

Ficha Técnica

Titulación:	Grado en Ingeniería Informática		
Plan BOE:	BOE número 108 de 6 de mayo de 2015		
Asignatura:	Paradigmas de la Programación		
Módulo:	Programación y estructuras de datos		
Curso:	2º	Créditos ECTS:	6
Tipo de asignatura:	Obligatoria	Tipo de formación:	Teórica-Práctica

Presentación

Paradigmas de la Programación es una asignatura de carácter obligatorio y consta de 6 créditos. Esta asignatura aborda distintos modelos de programación o paradigmas para resolver problemas. Uno de los principales objetivos de esta asignatura es que el estudiante sea capaz de elegir el modelo adecuado para resolver cada problema. Los paradigmas de programación son pilares que definen el desarrollo informático de los programas y por extensión el desarrollo computacional. La aparición de cada uno de ellos ha marcado un hito en el modelo de desarrollo de un programa informático. En esta asignatura se estudiarán los paradigmas de programación más representativos de las diferentes formas de modelar un programa informático. Además, la asignatura “Paradigmas de la Programación” proporciona al estudiante una base en programación necesaria para afrontar con éxito asignaturas más avanzadas en los cursos de tercero y cuarto. Para el desarrollo de esta asignatura, se podrán usar los conocimientos previos de programación del estudiante, así como los conocimientos de Estructuras de Datos.

En esta asignatura se abordarán principalmente los siguientes temas:

- Breve descripción de los principales paradigmas de la programación.
- Paradigma procedimental: imperativo y orientado a objetos.
- Paradigma declarativo: lógico, funcional, dataflow y end-user development.
- Paradigma demostrativo.

Los requisitos previos obligatorios antes de abordar esta asignatura son:

- Es obligatorio que el estudiante haya superado la asignatura “Metodología de la Programación”.

Competencias y/o resultados del aprendizaje

- Capacidad para conocer e identificar la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.
- Capacidad para conocer y aplicar los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
- Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Contenidos Didácticos

- 1 Paradigmas de Programación. Concepto y Tipos de Paradigmas.
 - 1.1 Introducción y objetivos
 - 1.2 Concepto de Paradigma de Programación
 - 1.3 Tipos de Paradigmas de Programación
 - 1.3.1 Paradigmas Procedimentales
 - 1.3.2 Paradigmas Declarativos
 - 1.3.3 Paradigmas Demostrativos
 - 1.3.4 Secuencias de Control en Paradigmas
 - 1.4 Ejercicios de Auto comprobación
 - 1.5 Conclusiones
 - 1.6 Bibliografía
- 2 El Paradigma Funcional
 - 2.1 Introducción y objetivos
 - 2.2 El Paradigma Funcional. Concepto y Características
 - 2.2.1 La Función
 - 2.2.2 Especificación Procedimental de la Función
 - 2.2.3 Especificación de Declarativa de la Función
 - 2.2.4 Características del Paradigma Funcional
 - 2.3 Transferencia Referencial
 - 2.4 Composición Funcional
 - 2.5 Funciones de Orden Superior
 - 2.6 Concepto de Estado
 - 2.7 Funciones Polimórficas
 - 2.8 Evaluación Retardada o Diferida
 - 2.9 Mónadas
 - 2.10 Tipos de Datos en el Paradigma Funcional
 - 2.11 Programas Funcionales
 - 2.12 Ejercicios de Auto comprobación
 - 2.13 Conclusiones
 - 2.14 Bibliografía
- 3 El Paradigma Funcional. Metodología y Entorno.
 - 3.1 Introducción y objetivos
 - 3.2 Metodología FAD
 - 3.2.1 Lenguaje de Modelado FAD
 - 3.2.2 Fases de la Metodología FAD
 - 3.3 Lenguajes Funcionales
 - 3.3.1 Hope
 - 3.3.2 Haskell
 - 3.3.3 Otros lenguajes
 - 3.4 Entornos de Desarrollo
 - 3.4.1 Entornos para Hope
 - 3.4.2 Entornos para Haskell
 - 3.4.3 Entornos para otros Lenguajes
 - 3.5 Ejercicios de Auto comprobación

