

**UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y
TECNOLOGÍAS**

PLANIFICACIÓN ANUAL 2023

ASIGNATURA: LABORATORIO III

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

Equipo cátedra:

Equipo cátedra:

Profesor Adjunto: Méndez Analía

Aux.Doc.de 1º Titulado: Lic. Federico Rosenzvaig



PLANIFICACIÓN DE LA ASIGNATURA

1- IDENTIFICACIÓN:

- 1.1. **Nombre de Asignatura:** Laboratorio 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:** Tercer 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	10
Algoritmia básica	15
Paradigmas de Programación	35
CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR	60

Tabla 1: Carga horaria por área/bloque/tramo

1.6.3. Correlativas

1.6.3.1 Anteriores:

Débiles: Inglés II, Laboratorio II, Programación I.

Fuerte: Inglés I, Laboratorio I.

1.6.3.2. Posteriores:

Débiles: Programación III.

Fuertes: Laboratorio V, Redes.



1.7. Carga horaria:

1.7.1. Carga horaria semanal total

1.7.1.1. **Presencial:** 4.

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 conduce al alumno al conocimiento de un lenguaje moderno, indispensable para la actuación profesional de acuerdo a los requerimientos actuales de los desarrollos de sistemas. Propone, así mismo, una instancia de aplicación, ejercitación, revisión y estudio de estructuras de datos, teorías y técnicas aprendidas en otras asignaturas, integrándolas en el desarrollo de aplicaciones utilizando el lenguaje de programación Java.

En la asignatura Laboratorio III se analiza y estudian conceptos avanzados referidos al desarrollo de programas utilizando el lenguaje Java.

Los conceptos relativos a la Programación Orientada a Objetos tales como herencia, polimorfismo, objetos functor, etc., encuentran en Java el contexto ideal de aplicación proporcionando al alumno la oportunidad de desarrollar sus destrezas en el desarrollo de sistemas.

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

En relación a la asignatura Laboratorio I:

- Destrezas relativas al desarrollo de programas básicos en lenguaje C.
- Comprensión y aplicación de conceptos básicos de la programación en computadora.

En relación a la asignatura Laboratorio II:

- Destrezas en el uso y aplicación de lenguajes de programación como C y Java.
- Conocimiento y habilidad en el manejo de herramientas de desarrollo (IDE).

En relación a la asignatura Programación I:

- Conocimiento y comprensión de estructuras de datos básicas, sus operaciones e implementación.
- Comprensión de los conceptos básicos relativos al Paradigma de Programación Orientada a Objetos (POO).



2.3. Aspectos del Perfil Profesional a los que contribuye la Asignatura.

La asignatura Laboratorio III contribuye con aspectos del perfil profesional tales como:

- Otorga experiencia práctica en el uso del lenguaje de programación Java.
- Capacita en la utilización fluida de lenguajes de programación tales como Java.
- Motiva al alumno en la aplicación de metodologías y técnicas informáticas.
- Motiva al alumno en el desarrollo de una actitud crítica y flexible frente a los avances tecnológicos que se manifiestan en el campo de las aplicaciones de software, específicamente en relación al desarrollo de software y lenguajes de programación.
- Colabora con 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.
- Prepara recursos humanos calificados y competentes, de nivel universitario, en el ámbito técnico de la Informática.

