



MINISTERIO DE EDUCACIÓN
DIRECCIÓN DE FORMACIÓN TÉCNICA SUPERIOR
Instituto de Formación Técnica Superior N° 29

Materia Desarrollo de Sistemas Orientado a Objetos	Año 2025 – 1° cuatrimestre
Régimen Cuatrimestral	

FUNDAMENTACIÓN

La asignatura **Desarrollo de Sistemas Orientado a Objetos (DSOO)** es la puerta de entrada a la programación orientada a objetos. Su cursada permite al futuro egresado de la Tecnicatura Superior en Desarrollo de Software, aplicar técnicas de programación orientada a objetos, desarrollando sistemas mediante el lenguaje adecuado a tal fin; modelizar y diseñar aplicaciones mediante el Lenguaje de Modelado Unificado (UML), utilizando las reglas de buenas prácticas de programación y las normas de calidad del software y verificar la funcionalidad de las clases diseñando sus respectivas clases de prueba, elaborando la documentación técnica correspondiente.

Para la organización de la enseñanza de esta unidad curricular se han organizado los contenidos en dos ejes: “Paradigmas de Programación Orientada a Objetos”, y “Desarrollo de Sistemas”. El eje de la Programación Orientada a Objetos se centra en la presentación de conceptos básicos que sustentan y dan sentido a este paradigma de programación: abstracción, encapsulamiento, modularización, jerarquía de clases, herencia, polimorfismo y relaciones entre clases.

El núcleo central del bloque Desarrollo de Sistemas es la elaboración y la construcción de sistemas computacionales implementando los conceptos del paradigma orientado a objetos y el modelado de situaciones problemáticas en un lenguaje adecuado al paradigma en un entorno de desarrollo corporativo o abierto.

PROPÓSITOS GENERALES

- Construir saberes, conocimientos y habilidades referidos al paradigma de programación orientada a objetos, en particular componentes y técnicas, construyendo un sistema en donde aplique el paradigma y las técnicas de programación orientada a objetos.
- Conceptualizar la utilización de objetos para la codificación de algoritmos, la reutilización del código, la programación ordenada y la documentación específica de la programación.

OBJETIVO DE LA MATERIA

El **Diseño de Sistemas Orientado a Objetos** es una materia que busca construir saberes, conocimientos y habilidades referidos al paradigma de programación orientada a objetos, en particular componentes y técnicas, construyendo un sistema en donde aplique este paradigma.

Se espera que al finalizar el cursado de la materia los estudiantes sean capaces de:

- Aplicar técnicas de programación orientada a objetos, desarrollando sistemas mediante el lenguaje adecuado a tal fin.
- Modelizar y diseñar aplicaciones mediante el Lenguaje de Modelado Unificado (UML), utilizando las reglas de buenas prácticas de programación y las normas de calidad del software.
- Verificar la funcionalidad de las clases diseñando sus respectivas clases de prueba, elaborando la documentación técnica correspondiente.

CONTENIDOS

Unidad 1: Introducción al paradigma orientado objetos

Introducción al paradigma de objetos. Comparación entre paradigmas. Clases y objetos. Subclases. Lenguaje #C. Entorno visual. Introducción a la codificación. Atributos. Modularidad. Conceptos sobre Sistemas.

Unidad 2: Pilares en Programación Orientada a Objetos

Sobrecarga de métodos. Concepto de acoplamiento. Pilares de la POO: Abstracción y Encapsulamiento. Sobreescritura. Controles básicos del entorno visual. Interface de usuario. Agregación/composición: por valor y por referencia. Arraylist. Asociación: multiplicidad - colecciones. Modificadores de visibilidad.

Unidad 3: Metodología de desarrollo

Pilares de la POO, continuación: Herencia y Polimorfismo. Generalización. Especialización. Modelo de tres capas. Jerarquía de clases. Sobreescritura de métodos. Entorno de datos: conexión y manipulación. Codificación de subsistemas. Administración de excepciones. El modelo computacional de objetos. Tiempo de vida de los objetos.

