

**UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y
TECNOLOGÍAS**

PLANIFICACIÓN ANUAL 2023

ASIGNATURA: LABORATORIO II

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

Equipo cátedra:

Profesor Adjunto: Méndez Analía

JTP: Montalvetti Mario

Auxiliar Docente de Primera: Gómez Pizarro María Teresa

Auxiliar Docente de Primera: Rosenzvaig Federico



PLANIFICACIÓN DE LA ASIGNATURA

1- IDENTIFICACIÓN:

1.1. **Nombre de Asignatura:** Laboratorio II.

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:** Segundo módulo – Primer 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	15
Algoritmia básica	20
Paradigmas de Programación	25
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:** Inglés I, Fundamentos de la Programación, Laboratorio I.

1.6.3.2 **Posteriores:** Programación III, Laboratorio III.

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: el dictado de la asignatura se organiza en dos comisiones de práctica con el objetivo de fortalecer en el alumno el desarrollo de habilidades en la programación de computadoras.

2- PRESENTACIÓN

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

En esta asignatura se estudian aspectos avanzados del Lenguaje C, tales como punteros, convocatorias de funciones por valor y por referencia, y el tipo struct. Se introducen los conceptos de Programación Orientada a Objetos mediante la programación en lenguaje Java. La asignatura se desarrolla en base a la utilización de una modalidad netamente teórico – práctica, motivando el estudio y aplicación de dichos lenguajes de programación a partir del planteamiento de la resolución de problemas seleccionados para la mejor y mayor comprensión de los aspectos teóricos.

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

- Conocimientos básicos de operación de un sistema de computación y programación en lenguaje C, adquiridos en Laboratorio I.
- Conceptos esenciales referentes a paradigmas de programación y enfoque algorítmico aplicado en la resolución de problemas, adquiridos en la asignatura Fundamentos de la Programación.

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

Esta asignatura contribuye a:

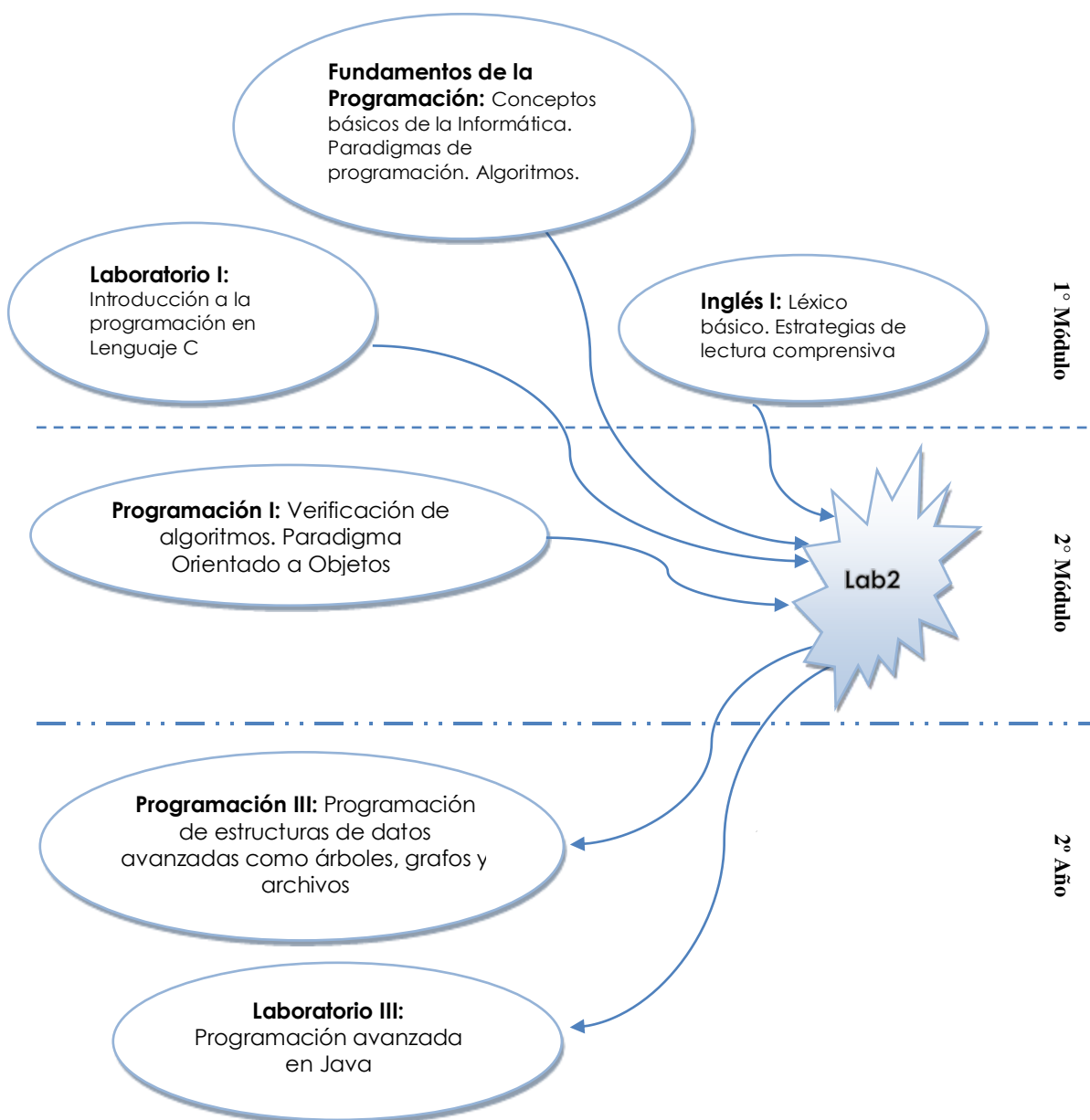
- La formación técnica del alumno, abarcando los conocimientos destinados al aprendizaje y al uso de lenguajes de programación y herramientas informáticas en entornos específicos para la resolución de los problemas concretos.
- Preparar recursos humanos calificados y competentes, de nivel universitario,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



