

	Asignatura: Programación II
	Carrera: Tecnicatura Universitaria en Programación (a distancia)
	Cursado: Segundo Trimestre Horas semanales: 8 hs
	Tipo: Troncal Horas semestrales: 128 hs
	Área: Disciplinas Troncales Nivel (Año):
	<input checked="" type="checkbox"/> 1° <input type="checkbox"/> 2° <input type="checkbox"/> <input type="checkbox"/> 4° <input type="checkbox"/> 5° <input type="checkbox"/> 6°

Programa de Cátedra

Integrantes de la Cátedra:

Profesor Coordinador: Carlos Martinez

Profesor a cargo comisión: Cinthia Rigoni

Ariel Enrerrel

Alberto Cortez

Fundamentación

La asignatura Programación II es una continuación de Programación I y se dicta en el segundo trimestre de la Tecnicatura Universitaria en Programación (modalidad a distancia). Su propósito es profundizar los conocimientos adquiridos previamente, enfocándose en la programación orientada a objetos (POO), el manejo de estructuras de datos avanzadas, el acceso a bases de datos relacionales y la integración con servicios web.

A lo largo del curso, los estudiantes fortalecerán su pensamiento lógico y algorítmico mediante la resolución de problemas complejos, utilizando estructuras de control avanzadas, funciones, procedimientos y técnicas para el manejo adecuado de errores. Se abordará el uso eficiente de estructuras de datos como arreglos y colecciones, permitiendo mejorar la organización, manipulación y gestión de la información.

Uno de los ejes centrales de la asignatura es el paradigma de programación orientada a objetos, lo que permitirá a los estudiantes desarrollar software modular, reutilizable y mantenable. Para ello, se trabajará con conceptos clave como clases, objetos,

encapsulamiento, herencia y polimorfismo, consolidando la capacidad de diseñar aplicaciones escalables y bien estructuradas.

Además, se abordará el acceso a bases de datos relacionales, enseñando a los estudiantes a realizar consultas, gestionar transacciones y optimizar la interacción con sistemas de almacenamiento de datos persistentes. También se explorará la integración de aplicaciones con servicios web y el consumo de APIs, proporcionando herramientas para la comunicación con sistemas externos y la construcción de aplicaciones interconectadas.

Descripción del espacio y propuesta general

Descripción del Espacio

La materia "Programación II" se desarrolla en un entorno virtual, dado que es parte de la modalidad a distancia de la Tecnicatura Universitaria de Programación. Este espacio virtual está diseñado para facilitar el aprendizaje autónomo y colaborativo, utilizando plataformas de e-learning que permiten el acceso a materiales didácticos, foros de discusión, y herramientas de evaluación en línea. Los estudiantes tendrán acceso a:

- Plataforma de Aprendizaje: Donde se encuentran los módulos de estudio, videos tutoriales, y ejercicios prácticos.
- Foros y Meets: Para la interacción con compañeros y profesores, resolución de dudas, y discusión de temas.
- Entornos de Desarrollo: Acceso a entornos de desarrollo integrados (IDEs) en línea o descargables para la práctica de programación.

Propuesta General

La asignatura **Programación II** profundiza los conocimientos adquiridos en **Programación I**, consolidando el pensamiento lógico y la capacidad de resolución de problemas a través de la **programación orientada a objetos (POO)**. Su enfoque permite a los estudiantes diseñar y desarrollar aplicaciones modulares, reutilizables y escalables, integrando estructuras de datos avanzadas y acceso a bases de datos.

La propuesta incluye:

- **Desarrollo del Pensamiento Algorítmico:** Aplicación de estructuras de control avanzadas y manejo eficiente de datos.
- **Fundamentos de la POO:** Introducción y aplicación de conceptos clave como clases, objetos, encapsulamiento, herencia y polimorfismo.
- **Entorno de Desarrollo Profesional:** Simulación de escenarios reales mediante proyectos prácticos y resolución de problemas del mundo laboral.

