UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO

FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS

PLANIFICACIÓN ANUAL 2024

ASIGNATURA: ESTRUCTURAS DE DATOS Y PROGRAMACIÓN II

LICENCIATURA EN SISTEMAS DE INFORMACIÓN

Plan de Estudio: 2011

Equipo cátedra:

Profesor Asociado: Méndez Analía

JTP: Zarco Raquel

Ayudante de 1º: Gómez Claudia

PLANIFICACIÓN DE LA ASIGNATURA

1. IDENTIFICACIÓN:

1.1. Nombre de Asignatura: Estructuras de Datos y Programación II.

1.2. Carrera: Licenciatura en Sistemas de Información.

1.3. Plan de Estudios: 2011 – Innovación Curricular 2022.

1.4. Año académico: 2024.

1.5. Carácter: Obligatoria.

1.6. Ubicación de la Asignatura en el Plan de Estudios

1.6.1. Módulo – Año: 4º Módulo – 2º Año.

1.6.2. Trayecto al que pertenece la Asignatura/Obligación Curricular

TRAYECTO	CARGA HORARIA PRESENCIAL
Ciencias Básicas y Específicas	-----
Algoritmos y Lenguajes	90 horas
Arquitectura, Sistemas Operativos y Redes	-----
Ingeniería de Software, Bases de Datos y Sistemas de Información	-----
Aspectos Sociales y Profesionales	-----
Otros contenidos	-----
CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR	**90 horas**

Tabla 1: Carga horaria por trayecto

1.6.3. Correlativas

1.6.3.1. Anteriores:

Débiles: Inglés I, Estructuras de Datos y Programación I, Organización del Computador.

1.6.3.2. Posteriores:

Débiles: Métodos Numéricos, Bases de Datos I, Sistemas Operativos, Teoría de la Computación, Programación Lógica y Funcional.

Fuerte: Simulación.

1.7. Carga horaria:

1.7.1. Carga horaria semanal total: 6 horas.

1.7.2. Carga horaria semanal destinada a la formación práctica: 4 horas.

1.7.3. Carga horaria total dedicada a las distintas actividades de formación práctica: 60 horas.

1.8. Ámbitos donde se desarrollan las actividades de formación práctica a las que se hace referencia en el punto anterior: aulas y laboratorio de computadoras.

1.9. Indique la cantidad de comisiones en las que se dicta la asignatura: 1 comisión.

2. PRESENTACIÓN

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

Esta asignatura corresponde al Trayecto Algoritmos y Lenguajes. En ella se estudian estructuras de datos no lineales, archivos de datos, estrategias de diseño de algoritmos y se aplica la Programación Orientada a Objetos (POO). En esta asignatura se abordan:

- Aspectos avanzados de la POO, tales como polimorfismo, genericidad y manejo de excepciones, aplicados en la resolución de problemas y el desarrollo de algoritmos.
- Estructuras de datos no lineales, tales como árboles y grafos.
- Organización de archivos y tablas de dispersión.
- Algoritmos de ordenamiento y búsqueda así como estrategias de diseño de algoritmos aplicados sobre las estructuras de datos estudiadas.

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

- Competencias desarrolladas en Estructuras de Datos y Programación I en relación al uso y aplicación de las estructuras de datos lineales, la recursividad y el Paradigma de Programación Orientado a Objetos.
- Conocimientos adquiridos respecto a la representación de los datos a nivel de máquina, los componentes básicos de la computadora, y el manejo de interrupciones, adquiridos en Organización del Computador.
- El vocabulario específico sobre Informática, formas de dar instrucciones, y formas de describir un problema adquiridos en la asignatura Inglés I.

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

Esta asignatura contribuye a que el egresado posea:

- Los conocimientos básicos (lógico-matemáticos y computacionales) para una adecuada fundamentación teórica de su quehacer profesional específico.

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

- Analizar y seleccionar las estructuras de datos, necesarias para los diferentes Sistemas de Información.
- Participar con otros especialistas de su disciplina, en la selección y configuración de Sistemas de Computación, según requerimientos de las distintas áreas de aplicación.

2.4.Integración horizontal y vertical con otras asignaturas.

