



CENTRO: 180 - Escuela de Ingeniería Informática

TITULACIÓN: 4008 - Grado en Ingeniería Informática

ASIGNATURA: 40813 - PROGRAMACIÓN I

Vinculado a : (Titulación - Asignatura - Especialidad)

4801-Doble Grado en Ingeniería Informática y - 48113-PROGRAMACIÓN I - 00

4801-Doble Grado en Ingeniería Informática y - 48315-PROGRAMACIÓN I - 00

5038-M.U. Ingeniería Informática - 50978-PROGRAMACIÓN I - 22

5038-M.U. Ingeniería Informática - 50978-PROGRAMACIÓN I - 32

5038-M.U. Ingeniería Informática - 50978-PROGRAMACIÓN I - 42

CÓDIGO UNESCO: 1203

TIPO: Obligatoria

CURSO: 2

SEMESTRE: 1º semestre

CRÉDITOS ECTS: 6

Especificar créditos de cada lengua:

ESPAÑOL: 6

INGLÉS:

SUMMARY

REQUISITOS PREVIOS

- Introducción a la Informática
- Fundamentos de Programación

Plan de Enseñanza (Plan de trabajo del profesorado)

Contribución de la asignatura al perfil profesional:

Programación I es la primera 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 la materia Informática, adscrita al módulo de Fundamentos Básicos, que se imparte en el primer curso.

La materia Informática proporciona al alumno una formación sólida en el modelo de programación imperativa, utilizando un lenguaje de programación de propósito general, imperativo, estructurado y fuertemente tipado.

Por lo tanto, Programación I es una asignatura que 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 los elementos de la programación guiada por eventos, ilustrada esta última con su aplicación al desarrollo de interfaces gráficas. Además, se introducen los conceptos de tipos abstracto de datos y contenedor como elementos útiles en la resolución de gran número de problemas.

Asimismo, se inicia al estudiante en las técnicas de prueba de programas utilizando paquetes de apoyo específicos.

Competencias que tiene asignadas:

G1, G2, G3, G5, N4, T3, T7, T8, T9, CII05, CII08.

Objetivos:

- Ob1) Aplicar el paradigma de programación orientada a objetos en el desarrollo de programas y usar herramientas específicas de apoyo.
- Ob2) Probar programas utilizando pruebas de unidades.
- Ob3) Utilizar contenedores de forma apropiada en el desarrollo de programas.
- Ob4) Aplicar el modelo de programación guiada por eventos de forma adecuada en el desarrollo de programas.

Contenidos:

CONTENIDOS TEÓRICOS

- 1. Programación Orientada a Objetos
 - 1.1. Introducción a la Programación Orientada a Objetos
 - 1.2. Clases, objetos
 - 1.3. Modularidad
 - 1.4. Herencia
 - 1.5. Polimorfismo
 - 1.6. Clases abstractas e interfaces
 - 1.7. Documentación de programas
 - 1.8. Introducción a UML

Bibliografía: [1] [2] [3] [5]

- 2. Robustez y prueba de programas
 - 2.1. Aserciones
 - 2.2. Excepciones
 - 2.3. Técnicas de prueba de programas
 - 2.4. Diseño de casos de prueba
 - 2.5. Pruebas unitarias

Bibliografía: [1] [4] [6] [7]

- 3. Genericidad y contenedores
 - 3.1. Clases y métodos genéricos
 - 3.2. Clases contenedoras

Bibliografía: [1] [3]

- 4. Fundamentos de Programación Guiada por Eventos

- 5.1. Introducción
- 5.1. Gestores de eventos
- 5.2. Interfaz gráfica de usuarios

Bibliografía: [1] [3]

CONTENIDOS PRÁCTICOS

- 1. Clases, métodos y constructores
- 2. Prueba de programas

3. Herencia y polimorfismo
4. Contenedores
5. Interfaz gráfica de usuarios

Metodología:

La metodología a desarrollar incluirá las siguientes actividades formativas:

AF1) Sesiones académicas teóricas. El profesor expondrá aspectos teóricos relevantes de los contenidos de la asignatura que capacitan al estudiante para la realización de las actividades prácticas, dando respuesta colectiva a las dudas que puedan surgir durante el estudio personal de estos.

