



Universidad Nacional del Nordeste  
Facultad de Ciencias Exactas y  
Naturales y Agrimensura

RESOLUCIÓN: 0393/23  
CORRIENTES: 08 JUN 2023



VISTO el Expediente N° 09-2023-03060 por el cual la Directora de la carrera Licenciatura en Sistemas de Información, Mgter. Gladys N. Dapozzo, solicita la aprobación del Programa Analítico y de Examen de la Asignatura “Programación Orientada a Objetos”, y

CONSIDERANDO que corresponde a la propuesta presentada por la Profesora Responsable de la asignatura, Mgter. Cristina Liliam Greiner;

QUE se ajusta a las adecuaciones necesarias en función de los estándares de la convocatoria de acreditación;

QUE la presentación cumple con lo requerido en las Resoluciones N° 1074/22 C.D. y N° 1075/22 C.D.

QUE la solicitud cuenta con el aval de la Comisión de Carrera respectiva.

QUE obra el informe del Gabinete Psicopedagógico de FaCENA.

QUE cuenta con el informe de la Secretaría Académica.

LO aconsejado por la Comisión de Enseñanza y Planes de Estudios, criterio compartido por este cuerpo en la sesión del día 08-06-2023;

POR ELLO:

EL CONSEJO DIRECTIVO DE LA FACULTAD DE  
CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA  
R E S U E L V E:

ARTICULO 1º) APROBAR el Programa Analítico y de Examen para la asignatura “Programación Orientada a Objetos” del Plan de Estudio de la carrera Licenciatura en Sistemas de Información, conforme a los Anexos de la presente Resolución.

ARTICULO 2º) REMITIR copia al Profesor/a Responsable, Secretaría Académica, Dirección de Gestión Académica, Dirección de Gestión Estudios, Dirección de Gestión Biblioteca, Departamento Concurso y Carrera Docente, División Bedelía, Acreditación de Carreras, Secretaría de Departamento.

ARTICULO 3º) REGÍSTRESE, Comuníquese y archívese.  
RMR/YV

Lic. YANINA MEDINA  
Secretaría Académica  
FaCENA - UNNE



CRISTINA GLADIS ESCOBAR  
Protocolización y Archivo  
FaCENA - UNNE

Dr. ENRIQUE RAFAEL LAFFONT  
Vice Decano  
A/Cargo Decanato  
F.A.C.E.N.A.



Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y  
Naturales y Agrimensura



RESOLUCIÓN: 0395 23  
CORRIENTES: 08 JUN 2023

## ANEXO I

### PROGRAMA ANALÍTICO Y DE EXAMEN

#### 1. IDENTIFICACION

FACULTAD DE CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA

Departamento: Informática  
Área: Programación  
Bloque/s de conocimiento o  
Trayecto/s de Formación:  
Nombre de la asignatura: Programación Orientada a Objetos  
Carácter (*Obligatoria/Optativa*): Obligatoria  
Carrera: Licenciatura en Sistemas de Información  
Año en que se dicta: 2º  
Régimen de cursado (*Bim, Trim, Cuat, Anual*): Cuatrimestral Ubicación (1º, 2º, ...): 2º  
Nombre del profesor/a responsable: Greiner, Cristina Lilian  
Máximo título alcanzado: Mgter. en Informática y Computación

Carga horaria total: 128  
Carga horaria semanal: 8  
Teórica:  
Teórico/ Práctica: 3  
Práctica: 2,30  
Laboratorio: 2,30  
Seminario:  
Otro (*Especificar*):

#### 2. DESCRIPCION:

##### 2.1. Fundamentación

El software se ha convertido en un elemento crucial en la economía mundial. En consecuencia, se realizan grandes esfuerzos por lograr procesos de desarrollo de software eficientes, y que permitan obtener productos de calidad. La calidad en los productos software es una exigencia, dado que cada vez es más amplio su uso en procesos que son críticos para las organizaciones. Ante estas exigencias, surge la ingeniería del software, que, según Ivar Jacobson, es "...un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" y "...estudia la aplicación de técnicas formales de desarrollo a la construcción de software, buscando obtener productos de calidad, con la mejor relación costo/eficiencia".

En este sentido, la Programación Orientada a Objetos (POO) es una metodología de desarrollo de software que permite crear unidades funcionales extensibles, de forma que el desarrollador las pueda reutilizar. Es un paradigma que se fundamenta en concebir a un sistema como un conjunto de entidades que representan al mundo real (los objetos), que tienen distribuida y encapsulada la información y funcionalidad necesaria, y que cooperan entre sí para el logro de un objetivo común. Esta metodología de desarrollo se enfoca principalmente a la reutilización del código y a facilitar su mantenimiento. Esto se consigue mediante mecanismos de abstracción y otros principios, como





Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y  
Naturales y Agrimensura



RESOLUCIÓN:

CORRIENTES:

jerarquía, encapsulación y polimorfismo, características propias de la POO, y que redundan en un menor tiempo en el proceso, tanto de desarrollo como del posterior mantenimiento y extensibilidad.

