intercambiarán ideas, y se supervisarán las actividades sincrónicas.

Además, se dispondrá de tutorías, individuales o en grupo, para resolver dudas y proporcionar refuerzo adicional a quienes lo necesiten. Estas tutorías, aunque podrán ser presenciales, se realizarán preferentemente usando medios electrónicos.

Actividades formativas:

AF1 Sesiones académicas de fundamentación (RA1, RA2, RA3, RA4)

AF2 Sesiones académicas de interacción (RA1, RA2, RA3, RA4)

AF3 Sesiones académicas de aplicación (RA1, RA2, RA3, RA4)

AF4 Sesiones de tutorización (RA1, RA2, RA3, RA4)

AF5 Estudio (RA1, RA2, RA3, RA4)

Evaluación:

Criterios de evaluación

En el proceso de evaluación de la asignatura se emplearán las siguientes fuentes de evaluación:

FE1) Exámenes sincrónicos individuales. Esta fuente de evaluación está relacionada con las actividades formativas AF1, AF2, AF3 y AF4.

FE2) Trabajos prácticos: individuales y/o en equipo. Esta fuente de evaluación está relacionada con las actividades formativas AF1, AF2, AF3 y AF4.

FE1) Exámenes sincrónicos individuales

* Criterios relativos a las cuestiones:

- Conocimiento y comprensión de los contenidos
- Rigor y fluidez en la comunicación escrita

* Criterios relativos al desarrollo de un programa:

Los ejercicios de programación se evaluarán atendiendo al grado de cumplimiento de los siguientes criterios:

- La solución propuesta es operativa (no contiene errores que impidan su ejecución) y resuelve correcta y completamente el problema planteado.
- La solución propuesta es la mejor de las soluciones posibles, en cuanto a su funcionamiento, en el contexto formativo de la asignatura.
- La solución propuesta es acorde con los criterios de calidad del código propuestos por la asignatura.

FE2) Trabajos prácticos: individuales y en equipo

* Criterios relativos a una práctica:

-El profesor someterá a los estudiantes a través de trabajos o a cuestionarios de clase. Los cuestionarios o programas entregados, serán sometidos a pruebas oportunas para formular una valoración global que tenga en cuenta todos los aspectos implicados en su realización, con especial énfasis en la ejecución (funcionamiento, adecuación a las especificaciones, robustez, ...), en el

estilo (formato, comentarios, elección de identificadores, ...) y en la autoría. Cuando lo estime conveniente, el profesor podrá citar a los alumnos para formularles cuestiones que considere relevantes para la valoración global.

Sistemas de evaluación

Los diferentes sistemas de evaluación de la asignatura se realizarán atendiendo a las distintas fuentes de evaluación:

Para la convocatoria ordinaria se usarán las pruebas síncronas (FE1) realizadas a lo largo del periodo docente (No habrá un examen específico de convocatoria); en cada prueba de las que componen la fuente FE1 se especificará la ponderación de la prueba respecto al total de la fuente FE1. Además, de un trabajo práctico obligatorio que se llevará a cabo a través de la plataforma de aprendizaje.

Para las convocatorias extraordinaria y especial, habrá una única prueba síncrona presencial compuesta por varios ejercicios y cuestiones.

Para todas las convocatorias (ordinaria, extraordinaria y especial) se usarán las pruebas correspondientes a los trabajos prácticos (FE2) realizados durante el periodo de docencia, no existiendo la posibilidad de realizar nuevas entregas para las convocatorias extraordinaria y especial.

Criterios de calificación

En cada prueba (Pi) de las que integran la fuente FE1 se especificará la ponderación asignable a los ejercicios que la componen. Cada prueba Pi se calificará entre 0 y 10, debiendo obtenerse un mínimo de 5 puntos en cada Pi, para considerarla superada. Los aspectos relacionados con trabajos prácticos (FE2) serán calificados entre 0 y 10, debiendo obtenerse un mínimo de 5 puntos, en cada uno, para considerarlos superados, y tendrán su correspondiente peso en la nota final de la asignatura.

De tal forma, esta nota final (NF) será calculada de la siguiente manera:

$$NF = FE1 * 0.75 + FE2 * 0.25$$

La evaluación de la asignatura se realizará en la convocatoria ordinaria únicamente de forma continua.

La calificación en las convocatorias extraordinaria y especial se computará de la misma forma, siempre que el estudiante haya aprobado alguna de las pruebas.

Plan de Aprendizaje (Plan de trabajo de cada estudiante)

Tareas y actividades que realizará según distintos contextos profesionales (científico, profesional, institucional, social)

Ta1) Búsqueda, estudio y consulta de información

Ta2) Realización de ejercicios de programación autoevaluables.

Ta3) Realización de prácticas de programación individuales

Ta4) Realización de prácticas de programación en equipo que requieren del desarrollo individual para una posterior discusión colectiva.

Ta5) Uso del entorno de desarrollo integrado para desarrollo de aplicaciones con interfaz gráfica y aplicaciones concurrentes

Científico: Ta1, Ta2, Ta3, Ta4, Ta5
Profesional: Ta1, Ta2, Ta3, Ta4, Ta5
Institucional:
Social: Ta1, Ta4

Temporalización semanal de tareas y actividades (distribución de tiempos en distintas actividades y en presencialidad - no presencialidad)

PRESENCIAL

Los estudiantes dedicarán 2 horas semanales a atender a sesiones académicas teóricas y otras 2 horas a sesiones académicas prácticas, que incluirán Ta2, Ta3, Ta4 y Ta5 (60 horas en 15 semanas). Los exámenes que se realizarán a lo largo del semestre consumen 7.5 horas en total.

