

CARRERA	Tecnicatura Superior en Análisis de Sistemas con Orientación en Redes Informáticas			
PLAN DE ESTUDIOS	Resolución 2561/06			
DOCENTE	Daniel Reyes			
UNIDAD CURRICULAR	Período	Año	Régimen	Hs. Cátedra
Programación IV (PIV)	2024	3ro.	Anual	3 (tres)

1. Fundamentación

La cátedra de Programación IV es un espacio fundamental de la carrera Tecnicatura Superior en Análisis de Sistemas con Orientación a Redes Informáticas, puesto que brindará al futuro profesional, el conocimiento suficiente para desarrollar programas de aplicación siguiendo el Paradigma Orientado a Objetos (POO) como alternativa a la programación procedimental.

Intentar conjugar los distintos aspectos teóricos y prácticos, con el objetivo de que el alumno complete el ciclo del problema hasta obtener su solución con computadora, mediante el planteamiento de un programa como una modelización de los elementos del mundo real, es una tarea que no puede estar sujeta a improvisaciones, y que conlleva una serie de actividades de decisión y planificación complejas y variadas, para las cuales es necesario tener una adecuada respuesta metodológica.

Durante el cursado, se introducirán los conceptos fundamentales y necesarios para abstraer un problema del mundo real aplicando el paradigma de POO, convertirlo en un modelo computable, escribir una solución simbólica del mismo y traducirla en una solución concreta y ejecutable. La suma de estos conocimientos permitirá a los futuros profesionales incorporar el enfoque basado en objetos para el desarrollo de programas, reconocer las diferencias que dicho paradigma presenta con respecto a la Programación Imperativa y adquirir un dominio amplio de los fundamentos del Lenguaje de Programación Java, así como explorar y comprender aspectos más avanzados para su aplicación efectiva en el desarrollo de software.

En el desarrollo de la Unidad Curricular, los alumnos incorporarán habilidades de modelado de datos e información, las cuales se integrarán y complementarán con las capacidades adquiridas en los trayectos formativos de las cátedras de “Programación III” “Análisis de Sistemas I”, “Análisis de Sistemas II” y “Práctica Profesional II”, completando de esta manera las competencias de los alumnos necesarias para el modelado de sistemas software.

Como resultado de dicha integración, los alumnos desarrollarán una aplicación software, construida aplicando el enfoque de la POO y utilizando Java. Esta aplicación deberá desarrollarse por grupos de alumnos y constituirá su proyecto grupal de cursada. La especificación de requisitos del trabajo a desarrollar será fijada por la Cátedra.

2. Objetivos

- Dominar los conceptos fundamentales del paradigma de la Programación Orientada a Objetos (POO) y sus diferencias con otros paradigmas.
- Adquirir habilidades en el diseño, construcción y ejecución de algoritmos basados en POO para resolver problemas de complejidad media-alta.
- Conocer los aspectos históricos, tecnológicos y los lenguajes asociados a la POO, con énfasis en Java.
- Diseñar e implementar soluciones utilizando POO, incluyendo la creación y manipulación de archivos.
- Comprender las reglas semánticas y sintácticas de Java, así como sus estructuras de control.
- Familiarizarse con mecanismos avanzados para la prevención y gestión de errores de ejecución.
- Adquirir conceptos para la construcción de interfaces gráficas de usuario (GUI).
- Reconocer y aplicar los aspectos más importantes de los diagramas de estructura estática de UML para el modelado de clases, objetos, atributos y relaciones.
- Dominar un Entorno de Desarrollo Integrado (IDE) para la creación de aplicaciones en Java.

3. Contenidos

Unidad 1: Introducción al Paradigma de Programación Orientado a Objetos (POO)

¿Qué es un Paradigma? ¿Qué es un Paradigma de Programación? El Paradigma Orientado a Objetos. Pilares de la POO: Abstracción, Encapsulación, Herencia y Polimorfismo.

Mecanismos básicos de la POO: Clases, Objetos, Métodos, Constructores, Destructores, Mensajes. Reusabilidad y extensibilidad. Ventajas de la POO. Desventajas de POO.

