

**UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y
TECNOLOGÍAS**

PLANIFICACIÓN ANUAL 2023

ASIGNATURA: PROGRAMACIÓN III

**PROGRAMADOR UNIVERSITARIO EN
INFORMÁTICA
Plan de Estudio: 2017**

Equipo cátedra:

Equipo cátedra:

Profesor Adjunto: Méndez Analía

JTP: Zarco Raquel

Auxiliar Docente: Gómez Claudia

Ayudante Estudiantil: Autalán Ana



PLANIFICACIÓN DE LA ASIGNATURA

1- IDENTIFICACIÓN:

- 1.1. **Nombre de Asignatura:** Programación III.
- 1.2. **Carrera:** Programador Universitario en Informática.
- 1.3. **Plan de Estudios:** 2017.
- 1.4. **Año académico:** 2023.
- 1.5. **Carácter:** obligatoria.
- 1.6. **Ubicación de la Asignatura en el Plan de Estudios**
 - 1.6.1. **Módulo – Año:** Cuarto módulo – Segundo año.
 - 1.6.2. **Área/Bloque/Tramo al que pertenece la Asignatura/Obligación Curricular, según la organización del Plan de Estudios:**

ÁREAS/BLOQUE/TRAMO	CARGA HORARIA PRESENCIAL
Estructuras de Datos	25
Algoritmia básica	25
Paradigmas de Programación	25
CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR	75

Tabla 1: Carga horaria por área/bloque/tramo

- 1.6.3. **Correlativas**
 - 1.6.3.1 **Anteriores:** Laboratorio II, Programación I, Programación II, Laboratorio III.
 - 1.6.3.2 **Posteriores:** Seminario de Ética y Deontología, Práctica Profesional.
- 1.7. **Carga horaria:**
 - 1.7.1. **Carga horaria semanal total**
 - 1.7.1.1. **Presencial:** 5.
 - 1.7.1.2. **No Presencial:** no indicado.
 - 1.7.2. **Carga horaria semanal destinada a la formación práctica**
 - 1.7.1.3. **Presencial:** 3.
 - 1.7.1.4. **No Presencial:** no indicado.
 - 1.7.3. **Carga horaria total dedicada a la formación práctica:** 45.



1.8. Ámbitos donde se desarrollan las actividades de formación práctica a las que se hace referencia en el punto anterior.

Las actividades de formación práctica se desarrollan en laboratorios de computadoras pertenecientes al Departamento de Informática.

1.9. Indique si la asignatura se dicta en más de una comisión: una comisión.

2- PRESENTACIÓN

2.1. Ubicación de la Asignatura como tramo de conocimiento de una disciplina.

Esta asignatura está orientada fundamentalmente al estudio, análisis y manejo de estructuras de datos avanzadas, la recursividad en la programación y los algoritmos de ordenamiento y búsqueda. En ella se abarcan:

- Estructuras de datos avanzadas, como árboles, grafos y archivos.
- Recursividad.
- Algoritmos de ordenamiento y búsqueda.
- Resolución de problemas y desarrollo de algoritmos mediante la aplicación del Paradigma de Programación Orientado a Objetos.

2.2. Conocimientos y habilidades previas que permiten encarar el aprendizaje de la Asignatura.

- Conocimientos y destrezas en programación de computadoras, tipos básicos y estructuras de datos encadenadas adquiridos en Programación I y Programación II.
- Destrezas en la programación avanzada en lenguaje Java y la programación orientada a objetos, adquiridas en Laboratorio II y Laboratorio III.

2.3. Aspectos del Perfil Profesional del Egresado a los que contribuye la asignatura.

Esta asignatura contribuye a:

- La formación técnica del alumno, abarcando los conocimientos destinados al aprendizaje y al uso de lenguajes de programación y herramientas informáticas en entornos específicos para la resolución de los problemas concretos.
- Preparar recursos humanos calificados y competentes, de nivel universitario, en el ámbito técnico de la Informática.

En esta asignatura se proporciona al alumno la capacitación básica necesaria para:

- Implementar, probar, documentar y mantener aplicaciones software.
- Participar en las soluciones de programación en el sector de SSI.