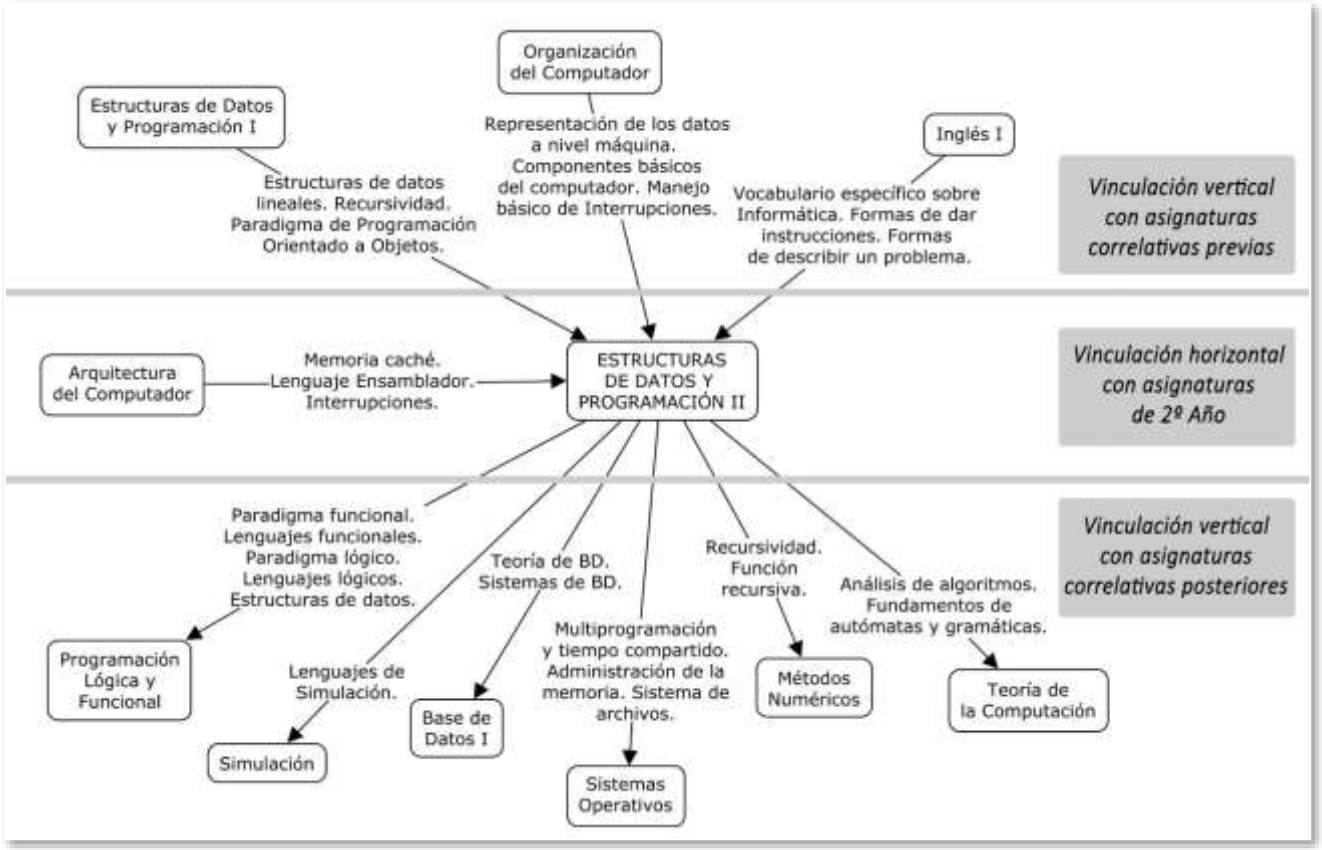

Figura 1. Integración horizontal y vertical de Estructuras de Datos y Programación II.

3. OBJETIVOS

3.1. Objetivos Generales

- Desarrollar creativamente soluciones algorítmicas seleccionando criteriosamente la alternativa adecuada, aplicando distintos paradigmas de programación.
- Emplear estructuras de control y estructuras de datos en la resolución de problemas, aplicando distintos paradigmas de programación.
- Verificar algoritmos desarrollados en los distintos paradigmas de programación.
- Aplicar distintos lenguajes de programación como herramientas computacionales para la resolución de problemas.

3.2.Objetivos Específicos

- Que el alumno desarrolle las siguientes competencias básicas:
    - Seleccionar y procesar la información necesaria para la resolución de problemas y verificar las soluciones encontradas.
- Que el alumno desarrolle las siguientes competencias específicas:
    - Analizar problemas y formular soluciones algorítmicas utilizando el Paradigma Orientado a Objetos, aplicando modularidad y reutilización de código, e implementando estructuras de datos avanzadas, tales como archivos y árboles.
    - Describir el funcionamiento de la resolución de problemas de computadora mediante la diagramación de flujo y el modelado de las resoluciones usando diagramación UML de clases para distinguir los componentes del sistema y las relaciones entre clases.
    - Utilizar herramientas modernas de desarrollo de software, tal como un entorno de desarrollo integrado (IDEs), que facilite la implementación y manipulación de estructuras de datos como archivos y árboles, para el procesamiento y manejo de volúmenes de datos.
- Que el alumno desarrolle las siguientes competencias transversales:
    - Colaborar en el desarrollo de proyectos de programación en equipo, asegurando la integración efectiva de archivos y árboles en la solución final y la correcta aplicación del Paradigma Orientado a Objetos.
    - Explicar y justificar soluciones técnicas relacionadas con estructuras de datos, utilizando un lenguaje claro y preciso que sea comprensible.
    - Utilizar las tecnologías y técnicas de programación de manera responsable, considerando los impactos sociales, ambientales y éticos de las soluciones desarrolladas.

4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

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

Estructuras de datos no lineales: árboles y grafos. Operaciones e implementación. Estrategias de diseño de algoritmos. Algoritmos de ordenamiento y búsqueda. Archivos de datos: operaciones básicas. Organización de archivos. Tablas de dispersión. Programación Orientada a Objetos. Polimorfismo y genericidad. Patrones. Manejo de excepciones.

4.2.Programa Sintético sobre la base de los contenidos mínimos

Unidad 1: Programación Orientada a Objetos

Programación Orientada a Objetos. Polimorfismo y genericidad. Patrones. Manejo de excepciones.

Unidad 2: Archivos.

Archivos de datos: operaciones básicas. Organización de archivos. Tablas de dispersión.

Unidad 3: Estructuras de datos no lineales

Estructuras de datos no lineales: árboles y grafos. Operaciones e implementación.

Unidad 4: Estrategias de diseño de algoritmos.

Estrategias de diseño de algoritmos. Algoritmos de ordenamiento y búsqueda.

