

RESOLUCIÓN "C.D." N° 054/25

CONCORDIA, 22/04/2025

**VISTO:** el proyecto de programa de la asignatura Programación II de la carrera de Licenciatura en Sistemas, presentado por la Prof. Elizabeth Silva Laves de fs. 2 a 11, el Acta del Comité Académico de la misma de fs. 12 y 13, lo manifestado por el Secretario Académico y la Directora de la carrera a fs. 14 y el resto de la documentación agregada al EXP\_FCAD-UER: 0000063/2025, y

**CONSIDERANDO:**

Que la Directora de la carrera de Licenciatura en Sistemas informa que el programa se adecua a los contenidos mínimos especificados para la precitada asignatura en el plan de estudios y la bibliografía es pertinente.

Que el Secretario Académico indicó que el régimen de evaluación y acreditación propuesto se encuadra en la normativa aplicable.

Que el Comité Académico de la carrera y la Comisión de Enseñanza se han expedido favorablemente aconsejando su aprobación, de fs. 12 a 13 y 29, respectivamente.

Que, conforme lo establece el Artículo 23, Inciso 6, del Estatuto, (aprobado por la Resolución "A.U." N° 41), este cuerpo es competente para decidir sobre el particular.

Por ello,

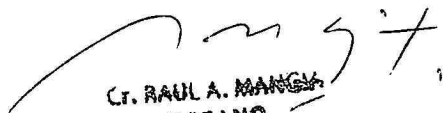
**EL CONSEJO DIRECTIVO  
DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN  
RESUELVE:**

**ARTÍCULO 1°.-** Aprobar el programa de la asignatura Programación II de la carrera de Licenciatura en Sistemas, presentado por la Prof. Elizabeth Silva Laves, que como Anexo Único integra la presente.

**ARTÍCULO 2°.-** Precisar que el régimen de evaluación y acreditación de la asignatura, cuya descripción obra en el anexo aprobado por el artículo precedente, se encuadra en las previsiones de los incisos b) y c), del Artículo 26, siguientes y concordantes, del Reglamento Académico aprobado por la Resolución "C.D." N° 509/16 y modificatorias (t.o. por la Resolución de Decano N° 135/19).

**ARTÍCULO 3°.-** Regístrese, notifíquese a la presentante, a la División Bedelía y a la Secretaría Académica, publíquese en el Boletín que se incluye en el Digesto Electrónico de la Universidad Nacional de Entre Ríos y cumplido, archívese.

  
Abeg. Francisco Daniel Victorio  
Secretario Consejo Directivo

  
Ct. RAUL A. MANGIN  
DECANO

RESOLUCIÓN "C.D." N° 054/25

ANEXO ÚNICO

Carrera: Licenciatura en Sistemas

PROGRAMACIÓN II

Año: 2°

Régimen de cursada: Cuatrimestral

Carácter: Obligatoria

Profesora: Elizabeth Silva Layes

Horas semanales: 8

Carga horaria total: 128

FUNDAMENTACIÓN

La asignatura Programación II es fundamental en la formación de los profesionales en el área de sistemas e informática, ya que proporciona los conocimientos y habilidades necesarios para el diseño, desarrollo y mantenimiento de *software* basado en paradigmas modernos de programación. Esta metodología permite mejorar la modularidad, reutilización y escalabilidad de los sistemas, aspectos esenciales en la ingeniería del *software* y en el desarrollo de sistemas complejos y de gran escala.

La materia contribuye directamente a varias de las actividades profesionales reservadas establecidas en el Anexo XXX de la Resolución ME N° 1254/2018.

Particularmente la materia aporta a las siguientes actividades profesionales reservadas al título de Licenciado en Sistemas:

AR1. Especificar, proyectar y desarrollar sistemas de información, sistemas de comunicación de datos y *software* cuya utilización pueda afectar la seguridad, salud, bienes o derechos.

AR5. Dirigir y controlar la implementación, operación y mantenimiento de lo anteriormente mencionado.

Y a los siguientes alcances del título:

AL1. Planificar, dirigir, realizar y evaluar proyectos de relevamiento de problemas del mundo real. Realizar la especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de *software* que se ejecuten sobre sistemas de procesamiento de datos.

AL2. Organizar, dirigir y controlar las áreas informáticas de las organizaciones, seleccionando y capacitando al personal técnico de los mismos.