2.4. Integración horizontal y vertical con otras asignaturas.

- **Integración vertical.**

La integración vertical de esta asignatura se establece, en principio, con la asignatura Programación II donde los alumnos aprendieron a utilizar estructuras de datos encadenadas.

Con la asignatura Laboratorio III, que se dicta durante el primer cuatrimestre del segundo año de la carrera, también se organiza una integración vertical. El desarrollo de destrezas para la creación de aplicaciones de software aplicando el paradigma de POO, preparan al alumno para el cursado de Programación III.

- **Integración horizontal**

Así mismo, los temas que se dictan en Arquitectura del Computador le sirven al alumno para poder interpretar como se lleva a cabo la ejecución de programas que se desarrollan en Programación III utilizando el lenguaje de programación Java.

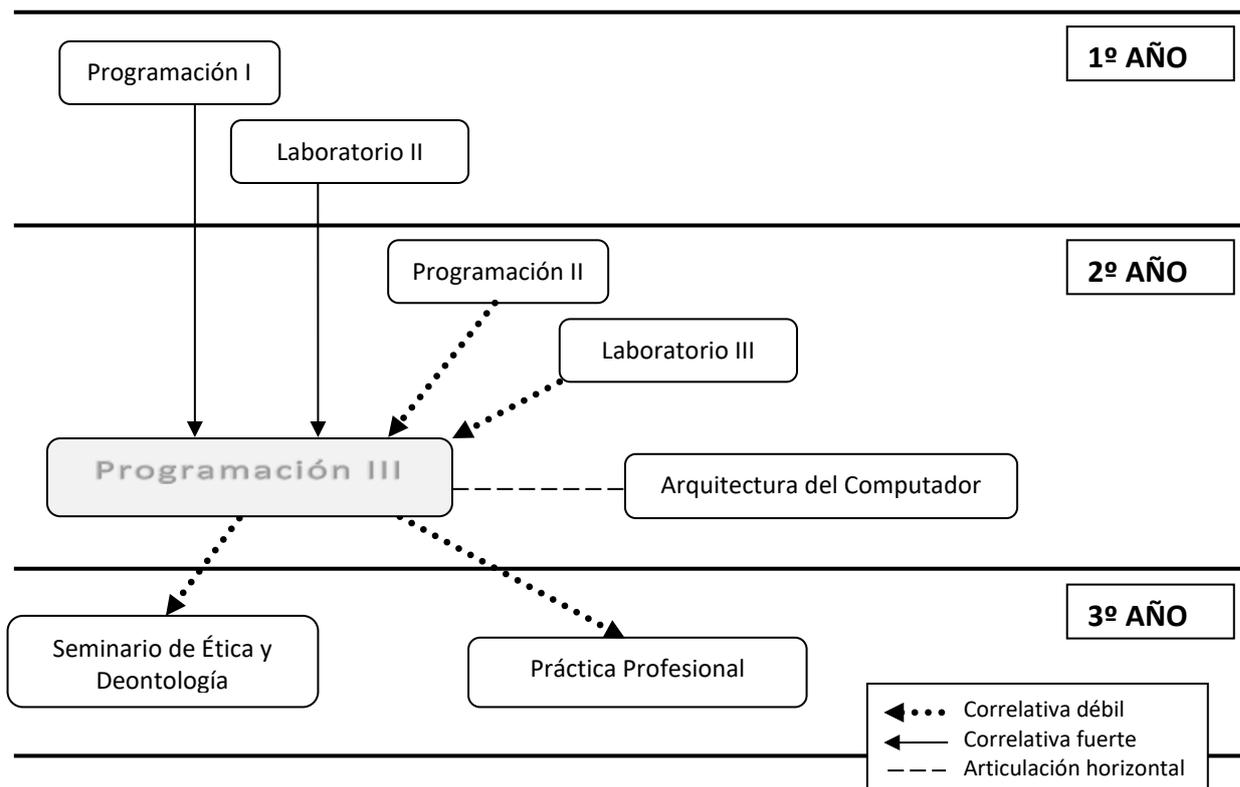


Figura 1. Integración vertical y horizontal con otras asignaturas.