2.4. Integración horizontal y vertical con otras asignaturas.

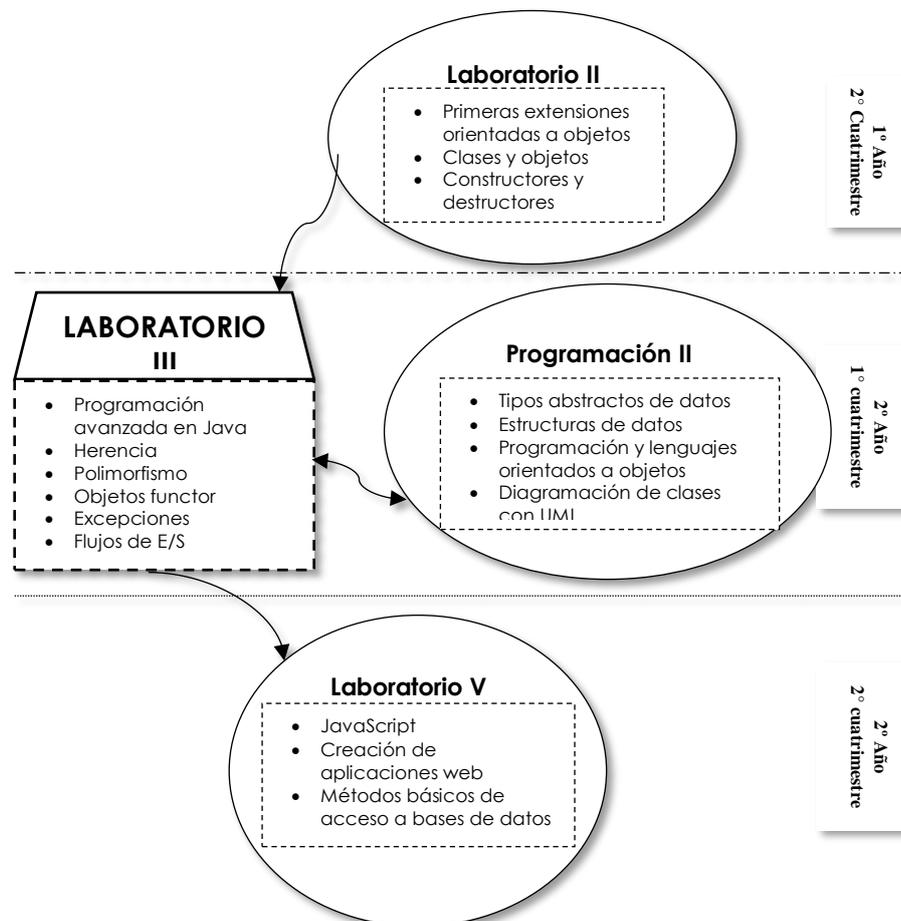


Figura 1. Integración horizontal y vertical de Laboratorio III con otras asignaturas.



3- OBJETIVOS: Que el alumno desarrolle las siguientes competencias...

Competencias transversales o genéricas.

Competencias instrumentales:

- Capacidad de organización, de abstracción y comprensión.
- Capacidad para la resolución de problemas.

Competencias personales:

- Habilidad para documentar y realizar el seguimiento de sus procesos mentales.
- Capacidad de trabajo en equipo.
- Destreza en el razonamiento lógico.

Competencias sistémicas:

- Motivación por la calidad.

Competencias específicas.

Competencias cognitivas (saber):

- Reconocer las propiedades de la POO y su aplicación usando Java.
- Reconocer la función de las excepciones en la construcción de programas.
- Interpretar el concepto de Herencia en la POO y sus características y ventajas.
- Comparar las características de las Interfaces y las clases abstractas.
- Clasificar las principales clases para el manejo de flujos de E/S.

Competencias procedimentales e instrumentales (saber hacer):

- Usar, en forma eficiente, la diagramación UML (Unified Modeling Language) en la documentación de aplicaciones de software.
- Ejecutar y verificar programas en lenguaje de programación JAVA aplicando Herencia, Polimorfismo, Clases Abstractas e Interfaces y reutilización de código.
- Implementar el uso y creación de excepciones en la construcción de programas.

Competencias actitudinales (ser):

- Trabajar de forma autónoma y con inquietud para la eficiencia y el rigor.
- Comunicar resultados de forma clara y precisa.
- Reaccionar frente al error identificando y verificando nuevas estrategias para alcanzar el logro exigido.
- Llevar a cabo la búsqueda creativa de soluciones algorítmicas en grupo e individualmente.



4- SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

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

Programación avanzada en Java. Objetos y referencias. Miembros de instancia y miembros estáticos. Paquetes. Reutilización de clases. Herencia. Tipos de herencia. Polimorfismo. El tipo Object. Clases y métodos abstractos. Implementación de componentes genéricos. Interfaces y clases genéricas. Autoboxing/unboxing. Objetos función o functor. Clases anidadas, locales y anónimas. Tratamiento de excepciones. La clase Exception en Java. Generación y captura de excepciones en Java. Flujos o streams de entrada/salida.

4.2. Programa Sintético sobre la base de los contenidos mínimos

Unidad 1. Objetos y clases.

Programación avanzada en Java. Objetos y referencias. Método toString. Método equals. La referencia this. Miembros de instancia y métodos estáticos. Paquetes.

Unidad 2. Herencia y genericidad en Java.

Herencia. Reutilización de clases. Tipos de herencia. Polimorfismo. El tipo Object. Clases y métodos abstractos. Implementación de componentes genéricos. Interfaces y clases genéricas. Autoboxing/unboxing. Objetos función o functor. Clases anidadas, locales y anónimas.

Unidad 3. Excepciones en Java.

Tratamiento de excepciones. La Clase Exception en Java. Generación y captura de excepciones en Java. Declaración de nuevos tipos de excepciones.

Unidad 4. Flujos de E/S.

El sistema de E/S de Java. Flujos o streams de E/S. Archivos y flujos. Lectura y escritura de archivos. La clase File. Serialización de objetos.

4.3. Articulación Temática de la Asignatura /Obligación Curricular

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

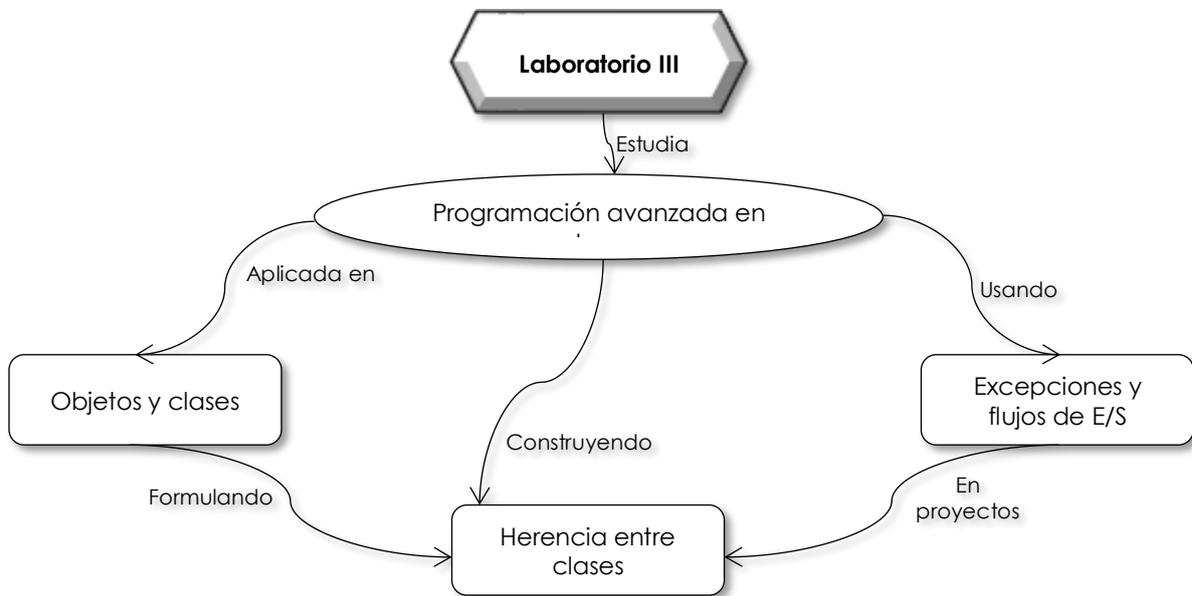


Figura 2. Articulación temática de la asignatura.

4.4. Programa Analítico

Unidad 1. Objetos y clases.

Programación avanzada en Java. Tipos referencia. Objetos y referencias. Objetos y clases. Constructores sobrecargados. Mutadores y accesores. Método toString. Método equals. La referencia this. Miembros de instancia y métodos estáticos. Paquetes. Relación de composición entre clases.