- 3.6 Conclusiones
- 3.7 Bibliografía
- 4 El Paradigma Lógico
 - 4.1 Introducción y objetivos
 - 4.2 Funciones versus Relaciones
 - 4.3 La Forma Clausal
 - 4.4 La Cláusula de Horn
 - 4.5 Procedimientos de Prueba. Reglas de Inferencia y Estrategias de Búsqueda
 - 4.6 Indeterminismo del Paradigma Lógico
 - 4.7 Programa Lógico
 - 4.8 Determinismo de los Lenguajes Lógicos
 - 4.9 Prolog
 - 4.10 El Operador Cut (!)
 - 4.11 Ejercicios de Auto comprobación
 - 4.12 Conclusiones
 - 4.13 Bibliografía
- 5 El Paradigma Lógico. Metodología y Entorno
 - 5.1 Introducción y objetivos
 - 5.2 Metodologías
 - 5.2.1 Lógica Tradicional
 - 5.2.2 GUPU
 - 5.2.3 Metodología basada en la Ingeniería del Conocimiento
 - 5.3 El Lenguaje de Prolog
 - 5.4 Otros Lenguajes
 - 5.5 Entornos de Desarrollo Prolog
 - 5.6 Ejercicios de Auto comprobación
 - 5.7 Conclusiones
 - 5.8 Bibliografía
- 6 El Paradigma Demostrativo. Programación Genética
 - 6.1 Introducción y objetivos
 - 6.2 Paradigma Demostrativo
 - 6.3 Computación Evolutiva
 - 6.4 Algoritmos Genéticos
 - 6.5 Fundamentos de la Programación Genética
 - 6.6 Variantes de la Programación Genética
 - 6.6.1 Programación Genética de Koza
 - 6.6.2 Programación Genética Guiada por Gramáticas
 - 6.7 Ejercicios de Auto comprobación
 - 6.8 Conclusiones
 - 6.9 Bibliografía
- 7 El Paradigma Dirigido por Eventos de Interfaz de Usuario
 - 7.1 Introducción y objetivos
 - 7.2 La Programación Dirigida por el Control
 - 7.3 La Programación Dirigida por los Datos
 - 7.4 Concepto de Evento. Eventos Secuenciales, Simultáneos y Concurrentes

- 7.5 La Programación Dirigida por los Eventos
- 7.6 Características de los Programas Dirigidos por Eventos
- 7.7 Tipos de Eventos de Interfaz Gráfica
 - 7.7.1 Eventos de Bajo Nivel
 - 7.7.2 Eventos Semánticos
- 7.8 Ejercicios de Auto comprobación
- 7.9 Conclusiones
- 7.10 Bibliografía
- 8 El Paradigma Orientado a Objetos y Eventos. Metodología y Entorno
 - 8.1 Introducción y objetivos
 - 8.2 Metodología Básica para el Paradigma Orientado a Objetos y Eventos
 - 8.3 Disciplinas del RUP
 - 8.4 Fases del RUP
 - 8.5 Librerías y Entornos de Desarrollo Orientado a Objetos y Dirigido por Eventos
 - 8.5.1 Lenguaje Java. Librería Swing.
 - 8.5.2 Adaptaciones a la Metodología al Emplear un Entorno de Desarrollo visual
 - 8.6 Ejercicios de Auto comprobación
 - 8.7 Conclusiones
 - 8.8 Bibliografía
- 9 El Paradigma Dataflow
 - 9.1 Introducción y objetivos
 - 9.2 Modelo Computacional Dirigido por el Flujo de Datos
 - 9.2.1 El Grafo de Flujo de Datos
 - 9.2.2 Dependencias Circulares
 - 9.2.3 Paradigma Dataflow versus Paradigma Funcional
 - 9.3 Tipos de Nodos de un GFD
 - 9.4 La Hoja de Cálculo
 - 9.5 Implementación del Paradigma Dataflow
 - 9.5.1 Implementación Data-Driven
 - 9.5.2 Implementación Demand-Driven
 - 9.5.3 Características de las Computaciones Data-Driven y Demand-Driven
 - 9.6 Lenguajes de Programación Visual
 - 9.7 Lenguajes Dataflow
 - 9.8 Ejercicios de Auto comprobación
 - 9.9 Conclusiones
 - 9.10 Bibliografía
- 10 El Paradigma End-User Development
 - 10.1 Introducción y objetivos
 - 10.2 Origen y Fines del Paradigma
 - 10.3 Clasificación del Paradigma EUD
 - 10.4 Metáforas del Paradigma
 - 10.5 Implementación del Paradigma en la Web
 - 10.6 Modelo de Desarrollo
 - 10.7 Lenguajes y Herramientas de Desarrollo
 - 10.8 Líneas Futuras del Paradigma EUD

10.9 Ejercicios de Auto comprobación

10.10 Conclusiones

10.11 Bibliografía

Contenidos Prácticos

Durante el desarrollo de la asignatura se realizarán las siguientes actividades prácticas:

- Realización de prácticas en lenguajes del Paradigma funcional.
- Realización de prácticas en lenguajes del Paradigma lógico.
- Resolución de casos prácticos del Paradigma demostrativo y Paradigma Orientado a Objetos.

Evaluación

El sistema de evaluación del aprendizaje de la UDIMA contempla la realización de diferentes tipos de actividades de evaluación y aprendizaje. El criterio de valoración establecido se detalla a continuación:

Actividades de aprendizaje	10%
Controles	10%
Actividades de Evaluación Continua (AEC)	20%
Examen final presencial	60%
TOTAL	100%

Bibliografía

- Alonso, F., Lara, J.L., Lizcano, D. y Martínez, L. Paradigmas de Programación, Ed. Administración Digital S.L.
- Ambler, A.L., Burnett, M. y Zimmerman, B. (1992) Operational versus Definitional: A Perspective on Programming Paradigms, IEEE Computer.