Bibliografía específica de la Unidad			
Autor	Título	Editorial	Lugar y año de edición
Booch, G.	Análisis y diseño OO con aplicaciones	Addison-Wesley	España, 1996
Joyanes, L y Zhonero, M.	Programación en Java 6. Algoritmos, Estructuras de datos y POO	Mc Graw Hill	México, 2011
Martin, J. y Odell, J.	Análisis y Diseño Orientado a Objetos	Prentice-Hall	México, 1992
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2024

Unidad 2: Introducción al lenguaje Java

Introducción a la plataforma Java. El lenguaje de programación Java: origen y características. Tipos de programas en Java. El proceso de compilación en Java. ¿Java es un lenguaje compilado o interpretado? ¿Qué es una API y para qué sirve? Versiones y distribuciones de Java. Evolución del Java estándar. Diferencias entre JDK, JRE y JVM.

Estructura de un programa en Java. La clase principal y el método *main*. Comentarios. Identificadores. Tipos primitivos de dato. Formato de representación de los datos numéricos reales. Declaraciones de variables. Declaración de variables final o constantes. Conversiones entre tipos de datos (*typecast*).

Operadores: concepto. Tipos de operadores: de asignación, aritméticos, aritméticos incrementales y combinados, relacionales, lógicos o booleanos y condicional. Operador de concatenación de cadenas. Separadores. Expresiones. Prioridad entre operadores. Sentencias selectivas o condicionales. Sentencias de control del flujo de un programa. Sentencia *break* y *continue*.

Bibliografía específica de la Unidad			
Autor	Título	Editorial	Lugar y año de edición
Deitel, P y Deitel, H.	Cómo programar en Java, 9na. edición	Pearson	México , 2012
Dean, J y Dean, S.	Introducción a la programación con Java	Mc Graw Hill	México, 2009
Joyanes, L y Zhonero, M.	Programación en Java 6. Algoritmos, Estructuras de datos y POO	Mc Graw Hill	México, 2011
Lewis, J. y Chase, J.	Estructura de datos con Java	Pearson	México, 2009
Horstmann, C.	Java concepts, 6th ed.	John Wiley	EEUU, 2009
Wu, T	Introducción a la Programación Orientada a Objetos con Java	Mc Graw Hill	España, 2000
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2024

Unidad 3: Java para Programación Orientada a Objetos

Declaración y uso de métodos: Parámetros, *return* y *void*. Sobrecarga de métodos. Objetos y clases. Modificadores de visibilidad. Clases anidadas e internas. El operador *instanceof*. Paquetes: Identificación de un paquete, Acceso a los componentes de un paquete, Modificadores de acceso. Tipos de atributos: Variables de instancia, de clase y variables finales. Constructores. Constructor por defecto. Declaración de un constructor. Llamada a un constructor desde otro constructor. Constructores sobrecargados. La referencia *this*. Tipos de métodos: de instancia y de clase. Parámetros. Variables locales. Referencias estáticas.

Herencia. Jerarquía de clases. Acceso a miembros y herencia. La referencia *super*. La clase *Object*. Herencia y constructores. Constructores de subclases. Sobrescritura de métodos. Polimorfismo. *Casting* de objetos y herencia. Clases y métodos finales. Clases abstractas. Métodos abstractos. Interfaces. Declaración e implementación de una interfaz en una clase. Implementación de múltiples interfaces en una clase. Jerarquía entre interfaces. Interfaz como un tipo de dato. Interfaces y clases abstractas. El *Garbage Collector* (recolector de basura). El método *finalize*.

Bibliografía específica de la Unidad			
Autor	Título	Editorial	Lugar y año de edición
Deitel, P y Deitel, H.	Cómo programar en Java, 9na. edición	Pearson	México , 2012
Dean, J y Dean, S.	Introducción a la programación con Java	Mc Graw Hill	México, 2009
Joyanes, L y Zhonero, M.	Programación en Java 6. Algoritmos, Estructuras de datos y POO	Mc Graw Hill	México, 2011
Lewis, J. y Chase, J.	Estructura de datos con Java	Pearson	México, 2009
Weiss, M.	Estructuras de datos en Java, 4ta. ed.	Pearson	México , 2013
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2024

Unidad 4: Arrays, Genéricos, Colecciones y Manejo de Errores en Java

Arrays en Java. Arrays unidimensionales. Recorrido con *for-each*. Arrays como parámetros de métodos. Arrays como resultado de métodos. Arrays multidimensionales. La clase *Arrays*. Array con números aleatorios. La clase *Random*. Array de objetos.

