23-24

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
SEGUNDO CURSO

# GUÍA DE ESTUDIO PÚBLICA

# LENGUAJES DE PROGRAMACIÓN Y PROCESADORES

CÓDIGO 71022028

UNED

23-24

**LENGUAJES DE PROGRAMACIÓN Y PROCESADORES**

**CÓDIGO 71022028**

# ÍNDICE

PRESENTACIÓN Y CONTEXTUALIZACIÓN
REQUISITOS Y/O RECOMENDACIONES PARA CURSAR LA ASIGNATURA
EQUIPO DOCENTE
HORARIO DE ATENCIÓN AL ESTUDIANTE
TUTORIZACIÓN EN CENTROS ASOCIADOS
COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE
RESULTADOS DE APRENDIZAJE
CONTENIDOS
METODOLOGÍA
SISTEMA DE EVALUACIÓN
BIBLIOGRAFÍA BÁSICA
BIBLIOGRAFÍA COMPLEMENTARIA
RECURSOS DE APOYO Y WEBGRAFÍA
TUTORIZACIÓN EN CENTROS ASOCIADOS
IGUALDAD DE GÉNERO

| | |
|---|---|
| Nombre de la asignatura | LENGUAJES DE PROGRAMACIÓN Y PROCESADORES |
| Código | 71022028 |
| Curso académico | 2023/2024 |
| Departamento | LENGUAJES Y SISTEMAS INFORMÁTICOS |
| Título en que se imparte | GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN |
| CURSO - PERIODO | GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN<br>- SEGUNDO CURSO<br>- SEMESTRE 2 |
| CURSO - PERIODO | ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE GESTIÓN EN UNED<br>- OPTATIVAS CURSO<br>- SEMESTRE 2 |
| CURSO - PERIODO | ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE GESTIÓN<br>- OPTATIVAS CURSO<br>- SEMESTRE 2 |
| Tipo | OBLIGATORIAS |
| Nº ETCS | 6 |
| Horas | 150.0 |
| Idiomas en que se imparte | CASTELLANO |

## PRESENTACIÓN Y CONTEXTUALIZACIÓN

Esta asignatura pertenece al grado de Ingeniería en Tecnología de la Información y se imparte en el segundo semestre del segundo curso. Tiene carácter obligatorio y consta de 6 créditos ECTS.

La asignatura introduce el estudio de la especificación formal de los lenguajes de programación y presenta los mecanismos de los lenguajes de programación, básicamente: sus sistemas de tipos, el control de secuencia y la abstracción. También se introduce el concepto de paradigma de lenguajes de programación, se identifican los paradigmas existentes, enmarcando los principales lenguajes de programación en paradigmas y destacando las características que los hacen adecuados para un determinado dominio o tipo de problemas. También introduce al alumno en el estudio de los procesadores de lenguajes, en particular en la estructura de un compilador. La asignatura se completa con un estudio de los lenguajes de marcado, en particular XML, y las características fundamentales de los lenguajes de script.

Esta asignatura junto con la de Autómatas, Gramáticas y Lenguajes conforman la materia de Lenguajes de Programación. Dicha asignatura proporciona los fundamentos para poder abordar en ésta la especificación de los lenguajes de programación, sus mecanismos y el estudio de los procesadores de lenguajes.

## REQUISITOS Y/O RECOMENDACIONES PARA CURSAR LA ASIGNATURA

El estudiante debería cursar con anterioridad las siguientes asignaturas:
- Autómatas, Gramáticas y Lenguajes, ya que proporciona los fundamentos sobre los que asentar el estudio de los lenguajes de programación y los procesadores.
- Fundamentos de Programación, ya que proporciona los conocimientos básicos de programación imperativa.
- Programación Orientada a Objetos, ya que proporciona conocimientos de orientación a objetos y el lenguaje de programación Java.
- Estrategias de Programación y Estructuras de Datos, ya que proporciona los conocimientos de las estructuras de datos básicas, análisis de algoritmos y recursividad.
- Programación y estructuras de datos avanzadas, ya que proporciona los conocimientos necesarios sobre los principales esquemas algorítmicos y su uso, así como el de las estructuras de datos avanzadas.

