

## Plan Anual de Actividades Académicas 2025

**Carrera: Técnico Universitario en Programación**

**Asignatura: Laboratorio de Computación III**

<b>Año de ubicación del curso</b> <b>2° (segundo)</b>	<b>Cuatrimestre</b> <b>1° (primero)</b>
--	--

### CONSTITUCIÓN DEL EQUIPO DOCENTE

<b>Apellido y nombre</b>	<b>Título habilitante</b>	<b>Cargo</b>
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 un trabajo práctico final obligatorio se aprueba con nota igual o superior a 6 (seis). 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 y TP final 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	viernes, 14 de marzo	Introducción a la materia. Introducción a fundamentos de POO: clases, objetos, métodos, atributos, parámetros, constructores, campos de instancia y de clase.
	viernes, 14 de marzo	
2	viernes, 21 de marzo	<b>Herencia y polimorfismo:</b> 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.
	viernes, 21 de marzo	
3	viernes, 28 de marzo	<b>Abstracción y encapsulamiento:</b> 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.
	viernes, 28 de marzo	
4	viernes, 4 de abril	<b>POO con Java:</b> String vs StringBuilder. Introducción a los Arrays (Arreglos) en Java. Cómo Declarar, Instanciar e Inicializar un array. Manipulación de los arrays. Guía #1
	viernes, 4 de abril	
5	viernes, 11 de abril	<b>POO con Java:</b> Clase Object. Métodos toString(), equals(), hashCode() y getClass(). Sentencia instanceof. Enumeradores. Interfaces. Herencia simple vs multiple. Extendiendo y diseñando interfaces. Paquetes.
	viernes, 11 de abril	<b>POO con Java:</b> Guía #1 y Guía #2
6	viernes, 18 de abril	<b>FERIADO - Viernes Santo</b>
	viernes, 18 de abril	<b>FERIADO - Viernes Santo</b>
7	viernes, 25 de abril	Consultas previas al parcial.
	viernes, 25 de abril	<b>Primer parcial</b>
8	viernes, 02 de mayo	<b>FERIADO - Día del Docente Tecnológico</b>
	viernes, 02 de mayo	<b>FERIADO - Día del Docente Tecnológico</b>
9	viernes, 09 de mayo	<b>Collections:</b> Set, List, Queue, Map. Cuándo utilizar cada una. Rendimiento en las operaciones de búsqueda, acceso aleatorio e inserción. Clases contenedoras. Guía #4
	viernes, 09 de mayo	<b>Collections:</b> Resolución de Guía #4.
10	viernes, 16 de mayo	<b>Exceptions:</b> throw, throws, try, catch y finally. Cuando usar excepciones. Excepciones verificadas (checked exceptions) y excepciones no verificadas (unchecked exceptions). Excepciones personalizadas. Guía #5
	viernes, 16 de mayo	<b>Exceptions:</b> Resolución de Guía #5.
11	viernes, 23 de mayo	<b>Genericidad:</b> Clases, métodos e interfaces genéricas. Ventajas y limitaciones. Crear objetos de una clase genérica. Tipo genérico simple. Tipos comodín. Colecciones genéricas en Java. Guía #6
	viernes, 23 de mayo	<b>Genericidad:</b> Resolución de Guía #6.
12	viernes, 30 de mayo	<b>JSON:</b> Definición y características. Sintaxis. Serialización y deserialización con Java. Guía #7
	viernes, 30 de mayo	<b>JSON:</b> Resolución de Guía #7. Consultas previas al examen.
13	viernes, 06 de junio	Consultas previas al examen.
	viernes, 06 de junio	<b>Segundo Parcial</b>
14	viernes, 13 de junio	Resolución de Segundo parcial. Entregas de TP Final.
	viernes, 13 de junio	Entregas de TP Final.
15	viernes, 20 de junio	<b>FERIADO - Día de la Bandera</b>
	viernes, 20 de junio	<b>FERIADO - Día de la Bandera</b>
16	viernes, 27 de junio	<b>Recuperatorio de parcial 1 y TP final</b>
	viernes, 27 de junio	<b>Recuperatorio de parcial 2 y TP final</b>

Comisión 22		
Semana	Día	Tema
1	13/03/25	Introducción a POO: Clases, objetos, métodos, atributos, parámetros y constructores.
	13/03/25	Introducción a POO: control de acceso, metodos como control de acceso, campos de instancia y de clase.
2	20/03/25	POO con Java: Herencia y polimorfismo. Una clase extendida. Miembros de clase protegidos. Sobreescritura y sobrecarga de métodos y ocultación de información. La palabra "super".
	20/03/25	POO con Java: Métodos y clases finales. Clases y métodos abstractos. Cómo y cuando extender clases. Diseñando una clase para que sea extensible.
3	27/03/25	Resolución de Guia #2. Repaso y consultas.
	27/03/25	POO con Java: Clase Object. Metodos toString(), equals(), hashCode() y getClass(). Sentencia instanceof. Enumeradores.
4	03/04/25	POO con Java: Interfaces. Herencia simple vs multiple. Extendiendo y diseñando interfaces. Paquetes.
	03/04/25	Guia #3
5	10/04/25	Collections: Repaso Collections. Clases contenedoras. Collection de collections. Ejercitación en conjunto.
	10/04/25	Guia #4
6	17/04/25	Sin Actividad- Jueves Santo
	17/04/25	Sin Actividad- Jueves Santo
7	24/04/25	- Primer parcial LABORATORIO-
	24/04/25	- Primer parcial LABORATORIO-
8	01/05/25	Sin Actividad – Día del Trabajador (Ley 21329)
	01/05/25	Sin Actividad – Día del Trabajador (Ley 21329)
9	08/05/25	Genericidad: Clases, métodos e interfaces genericas. Ventajas y limitaciones. Crear objetos de una clase generica. Tipo genérico simple. Tipos comodín. Colecciones genéricas en Java.
	08/05/25	Genericidad: Clases, métodos e interfaces genericas. Ventajas y limitaciones. Crear objetos de una clase generica. Tipo genérico simple. Tipos comodín. Colecciones genéricas en Java.
10	15/05/25	Archivos: Manejar archivos como objetos de una clase. Abrir, leer y cerrar un archivo. Leer y escribir datos en un archivo secuencial. Utilizar la jerarquía de clases definida en el entorno de Java para el manejo de archivos. Utilizar la clase File para procesar archivos.
	15/05/25	JSON: Definición y características. Sintaxis. Serialización y deserialización con Java
11	22/05/25	Ejercitación integradora.
	22/05/25	Revisión y consultas.
12	29/05/25	- Segundo parcial LABORATORIO-
	29/05/25	- Segundo parcial LABORATORIO-

13	05/06/25	Entrega de TP Final.
	05/06/25	Entrega de TP Final.
14	12/06/25	Recuperatorio 1er Parcial LABORATORIO
	12/06/25	Recuperatorio 1er Parcial LABORATORIO
15	19/06/25	-Recuperatorio de TP FINAL-
	19/06/25	-Recuperatorio de TP FINAL-
16	26/06/25	Recuperatorio 2do Parcial LABORATORIO
	26/06/25	Recuperatorio 2do Parcial LABORATORIO