Clases Anónimas en Java. Genéricos en Java. Parámetros de tipo. Clase genérica. Métodos Genéricos. Herencia de genéricos. Conversión de tipos con genéricos. Sobrecarga de Métodos Genéricos. Tipos crudos (*raw types*). Comodines. Clases autoreferenciales. Genéricos y clases anónimas.

Colecciones en Java. El Framework de Colecciones de Java. Tipos de colecciones en Java. Métodos comunes. Implementaciones más comunes de las interfaces de colecciones. Las interfaces *Iterable* e *Iterator*. Las interfaces *Comparable* y *Comparator*.

Fundamentos de la gestión de excepciones. Tipos de excepciones. Excepciones no capturadas. Uso de *try* y *catch*. Descripción de una excepción. Cláusulas *catch* múltiples. Sentencias *try* anidadas. Cláusulas *throw*, *throws* y *finally*. Excepciones integradas en Java. Excepciones propias. Excepciones en Colecciones. Excepciones comunes en operaciones de colecciones. Excepciones integradas en operaciones de colecciones. Casos prácticos y ejemplos.

Bibliografía específica de la Unidad			
Autor	Título	Editorial	Lugar y año de edición
Collins, W.	Data Structures and the Java Collections Framework, 3rd. Ed.	Wiley	EEUU, 2011
Deitel, P y Deitel, H.	Cómo programar en Java, 9na. edición	Pearson	México , 2012



Dean, J y Dean, S.	Introducción a la programación con Java	Mc Graw Hill	México, 2009
Joyanes, L y Zhonero, M.	Programación en Java 6. Algoritmos, Estructuras de datos y POO	Mc Graw Hill	México, 2011
Liskov, B. y Guttag, J.	Program Development in Java: Abstraction, Specification, and Object-Oriented Design, 1st. ed.	Addison-Wesley	EEUU, 2000
Lewis, J. y Chase, J.	Estructura de datos con Java	Pearson	México, 2009
Naftalin, M. y Wadler, P.	Java Generics and Collections	O'Reilly Media	EEUU, 2006
Weiss, M.	Estructuras de datos en Java, 4ta. ed.	Pearson	México, 2013
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2024

Unidad 5: Fundamentos de Programación GUI con Java

Programación Dirigida por Eventos, Introducción a la programación dirigida por eventos, Concepto de interfaces gráficas de usuario (GUI), Exploración de AWT y Swing en Java, Jerarquía de componentes Swing, Clase JFrame y su papel en la creación de ventanas. Componentes y Contenedores, Tipos de componentes y su uso en Java, Detalles sobre componentes específicos como JLabel, JTextField, JButton, etc., Clases internas y clases internas anónimas en Java.

Gestión de Eventos, Modelo de delegación de eventos en Java, Eventos, fuentes y oyentes, Clases de eventos y interfaces de oyentes, Registro de eventos y manejo de eventos mediante clases adaptadoras.

Distribución y Diseño GUI, Distribución de componentes en una interfaz gráfica, Conceptos de diseño GUI y posicionamiento de componentes, Uso de gestores de posicionamiento y la clase JPanel en Java, Exploración de componentes adicionales como JTextArea, JCheckBox, JRadioButton, JComboBox, etc. Hilos (Threads), Introducción a los hilos en Java, Importancia del manejo de hilos en aplicaciones GUI, Principios básicos de programación multihilo en Java.

Bibliografía específica de la Unidad			
Autor	Título	Editorial	Lugar y año de edición
Deitel, P y Deitel, H.	Cómo programar en Java, 9na. edición	Pearson	México, 2012
Dean, J y Dean, S.	Introducción a la programación con Java	Mc Graw Hill	México, 2009
Lewis, J. y Chase, J.	Estructura de datos con Java	Pearson	México, 2009
Robinson, M. y Vorobiev, P.	Swing, Second Edition, 2nd. ed.	Manning	EEUU, 2003
Schildt, H.	Swing a beginner's guide	Mc Graw Hill	EEUU, 2009
Weiss, M.	Estructuras de datos en Java, 4ta. ed.	Pearson	México, 2013
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2024

4. Estrategias de Enseñanza

- Trabajos grupales e individuales.
- Elaboración de informes escritos.
- Exposiciones grupales e individuales
- Aprendizaje cooperativo.
- Aula invertida.

