



INSTITUTO/S: Tecnología e Ingeniería

CARRERA/S: Licenciatura en Informática

MATERIA: Programación con Objetos II

NOMBRE DEL RESPONSABLE DE LA ASIGNATURA: Andrés Juárez

EQUIPO DOCENTE: Cristian Ciarallo

CUATRIMESTRE: 2^{do}

AÑO: 2^{do}

PROGRAMA N°: 10

(Aprob. Por Cons.Directivo fecha XX)

Instituto/s: Tecnología e Ingeniería

Carrera/s: Licenciatura en Informática

Nombre de la materia: Programación con Objetos II

Responsable de la asignatura y equipo docente: Andrés Juárez, Cristian Ciarallo

Cuatrimestre y año: 1^{er} cuat. del 2^{do} año

Carga horaria semanal: 6 hs

Programa N°: 10

Código de la materia en SIU: 765

Programación con Objetos II

1. Fundamentación

En esta materia, perteneciente al área de Algoritmos y Programación, los estudiantes continúan formándose en la Programación Orientada a Objetos, comenzado a estudiarse en la materia precedente, pero ahondando en algunas características que les serán de utilidad para la vida profesional, como ser el Diseño, el lenguaje UML, el Desarrollo Dirigido mediante Pruebas, los Patrones de Diseño, las Excepciones, etcétera. También, hacemos hincapié en las buenas prácticas de programación, que incluyen nombres de atributos, constantes, métodos, clases y objetos, entre otras cuestiones.

2. Propósitos y/u objetivos

Propósitos

- Profundizar en la Programación Orientada a Objetos que comenzó a verse en la materia precedente.
- Análisis y Diseño Orientado a Objetos. UML.
- Adquieran conocimiento y manejo de herramientas para realizar pruebas unitarias y Desarrollo Guiado por Pruebas (TDD).
- Introducir a los Patrones de Diseño. Conocimiento e implementación de algunos.
- Manejo de excepciones.

3. Programa sintético:

Aproximación al diseño de software. Noción de decisión de diseño, el diseño como proceso de toma de decisiones. Conceptos de acoplamiento y cohesión. Problemas que derivan de un grado de acoplamiento inadecuado. Vinculación entre las ideas básicas de diseño y el paradigma de objetos. Características deseadas en un diseño de objetos. Patrones de diseño. Nociones sobre

proceso de diseño. Eventos. Meta programación. Uso de un entorno integrado de software. Notación UML de los diagramas de clases, de objetos y de secuencia. Testeo unitario y automático. Manejo de errores, impacto del manejo de errores en el diseño.

4. Programa analítico

4.1 Organización del contenido:

Lenguaje Java

Características del lenguaje. Máquina virtual. IDEs. Sintaxis básica. Estructuras secuenciales, condicionales y repetitivas. Clases. Objetos. Polimorfismo. Entrada y salida. Clases parametrizadas.

UML

Diseño. Diagramas de clase. Diagramas de relación de clases. Diagramas de secuencia e interacción. Casos de uso. User stories.

Pruebas Unitarias – TDD

Test Unitarios. Instalación de paquetes. Configuración. TDD: Test Development Driven. Filosofía. Implementación y ejercitación de ejemplos varios.

Patrones de diseño

Definición. Utilidad. Clasificación. Patrones de creación: Factory. Builder. Prototype. Singleton. Patrones estructurales: Adapter. Bridge. Composite. Decorator. Patrones de Comportamiento: Observer. Template. Strategy. Iterator. Implementación y uso de al menos uno de cada tipo.

Manejo de excepciones

Excepciones. Manejo de excepciones por parte del usuario. Creación de clases de manejo de excepciones. La clase Exception.

4.2 Bibliografía y recursos obligatorios:

- **Wirfs-Brock, Rebecca y Wilkerson.** (1990). *Designing Object-Oriented Software*. Nueva Jersey, USA: Editorial Prentice Hall.
- **Budd, T.** (2001). *Introduction to Object-Oriented Programming*. Oregon, USA: Ed. Addison Wesley.
- **Weisfeld, M.** (2013). *The Object-Oriented Thought Process*. USA: Pearson Education.
- **Gamma, E., Helm, H.; Johnson, R.; Vlissides, J.** (1994). *Design Patterns. Elements of Reusable Objects Oriented Software*. Addison – Wesley.
- **Fontela, C.** (2018). *Programación Orientada a Objetos*. Facultad de Ingeniería de la Universidad de Buenos Aires.
- **Juárez, A.** (2014). *Java + Programación Orientada a Objetos*. EUDEBA. Facultad de Ingeniería de la Universidad de Buenos Aires.

4.3 Bibliografía optativa:

- **Fowler, M. y Scott, K.** (1999). *UML gota a gota*. México: Addison – Wesley.
- **Gosling, J.; Steele, G.; Bracha, G.; Buckley, A.** (2015). *The Java Language Specification*. California, USA: Oracle America, Inc.

5. Metodología de enseñanza:

Las clases son teóricas / prácticas, virtuales de manera sincrónica. En algunas de ellas, presentamos y explicamos los conceptos, luego pedimos que realicen un ejercicio que es presentado por uno o más estudiantes voluntarios, y es revisado y analizado, en primer lugar, por sus pares, y luego, los/as docentes podemos agregar alguna observación o realizar aclaraciones sobre los temas discutidos.

Algunas tienen el enfoque de “clase invertida” que consiste en brindarles previamente algún material teórico / práctico para que lean u observen (apuntes y/o vídeos) y en clase, aclarar dudas, discutir distintos escenarios, etc.

Además, hay trabajos para desarrollar de forma asincrónica, con el formato de un trabajo práctico que pueden ser individuales o grupales.

Plan de trabajo en el campus:

Se combinarán clases teóricas con prácticas en las clases sincrónicas. Además, habrá material para que consulten e investiguen de forma asincrónica.

6. Actividades de investigación y extensión (si hubiera)

No aplica

7. Evaluación y régimen de aprobación

7.1 Aprobación de la cursada

Está previsto la entrega de trabajos prácticos con reentregas en caso de ser necesarias. Además, habrá un parcial de tipo Opción Múltiple con su correspondiente recuperatorio.

7.2 Aprobación 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 la cursada 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 administrado, 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.

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.

7.3 Criterios de calificación

En los criterios de evaluación tendremos en cuenta:

- Funcionalidad
- Diseño
- Buenas prácticas
- Modularización
- Optimización

8. Cronograma

Semana	Tema	Modalidad
1	Repaso POO - Java básico	Virtual
2	Java Avanzado - POO	Virtual
3	Diseño	Virtual
4	Diagramas de clase	Virtual
5	Herencia y polimorfismo	Virtual
6	Genericidad - Templates - TDD	Virtual
7	Casos de Uso - UML - User stories	Virtual
8	Diagramas de secuencia	Virtual
9	Excepciones	Virtual
10	Patrones de diseño	Virtual
11	Patrones de diseño	Virtual
12	Clase de repaso y consulta	Virtual
13	Parcial	Virtual
14	Consultas	Virtual
15	Recuperatorio	Virtual
16	Defensa Trabajos	Virtual