AL3. Dirigir el relevamiento y análisis de los procesos funcionales de una organización, con la finalidad de dirigir proyectos de diseño de sistemas de información asociados, así como los sistemas de *software* que hagan a su funcionamiento. Determinar, regular y administrar las pautas operativas y reglas de

RESOLUCIÓN "C.D." N° 054/25

control que hacen al funcionamiento de las áreas informáticas de las empresas y organizaciones.

**AL4.** Entender, planificar y participar de los estudios técnicos-económicos de factibilidad o referentes a la configuración y dimensionamiento de sistemas de procesamiento de información. Supervisar la implantación de los sistemas de información y organizar y capacitar al personal afectado por dichos sistemas.

**AL5.** Establecer métricas y normas de calidad y seguridad de *software*, controlando las mismas a fin de tener un producto industrial que respete las normas nacionales e internacionales. Controlar la especificación formal del producto, del proceso de diseño, desarrollo, implementación y mantenimiento. Establecer métricas de validación y certificación de calidad.

**AL6.** Planificar, dirigir, realizar y evaluar los sistemas de seguridad en el almacenamiento y procesamiento de la información. Realizar la especificación, diseño, desarrollo, implementación y mantenimiento de los componentes de seguridad de información embebidos en los sistemas físicos y en los sistemas de *software* de aplicación. Establecer y controlar las metodologías de procesamiento de datos orientadas a seguridad, incluyendo *data warehousing*.

**AL8.** Planificar, dirigir, realizar y evaluar proyectos de sistemas de administración de recursos. Realizar la especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de eficiencia-calidad de los sistemas de administración de recursos que se implanten como *software* sobre sistemas de procesamiento de datos.

**AL9.** Analizar y evaluar proyectos de especificación, diseño, implementación, verificación, puesta a punto, mantenimiento y actualización de sistemas de procesamiento de datos.

**AL10.** Analizar y evaluar proyectos de especificación, diseño, implementación, verificación, puesta a punto y mantenimiento de redes de comunicaciones que vinculen sistemas de procesamiento de datos.

El dominio de la programación orientada a objetos (Programación II) es crucial para la formación de los futuros profesionales, ya que les permite desarrollar sistemas confiables y seguros, cumpliendo con los estándares y regulaciones de calidad requeridos en la industria del *software* y en el ámbito académico. Además, su enseñanza fortalece el perfil del egresado al dotarlo de herramientas para abordar proyectos de *software* complejos y alineados con las necesidades del mercado y la sociedad.

EJES Y COMPETENCIAS DE LA CARRERA A LOS QUE APORTA LA ASIGNATURA

EJE Y COMPETENCIAS	APORTE		
	ALTO	MEDIO	BAJO
1. Seleccionar los elementos que contribuyen a la mejor estrategia algorítmica para lograr soluciones de <i>software</i> eficientes aplicando buenas prácticas de programación.		X	

Handwritten notes and signatures on the left margin, including a large signature at the bottom.

RESOLUCIÓN "C.D." N° 054/25

2. Interpretar especificaciones de modelos para el diseño e implementación de programas mediante la correcta aplicación de principios de los paradigmas de programación.		X	
3. Verificar componentes de <i>software</i> para determinar su grado de correctitud realizando análisis conforme a métricas establecidas.			X
4. Evaluar lenguajes de programación para determinar su grado de adecuación a un contexto de desarrollo específico analizando las características propias del paradigma y de la herramienta			X

RESULTADOS DEL APRENDIZAJE

Competencias	Descripción de la competencia	Nivel de tributación	Resultados de aprendizaje
Genéricas tecnológicas	1. Identificar, formular y resolver problemas de informática	2	Analiza problemas de programación mediante la identificación de requerimientos y alcances, para formular soluciones efectivas aplicando buenas prácticas de programación
	4. Utilizar las técnicas y herramientas de aplicación en la informática	2	Emplea herramientas y técnicas de programación en el diseño de <i>software</i> , para llevar a cabo el proceso de desarrollo de <i>software</i> aplicando el paradigma de objetos
Genéricas sociales, políticas y actitudinales	1. Desempeñarse en equipos de trabajo	1	Colabora en actividades de programación para lograr resultados eficientes y conocimientos compartidos, aplicando la división de tareas y el uso de herramientas colaborativas en el marco del trabajo en equipo

Handwritten signature and scribbles on the left margin.