## EQUIPO DOCENTE

| | |
|---|---|
| Nombre y Apellidos | RAQUEL MARTINEZ UNANUE (Coordinador de asignatura) |
| Correo Electrónico | raquel@lsi.uned.es |
| Teléfono | 91398-8725 |
| Facultad | ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA |
| Departamento | LENGUAJES Y SISTEMAS INFORMÁTICOS |

| | |
|---|---|
| Nombre y Apellidos | VICTOR DIEGO FRESNO FERNANDEZ |
| Correo Electrónico | vfresno@lsi.uned.es |
| Teléfono | 91398-8217 |
| Facultad | ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA |
| Departamento | LENGUAJES Y SISTEMAS INFORMÁTICOS |

## HORARIO DE ATENCIÓN AL ESTUDIANTE

El alumno dispone de varios mecanismos para facilitar su aprendizaje:

* Tutores en los centros asociados. Serán los encargados de desarrollar las sesiones presenciales de seguimiento y control de las prácticas. Para realizarlas, el alumno debe ponerse en contacto con su tutor al comienzo del semestre, para conocer cuanto antes los horarios de las sesiones previstas.

* Tutorías presenciales o virtuales en el centro asociado correspondiente.

* Página web de la asignatura, en la que se ofrecerá una visión general de la asignatura e información actualizada sobre los contenidos y recursos. Puede encontrarse en www.lsi.uned.es.

* Entorno virtual de la asignatura (accesible desde el portal de la UNED, www.uned.es). Este entorno será el mecanismo básico de comunicación del equipo docente con estudiantes

y tutores. Dispone además de foros donde los alumnos podrán plantear sus dudas para que sean respondidas por los tutores o por el propio equipo docente.

* Tutorías con el equipo docente. El equipo docente atenderá dudas de carácter conceptual vía correo electrónico, teléfono o presencialmente. El Equipo Docente no resolverá problemas producidos a la hora de instalar, compilar o depurar los programas que vayan generando para el estudio de la asignatura o la práctica. Las dudas a este respecto deben plantearse al tutor asignado por el centro asociado.

Horario de tutorías:

Raquel Martínez Unanue: Martes de 09:30 a 13:30, raquel@lsi.uned.es, 913988725.

Víctor Fresno Fernández: Martes y Miércoles de 11:30 a 13:30, vfresno@lsi.uned.es, 913988217.

Dirección Postal:

Dpto. Lenguajes y Sistemas Informáticos
ETSI Informática
C/ Juan del Rosal nº 16 28040 - Madrid

## TUTORIZACIÓN EN CENTROS ASOCIADOS

En el enlace que aparece a continuación se muestran los centros asociados y extensiones en las que se imparten tutorías de la asignatura. Estas pueden ser:

•**Tutorías de centro o presenciales**: se puede asistir físicamente en un aula o despacho del centro asociado.

•**Tutorías campus/intercampus**: se puede acceder vía internet.

Consultar horarios de tutorización de la asignatura 71022028

## COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

Competencias generales:

CG.2 - Competencias cognitivas superiores: selección y manejo adecuado de conocimientos, recursos y estrategias cognitivas de nivel superior apropiados para el afrontamiento y resolución de diversos tipos de tareas/problemas con distinto nivel de complejidad y novedad: Análisis y Síntesis. Aplicación de los conocimientos a la práctica Resolución de problemas en entornos nuevos o poco conocidos. Pensamiento creativo. Razonamiento crítico. Toma de decisiones.

CG.4 - Competencias de expresión y comunicación (a través de distintos medios y con distinto tipo de interlocutores): Comunicación y expresión escrita. Comunicación y expresión oral. Comunicación y expresión en otras lenguas (con especial énfasis en el inglés). Comunicación y expresión matemática, científica y tecnológica (cuando sea requerido y estableciendo los niveles oportunos).

G.5 - Competencias en el uso de las herramientas y recursos de la Sociedad del Conocimiento: Manejo de las TIC. Competencia en la búsqueda de información relevante. Competencia en la gestión y organización de la información. Competencia en la recolección

de datos, el manejo de bases de datos y su presentación.

Además, durante el estudio de esta asignatura se pretende alcanzar las siguientes competencias específicas de la materia:
FB.4 - Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
FB.5 - Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, así como de los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.
BC.1 - Capacidad para diseñar, desarrollar, seleccionar y evaluar, aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a los principios éticos y a la legislación y normativa vigente.
BC.6 - Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
BC.7 - Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.
BC.8 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
BC.9 - Capacidad para conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.
BTEti.2 - Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.

