

**UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y
TECNOLOGÍAS**

PLANIFICACIÓN ANUAL 2024

ASIGNATURA: ESTRUCTURAS DE DATOS Y PROGRAMACIÓN I

**LICENCIATURA EN SISTEMAS DE
INFORMACIÓN**

Plan de Estudio: 2011

Equipo cátedra:

Profesor Adjunto: Méndez Analía

JTP: Zarco Raquel

Ayudante de 1º: Gómez Claudia

Ayudantes Estudiantiles: 1



PLANIFICACIÓN DE LA ASIGNATURA

1. IDENTIFICACIÓN:

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

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: 3° Módulo – 2° Año.

1.6.2. Trayecto al que pertenece la Asignatura/Obligación Curricular

TRAYECTO	CARGA HORARIA PRESENCIAL
Ciencias Básicas y Específicas	-----
Algoritmos y Lenguajes	75 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	75 horas

Tabla 1: Carga horaria por trayecto

1.6.3. Correlativas

1.6.3.1. Anteriores:

Débiles: Álgebra I, Fundamentos de la Programación.

1.6.3.2. Posteriores:

Débiles: Estructuras de Datos y Programación II.

Fuerte: Programación Lógica y Funcional.

1.7.Carga horaria:

1.7.1. Carga horaria semanal total: 5 horas.

1.7.2. Carga horaria semanal destinada a la formación práctica: 3 horas.



1.7.3. Carga horaria total dedicada a las distintas actividades de formación práctica: 45 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. Está orientada fundamentalmente al estudio, análisis y manejo de estructuras de datos encadenadas y a la aplicación de recursividad en el desarrollo de algoritmos, e introduce el estudio del Paradigma de Programación Orientada a Objetos. En ella se abarcan:

- Conceptos básicos del Paradigma de Programación Orientada a Objetos (POO) aplicados en la resolución de problemas y el desarrollo de algoritmos, tales como abstracción, encapsulamiento e introducción a la herencia entre clases.
- Estructuras de datos encadenadas.
- Recursividad.
- Representación de datos en memoria y manejo de memoria en ejecución.
- Algoritmos fundamentales de recorrido, búsqueda, ordenamiento, y actualización aplicados sobre listas, pilas y colas.

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

- Conocimientos matemáticos adquiridos en Álgebra I.
- Paradigmas y lenguajes, análisis y diseño de algoritmos, estructuras de datos elementales, tipos de datos abstractos, verificación de algoritmos, aplicando el Paradigma Imperativo y lenguajes de programación, adquiridos en Fundamentos de la Programación.

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.