- **Interacción con Bases de Datos y Servicios Web:** Implementación de consultas, gestión de transacciones y consumo de APIs para el desarrollo de aplicaciones conectadas.

Esta propuesta busca no solo reforzar la lógica de programación, sino también fomentar la **capacidad de diseño y abstracción**, preparando a los estudiantes para desafíos más avanzados en el desarrollo de software.

Intencionalidades educativas (objetivos)

Generales

- Desarrollar la capacidad de aplicar los principios de la **programación orientada a objetos (POO)** para diseñar soluciones modulares y reutilizables.
- Conocer e integrar **estructuras de datos avanzadas**, bases de datos y servicios web en el desarrollo de aplicaciones.
- Promover **buenas prácticas de programación**, enfatizando la escritura de código limpio, mantenable y eficiente.

Específicos

1. **Profundizar en el Pensamiento Algorítmico:** Fortalecer la capacidad de resolver problemas complejos mediante el uso de estructuras de control avanzadas y el diseño eficiente de algoritmos.
2. **Comprender y Aplicar los Fundamentos de la POO:** Utilizar conceptos clave como **clases, objetos, encapsulamiento, herencia y polimorfismo** en la construcción de software.
3. **Implementar Soluciones con Manejo de Datos:** Desarrollar aplicaciones que integren estructuras de datos dinámicas, bases de datos relacionales y consumo de APIs.
4. **Simular un Entorno de Desarrollo Profesional:** Aplicar conocimientos en **proyectos prácticos** que reflejen escenarios reales del mundo del desarrollo de software.

Resultados de aprendizaje

RA1: Analiza problemas y define modelos de solución aplicando los principios de la **programación orientada a objetos (POO)**, asegurando un diseño modular y escalable.

RA2: Desarrolla soluciones implementando estructuras de datos avanzadas y utilizando **clases, objetos, encapsulamiento, herencia y polimorfismo** para mejorar la organización del código.

RA3: Integra bases de datos en sus aplicaciones mediante consultas y transacciones, garantizando un acceso y manipulación eficiente de la información.

RA4: Implementa y verifica aplicaciones conectadas a **servicios web y APIs**, completando el proceso de desarrollo de software con prácticas de integración y prueba.

Competencias específicas

CE1.3: Especificar, proyectar y desarrollar software para la elaboración de soluciones informáticas con el propósito de resolver problemas estratégicos y operativos, así como de servicios y de negocios, en el marco de una actividad económica que sea social y ambientalmente sustentable.

Recursos Digitales

El curso utilizará la plataforma **Moodle** como entorno principal para la gestión de contenidos, actividades y seguimiento académico. Se guiará a los estudiantes en la configuración de un **entorno de desarrollo adecuado** para la implementación de soluciones basadas en **programación orientada a objetos (POO)**, asegurando un diseño modular y estructurado.

Para reforzar los conceptos clave, se dispondrá de **videos tutoriales** de producción propia y seleccionados de fuentes especializadas, abordando temas como **clases, herencia, polimorfismo y manejo de bases de datos**.

Asimismo, se integrarán **herramientas interactivas como Genially**, utilizadas en cada unidad para el diseño de **presentaciones, infografías y recursos interactivos**, fomentando un aprendizaje visual e intuitivo.

Esta combinación de recursos busca proporcionar una **experiencia de aprendizaje dinámica y enriquecedora**, promoviendo la aplicación práctica de los conceptos en escenarios reales del desarrollo de software.

Plan de Tutoría y Acompañamiento

El objetivo es brindar un soporte integral a los estudiantes a lo largo de su proceso de aprendizaje. Para ello, el tutor virtual jugará un papel fundamental, estando disponible para resolver dudas, ofrecer apoyo técnico y pedagógico, y realizar un seguimiento personalizado del progreso de cada estudiante.

Para interactuar y estar en contacto durante toda la cursada se pone a disposición:

- Se propone una clase de consulta sincrónica de una hora de duración para resolver dudas surgidas durante la semana o en el desarrollo de las actividades.
- Se propone la realización de un trabajo práctico. Este tendrá como objetivo aplicar los conceptos estudiados.
- Al finalizar el módulo, se pondrá a disposición de los estudiantes un documento con los ejercicios resueltos para facilitar la autoevaluación.
- La actividad final de la unidad consistirá en un cuestionario de autoevaluación.