Unidad 2. Herencia y genericidad en Java.

Herencia. Reutilización de clases. Tipos de herencia. Clases derivadas. Visibilidad de clases. Upcasting. Clases y métodos final. Polimorfismo. El tipo Object. Clases y métodos abstractos. Tipo genérico. Implementación de componentes genéricos. Interfaces y clases genéricas. Autoboxing/unboxing. Objetos función o functor. Clases anidadas, locales y anónimas.

Unidad 3. Excepciones en Java.

Tratamiento de excepciones. Definición de excepciones. La Clase Exception en Java. Generación y captura de excepciones en Java. La cláusula finally. Excepciones comunes. Cláusulas throw y throws. Declaración de nuevos tipos de excepciones.

Unidad 4. Flujos de entrada/salida.

El sistema de E/S de Java. Flujos o streams de E/S. Concepto de flujo en Java. Tipos de flujos. Uso de flujos de E/S. Archivos y flujos. Lectura y escritura de archivos. La clase File. Archivos de texto de acceso secuencial. Serialización de objetos.



4.5. Cronograma para el desarrollo de las Unidades Temáticas

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Unidad 1: Objetos y clases	2	Semanas 1 a 2
Unidad 2: Herencia y genericidad en Java	7	Semanas 3 a 9
Unidad 3: Excepciones en Java	2	Semanas 10 a 11
Unidad 4: Flujos de E/S	4	Semanas 12 a 15
TOTAL	15	

Tabla 2: Cronograma para el desarrollo de las unidades temáticas

4.6. Programa y cronograma de formación práctica

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
TP1. Objetos y referencias	6	Semanas 1 a 2
TP2. Herencia	9	Semanas 3 a 5
TP3. Tipos genéricos	12	Semanas 6 a 9
TP4. Creación de excepciones	6	Semanas 10 a 11
TP5. Flujos de E/S	12	Semanas 12 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
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 4: Bibliografía

Tabla 4: Bibliografía (continuación)



TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Introducción a la programación, lógica y diseño (7° 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/		
Programación orientada a objetos usando java	Flórez Fernández, H. A.	Ecoe Ediciones	Biblioteca Digital eLibro.net	2012
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/69236		
Java 2: manual de usuario y tutorial (5a. ed.)	Froufe Quintas, A.	RA-MA Editorial	Biblioteca Digital eLibro.net	2014
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/106400		
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/		
Jiménez de Parga, C. (2021). UML: arquitectura de aplicaciones en Java, C++ y Python (2a. ed.)	Jiménez de Parga C..	RA-MA Editorial	Biblioteca Digital eLibro.net	2014
	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/		
Programación Java del análisis de problemas al diseño de programas (5a. ed.)	Malik, D. S.	Cengage Learning.	Biblioteca Digital eLibro.net	2013
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/93217		
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/		

Tabla 4: Bibliografía (continuación)



TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Empezar a programar usando Java (3° 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/		
Fundamentos de la programación orientada a objetos: una aplicación a las estructuras de datos en Java	Ruiz Rodríguez, R.	El Cid Editor.	Biblioteca Digital eLibro.net	2009
	ENLACE DE ACCESO	https://elibro.net/es/lc/unsebiblio/titulos/34869		
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 la Programación Orientada a Objetos usando UML y Java	Vélez Serrano, J.	Dykinson	Biblioteca Digital eLibro.net	2011
	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

6- ESTRATEGIAS METODOLÓGICAS.

6.1. Aspectos pedagógicos y didácticos

El desarrollo de la asignatura estará caracterizado por una combinación de exposición oral explicativa por parte de los docentes y de un estudio dirigido de actividades prácticas en el que los alumnos trabajarán en el laboratorio de computadoras desarrollando programas en el lenguaje de programación Java y donde se facilitarán y promoverán actividades relacionadas con el desarrollo de aplicaciones informáticas.