2.4. Integración horizontal y vertical con otras asignaturas.

Se presenta a continuación un esquema gráfico representativo de la integración horizontal y vertical con otras asignaturas.

En este esquema se expresan las vinculaciones entre los temas principales de la asignatura Laboratorio II (LAB2) y las asignaturas correlativas previas y posteriores más directas, así como también con la asignatura que se desarrolla en el mismo módulo.





3- OBJETIVOS

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

Competencias básicas.

- Habilidad para trabajar de forma autónoma.
- Inquietud para la eficiencia y el rigor.
- Capacidad para reaccionar frente al error identificando y verificando nuevas estrategias para alcanzar el logro exigido.
- Disposición al análisis crítico y auto-crítico sobre código de programa.

Competencias transversales o genéricas.

- Capacidad de organización, de abstracción y comprensión.
- Habilidad para documentar y realizar el seguimiento de sus procesos mentales en la resolución de problemas.
- Destreza en el razonamiento lógico.
- Capacidad para identificar, plantear y resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.

Competencias específicas.

- Desarrollar y verificar programas en lenguaje C usando funciones convocadas por valor y por referencia, punteros, y la definición de registros de datos mediante el tipo struct, para resolver problemas de computación.
- Aplicar los conceptos básicos de la POO, abstracción y encapsulamiento, identificando objetos y clases para representarlos en la resolución de problemas usando lenguaje de programación Java.

4- SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

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

Programación avanzada en lenguaje C. Variables puntero. Aritmética de punteros. Punteros y arreglos. Punteros a punteros. El tipo struct. Punteros y el tipo struct. Funciones: Prototipo de función y tipos de funciones. Llamada de funciones por valor y por referencia. Introducción a la programación orientada a objetos en Java. Características de Java. Tipos de datos primitivos. Operadores y expresiones. Estructuras de selección y de control. Clases, objetos y mensajes en Java. Métodos básicos. Constructores. Encapsulamiento de la información en Java.



4.2. Programa Sintético sobre la base de los contenidos mínimos

Unidad 1. Programación avanzada en lenguaje C.

Programación avanzada en lenguaje C. Variables puntero. Aritmética de punteros. Punteros y arreglos. Punteros a punteros. El tipo struct. Punteros y el tipo struct.