AF2) Sesiones académicas prácticas. El profesor expondrá las actividades prácticas a realizar, mostrando el uso de las herramientas necesarias para su realización y guiando en las dudas que se planteen en su desarrollo.

AF3) Ejercicios de autoevaluación. Se pondrá a disposición de los estudiantes ejercicios que permitan una autoevaluación sencilla de las habilidades adquiridas. Estas actividades se pueden realizar de forma no presencial.

AF4) Tutorías. Se usarán para resolver dudas y hacer un seguimiento de las actividades.

Evaluación:

Criterios de evaluación

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

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

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

FE3) Coevaluación. Esta fuente de evaluación está relacionada con las actividades formativas AF2 y AF4.

FE1) Exámenes presenciales

* 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:

- El profesor someterá el programa a las pruebas oportunas para formular una valoración que tenga en cuenta todos los aspectos implicados en su realización, con especial énfasis en la ejecución (funcionamiento, adecuación a las especificaciones, robustez, ...) y en el estilo (formato, comentarios, elección de identificadores, ...).

FE2) Trabajos prácticos: individuales y en equipo

* Criterios relativos a una práctica:

- El profesor someterá los programas e información entregados a los análisis y pruebas oportunos 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.

FE3) Coevaluación

Cada miembro del equipo valora a sus compañeros teniendo en cuenta su grado de: colaboración, respeto, integración, distribución del trabajo, motivación, compromiso y conciliación. El profesor supervisará la apropiada y justa evaluación entre los miembros de los equipos.

Sistemas de evaluación

1) Convocatoria ordinaria:

La evaluación de la asignatura se realizará en esta convocatoria únicamente de forma continua, atendiendo a las distintas fuentes de evaluación:

* Exámenes presenciales individuales (FE1)

La escala que se utilizará para valorar esta parte es de 0 a 10 puntos y se efectuará a partir de exámenes individuales que se irán realizando regularmente, este sistema se basa en tests y/o cuestiones de respuesta corta y/o desarrollo de programas relativos a los temas tratados. Cada examen se valora de 0 a 10 puntos aunque pondera en la nota FE1 según los temas evaluados (25% cada tema).

* Trabajos prácticos: individuales y en equipo (FE2)

La escala que se utilizará para valorar el trabajo práctico es de 0 a 10 puntos y su evaluación se efectuará a partir de las prácticas de laboratorio. El enunciado de cada práctica incluirá las especificaciones, plazos de entrega y porcentaje que aporta a la evaluación; asimismo, se indicará el tipo de práctica: individual o en equipo. Se deberá entregar, dentro de los plazos establecidos, los ficheros conteniendo el código fuente y demás información requerida, usando el medio que se establezca. Para los trabajos en equipo, la nota obtenida en cada práctica es la que le corresponderá a cada alumno componente del equipo que la realiza.

* Coevaluación (FE3)

Cada alumno interviene en la evaluación al calificar la participación activa en el trabajo colaborativo de sus compañeros de equipo por medio de un cuestionario que se valorará de 0 a 10 puntos. La nota de un alumno por cada coevaluación realizada será la media de la otorgada por sus compañeros de equipo. La nota final de esta parte será la media de los resultados obtenidos en las coevaluaciones efectuadas a lo largo del semestre y se valorará de 0 a 10 puntos. Requiere la asistencia a las sesiones académicas, con un mínimo del 80%. Los resultados de la coevaluación serán supervisados por el profesor para evitar posibles disfunciones en la nota.

2) Convocatorias extraordinaria y especial:

Se realizará un examen presencial de convocatoria (FE1). El examen podrá estar dividido en tests y/o cuestiones de respuesta corta y/o desarrollo de programas, cada uno de los cuales aportará un porcentaje especificado a la nota. Se valorará de 0 a 10 puntos.

Criterios de calificación

La calificación en la evaluación se obtendrá (en todas las convocatorias) aplicando la siguiente fórmula:

$$NF = FE1 * 0.7 + FE2 * 0.2 + FE3 * 0.1$$

donde:

- NF representa la nota final de la convocatoria.
- FE1 representa la nota de los exámenes presenciales individuales, o la nota del examen presencial de convocatoria.
- FE2 representa la nota de los trabajos prácticos.
- FE3 representa la nota de la coevaluación.