Las actividades de estudio tanto prácticas como de tipo teórico serán apoyadas y desarrolladas mediante la plataforma Moodle del Centro Universitario Virtual (CUV). Mediante esta plataforma se llevarán a cabo actividades tales como cuestionarios evaluativos y de diagnóstico, talleres, actividades domiciliarias, etc.

En el desarrollo de las actividades áulicas se destaca el uso de las instalaciones del laboratorio de computadoras, tanto para llevar a cabo ejercicios prácticos respectivos a los temas en estudio así como para el uso de herramientas de apoyo didáctico. Particularmente, herramientas como el videoproector se utilizarán para acompañar al alumno en las actividades prácticas y ejercicios que se desarrollen tanto en las clases



prácticas como en las teóricas. Las actividades áulicas se caracterizarán por una integración entre los conceptos teóricos impartidos y su aplicación en ejercicios prácticos. Cada uno de los trabajos prácticos propuesto en esta asignatura incluye el desarrollo de problemas mediante diagramación UML de clases y 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 plantea coordinación horizontal con la asignatura Programación II. Ésta integra el mismo módulo y año de la carrera de Programador Universitario en Informática y capacita al alumno en el tratamiento de estructuras de datos así como respecto a los principales conceptos referidos al desarrollo de programación en computadoras.

La asignatura Laboratorio III incluye en sus trabajos prácticos y taller la resolución de problemas aplicando las estructuras de datos cuyo tratamiento y estudio en profundidad se lleva a cabo en la asignatura Programación II. Para la coordinación de estas actividades se realizan reuniones entre los docentes de la práctica.

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, y el equipamiento computacional de los laboratorios del Departamento de Informática. Estos constituyen los principales recursos que se utilizan para el desarrollo de las clases tanto teóricas como prácticas y en particular de los talleres. Es con los trabajos prácticos y los talleres que los alumnos logran adquirir habilidad para programar en 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 y apuntes elaborados por el equipo docente de la asignatura.
- En las clases teóricas y en las clases prácticas se utilizará pizarra, proyector, Internet, enunciados tomados de las actividades prácticas propuestas, diapositivas elaboradas por los docentes de la asignatura, demos.

Además, se contará con el espacio de la asignatura en la plataforma Moodle del Centro Universitario Virtual (CUV), así como un grupo en FaceBook dedicado al curso, que permitirán al alumno acceder a consultas virtuales (on-line o secuencial) a distancia,



acceso a los recursos digitales, apoyo de prácticas y talleres. 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

Durante la primera semana de clase se realizará una evaluación diagnóstica que permita constatar el nivel de conocimientos previos del alumnado. Se llevará a cabo de modo asincrónico mediante recursos aplicables en la plataforma del CUV.

Esta estrategia proporcionará información útil para el desenvolvimiento inicial de las actividades a fin de preparar al alumnado adecuadamente para el logro de los objetivos relativos al desarrollo de esta asignatura.

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. Se implementará utilizando la herramienta cuestionario de la plataforma del CUV.

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.

7.3. Evaluación Parcial

7.3.1. Programa de Evaluaciones Parciales

EVALUACIÓN	TEMA	MODALIDAD	SEM	FECHA	DEVOLUCIÓN DE RESULTADO
Parcial Práctico 1	Herencia y tipos genéricos	Individual. Escrito. En computadora. Resolución de un problema propuesto mediante: <ul style="list-style-type: none">• Diagramación UML de clases aplicando POO.• Diagramación de flujo.• Desarrollo de proyecto de POO con lenguaje Java.	9	16/05	19/05
Recuperatorio 1			11	30/05	02/06
Parcial Práctico 2	Objetos functor y excepciones		14	21/06	23/06
Recuperatorio 2			15	28/06	30/06