3- OBJETIVOS

Que el alumno desarrolle las siguientes competencias básicas, transversales y específicas.

Competencias básicas

- Capacidad para buscar, seleccionar y procesar la información necesaria para la resolución de problemas.
- Capacidad para verificar las soluciones encontradas.

Competencias transversales

- Capacidad para relacionar la teoría a la práctica.
- Capacidad para identificar, plantear y resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.
- Capacidad para la comunicación oral y escrita enfocada en el intercambio y transmisión de los conocimientos.

Competencias específicas.

- Emplear los conceptos y técnicas de la POO en la resolución de problemas.
- Desarrollar creativamente soluciones algorítmicas seleccionando criteriosamente la alternativa de estructura de datos más adecuada aplicando POO.
- Crear aplicaciones utilizando estructuras de datos avanzadas aplicando el paradigma de programación orientado a objetos.
- Distinguir las características básicas de un algoritmo recursivo para diferenciarlo de un algoritmo iterativo.
- Modelar el manejo de memoria en ejecución durante el desempeño de un algoritmo recursivo.
- Componer aplicaciones de software siguiendo el paradigma de orientación a objetos mediante el uso de un lenguaje de programación orientado a objetos.

4- SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

4.1. Contenidos mínimos establecidos en el Plan de Estudios para la Asignatura:

Recursividad. Características. Algoritmos recursivos. Árboles: conceptos básicos. Grafos: conceptos básicos. Algoritmos de ordenamiento y búsqueda: Heap Sort y Quick Sort. Archivos de datos. Claves, registro físico o bloque. Factor de bloqueo. Organización de archivos. Operaciones básicas sobre archivos.



4.2. Programa Sintético sobre la base de los contenidos mínimos

Unidades 1: Archivos

Archivos de datos. Claves, registro físico o bloque. Factor de bloqueo. Organización de archivos. Operaciones básicas sobre archivos.

Unidad 2: Recursividad.

Recursividad: definición y características. Algoritmos recursivos. Creación, seguimiento, convocatoria.

Unidad 3: Estructuras de datos no lineales.

Árboles: conceptos básicos. Árboles binarios de búsqueda. Grafos: conceptos básicos. Camino más corto con un solo origen. Árbol de expansión de costo mínimo.

Unidad 4: Algoritmos de ordenamiento y búsqueda

Algoritmos de ordenamiento y búsqueda: Heap Sort y Quick Sort. Montículo. Intercalación y ordenamiento de archivos.

4.3. Articulación Temática de la Asignatura /Obligación Curricular

En la figura 2 se presentan los principales conceptos a tratar en la asignatura.