La asignatura aporta al perfil del egresado los conocimientos, habilidades y metodologías del paradigma de objetos, así como las capacidades de análisis y síntesis para plantear soluciones a diferentes problemas informáticos.

El curso comienza con un análisis de las características que hacen a la calidad del software. Este análisis sirve para fundamentar el surgimiento de la POO, como una solución a las carencias propias de la programación imperativa y de otras técnicas de diseño más clásicas.

El núcleo del curso se ocupa del estudio de los mecanismos que definen el paradigma de la POO, como son el desarrollo de aplicaciones en términos de clases de objetos, la organización de las clases en jerarquías de herencia, el polimorfismo y la vinculación dinámica.

Adicionalmente, se integran aspectos distintivos de los lenguajes de POO, tal como el tratamiento de excepciones, dada la importancia que estas tecnologías tienen en la actualidad. Así también, se da relevancia al rol que tiene la documentación en todo proceso de desarrollo de software.

## 2.2. Objetivos generales de enseñanza de la asignatura

El objetivo general de la enseñanza de la programación es capacitar a los alumnos para construir metódicamente programas legibles, bien documentados, correctos, eficientes y fáciles de mantener y reutilizar. En este marco, la POO introduce una serie de técnicas y mecanismos que favorecen fundamentalmente los dos últimos criterios de calidad: la facilidad para mantener y modificar los programas, y la posibilidad de desarrollar programas o componentes fácilmente reutilizables.

Se espera que los alumnos apliquen los conceptos del paradigma de POO, percibiendo las ventajas que proporciona en la resolución de problemas informáticos, utilizando correctamente las técnicas que facilitan el mantenimiento y la reutilización, tales como el encapsulamiento, la herencia, el polimorfismo, entre otros.

## 2.3. Metodología

Con el propósito de promover el aprendizaje significativo<sup>1</sup>, los contenidos están estructurados de modo de incorporar progresivamente los conceptos. Para cada concepto concreto se exploran distintas alternativas de ejemplos, y la implementación práctica, lo que propicia su comprensión.

Las actividades se organizan en el dictado de clases teórico-prácticas, prácticas de resolución de problemas, y práctica experimental en el Laboratorio de Informática.

Las primeras se desarrollan en clases expositivas dialogadas, donde en primer lugar se brindan los conceptos, y luego se propone a los alumnos la resolución de un problema sencillo, que requiere la aplicación de los conceptos recibidos, y propicia la consulta y el refuerzo de los conceptos.

Las clases prácticas de resolución de problemas tienen como propósito que los alumnos comprendan cómo se implementan en forma práctica (en el software) los principios que sustentan el paradigma. Se muestran ejemplos, y luego los alumnos resuelven los problemas en forma individual o de a pares. Las clases de laboratorio favorecen este proceso, ya que permiten observar los resultados en forma directa. Todas estas actividades son presenciales y cuentan con la supervisión del docente. Los alumnos deben complementar estos espacios con tareas domiciliarias, que luego entregan en el recurso “Tarea” del Aula Virtual, para ser evaluadas posteriormente.

La modalidad de evaluación forma parte de la metodología, y cobra relevancia, ya que los alumnos son evaluados en forma individual, o de a pares, en los casos previstos, durante el proceso, a lo largo del cuatrimestre, en un

<sup>1</sup> “.. las nuevas adquisiciones se relacionan con lo que el alumno ya sabe, siguiendo una lógica, con sentido, y no arbitrariamente”. Ausubel-Novak-Hanesian (1983) Psicología Educativa: Un punto de vista cognoscitivo. 2º Ed. TRILLAS México.



RESOLUCIÓN:

CORRIENTES:

0395 23  
 08 JUN 2023

coloquio donde deben exponer (en fechas preestablecidas) cada una de las guías de trabajos prácticos entregadas en el Aula Virtual. Al momento del coloquio deben presentar además un informe, con estructura y formato establecidos, donde deben exponer los conceptos del paradigma de objetos aplicados, y el trozo de código correspondiente. En el coloquio también se realizan preguntas sobre los conceptos teóricos y su implementación práctica en el software. Esta modalidad permite detectar los logros y dificultades de cada alumno, y la devolución realizada por el docente es de gran valor para el alumno. Como parte de la evaluación se incluyen un examen parcial práctico y dos exámenes parciales de conceptos teóricos, que se implementan como cuestionarios en la plataforma de UNNE-Virtual. Finalizando el cuatrimestre, organizados en equipos, los alumnos deben resolver y presentar un trabajo integrador.

