

MODALIDAD ACADÉMICA

Asignatura	Paradigmas de Programación	
Carrera	INGENIERÍA EN SISTEMAS DE INFORMACIÓN	
Ciclo Lectivo	2019	
Vigencia del programa	Desde el inicio del ciclo lectivo 2019	
Plan	2008	
Nivel <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1er. Nivel 2do. Nivel 3er. Nivel 4to. Nivel 5to. Nivel	
Coordinador de la Cátedra	Dr. Ing. Marcelo Marciszack	
Área de Conocimiento <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Programación Computación Sistemas de Información Gestión Ingenieril Modelos Complementaria	
Carga horaria semanal	8 Horas	
Anual/ cuatrimestral	Cuatrimestral	
Contenidos Mínimos, según Diseño Curricular-Ordenanza 1150 (sólo para asignaturas curriculares)	Concepto de Paradigmas de Programación. Paradigmas Fundamentales. Paradigma Funcional. Cálculo Lambda. Lenguajes de Programación Funcional. Paradigma Lógico. Lógica de Predicados de Primer Orden y Formas Restringidas. Regla inferencia Resolución. Lenguajes de Programación Lógica. Paradigma Orientado a Objetos. Conceptos Básicos. Clasificación, Clase y Objeto. Método y Mensaje. Clase Abstracta y Concreta. Herencia y Tipos de Herencia. Polimorfismo y Tipos de Polimorfismo en el Modelo de Objetos. Lenguajes de Programación Orientados a Objetos. Extensiones al Modelo Básico de Objeto en un Lenguaje Particular.	
Correlativas para cursar (según Diseño Curricular-Ordenanza 1150)	Regulares	Aprobadas
	<ul style="list-style-type: none"> • Matemática discreta • Algoritmos y estructuras de Datos 	
	Regulares	Aprobadas

<p>Correlativas para rendir (según Diseño Curricular-Ordenanza 1150)</p>	<ul style="list-style-type: none"> • Paradigmas de Programación 	<ul style="list-style-type: none"> • Matemática discreta • Algoritmos y estructuras de Datos
<p>Objetivos de la Asignatura</p>	<p><u>Objetivos generales</u></p> <p>Que el estudiante:</p> <ul style="list-style-type: none"> • Comprenda los fundamentos de los paradigmas básicos que son utilizados por los lenguajes de programación actuales. • Conozca el modelo formal o semiformal subyacente de cada paradigma y la forma en que el mismo es incorporado en un lenguaje de programación correcto. • Aplique los diferentes paradigmas en la resolución de problemas. <p><u>Resultados generales de aprendizajes:</u></p> <p>Demostrar capacidad en el dominio de los elementos conceptuales de los diferentes paradigmas de programación, como así también sus influencias en las formas de analizar y modelar soluciones a una amplia variedad de dominios de problemas, para justificar la aplicación entre uno u otro paradigma para su resolución.</p> <p>Manifestar conocimiento conceptual respecto al modelo formal o semiformal subyacente de cada paradigma de programación y la forma en que el mismo es incorporado en un lenguaje de programación para diseñar, implementar y evaluar modelos de soluciones que permitan resolver problemas de información en diferentes ámbitos de aplicación.</p> <p>Promover las habilidades de programación encaminadas a la resolución de problemas (ya iniciadas en la asignatura correlativa anterior, bajo el paradigma imperativo), para desarrollar soluciones eficientes a problemas ingenieriles del área de sistemas de información, dentro del paradigma que se haya considera el más apropiado para tal problema, utilizando lenguajes de programación puros.</p>	

Programa Analítico

Unidad Nro. 1: Introducción a los Lenguajes y Paradigmas de Programación.

Resultados de aprendizajes:

Evidenciar las características constitutivas de cada uno de los paradigmas y lenguajes de programación para asociarlos con diferentes prototipos de problemas que sean factibles de resolver con cada uno de ellos de la forma más eficiente.