4.3.Articulación Temática de la Asignatura.

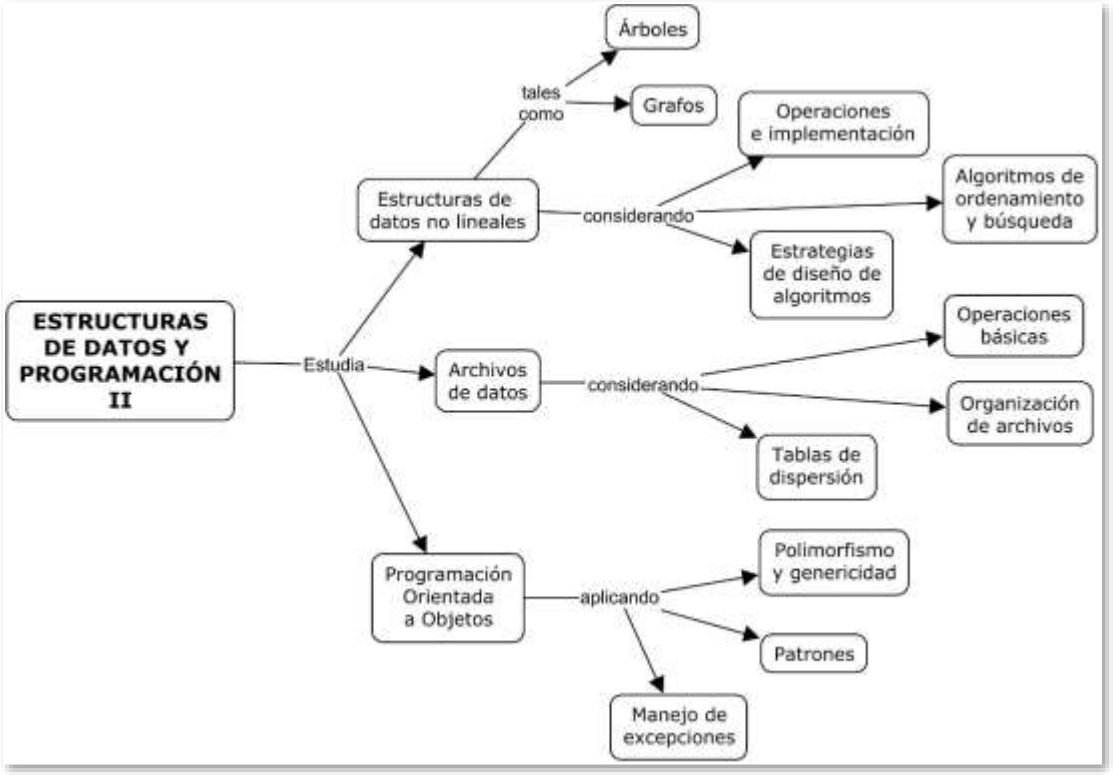

Figura 2. Articulación temática de la asignatura.

4.4.Programa Analítico

Unidad 1: Programación Orientada a Objetos

Programación Orientada a Objetos (POO). Clases y métodos abstractos. Polimorfismo y genericidad. Interfaces y clases genéricas. Manejo de excepciones. Generación y captura de excepciones. Patrones. Modelado y desarrollo de proyectos aplicando POO.

Unidad 2: Archivos.

Archivos. Tipos de archivos. Archivos de datos: operaciones básicas. Acceso secuencial. Acceso directo. Organización de archivos. Gestión de archivos (ABM). Tablas de dispersión. Función de dispersión, manejo de colisiones. Archivos hash. Implementaciones y ejemplos.

Unidad 3: Estructuras de datos no lineales

Estructuras de datos no lineales: árboles y grafos. Representación. Operaciones e implementación. Árboles generales y árboles binarios. Árboles binarios de búsqueda. Árboles AVL. Algoritmos fundamentales sobre árboles y grafos. Aplicaciones y resolución de problemas.

Unidad 4: Estrategias de diseño de algoritmos.

Estrategias de diseño de algoritmos. Algoritmos voraces. Árbol abarcador de costo mínimo. Algoritmos de Prim y Kruskal. Algoritmos divide y vencerás. Backtracking. Camino de costo mínimo. Algoritmo de Dijkstra. Algoritmos de ordenamiento y búsqueda.

4.5.Cronograma para el desarrollo de las Unidades Temáticas.

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Unidad 1: Programación Orientada a Objetos.	7	Semanas 2 a 6, 9, 10 y 15
Unidad 2: Archivos.	8	Semanas 1 a 6
Unidad 3: Estructuras de datos no lineales.	8	Semanas 7 a 10, 14 y 15
Unidad 4: Estrategias de diseño de algoritmos.	7	Semanas 11 a 14
TOTAL	**30**	

Tabla 2: Cronograma para el desarrollo teórico de las unidades temáticas

5. FORMACIÓN PRÁCTICA

5.1.Descripción de las actividades de formación práctica

Las actividades prácticas de la asignatura se llevan adelante mediante la Guía de Trabajos Prácticos que se organiza temáticamente y se proporciona al alumnado al comienzo del curso.

Mediante las actividades prácticas los alumnos aprenden sobre estructuras de datos no lineales y archivos, considerando algoritmos de aplicación que se expresan en el desarrollo de proyectos de programación en un lenguaje orientado a objetos.