Unidad 4: Desarrollo e implementación

Clases y métodos abstractos. Diseño UML. Diagrama de clases. Relaciones entre clases: herencia, asociación, composición y agregación. Diagrama de Casos de Uso. Diagrama de Secuencia. Construcción de un diseño. Carga de grillas. Implementación de interfaces. Control del Código. Pruebas al Sistema. Atributos y métodos estáticos. Puesta a punto del Sistema.

A continuación se presenta una distribución de los contenidos por semana:

PLAN DE CONTENIDOS

La asignatura se organiza en momentos 4 etapas consecutivas, que son propias del desarrollo de un sistema de gestión: E1 Preliminar, E2 Análisis, E3 Programación, E4 Implementación. A continuación se presenta una distribución de los contenidos por semana y según las etapas:

Semana	Unidad/ Eje/ Recorrido	Contenidos
1 Nuestro lenguaje	Unidad 1 Conceptos básicos de la programación orientada a objetos	Conceptos sobre Sistemas Lenguaje C# <ul style="list-style-type: none"> • Instalación de Visual Studio. • Reconocimiento del entorno de desarrollo (Área de trabajo). • Conceptos básicos (Variables, Constantes, Operadores, estructuras de decisión y Ciclos).
2 Haciendo foco	Unidad 1 Conceptos básicos de la programación orientada a objetos	Profundización en el paradigma orientado a Objetos <ul style="list-style-type: none"> • Clases. • Instanciación. • Constructor. • Introducción a la codificación en entorno visual.
3 El entorno y sus métodos	Unidad 2 Pilares en Programación Orientada a Objetos	Sobrecarga de métodos Asociación simple. Controles básicos entorno visual: combobox, cajas de verificación, opciones.

4 Pilares de la POO (Parte 1)	Unidad 2 Pilares en Programación Orientada a Objetos	<p>Visibilidad. Abstracción, encapsulamiento. Tips para la creación del entorno visual.</p> <p>Presentación del Proyecto Integrador (PI) cuyo objetivo es analizar, desarrollar y poner en funcionamiento un sistema que satisfaga los requerimientos de un cliente.</p> <p>Fase 1 de PI: Análisis del funcionamiento del club deportivo.</p>
5 Colecciones	Unidad 2 Pilares en Programación Orientada a Objetos	<p>Concepto de colección ArrayList. Tips para la creación de clases Fase previa y funcional del sistema Bocetos de la interface</p> <p>PI.El club deportivo.</p> <p>Fase 2 de PI: Análisis y documentación de la fase previa y funcional del sistema "Club deportivo".</p>
6 Repaso		Revisión de contenidos
7 Pilares de la POO (Parte 2)	Unidad 3 Metodología de desarrollo	<ul style="list-style-type: none"> • Herencia. • Generalización. • Especialización. • Jerarquía de clases. • Tipos de herencia. • Métodos en clases derivadas. • Fase de análisis orgánico del Sistema de la cursada. <ul style="list-style-type: none"> ◦ Construir el caso de uso de los procesos. • Esquema del análisis orgánico del Sistema. <ul style="list-style-type: none"> ◦ Explicación del contenido PI.El club deportivo. <p>Fase 3 de PI: Análisis orgánico. Análisis y creación de las plantillas de caso de uso del sistema.</p>
	Entrega del 1er TP obligatorio	Trabajo práctico integrador.
8 Metodología de especialización	Unidad 3 Metodología de desarrollo	<ul style="list-style-type: none"> • Sobreescritura de métodos. • Constructores en herencia. • Entorno de datos y codificación del Sistema de la cursada. • Conexión con el entorno de datos. • Manipulación de datos desde el sistema. <p>Fase 4 de PI: Conexión de la base de datos al sistema.</p>
9 Último pilar de la POO	Unidad 3 Metodología de desarrollo	<ul style="list-style-type: none"> • Polimorfismo. • Ingreso al sistema y codificación de los Sub_Sistemas (Parte_I). <ul style="list-style-type: none"> *Ventana Principal del Sistema. *Llamada a la Ventana Principal. *Programación de los procesos. *Inscribir Postulante (Insert a la tabla). *Método para el Nuevo Postulante.