## RESULTADOS DE APRENDIZAJE

- Conocer los lenguajes de programación en diferentes paradigmas, así como las técnicas para su implementación.
- Saber elegir el lenguaje más apropiado para el desarrollo de una aplicación. Conocer las alternativas y sus características.
- Conocer y utilizar las máquinas virtuales.
- Conectar los lenguajes de programación con gestores de bases de datos, aplicaciones, herramientas y servicios.
- Conocer y comprender las distintas etapas en la traducción de lenguajes de programación.
- Procesar lenguajes de marcado.

# CONTENIDOS

## Tema 1. Lenguajes de programación

Este capítulo expone las nociones básicas necesarias para entender los lenguajes de programación. Se presentan los conceptos de sintaxis y semántica de un lenguaje de programación, así como la notación BNF para la descripción de la sintaxis de estos lenguajes. Posteriormente se profundiza en los diferentes elementos que los componen: expresiones y enunciados, tipos de datos, ámbito de variables, etc. Cada uno de estos elementos es presentado enumerando sus principales características e indicando con ejemplos la forma en que se han incluido en diferentes lenguajes de programación. Los ejercicios de este capítulo hacen hincapié en la sintaxis de los lenguajes de programación y en el aprendizaje de los diferentes conceptos presentados a lo largo del tema.

•Introducción.
•Sintaxis.
•Semántica básica.
•Tipos de datos.
•Expresiones y enunciados.
•Procedimientos y ambientes.
•Tipos abstractos de datos y módulos.

## Tema 2. Procesadores de lenguajes.

Este capítulo ofrece una visión general de las diferentes etapas de transformación de un programa, desde un código fuente escrito por un programador, hasta un fichero ejecutable. De entre los diferentes tipos de procesadores de lenguajes se destacan los compiladores, de los que se ofrece una explicación más en detalle como ejemplo paradigmático de un procesador de lenguajes. La descripción de las fases de un compilador se centra fundamentalmente en las etapas de análisis más que en las de síntesis. Los ejercicios de este capítulo se centran en aspectos relacionados con la fase de análisis de un compilador, así como en la posterior etapa de traducción dirigida por sintaxis.

•Introducción
•Tipos de procesadores
•Estructura de un compilador
•Traducción dirigida por sintaxis.

## Tema 3. Paradigmas y modelos de programación.

Este capítulo define el concepto de paradigma de programación y describe brevemente algunos de los paradigmas existentes: imperativo, orientado a objetos, concurrente, funcional y lógico. Para introducir cada paradigma se utiliza un lenguaje representativo del mismo del cual se explican los conceptos sintácticos fundamentales para comprender el paradigma y poder realizar ejercicios sencillos. El capítulo también presenta otros lenguajes de computadora que a pesar de que no representan un paradigma concreto, son de un uso tan generalizado que merecen al menos una breve introducción: los lenguajes dinámicos y los lenguajes de marcado. Los ejercicios de este capítulo buscan reforzar el conocimiento del lector sobre cada uno de los paradigmas y lenguajes presentados.

•Programación funcional.
•Programación lógica.
•Programación Orientada a Objetos.
•Programación concurrente.
•Programación con lenguajes dinámicos.

## Tema 4. Lenguajes de marcado. XML.

Este capítulo introduce al lector en los fundamentos del lenguaje de marcado XML. Se presentan las principales tecnologías asociadas para que el lector conozca sus posibilidades y principales características. Además se presentan los procesadores de documentos XML en términos de los tipos de análisis que realizan. Los ejemplos, ejercicios resueltos, junto con el uso de las herramientas recomendadas en el propio capítulo permitirán al lector familiarizarse con este lenguaje de marcado.

•Introducción.
•Componentes de un documento XML.
•Modelado de datos en XML.
•Fundamentos de la DTD.
•Espacios de nombres.
•Fundamentos del XML-Schema.
•Procesadores de documentos XML.
•Vinculación entre documentos

## Tema 5. Lenguajes de script.

Este capítulo introduce los lenguajes de *script* comenzando con una breve descripción de sus orígenes. La presentación de los dominios en los que normalmente se utilizan sirve de marco para presentar y comentar algunas de las herramientas y lenguajes de *script* más

populares. En este capítulo tienen especial relevancia los ejercicios resueltos ya que son los que van a mostrar la potencia y diferencias de este tipo de lenguajes con respecto a otros lenguajes de programación. Se tratan dos lenguajes de *script* con un poco más de detalle, Perl y PHP, y en particular este último se relaciona con XML y algunos procesadores de documentos.