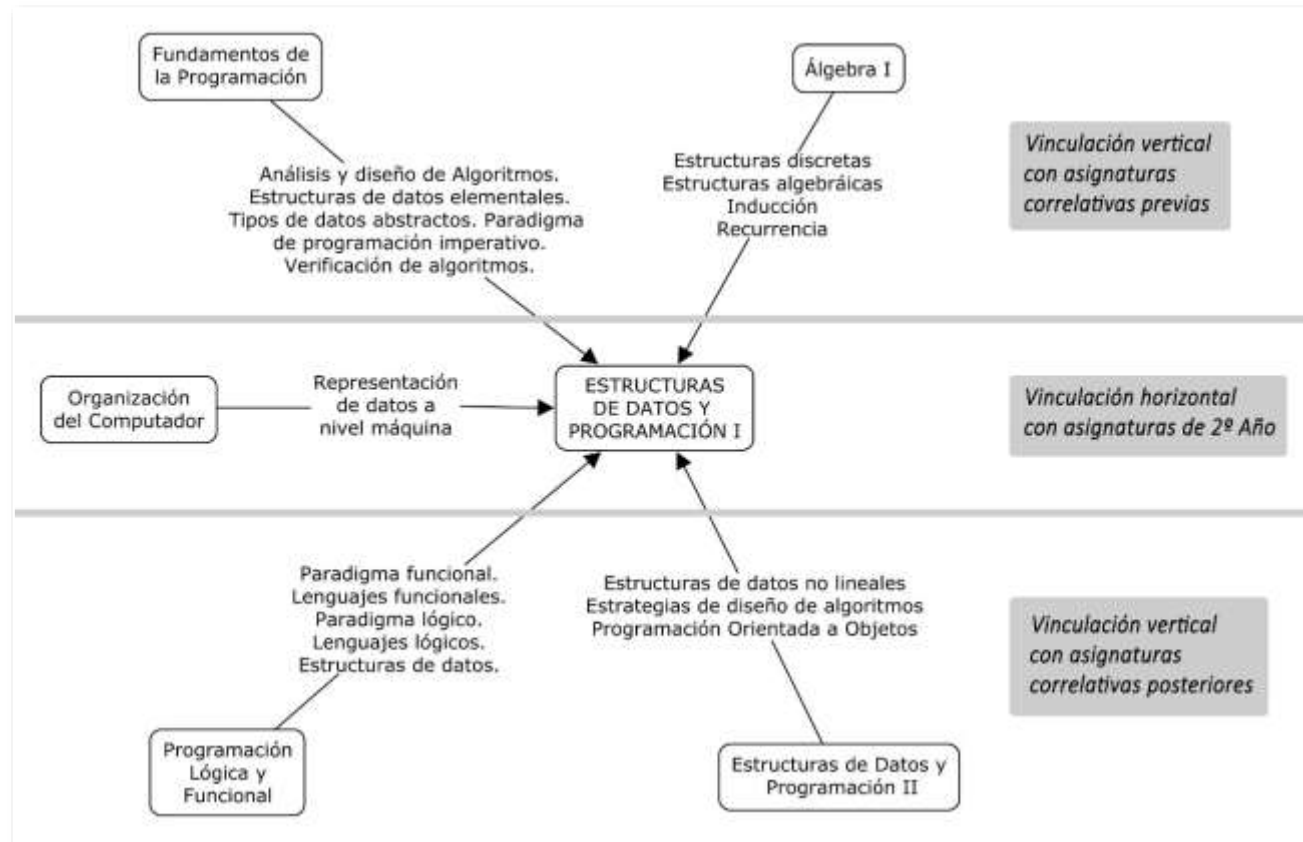


Figura 1. Integración horizontal y vertical de Estructuras de Datos y Programación.



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, e implementando estructuras de datos lineales, tales como listas, pilas y colas.
 - 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 sus relaciones de uso, de composición y de herencia.
 - 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, empleando la orientación a objetos y aplicando sus conceptos básicos, tales como abstracción, encapsulamiento y herencia entre clases.
- Que el alumno desarrolle las siguientes competencias transversales:
 - Colaborar en el desarrollo de proyectos de programación en equipo, utilizando en la solución final estructuras de datos, como listas, pilas o colas y aplicando el Paradigma Orientado a Objetos.

4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

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

Estructuras de datos lineales: listas, pilas y colas. Representación de datos en memoria. Operaciones e implementación. Estrategias de implementación. Algoritmos fundamentales de recorrido, búsqueda, inserción y eliminación aplicados sobre listas. Resolución de problemas y algoritmos. Recursividad. Tipos de datos recursivos. Algoritmos recursivos. Manejo de memoria en ejecución. Paradigma de Programación Orientado a Objetos: objeto, clases, encapsulamiento, herencia. Lenguajes de programación orientados a objetos.

4.2. Programa Sintético sobre la base de los contenidos mínimos

Unidad 1: Paradigma de Programación Orientado a Objetos

Paradigma de Programación Orientada a Objetos. Objeto, clases, encapsulamiento, herencia. Lenguajes de programación orientados a objetos.

Unidad 2: Estructuras de datos lineales.

Representación de datos en memoria. Estructuras de datos lineales: listas, pilas y colas. Operaciones e implementación. Estrategias de implementación. Algoritmos fundamentales de recorrido, búsqueda, inserción y eliminación aplicados sobre listas. Resolución de problemas y algoritmos.

Unidad 3: Recursividad.