RESOLUCIÓN "C.D." N° 054/25

	2. Comunicarse con efectividad	1	Comunica de manera escrita y oral ideas y hallazgos técnicos en la resolución de problemas de programación, para mejorar la comprensión y el intercambio de información en el equipo aplicando el uso de recursos de documentación y herramientas colaborativas
	5. Aprender en forma continua y autónoma	1	Desarrolla estrategias para la autoevaluación y actualización continua en tecnologías de desarrollo de <i>software</i> , aplicando las mejores prácticas de programación
Específicas	1. Seleccionar los elementos que contribuyen a la mejor estrategia algorítmica para lograr soluciones de <i>software</i> eficientes aplicando buenas prácticas de programación	2	Selecciona estrategias algorítmicas eficientes, para desarrollar soluciones de <i>software</i> óptimas aplicando buenas prácticas de programación con el paradigma orientado a objetos
	2. Interpretar especificaciones de modelos para el diseño e implementación de programas mediante la correcta aplicación de principios de los paradigmas de programación	2	Interpreta especificaciones orientadas a objetos para el desarrollo de <i>software</i> , con el fin de implementar soluciones alineadas al paradigma de programación establecido, aplicando principios de encapsulamiento, herencia y polimorfismo

CONTENIDOS MÍNIMOS

Programación orientada a objetos. Clases. Objetos. Métodos. Encapsulamiento. Modelización de datos. Herencia. Polimorfismo. Ligamiento estático y dinámico. Interfaces. Excepciones. Manejo de eventos. Algoritmos concurrentes y paralelos. Programación concurrente. Hilos y multihilos.

UNIDADES TEMÁTICAS

## RESOLUCIÓN "C.D." N° 054/25

**UNIDAD 1:** Introducción al paradigma y modelado orientado a objetos. Concepto de objetos y clases. Características generales. Abstracción. Encapsulamiento. Herencia. Especialización y generalización. Herencia simple y múltiple. Polimorfismo. *Binding* estático y dinámico. Modelado orientado a objetos. Herramienta UML. Diagramas de clases.

**UNIDAD 2:** Conceptos básicos de la programación Java. Orígenes y características. Generalidades. Tipos de datos. Variables. Declaración de variables. Inicialización dinámica. Ámbito y tiempo de vida de una variable. Conversión de tipos básicos. Manejo de operación de tipos en expresiones. Entrada y salida. Concepto de *stream*. Biblioteca básica de flujos. El paquete *java.io*. Operadores. Operadores aritméticos. Operadores a nivel de *bit*. Operadores relacionales. Operador de asignación. Operador ternario. Precedencia de operadores. Sentencias de control. Sentencias de selección. Sentencias de iteración. Sentencias de salto. Arreglos. Arreglos unidimensionales. Arreglos multidimensionales.

**UNIDAD 3:** Clases y métodos. Fundamentos de las clases. Forma general de una clase. Declaración de objetos. Operador *new*. Asignación de variables de referencia a objetos. Introducción básica a métodos. Constructores. Referencia al objeto actual: *this*. Recolección de basura. Métodos. Sobrecarga de métodos. Pasaje de parámetros. Miembros *static* y final. Paquetes. Introducción al control de accesos.

**UNIDAD 4:** Herencia y polimorfismo. Fundamentos de la herencia. Superclases: acceso a miembros. Referencia a la superclase: *super*. Creación de una jerarquía multinivel. Sobrescritura de métodos. Selección de métodos en forma dinámica. Utilización de *final* con la herencia. Control de acceso. Clases abstractas. La clase *object*. Uso de *InstanceOf*. Interfaces. Definición de interfaces. Implementación de una interfaz. Implementación parcial. Utilización de las interfaces. Herencia en interfaces.

**UNIDAD 5:** Colecciones. Recorrido de colecciones: *Iterator* y *listIterator*. Comparación de elementos: *comparable* y *comparator*.

**UNIDAD 6:** Excepciones. Fundamentos de la gestión de excepciones. Tipos de excepciones. Captura de excepciones. Excepciones predefinidas. Lanzamiento de excepciones. Creación de excepciones propias. Propagación de excepciones.

**UNIDAD 7:** Interfaces gráficas y manejo de eventos. Interfaces gráficas. Conceptos generales. Control de eventos. Tipos de eventos. Generación y propagación de eventos. Métodos de control de eventos.

**UNIDAD 8:** Hilos y multihilos. Concepto de hilos. Estados de un hilo. Creación de hilos. Prioridades. Comunicación entre hilos. Sincronización. Suspensión, reanudación y finalización de hilos. Tipos de interacción entre procesos concurrentes.