•Introducción.

•Dominios de aplicación.

•Algunos lenguajes de script destacados.

Tema 6. Aspectos pragmáticos de los lenguajes de programación.

Este capítulo ofrece al lector algunos aspectos que pueden resultar clave en la elección de un determinado lenguaje frente a otro, a la hora de crear un programa informático, o de estudiar la interoperabilidad entre aplicaciones escritas en diferentes lenguajes de programación.

•Principios de diseño de los lenguajes.

•Interacción e interoperatividad con aplicaciones, herramientas y servicios.

•Lenguajes embebidos.

•Criterios de selección de lenguajes.

## METODOLOGÍA

El estudiante trabajará con contenidos teórico-prácticos utilizando el texto base, la guía de estudio y el material complementario. Además se realizarán dos prácticas obligatorias bajo la supervisión del tutor, con las herramientas y directrices preparadas por el equipo docente. Se requerirá la asistencia, al menos, a dos de las sesiones presenciales de prácticas organizadas por el tutor en el centro asociado.

La metodología de estudio que se contempla es la siguiente:

•Trabajo autónomo de estudio de cada tema, utilizando la bibliografía básica y complementaria que contempla las siguientes actividades:

•Estudio **de los contenidos teóricos presentados y desarrollados en el capítulo.**

•Resolución de los **ejemplos y ejercicios resueltos** en el capítulo. Estos ejemplos y ejercicios se desarrollan completamente en el texto base, por lo que el estudiante podrá comprobar la corrección de su resolución.

•Resolución de los **ejercicios propuestos** al final del capítulo. El estudiante debe resolver los ejercicios y comentar sus dudas con el Tutor, en el centro asociado o en el Curso Virtual.

•Prácticas de laboratorio:

•Se facilita al estudiante el enunciado de dos prácticas obligatorias al comienzo del semestre.

•Se explica en el curso virtual cuál es el procedimiento para su realización.

•Para solicitar plaza/turno de prácticas de laboratorio/experimentales, el estudiante tendrá que acceder a la aplicación de prácticas desde su escritorio. En estas imágenes puede ver desde dónde se puede realizar el acceso a dicha aplicación: https://descargas.uned.es/publico/pdf/guias/ACCESO_PRACTICAS_GRADOS_2017.pdf Si al acceder a ella no encuentra ninguna oferta, deberá ponerse en contacto con el centro asociado donde está matriculado. Las dudas sobre la realización de las prácticas deben consultarse con el Tutor.

## SISTEMA DE EVALUACIÓN

**TIPO DE PRUEBA PRESENCIAL**

| | |
|---|---|
| Tipo de examen | Examen de desarrollo |
| Preguntas desarrollo | 5 |
| Duración del examen | 120 (minutos) |
| Material permitido en el examen | |

Ninguno

Criterios de evaluación

La prueba presencial, que será de carácter teórico-práctico, será evaluada por el equipo docente de la asignatura, y supondrá un 80% de la nota final. El examen, que podrá incluir preguntas sobre la práctica o prácticas realizadas, debe aprobarse con una nota igual o superior a 5 para promediar con la nota de prácticas.

| | |
|---|---|
| % del examen sobre la nota final | 80 |
| Nota del examen para aprobar sin PEC | |
| Nota máxima que aporta el examen a la calificación final sin PEC | |
| Nota mínima en el examen para sumar la PEC | |
| Comentarios y observaciones | |

**PRUEBAS DE EVALUACIÓN CONTINUA (PEC)**

| | |
|---|---|
| ¿Hay PEC? | No |
| Descripción | |

Esta asignatura tiene prácticas obligatorias que se describen en otras actividades evaluables.

Criterios de evaluación

| | |
|---|---|
| Ponderación de la PEC en la nota final | 0 |
| Fecha aproximada de entrega | |
| Comentarios y observaciones | |

**OTRAS ACTIVIDADES EVALUABLES**

¿Hay otra/s actividad/es evaluable/s? Si

Descripción

Prácticas obligatorias

Criterios de evaluación

Los alumnos deben realizar dos prácticas obligatorias. Cada práctica se puntuará sobre 10, siendo el aprobado el 5. Ambas deberán aprobarse.