Los alumnos resuelven problemas básicos y problemas similares a los del mundo real desarrollando soluciones en computadora mediante la aplicación del Paradigma de Programación Orientado a Objetos, utilizando un Entorno de Desarrollo Integrado (IDE).

Los Trabajos Prácticos de la asignatura son

- TP Procesamiento de Archivos,
- TP Árboles, y
- TP Algoritmos sobre Grafos

y consisten en el desarrollo de un conjunto de ejercicios aplicando…

- diagramación UML de clases,
- diagramación de flujo y
- el desarrollo de proyectos en lenguaje de programación Java aplicando los conceptos del Paradigma de Programación Orientado a Objetos –herencia entre clases, genericidad, entre otros– y la reutilización de código.

Las actividades prácticas que desarrollan los alumnos en el contexto de los Trabajos Prácticos de la asignatura son las siguientes:

- En relación a la Identificación, formulación y resolución de problemas de informática:

  A1. Expresar la resolución de problemas mediante el modelado usando diagramas UML de clases y diagramas de flujo estructurado.

- En relación a la Concepción, diseño y desarrollo de proyectos de informática:

  A2. Codificar algoritmos eficientes para la manipulación de estructuras de datos complejas, tales como archivos y árboles binarios de búsqueda.

- En relación a la Utilización de técnicas y herramientas de aplicación en la informática:

  A3. Usar un entorno de desarrollo integrado (IDEs), tal como NetBeans, para la codificación, depuración y prueba de programas que implementan estructuras de datos complejas

- En relación a la Generación de desarrollos tecnológicos y/o innovaciones tecnológicas:

  A4. Desarrollar proyectos de programación orientada a objetos para la gestión de grandes conjuntos de datos utilizando estructuras de datos avanzadas.

Durante el desarrollo de las clases de la asignatura, tanto en las clases prácticas como en las clases teóricas, se acompaña a los alumnos en el desarrollo de las actividades prácticas propuestas en los Trabajos Prácticos, aplicando dinámicas de grupo. En este sentido las actividades prácticas que desarrollan los alumnos son las siguientes:

- En relación a los Fundamentos para el desempeño en equipos de trabajo:

  A5. Colaborar con su equipo en el desarrollo de tareas de integración entre teoría y práctica considerando el Paradigma Orientado a Objetos y estructuras de datos avanzadas.

Mediante actividades de aula que se desarrollan en clases de teoría, y mediante actividades aplicativas que se presentan en el aula virtual de la asignatura, los alumnos fortalecen la vinculación entre los conceptos teóricos en estudio y su aplicación práctica. Se trata generalmente de actividades grupales que consisten en la elaboración de informes, resúmenes, mapas conceptuales, etc., la documentación del desarrollo de proyectos de programación y la exposición en clase de las tareas realizadas. En estas dinámicas, los alumnos desarrollarán la siguiente actividad:

- En relación a los Fundamentos para la comunicación efectiva:

  A6. Explicar de manera clara la lógica y el funcionamiento de estructuras de datos como árboles y grafos.

Así mismo, mediante actividades teóricas y áulicas, los estudiantes abordarán aspectos éticos de actualidad en relación al desarrollo de software y la Informática centrados en la temática de la asignatura. Los alumnos desarrollarán la siguiente actividad:

- En relación a los Fundamentos para la acción ética y responsable:

  A7. Analizar las decisiones tomadas en el diseño e implementación de estructuras de datos, evaluando su alineación con principios éticos.

De este modo, las actividades prácticas se llevan a cabo tanto en forma individual, procurando desarrollar habilidades de autogestión y superación frente al error, como también en modalidad grupal, fomentando la interacción con sus iguales, la cooperación y la discusión productiva con el objetivo de resolver el problema planteado.

Los problemas planteados en las prácticas de programación representan casos de la vida real en los cuales el cliente requiere interactuar con un programa en ejecución que le permita completar actividades comerciales, de administración de recursos o actividades, de gestión de información, entre otros. Estas prácticas proporcionan al alumno habilidades básicas para su futuro profesional en el desarrollo del software y la aplicación del Paradigma de Programación Orientado a Objetos.

En las actividades de formación práctica se pretende que el alumno desarrolle habilidades de eficiencia en la programación, observando organización y claridad en la codificación y documentación de los proyectos elaborados. Particularmente, se fomenta la prueba y verificación del código desarrollado.

5.2.Formación en Ejes Transversales

En la Tabla 3 se relacionan cada uno de los Ejes Transversales de Formación con las actividades y los resultados de aprendizaje esperados.

Los Resultados de Aprendizaje RA.1, RA.2, RA.3 y RA.4 se evalúan en las entregas de Trabajos Prácticos y en las evaluaciones Parcial Práctico 1, Parcial Práctico 2 y sus correspondientes Recuperatorios.

Tanto el RA.5 como el RA.7, se evalúan en las actividades de aula y en las actividades aplicativas que se proponen para integrar la teoría con la práctica. El Resultado de Aprendizaje RA.6 se evalúa en las Evaluaciones Prácticas Formativas, al igual que el RA.5.