- El foro Asíncrono “Avisos”: un espacio donde se publicarán los comunicados y recordatorios importantes relacionados con la unidad y el desarrollo del curso.
- El foro Asíncrono "Dudas y Consultas": un espacio donde podremos realizar todas las consultas respecto de los temas de la clase. Las consultas serán contestadas en un plazo máximo de 48hs.
- La mensajería de la plataforma es sólo para temas administrativos, no para temas académicos.

Consideraciones de Accesibilidad

En cuanto a las consideraciones de accesibilidad, se implementarán medidas para garantizar que todos los estudiantes puedan acceder al contenido del curso sin dificultades. Todos los videos explicativos contarán con subtítulos, facilitando el acceso a estudiantes con discapacidad auditiva. Además, los materiales del curso se proporcionarán en formatos compatibles con lectores de pantalla, como archivos PDF, para asegurar que la información sea fácilmente accesible y comprensible.

Asimismo, se ofrecerá flexibilidad en los plazos de entrega para aquellos estudiantes que lo requieran, previa justificación, con el objetivo de adaptar las evaluaciones a sus circunstancias particulares. Estas medidas buscan promover una experiencia educativa inclusiva y equitativa para todos.

Contenidos

Módulo 1: Introducción a Java

Objetivo: Familiarizarse con el lenguaje Java, su entorno de desarrollo y su sintaxis básica, permitiendo a los estudiantes instalar, configurar y ejecutar sus primeros programas de manera efectiva.

- Introducción a Java: historia, características y plataforma.
- Bytecode y Máquina Virtual Java (JVM).
- Instalación y configuración del entorno de desarrollo (JDK, NetBeans, Eclipse).
- Estructura básica de un programa en Java.
- Tipos de datos, variables y constantes.
- Operadores y expresiones.
- Entrada y salida básica (Scanner, System.out).
- Modo oscuro en NetBeans y configuración del editor.

Módulo 2: Programación Estructurada

Objetivo: Desarrollar la capacidad de construir programas utilizando estructuras de control y funciones, aplicando principios de la programación estructurada para mejorar la claridad y eficiencia del código.

- Estructuras de control:
 - Secuenciales.

- Condicionales (if, else, switch).
- Repetitivas (for, while, do-while).
- Funciones y procedimientos:
 - Definición y llamada.
 - Parámetros y retorno de valores.
 - Ámbito de variables (locales y globales).
- Manejo de errores básicos.

Módulo 3: Introducción a la Programación Orientada a Objetos (POO)

Objetivo: Comprender los fundamentos de la Programación Orientada a Objetos, incluyendo clases, objetos, atributos y métodos, para estructurar programas de manera modular y reutilizable en Java.

- Paradigmas de programación: comparación entre programación estructurada y POO.
- Conceptos básicos de POO: clases, objetos, atributos y métodos.
- Diferencia entre clase y objeto.
- Estado de un objeto: atributos y valores iniciales.
- Comportamiento de un objeto: Métodos en Java
- Identidad de un objeto
- Encapsulamiento: ocultamiento de detalles de implementación.
- Modificadores de acceso: public, private, protected.
- Getters y setters: acceso controlado a atributos.

Módulo 4: Programación Orientada a Objetos

Objetivo: Comprender y aplicar el uso de this, constructores, sobrecarga de métodos, métodos estáticos, diagramas de clases en UML y relaciones entre objetos (asociación, composición y agregación) para el diseño modular en Java.

- Uso de this para referenciar atributos y métodos de la clase.
- Constructores: definición y uso.
- Sobrecarga de constructores y métodos.
- Métodos estáticos (static).
- Introducción a UML: diagramas de clases.
- Relaciones entre objetos: asociación, composición y agregación.