### BIBLIOGRAFÍA

#### Básica

- Bloch, J. y Wesley, A. (2017). *Effective Java* – 3ra. edición. Professional.
- Booch, G. y otros. (2017). *The Unified Modeling Language User Guide*. 2da. edición. Professional.

## RESOLUCIÓN "C.D." N° 054/25

- Horstmann, C. (2024). *Core Java, Volume 1: Fundamentals*. 13ra. edición. Oracle Press.
- Horstmann, C. (2022). *Core Java: Advanced Features, Volume 2*. 12da. edición. Oracle Press.
- Kennedy, S. y Van Putten, M. (2023). *Learn Java with Projects: A concise practical guide to learning everything a Java professional really needs to know*. Packt Publishing.
- Loy, M. y otros. (2023). *Learning Java: An Introduction to Real-World Programming with Java*. 6ta. edición. O'Reilly Media.
- Sierra, K. y otros. (2022). *Head First Java: A Brain-Friendly Guide* 3ra edición. O'Reilly.

### Complementaria

- Debrauwer, L. y Van Der Heyde, F. (2016). *UML 2.5 - Iniciación, ejemplos y ejercicios corregidos*, 4ta. edición. Ediciones ENI.

## METODOLOGÍA DE ENSEÑANZA

La asignatura se desarrolla a través de una metodología activa y centrada en el estudiante, fomentando la construcción del conocimiento mediante la experimentación, la resolución de problemas y el desarrollo de proyectos aplicados. Se busca que los estudiantes adquieran una comprensión profunda de los principios del paradigma de objetos, promoviendo su aplicación en escenarios reales, integrando estrategias didácticas que favorecen el desarrollo de los resultados de aprendizaje y su vinculación con las competencias de egreso, asegurando que los estudiantes adquieran habilidades relevantes y fortalezcan las competencias necesarias para su formación profesional.

La asignatura combina clases teóricas con actividades prácticas en laboratorios físicos o virtuales, asegurando que los estudiantes desarrollen habilidades tanto conceptuales como técnicas. La carga horaria se distribuye de la siguiente manera:

- **Clases teórico-aplicadas:** donde se presentan conceptos, se realizan demostraciones y se resuelven ejercicios guiados.
- **Prácticas de laboratorio:** enfocadas en el desarrollo de programas utilizando los principios del paradigma de objetos, donde se incluyen ejercicios individuales y colaborativos.
- **Foros de discusión y actividades en el campus virtual:** donde se promueve la interacción entre estudiantes y docentes a través de la resolución de dudas, análisis de código y discusión de problemas.

### Uso de Laboratorios y Recursos Digitales

- Se utilizan laboratorios físicos equipados con computadoras y entornos de desarrollo para la implementación de ejercicios prácticos.
- Se aprovecha el campus virtual como plataforma de apoyo para compartir materiales, realizar evaluaciones y gestionar proyectos colaborativos.
- Se fomenta el uso de repositorios de código para la gestión y seguimiento de proyectos, promoviendo buenas prácticas en el desarrollo de *software*.

Esta metodología garantiza que los estudiantes no sólo comprendan los fundamentos del paradigma de objetos, sino que también sean capaces de aplicarlos en situaciones reales, fortaleciendo su aprendizaje autónomo y también el colaborativo.

## METODOLOGÍA DE EVALUACIÓN

## RESOLUCIÓN "C.D." N° 054/25

La evaluación de la asignatura se llevará a cabo de manera formativa y continua, asegurando que los estudiantes desarrollen las competencias establecidas mediante diversas estrategias e instrumentos de evaluación. Se fomentará la participación activa en clase, el trabajo colaborativo y la autoevaluación como parte del proceso de aprendizaje.

La metodología de evaluación se realizará teniendo presente los siguientes resultados de aprendizaje:

1. Analizar problemas de programación mediante la identificación de requerimientos y alcances, para formular soluciones efectivas aplicando buenas prácticas de programación.

Se evaluará la identificación de requerimientos y el uso de buenas prácticas a través de trabajos prácticos, y rúbricas. También se valorará la participación en foros de discusión sobre casos de estudio.

2. Emplear herramientas y técnicas de programación en el diseño de *software*, para llevar a cabo el proceso de desarrollo de *software* aplicando el paradigma de objetos.

Se desarrollarán programas aplicando el paradigma orientado a objetos, evaluando la calidad del código y estructuración mediante rúbricas y presentaciones individuales o grupales.

