

INSTITUTO DE DESARROLLO ECONÓMICO E INNOVACIÓN

Año: 2023



PROGRAMA DE LA ASIGNATURA:

Programación y Diseño Orientado a Objetos (IF030)

CÓDIGO: IF030

AÑO DE UBICACIÓN EN EL PLAN DE ESTUDIOS:

2 año

FECHA ULTIMA REVISIÓN DE LA ASIGNATURA:

2021-06-30

CARRERA/S: Licenciatura en Sistemas 049/2017, Analista Universitario de Sistemas 050/2017,

CARÁCTER: CUATRIMESTRAL (2do)

TIPO: OBLIGATORIA

NIVEL: GRADO

MODALIDAD DEL DICTADO: PRESENCIAL

MODALIDAD PROMOCION DIRECTA: NO

CARGA HORARIA SEMANAL: 8 HS

CARGA HORARIA TOTAL: 120 HS

EQUIPO DOCENTE

Nombre y Apellido	Cargo	e-mail
Martin Villarreal	Profesor Adjunto	mvillarreal@untdf.edu.ar
Ariel Parson	Profesor Adjunto	aparson@untdf.edu.ar

1. FUNDAMENTACION

Permite al alumno obtener nuevas técnicas y herramientas orientadas a objetos, para la resolución de problemas. La aplicación de estas técnicas, permite obtener los beneficios de la programación orientada a objetos tales como abstracción, reuso, extensibilidad y facilidad de mantenimiento entre otras.

La asignatura es correlativa a la asignatura Algorítmica de programación II, que permite al alumno resolver algoritmos dentro del paradigma imperativo. Esta Asignatura, por su parte, introduce al alumno en los conceptos fundamentales de un nuevo paradigma de programación, el paradigma orientado a objetos. Estos conceptos serán ampliados en la asignatura Paradigmas y Lenguajes de Programación desde un enfoque más Teórico.

La asignatura se encuentra fuertemente ligada a la asignatura Ingeniería de Software I, ya que esta materia complementa los contenidos de "Arquitectura y diseño" y "desarrollo de software". Estos conceptos podrían ser también utilizados en la asignatura Laboratorio de software si se optara por este paradigma de programación.

2. OBJETIVOS

a) OBJETIVOS GENERALES

- Que el alumno comprenda las características y alcances de los distintos paradigmas de programación.
- Que el alumno aprenda los fundamentos de la programación orientada a objetos, así como la introducción de estos conceptos en el desarrollo de software y los objetivos buscados por los métodos de diseño orientado a objetos.

b) OBJETIVOS ESPECIFICOS

- Utilizar el lenguaje Java como herramienta para la programación orientada a objetos.
- Conocer con detalle sus características relativas a la POO, y como herramienta para implementar interfaces gráficas de usuario.
- Presentar un lenguaje standard para representar los diferentes aspectos de un sistema: UML. Diagramas UML para modelar los aspectos estáticos de un sistema y diagramas para modelar los aspectos de comportamiento de un sistema.
- Desarrollar capacidades de programación e implementación de problemas, adecuadas a los requerimientos de las prácticas.

3. CONDICIONES DE REGULARIDAD Y APROBACION DE LA ASIGNATURA

La evaluación de la cursada se realiza mediante una prueba de carácter práctico para los temas incluídos hasta la unidad 3. El mismo se aprueba con nota mayor o igual a 6, y se da una instancia recuperatoria para el mismo. El segundo parcial abarca los temas siguientes hasta los incluídos en la unidad 6. El resto de los temas se evalúan con la aprobación de un trabajo de programación integrador en el lenguaje Java, el cual se presenta al terminar la unidad 5, y se entrega al final de la cursada. Se pautan requisitos mínimos para su aprobación y se brinda la posibilidad de rehacer los aspectos inconclusos o poco satisfactorios de la resolución entregada en primera instancia.

El 2° parcial y el trabajo integrador se aprueban con iguales condiciones que el primer parcial. Para la evaluación final de la materia se realiza mediante un trabajo de programación integrador del tema de programación orientada a objetos, el cual se presenta en fecha de finales realizando una defensa del mismo y un coloquio con el profesor que realiza las consultas que considera apropiadas.

4. CONTENIDOS DE LA ASIGNATURA

Contenidos mínimos:

- Objetos. Clases e instancias.
- Herencia. Jerarquías de clases.
- Encapsulamiento y abstracción en OO.
- Polimorfismo.
- Lenguajes OO.
- Interfaces gráficas. Eventos
- Metodologías de diseño orientado a objetos.
- Patrones.
- Diseño de aplicaciones.

1) Conceptos básicos del Paradigma Orientado a Objetos.

Problemas que resuelve este paradigma. Características del Modelo. Tipos abstractos de datos. Encapsulación. Ocultamiento de la información. Objetos. Cooperación entre objetos: envío de mensajes. Métodos. Clases e instancias. Jerarquías de clases. Subclases. Superclases. Clases Abstractas. Herencia. Polimorfismo. Introducción al Diseño Orientado a Objetos: Identificación de clases, responsabilidades y colaboraciones.

2) El paradigma orientado a objetos en JAVA.

