

Módulo: Algoritmos y Estructuras de Datos II

Carga Horaria: 64 horas reloj

Síntesis introductoria

Este módulo considera lo abordado en Algoritmos y Estructuras de Datos I. En tal sentido, se propone acercar a los alumnos a los distintos paradigmas que ofrece la programación, profundizando en aspectos de la programación orientada a objetos sus componentes y técnicas. Particularmente, el paradigma de programación orientado a objetos les brinda a los futuros profesionales, técnicos que combinan la abstracción, modularización, encapsulamiento, polimorfismo y herencia. Así, la combinación de estos aspectos promueve una forma de resolver problemas, no abstracta, sino más cercana a la realidad.

Capacidades profesionales

Se espera que al finalizar el cursado del módulo los estudiantes sean capaces de:

- Conocer los distintos paradigmas de la programación.
- Conocer y utilizar el paradigma de objetos, sus características, ventajas y aplicaciones dentro del desarrollo de sistemas.
- Adquirir técnicas de resolución de problemas reales.
- Diseñar aplicaciones con frameworks orientado a objetos.

Contenidos

Conceptos y paradigmas de lenguajes de programación. Comparación entre paradigmas. Paradigma funcional. Paradigma lógico. Paradigma orientado a objetos. Clases y objetos. Subclases. Atributos. Métodos. Recursividad. Modificadores de visibilidad. Encapsulación. Sobrecarga de métodos. Concepto de acoplamiento. Herencia. Sobreescritura. Clases abstractas y concretas. Cardinalidad. Atributos y comportamiento. Diseño UML. Diagrama de clases. Relaciones entre clases: herencia, asociación, composición y agregación. Diagrama de Secuencia. Diagramas de Interacción y casos de Uso. Patrones de Diseño: introducción, definición, descripción, catálogo, utilidad, selección y usos de un patrón. Patrones creacionales, estructurales y de comportamiento. Mover aspectos entre objetos. Simplificación de invocación de métodos. Manipulación de la generalización. Patrones creacionales (Abstract Factory, Singleton). Patrones estructurales (Composite, Decorador, Adapter, Proxy. Patrones de comportamiento: Observer, State, Strategy, Template Method, Command). Refactoring: Introducción, utilidad y técnicas de aplicación del Refactoring. Catálogo de refactoring. Refactoring hacia patrones. Unificación de interfaces con el patrón adaptador. Reemplazar lógica condicional con el patrón estrategia. Reemplazar estados condicionales con el patrón estado. Reemplazar notificaciones con el patrón observador. Mover código embebido al patrón decorador. Aplicaciones con frameworks orientado a objetos. Introducción a Frameworks. Reutilización de software vs. Reutilización de diseño. Clasificación de frameworks según su propósito Testing. Tipos de tests: de unidad, de integración, de aceptación. Metodología de desarrollo ágil TDD (Test Driven Development). Relación entre refactoring y testing.

Perfil docente

Graduado en el nivel Superior que posea formación específica en los contenidos enunciados en este espacio curricular, con formación pedagógica que califique su ingreso y promoción en la carrera docente.