



000027

Programación con Objetos 1

INSTITUTOS/S: INSTITUTO DE TECNOLOGÍA E INGENIERÍA

CARRERA/S: Licenciatura en Informática

RESPONSABLE DE LA ASIGNATURA Y EQUIPO DOCENTE:

Lucas Spigariol,
Hernán Coniglio

AÑO: 2025.

CRÉDITOS: 7

CARGA HORARIA DE INTERACCIÓN PEDAGÓGICA: 96

CARGA HORARIA TOTAL: 175

CÓDIGO DE LA MATERIA EN SIU: 753

000027

1. Fundamentación

La materia Programación con objetos I pertenece al área Algoritmos y Lenguajes y se dicta cuatrimestralmente en el primer año según el plan de estudios vigente, además tiene como correlativas precedentes a la asignatura Introducción a la Programación, y como espacio curricular subsiguiente a la asignatura Programación con objetos II.

La materia reviste vital importancia para el/la estudiante ya que en los entornos dinámicos y cambiantes donde participan sistemas informáticos los problemas requieren soluciones de sistemas de información que pueden ser de muy variado tipo y con características muy distintas unos de otros. Es indispensable que el/la estudiante comprenda y conozca el enfoque de la programación con objetos y que pueda aplicarlo en la solución de los problemas.

2. Propósitos y/u objetivos

- *Introducir formalmente a los/as estudiantes en la programación con objetos, que conozcan y comprendan sus conceptos fundamentales y ventajas.*
- *Brindar las herramientas necesarias para que los/as estudiantes adquieran destreza modelando y desarrollando prácticas concretas de complejidad simple, moderada y avanzada.*
- *Instalar la noción de cultura colaborativa como fundamento de la comunidad UNA HUR.*

3. Contenidos mínimos

Conceptos fundantes del paradigma: objeto y mensaje. Concepto de polimorfismo en objetos, comprensión de las ventajas de aprovecharlo. Protocolo/interfaz, concepto de tipo en objetos, comprensión de que un objeto puede asumir distintos tipos. La interfaz como contrato al que se compromete el objeto modelado, posibilidad de reforzar ese contrato. Estado en el paradigma de objetos: referencias, conocimiento, estado interno. Métodos, clases, herencia simple, method lookup. Conceptos de responsabilidad y delegación. Colecciones: conceptualización como objetos, caracterización a partir de los conceptos de

000027


protocolo y responsabilidad, protocolo, acceso a sus elementos. Testeo automático y repetible. Nociones básicas sobre manejo de errores. Interrupción del flujo de ejecución: modelado mediante estructuras de control, concepto de excepción.

4. Carga horaria

Créditos	Interacción pedagógica		Trabajo Autónomo	Total
7	Teoría: 32	Práctica: 64	79	175

4.1 Trabajo autónomo del/la estudiante

Actividad	Carga horaria
<i>Lectura de contenidos teóricos apuntes o bibliografía.</i>	14
<i>Investigar nuevos conceptos.</i>	5
<i>Desarrollo de los ejercicios prácticos propuestos de codificación.</i>	20
<i>Uso de plataforma práctica.</i>	20
<i>Realización de mini-proyectos.</i>	20

5. Programa analítico

UNIDAD 1: Programa, objeto y mensaje.

Definición de programa como objetos que se comunican entre ellos. Objetos como representaciones computacionales de entes que exhiben un comportamiento. Definición de un objeto. Noción de estado interno para recordar valores, concepto de atributo. Noción de mensaje como punto de acceso al comportamiento exhibido por los objetos. Conocimiento elemental de método como código que se ejecuta cuando un objeto recibe un mensaje. Comprobación del comportamiento y estado interno de los objetos.

UNIDAD 2: Referencias e interacción entre objetos, polimorfismo.

Relaciones entre objetos que conforman un modelo / programa. Referencias entre objetos como parte del estado interno de los mismos. Necesidad de interacción entre los objetos basadas en el comportamiento. Obtener información de un objeto y cambiar el estado

000027

interno. Polimorfismo. Explicación del polimorfismo modelando situaciones donde aparecen distintos objetos con los que un tercero puede interactuar indistintamente sin que necesite saber si con quien está interactuando. Igualdad e Identidad, diferencia.

UNIDAD 3: Colecciones y clousures.