NO PRESENCIAL

En promedio, realizarán cada semana 5 horas de trabajo autónomo: 2 horas dedicadas a la consulta y asimilación de materiales documentales (Ta1) y 3 horas para la realización de ejercicios y trabajos; ello supone 75 horas en 15 semanas.

Previsión de dedicación temporal en la relación de prácticas y actividades formativas:

- Semanas 1 a 3. Se realizarán Ta2 y Ta3.
- Semanas 4 a 7. Se realizarán Ta2 y Ta4.
- Semanas 8 a 11. Se realizarán Ta2 y Ta4 y Ta5.
- Semanas 12 a 15. Se realizarán Ta2, Ta4.

Las horas restantes las distribuirá el estudiante, en función de sus necesidades particulares, entre atención tutorial individual, tiempo adicional de estudio y realización de ejercicios y trabajos.

PLAN DE CONTINGENCIAS NO PRESENCIAL

En caso de que la enseñanza de esta asignatura tuviera que pasar por causa de fuerza mayor a modalidad no presencial, se seguirá este mismo proyecto docente, sustituyendo las actividades presenciales por sus equivalentes telemáticos.

Recursos que tendrá que utilizar adecuadamente en cada uno de los contextos profesionales.

Los recursos que utilizará el estudiante son los siguientes:

- Re1) Navegador web
- Re2) Moodle con VPL (Herramientas del campus virtual de la ULPGC, herramienta de programación online)
- Re3) Herramientas de desarrollo integrado (Java, interfaz gráfica en Java)

CONTEXTOS

Científico: Re1, Re2, Re3

Profesional: Re1, Re2, Re3
Institucional:
Social: Re1

Resultados de aprendizaje que tendrá que alcanzar al finalizar las distintas tareas.

El estudiante deberá ser capaz de:

RA1) Desarrollar programas usando elementos avanzados de la programación orientada a objetos. Se adquiere con las actividades formativas AF1, AF2 y AF3.

RA2) Utilizar programación guiada por eventos para gestionar la interacción con los usuarios mediante interfaces gráficas. Se adquiere con las actividades formativas AF1 y AF2.

RA3) Construir interfaces gráficas de usuario utilizando componentes visuales comunes y principios básicos de diseño. Se adquiere con las actividades formativas AF1 y AF2.

RA4) Solucionar problemas mediante técnicas básicas de programación concurrente y programación distribuida. Se adquiere con las actividades formativas AF1 y AF2.

Alcanzar cada uno de estos resultados de aprendizaje puede requerir, según las dificultades que surjan durante su adquisición por parte de los alumnos, de la actividad formativa AF4.

Plan Tutorial

Atención presencial individualizada (incluir las acciones dirigidas a estudiantes en 5ª, 6ª y 7ª convocatoria)

La atención individualizada se verificará en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas.

No procede la atención individualizada de los estudiantes en 5ª, 6ª y 7ª convocatoria al ser una asignatura de nueva implantación.

Atención presencial a grupos de trabajo

La atención a grupos de trabajo se verificará en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas.

Atención telefónica

La atención telefónica se verificará en los teléfonos del profesorado publicados en la web del Departamento de Informática y Sistemas, en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas.

Atención virtual (on-line)

La atención virtual (on-line) se llevará a cabo usando las herramientas disponibles en el Campus Virtual de la ULP GC. Se responderá a las consultas virtuales en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas.

Datos identificativos del profesorado que la imparte.

Datos identificativos del profesorado que la imparte

Dr./Dra. David Sebastián Freire Obregón

(COORDINADOR)

Departamento: 260 - INFORMÁTICA Y SISTEMAS

Ámbito: 570 - Lenguajes Y Sistemas Informáticos

Área: 570 - Lenguajes Y Sistemas Informáticos

Despacho: INFORMÁTICA Y SISTEMAS

Teléfono: **Correo Electrónico:** david.freire@ulpgc.es

Dr./Dra. Ignacio José López Rodríguez

Departamento: 260 - INFORMÁTICA Y SISTEMAS

Ámbito: 570 - Lenguajes Y Sistemas Informáticos

Área: 570 - Lenguajes Y Sistemas Informáticos

Despacho: INFORMÁTICA Y SISTEMAS

Teléfono: **Correo Electrónico:** ignaciojose.lopez@ulpgc.es

Bibliografía

[1 Básico] Piensa en Java /

Bruce Eckel.

Prentice Hall,, Madrid [etc.] : (2007) - (4ª ed.)

9788489660342

[2 Recomendado] Programador Java certificado: curso práctico /

Antonio J. Martín

Sierra.

Ra-Ma,, Paracuellos de Jarama, Madrid : (2010) - (3ª ed.)

978-84-7897-972-1

[3 Recomendado] Principles of concurrent and distributed programming /

M. Ben-Ari.

Pearson Education,, Harlow, England : (2006) - (2nd ed.)

9780321312839