**3. Nivel de aporte de la asignatura al desarrollo de las Competencias de Egreso de la carrera**

Categoría (CE, CGT, CGS)	Competencia	0	1	2	3
CE1	Especificar, proyectar y desarrollar sistemas de información, sistemas de comunicación de datos y <b>software</b> cuya utilización pueda afectar la seguridad, salud, bienes o derechos.			X	
CE2	Proyectar y dirigir lo referido a seguridad informática.	X			
CE3	Establecer métricas y normas de calidad de software.		X		
CE4	Certificar el funcionamiento, condición de uso o estado de lo mencionado anteriormente.	X			
CE5	Dirigir y controlar la implementación, operación y mantenimiento de anteriormente mencionado.	X			
CGT1	Identificación, formulación y resolución de problemas de informática.			X	
CGT2	Concepción, diseño y desarrollo de proyectos de informática.	X			
CGT3	Gestión, planificación, ejecución y control de proyectos de informática.	X			
CGT4	Utilización de técnicas y herramientas de aplicación en la informática.				X
CGT5	Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	X			
CGS1	Fundamentos para el desempeño en equipos de trabajo.		X		
CGS2	Fundamentos para la comunicación efectiva.			X	
CGS3	Fundamentos para la acción ética y responsable.		X		
CGS4	Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local.	X			
CGS5	Fundamentos para el aprendizaje continuo.			X	
CGS6	Fundamentos para la acción emprendedora.	X			

**4. Propuesta para el desarrollo de las competencias**





Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y  
Naturales y Agrimensura



RESOLUCIÓN:

CORRIENTES:

0395 23

08 JUN 2023

Resultado de Aprendizaje	Unidades/Temas	Guía de Trabajo Práctico	Actividad Formativa	Estrategia de enseñanza
RA01. Aplica los conceptos del paradigma de POO para desarrollar software utilizando correctamente las técnicas que facilitan el mantenimiento y la reutilización, contribuyendo a la de calidad del software (CE)	Tema 1, 2, 3, 4, 5, 6, 7	TP 2, 3, 4, 5, 7	1- Clase Teórico-Práctica 2- Clase invertida 3- Clases Prácticas 4- Laboratorio	<ul style="list-style-type: none"> <li>1.1. Clase expositiva dialogada</li> <li>1.2. Resolución de Problemas: Finalizada la exposición, se plantea un problema de escasa complejidad, que los alumnos deben resolver en clase, aplicando los conceptos recién incorporados, con el apoyo del docente.</li> <li>2.1. Se dan las indicaciones en el aula virtual o en el encuentro presencial previo, sobre las actividades a realizar con el tema seleccionado</li> <li>2.2. En el encuentro presencial se realiza una construcción colectiva del tema</li> <li>2.3. El docente finaliza con una síntesis del tema</li> <li>3.1. Resolución de Problemas           <ul style="list-style-type: none"> <li>3.1.1. El docente recupera los conceptos teóricos relacionados al TP en desarrollo, informa los logros esperados al finalizar el TP, resuelve con la participación de los alumnos un problema tipo.</li> <li>3.1.2. El alumno resuelve los problemas del TP, aplicando todas las técnicas derivadas de los principios de la POO</li> </ul> </li> <li>4.1. El alumno implementa las soluciones a problemas del TP, aplicando todas las técnicas derivadas de los principios de la POO:           <ul style="list-style-type: none"> <li>4.1.1. Presencial, bajo la supervisión del docente</li> <li>4.1.2. No presencial, con tarea domiciliaria que entrega en el recurso "Tarea" del Aula Virtual, para luego ser evaluada</li> </ul> </li> </ul>
RA02. Implementa mecanismos de los lenguajes de programación para lograr mayor eficiencia y robustez, haciendo uso de las características de los lenguajes de programación (manejo de excepciones) que contribuyen a estos factores de calidad del software (CE)	Tema 8	TP 6	Clase Teórico-Práctica	<ul style="list-style-type: none"> <li>1. Clase expositiva dialogada</li> <li>2. Resolución de Problemas           <ul style="list-style-type: none"> <li>2.1. El docente resuelve con la participación de los alumnos un problema tipo.</li> <li>2.2. El alumno resuelve los problemas del TP, aplicando los conceptos recién incorporados, con el apoyo del docente.</li> </ul> </li> </ul>
RA03. Utiliza herramientas y tecnologías para implementar las		TP 1, 2, 3, 4, 5, 7	Laboratorio	El alumno implementa las soluciones a problemas del TP utilizando las herramientas y el entorno de desarrollo propuesto por la asignatura



*Universidad Nacional del Nordeste*

*Facultad de Ciencias Exactas y  
Naturales y Agrimensura*



RESOLUCIÓN:  
CORRIENTES:

0395 23  
08 JUN 2023

Resultado de Aprendizaje	Unidades/Temas	Guía de Trabajo Práctico	Actividad Formativa	Estrategia de enseñanza
soluciones de problemas informáticos recurriendo a la documentación de software (CGT-4)				<ul style="list-style-type: none"> <li>• Presencial, bajo la supervisión del docente</li> <li>• No presencial, con tarea domiciliaria que entrega en el recurso “Tarea” del Aula Virtual, para luego ser evaluada</li> </ul>
RA04. Integra equipos de trabajo para aprovechar las habilidades y capacidades de los pares, aportando las propias. Actúa en forma ética y responsable (CGS-1/CGS-3)		TP 7		<ul style="list-style-type: none"> <li>- Determinación de roles</li> <li>- Distribución de tarea</li> <li>- Elaboración de módulo</li> <li>- Integración de módulos</li> <li>- Elaboración de informe</li> </ul>
RA05. Utiliza el lenguaje oral y escrito para afianzar habilidades de comprensión y expresión lingüística y desenvolvimiento frente a su interlocutor, empleando los términos técnicos apropiados. (CGS-2)		Defensa de los TP		<ul style="list-style-type: none"> <li>- Exposición del alumno en coloquio</li> <li>- Presentación de informe de cada trabajo práctico evaluado</li> </ul>
RA06. Gestiona información proveniente de diversas fuentes para incorporar ideas y conocimientos nuevos para la resolución de problemas informáticos. (CGS-5)				<ul style="list-style-type: none"> <li>- Autoestudio de material específico puesto a disposición en Aula virtual, con la indicación de lectura obligatoria</li> <li>- Acceso a la API del lenguaje de programación utilizado</li> </ul>





RESOLUCIÓN: 0395 23  
 CORRIENTES: 08 JUN 2023

## 5. Sistema de Evaluación

Tipo de evaluación	Criterio de Evaluación	RA Nº	Técnica de evaluación
<b>Formativa</b>	Correcta aplicación de los conceptos teóricos. Utilización apropiada del lenguaje técnico específico. Correcta aplicación de las técnicas de POO.	01, 02, 03, 04, 05, 06	Revisión del código. Coloquios. Informes escritos.
<b>Sumativa</b>	Correcta aplicación de los conceptos teóricos y las técnicas de POO	01, 03	Examen práctico escrito. Cuestionario en el Aula Virtual.

## 6. Régimen de acreditación

### 6.1. Condiciones para regularizar la materia:

- 6.1.1. Aprobar un Examen Parcial Práctico (EPP). Se contempla 1 recuperatorio y 1 extraordinario
- 6.1.2. Presentar en tiempo y forma y Aprobar los Trabajos Prácticos (TP) 3, 4, 5 y 7. Se contempla 1 recuperatorio para cada instancia y 1 único recuperatorio extraordinario para uno cualquiera de los TP
- 6.1.3. Asistir a las clases de práctica y laboratorio (75%)

### 6.2. Condiciones para aprobar la materia sin examen final (promoción):

- 6.2.1. Cumplir las condiciones para regularizar, obteniendo un promedio de notas igual o mayor a 8. Para calcular el promedio se consideran las notas de los TP evaluados en laboratorio y el EPP.
- 6.2.2. Aprobar 2 exámenes parciales de conceptos teóricos, con 1 recuperatorio para cada parcial, y 1 único extraordinario
- 6.2.3. Asistir a las clases de teoría (75%)

La nota final será el promedio obtenido de las notas de los TP evaluados en laboratorio y el EPP.

### 6.3. Condiciones para aprobar la materia con examen final:

#### 6.3.1. Regular

Los alumnos en condición de Regular rinden un examen oral ó escrito sobre los contenidos del programa vigente.

#### 6.3.2. Libre

Los alumnos en condición de Libre deben:

- 6.3.2.1. Enviar a través del SIU-Guaraní un archivo comprimido con la resolución de los Trabajos Prácticos 3, 4 y 5. El envío debe ser realizado con al menos 1 día de anticipación de la fecha de la mesa de examen.

#### 6.3.2.2. Aprobar un examen práctico.

Cumplidos estos dos requisitos, el alumno está en condiciones de rendir el examen teórico como alumno regular, según la normativa vigente.

## 7. Programa Analítico

### 7.1. Contenidos mínimos (del Plan de Estudios)

Conceptos fundamentales del paradigma de POO. Encapsulamiento. Abstracción. Jerarquía. Modularidad. Objetos y clases. Nociones del Lenguaje Unificado de Modelado (UML). Diagrama de Clases. Mecanismos de abstracción. Herencia. Polimorfismo. Tratamiento de problemas en tiempo de ejecución.

### 7.2. Contenidos por unidad/tema

#### Unidad I: Conceptos fundamentales de Programación Orientada a Objetos

##### Tema 1: Introducción a la POO.

Introducción. La complejidad del software. La crisis del software. Ingeniería del Software. Factores de calidad del software. La orientación a objetos. Características de la programación orientada a objetos: abstracción,





Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y  
Naturales y Agrimensura



RESOLUCIÓN: 0395 23  
CORRIENTES: 08 JUN 2023

encapsulamiento, modularidad, jerarquía. Beneficios de la programación orientada a objetos. Lenguajes de programación orientados a objetos.