Colección como un objeto que agrupa y mantiene referencias a una cantidad indeterminada de otros objetos. Listas y conjuntos, definición y diferencias entre ambos. Comportamiento esperado. Noción de Clousure como porción de código que puede convertirse en un objeto y que puede diferir su ejecución.

Mensajes básicos que entienden las colecciones. Mensajes sin y con efecto. Comandos, condiciones, transformaciones y otros mensajes útiles sin y con efecto. Mensajes que solo entienden las listas sin y con efecto. Mensajes que solo entienden los conjuntos sin y con efecto.

UNIDAD 4: Clases.

Motivado por la necesidad de clasificar los objetos, agrupándolos en función a sus semejanzas. Concepto de clase como patrón o modelo para crear objetos que describe completa y detalladamente la estructura de información y comportamiento que tendrá todo objeto de la clase. Instanciación como creación de objetos a partir de una clase. Constructores de clase. Pseudovariable self.

UNIDAD 5: Herencia simple.

Definición de nuevas clases basadas en clases existentes. Superclase y subclase que permite agregar nuevas variables y métodos y modificar los métodos heredados. Mecanismo de búsqueda con y sin herencia. Redefinición de métodos. Pseudovariable super. Herencia como herramienta que potenciar y ampliar el polimorfismo. Noción de Clase Abstracta. Template Method.

UNIDAD 6: Prueba unitaria.

Importancia de las pruebas unitarias en la calidad del desarrollo de programas. Utilización como documentación del programa. Elementos que conforman una prueba: objetos de prueba, configuración de escenarios, aserciones.

UNIDAD 7: Errores y excepciones.

000027



Errores básicos del tipo objetos que no entienden mensajes y no permiten la relación entre objetos. Errores relacionados con las condiciones y la posibilidad de control de estos. Definición de excepción como evento que altera el flujo normal de mensajes entre los objetos. Noción de propagar la excepción hacia arriba.

6. Ejes y enunciados Multidimensionales y transversales

Eje	Nivel de logro de Aprendizaje	Acciones de enseñanza
C1. Identificación, formulación y resolución de problemas de informática	Alto	Resolución de ejercicios y problemas: Permite a los estudiantes desarrollar habilidades prácticas para identificar y resolver problemas concretos. Aprendizaje Basado en Problemas (ABP): Fomenta la resolución de problemas reales, permitiendo aplicar conocimientos teóricos.
C2. Concepción, diseño y desarrollo de proyectos de informática	Alto	Aprendizaje orientado a Proyectos: Ayuda a los estudiantes a desarrollar un proyecto de informática desde la planificación hasta la ejecución. Clases prácticas: Permiten aplicar el diseño de proyectos en situaciones reales o simuladas.
C3. Gestión, planificación, ejecución y control de proyectos de informática	Bajo	Método de Casos: Ayuda a los estudiantes a enfrentarse a problemas reales de gestión y planificación de proyectos. Clases expositivas dialogadas: Para exponer y discutir conceptos clave sobre gestión de proyectos.
C4. Utilización de técnicas y herramientas de aplicación en la informática	Alto	Clases prácticas: Los estudiantes pueden practicar el uso de herramientas informáticas en escenarios controlados. Ilustraciones: Pueden utilizarse para representar visualmente conceptos complejos relacionados con las herramientas técnicas.

000027



C5. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas	Alto	Aprendizaje orientado a Proyectos: Fomenta el desarrollo de soluciones innovadoras a problemas reales. Lluvia de ideas: Para fomentar la generación de ideas creativas y originales en el ámbito tecnológico.
C6. Fundamentos para el desempeño en equipos de trabajo	Bajo	Aprendizaje cooperativo: Promueve el trabajo en equipo, la interacción y la responsabilidad compartida entre los estudiantes. Tutoría: Fomenta el apoyo grupal e individual para mejorar el trabajo en equipo.
C7. Fundamentos para la comunicación efectiva	No aporta	.
C8. Fundamentos para la acción ética y responsable	No aporta	.
C9. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local	No aporta	.
C10. Fundamentos para el aprendizaje continuo	Bajo	Aula invertida: Los estudiantes pueden desarrollar autonomía en su aprendizaje al investigar por su cuenta fuera del aula. Tutoría: Facilita el desarrollo de un plan de aprendizaje individual a lo largo del tiempo.
C11. Fundamentos para la acción emprendedora	No aporta	.

