

GUÍA DOCENTE 2024-2025

DATOS GENERALES DE LA ASIGNATURA

ASIGNATURA:	Lenguajes de programación		
PLAN DE ESTUDIOS:	Grado en Ingeniería Informática		
FACULTAD:	Escuela Politécnica Superior		
CARÁCTER DE LA ASIGNATURA:	Obligatoria		
ECTS:	6		
CURSO:	Segundo		
SEMESTRE:	Segundo		
IDIOMA EN QUE SE IMPARTE:	Castellano		
PROFESORADO:	Dr. Jose Breñosa		
DIRECCIÓN DE CORREO ELECTRÓNICO:	josemanuel.brenosa@uneatlantico.es		

DATOS ESPECÍFICOS DE LA ASIGNATURA

REQUISITOS PREVIOS:
No Aplica
CONTENIDOS:
1. Lenguajes, gramáticas y autómatas 1.1. Introducción 1.2. Definiciones previas 1.3. Definición formal de gramática libre de contexto. 1.4. Relaciones entre cadenas 1.5. Definición formal de lenguaje 2. Fundamentos del análisis léxico. 2.1. El Análisis Léxico 2.2. Funciones del analizador léxico y ventajas 2.3. Definiciones básicas 2.4. ¿Cómo funciona el analizador léxico? 3. Fundamentos del análisis sintáctico

- 3.1. Análisis sintáctico descendente
- 3.2. Análisis sintáctico ascendente
4. Análisis semántico.
 - 4.1. Especificación Semántica de Lenguajes de Programación
 - 4.2. Especificación Formal de Semántica
 - 4.3. Tareas y Objetivos del Análisis Semántico
5. Las tablas de símbolos.
 - 5.1. Necesidad de las Tablas de símbolos
 - 5.2. Objetivos de la Tabla de Símbolos (TS)
 - 5.3. Contenidos de la TS
 - 5.4. Operaciones con la TS
6. Diseño de lenguajes de programación
 - 6.1. Organización de la memoria en tiempo de ejecución.
 - 6.2. Generación de código. Validación, documentación y entorno de desarrollo.
 - 6.3. Intérpretes.

COMPETENCIAS

COMPETENCIAS GENERALES:

CG6 - Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.

CG8 - Capacidad de explicar y aplicar las materias básicas y tecnologías, que permitan el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

CG9 - Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

COMPETENCIAS ESPECÍFICAS:

CE14 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

RESULTADOS DE APRENDIZAJE:

- Conocer los fundamentos de un lenguaje de programación de alto nivel
- Conocer los fundamentos para la creación de un lenguaje de programación
- Saber manejar algunos tipos de datos, estructuras de datos según el lenguaje
- Comprender y saber utilizar eficientemente la orientación a objetos y sus conceptos derivados: herencia, polimorfismo, abstracción; que se da en algunos lenguajes de alto nivel.

METODOLOGÍAS DOCENTES Y ACTIVIDADES FORMATIVAS

METODOLOGÍAS DOCENTES:

En esta asignatura se ponen en práctica diferentes metodologías docentes con el objetivo de que los alumnos puedan obtener los resultados de aprendizaje definidos anteriormente:

- MD1 Método expositivo
- MD2 Estudio y análisis de casos
- MD3 Resolución de ejercicios
- MD4 Aprendizaje basado en problemas
- MD5 Aprendizaje orientado a proyectos
- MD6 Aprendizaje cooperativo / Trabajo en grupos
- MD7 Trabajo autónomo

ACTIVIDADES FORMATIVAS:

A partir de las metodologías docentes especificadas anteriormente, en esta asignatura, el alumno participará en las siguientes actividades formativas:

Actividades formativas		Horas
Actividades dirigidas	Clases expositivas	7,5
	Clases prácticas	22,5
	Seminarios y talleres	15
Actividades supervisadas	Supervisión de actividades	3,75
	Tutorías (individual / en grupo)	3,75
Actividades autónomas	Preparación de clases	15
	Estudio personal y lecturas	37,5
	Elaboración de trabajos	30
	Trabajo en campus virtual	7,5

El primer día de clase, el profesor/a proporcionará información más detallada al respecto.