#### **Tema 2: Concepto de Objeto y Clase.**

Concepto de Objeto. Propiedades de los Objetos. Objeto y encapsulación. Interfaz pública. Interacción entre objetos: mensajes. Concepto de Clase. Propiedades de una clase: atributos y métodos. Instancias de una clase. Encapsulación – Visibilidad. Implementación de una clase en Java. Métodos Constructores. La Programación Orientada a Objetos.

#### **Tema 3: La documentación en el desarrollo del software.**

Introducción. Documentación externa. Modelado. Modelo. Lenguaje de modelado. UML. Diagramas. Documentación Interna. El estilo de programación - Reglas de buen estilo. Los Comentarios. Qué documentar. Tipos de Comentarios. Javadoc. Nomenclatura. Indentación.

#### **Tema 4: Objeto, Clase, UML y Java: un enfoque práctico**

Objeto. Comportamiento. ¿Qué hace? ¿Cómo lo hace? Implementación: método vs. mensaje. El estado interno. Envío de mensajes. Formato de mensajes. Métodos. Especificación de un método. Observadores y mutadores. Objeto y clase. Clase y UML. Envío de mensajes considerando el concepto de clases. Atributos y mensajes de clase. Formas de conocimiento. Seudovariable this.

#### **Tema 5: Reutilización: Clases predefinidas**

Colecciones/Contenedores. Colecciones en los lenguajes OO. Tipos de colecciones: homogéneas/heterogéneas, estáticas/dinámicas, ordenadas/no ordenadas. Operaciones con colecciones. Colecciones en Java. Jerarquía de Interfaces y de Clases de la Java Collections Framework. Persistencia de objetos. Reutilizando clases para entrada/salida en Java. Reutilizando clases para tratar fechas en Java.

#### **Unidad II: Conceptos avanzados de Programación Orientada a Objetos**

#### **Tema 6: Mecanismos de abstracción. Herencia.**

Introducción. Mecanismos de abstracción. Reutilización y mecanismos de abstracción. Herencia. Herencia simple. Notación UML. Redefinición. Constructores y destructores considerando herencia. Herencia múltiple. Interfaces. Herencia y reutilización. Mecanismo de respuesta a mensajes en una jerarquía de clases. Herencia y ocultación de información.

#### **Tema 7: Polimorfismo.**

Introducción. Aplicación del Polimorfismo. Ventajas del polimorfismo. Redefinición. Sobrecarga y redefinición. Ligadura dinámica. Métodos virtuales. Clases y métodos abstractos. Notación UML. Transformación de tipos en tiempo de ejecución. Transformación implícita y explícita. Información de tipos en tiempo de ejecución.

#### **Unidad III: Conceptos de los lenguajes de programación que contribuyen a la calidad del software**

#### **Tema 8: Tratamiento de problemas en tiempo de ejecución: Excepciones**

Comprobación de estados. Enfoques conservadores. Enfoques optimistas. Errores y excepciones. Formas de tratar excepciones. Lanzamiento de excepciones. Captura de excepciones: manejadores. Atributos y métodos en excepciones creadas por el programador. Jerarquía de excepciones en Java.





1983 2023  
REPUBLICA ARGENTINA 40 AÑOS DE DEMOCRACIA EN LA REPUBLICA

RESOLUCIÓN: 039523  
CORRIENTES: 08 JUN 2023

Universidad Nacional del Nordeste  
Facultad de Ciencias Exactas y  
Naturales y Agrimensura

#### 8. BIBLIOGRAFIA:

##### 8.1. Bibliografía Específica

Título	Autores	Editorial	Ejemplares disponibles	Año de edición
Orientación A Objetos - Diseño y Programación	Carlos Fontela	Nueva Librería ISBN: 9871104642	1	2008
Orientación a Objetos con Java y UML	Carlos Fontela	Nueva Librería ISBN: 9871104200	4	2004
Java: cómo programar	Deitel, Paul J.; Deitel, Harvey M.; Campos García, Gabriela Azucena; Martínez Román, Roberto	Pearson Educación ISBN: 789702611905	8	2008
Java: cómo programar	Deitel, Paul J.; Deitel, Harvey M.	Pearson Educación ISBN: 789702611905	1	2012
Piensa en Java. 4ta. Ed.	Bruce Eckel	Pearson ISBN: 978848966034 2	3	2007
JAVA. La Guía total del Programador	Dos Santos, Sergio Alejandro	MP Ediciones ISBN: 9875262587	1	2005
Java 2: manual de usuario y tutorial	Froufe Quintas, Agustín	Alfaomega ISBN: 9701508467	1	2002
Programación orientada a objetos	Luis Joyanes Aguilar	Mc Graw-Hill ISBN: 8448120159	1	1998
Cómo programar en C++	Deitel, Harvey M.; Deitel, Paul J.	Pearson Educación ISBN: 789702612735	3	2009
Programación en C++	Luis Joyanes Aguilar	Mc Graw-Hill ISBN: 844814645X	1	2006
Fundamentos de programación: algoritmos estructuras de datos y objetos	Luis Joyanes Aguilar	Mc Graw-Hill ISBN: 8448136640	1	2003
Programación en C, C++, Java y UML	Joyanes Aguilar, Luis; Zahonero Martínez, Ignacio	Mc Graw-Hill ISBN: 786071512123	4	2014
Programación en C++ : un enfoque práctico	Joyanes Aguilar, Luis; Sánchez García, Lucas	Mc Graw-Hill ISBN: 8448146433	1	2006
Programación en C,++, Java y UML	Joyanes Aguilar, Luis; Zahonero Martínez, Ignacio	Mc Graw-Hill ISBN: 789701069493	1	2010
Programación orientada a objetos con C++	Ceballos Sierra, Francisco Javier	Alfaomega; Ra-Ma 9701509560	2	2004



