

Plan Anual de Actividades Académicas 2025

Carrera: Técnico Universitario en Programación

Asignatura: Programación III

Año de ubicación del curso 2° (segundo)	Cuatrimestre 1° (primero)
--	--

CONSTITUCIÓN DEL EQUIPO DOCENTE

Apellido y nombre	Título habilitante	Cargo
Bazán Lucrecia	Técnico Universitario en Programación	Profesor adjunto
Diaz Daniel	Técnico Universitario en Programación	Profesor adjunto

Planificación de la asignatura

▪ Fundamentación

Programación III es la tercera de las materias de programación incluidas en el plan de la carrera Técnico Universitario en Programación (TUP). El objetivo fundamental de esta materia es que los alumnos adquieran la capacidad de conceptualizar aplicaciones en función del paradigma de la Programación Orientada a Objetos (POO). El desarrollo de esta capacidad implica el aprendizaje de contenidos conceptuales, así como también principios, técnicas, criterios y buenas prácticas de la POO.

▪ Objetivos

Conceptualizar y modelar software según el paradigma de la POO.

Evaluar factores de calidad del software: encapsulamiento, reusabilidad, modularidad, cohesión y acoplamiento.

Diseñar software para entornos de escritorio en función de la POO.

Conceptualizar y modelar software utilizando la POO que posea una capa de acceso a datos con persistencia en archivos JSON.

Comprender que la POO no es una conceptualización nacida de algún lenguaje orientado a objetos sino que los mismos son consecuencia de la existencia del paradigma.

▪ **Contenidos.**

	Descripción	Contenidos	Bibliografía
I	Introducción a la POO	Descripción del paradigma, diferencias entre programación estructurada y orientada a objetos. Principios de la POO: Encapsulamiento, reusabilidad, modularidad, cohesión y acoplamiento. Objetos como concepto. Características de un Objeto. Herencia y Polimorfismo como núcleo conceptual de la POO.	1,2
II	Diseño orientado a objetos	Identificando clases y sus responsabilidades, representación gráfica de clases y relaciones entre objetos con UML. Introducción a los principios SOLID y GRASP	1,2,6
III	Clases y Objetos	Clases: campos, métodos, parámetros, control de acceso, métodos como control de acceso, creación de objetos, constructores. Miembros estáticos (campos, bloques, métodos), miembros de instancia y de clase.	1,2,3,6
IV	Herencia y polimorfismo	La clase Object. Una clase extendida. Miembros de clase protegidos. Constructores. Sobreescritura de métodos y ocultación de información. La palabra "super". Métodos y clases finales. Clases y métodos abstractos. Clonando objetos. Cómo y cuando extender clases. Diseñando una clase para que sea extensible. Herencia simple versus múltiple. Extendiendo interfaces. Conflicto de nombres. Implementando interfaces. Clases anónimas.	1,2,3,6
V	Genericidad	Diferenciar entre ente genérico y ente concreto. Declarar una clase genérica. Definición de métodos genéricos. Criterios para la generalización. Crear objetos de una clase genérica.	1,2
VI	Interfaces gráficas	Concepto de interfaz gráfica. Introducción al diseño de interfaces gráficas. Eventos de usuario. Eventos de sistema.	6,5,1

▪ Bibliografía

Bibliografía Obligatoria :

1. ***Programación en Java 6. Algoritmos y programación orientada a objetos***. Luis Joyanes Aguilar, Ignacio Zahonero Martínez. *McGraw Hill*.
2. ***“Introducción a la programación orientada a objetos con Java”*** . Rafael Llobet Azpitarte , Pedro Alonso Jordá , Devesa Llinares , Emili Miedes De Elías ,María Idoia Ruiz Fuertes, Francisco Torres Goterris. *Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia*

Bibliografía Complementaria:

3. ***“Programación orientada a objetos con Java”***. David J. Barnes, Michael Kölling. *Prentice Hall*.
4. ***“Head first Java”***. Segunda Edición. Kathy Sierra, Bert Bates. *O'Reilly*.
5. ***“Beginning Java 8 APIs, Extensions and Libraries”***. Kishori Sharan. *Apress*
6. *Apuntes de cátedra*
7. ***“Learning Java”*** Primera edición. Pat Niemeyer, Jonathan Knudsen. *O'Reilly*

▪ Metodología de Trabajo

Los contenidos Programación III tienen su desarrollo práctico en Laboratorio de Computación III que se desarrollarán en forma paralela. Esto significa que ambas materias están fuertemente relacionadas y que se complementan mutuamente.