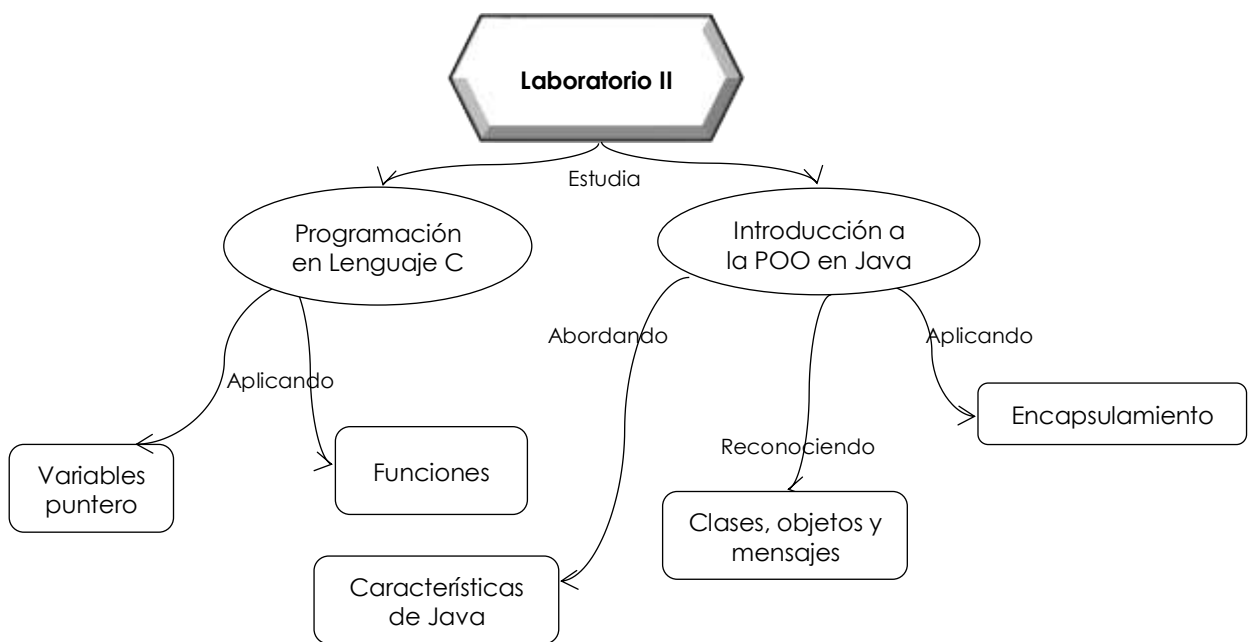
Unidad 2. Funciones.

Funciones: Prototipo de función y tipos de funciones. Llamada de funciones por valor y por referencia.

Unidad 3. Programación orientada a objetos en Java.

Introducción a la programación orientada a objetos en Java. Características de Java. Tipos de datos primitivos. Operadores y expresiones. Estructuras de selección y de control. Clases, objetos y mensajes en Java. Métodos básicos. Constructores. Encapsulamiento de la información en Java.

4.3. Articulación Temática de la Asignatura





4.4. Programa Analítico

Unidad 1. Programación avanzada en lenguaje C.

Programación avanzada en lenguaje C. Variables puntero. Aritmética de punteros. Punteros y arreglos. Punteros a punteros. El tipo struct. Punteros y el tipo struct.

Unidad 2. Funciones en C.

Funciones: Prototipo de función y tipos de funciones. Clases de almacenamiento y funciones. Función de tipo void. Funciones de tipo char. Funciones enteras. Funciones de tipo punto flotante. Argumentos de funciones. Parámetros formales y parámetros actuales de una función. Uso de void como parámetro. Argumentos enteros. Argumentos de punto flotante. Llamadas de funciones por valor y por referencia.

Unidad 3. Programación orientada a objetos en Java.

Introducción a la programación orientada a objetos en Java. Características de Java. Tipos de datos primitivos. Operadores y expresiones. Estructuras de selección y de control. Entrada/Salida estándar en Java. Clases, objetos y mensajes en Java. Métodos básicos. Constructores. Encapsulamiento de la información en Java.