Título	Autores	Editorial	Ejemplares disponibles	Año de edición
Programación orientada a objetos en Java	Blasco, F.	RA-MA Editorial	<a href="https://elibro.net/es/lc/unne/titulos/222723">https://elibro.net/es/lc/unne/titulos/222723</a> acceso <u>multiusuario</u>	2019
Programación orientada a objetos con C++.	Ceballos Sierra, F. J.	RA-MA Editorial	<a href="https://elibro.net/es/lc/unne/titulos/106519">https://elibro.net/es/lc/unne/titulos/106519</a> acceso <u>multiusuario</u>	2018 - 5a. ed.
Programación orientada a objetos con C++	Ceballos Sierra, F. J	RA-MA Editorial	<a href="https://elibro.net/es/lc/unne/titulos/62459">https://elibro.net/es/lc/unne/titulos/62459</a> acceso <u>multiusuario</u>	2015 - 4a. ed.

#### 8.2. Bibliografia Complementaria.

Los alumnos acceden al repositorio eLibro desde la plataforma de UNNE Virtual. Contacto: e-libros@bib.unne.edu.ar

Título	Autores	Editorial	Ejemplares disponibles	Año edición
Programación orientada a objetos C++ y Java: un acercamiento interdisciplinario	López Goytia, J. L.	Grupo Editorial Patria	<a href="https://elibro.net/es/lc/unne/titulos/39461">https://elibro.net/es/lc/unne/titulos/39461</a>	2015
Programación orientada a objetos.	Moreno Pérez, J.	RA-MA Editorial.	<a href="https://elibro.net/es/lc/unne/titulos/106461">https://elibro.net/es/lc/unne/titulos/106461</a>	2015
Programación orientada a objetos usando java	Flórez Fernández, H. A.	Ecoe Ediciones	<a href="https://elibro.net/es/lc/unne/titulos/69236">https://elibro.net/es/lc/unne/titulos/69236</a>	2012
Diseñar y programar, todo es empezar: una introducción a la Programación Orientada a Objetos usando UML y Java	Vélez Serrano, J.	Dykinson	<a href="https://elibro.net/es/lc/unne/titulos/63076">https://elibro.net/es/lc/unne/titulos/63076</a>	2011
Todo lo básico que debería saber: sobre programación orientada a objetos en Java	García Llinás, L. F.	Ediciones de la U.	<a href="https://elibro.net/es/lc/unne/titulos/69813">https://elibro.net/es/lc/unne/titulos/69813</a>	2010
Fundamentos de la programación orientada a objetos: una aplicación a las estructuras de datos en Java	Ruiz Rodríguez, R.	El Cid Editor.	<a href="https://elibro.net/es/lc/unne/titulos/34869">https://elibro.net/es/lc/unne/titulos/34869</a>	2009
Lógica de programación orientada a objetos	Oviedo Regino, E. M.	Ecoe Ediciones	<a href="https://elibro.net/es/lc/unne/titulos/70431">https://elibro.net/es/lc/unne/titulos/70431</a>	2015

#### 9. PROGRAMA DE EXAMEN:

No corresponde.





#### 10. NOMINA DE TRABAJOS PRACTICOS:

10.1. Resolución de situaciones problemáticas. Las clases se desarrollan en aulas. Los alumnos pueden trabajar en sus propias notebooks o con papel y lápiz.