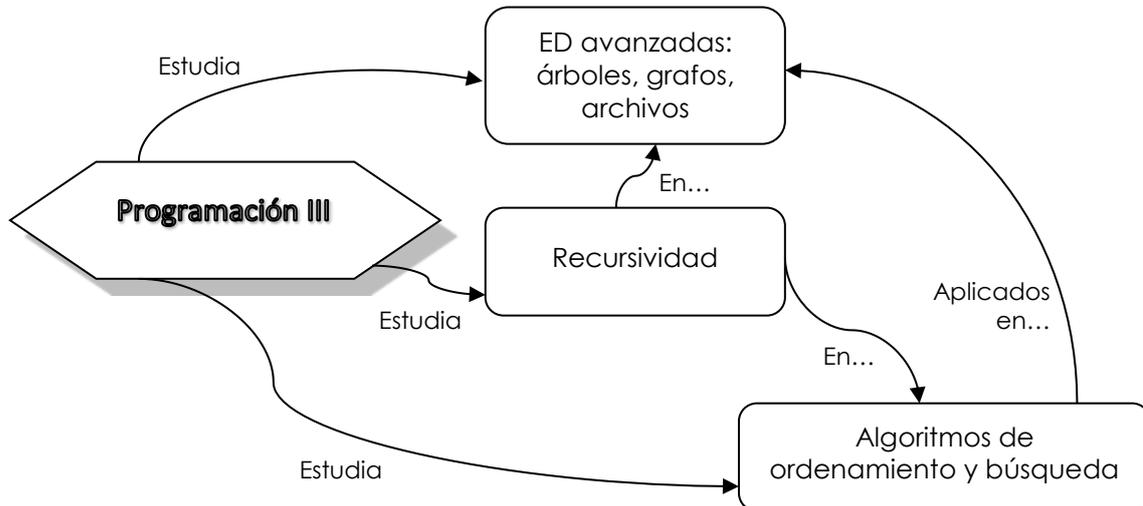


Figura 2. Articulación temática de la asignatura Programación III.



4.4. Programa Analítico

Unidades 1: Archivos

Archivos de datos: Definiciones. Campos, claves, registro físico o bloque. Factor de bloqueo. Organización de archivos: secuencial, directa, secuencial indexada. Modos de acceso. Operaciones básicas sobre archivos.

Unidad 2: Recursividad.

Recursividad: definición y características. Requisitos para la recursión. Tipos de datos recursivos. Algoritmos recursivos. Creación, seguimiento, convocatoria. Eficiencia de la recursión. Manejo de memoria en ejecución.

Unidad 3: Estructuras de datos no lineales.

Árboles: conceptos básicos. Terminología. Árboles generales. Árboles binarios. TAD árbol binario. Formas de representación. Árboles binarios de búsqueda. Grafos: conceptos básicos. TAD grafo. Camino más corto con un solo origen: Algoritmo de Dijkstra. Árbol de expansión de costo mínimo: Algoritmo de Kruskal, Algoritmo de Prim.

Unidad 4: Algoritmos de ordenamiento y búsqueda

Algoritmos de ordenamiento y búsqueda: Heap Sort y Quick Sort. Montículo. Intercalación y ordenamiento de archivos. Mezcla directa. Mezcla equilibrada. Búsqueda externa. Búsqueda en archivos secuenciales. Búsqueda secuencial. Búsqueda Binaria. Búsqueda por transformación de claves en archivos y solución de colisiones.

4.5. Cronograma para el desarrollo de las Unidades Temáticas

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Unidad 1: Archivos	8	Semanas 1 a 5
Unidad 2: Recursividad.	4	Semanas 6 a 7
Unidad 3: Estructuras de datos avanzadas: Árboles y Grafos.	10	Semanas 8 a 11
Unidad 4: Algoritmos de ordenamiento y búsqueda	8	Semanas 12 a 15
TOTAL	30	

Tabla 2: Cronograma para el desarrollo de las unidades temáticas



4.6. Programa y cronograma de formación práctica

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
TP Procesamiento de Archivos	12	Semanas 1 a 6
TP Recursividad	9	Semanas 7 a 8
TP Árboles	15	Semanas 9 a 12
TP Algoritmos sobre Grafos	9	Semanas 13 a 15
TOTAL	45	

Tabla 3: Cronograma para el desarrollo de los trabajos prácticos

5- BIBLIOGRAFÍA.

TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Manual de algorítmica: recursividad, complejidad y diseño de algoritmos	Bisbal Riera J.	Editoria UOC	Biblioteca Digital eLibro.net	2013
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/56561/		
Estructuras de datos. 3º Edición	Cairo Osvaldo – Guardati Silvia	McGraw-Hill	1 en Centro de Documentación	2006
Estructuras de Datos y Algoritmos	Fritelli, Valerio - Guzman, A. - Tymoschuk, J.	Jorge Sarmiento Editor - Universitas	Biblioteca Digital eLibro.net	2020
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/175249/		
UML: arquitectura de aplicaciones en Java, C++ y Python (2a. ed.)	Jiménez de Parga C..	RA-MA Editorial	Biblioteca Digital eLibro.net	2021
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/222720/		
Estructuras de datos en Java	Joyanes Aguilar Luis, Zahonero Martinez Ignacio	McGraw-Hill	1 en Centro de Documentación	2008
Programación orientada a objetos en C++ y Java: un acercamiento interdisciplinario	López Goytia J.L.	Grupo Editorial Patria	Biblioteca Digital eLibro.net	2015
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/39461/		