7. Bibliografía y recursos

7.1 Bibliografía obligatoria

Barnes, D., Kölling, M. (2017). "Programación Orientada a Objetos con Java usando Blue", 6ta edición. Madrid, España: Pearson.

000027

Barnes, D., Kölling, M. (2017). *"Programación Orientada a Objetos con Java: una introducción práctica usando Blue"*. Madrid, España: Pearson.

Juárez Andrés (2019) *"Java + programación orientada a objetos - Curso teórico -práctico"*, -1ra Ed.- Editorial EUDEBA.

7.2 Bibliografía optativa

Fontenla, C. y Páez, N. (2015). *"Programación Orientada a Objetos"*. Recuperado de <https://legacy.gitbook.com/book/nicopaez/poo/details>.

Izquierdo, L.R. *"Introducción a la Programación Orientada a Objetos"*. Recuperado de <http://luis.izqui.org/resources/ProgOrientadaObjetos.pdf>.

Gamma, E. (2002). *"Patrones de diseño"*. (C. Fernández Acebal, Trad.). Pearson Educación.

7.3 Recursos

Guías de ejercicios y material adicional generado por el grupo docente disponible en página de la materia: <https://obj1-unahur.github.io/material/>

Página de la Materia: <https://obj1-unahur.github.io>

Wollok: <http://www.wollok.org>

Se trata de un lenguaje de programación acompañado de un entorno integrado de programación, desarrollado en Argentina por docentes de distintas universidades públicas.

Wollok combina un enfoque pedagógico orientado a estudiantes que están haciendo sus primeras armas en programación orientada a objetos, y con poca experiencia en programación. Al mismo tiempo, las características del lenguaje y del entorno de programación son similares a las de lenguajes y herramientas ampliamente utilizados en la industria, como ser los lenguajes Java y JavaScript, y el entorno Visual Studio Code y/o Eclipse.

000027

8. Metodología de enseñanza

8.1 Modalidades u opciones pedagógicas

La asignatura está constituida por 7 unidades temáticas, las cuales se dictarán durante un cuatrimestre en clases teóricas y prácticas, 50% en formato presencial y 50% en formato virtual de clase sincrónica, con resolución de problemas a cargo de los/las alumnos/as supervisados por los/las docentes. Las clases están organizadas para integrar los contenidos teóricos en la solución de problemas prácticos presentados a los/las estudiantes. El nivel de complejidad de los problemas aumenta en función a la presentación de los contenidos teóricos de cada unidad.

El objetivo de esta metodología busca que el saber de los conceptos introducidos en cada clase se refleje en el saber y hacer práctico de la solución de un problema planteado. Se presentan los conceptos teóricos, se complementa con videos teórico/práctico y lecturas adicionales recomendadas, y se verifica en el hacer práctico que se apliquen los conceptos presentados.

Cada problema es presentado y discutido teóricamente, y luego es resuelto por los/las estudiantes directamente mediante la implementación de programas. Esta estrategia unifica los conceptos, el modelado y solución del problema, y la implementación como clase de laboratorio.

La carga horaria de la materia permite la discusión teórica incentivando el espíritu de colaboración e investigación, como así también la implementación de las soluciones modeladas en las computadoras, donde cada estudiante trabaja con la asistencia de los/las docentes y de sus compañeros/as.

8.2 Formación práctica

La organización propuesta para la cursada está fuertemente basada en la existencia de una serie de guías de ejercicios, que están graduados de forma de acompañar la evolución de los/las estudiantes y la secuencia pedagógica propuesta.

9. Evaluación y régimen de aprobación

9.1 Modalidad de evaluación

000027

El sistema normal de evaluación consistirá en 2 Exámenes Parciales con un recuperatorio por cada uno de ellos, según el cronograma previsto. Tanto en los parciales como en los recuperatorios, los contenidos a evaluar representan la totalidad de la materia descrita en el programa. Los mismos se realizarán en las fechas que, a tal efecto, se establezcan en el cronograma correspondiente.

9.2 Aprobación de la cursada

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

9.3 Acreditación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: *tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.*

Evaluación integradora: *tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado lo cursado con una nota de entre cuatro (4) y seis (6) puntos.*

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrada, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

000027

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.