Contenidos:

Introducción y conceptos generales: Programas, paradigmas, lenguajes de programación, y programación.
Paradigmas fundamentales: Clasificación y evolución histórica, definición, lenguajes asociados, ventajas, limitaciones y áreas de aplicación.

Diferencia entre lenguaje y paradigma de programación.

Lenguajes de programación: Conceptos, criterios de evaluación, reseña histórica y evolución y tipos de lenguajes: híbridos y puros.

Bibliografía Obligatoria

- Material correspondiente a esta Unidad elaborado por los docentes de la Cátedra y puesto a disposición de los alumnos en la UV

Bibliografía Complementaria:

- David A. Watt, William Findlay - 2004 - Programming Language Design Concepts
- Robert Sebesta - 2011 - Concepts of Programming Languages
- Ghezzi, Carlo - Mehdi Jazayeri - 1996 - Programming Languages Concepts - 3a Edición

Evaluación:

La evaluación sumativa, de los contenidos de la unidad, se realiza en el primer parcial, junto a las unidades temáticas N° 2, 3 y 4. Son preguntas de desarrollo sobre los temas trabajados de esta unidad.

Unidad Nro. 2: Elementos constitutivos de Lenguajes y Paradigmas

Resultados de Aprendizaje:

Reconocer los elementos constitutivos, mecanismos y formas de implementación de las características de los lenguajes de programación dentro del paradigma de programación al que corresponda, para generar la habilidad de aplicarlos eficientemente en la resolución de diferentes problemas ingenieriles del área de sistemas con los que lleguen a enfrenarse los estudiantes tanto en el ámbito académico como en el profesional.

Contenidos:

Conceptos lógicos y transversales.

Tipos de datos: Teoría, clasificación, verificación, sistema de tipos, conversión y ejemplos en los diferentes lenguajes de programación.

Mecanismos de control de flujo: Organización y ejemplos en los diferentes lenguajes de programación.

Abstracción y modularización: definición y mecanismos de implementación.

Bibliografía Obligatoria:

- Material correspondiente a esta Unidad elaborado por los docentes de la Cátedra y puesto a disposición de los alumnos en la UV

Bibliografía Complementaria:

- Apunte de la Cátedra P.P.R. -UTN FRC Tymoschuk/Serra/Castillo Ed. Educa
- Piensa en Java Bruce Heckel Ed. PEARSON Prentice Hall
- Lenguajes de Programación - Conceptos y Constructores Ravi Sethy Ed. Addison Wesley
- Compiladores. Análisis semántico y chequeo de tipos. Universidad Galileo
- Diferencias entre Paradigmas de Programación. Oscar Campos

Evaluación:

La evaluación sumativa, de los contenidos de la unidad, se realiza en el primer parcial, junto a las unidades

temáticas N° 1, 3 y 4. Son preguntas de desarrollo sobre los temas trabajados de esta unidad.

Unidad Nro. 3: Paradigma Imperativo: Secuencial o Estructurado.

Resultados de Aprendizaje:

Valorar las ventajas de la descomposición de un problema en subproblemas, y la correspondencia con la separación de código en funciones, para la construcción y verificación de programas escritos en lenguajes propios del paradigma imperativo.

Distinguir el impacto de la declaración de variables locales y globales en un programa estructurado para poder disponer de un control más preciso sobre el acceso a los datos durante su diseño e implementación.

Manifestar dominio en la aplicación de las diferentes estructuras de control para diseñar e implementar programas codificados con lenguajes de programación pertenecientes al paradigma estructurado, valorando las ventajas de su utilización.

Contenidos:

Introducción.

Programación estructurada.

Características generales: Variables locales y globales. Estructuras de control. Modularización (funciones y procedimientos).

Introducción al lenguaje C: Tipos de datos. Declaración de variables. Operadores. Procedimientos. Funciones. Estructuras de control. Funciones de entrada y salida.

Lenguaje asociado: Python / C++. Entornos Integrados de Desarrollo.