Módulo 5: UML Basico

- ¿Qué es UML?
- Primer diagrama con UMLetino
- Relaciones 1 a 1 en Clases UML
- Asociacion 1 a 1 Unidireccional en Clases UML

- Agregacion 1 a 1 en Clases UML
- Composicion 1 a 1 en Clases UML
- Asociacion Bidireccional 1 a 1 en Clases UML
- Uso o Dependencia en Java: Código y UML
- Dependencia de Uso
- Dependencia de Creación
- Alta cohesion y Bajo Acoplamiento

Módulo 6: Colecciones

Objetivo: Desarrollar estructuras de datos dinámicas en Java mediante el uso de colecciones y enumeraciones, optimizando su manipulación y recorrido.

- Arreglos unidimensionales y multidimensionales.
- Colecciones: ArrayList, métodos útiles.
- Relaciones 1 a N en Clases UML
- Atributos tipo colección en Java
- Ciclo for-each para recorrer colecciones.
- Búsqueda y filtrado en colecciones.
- Enumeraciones

Módulo 7: Herencia y Polimorfismo

Objetivo: Comprender y aplicar los principios de herencia y polimorfismo en Java, permitiendo la creación de jerarquías de clases y la reutilización de código mediante la sobreescritura de métodos y la implementación de interfaces genéricas.

- Herencia: clases base y derivadas.
- Sobreescritura de métodos.
- Polimorfismo: métodos con múltiples formas.
- Uso de super para acceder a métodos de la clase base.
- Clases abstractas y métodos abstractos.
- Interfaces: definición e implementación.
- Relaciones entre interfaces y clases.
- Uso de Generics en interfaces
- Implementación de interfaces genéricas en clases concretas
- Polimorfismo con Generics
-

Módulo 8: Interfaces y Excepciones

Objetivo: Desarrollar competencias avanzadas en Programación Orientada a Objetos (POO) en Java, comprendiendo y aplicando la herencia múltiple, el uso de interfaces y el manejo de excepciones para diseñar código flexible, extensible y robusto.

- Herencia Múltiple y su Problemática
- Uso de Interfaces en Java
- Manejo de Excepciones en Java
- Técnicas Avanzadas de Manejo de Excepciones

Módulo 9: Genéricos

Objetivo: Desarrollar habilidades en el uso de Genéricos en Java para mejorar la seguridad, reutilización y escalabilidad del código. Comprender la implementación de clases, métodos e interfaces genéricas en estructuras de datos dinámicas. Aplicar comodines (? , extends, super) para gestionar diferentes tipos de datos en colecciones. Utilizar Comparable y Comparator para ordenar y buscar elementos en colecciones de manera flexible. Integrar Genéricos en el diseño modular del software, optimizando su organización y mantenimiento.

- Introducción a los Genéricos en Java
- Creación de Clases y Métodos Genéricos
- Uso de Comodines en Genéricos
- Interfaces Genéricas
- Ordenamiento y búsqueda en colecciones con Genéricos
-

Módulo 10: Acceso a Datos con JDBC

Objetivo: Desarrollar habilidades para conectar aplicaciones Java con bases de datos relacionales mediante JDBC, ejecutando consultas SQL y manejando resultados de forma eficiente, garantizando el manejo adecuado de transacciones y excepciones.

- Introducción a JDBC (Java Database Connectivity).
- Conexión a bases de datos relacionales.
- Ejecución de consultas SQL (SELECT, INSERT, UPDATE, DELETE).
- Manejo de resultados (ResultSet).
- Transacciones y manejo de excepciones en JDBC.

Estrategia de Evaluación

Actividades en el aula

Los estudiantes deberán recorrer cada Unidad respetando la Hoja de Ruta propuesta y las indicaciones de Docentes y Tutores, donde se establecerán actividades obligatorias y opcionales que representan la secuencia didáctica de esa semana.

Las actividades de autoevaluación como lecciones, cuestionarios, etc., que se presentarán semanalmente como aplicación de los temas desarrollados a través de videos,

apuntes de cátedra y otros recursos, deberán ser completadas en su totalidad. Se considera como válida la obtención de una calificación igual o superior al 60%, estas actividades podrán ser repetidas para que el alumno llegue a la calificación indicada en caso de ser necesario, manteniendo la nota más alta obtenida.