Tabla 5: Descripción y cronograma de Evaluaciones Parciales



7.3.2. Criterios de Evaluación

ESTRUCT.	OBJETIVO	CRITERIOS DE EVALUACIÓN	POND.DE CALIFIC.
Parcial Práctico y Recuperatorio: Herencia y tipos genéricos			
Diag. UML de clases y diag.de flujo de la resolución de un problema.	Determinar si los alumnos han adquirido las habilidades necesarias para resolver un problema de computadora aplicando análisis orientado a objetos y diagramación UML de clases.	Correcta aplicación de diagramación UML de clases. Correcta identificación de clases, variables miembros y métodos en el problema planteado. Resolución algorítmica en tiempo y forma adecuados.	35%
Resolución de problemas mediante la programación en Java	Determinar si los alumnos han adquirido las competencias necesarias para: <ul style="list-style-type: none"> • Reconocer las propiedades de la POO y su aplicación usando Java. • Construir proyectos de varias clases de acuerdo con los conceptos de la POO. • Desarrollar programas en Java aplicando Herencia, Polimorfismo, Clases Abstractas e Interfaces. • Emplear adecuadamente el tipo genérico en el desarrollo de programas. 	Adecuada resolución del planteamiento propuesto aplicando POO y los conceptos de herencia y tipos genéricos. Correcta organización de las jerarquías de clases y eficiente reutilización de código.	65%
Parcial Práctico y Recuperatorio: Objetos functor y excepciones			
Diagramación UML de clases y diagramación de flujo de la resolución de un problema.	Determinar si los alumnos han adquirido las habilidades necesarias para resolver un problema de computadora aplicando análisis orientado a objetos y diagramación UML de clases usando archivos.	Correcta aplicación de diagramación UML de clases. Correcta identificación de clases, variables miembros y métodos en el problema planteado. Resolución algorítmica en tiempo y forma adecuados.	35%
Resolución de problemas mediante la programación en Java	Determinar si los alumnos han adquirido las competencias necesarias para: <ul style="list-style-type: none"> • Desarrollar excepciones propias para utilizar en la construcción de programas. • Aplicar el concepto de flujos en Java. • Implementar lectura y escritura de archivos de texto mediante el uso de las clases del paquete java.io. 	Adecuada resolución del planteamiento propuesto aplicando POO, flujos en Java y excepciones. Correcta organización de las jerarquías de clases y eficiente reutilización de código.	65%

Tabla 6: Estructura, objetivos y criterios de Evaluaciones Parciales



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 en modalidad individual mediante el desarrollo de un proyecto de software aplicando POO y utilizando tipos genéricos, herencia, clases abstractas o interfaces, objetos functor, excepciones y, en caso de que corresponda, flujos de E/S. La presentación de este trabajo deberá incluir la debida documentación consistente de diagramación UML de clases y diagramación de flujo.

7.5. Evaluación Sumativa

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

- a) Asistir como mínimo al 80% del total de sesiones que serán registradas en la asistencia del CUV.
- b) Presentar y/o exponer todas las actividades de teoría propuestas, en tiempo y forma.
- c) Presentar todos los trabajos prácticos en tiempo y forma, correctamente documentados.
- d) Aprobar los Parciales Prácticos.
- e) Aprobar el Trabajo Integrador.

7.5.2. Condiciones para lograr la Regularidad de la Asignatura.

- a) Haber completado como mínimo un 40% de todas las actividades teóricas propuestas mediante el CUV, en tiempo y forma.
- b) Presentar mediante el CUV y personalmente como mínimo los Trabajos Prácticos 3 y 4 en tiempo y forma, o en la extensión de su presentación. correctamente documentados.
- c) Aprobar los Parciales Prácticos o sus correspondientes recuperatorios.

7.6. Examen Final.

El examen es de carácter oral y escrito. El alumno deberá presentar una exposición oral respondiendo preguntas teórico-prácticas, y/o resolver planteamientos mediante computadora, aplicando POO y utilizando lenguaje Java.

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. El trabajo deberá ser presentado mediante la plataforma del CUV, cumplimentando las pautas establecidas en el enunciado, la presentación deberá realizarse al término del plazo de días hábiles indicado a partir de la entrega del enunciado.



Segunda Etapa: Aprobar una evaluación escrita de contenidos teóricos del programa analítico.

Tercera Etapa: Aprobar una evaluación en la computadora similar al examen final de los alumnos regulares.

Méndez Analía

.....
Profesor responsable de Asignatura