4.5. Cronograma para el desarrollo de las Unidades Temáticas

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Programación avanzada en lenguaje C	14	Semanas 1 a 4
Funciones en C	18	Semanas 3 a 8
Programación orientada a objetos en Java	28	Semanas 9 a 15
TOTAL	60	

Tabla 2: Cronograma para el desarrollo de las unidades temáticas

4.6. Programa y cronograma de formación práctica

4.6.1. Cronograma de trabajos prácticos

ACTIVIDAD	CARGA HORAS	CRONOGRAMA DE DESARROLLO		
		Semanas	Carácter	Fecha presentación
Trabajo Práctico 1: Arreglos en lenguaje C. Conversiones de tipo. Funciones de biblioteca de Entrada – Salida. Strings.	6	1 – 2	Obligatorio	03/09
Trabajo Práctico 2: Variables puntero	4	3 – 4		10/09
Trabajo Práctico 3: El tipo struct	3	5		17/09
Trabajo Práctico 4: Funciones definidas por el usuario.	4	5 – 6		24/09
Trabajo Práctico 5: Convocación de funciones por valor y por referencia.	3	6		28/09
Trabajo Práctico 6: Introducción a la POO en Java.	3	10		22/10
Trabajo Práctico 7: Encapsulamiento de la información en Java.	3	11		29/10
Trabajo Práctico 8: Tipos primitivos y tipos referencia.	4	12 – 13		12/11
TOTAL	30			

Tabla 3.a: Cronograma para el desarrollo de los trabajos prácticos

4.6.2. Cronograma de formación experimental

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO		
		Semanas	Carácter	Fecha presentación
Taller 1: Programación avanzada en Lenguaje C: Punteros. Tipos estructurados (struct). Funciones.	9	6 – 7 – 8	Obligatorio	Semana de 03/10
Taller 2: Introducción a la POO en Java	6	13 – 14		Semana de 14/11
TOTAL	15			

Tabla 3.b: Cronograma para el desarrollo de la formación experimental



5- BIBLIOGRAFÍA.

TÍTULO	AUTORES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Lenguaje C.	Kernighan & Ritchie	Prentice – Hall. Hispanoamericana S.A.	1	1991
Como programar en C/C++ 2º Edición.	Deitel Harvey		1	1995
Manual de Borland C++.	Pappas Chris & Murray William.	Osborne/MacGraw Hill. Interamericana de España. S.A.	1	1993
Estructura de datos en JAVA	Weiss Mark Allen.	Pearson	1	2013
Estructuras de datos en Java	Joyanes Aguilar Luis, Zahonero Martinez Ignacio	McGraw-Hill	1	2008
Metodología de la programación: Diagramas de flujos, algoritmos y programación estructurada.	Joyanes Aguilar Luis		1	1987
Fundamentos de programación. Algoritmos y estructura de datos.			1	2000
Piensa en JAVA	Bruce Eckel	Prentice Hall	1	2002
UML y Patrones	Larman, Craig	Pearson - Prentice Hall	1	2003

Tabla 4: Bibliografía

6- ESTRATEGIAS METODOLÓGICAS.

6.1. Aspectos pedagógicos y didácticos

Las clases de la asignatura serán de carácter: (a) teórico – práctico y (b) prácticas.

Las explicaciones teóricas se conjugarán con la exposición y seguimiento de ejemplos. En la presentación de los contenidos teóricos se emplearán medios audiovisuales disponibles con el objeto de clarificar y profundizar en los detalles referentes a los lenguajes de programación utilizados y brindar al alumno una clara descripción.

Las clases teórico – prácticas conformarán el fundamento de desarrollo de las clases prácticas; las clases se coordinarán a fin de proporcionar al alumno el marco conceptual, la ejemplificación y la ejercitación suficiente para alcanzar el desenvolvimiento de fuertes destrezas en el desarrollo de programas utilizando los lenguajes C y Java.

Mediante la plataforma Moodle del CUV (Centro Universitario Virtual) se llevarán a cabo las entregas de las tareas prácticas y de taller y las actividades de evaluación formativa que complementarán el desarrollo de la teoría.



Las actividades prácticas se organizarán en función del desarrollo de prácticos y Talleres, que implicarán una dedicación extra clase por parte del alumno.

Ya sea en grupos de estudio o individualmente, el alumno deberá dedicar horas fuera de clase para el desarrollo de las habilidades propias de la programación en computadora.

En las clases prácticas los docentes a cargo resolverán consultas de la clase, utilizarán ejercicios tipo para explicar la aplicación del lenguaje C y el lenguaje Java, y orientarán al alumno en la resolución de los prácticos y Talleres, haciendo hincapié en metodologías, organización, y prácticas adecuadas a la programación eficiente en computadora.

Las actividades áulicas implicarán la realización de tareas tales como:

- Revisión bibliográfica e investigación previamente a las clases.
- Comprensión y ejercitación práctica sobre conceptos en estudio utilizando la diagramación estructurada y el entorno de programación.
- Resolución de problemas mediante la aplicación de estructuras, sentencias, y otros componentes del lenguaje estudiados.
- Actividades de discusión orientadas por el docente para una mejor comprensión de criterios de programación eficiente.
- Planteamientos para resolver en grupo.
- Actividades de resolución de problemas y taller.