Tabla 3: Formación en Ejes Transversales

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GPT (*)
Identificación, formulación y resolución de problemas de informática	A1. Expresar la resolución de problemas mediante el modelado usando diagramas UML de clases y diagramas de flujo estructurado.	RA.1. Desarrolla el diagrama UML de clases destacando entre ellas el tipo de relaciones, tales como relaciones de uso, de agregación o de composición según corresponda, y expresa el algoritmo solución en diagramas de flujo, para cada uno de los métodos de las clases identificadas.	Alto
Concepción, diseño y desarrollo de proyectos de informática	A2. Codificar algoritmos eficientes para la manipulación de estructuras de datos complejas, tales como archivos y árboles binarios de búsqueda.	RA.2. Implementa correctamente algoritmos para la manipulación de archivos y árboles, demostrando competencia en la resolución de problemas de programación usando el Paradigma Orientado a Objetos.	Bajo
Utilización de técnicas y herramientas de aplicación en la informática	A3. Usar un entorno de desarrollo integrado (IDEs), tal como NetBeans, para la codificación, depuración y prueba de programas que implementan estructuras de datos complejas	RA.3. Utiliza un entorno de desarrollo integrado (IDEs), como NetBeans, para implementar, depurar y gestionar proyectos de programación orientada a objetos que involucren estructuras de datos complejas como archivos y árboles.	Medio
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	A4. Desarrollar proyectos de programación orientada a objetos para la gestión de grandes conjuntos de datos utilizando estructuras de datos avanzadas.	RA.4. Aplica su conocimiento de estructuras de datos en el desarrollo de proyectos de programación orientada a objetos, demostrando capacidad para integrar la manipulación de archivos y árboles en soluciones prácticas.	Medio
Fundamentos para el desempeño en equipos de trabajo	A5. Colaborar con su equipo en el desarrollo de tareas de integración entre teoría y práctica considerando el Paradigma Orientado a Objetos y estructuras de datos avanzadas.	RA.5. Colabora eficazmente con otros miembros del equipo en la elaboración de trabajos grupales, contribuyendo activamente en la resolución del planteamiento.	Medio
Fundamentos para la comunicación efectiva	A6. Explicar de manera clara la lógica y el funcionamiento de estructuras de datos como árboles y grafos.	RA.6. Explica y justifica de manera clara y precisa las soluciones técnicas relacionadas con archivos, árboles y grafos, demostrando comprensión profunda y capacidad de comunicación efectiva.	Bajo
Fundamentos para la acción ética y responsable	A7. Analizar las decisiones tomadas en el diseño e implementación de estructuras de datos, evaluando su alineación con principios éticos.	RA.7. Reflexiona sobre su práctica de programación, considerando los impactos éticos de las soluciones desarrolladas con estructuras de datos como archivos, árboles y grafos.	Bajo

(*) GPT: Grado de Profundidad en el Tratamiento.

5.3.Cronograma de formación práctica

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
TP Procesamiento de Archivos	28	Semanas 1 a 7
TP Árboles	16	Semanas 8 a 10, 14, 15
TP Algoritmos sobre Grafos	16	Semanas 11 a 15
TOTAL	**60**	

Tabla 4: Cronograma para el desarrollo de las Actividades Prácticas

6. BIBLIOGRAFÍA.

TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Programación orientada a objetos en Java	Blasco F.	Ediciones de la U	Biblioteca Digital eLibro.net	2019
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/222723/		
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 5: Bibliografía

Tabla 5: 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/		
Java: curso práctico	Vegas Gertrudix, J.	RA-MA Editorial	Biblioteca Digital eLibro.net	2020
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/222715/		
Estructura de datos en JAVA	Weiss Mark Allen.	Pearson	2 en Centro de Documentación	2013

7. ESTRATEGIAS METODOLÓGICAS.

7.1.Aspectos pedagógicos y didácticos

La metodología de enseñanza que se aplicará en el aula será la de 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 el aula virtual de la asignatura alojada en 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 en las clases 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 presentar o enviar mediante la plataforma del CUV.

El estudio de las estructuras de datos en cada uno de los trabajos prácticos propuestos en esta asignatura, incluye el desarrollo de la resolución de problemas mediante diagramación UML

de clases, diagramación de flujo y desarrollo de proyectos en lenguaje Java aplicando el Paradigma de Programación Orientado a Objetos.

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 resolución algorítmica en el alumno. En la práctica se plantean a los alumnos diferentes problemas en los que se solicita la programación completa en lenguaje Java de la resolución del enunciado a fin de fortalecer las habilidades de programación en el alumno aplicando la orientación a objetos.

Se utiliza software libre en la tarea de desarrollar habilidades relacionadas con la programación en un lenguaje orientado a objetos. Dichas habilidades son las necesarias para la resolución de problemas y el diseño de algoritmos y para preparar a los alumnos a enfrentar los distintos cambios del mundo tecnológico.

En el desarrollo de las actividades de la asignatura se promueve por parte del alumno la integración participativa en donde aporte al grupo de estudio tanto sus habilidades como sus conocimientos.

Mediante actividades de aula y actividades aplicativas se propone al alumno la vinculación entre teoría y práctica. En estas actividades se fomenta el desarrollo de habilidades para la comunicación efectiva mediante la expresión oral y/o escrita de los conceptos teóricos relacionándolos con la práctica, así como el seguimiento documentado de sus procesos mentales en la resolución de problemas fundamentándolos teóricamente. En dichas actividades los estudiantes deberán, entre otras tareas, presentar en grupo o individualmente la descripción de casos e informes, exponer en clase la actividad realizada y documentar el desarrollo de proyectos de programación.

Las competencias relacionadas con la acción ética y responsable serán desarrolladas mediante actividades teóricas y áulicas en las cuales los estudiantes deberán, entre otras tareas, analizar problemas éticos de actualidad relacionados con el desarrollo de software y la Informática y reflexionar sobre los propios mecanismos éticos personales para privilegiar los valores de la ética.