3. Colaborar en actividades de programación para lograr resultados eficientes y conocimientos compartidos, aplicando la división de tareas y el uso de herramientas colaborativas en el marco del trabajo en equipo.

Se realizarán trabajos en equipo con distribución de tareas, uso de herramientas colaborativas y evaluaciones entre pares para valorar la contribución de cada estudiante.

4. Comunicar de manera escrita y oral ideas y hallazgos técnicos en la resolución de problemas de programación, para mejorar la comprensión y el intercambio de información en el equipo aplicando el uso de recursos de documentación y herramientas colaborativas.

Se fomentará la elaboración de documentación y exposiciones orales sobre soluciones de programación, evaluando claridad y uso de términos técnicos.

5. Desarrollar estrategias para la autoevaluación y actualización continua en tecnologías de desarrollo de *software*, aplicando las mejores prácticas de programación.

Se promueve la autoevaluación, investigación de tendencias en programación y resolución de ejercicios prácticos adicionales para reforzar el aprendizaje.

6. Seleccionar estrategias algorítmicas eficientes, para desarrollar soluciones de *software* óptimas aplicando buenas prácticas de programación con el paradigma orientado a objetos.

Se evaluará la selección de algoritmos eficientes mediante resolución de problemas y análisis de rendimiento.

7. Interpretar especificaciones orientadas a objetos para el desarrollo de *software*, con el fin de implementar soluciones alineadas al paradigma de programación establecido, aplicando principios de encapsulamiento, herencia y polimorfismo.

## RESOLUCIÓN "C.D." N° 054/25

Se valorará la correcta implementación de encapsulamiento, herencia y polimorfismo a partir de especificaciones y diagramas UML, incluyendo ejercicios de modelado y desarrollo de código en Java.

### Instrumentos de Evaluación:

Para garantizar una evaluación integral, se emplearán los siguientes instrumentos:

- Rúbricas para la evaluación de trabajos prácticos y proyectos.
- Corrección de guías de ejercicios para medir la comprensión de los conceptos clave.
- Exposiciones orales en las que se valorarán los criterios de claridad, coherencia y uso de terminología técnica.
- Autoevaluaciones y coevaluaciones en proyectos grupales.
- Ejercicios teórico-prácticos para evaluar conocimientos y habilidades en resolución de problemas.

Esta metodología de evaluación permitirá un seguimiento continuo del aprendizaje, promoviendo el desarrollo de habilidades técnicas y actitudinales en el ámbito de la programación orientada a objetos.

### RÉGIMEN DE EVALUACIÓN Y ACREDITACIÓN

Los estudiantes para regularizar la asignatura deberán:

- Aprobar los trabajos prácticos y evaluaciones -por unidad- propuestos.
- Aprobar un parcial teórico-práctico o su recuperatorio al que podrán acceder si han cumplido con el requisito mencionado en el punto anterior.

Los estudiantes para acreditar la asignatura deberán:

- Además de lo expresado en el párrafo anterior, presentar y aprobar un trabajo final integrador para aprobar por el régimen de promoción directa.
- El estudiante que no apruebe el trabajo final integrador deberá presentarse a mesa de examen como estudiante regular en los turnos establecidos en el calendario académico.

### Trabajos prácticos:

Se realizarán trabajos prácticos individuales o en equipos durante el transcurso del cuatrimestre, los que consistirán en preguntas y ejercicios que el estudiante deberá resolver y entregar en fechas establecidas a tal fin.

### Parciales teórico-prácticos:

La parte teórico-práctica se evaluará mediante un parcial escrito que se realizará al término del cuatrimestre. Este parcial consistirá principalmente en aspectos teórico-prácticos y en temas teóricos a desarrollar. Los aspectos teórico-prácticos podrán versar sobre algunas temáticas abiertas propuestas durante las clases o preguntas sobre material entregado al estudiante, como artículos o temas específicos.


### Trabajo final integrador:

El Trabajo final integrador pretende englobar todos los conocimientos impartidos durante el cuatrimestre, de manera de culminar el aprendizaje del estudiante integrando los mismos en un producto final que luego deberá defender.

### Estudiantes libres:

Deberán rendir una instancia integrada de evaluación escrita y oral que versará sobre todos los contenidos de la asignatura.

  
Abag. Francisco Daniel Victorio  
Secretario Consejo Directivo

  
Cf. RAUL A. MANGIA  
DECAMO