6.2. Mecanismos para la integración de docentes

Se plantea principalmente la integración con las asignaturas Fundamentos de la Programación y Laboratorio I, en las que los alumnos apropiaran conceptos de programación, diagramación estructurada y resolución de problemas y la introducción a la programación con Lenguaje C, respectivamente.

Las actividades prácticas de Laboratorio II se plantean teniendo en cuenta y reforzando los conocimientos apropiados en Programación I, para esto se llevan a cabo encuentros de coordinación entre representantes de los equipos cátedra de ambas asignaturas.

La integración con asignaturas posteriores, como Laboratorio III y Programación III, se lleva adelante a través de encuentros formales e informales con representantes de la asignatura, identificando habilidades requeridas, analizando los conocimientos previos y posteriores del alumnado, a fin de prepararlo adecuadamente y/o ajustar el dictado en cuanto a aspectos metodológicos y prácticas.

6.3. Recursos Didácticos

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

- Ambiente integrado de desarrollo (IDE) para la programación en los lenguajes C y Java, y equipamiento computacional de los laboratorios del Departamento de Informática. Estos constituyen los principales recursos que se utilizan para el desarrollo de los talleres. Mediante ellos, los alumnos logran adquirir habilidad para programar en Lenguaje C y Java.
- Compilador de C y Java.



- Bibliografía actualizada y apuntes elaborados por el equipo docente de la asignatura. Estos se utilizarán como una manera de acercar a los alumnos a documentación técnica referida al uso del Lenguaje C y de Java.
- En las clases teóricas y en las clases prácticas se utilizará enunciados tomados de las actividades prácticas propuestas, diapositivas y videos elaborados por los docentes de la asignatura, demos, tutoriales, etc.
- Se contará con el espacio de la asignatura en la plataforma Moodle del Centro Universitario Virtual (CUV), que permitirá al alumno acceder a consultas, acceso a los recursos digitales, apoyo de prácticas, etc. También brindará a los alumnos un canal de comunicación permanente donde podrán acceder a información actualizada de la asignatura: fechas de evaluaciones, resultados de parciales, condición final de la cursada, etc.

7- EVALUACIÓN

7.1. Evaluación Diagnóstica

Mediante la utilización de la herramienta Cuestionario, propia de la plataforma Moodle del Centro Universitario Virtual (CUV), se llevará a cabo durante las dos primeras semanas del cuatrimestre la Evaluación diagnóstica.

Esta evaluación estará orientada con el objetivo de constatar el nivel de conocimientos previos del alumnado. Así mismo, 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.

7.2. Evaluación Formativa

Esta modalidad de evaluación 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. Se evaluará:

- Participación del alumno en clase y en actividades propuestas a través del CUV.
- Disposición y desempeño del alumno en la resolución de las actividades prácticas.
- Presentación en tiempo y forma de las prácticas propuestas.
- Capacidad de resolución y de análisis de los problemas de carácter teórico y práctico que se le presenten al alumno.

Las actividades que se aplicarán para la Evaluación Formativa serán:

- Consultas en la finalización de cada clase destinadas a obtener una retroalimentación del alumno respecto a su nivel de comprensión del tema tratado.
- Desarrollo de enunciados de los trabajos prácticos que deberán ser enviados en fechas determinadas mediante la plataforma del CUV.



7.3. Evaluación Parcial

7.3.1. Programa de Evaluaciones Parciales

PARCIALES	TEMAS	MODALIDAD	SEM	DEVOLUCIÓN DE RESULTADOS
Parcial Práctico 1	Arreglos. Funciones en lenguaje C. Strings. Punteros. Registros en C	• Cuestionario multiple-choice sincrónico sobre conceptos teóricos mediante plataforma del CUV.	7	Semana 8
Recuperatorio Parcial Práctico 1			9	Semana 10
Parcial Práctico 2	Introducción a la POO y el lenguaje Java	• Resolución en computadora de un enunciado propuesto mediante el uso del lenguaje de programación en laboratorio.	14	18/11
Recuperatorio Parcial Práctico 2			15	24/11