Los juegos, salas de escape y otros espacios de esparcimiento serán de cumplimiento obligatorio sin evaluación.

Exámenes Parciales

Se tomarán al menos dos evaluaciones parciales, una al promediar el cursado, y la segunda antes de finalizarlo.

Para poder acceder al examen parcial, el alumno debe haber completado todas las actividades correspondientes a las unidades o módulos que se evalúen en dicho examen. Se configurará el examen para que solo esté disponible si el alumno ha completado con un mínimo de calificación del 60% en cada una de las actividades de autoevaluación o Trabajos Prácticos que el docente haya propuesto para las unidades a evaluar.

El examen se tomará a través del aula virtual en una fecha que se indicará al principio del cursado y dentro de una franja horaria, tal que se puedan conectar los alumnos, más allá de sus obligaciones personales.

De acuerdo al Reglamento de Estudio, los exámenes parciales deberán aprobarse con una calificación igual o superior al 60%.

Recuperatorios

Se habilitará una semana previa a la finalización del cursado, para que el alumno recupere las actividades de autoevaluación, trabajos prácticos y exámenes parciales.

Trabajo Práctico Integrador

Se deberá realizar un Trabajo Práctico Integrador, de carácter obligatorio, que servirá como cierre del cursado de la asignatura.

Podrán realizar este trabajo aquellos estudiantes que hayan obtenido al menos un 40% de calificación en ambos exámenes parciales o en sus respectivos recuperatorios.

El Trabajo Práctico Integrador se aprueba con una calificación mínima del 60%.

Su aprobación es un requisito indispensable tanto para promocionar la materia (es decir, aprobarla sin rendir examen final) como para acceder al examen final en caso de haber alcanzado solo la condición de regular.

Aquellos estudiantes que no alcancen el mínimo del 40% en los parciales (ni en sus recuperatorios) no podrán realizar el Trabajo Práctico Integrador y quedarán en condición de alumno libre.

Condiciones de Aprobación

Condiciones para obtener la APROBACIÓN DIRECTA:

- ✓ Completar las auto-evaluaciones del aula virtual
- ✓ Rendir los 2 (dos) parciales teórico – prácticos y aprobarlos con una calificación de 60 % o más, en la primera instancia o en su instancia recuperatoria.
- ✓ Rendir y aprobar el examen integrador con un puntaje igual o superior a 60 % en la primera instancia o en su instancia recuperatoria.

La Nota final se obtiene de la siguiente forma:

$$NF = [(Parcial1 + Parcial2) / 2] * 0,4 + Nota Examen Global Integrador * 0,6$$

Condiciones para obtener el CURSADO APROBADO:

- ✓ Completar las auto-evaluaciones del aula virtual.
- ✓ Rendir los 2 (dos) parciales teórico – prácticos y obtener una calificación de 40% o más, en la primera instancia o en su instancia recuperatoria.

Instancias de recuperación

Se implementará al menos una (1) instancia recuperatoria para todas las evaluaciones.

Si algún estudiante no aprueba el Trabajo Práctico Integrador o su recuperatorio no perderá la condición de Cursado Aprobado antes obtenida. Pero deberá entregar y aprobar dicho trabajo para poder presentar el Examen Final de la materia.

Si algún estudiante no logra aprobar los dos parciales (o sus recuperatorios), se puede tomar un segundo recuperatorio, sólo para lograr el Cursado Aprobado.

Si algún estudiante no cumple con las condiciones de Cursado Aprobado, quedará en condición de ALUMNO LIBRE por notas o ALUMNO LIBRE por inasistencia a los exámenes respectivamente, y deberá RECURSAR la materia.

Es requisito para presentarse a la mesa de examen final tener el trabajo integrador aprobado.