Recursividad. Tipos de datos recursivos. Algoritmos recursivos. Manejo de memoria en ejecución.

4.3. Articulación Temática de la Asignatura.

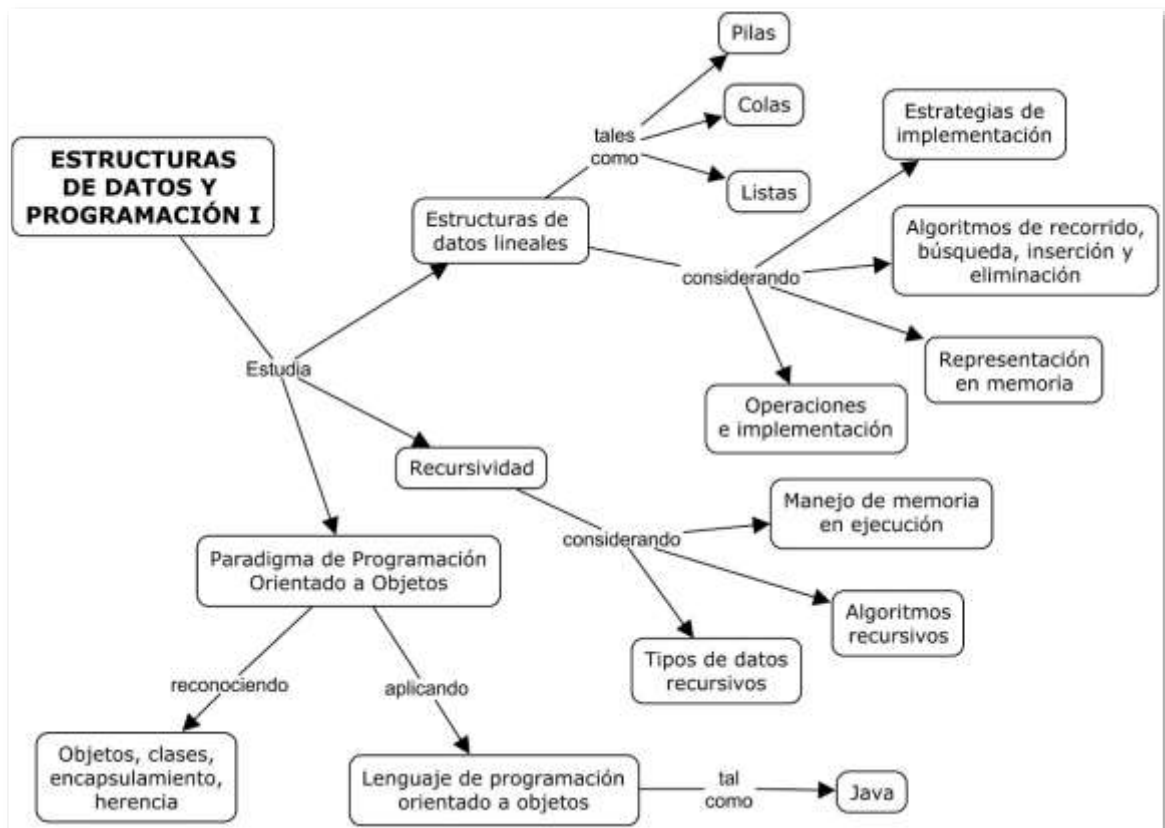


Figura 2. Articulación temática de la asignatura.



4.4. Programa Analítico

Unidad 1: Paradigma de Programación Orientado a Objetos

Paradigma de Programación Orientada a Objetos. Objeto, clases, encapsulamiento, herencia. Tipos de herencia y jerarquía de clases. Generalización y especialización. Lenguajes de Programación Orientados a Objetos. Lenguaje Unificado de Modelado (UML). Vista estática. Diagramación UML de clases. Clasificadores y relaciones. Modelado basado en clases.

Unidad 2: Estructuras de datos lineales.