7.3.2. Criterios de Evaluación

	OBJETIVOS	CRITERIOS DE EVALUACIÓN
PARCIAL 1 Y RECUPERATORIO	Determinar si los alumnos han adquirido las habilidades necesarias para aplicar funciones definidas por el usuario, funciones de biblioteca de tratamiento de strings y de E/S, manejo de punteros y el tipo struct.	<ul style="list-style-type: none">• Capacidad para aplicar prototipo, definición y convocación de funciones.• Capacidad para aplicar variables punteros.• Habilidad para reconocer la necesidad de uso del tipo struct y aplicarlo en la resolución de problemas.
PARCIAL 2 Y RECUPERATORIO	Determinar si los alumnos han adquirido los conocimientos suficientes relativos a la POO en Java, identificación de clases, objetos, atributos y métodos para la resolución de problemas en computadoras, aplicando los conceptos de abstracción, y encapsulamiento.	<ul style="list-style-type: none">• Capacidad para desarrollar programas usando sentencias de Java.• Capacidad para aplicar el encapsulamiento en la programación orientada a objetos.• Comprensión algorítmica y aplicación eficiente de los conceptos fundamentales de la programación orientada a objetos.



7.3.3. Escala de Valoración

ACTIVIDAD	ESCALA DE VALORACIÓN	RECUPERACIÓN	PROMOCIÓN
Trabajos Prácticos	Cualitativa dicotómica (aprobado – desaprobado)	Instancias recuperatorias mediante extensión de presentaciones en plataforma del CUV	Presentados todos los trabajos prácticos en tiempo y forma.
Parcial Práctico 1	Cuantitativa politómica. Escala: 1 al 100. El puntaje mínimo para aprobar es de cinco (5) puntos, sobre una calificación máxima de 10.	Recuperatorio Práctico 1	Obtener como mínimo siete (7) puntos sin llegar a la instancia de recuperación.
Parcial Práctico 2		Recuperatorio Práctico 2	
Taller 1	Cualitativa dicotómica (aprobado – desaprobado)	Recuperatorio Práctico 1	Taller aprobado en su instancia inicial.
Taller 2		Recuperatorio Práctico 2	

7.4. Evaluación Integradora

La evaluación integradora se llevará a cabo a través de una actividad en la que los alumnos, de manera individual, realizarán el análisis y corrección de códigos de programas y la elaboración de informes referentes a componentes del lenguaje C y la comparación entre los lenguajes C y Java.

7.5. Evaluación Sumativa

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

- Asistir como mínimo al 80% del total de sesiones.
- Aprobar los Parciales Prácticos con un puntaje ≥ 7 puntos.
- Aprobar los talleres y los trabajos prácticos sin llegar a la instancia de recuperación.
- Aprobar el Trabajo Integrador.

7.5.2. Condiciones para lograr la Regularidad de la Asignatura.

- Aprobar los Parciales o sus recuperatorios.
- Presentar los trabajos prácticos y aprobar los talleres propuestos.



-
- c) Todo alumno que no haya completado la presentación de los trabajos prácticos y/o no haya aprobado un taller deberá aprobar la evaluación recuperatoria correspondiente al tema.

7.6. Examen Final

Podrá ser una evaluación oral o escrita.

En el caso de ser oral el alumno dispondrá de una computadora para responder las requisitorias que se le realicen, se tratará de solicitudes puntuales de programación.

En el caso de la evaluación escrita el alumno deberá resolver un enunciado en computadora.

Se evaluará el nivel de conocimiento en la aplicación práctica de los lenguajes C y Java en la resolución de problemas, respondiendo a los contenidos temáticos estudiados en la asignatura.

7.7. Examen Libre

Primera Etapa: Examen escrito sobre los aspectos prácticos fundamentales de la asignatura. Esta instancia se llevará a cabo mediante la resolución en computadora de un enunciado práctico que implicará la aplicación de los lenguajes de programación estudiados.

Segunda Etapa: Aprobar una evaluación de contenidos teóricos del programa analítico. Esta instancia se llevará a cabo utilizando la plataforma del CUV para responder un cuestionario multiple-choice sincrónico (escrito), o mediante una exposición respondiendo preguntas (oral).

Tercera Etapa: Presentación, prueba y defensa de un planteamiento práctico. El trabajo deberá ser presentado en soporte digital e impreso, cumplimentando las pautas establecidas en el enunciado.

Méndez Analía

.....
Profesor responsable de Asignatura