Tabla 4: Bibliografía



Tabla 4: Bibliografía (continuación)

TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Grafos: fundamentos y algoritmos	Moreno E.	Editorial ebooks Patagonia – J.C. Sáez	Biblioteca Digital eLibro.net	2012
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/68438/		
Lógica de programación orientada a objetos	Oviedo Regino E. M.	Ecoe Ediciones	Biblioteca Digital eLibro.net	2015
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/70431/		
Estructura de datos en JAVA	Weiss Mark Allen.	Pearson	2 en Centro de Documentación	2013

6- ESTRATEGIAS METODOLÓGICAS.

6.1. Aspectos pedagógicos y didácticos

La metodología de enseñanza que se aplicará en el aula será en un espacio de diálogo y construcción, en el que se trabaje interactuando permanentemente, y de esta manera tanto los alumnos como el docente se consideran fuente de información.

Las actividades de estudio tanto prácticas como de tipo teórico serán apoyadas y desarrolladas mediante la plataforma Moodle del Centro Universitario Virtual (CUV - <http://cuv.unse.edu.ar>). Mediante esta plataforma se acercarán al estudiante diferentes recursos tales como videos tutoriales y materiales de estudio creados por los docentes, se utilizarán herramientas como cuestionarios evaluativos y de diagnóstico, así como la presentación digital de las actividades y proyectos propuestos en las prácticas. Cada contenido desarrollado es mediado para su mejor comprensión y con el fin de propiciar el diálogo y la discusión.

Tanto en las clases teóricas como prácticas se aplicará la metodología de aula invertida. Los alumnos realizarán actividades de aprendizaje y de evaluación formativa previamente a los encuentros con los docentes. Durante las clases se completarán las actividades requeridas para el abordaje de cada tema y se organizarán tareas individuales y grupales colaborativas que favorezcan la apropiación del estudio, y que el alumno deberá completar y enviar mediante la plataforma del CUV, al finalizar la clase.

Cada uno de los trabajos prácticos propuesto en esta asignatura incluye el desarrollo de problemas mediante diagramación UML de clases y diagramación de flujo, así como ejercicios en los que se solicita la programación completa en lenguaje Java de la resolución del enunciado. Los conceptos teóricos y los aspectos metodológicos de la



práctica requerida son abordados tanto en los encuentros teóricos como en los prácticos, a fin de fortalecer las habilidades de programación en el alumno.

Se utiliza software libre en la tarea de desarrollar habilidades prácticas en el alumno, necesarias para la resolución de problemas y el diseño de algoritmos y para prepararlos a enfrentar los distintos cambios del mundo tecnológico.

6.2. Mecanismos para la integración de docentes

Se efectuarán reuniones con la cátedra de la asignatura Laboratorio III en la cual los alumnos aprenden lenguaje Java avanzado. Estas reuniones permitirán realizar un seguimiento temático acorde.

Por otra parte, con las cátedras de asignaturas como Programación I y Programación II, la coordinación se realiza mediante encuentros anuales en los que se revisan los logros alcanzados en la comprensión y aplicación del paradigma de programación estudiado y las actividades prácticas.

6.3. Recursos Didácticos

Se utilizarán diversos recursos didácticos a fin del logro de los objetivos propuestos, a saber:

- Ambiente integrado de desarrollo (IDE) para la programación en lenguaje Java.
- Para desarrollar aplicaciones en Java se necesita la herramienta de desarrollo Java Development Kit (JDK). JDK incluye Java Runtime Environment, el compilador Java y las API de Java.
- Bibliografía actualizada digital y apuntes elaborados por el equipo docente de la asignatura, disponibles en la plataforma del CUV.
- En las clases teóricas y en las clases prácticas se utilizará enunciados tomados de las actividades prácticas propuestas, diapositivas y videos elaborados por los docentes de la asignatura, demos, tutoriales, etc.
- Se contará con el espacio de la asignatura en la plataforma Moodle del Centro Universitario Virtual (CUV), que permitirá al alumno acceder a consultas virtuales (on-line o secuencial) a distancia, acceso a los recursos digitales, apoyo de prácticas, etc. También brindará a los alumnos un canal de comunicación permanente donde podrán acceder a información actualizada de la asignatura: fechas de evaluaciones, resultados de parciales, condición final de la cursada, etc.