Estructuras de datos. Representación de datos en memoria. Estructuras de datos lineales: listas, pilas y colas. Operaciones e implementación dinámica. Estrategias de implementación. Algoritmos fundamentales de recorrido, búsqueda, inserción y eliminación sobre listas. Resolución de problemas y algoritmos aplicando listas, pilas y colas.

Unidad 3: Recursividad.

Recursividad. Concepto. Tipos de datos recursivos. Algoritmos recursivos. Características. Tipos de recursividad. Manejo de memoria en ejecución. Análisis de algoritmos recursivos. Ejemplos de casos.

4.5. Cronograma para el desarrollo de las Unidades Temáticas.

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Unidad 1: Paradigma de Programación Orientado a Objetos.	16	Semanas 1 a 15
Unidad 2: Estructuras de datos lineales.	7	Semanas 2 a 8
Unidad 3: Recursividad.	7	Semanas 9 a 15
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 tales como listas, pilas, y colas, sus operaciones básicas y los algoritmos fundamentales de recorrido, búsqueda, ordenamiento y actualización aplicados sobre esas estructuras de datos estudiadas.



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 los siguientes:

- TP Paradigma de POO y diagramación UML de clases,
- TP LSE y LSEO,
- TP Pilas y Colas dinámicas,
- TP Introducción a la recursividad, y
- TP Desarrollo de algoritmos recursivos

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 básicos del Paradigma de Programación Orientado a Objetos, abstracción, encapsulamiento y herencia entre clases.

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 lineales, tales como listas, pilas y colas.
- 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 lineales.
- En relación a la Generación de desarrollos tecnológicos y/o innovaciones tecnológicas:
A4. Desarrollar proyectos de programación orientada a objetos para la gestión de datos utilizando estructuras de datos lineales.

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 diseño e implementación de soluciones usando el Paradigma Orientado a Objetos y aplicando estructuras de datos lineales.

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 y la cooperación a fin de resolver problemas dados.

Los problemas planteados en las prácticas de programación representan casos sencillos 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 en el desarrollo del software y la aplicación del Paradigma de Programación Orientado a Objetos.

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.

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.	Medio
Concepción, diseño y desarrollo de proyectos de informática	A2. Codificar algoritmos eficientes para la manipulación de estructuras de datos lineales, tales como listas, pilas y colas.	RA.2. Implementa correctamente algoritmos para la manipulación de listas, pilas y colas, 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 lineales.	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 lineales como listas, pilas y colas.	Bajo
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	A4. Desarrollar proyectos de programación orientada a objetos para la gestión de datos utilizando estructuras de datos lineales.	RA.4. Aplica su conocimiento de estructuras de datos en el desarrollo de proyectos de programación orientada a objetos, demostrando capacidad para integrar listas, pilas y colas en soluciones prácticas.	Bajo



Tabla 3: Formación en Ejes Transversales

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GPT (*)
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 las estructuras de datos lineales.	RA.5. Colabora eficazmente con otros miembros del equipo en la elaboración de trabajos grupales, contribuyendo activamente en la resolución del planteamiento.	Bajo

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

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.

El Resultado de Aprendizaje RA.5 se evalúa principalmente durante el desarrollo de actividades de aula y actividades aplicativas que se proponen al alumno a fin de integrar la teoría y la práctica.

5.3. Cronograma de formación práctica

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
TP Paradigma de POO y diagramación UML de clases	6	Semanas 1 y 2
TP LSE y LSEO	9	Semanas 3, 4 y 5
TP Pilas y Colas dinámicas	9	Semanas 6, 7 y 8
TP Introducción a la recursividad	6	Semanas 9 y 10
TP Desarrollo de algoritmos recursivos	15	Semanas 11 a 15
TOTAL	45	

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/		

Tabla 5: Bibliografía



Tabla 5: Bibliografía (continuación)

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
Introducción a la programación, lógica y diseño (7º ed.)	Farrel, J.	Cengage Learning	Biblioteca Digital eLibro.net	2013
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/93265/		
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 Aplicaciones en Java y C++	Jiménez de Parga C..	RA-MA Editorial	Biblioteca Digital eLibro.net	2014
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/106466/		
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/		
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/		
Empezar a programar usando Java (3º ed.)	Prieto Sáez, N & Casanova Faus, A.	Editorial de la Universidad Politécnica de Valencia	Biblioteca Digital eLibro.net	2016
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/57434/		
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/		
Diseñar y programar, todo es empezar: una introducción a	Vélez Serrano, J.	Dykinson	Biblioteca Digital eLibro.net	2011