7.2.Mecanismos para la integración de docentes

Se realizarán actividades de revisión y coordinación en el Trayecto Algoritmos y Lenguajes en el marco de la Comisión de Seguimiento del Plan de Estudios de la carrera de Licenciatura en Sistemas de Información. Además, se efectuarán reuniones periódicas con las asignaturas del Trayecto con el fin de aunar tareas conjuntas de integración.

La coordinación entre las asignaturas 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.

7.3.Recursos Didácticos

Se utilizarán diversos recursos didácticos a fin de lograr 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 Kid (JDK). JDK incluye Java Runtime Environment, el compilador Java y las API de Java.
- Bibliografía actualizada impresa y digital (Biblioteca eLibro) y apuntes elaborados por el equipo docente de la asignatura, disponibles en la plataforma del CUV y en el Centro de Documentación del Departamento de Informática.
- En las clases teóricas y en las clases prácticas se utilizarán 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 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.

8. EVALUACIÓN

8.1.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 de datos lineales, recursividad y Paradigma de Programación Orientado a Objetos. Se utilizará la herramienta cuestionario de la plataforma Moodle del CUV.

La evaluación diagnóstica será especialmente diseñada, individual, 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.

8.2.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 y el proceso de aprendizaje que permita comprender cómo se producen dichos procesos 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.

8.2.1. Programa y Cronograma de Evaluaciones Formativas.

Evaluación formativa de los contenidos teóricos

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.

Los cuestionarios se llevarán a cabo mediante la plataforma del CUV o en su defecto mediante prueba escrita y se realizarán en horarios de clase de teoría.

		MESES	
ACTIVIDAD	**TEMA**	**Septiembre**	**Noviembre**
Cuestionario 1	Archivos. Tablas de dispersión. Excepciones. Interfaces. Clases abstractas.	1º Semana	-----
Cuestionario 2	Árboles binarios de búsqueda. Árboles AVL. Grafos. Estrategias de diseño de algoritmos. Polimorfismo. Genericidad.	-----	1ª Semana

Tabla 6: Cronograma de evaluaciones teóricas formativas

Evaluación formativa de la Práctica

La evaluación formativa de la práctica se llevará adelante aplicando la técnica de casos y se estructurará mediante el desarrollo de proyectos de programación con entregas reguladas según se detalla a continuación:

- Diseño orientado a objetos (DOO), utilizando diagramación UML de clases y diagramación de flujo estructurado.
- Desarrollo, verificación y ejecución aplicando el lenguaje de programación Java y la POO.

En la Tabla 7 se detalla el Cronograma de evaluaciones prácticas formativas con sus fechas de entrega y modalidad de desarrollo. Cada una de las actividades propuestas en el cronograma se desarrollará mediante práctica asistida. En clase de práctica y en horarios de consulta se promoverá el intercambio con los estudiantes para motivarlos, asesorarlos, aclarar sus dudas, procurando potenciar su aprendizaje autónomo y su integración con su grupo de trabajo.

La escala de valoración que se aplicará a las entregas de las actividades de evaluaciones prácticas formativas será cualitativa dicotómica (aprobado – desaprobado).

Para el fortalecimiento de la aplicación práctica de los conceptos aprendidos los estudiantes desarrollarán una Carpeta de Prácticas para lo que contarán con la Guía de Trabajos Prácticos con ejercicios que implican:

- la resolución de problemas usando las estructuras de datos en estudio, aplicando el Paradigma Orientado a Objetos, la Diagramación de flujo estructurada y la Diagramación UML de clases,
- el desarrollo de proyectos de programación que hacen uso de las estructuras de datos en estudio, aplicando el Paradigma Orientado a Objetos, la Diagramación de flujo estructurada, la Diagramación UML de clases y el Lenguaje de programación Java.

ACTIVIDAD	TEMA	MODALIDAD	SEM	ENTREGA	DEVOLUCIÓN
Diseño del proyecto	Archivos. Flujos en Java. Excepciones. Interfaces. Herencia.	Domiciliario. Grupal. Asistido. Escrito. Aplicar: • Diag. UML de clases. • Diagramación de flujo.	4	06/09	09/09
Codificación y ejecución del proyecto		Domiciliario. Grupal. Asistido. Desarrollo de proyecto de POO aplicando lenguaje Java	6	18/09	18/09
Diseño del proyecto	Árboles binarios de búsqueda. Clases abstractas. Polimorfismo. Genericidad.	Domiciliario. Grupal. Asistido. Escrito. Aplicar: • Diag. UML de clases. • Diagramación de flujo.	9	09/10	14/10
Codificación y ejecución del proyecto		Domiciliario. Grupal. Asistido. Desarrollo de proyecto de POO aplicando lenguaje Java	11	23/10	23/10

Tabla 7: Cronograma de evaluaciones prácticas formativas

En la Tabla 8 se detallan objetivos, y en la Tabla 9 se indica modalidad y las consignas de resolución de los trabajos prácticos.

Tabla 8: Programa de Trabajos Prácticos. Objetivos.

