

Nombre de la asignatura: Fundamentos de Programación

Créditos: 3-2-5

Aportación al perfil

- Aplicar conocimientos científicos y tecnológicos en la solución de problemas en el área informática con un enfoque interdisciplinario.
- Analizar, modelar, desarrollar, implementar y administrar sistemas de información para aumentar la productividad y competitividad de las organizaciones.
- Seleccionar y utilizar de manera óptima técnicas y herramientas computacionales actuales y emergentes.

Objetivo de aprendizaje

- Diseñar metodológicamente la solución de problemas, utilizando técnicas de análisis, diseño y formulación de soluciones, con un enfoque de programación orientada a objetos.

Competencias previas

- Conocimientos básicos de computación.
- Razonamiento lógico y matemático.
- Conocimientos de software de aplicación.

Temario

- Evolución de los paradigmas de programación
 - Programación lineal.
 - Programación estructurada.
 - Programación Orientada a Objetos.
 - Relación entre la programación orientada a objetos y la programación estructurada.
 - Metodología para la solución de problemas basados en objetos.
- Fundamentos del paradigma orientado a objetos
 - Estructura de un programa orientado a objetos.
 - Clases.
 - Objetos.
 - Métodos.
 - Tipos de datos.
 - Variables.
 - Constantes.
 - Sentencias condicionales.
 - Sentencias repetitivas.
 - Herencia y Polimorfismo.
- Representación de algoritmos
 - Algoritmos tradicionales.
 - Algoritmos orientados a objetos.
 - Otras tendencias.
- Modelado de aplicaciones utilizando la POO (UML)
 - Importancia del modelado.
 - Diagramas.
 - Modelado de requerimientos y casos de uso.
 - Modelado de acciones.
 - Máquinas de estado.
 - Modelado avanzado.

Definición de las competencias específicas

- Adquirir bases y fundamentos sobre metodologías de programación y las implementa en la solución de problemas informáticos.
- Aplicar conocimientos para instalar y configurar herramientas de programación orientado a objetos en diferentes plataformas.
- Aplicar técnicas actuales de programación orientada a objetos en las soluciones que propone.
- Aplicar técnicas orientadas a objetos para el desarrollo de algoritmos.
- Conocer los elementos para la construcción de modelos.
- Razonar modelos que reflejen los comportamientos de los sistemas.

Sugerencias didácticas transversales para el desarrollo de competencias profesionales

- Investigar antes de la clase y ubicar en el tiempo la evolución de los diferentes paradigmas de programación, llegar a conclusiones mediante una lluvia de ideas.
- Hacer comparaciones entre los diferentes paradigmas de programación.
- Promover durante el curso actividades de exposición que propicie el intercambio de ideas y la discusión dirigida.
- Propiciar la lectura y realización de informes sobre paradigmas de programación actuales emergentes.
- Promover la investigación de tópicos relacionados con la asignatura.
- Promover el trabajo en equipo para que el estudiante construya su aprendizaje.
- Organizar paneles con estudiantes de semestres avanzados para sensibilizar a los estudiantes de nuevo ingreso sobre la importancia de la materia en su estructura reticular y el perfil de su carrera.
- Elaborar en coordinación con los estudiantes un conjunto de problemas actuales asociados al entorno.
- Planear problemas para que resueltos en equipos durante el curso.

Prácticas

- Realizar un cuadro comparativo que describa las características de los paradigmas de programación.
- Realizar prácticas de ejercicios sobre identificación de objetos.
- Ejercicio 1: Representar los objetos físicos: silla, computadora, casa, escuela, empresa y persona a su concepción lógica.
- Ejercicio 2: Identificar objetos lógicos (no tangibles) tales como: reacciones del cuerpo (caliente, frío, enojo, alegría, impulsos).
- Realizar prácticas mediante codificación para la representación de clases atendiendo el paradigma orientado a objetos.
- Ejercicio 1. Describir las características y funciones de componentes de objetos utilizados cotidianamente tales como: video casetera, horno de microondas, teléfono, refrigerador, expendedor de refresco, etc.
- Ejercicio 2: Codificar las clases anteriores desarrollando sus métodos correspondientes.
- Ejercicio 3: Codificar la clase principal para instanciar objetos en sus múltiples modalidades (sin constructor, con constructor).
- Ejercicio 4: Utilizar la variable instanciada para llamar a los métodos sin parámetros y con parámetros de entradas y métodos que retornen parámetros.
- Ejercicio 5: Realizar codificaciones donde aplique las terminologías de herencia y polimorfismo.
- Realizar prácticas de ejercicios mediante codificación de algoritmos tradicionales.
- Realizar prácticas de ejercicios mediante codificación de algoritmos deterministas.
- Realizar prácticas de ejercicios mediante codificación de algoritmos estadísticos.
- Realizar prácticas de ejercicios mediante codificación de algoritmos de nuevas tendencias.
- Realizar un diagrama del modelado de un caso práctico mediante UML.

Criterios de evaluación

La evaluación de la asignatura se hará con base en siguiente desempeño:

- Trabajos de investigación.
- Tareas relacionadas con el modelado de la solución de problemas.
- Exámenes escritos.
- Ejercicios dentro del aula.
- Establecer de común acuerdo con los estudiantes, la ponderación de las diferentes actividades del curso.
- Desempeño académico del estudiante.