5. Evaluación

5.1. Criterios de Evaluación durante el Cursado

- Participación comprometida en los trabajos propuestos, ya sean individuales y/o grupales.
- Compromiso y responsabilidad en las tareas a desarrollar.
- Presentación de trabajos en tiempo y forma.
- Manejo adecuado de la bibliografía.
- Coherencia, claridad conceptual, integración y pertinencia tanto en oratoria como en el discurso escrito.
- Dominio, comprensión y transposición de los contenidos del programa.
- Capacidad de resolución de problemáticas presentadas.
- Capacidad de organizar, coordinar y elaborar la tarea encomendada.
- Aplicación de conocimientos y dominio de técnicas, mediante la respuesta a preguntas y la resolución de problema.
- Participación activa y pertinente en el desarrollo de las clases.

5.2. Instrumentos de Evaluación

Para valorar los aprendizajes de los alumnos y sistematizar dichas valoraciones, se utilizarán los siguientes instrumentos:

- Evaluaciones escritas
- Evaluaciones orales
- Trabajos prácticos
- Coloquios
- Informes

5.3. Criterios de Acreditación

5.3.1. Para Regularizar la unidad curricular

Para regularizar la asignatura, el alumno deberá:

- Tener el 80% de asistencia a las clases o de participación en el aula virtual.
- Presentar a término y aprobar el 90% de los trabajos prácticos o actividades propuesto.
- Presentar y aprobar un portfolio de evidencias de las experiencias del alumno.
- Aprobar los exámenes parciales propuestos o sus respectivos exámenes recuperatorios. La nota mínima obtenida no deber ser menor de 5 (cinco), en una escala de 1 (uno) a diez (10).
- Obtener una calificación final mínima de 5 (cinco) puntos, en una escala de 1(uno) a 10(diez), al terminar el cursado de la asignatura.

Las condiciones de regularización descriptas serán expuestas a los alumnos durante la presentación de la Unidad Curricular y del plan de trabajo de la misma.

5.3.2. Para Promocionar la Unidad Curricular

Según lo establece la normativa vigente (RAM), la Unidad Curricular NO ES promocionable.

5.4. Para Finalizar la Unidad Curricular como alumno Regular

En base a la normativa vigente (RAM), para rendir esta Unidad Curricular en calidad de alumno regular el alumno deberá rendir un primer examen final escrito sobre aspectos teóricos de la misma. Este examen versará sobre las temáticas comprendidas en el programa de la asignatura, desde la perspectiva de la bibliografía obligatoria. Los criterios de evaluación se basan en la acreditación de la lectura de la bibliografía, la argumentación sólida que utilice la terminología y los conceptos del campo de estudio y la apropiación crítica de los mismos.

5.5. Para finalizar la Unidad Curricular como alumno Libre

De acuerdo con la normativa actual (RAM), las condiciones establecidas para rendir esta Unidad Curricular en calidad de alumno libre comprenden:

- Presentación y aprobación de la totalidad de los trabajos prácticos o de un trabajo práctico integrador equivalente, en ambos casos, propuestos por la Cátedra. Para ello, el alumno deberá comunicarse con la Cátedra con treinta (30) días de anticipación a la fecha del examen final, a fin de obtener el conjunto de trabajos prácticos que deberá resolver. Los mismos deberán ser entregados al docente responsable siete (7) días antes de la fecha de examen final.
- Superada la instancia anterior, el alumno estará en condiciones de rendir un primer examen final escrito sobre aspectos teóricos (25%) y prácticos (75%) de la asignatura.
- En caso de aprobar el examen escrito, el alumno estará en condiciones de rendir un segundo examen final de características similares al que rinden los alumnos regulares. Este examen será tomado en forma oral ó escrita y versará sobre las temáticas comprendidas en el programa de la asignatura, desde la perspectiva de la bibliografía obligatoria. Los criterios de evaluación se basan en la acreditación de la lectura de la bibliografía, la argumentación sólida que utilice la terminología y los conceptos del campo de estudio y la apropiación crítica de los mismos.

6. Materiales para el aprendizaje

- Gabinete de Informática, con la suficiente cantidad de computadoras con el siguiente software debidamente instalado: Windows 10+ y una herramienta CASE para modelado del Diseño (ArgoUML, StartUML o similar).
- Apuntes de Cátedra.
- Guía de trabajos prácticos
- Pizarra y fibrones.
- Notebook.
- Cañón proyector.