**Para que se evalúe la práctica es imprescindible que, si es un programa, compile y funcione adecuadamente y, en otro caso, que responda a las cuestiones planteadas en la práctica de forma correcta. En la evaluación que realizará el tutor se tendrán en cuenta los siguientes aspectos:**

Corrección en las respuestas a las cuestiones.

Correcta utilización de los elementos objetos de estudio en la práctica.

Calidad del código y estilo de programación.

Posibles mejoras introducidas por el alumno a los requisitos básicos de la práctica.

Ponderación en la nota final 20%

Fecha aproximada de entrega

Comentarios y observaciones

Las prácticas tienen sesiones de asistencia obligatoria y se organizan desde los centros asociados bajo la responsabilidad de cada tutor, por lo que los alumnos deben ponerse en contacto con ellos lo antes posible al comienzo del curso para conocer:

El calendario de entrega de prácticas, tanto para la convocatoria de junio como para la de septiembre, y de las sesiones presenciales obligatorias de las prácticas.

La forma de entrega.

**¿CÓMO SE OBTIENE LA NOTA FINAL?**

Las prácticas y el examen deben aprobarse por separado. El examen solo se evalúa si las prácticas están aprobadas.

**La nota del examen representa el 80% de la valoración final de la asignatura y las prácticas el 20% restante.**

**La nota de las prácticas podrá incrementar hasta un máximo de 2 puntos la nota final de la asignatura. El aporte de las calificaciones de cada práctica a la nota final es el siguiente:**

**Sobresaliente -> 1 punto**

**Notable -> 0.75 puntos**

**Aprobado -> 0.5 puntos.**

**Las notas de las prácticas aprobadas son válidas para las convocatorias de junio y de septiembre del curso en el que se realizan y para el curso siguiente.**

## BIBLIOGRAFÍA BÁSICA

LENGUAJES DE PROGRAMACIÓN Y PROCESADORES,
Francisco Gortázar Bellas, Raquel Martínez Unanue, Víctor Fresno Fernández.
Ed. Ramón Areces, segunda edición 2016.
ISBN: 978-84-9961-249-2

## BIBLIOGRAFÍA COMPLEMENTARIA

ISBN(13):9789706862846
Título:LENGUAJES DE PROGRAMACIÓN2ª
Autor/es:Louden, Kenneth C. ;
Editorial:THOMSON PARANINFO,S.A.

ISBN(13):9789706862990
Título:CONSTRUCCIÓN DE COMPILADORES: PRINCIPIOS Y PRÁCTICAnull
Autor/es:Louden, Kenneth C. ;
Editorial:THOMSON PARANINFO,S.A.

Además de los libros anteriores, también recomendamos el siguiente sobre XML.

- ISBN: 9782746049581
Título: XML Practico. Bases esenciales, conceptos y casos prácticos.
Autor/es: BOULANGER, THIERRY
Editorial: ENI

En caso de no encontrar este libro, podría ser útil cualquiera sobre XML que incluya también las tecnologías XML.

## RECURSOS DE APOYO Y WEBGRAFÍA

Además de esta guía del curso y la guía de estudio que detalla el plan de trabajo propuesto para el alumno, el equipo docente proporcionará material adicional a través del entorno de enseñanza virtual de la asignatura.

En los centros asociados los alumnos dispondrán de ordenadores en los que se habrá instalado un servidor Apache con módulo PHP. Además, los alumnos que dispongan de

ordenador personal podrán instalarse este software.

## TUTORIZACIÓN EN CENTROS ASOCIADOS

En el enlace que aparece a continuación se muestran los centros asociados y extensiones en las que se imparten tutorías de la asignatura. Estas pueden ser:

•**Tutorías de centro o presenciales**: se puede asistir físicamente en un aula o despacho del centro asociado.

•**Tutorías campus/intercampus**: se puede acceder vía internet.

Consultar horarios de tutorización de la asignatura 71022028

---

## IGUALDAD DE GÉNERO

En coherencia con el valor asumido de la igualdad de género, todas las denominaciones que en esta Guía hacen referencia a órganos de gobierno unipersonales, de representación, o miembros de la comunidad universitaria y se efectúan en género masculino, cuando no se hayan sustituido por términos genéricos, se entenderán hechas indistintamente en género femenino o masculino, según el sexo del titular que los desempeñe.