Nro. TP	Modalidad	Tema/Descripción
1	De a pares	Objetivo: Que el alumno se familiarice con la sintaxis del lenguaje de programación y el uso del entorno de trabajo (IDE). Resolución de una Guía de Trabajos Prácticos de ejercicios simples, utilizando el lenguaje de programación y el IDE propuesto por la asignatura. El trabajo de a pares contribuye a la incorporación de los nuevos conocimientos.
2	De a pares	Objetivo: que el alumno aplique los conceptos fundamentales del paradigma de objetos para definir clases. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la definición de clases con sus constructores y métodos, considerando el encapsulamiento, e instanciando objetos y enviándole mensajes para verificar su correcta implementación, utilizando el lenguaje de programación y el IDE propuesto por la asignatura. El trabajo de a pares contribuye a la incorporación de los nuevos conocimientos.
3	Individual	Objetivo: que el alumno aplique los conceptos fundamentales del paradigma de objetos para implementar la interacción entre objetos y la reutilización de clases definidas previamente. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la definición de clases con constructores y métodos que incorporan otros objetos, e instanciando objetos que se envían mensajes entre sí, para verificar su correcta implementación, utilizando el lenguaje de programación y el IDE propuesto por la asignatura. El trabajo individual permite al alumno <b>evaluar</b> su autonomía para aplicar los nuevos conocimientos.
4	Individual	Objetivo: que el alumno aplique los conceptos del paradigma de objetos para implementar la interacción múltiple entre objetos, utilizando colecciones de objetos, y reutilizando clases predefinidas del lenguaje de programación. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la utilización de colecciones de objetos de clases con sus constructores y métodos que incorporan otros objetos (únicos o en colecciones), e instanciando objetos que se envían mensajes entre sí para verificar su correcta implementación. El trabajo individual permite al alumno <b>afianzar</b> su autonomía para aplicar los nuevos conocimientos.
5	Individual	Objetivo: que el alumno aplique los conceptos avanzados del paradigma de objetos para implementar la herencia y el polimorfismo en jerarquías de clases. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la implementación de la herencia y el polimorfismo en jerarquías de clases, definiendo adecuadamente los constructores dentro de la jerarquía, valorando la reutilización de código mediante el mecanismo de abstracción de la herencia, e instanciando objetos que se envían mensajes entre sí para verificar su correcta implementación.
6	Individual	Objetivo: que el alumno aplique conceptos de los lenguajes de programación orientado a objetos que contribuyen a la robustez y a la calidad del software, mediante el tratamiento de excepciones. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la implementación del tratamiento de excepciones y la verificación de su correcta implementación y utilidad.
7	Grupal	Objetivo: que el alumno integre y aplique todos los conceptos del paradigma de programación orientado a objetos aprendidos y que trabaje en equipo. Resolución de un problema de mayor complejidad que requiere la aplicación de todos los conceptos aprendidos. La guía, además del enunciado del problema, contiene las consignas para el trabajo en equipo, que incluye la resolución de distintos módulos en subgrupos y la integración final de todos los módulos para la resolución completa del problema.

#### 10.2. Laboratorio / Trabajo de campo

Se implementan las mismas Guías de Trabajos Prácticos. Las clases se desarrollan en los laboratorios de informática. Los alumnos tienen a disposición computadoras para llevar a cabo la tarea.





*Universidad Nacional del Nordeste  
Facultad de Ciencias Exactas y  
Naturales y Agrimensura*

1983 - 2023  
REPÚBLICA ARGENTINA 40 AÑOS DE DEMOCRACIA EN LA REPÚBLICA  
RESOLUCIÓN: 039523  
CORRIENTES: 08 JUN 2023

**ANEXO II**  
**CARGA HORARIA**

**1. IDENTIFICACION**

FACULTAD DE CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA

Departamento: Informática

Área: Programación

Bloque/s de conocimiento o Algoritmos y Lenguajes

Trayecto/s de Formación:

Nombre de la asignatura: Programación Orientada a Objetos

Carácter (*Obligatoria/Optativa*): Obligatoria

Carrera: Licenciatura en Sistemas de Información

Año en que se dicta: 2º

Régimen de cursado (*Bim, Trim, Cuat, Anual*): Cuatrimestral Ubicación (1º, 2º, ...): 2º

Nombre del profesor/a responsable: Greiner, Cristina Lilian

Máximo título alcanzado: Mgter. en Informática y Computación

**Carga horaria total:** 128

**Carga horaria semanal:** 8

Teórica:

Teórico/ Práctica: 3

Práctica: 2,30

Laboratorio: 2,30

Seminario:

Otro (*Especificar*):

**2. RÉGIMEN DE ACREDITACIÓN**

**2.1. Condiciones para regularizar la materia:**

- 2.1.1. Aprobar un Examen Parcial Práctico (EPP). Se contempla 1 recuperatorio y 1 extraordinario
- 2.1.2. Presentar en tiempo y forma y Aprobar los Trabajos Prácticos (TP) 3, 4, 5 y 7. Se contempla 1 recuperatorio para cada instancia y 1 único recuperatorio extraordinario para **uno** cualquiera de los TP
- 2.1.3. Asistir a las clases de práctica y laboratorio (75%)

**2.2. Condiciones para aprobar la materia sin examen final (promoción):**

- 2.2.1. Cumplir las condiciones para regularizar, obteniendo un promedio de notas igual o mayor a 8. Para calcular el promedio se consideran las notas de los TP evaluados en laboratorio y el EPP.
- 2.2.2. Aprobar 2 exámenes parciales de conceptos teóricos, con 1 recuperatorio para cada parcial, y 1 único extraordinario
- 2.2.3. Asistir a las clases de teoría (75%)

La nota final será el promedio obtenido de las notas de los TP evaluados en laboratorio y el EPP.