TRABAJOS PRÁCTICOS	OBJETIVOS
TP Procesamiento de Archivos	• Resolver problemas utilizando Archivos directos aplicando la orientación a objetos y el lenguaje Java. • Desarrollar proyectos en lenguaje Java utilizando flujos de E/S, excepciones, interfaces y herencia en la resolución de problemas usando Archivos.
TP Árboles	• Resolver problemas utilizando la estructura de datos Árbol aplicando la orientación a objetos y el lenguaje Java. • Desarrollar proyectos en lenguaje Java utilizando clases abstractas, clases genéricas y polimorfismo en la resolución de problemas usando el TAD Árbol.
TP Algoritmos sobre Grafos	• Reconocer las principales características de la estructura de datos Grafo mediante el análisis de sus operaciones en la resolución de problemas usando la orientación a objetos. • Aplicar algoritmos de Grafos para modelar la resolución de problemas mediante el desarrollo de proyectos en lenguaje Java.

ASPECTOS COMUNES A TODOS LOS TRABAJOS PRÁCTICOS	
OBJETIVOS GENERALES	Que el estudiante adquiera las competencias para: • Usar las estructuras de datos estudiadas con sus operaciones básicas en la resolución de problemas. • Desarrollar aplicaciones de software mediante la utilización del paradigma de POO.
MODALIDAD	• Resolución individual de problemas mediante la diagramación UML de clases, y la diagramación de flujo, aplicando el Paradigma Orientado a Objetos. • Desarrollo, individual y grupal, de proyectos de programación que hacen uso de las estructuras de datos en estudio
CONSIGNAS DE RESOLUCIÓN	• El trabajo práctico debe estar desarrollado completamente. • La identificación de clases, sus propiedades y comportamiento, así como sus relaciones debe realizarse aplicando el DOO. • La diagramación de flujo, así como la diagramación UML de clases, deben haber sido desarrolladas por el alumno siguiendo las pautas fijadas en los Recursos Bibliográficos. • La codificación en lenguaje Java debe realizarse siguiendo los principios de la POO. • La presentación del código de los enunciados solicitados deberá realizarse en tiempo y forma mediante la plataforma CUV.FCEyT.

Tabla 9: Programa de Trabajos Prácticos. Aspectos comunes a todos los trabajos prácticos.

8.3.Evaluación Parcial

8.3.1. Programa de Evaluaciones Parciales

EVALUACIÓN	TEMA	MODALIDAD	SEM	FECHA	DEVOLUCIÓN DE RESULTADO
Parcial Práctico 1	Archivos. Flujos en Java. Excepciones. Interfaces. Herencia.	Individual. Escrito. En computadora. Resolución de un problema propuesto mediante: • Diagramación UML de clases aplicando Orientación a Objetos. • Diagramación de flujo. • Reutilización de proyecto de POO en computadora aplicando lenguaje Java.	7	25/09	02/10
Recuperatorio 1			9	20/11	25/11
Parcial Práctico 2	Árboles binarios de búsqueda. Polimorfismo. Genericidad.		14	13/11	20/11
Recuperatorio 2			15	20/11	25/11

Tabla 10: Programa de Evaluaciones Parciales

8.3.2. Criterios de Evaluación

Los Criterios de Evaluación generales que serán aplicados en el Parcial Práctico 1 y en el Parcial Práctico 2, así como en sus correspondientes Recuperatorios, son los siguientes:

- En relación al Resultado de Aprendizaje RA.1:

  CE.1.1. El estudiante identifica y representa correctamente las clases y sus relaciones en el diagrama UML, reflejando de manera precisa la estructura y el comportamiento del sistema.

  CE.1.2. El estudiante representa correctamente la lógica algorítmica mediante la diagramación de flujo estructurado, asegurando que todas las operaciones, decisiones y ciclos estén adecuadamente reflejados para todos los métodos de las clases identificadas en el problema a resolver.

- En relación al Resultado de Aprendizaje RA.3:

  CE.3. El estudiante utiliza el Entorno de Desarrollo Integrado (IDE) NetBeans de manera eficiente para la gestión, depuración y desarrollo del proyecto.

Los Criterios de Evaluación específicos al Parcial Práctico 1 y su correspondiente Recuperatorio, son los siguientes:

- En relación al Resultado de Aprendizaje RA.2:

  CE.2.1. Demuestra comprensión en la aplicación de flujos de entrada/salida, interfaces y relación de herencia en la programación orientada a objetos (POO) aplicándolos en el desarrollo del proyecto solución para el problema planteado.

- En relación al Resultado de Aprendizaje RA.4:

  CE.4.1. Implementa correctamente un proyecto de programación orientada a objetos (POO) en lenguaje Java, utilizando archivos directos para resolver el problema planteado.

Los Criterios de Evaluación específicos al Parcial Práctico 2 y su correspondiente Recuperatorio, son los siguientes:

- En relación al Resultado de Aprendizaje RA.2:

  CE.2.2. Demuestra comprensión en la aplicación de clases abstractas, polimorfismo y genericidad en la programación orientada a objetos (POO) aplicándolos en el desarrollo del proyecto solución para el problema planteado.

- En relación al Resultado de Aprendizaje RA.4:

  CE.4.2. Implementa correctamente un proyecto de programación orientada a objetos (POO) en lenguaje Java, utilizando árboles binarios de búsqueda para resolver el problema planteado.

Lo descripto se resume en la siguiente Tabla.