Bibliografía Obligatoria:

- Material correspondiente a esta Unidad elaborado por los docentes de la Cátedra y puesto a disposición de los alumnos en la UV

Bibliografía Complementaria:

- C/C++ Cómo Programar de Harvey M. Deitel y Paul J. Deitel. ISBN: 970-2605318.
- Java – Fundamentos de Programación Judy M. Bishop – Addison Wesley-
- Frittelli, V., et al. (2016). Publicaciones Cátedra de AED. Fichas de Estudio para AED (disponible en Aula Virtual AED). Cordoba: UTN – Facultad Regional Córdoba.
- Python Software Foundation. (2015). Python Documentation. Retrieved February 24, 2016, from <https://docs.python.org/3/>.

Evaluación:

La evaluación sumativa, de los contenidos de la unidad, se realiza en el primer parcial, junto a las unidades temáticas N° 1, 2, y 4. Son preguntas de desarrollo sobre los temas trabajados de esta unidad.

Unidad N° 4: Paradigma de Programación con Orientación a Objetos.

Resultados de aprendizajes:

Manifestar dominio en la aplicación de los mecanismos fundamentales de la programación orientada a objetos

para diseñar e implementar programas mediante lenguajes puros pertenecientes al paradigma orientado a objetos.

Comprender cómo mediante la colaboración de objetos a través del envío de mensajes entre sí y la delegación de responsabilidades, se pueden resolver problemas ingenieriles de diferentes niveles de complejidad del área de sistemas, para así diseñar, implementar y evaluar programas mediante el lenguaje de programación Smalltalk, aplicando todas las propiedades de este paradigma, y haciendo uso eficiente de las diferentes colecciones que este lenguaje soporta.

Contenidos:

Conceptos fundamentales. Abstracción de datos y ocultamiento de la información. Estructura de un objeto. Métodos y mensajes. Clasificación. Clase. Concepto de generalización-especialización. Composición Herencia: Estrategias y modelos. Polimorfismo. Polimorfismo: Definición, tipos.

Conceptos del modelo de objetos en SmallTalk. Desarrollo de la sintaxis de objetos en Smalltalk. Expresiones literales. Caracteres, secuencia de caracteres, símbolos y números. Expresiones de asignación y variables. Asignación. Tipos de variables. Variables privadas: de instancia nombradas e indexadas, argumentos y temporales. Variables compartidas: de clase, globales y pool. Variables especiales: self y super. Expresiones de mensaje. Sintaxis de un mensaje. Tipos de mensajes: unario, binario y palabra clave. Orden de precedencia en la evaluación de expresiones. Expresión de mensajes en cascada. Expresiones de bloque. Clase Context. Bloques con y sin argumentos. Evaluación de bloques. Métodos y expresiones de retorno. Sintaxis de la definición de un método. Significado de la expresión de retorno. Métodos de clase e instancia. Implementación de Composición en Smalltalk. Implementación de Herencia en Smalltalk (Definición de una Subclase, uso de super, herencia de variables, inicialización de atributos en una clase Hija, herencia de métodos, clases abstractas). Implementación de Polimorfismo en Smalltalk. Colecciones en Smalltalk: Introducción, jerarquía, colecciones básicas (Set, Bag, OrderedCollection, SortedCollection, Array, Dictionary), operaciones básicas, conversión entre colecciones.

Lenguaje asociado: Smalltalk

Entorno asociado: Pharo. Imagen, ambiente de objetos, definición y uso de clases y objetos.

Bibliografía Obligatoria:

- Material correspondiente a esta Unidad elaborado por los docentes de la Cátedra y puesto a disposición de los alumnos en la UV

Bibliografía Complementaria:

- Pharo por Ejemplo Andrew P. Black Stéphane Ducasse Oscar Nierstrasz Damien Pollet <http://pharobyexample.org/es/PBE1-sp.pdf>
- Inside Smalltalk. Vol. I, II: Lalonde y J. Pugh; Prentice Hall International; 1991.
- Adele Goldberg and David Robson , Smalltalk 80- The Language.
- Wirfs- Brock, R. y otros, Designing Object-Oriented Software, Prentice Hall
- Tomas Khum – “Estructura de las Revoluciones Científicas” .
- Notas de clase de Tópicos II, curso de la Maestría en Ingeniería de Software UNLP, dictada en 2002 por Máximo Prieto. Documentos de Juan Carlos Vazquez.
- Apunte: Introducción a la orientación a objetos de la Asignatura Paradigmas de Programación UTN – FRBA Autores: Carlos Lombardi – Nicolás Passerini.