Fechas y turnos para encuentros sincrónicos

Los encuentros se realizarán en alguno de los siguientes turnos:

Mañana: de 9:00 a 12:00 hs

Tarde: de 14:00 a 17:00 hs

Noche: de 18:00 a 21:00 hs

Las fechas exactas serán publicadas en el aula virtual.

Importante

El trabajo en parejas es obligatorio. Cualquier situación excepcional deberá ser justificada ante la cátedra.

Toda comunicación oficial se realizará exclusivamente a través de la plataforma institucional y del correo electrónico.

Modalidad de Examen Final

Evaluación teórica práctica de:

- ✓ Trabajos prácticos Individuales.
- ✓ Participación en las mesas de trabajo.
- ✓ Temas teóricos tratados durante el cursado.

Acceden a la instancia de Examen Final los alumnos que hubiesen obtenido la condición de Cursado Aprobado. El Examen Final debe ser aprobado con una calificación igual o superior al 60%.

Fechas y comunicación oficial

Las **mesas de examen** se llevarán a cabo en **fechas y horarios específicos**, que serán publicados con anticipación en el **aula virtual**. La **devolución de resultados** y las **citas para el coloquio** se comunicarán oficialmente por **correo electrónico institucional**.

Importante

El **incumplimiento de los plazos** de entrega del Trabajo Integrador implicará la **postergación del examen** para la siguiente mesa disponible. Toda comunicación oficial se realizará exclusivamente a través de los **canales institucionales**: plataforma virtual y correo electrónico.

Retroalimentación

A lo largo de todo el proceso evaluativo, se proporcionará retroalimentación personalizada a los estudiantes con el objetivo de identificar sus fortalezas y áreas de mejora, y así guiar su aprendizaje de manera efectiva.

ESTRATEGIAS DEL DOCENTE:

1. **Integración de recursos didácticos audiovisuales**
2. **Uso de herramientas digitales para la comunicación y el intercambio de información:**
Implementación de plataformas de mensajería, correo electrónico y entornos colaborativos para mantener una comunicación fluida y constante con los estudiantes.
3. **Aprovechamiento del aula virtual para el aprendizaje colaborativo:** Fomento de la interacción a través de foros de discusión, evaluaciones en línea, actividades prácticas y entrega de trabajos, promoviendo la construcción colectiva del conocimiento.
4. **Enfoque de Aprendizaje Basado en Problemas (ABP):** Planteamiento de situaciones reales o simuladas que desafíen a los estudiantes a investigar, analizar y desarrollar soluciones mediante la aplicación de los conceptos aprendidos.
5. **Uso de preguntas insertas para estimular el pensamiento crítico:** Incorporación de cuestionamientos estratégicos dentro de las actividades y materiales didácticos para guiar la reflexión, la autoevaluación y el aprendizaje significativo.

Semana	Tema	Recursos
Semana 1 Introducción a Java	Módulo 1: - Actividades: - Prácticos: - Autoevaluación: Verificamos nuestro progreso.	Materiales y Recursos Apuntes y bibliografía proporcionada por el equipo docente. Videos explicativos.
Semana 2 Programación Estructurada	Módulo 2: - Actividades - Actividad Lúdica: - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso.	Materiales y Recursos Apuntes y bibliografía proporcionada por el equipo docente. Videos explicativos.

Semana 3 Introducción a la Programación Orientada a Objetos (POO)	Módulo 3: -Actividades: 1_ Estudiar videos sobre: Introducción a la OO. Clase y Objetos. 2_ Estudiar videos sobre: Atributos, objetos e identidad de un objeto 3_ Estudiar videos sobre: Encapsulamiento en OO Resolver el cuestionario correspondiente a cada actividad. - Prácticos: Aplicar los conocimientos adquiridos mediante la resolución de ejercicios prácticos.. - Autoevaluación: Verificar el progreso y comprensión de los temas abordados.	Materiales y Recursos Apuntes y bibliografía proporcionada por el equipo docente. Videos explicativos.
Semana 4 y Semana 5 Programación Orientada a Objetos	Módulo 4: -Actividades: 1_ Estudiar videos sobre: Uso de this, constructores y sobrecarga de constructores y métodos en Java. 2_ Estudiar videos sobre: Métodos toString, uso de static, introducción a UML y diagramas de clases UML. 3_ Estudiar videos sobre: Relaciones entre clases en Java, atributos de tipo objeto, relaciones uno a uno en UML y principios de encapsulamiento. 4_ Estudiar videos sobre: Resolución paso a paso de un ejercicio práctico de Programación Orientada a Objetos. Resolver el cuestionario correspondiente a cada actividad. - Prácticos: Aplicar los conocimientos adquiridos mediante la resolución de ejercicios prácticos..	Materiales y Recursos Apuntes y bibliografía proporcionada por el equipo docente. Videos explicativos.