Para superar la asignatura es requisito obtener una nota FE1 en la evaluación continua y NF mínimas de 5. La nota FE1 podrá ser sustituida por la obtenida en el examen de convocatoria (extraordinaria y especial).

El cálculo de la nota final de los alumnos que no cumplan las condiciones establecidas para superar la asignatura se obtendrá aplicando la misma fórmula pero considerando que las notas FE2 y FE3 valen cero.

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 (1ª semana) para una posterior discusión colectiva (2ª semana)
- Ta5) Uso del entorno de desarrollo integrado para desarrollo de aplicaciones con interfaz gráfica

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)

PRESENCIALES

Los estudiantes dedicarán 2 horas semanales a asistir 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 presenciales en total.

NO PRESENCIALES

En promedio, realizarán cada semana 5 horas de trabajo no presencial: 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 12. Se realizarán Ta2 y Ta4.
- Semanas 13 a 15. Se realizarán Ta2, Ta4 y Ta5.

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.

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 desarrollo online)

Re3) Herramientas de desarrollo (Java, interfaz gráfica en Java, prueba de programas, diseño con UML)

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) Explicar el concepto, implementar y usar de forma adecuada los tipos abstractos de datos en el desarrollo de programas. Se consigue con las actividades formativas AF1 y AF3.

RA2) Describir los elementos básicos de la programación orientada a objetos y aplicarlos de forma adecuada en el desarrollo de programas. Se adquiere con las actividades formativas AF1, AF2 y AF3.

RA3) Explicar los conceptos de contenedor e iterador y usarlos de forma adecuada en el desarrollo de programas. Se alcanza con las actividades formativas AF1, AF2 y AF3

RA4) Emplear técnicas de prueba para detectar errores en programas de complejidad media. Se logra con las actividades formativas AF1 y AF2

RA5) Explicar los fundamentos de la programación guiada por eventos y usarla en el desarrollo de programas simples. Se adquiere con las actividades formativas AF1 y AF2

Todos estos resultados de aprendizaje pueden 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 presencial 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.

Atención presencial a grupos de trabajo

La atención presencial 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 ULPGC. 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

D/Dña. José Daniel González Domínguez (COORDINADOR)
Departamento: 260 - INFORMÁTICA Y SISTEMAS
Ámbito: 570 - Lenguajes Y Sistemas Informáticos
Área: 570 - Lenguajes Y Sistemas Informáticos
Despacho: INFORMÁTICA Y SISTEMAS
Teléfono: 928458751 **Correo Electrónico:** josedaniel.gonzalez@ulpgc.es

Dr./Dra. José Rafael Pérez Aguiar
Departamento: 260 - INFORMÁTICA Y SISTEMAS
Ámbito: 570 - Lenguajes Y Sistemas Informáticos
Área: 570 - Lenguajes Y Sistemas Informáticos
Despacho: INFORMÁTICA Y SISTEMAS
Teléfono: 928458731 **Correo Electrónico:** jose.perez@ulpgc.es

Bibliografía

[1 Básico] Piensa en Java /

Bruce Eckel.
Prentice Hall,, Madrid [etc.] : (2007) - (4ª ed.)
9788489660342

[2 Básico] UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado /

Larman Craig.
Prentice Hall,, Madrid : (2003) - (2ª ed.)
8420534382

[3 Básico] The Java tutorials [

Oracle and /or its affiliates.
Oracle,, [s.l.] : (1995)

[4 Básico] Java tools for Extreme Programming: mastering open source tools including Ant, JUnit, and Cactus /

Richard Hightower, Nicholas Lesiecki.
John Wiley & Sons., New York : (2002)
0-471-20708-X

[5 Recomendado] Programador Java certificado: curso práctico /

Antonio J. Martín
Sierra.
Ra-Ma., Paracuellos de Jarama, Madrid : (2010) - (3ª ed.)
978-84-7897-972-1

[6 Recomendado] Test-driven development: a practical guide /

Dave Astels.
Prentice Hall PTR., Upper Saddle River, N.J : (2003)
978-0-13-101649-1

[7 Recomendado] Ingeniería del software: un enfoque práctico /

Roger S. Pressman.
McGraw Hill., México [etc.] : (2010) - (7ª ed.)
9786071503145