		Fase 5 de PI: Programación del menú principal y los nuevos socios al club deportivo. Primer entregable del PI.
10 Manipulan- do datos	Unidad 4 Desarrollo e implementación	<ul style="list-style-type: none"> • Clases abstractas. • Carga de grillas desde entorno de datos. • Diseño de documentos e impresión. Fase 6 de PI: Diseño de los documentos de salida.
11 Control del código	Unidad 4 Desarrollo e implementación	<ul style="list-style-type: none"> • Interfaces. • Metodología para control del código. • Procesos: Diálogos entre el actor y el sistema. <ul style="list-style-type: none"> ◦ Flujo Normal. ◦ Flujo Alternativo. ◦ Postcondiciones.
12 Optimizan- do el código	Unidad 4 Desarrollo e implementación	<ul style="list-style-type: none"> • Pruebas al Sistema. <ul style="list-style-type: none"> ◦ Casos de Prueba. ◦ Plantilla de Prueba. • Atributos estáticos. • Métodos estáticos.
	Entrega del Cuestionario Obligatorio	Cuestionario obligatorio con nota.
13 Puesta a punto del sistema y prepara- ción del coloquio grupal		Puesta a punto del sistema. Preparación del video grupal del coloquio obligatorio.
14 Entrega Final PI y video del coloquio grupal.		Entrega completa del sistema y del video del coloquio.
15 Conclusio nes.		Retroalimentación grupal. Autoevaluación.
16 Devolucio nes y		Devoluciones y cierre de la materia.

cierre de la materia.		
-----------------------	--	--

EVALUACIÓN:

La asignatura tendrá un enfoque formativo de evaluación donde se priorizará la producción grupal a partir de la propuesta didáctica presentada. Se valorará el trabajo en equipo a partir de la identificación, reconocimiento y desarrollo de aquellas capacidades y habilidades que el/la estudiante pone en juego así como aquellas que se desarrollen durante el cursado.

Modos de aprobación:

- Participación mínima quincenal en la oficina de trabajo, interactuando activamente dentro del equipo según las pautas establecidas.
- Elaboración y aprobación de los Trabajos Prácticos Obligatorios que conforman un 50% de la nota final y deben desarrollarse correctamente en un 60% como mínimo.
- Entrega y aprobación del Proyecto Integrador y Coloquio Asincrónico Grupal que conforma el 50% restante de la nota final y debe estar adecuadamente implementado en un 60% como mínimo.

BIBLIOGRAFÍA

- Organización de Estados Iberoamericanos. (2010). Metas Educativas 2021. La Educación que queremos para la generación de los bicentenarios. Madrid: OEI.
- Organización Internacional del Trabajo (2005). Recomendación sobre el desarrollo de Recursos
- Humanos Educación, Formación y Aprendizaje Permanente N° 195. Ginebra: OTT. Recuperado de www.cinteffor.org.uy
- Pressman, R. (2010). Ingeniería del software: un enfoque práctico. México: Mc Graw-Hill.
- Ramakrishnan, R. y Gehrke, J. (2007). Sistemas de gestión de bases de datos. España: Mc Graw-Hill.
- Design Patterns in Java - Steven John Metsker, William C.Wake. (2006)
- Diseño de Algoritmos, Nivio Siviani. (Ed. Tomson, 2007)
- Sommerville, I. (2011). Ingeniería de Software (Novena ed.). Mexico: Addison- Wesley

- Jeff, P. (2014). User Story Mapping. California: O'Reilly Media Inc. Desafíos de Programación, Nivio Siviani. (Ed. Thomson, 2014)
- Fundamentos de Programación y Bases de Datos, Ángel Arias. (Ed. CreateSpace, 2014).