7- EVALUACIÓN

7.1. Evaluación Diagnóstica

Teniendo en cuenta que la evaluación diagnóstica no sólo es una estimación, sino que tiene como propósito contribuir al aprendizaje, se llevará a cabo una única evaluación



diagnóstica, al comienzo de las clases, cuya finalidad será determinar el nivel de conocimientos y habilidades previas que permitan encarar el aprendizaje de la asignatura.

Los contenidos sobre los que se evaluará serán: Estructuras encadenadas: listas, pilas y colas, conceptos avanzados sobre la POO: herencia, clases abstractas, interfaces, excepciones, entre otros.

La evaluación diagnóstica será especialmente diseñada, individual, escrita y objetiva.

Se hará una prueba de opción múltiple para que el alumno marque la opción correcta. El nivel de calificación será cuantitativa politómica. Escala: 1 al 10.

7.2. Evaluación Formativa

Se implementará un proceso de evaluación formativa de carácter sistemático, cualitativo y continuo, se pretende resaltar el papel de evaluación orientadora, reguladora y motivadora. Se busca obtener información relevante respecto al proceso de enseñanza – aprendizaje que permita comprender cómo se produce dicho proceso y tomar las decisiones pertinentes en vistas a mejorar tanto el desarrollo como los resultados del aprendizaje. Permitirá identificar la evolución en el aprendizaje de los alumnos y el grado de impacto de la propuesta educativa que lleva a cabo la cátedra.

Mediante la evaluación formativa se motivará, reforzará y proporcionará ayuda a los estudiantes, reconociendo y observando sus aprendizajes, dificultades y posibilidades.

La evaluación formativa de los contenidos teóricos se llevará a cabo mediante la aplicación de herramientas disponibles en la plataforma Moodle del CUV. En particular se utilizará la herramienta Cuestionario (CD), siguiendo la técnica de la pregunta. Regularmente, y como una actividad de motivación, al comienzo o finalización de clase teórica se implementarán dinámicas de evaluación formativa, tales como escrito de un minuto, tarjetas de aplicación, preguntas abiertas de respuesta breve, cuyos resultados serán luego discutidos en aula.

En cuanto a las competencias relacionadas a la práctica, la evaluación formativa se llevará adelante aplicando la técnica de casos y se estructurará mediante la entrega regulada de diferentes enunciados propuestos en cada trabajo práctico. El seguimiento del desempeño en la práctica por parte de cada alumno podrá ser monitorizado por el docente mediante las herramientas disponibles en la plataforma Moodle del CUV.

Los análisis del aprendizaje (Learning Analytics) provistos por la plataforma del CUV proporcionarán un claro registro del progreso o no del alumno durante el cursado de la asignatura.



7.3. Evaluación Parcial

7.3.1. Programa de Evaluaciones Parciales

EVALUACIÓN	TEMA	MODALIDAD	SEM	FECHA	DEVOLUCIÓN DE RESULTADO
Parcial Práctico 1	Archivos	Individual. Escrito. En computadora. Resolución de un problema propuesto mediante: <ul style="list-style-type: none">• Diagramación UML de clases aplicando Orientación a Objetos.	6	22/09	26/09
Recuperatorio 1			8	06/10	10/10
Parcial Práctico 2	Recursividad y Árboles	• Diagramación de flujo. • Desarrollo de proyecto de POO en computadora aplicando lenguaje Java.	13	10/11	14/11
Recuperatorio 2			14	17/11	21/11