Evaluación:

La evaluación sumativa, de los contenidos de la unidad, se realiza en el primer parcial, junto a las unidades temáticas N° 1, 2, y 3. Son preguntas de desarrollo sobre los temas trabajados de esta unidad.

Los contenidos prácticos se evaluarán en máquina donde se solicitará la codificación de la solución propuesta para un determinado problema, o la modificación o ampliación de determinadas partes de una solución

previamente planteada.

El alumno en forma grupal, deberá resolver y presentar en tiempo y forma el Práctico Integrador N° 1 correspondiente a esta Unidad.

Unidad N° 5: Paradigma de Programación Funcional.

Resultados de aprendizajes:

Demostrar dominio en la aplicación de los principios constitutivos y filosóficos que dan origen al paradigma funcional, para poder diseñar, implementar programas codificados mediante lenguajes puros pertenecientes a este paradigma.

Comprender cómo haciendo uso del concepto de funciones, bajo los principios constitutivos del paradigma funcional, se pueden resolver problemas ingenieriles para diferentes ámbitos de aplicación, con la finalidad de diseñar, implementar y evaluar programas mediante el lenguaje Haskell, aplicando en forma eficiente el gran poder expresivo que éste proporciona.

Contenidos:

Introducción. Historia. Características. Ventajas/Desventajas. Áreas de Aplicación. Ejemplos de implementaciones. Familia de Lenguajes. Conceptos generales: Funciones matemáticas, Sintaxis en el paradigma funcional. Abstracción Funcional. Funciones de orden superior. Cálculo Lambda, evaluación postergada.

Lenguaje Haskell. Introducción. Entorno de Haskell – HUGS. Sintaxis. Comentarios. Tipos de datos. Sistemas de inferencia de tipos. Flujo de control. Definición de Funciones. Currificación. Reducción de expresiones. Evaluación. Tuplas. Recursividad. Listas: definición, listas por comprensión. Tipos definidos por el usuario. Tipos polimórficos.

Lenguaje asociado: Haskell

Entorno asociado: WinHugs

Bibliografía Obligatoria:

- Material correspondiente a esta Unidad elaborado por los docentes de la Cátedra y puesto a disposición de los alumnos en la UV

Bibliografía Complementaria:

- Richard Bird. "Introduction to Functional Programming using Haskell". Prentice Hall International, 2nd Ed. New York, 1997
- Luca Cardelli. "Basic Polymorphic Typechecking". AT&T Bell Laboratories, Murray Hill, NJ 07974. 1998.
- Paul Hudak. "A Gentle Introduction to Haskell 98". Yale University. Department of Computer Science. John Peterson. 1999.
- Paul Hudak. "The Haskell School of Expression: Learning Functional Programming through Multimedia". Yale University. Cambridge University Press, New York, 2000.
- Simon Thompson. "Haskell: The Craft of Functional Programming". 3rd Edition. ISBN: 9780201882957. 1999.
- Jose E. Labra G. "Introducción al lenguaje Haskell". Universidad de Oviedo, Departamento de Informática, Octubre 1998.
- Kurt Normark. "Funcional Programming in Scheme". Department of Computer Science, Aalborg University, Denmark. 2003.