Los temas se expondrán de acuerdo a un creciente nivel de complejidad y abstracción y que irán requiriendo de una serie de conceptos previos. Acompañando el desarrollo teórico se propondrán trabajos prácticos adecuados que ejemplifiquen lo expuesto.

Se fomentará la participación activa de los alumnos, sobre todo en el desarrollo de ejemplos. Cabe aclarar que las clases serán mayormente expositivas pero aún así se fomentará el debate y participación de los alumnos.

Es importante notar que el núcleo de los trabajos prácticos de este curso es el análisis y el modelado. En paralelo a estas prácticas y durante la mayor parte del cursado se trabajará en análisis y modelado del trabajo práctico globalizador en conjunto con Laboratorio de Computación III

Durante el cursado de Programación III y separados por varias clases se realizarán trabajos prácticos grupales que engloban los temas vistos desde el último trabajo práctico. Estos trabajos prácticos se realizarán en papel con UML y pseudocódigo aplicando la POO abstrayendo los temas vistos del lenguaje de programación.

De los temas expuestos y prácticas realizadas se hará relación con la práctica profesional de modo tal que el alumno internalice la importancia y frecuencia de lo visto en la cursada.

Aprobación de cursada

Para la aprobación de la cursada se establecen: dos evaluaciones parciales individuales que se aprueban con nota igual o superior a 6 (seis). Habrá un recuperatorio para cada una de las instancias y además, se deberá cumplir con el 75% de asistencia.

Aprobación directa

Los alumnos que tengan nota superior o igual a 8 en la primera instancia de cada parcial o en solo una instancia de recuperación y cumplan con el 75% de asistencia, obtendrán la aprobación directa, es decir, no tendrán que rendir el final.

Examen Final

En caso de no aprobar directamente la materia, se debe aprobar un examen final en las fechas establecidas en el calendario académico.

Metodología

Se establece una clase de revisión integradora antes de cada escrito y la muestra y realización de la evaluación después de cada uno. Esto es para que el alumno vea sus errores y se oriente en el proceso de aprendizaje.

Fechas de parciales

Las fechas de los exámenes parciales y sus recuperatorios se muestran en el cronograma estimado de clases.

▪ Recursos didácticos a utilizar como apoyo a la enseñanza.

Según se requiera a lo largo de la cursada, y según la disponibilidad del recurso, los temas serán expuestos con el apoyo de la pizarra, proyector y presentaciones en uno de los laboratorios.