7.3.2. Criterios de Evaluación

ESTRUCTURA	OBJETIVO	CRITERIOS DE EVALUACIÓN
Diagramación UML de clases aplicando Orientación a Objetos	Determinar si los alumnos han adquirido las competencias para: <ul style="list-style-type: none">• Modelar y documentar la resolución de un problema mediante diagramación UML de clases, identificando las clases y las relaciones entre ellas.	<ul style="list-style-type: none">• Correcto uso de la diagramación UML, con identificación de relaciones entre clases.
Diagramación de flujo de operaciones básicas de la estructura de datos en estudio	<ul style="list-style-type: none">• Emplear las estructuras de datos estudiadas, y sus operaciones básicas, en la resolución de problemas mediante diagramación de flujo.	<ul style="list-style-type: none">• Correcta representación de las operaciones de la/s estructura/s de datos en estudio.
Diagramación de flujo de la solución de un problema planteado		<ul style="list-style-type: none">• Selección eficiente de la estructura de datos adecuada a la resolución del problema.• Correcta aplicación de estructuras de diagramación.• Resolución algorítmica en tiempo y forma adecuados.



ESTRUCTURA	OBJETIVO	CRITERIOS DE EVALUACIÓN
Desarrollo de proyecto de POO para la resolución de un problema planteado utilizando el lenguaje de programación Java.	Determinar si los alumnos han adquirido las competencias para: <ul style="list-style-type: none">• Desarrollar una solución de programación empleando la orientación a objetos mediante la creación de un proyecto en lenguaje Java utilizando un Entorno de Desarrollo Integrado y aplicando la reutilización de código.	<ul style="list-style-type: none">• Correcto análisis y diseño de la solución al problema propuesto aplicando la POO.• Eficiente codificación en lenguaje Java del enunciado, verificación y funcionamiento de la aplicación.• Capacidad para el análisis crítico y el aporte de mejoras sobre código desarrollado.• Código organizado y claro, incluyendo comentarios, indentación y modularidad.

7.3.3. Escala de Valoración

La escala de valoración a emplear en los parciales y recuperatorios será cualitativa dicotómica (aprobado – desaprobado).

7.4. Evaluación Integradora

La evaluación integradora se llevará a cabo a través de un Trabajo Integrador Final en el que los alumnos deberán elaborar y exponer un resumen sobre temas teóricos asignados respecto a la estructura de datos Grafos.

7.5. Evaluación Sumativa

7.5.1. Condiciones para lograr la promoción sin Examen Final de la Asignatura.

- Asistir como mínimo al 80% del total de sesiones.
- Aprobar los Cuestionarios Evaluativos con un puntaje ≥ 7 puntos.
- Aprobar los Parciales Prácticos.
- Aprobar el Trabajo Integrador.

7.5.2. Condiciones para lograr la Regularidad de la Asignatura.

- Aprobar los Cuestionarios Evaluativos o sus instancias recuperatorias.
- Aprobar los dos (2) Parciales Prácticos previstos, o sus correspondientes Recuperatorios.

7.6. Examen Final.

En el examen final los alumnos serán evaluados sobre los contenidos teóricos y prácticos de la asignatura. El examen podrá ser oral o escrito, e individual.

7.7. Examen Libre.

Para el examen Libre el alumno deberá aprobar las tres (3) instancias que se detallan a continuación, siendo cada una de ellas eliminatorias:



Primera Etapa.

Presentación, prueba y defensa de un planteamiento práctico en lenguaje Java, con evaluación en computador, cuya temática y planteo deberá ser solicitado a la cátedra con al menos 7 (siete) días de anticipación a la fecha de examen. El trabajo deberá ser presentado en soporte digital e impreso, cumplimentando las pautas establecidas en el enunciado. El trabajo se presentará en la fecha del examen y deberá ser aprobado por el tribunal.

Segunda Etapa.

Aprobar una evaluación presencial consistente en:

- Práctica escrita con diagramación UML de clases y diagramación de flujo, aplicando Orientación a Objetos.
- Práctica en computador, utilizando el lenguaje Java y Programación Orientada a Objetos.

Tercera Etapa.

Aprobar una evaluación oral/escrita de contenidos teóricos del programa analítico.

Méndez Analía

.....
Profesor responsable de Asignatura