Tabla 5: Bibliografía (continuación)

TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
la Programación Orientada a Objetos usando UML y Java	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/63076/		
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 de carácter grupal se propone al alumno la vinculación entre teoría y práctica. En estas actividades se fomenta el desarrollo de habilidades para el desempeño en equipos de trabajo, promoviendo 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.

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 Kit (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: Algoritmos, tipos de datos, arreglos, pilas y colas estáticas. 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.

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.

8.2.1. Programa y Cronograma de Evaluaciones Formativas.

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.



ACTIVIDAD	TEMA	MESES	
		Mayo	Junio
Cuestionario 1	Listas, pilas y colas	1° Semana	-----
Cuestionario 2	Recursividad	-----	3ª Semana

Tabla 6: Cronograma de evaluaciones teóricas formativas

8.3. Evaluación Parcial

8.3.1. Programa de Evaluaciones Parciales

EVALUACIÓN	TEMA	MODALIDAD	SEM	FECHA	DEVOLUCIÓN DE RESULTADO
Parcial Práctico 1	LSE, LSEO, Pilas y Colas	Individual. Escrito. En computadora. Resolución de un problema propuesto mediante: • Diagramación UML de clases aplicando Orientación a Objetos.	8	15/05	22/05
Recuperatorio 1			10	29/05	05/06
Parcial Práctico 2	Recursividad	• Diagramación de flujo. • Desarrollo de proyecto de POO en computadora aplicando lenguaje Java.	14	26/06	28/06
Recuperatorio 2			15	03/07	05/07

Tabla 8: 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. El estudiante demuestra comprensión de los conceptos de abstracción y encapsulamiento 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. El estudiante implementa correctamente un proyecto de programación orientada a objetos (POO) en lenguaje Java, utilizando listas simples encadenadas ordenadas, pilas y colas 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. El estudiante demuestra comprensión del concepto de herencia entre clases en la programación orientada a objetos (POO) aplicándolo en el desarrollo del proyecto solución para el problema planteado.

- En relación al Resultado de Aprendizaje RA.4:

CE.4.2. El estudiante implementa correctamente un proyecto de programación orientada a objetos (POO) en lenguaje Java, utilizando recursividad y aplicando herencia entre clases 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
	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.	Parcial Práctico 2 Recuperatorio 2
RA.2. Implementa correctamente algoritmos para la manipulación de listas, pilas y colas, demostrando competencia en la resolución de problemas de programación usando el Paradigma Orientado a Objetos.	CE.2.1. Demuestra comprensión de los conceptos de abstracción y encapsulamiento 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 del concepto de herencia entre clases en la programación orientada a objetos (POO) aplicándolo en el desarrollo del proyecto solución para el problema planteado.	Parcial Práctico 2 Recuperatorio 2
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 lineales como listas, pilas y colas.	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 listas, pilas y colas en soluciones prácticas.	CE.4.1. Implementa correctamente un proyecto de programación orientada a objetos (POO) en lenguaje Java, utilizando listas simples encadenadas ordenadas, pilas y colas 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 recursividad 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.	Se aplica en actividades de aula y actividades aplicativas que se proponen al alumno a fin de integrar la teoría y la práctica durante el cursado.



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.
- Aprobar los dos (2) Parciales Prácticos previstos.
- Completar la presentación de todos los ejercicios de Trabajos Prácticos solicitados.

8.5.2. Condiciones para lograr la Regularidad de la Asignatura.

Para regularizar la asignatura el alumno deberá:

- Aprobar los dos (2) Parciales Prácticos previstos, o sus correspondientes Recuperatorios.
- Completar la presentación de todos los ejercicios de Trabajos Prácticos solicitados.

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