	<ul style="list-style-type: none"> - Autoevaluación: Verificar el progreso y comprensión de los temas abordados. 	
Semana 6 y Semana 7 UML	<p>Módulo 5:</p> <ul style="list-style-type: none"> - Actividades: - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	<p>Materiales y Recursos</p> <p>Apuntes y bibliografía proporcionada por el equipo docente.</p> <p>Videos explicativos.</p>
PARCIAL 1 (módulos 1 al 5)	<ul style="list-style-type: none"> - Autoevaluación: Es momento de verificar nuestro progreso. 	
Semana 8 Colecciones	<p>Módulo 6:</p> <ul style="list-style-type: none"> - Actividades: - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	<p>Materiales y Recursos</p> <p>Apuntes y bibliografía proporcionada por el equipo docente.</p> <p>Videos explicativos.</p>
Semana 9 y Semana 10 Herencia y Polimorfismo	<p>Módulo 7:</p> <ul style="list-style-type: none"> - Actividades: - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	<p>Materiales y Recursos</p> <p>Apuntes y bibliografía proporcionada por el equipo docente.</p> <p>Videos explicativos.</p>
RECUPERATORIO PARCIAL 1 (módulos 1 al 5)		
Semana 11 Interfaces y Excepciones	<p>Módulo 8</p> <ul style="list-style-type: none"> - Actividades: - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	<p>Materiales y Recursos</p> <p>Apuntes y bibliografía proporcionada por el equipo docente.</p> <p>Videos explicativos.</p>
Inicio Integrador		

Semana 12 Genéricos PARCIAL 2 (módulos 6 al 8)	Módulo 9: - Actividades: - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso.	Materiales y Recursos Apuntes y bibliografía proporcionada por el equipo docente. Videos explicativos.
Semana 13 y Semana 14 Acceso a Datos con JDBC	Módulo 10: -- Actividades: - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso.	Materiales y Recursos Apuntes y bibliografía proporcionada por el equipo docente. Videos explicativos.
Semana 15 Acceso a Datos con JDBC	Módulo 10: -- Actividades: - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso.	Materiales y Recursos Apuntes y bibliografía proporcionada por el equipo docente. Videos explicativos.
Semana 16 RECUPERATORIO PARCIAL 2 (módulos 6 al 8) INTEGRADOR	TP integrador	Materiales y Recursos -Cuestionario de evaluación generado en la plataforma Moodle.
RECUPERATORIO INTEGRADOR	Examen integrador En la primera mesa de examen	Materiales y Recursos -Cuestionario de evaluación generado en la plataforma Moodle.

Bibliografía y Recursos

Libros

Deitel, P. J., & Deitel, H. M. (2016). *Cómo programar en Java* (10a ed.). Pearson Educación. ISBN: 978-607-32-3802-1.

file:///home/mesa/Descargas/Como%20programar%20a%20Java,%2010%C2%AA%20Edici%C3%B3n%20-%20Paul%20Deitel.pdf

Videos

Cimino, C. (s. f.). OBJETOS (POO) desde -10 en Java [Lista de reproducción]. YouTube. Recuperado el 14 de febrero de 2025, de <https://www.youtube.com/watch?v=voMOPqtnJto&list=PLOw7b-NX043aSC7ZNtEuVfY8xZoNzVqdJ>