▪ **Cronograma estimado de clases.**

Comisión 11		
Semana	Día	Tema
1	lunes, 10 de marzo	Introducción a la materia. Introducción a fundamentos de POO: clases, objetos, métodos, atributos, parámetros, constructores, campos de instancia y de clase.
	lunes, 10 de marzo	
2	lunes, 17 de marzo	Herencia y polimorfismo: Una clase extendida. La palabra "super". Sobreescritura y sobrecarga de métodos. Métodos y clases finales. Clases y métodos abstractos. Diseñando una clase para que sea extensible. Static y Non Static.
	lunes, 17 de marzo	
3	lunes, 24 de marzo	FERIADO - Día Nac. de la Memoria por la Verdad y la Justicia
	lunes, 24 de marzo	FERIADO - Día Nac. de la Memoria por la Verdad y la Justicia
4	lunes, 31 de marzo	Abstracción y encapsulamiento: Identificando clases y sus responsabilidades, representación gráfica de clases y relaciones entre objetos (UML). Control de acceso: public, private y protected para métodos y atributos.
	lunes, 31 de marzo	POO con Java: String vs StringBuilder. Introducción a los Arrays (Arreglos) en Java. Cómo Declarar, Instanciar e Inicializar un array. Manipulación de los arrays. Guia #1
5	lunes, 7 de abril	POO con Java: Resolución de Guia #1.
	lunes, 7 de abril	POO con Java: Clase Object. Métodos toString(), equals(), hashCode() y getClass(). Sentencia instanceof. Enumeradores. Interfaces. Herencia simple vs multiple. Extendiendo y diseñando interfaces. Paquetes.
6	lunes, 14 de abril	POO con Java: Guia #2
	lunes, 14 de abril	POO con Java: Repaso y consultas. Resolución Guia #2
7	lunes, 21 de abril	POO con Java: Guia #3.
	lunes, 21 de abril	POO con Java: Repaso y consultas. Resolución Guia #3. Simulacro de parcial.
8	lunes, 28 de abril	Consultas previas al parcial.
	lunes, 28 de abril	Primer parcial
9	lunes, 05 de mayo	Collections: Set, List, Queue, Map. Cuándo utilizar cada una. Rendimiento en las operaciones de búsqueda, acceso aleatorio e inserción.
	lunes, 05 de mayo	Collections: Clases contenedoras. Guia #4
10	lunes, 12 de mayo	Collections: Resolución de Guia #4.
	lunes, 12 de mayo	Exceptions: throw, throws, try, catch y finally. Cuando usar excepciones. Excepciones verificadas (checked exceptions) y excepciones no verificadas (unchecked exceptions). Excepciones personalizadas. Guia #5
11	lunes, 19 de mayo	Exceptions: Resolución de Guia #5.
	lunes, 19 de mayo	Genericidad: Clases, métodos e interfaces genéricas. Ventajas y limitaciones. Crear objetos de una clase genérica. Tipo genérico simple. Tipos comodín. Colecciones genéricas en Java. Guia #6
12	lunes, 26 de mayo	Genericidad: Resolución de Guia #6.
	lunes, 26 de mayo	JSON: Definición y características. Sintaxis. Serialización y deserialización con Java. Guia #7
13	lunes, 02 de junio	JSON: Resolución de Guia #7.
	lunes, 02 de junio	Repaso y consultas previas al examen. Simulacro de parcial.
14	lunes, 09 de junio	Consultas previas al examen.
	lunes, 09 de junio	Segundo Parcial
15	lunes, 16 de junio	FERIADO - Día de Paso a la Inmortalidad del Gral. Don Martin de Guemes
	lunes, 16 de junio	FERIADO - Día de Paso a la Inmortalidad del Gral. Don Martin de Guemes
16	lunes, 23 de junio	Recuperatorio primer parcial
	lunes, 23 de junio	Recuperatorio segundo parcial

Comisión 22 - PROGRAMACION

Semana	Día	Tema
1	11/03/25	Introducción a la materia. Introducción a Java. Sintaxis.
	11/03/25	Java - Sintaxis, Guia Inicial y resolución.
2	18/03/25	Guia #1
	18/03/25	Resolución de Guia #1. Repaso y consultas.
3	25/03/25	POO con Java: Abstracción y encapsulamiento. Como diseñar clases. Diagrama UML. Ejemplos.
	25/03/25	Guia #2
4	01/04/25	POO con Java: Referencias polimórficas y Casting. Composición vs Herencia. Aplicación a diagrama UML.
	01/04/25	POO con Java: String vs StringBuilder.
5	08/04/25	Resolución de Guia #3. Repaso y consultas.
	08/04/25	Collections: Set, List, Queue, Map. Cuando utilizar cada una. Rendimiento en las operaciones de búsqueda, acceso aleatorio e inserción.
6	15/04/25	Resolución Guia #4. Repaso y consultas.
	15/04/25	Ejercitación integradora. Revisión y consultas.
7	22/04/25	- Primer parcial PROGRAMACION-
	22/04/25	- Primer parcial PROGRAMACION-
8	29/04/25	Exceptions: throw, throws, try, catch y finally. Cuando usar excepciones. Excepciones verificadas (checked exceptions) y excepciones no verificadas (unchecked exceptions).
	29/04/25	Guia #5
9	06/05/25	Guia #6
	06/05/25	Resolución Guia #6. Repaso y consultas.
10	13/05/25	Guia #7
	13/05/25	Resolución Guia #7. Repaso y consultas.
11	20/05/25	- Segundo parcial PROGRAMACION-
	20/05/25	- Segundo parcial PROGRAMACION-
12	27/05/25	Consultas Recuperatorio
	27/05/25	Consultas Recuperatorio
13	03/06/25	Consultas Recuperatorio
	03/06/25	Consultas Recuperatorio
14	10/06/25	-Recuperatorio de Primer Parcial PROGRAMACION-
	10/06/25	-Recuperatorio de Primer Parcial PROGRAMACION-
15	17/06/25	- Recuperatorio de Segundo Parcial PROGRAMACION-
	17/06/25	- Recuperatorio de Segundo Parcial PROGRAMACION-
16	24/06/25	Cierre de Notas
	24/06/25	Cierre de Notas