**2.3. Condiciones para aprobar la materia con examen final:**

**2.3.1. Regular**

Los alumnos en condición de Regular rinden un examen oral ó escrito sobre los contenidos del programa vigente.

**2.3.2. Libre**

Los alumnos en condición de Libre deben:

- 2.3.2.1. Enviar a través del SIU-Guaraní un archivo comprimido con la resolución de los Trabajos Prácticos 3, 4 y 5. El envío debe ser realizado con al menos 1 día de anticipación de la fecha de la mesa de examen.

- 2.3.2.2. Aprobar un examen práctico.

Cumplidos estos dos requisitos, el alumno está en condiciones de rendir el examen teórico como alumno regular, según la normativa vigente.





*Universidad Nacional del Nordeste*

*Facultad de Ciencias Exactas y  
Naturales y Agrimensura*

RESOLUCIÓN:

CORRIENTES:

0395 23  
08 JUN 2023

### 3. NOMINA DE TRABAJOS PRACTICOS:

3.1. Resolución de situaciones problemáticas. Las clases se desarrollan en aulas. Los alumnos pueden trabajar en sus propias notebooks o con papel y lápiz.

TP	Modalidad	Tema/Descripción
1	De a pares	Objetivo: Que el alumno se familiarice con la sintaxis del lenguaje de programación y el uso del entorno de trabajo (IDE). Resolución de una Guía de Trabajos Prácticos de ejercicios simples, utilizando el lenguaje de programación y el IDE propuesto por la asignatura. El trabajo de a pares contribuye a la incorporación de los nuevos conocimientos.
2	De a pares	Objetivo: que el alumno aplique los conceptos fundamentales del paradigma de objetos para definir clases. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la definición de clases con sus constructores y métodos, considerando el encapsulamiento, e instanciando objetos y enviándole mensajes para verificar su correcta implementación, utilizando el lenguaje de programación y el IDE propuesto por la asignatura. El trabajo de a pares contribuye a la incorporación de los nuevos conocimientos.
3	Individual	Objetivo: que el alumno aplique los conceptos fundamentales del paradigma de objetos para implementar la interacción entre objetos y la reutilización de clases definidas previamente. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la definición de clases con constructores y métodos que incorporan otros objetos, e instanciando objetos que se envían mensajes entre sí, para verificar su correcta implementación, utilizando el lenguaje de programación y el IDE propuesto por la asignatura. El trabajo individual permite al alumno <b>evaluar</b> su autonomía para aplicar los nuevos conocimientos.
4	Individual	Objetivo: que el alumno aplique los conceptos del paradigma de objetos para implementar la interacción múltiple entre objetos, utilizando colecciones de objetos, y reutilizando clases predefinidas del lenguaje de programación. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la utilización de colecciones de objetos de clases con sus constructores y métodos que incorporan otros objetos (únicos o en colecciones), e instanciando objetos que se envían mensajes entre sí para verificar su correcta implementación. El trabajo individual permite al alumno <b>afianzar</b> su autonomía para aplicar los nuevos conocimientos.
5	Individual	Objetivo: que el alumno aplique los conceptos avanzados del paradigma de objetos para implementar la herencia y el polimorfismo en jerarquías de clases. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la implementación de la herencia y el polimorfismo en jerarquías de clases, definiendo adecuadamente los constructores dentro de la jerarquía, valorando la reutilización de código mediante el mecanismo de abstracción de la herencia, e instanciando objetos que se envían mensajes entre sí para verificar su correcta implementación.
6	Individual	Objetivo: que el alumno aplique conceptos de los lenguajes de programación orientado a objetos que contribuyen a la robustez y a la calidad del software, mediante el tratamiento de excepciones. Resolución de una Guía de Trabajos Prácticos de ejercicios que requieren la implementación del tratamiento de excepciones y la verificación de su correcta implementación y utilidad.
7	Grupal	Objetivo: que el alumno integre y aplique todos los conceptos del paradigma de programación orientado a objetos aprendidos y que trabaje en equipo. Resolución de un problema de mayor complejidad que requiere la aplicación de todos los conceptos aprendidos. La guía, además del enunciado del problema, contiene las consignas para el trabajo en equipo, que incluye la resolución de distintos módulos en subgrupos y la integración final de todos los módulos para la resolución completa del problema.

#### 3.2. Laboratorio / Trabajo de campo

Se implementan las mismas Guías de Trabajos Prácticos. Las clases se desarrollan en los laboratorios de informática. Los alumnos tienen a disposición computadoras para llevar a cabo la tarea.