Tabla 9: Aplicación de los Criterios de Evaluación

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN	EVALUACIÓN EN LA QUE SE APLICA
RA.1. Desarrolla el diagrama UML de clases destacando entre ellas el tipo de relaciones, tales como relaciones de uso, de agregación o de composición según corresponda, y expresa el algoritmo solución en diagramas de flujo, para cada uno de los métodos de las clases identificadas.	CE.1.1. El estudiante identifica y representa correctamente las clases y sus relaciones en el diagrama UML, reflejando de manera precisa la estructura y el comportamiento del sistema.	Parcial Práctico 1 Recuperatorio 1 Parcial Práctico 2 Recuperatorio 2
	CE.1.2. El estudiante representa correctamente la lógica algorítmica mediante la diagramación de flujo estructurado, asegurando que todas las operaciones, decisiones y ciclos estén adecuadamente reflejados para todos los métodos de las clases identificadas en el problema a resolver.	
RA.2. Implementa correctamente algoritmos para la manipulación de archivos y árboles, demostrando competencia en la resolución de problemas de programación usando el Paradigma Orientado a Objetos.	CE.2.1. Demuestra comprensión en la aplicación de flujos de entrada/salida, interfaces y relación de herencia en la programación orientada a objetos (POO) aplicándolos en el desarrollo del proyecto solución para el problema planteado.	Parcial Práctico 1 Recuperatorio 1
	CE.2.2. Demuestra comprensión en la aplicación de clases abstractas, polimorfismo y genericidad en la programación orientada a objetos (POO) aplicándolos en el desarrollo del proyecto solución para el problema planteado.	Parcial Práctico 2 Recuperatorio 2

Tabla 9: Aplicación de los Criterios de Evaluación

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN	EVALUACIÓN EN LA QUE SE APLICA
RA.3. Utiliza un entorno de desarrollo integrado (IDEs), como NetBeans, para implementar, depurar y gestionar proyectos de programación orientada a objetos que involucren estructuras de datos complejas como archivos y árboles.	CE.3. El estudiante utiliza el Entorno de Desarrollo Integrado (IDE) NetBeans de manera eficiente para la gestión, depuración y desarrollo del proyecto.	Parcial Práctico 1 Recuperatorio 1 Parcial Práctico 2 Recuperatorio 2
RA.4. Aplica su conocimiento de estructuras de datos en el desarrollo de proyectos de programación orientada a objetos, demostrando capacidad para integrar la manipulación de archivos y árboles en soluciones prácticas.	CE.4.1. Implementa correctamente un proyecto de programación orientada a objetos (POO) en lenguaje Java, utilizando archivos directos para resolver el problema planteado.	Parcial Práctico 1 Recuperatorio 1
	CE.4.2. Implementa correctamente un proyecto de programación orientada a objetos (POO) en lenguaje Java, utilizando árboles binarios de búsqueda para resolver el problema planteado.	Parcial Práctico 2 Recuperatorio 2
RA.5. Colabora eficazmente con otros miembros del equipo en la elaboración de trabajos grupales, contribuyendo activamente en la resolución del planteamiento.	CE.5. Cumple las tareas asignadas dentro del equipo, entregando trabajos de calidad en los plazos establecidos.	Actividades de aula Actividades aplicativas Evaluaciones Prácticas Formativas
RA.6. Explica y justifica de manera clara y precisa las soluciones técnicas relacionadas con archivos, árboles y grafos, demostrando comprensión profunda y capacidad de comunicación efectiva.	CE.6. Explica los conceptos técnicos relacionados con archivos, árboles y grafos de manera clara, utilizando un lenguaje preciso y apropiado para el contexto.	Evaluaciones Prácticas Formativas
RA.7. Reflexiona sobre su práctica de programación, considerando los impactos éticos de las soluciones desarrolladas con estructuras de datos como archivos, árboles y grafos.	CE. 7. El estudiante identifica correctamente los dilemas éticos que pueden surgir durante el desarrollo de soluciones que involucran archivos, árboles y grafos.	Actividades de aula Actividades aplicativas

8.3.3. Escala de Valoración

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

8.4.Evaluación Integradora

No se prevé.

8.5.Evaluación Sumativa

8.5.1. Condiciones para lograr la promoción sin Examen Final de la Asignatura. (Rige la Resolución HCD N° 135/00)

Para promocionar la asignatura el alumno deberá:

- Asistir como mínimo al 80% del total de sesiones.
- Aprobar los dos (2) Cuestionarios Evaluativos con un puntaje >= 7 puntos.
- Presentar y aprobar en su primera instancia todas las Actividades evaluativas prácticas formativas.
- Aprobar los dos (2) Parciales Prácticos previstos.
- Completar la presentación de la Carpeta de Trabajos Prácticos.

8.5.2. Condiciones para lograr la Regularidad de la Asignatura.

Para regularizar la asignatura el alumno deberá:

- Presentar las Actividades evaluativas prácticas formativas.
- Aprobar los dos (2) Parciales Prácticos previstos, o sus correspondientes Recuperatorios.

8.6.Examen Final

En el examen final los alumnos serán evaluados sobre los contenidos teóricos previstos en el programa de la asignatura. El examen será individual y podrá ser oral o escrito.

8.7.Examen Libre

Para el examen Libre el alumno deberá aprobar las dos (2) instancias que se detallan a continuación, siendo cada una de ellas eliminatorias:

Etapa 1.

a) 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.

b) Aprobar una evaluación escrita de tipo práctica, con diagramación UML de clases y diagramación de flujo, aplicando Orientación a Objetos.

c) Aprobar una evaluación de tipo practica en computador, utilizando el lenguaje Java y Programación Orientada a Objetos.

Etapa 2.

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

Méndez Analía

.....

Profesor responsable de Asignatura