SISTEMA DE EVALUACIÓN

CONVOCATORIA ORDINARIA:

En la convocatoria ordinaria de esta asignatura se aplican los siguientes instrumentos de evaluación:

Actividades de evaluación		Ponderación
Evaluación Continua	Entregas de Portfolio y Ejercicios	25 %
	Interés y participación del alumno en la asignatura	5 %
	Examen Teórico-Práctico parcial	20 %
Evaluación final	Examen Teórico-Práctico	50 %

La calificación del instrumento de la evaluación final (tanto de la convocatoria ordinaria de Junio como de la convocatoria extraordinaria de Julio/Septiembre, según corresponda) no podrá ser inferior, en ningún caso, a 4,0 puntos (escala 0 a 10) para aprobar la asignatura y consecuentemente poder realizar el cálculo de porcentajes en la calificación final (prueba final, prueba/s parcial/es, evaluación continua y evaluación del profesor).

En la circunstancia que en esa Prueba o Examen Final se obtenga una calificación de 3.9 o inferior, en el Acta de Evaluación correspondiente se registrará la nota o calificación obtenida en dicho Examen Final, sin hacer media aritmética con las notas obtenidas en la Evaluación continua.

El alumno que suspenda la asignatura en la convocatoria ordinaria podrá examinarse en la convocatoria extraordinaria, donde se tendrán en cuenta las calificaciones obtenidas en la Evaluación continua a lo largo del semestre.

CONVOCATORIA EXTRAORDINARIA:

La convocatoria extraordinaria tendrá lugar durante el mes de julio (consúltese el calendario académico fijado por la universidad). Esta consistirá en la realización de uno o dos Exámenes Teórico-Prácticos con un valor de hasta el 50% de la nota final de la asignatura. El resto de la nota se complementará con la calificación obtenida en la evaluación continua de la convocatoria ordinaria.

BIBLIOGRAFÍA Y RECURSOS DE REFERENCIA GENERALES

BIBLIOGRAFÍA BÁSICA:

Las siguientes referencias son de consulta obligatoria:

- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman TEORÍA DE AUTÓMATAS. LENGUAJES Y COMPUTACIÓN. Ed. Addison-Wesley. PEARSON
- Fernando Lopez Ostenero; Ana Maria Garcia Serrano. TEORIA DE LOS LENGUAJES DE PROGRAMACION. Editorial Universitaria Ramon Areces
- VV.AA. Lenguajes Formales Y Teorías De Autómatas. ADDISON WESLEY. MARCOMBO

BIBLIOGRAFÍA COMPLEMENTARIA:

Las siguientes referencias no se consideran de consulta obligatoria, pero su lectura es muy recomendable para aquellos estudiantes que quieran profundizar en los temas que se abordan en la asignatura.

- Alfred V. Aho, Effrey D. Ullman, Ravi Sethi. COMPILADORES: PRINCIPIOS, TÉCNICAS Y HERRAMIENTAS. Ed. Prentice Hall
- Alejandro Ramallo Martínez.,TEORÍA, DISEÑO E IMPLEMENTACIÓN DE COMPILADORES. Ra-Ma
- Jacinto Ruiz Catalán. Compiladores : teoría e implementación. Ed. RC Libros
- Des Watson. A Practical Approach to Compiler Construction. Ed. Springer
- Alejandro Ramallo Martinez. Teoría, Diseño E Implementación De Compiladores De Lenguajes. Ed. RA-MA

WEBS DE REFERENCIA:

<https://prezi.com/-y-iclK5lxsq/disenio-de-los-lenguajes-de-programacion/>

<https://es.scribd.com/document/63549009/criterios-para-diseno-de-lenguajes-de-programacion>



OTRAS FUENTES DE CONSULTA:

No Aplica