Creación y manipulación de objetos. Tipos primitivos del lenguaje. Arreglos. Ambito. Creación de clases. Métodos, argumentos y valores de retorno. Estilo de código. Operadores. Expresiones de control de flujo. Bucles. Constructores. Sobrecarga de métodos. Inicialización. Inicialización de arreglos. Especificadores de acceso. Reuso. Composición. Herencia. Combinación de herencia y composición. Polimorfismo. Overriding. Diferencia con sobrecarga. Clases abstractas y métodos.

Polimorfismo y constructores. Binding. Interfaces y Herencia múltiple. Clases Inner. Containers: Clasificación y uso.

3)Modelado de Diseño OO con UML.

Diseño en UML. Modelando el Diseño de la Estructura estática del Sistema. Guías de diseño flexible: Cohesión y Acoplamiento. Generalización y especialización. Especialización vs. Agregación. Agregación. Principios SOLID.

4)Clases del Modelo de Diseño.

Clases de Diseño. Asociación con las Estructuras de los lenguajes de Programación. Visibilidad: de atributos, de parámetro, local. Clases Conceptuales (Dominio). Diseño de una clase. Clases Conceptuales vs. Diseño. Construcción y destrucción de Instancias. Asociaciones. Operaciones fundamentales. Herencia. Transformaciones Diseño-Código.

5)Interfaz gráfica de usuario. Modelo conceptual y desarrollo en Java

Modelo conceptual de una interfaz (GUI's). Componentes de GUIs básicos. Tipos de interfaces. Factores humanos en software interactivo. Diseño gráfico y consideraciones de estilo. Guías y reglas en el diseño. Tiempos de respuesta y despliegue. El Proceso de Diseño de interfaces. Análisis de tareas. Presentación de información. Interacción y mecanismos de control. Modelo Sintáctico-Semántico (SSOA). Modelo MVC (Model - View - Control). Arquitectura de sistemas GUI's. Manejo de eventos. Implementación de ventanas. Componentes de GUIs utilizados en Java. Applets. Librerías de componentes gráficas. Botones, campos de texto, áreas de texto, layouts. Canvas, scrollbars, frames y menús. Manejo de Eventos para componentes de GUIs. Pasos en la creación de GUIs. Modelo Swing. Catálogo de componentes Swing.

6)Aspectos avanzados de la POO.

Manejo de errores y excepciones en Java: Excepciones básicas. Captura de una excepción. Creación de excepciones. Excepciones estándar de Java. Mecanismo de manejo de excepciones try-catch-finally. Restricciones. Constructores. El Sistema de Entrada / Salida en Java: Jerarquía de clases de E/S. La clase File. Tipos de Streams: Input y Output. Interfaces más utilizadas. Readers & Writers. Creación de un archivo secuencial. Lectura Acceso directo. Escritura aleatoria de datos. Usos típicos de los I/O streams. Entrada / Salida estándar. Concurrencia: Programación concurrente y en paralelo. Concepto de proceso y tarea. Ciclo de vida de un thread. Compartiendo recursos en Java. Java Beans. Planificación por prioridades. Múltiples threads.

7)Patrones de Diseño.

Introducción a Patrones. Definición de patrón. Descripción de un patrón. Catálogo de Patrones. Patrones de diseño. Definición, descripción. Su utilidad, selección y uso. Catálogo de patrones de diseño. Patrones creacionales: Abstract Factory, Singleton. Patrones estructurales: Composite, Bridge, Decorator, Adapter. Patrones de comportamiento: Observer, State, Strategy, Template Method, Command, Mediator, Iterator.

5. RECURSOS NECESARIOS

- Proyector
- Laboratorio Informatica

6. PROGRAMACIÓN SEMANAL

Semana	Unidad / Módulo	Descripción	Bibliografía
--------	-----------------	-------------	--------------

1	1		Rumbaugh96, Eckel00, Deitel98
2	2		Eckel00, Deitel98
3	2		Eckel00, Deitel98
4	2		Eckel00, Deitel98
5	primer parcial		
6	3		Fowler99
7	4		Eckel00, Deitel98
8	4		Eckel00, Deitel98
9	5		Eckel00, Deitel98
10	5		Eckel00, Deitel98
11	5		Eckel00, Deitel98
12	5		Eckel00, Deitel98
13	6		Eckel00, Deitel98
14	7		Gamma97
15	7		Gamma97
16	segundo parcial		
17	Recepción y defensa del trabajo Práctico Final integrador. Informe fina,		

7. BIBLIOGRAFIA DE LA ASIGNATURA

Autor	Año	Título	Capítulo/s	Lugar de la Edición	Editor / Sitio Web
Rumbaugh y otros	1996	Modelado y diseño orientado a objetos			Editorial Prentice-Hall
Bruce Eckel	2000	Thinking in Java 2º ed			Editorial Prentice Hall
Deitel y Deitel	1998	Cómo programar en Java			Editorial Pearson Educación
Martin Fowler, Scott Kendall	1999	UML Gota a gota			Editorial Addison-Wesley Iberoamericana
Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides	1997	Design Patterns			Editorial Addison-Wesley

Firma del docente-investigador responsable

VISADO		
COORDINADOR DE LA CARRERA	DIRECTOR DEL INSTITUTO	SECRETARIO ACADEMICO UNTDF
Fecha :	Fecha :	

Este programa de estudio tiene una validez de hasta tres años o hasta que otro programa lo reemplace en ese periodo