7. Cronograma tentativo de los Contenidos Programados

El siguiente cronograma representa el avance semanal de los contenidos desarrollados, considerando el calendario escolar en curso, incluyendo los días feriados y/o no laborables.

Semana	Fecha	Tema
1	11/03/2024 - 17/03/2024	¿Qué es un Paradigma? ¿Qué es un Paradigma de Programación? El Paradigma Orientado a Objetos.
2	18/03/2024 - 24/03/2024	Pilares de la POO: Abstracción, Encapsulación, Herencia y Polimorfismo. Introducción a la plataforma Java.
3	25/03/2024 - 31/03/2024	El lenguaje de programación Java: origen y características. Tipos de programas en Java. El proceso de compilación en Java. Versiones y distribuciones de Java.



INSTITUTO DE EDUCACIÓN SUPERIOR N° 6.012

Pellegrini N° 295 - El Carril (Salta) – Rep. Argentina
FAX - T.E. N° 4-908015 - Email: isfd6012@gmail.com



Semana	Fecha	Tema
4	01/04/2024 - 07/04/2024	Estructura de un programa en Java. La clase principal y el método main. Comentarios. Identificadores. Tipos primitivos de dato. Declaraciones de variables. Declaración de variables final o constantes.
5	08/04/2024 - 14/04/2024	Conversiones entre tipos de datos (typecasting). Operadores. Sentencias selectivas o condicionales. Sentencias de control del flujo de un programa. Sentencia break y continue.
6	15/04/2024 - 21/04/2024	Declaración y uso de métodos: Parámetros. return y void. Sobrecarga de métodos.
7	22/04/2024 - 28/04/2024	Objetos y clases. Modificadores de visibilidad. Clases anidadas e internas.
8	06/05/2024 - 12/05/2024	Paquetes. Tipos de atributos. Constructores. Referencias estáticas.
9	13/05/2024 - 19/05/2024	Herencia. Jerarquía de clases. Acceso a miembros y herencia. Sobrescritura de métodos.
10	20/05/2024 - 26/05/2024	Polimorfismo. Tipos de polimorfismo. Clases y métodos finales. Clases abstractas. Interfaces.
11	27/05/2024 - 02/06/2024	Manejo de errores en Java. Excepciones.
12	03/06/2024 - 09/06/2024	Manejo de excepciones. Excepciones personalizadas.
13	10/06/2024 - 16/06/2024	Primer examen parcial
14	24/06/2024 - 30/06/2024	Recuperatorio del primer examen parcial
15	12/08/2024 - 18/08/2024	Arrays en Java. Recorrido con for-each. Arrays como parámetros de métodos y como resultado de métodos. La clase Arrays. La clase Random. Array de objetos.
16	19/08/2024 - 25/08/2024	Clases Anónimas en Java. Genéricos en Java. Parámetros de tipo. Clase genérica. Métodos Genéricos.
17	26/08/2024 - 01/09/2024	Sobrecarga de Métodos Genéricos. Comodines. Genéricos y clases anónimas.
18	02/09/2024 - 08/09/2024	Colecciones en Java. El Framework de Colecciones de Java. Tipos de colecciones.
19	16/09/2024 - 22/09/2024	Tipos de colecciones en Java (continuación). Métodos comunes. Implementaciones más comunes de las interfaces de colecciones.
20	23/09/2024 - 29/09/2024	Las interfaces Iterable e Iterator. Las interfaces Comparable y Comparator.
21	30/09/2024 - 06/10/2024	Programación Dirigida por Eventos. Introducción a la Programación Dirigida por Eventos.
22	07/10/2024 - 13/10/2024	Concepto de interfaces gráficas de usuario (GUI). Las librerías AWT y Swing en Java.
23	14/10/2024 - 20/10/2024	Jerarquía de componentes Swing. Clase JFrame y su papel en la creación de ventanas.
24	21/10/2024 - 27/10/2024	Componentes y Contenedores. Tipos de componentes y su uso en Java.
25	28/10/2024 - 03/11/2024	Gestión de Eventos. Modelo de delegación de eventos en Java.
26	04/11/2024 - 10/11/2024	Distribución y Diseño GUI. Distribución de componentes en una interfaz gráfica.
27	11/11/2024 - 17/11/2024	Segundo examen parcial
28	18/11/2024 - 24/11/2024	Recuperatorio del segundo examen parcial



Firma del docente