- Richard Kelsey, William Clinger and Jonathan Rees. "Revised Report on the Algorithmic Language Scheme". Higher-Order and Symbolic Computation, Vol. 11, No. 1, August 1998, pp. 7--105.
- Barendregt, Hendrik Pieter. "The Lambda Calculus: Its Syntax and Semantics, Studies in Logic and the Foundations of Mathematics". 103 (Revised ed.), North Holland, Amsterdam. [Corrections](#). 1984.
- Carlos Varela. "Programming Languages". Rennselaer Polytechnic Institute. USA. 2011.
- Lucas Spigariol. "Paradigma Funcional". Paradigmas de Programación - FRBA – UTN. 2007.
- Alonso Jimenez, José. "Programación Declarativa: Definición de listas por comprensión". Departamento de Ciencias de la Computación e I. A. Universidad de Sevilla. 2010.

Evaluación:

La evaluación sumativa, de los contenidos de la unidad, se realiza en el segundo parcial, junto a la unidad temática N° 6. Son preguntas de desarrollo sobre los temas trabajados de esta unidad.

Los contenidos prácticos se evaluarán en máquina donde se solicitará la codificación de la solución propuesta para un determinado problema, o la modificación o ampliación de determinadas partes de una solución previamente planteada.

El alumno en forma grupal, deberá resolver y presentar en tiempo y forma el Práctico Integrador N° 2 correspondiente a esta Unidad.

Unidad N° 6: Paradigma de Programación Lógico.

Resultados de aprendizajes:

Demostrar dominio de la aplicación de los principios constitutivos y filosóficos que dan origen al paradigma lógico, para poder diseñar, implementar y evaluar programas codificados mediante lenguajes puros pertenecientes a este paradigma.

Comprender cómo mediante el motor de inferencia, intérpretes deterministas y no deterministas, y todos los mecanismos propios de este paradigma, se pueden resolver problemas de diferentes ámbitos de aplicación, para así diseñar e implementar programas mediante el lenguaje de programación Prolog, aplicando predicados y cláusulas de primer orden mediante razonamientos y silogismos.

Contenidos:

Introducción a la Programación Lógica. Fundamentación lógica. Predicados y términos. Razonamientos y silogismos. Relaciones, hechos y reglas. Consultas. Tipos de consultas. Definición de programa en Paradigma Lógico. Motor de inferencia, ubicación del control en un programa lógico. Interpretación algorítmica: Procedimientos y programación. Intérprete no determinista. Estrategia de evaluación. PROLOG Intérprete determinístico, "backtracking". Orden de evaluación de cláusulas. Terminación. Sintaxis PROLOG. Cláusulas, predicados y términos. Distintos tipos de datos. Recursión en PROLOG. Tipos de datos recursivos, lista. Concepto de variable o incógnita. Unificación. Múltiples resultados. Inversibilidad. Aritmética, evaluación de expresiones aritméticas. Negación. Functores.

Lenguaje asociado: Prolog.

Entorno asociado: Win Prolog.

Bibliografía Obligatoria:

- Material correspondiente a esta Unidad elaborado por los docentes de la Cátedra y puesto a disposición de los alumnos en la UV

Bibliografía Complementaria:

- Tasistro A., J. Vidart. "Programación Lógica y funcional". EBAI (1988).
- Tutorial Prolog Universidad de Oviedo
<http://www.fdi.ucm.es/profesor/evah/IAIC/prolog/PPProlog.pdf>
- Apunte de la asignatura Paradigmas de Programación UTN-FRC disponible en el Sitio: Labsys.frc.utn.edu.ar, Sitios de las Cátedra, Paradigmas de Programación.

Evaluación:

La evaluación sumativa, de los contenidos de la unidad, se realiza en el segundo parcial, junto a la unidad temática N° 5. Son preguntas de desarrollo sobre los temas trabajados de esta unidad.

Los contenidos prácticos se evaluarán en máquina donde se solicitará la codificación de la solución propuesta para un determinado problema, o la modificación o ampliación de determinadas partes de una solución previamente planteada.

El alumno en forma grupal, deberá resolver y presentar en tiempo y forma el Práctico Integrador N° 3 correspondiente a esta Unidad.

<p>Metodología de enseñanza y aprendizaje</p>	<p>Los contenidos teóricos que serán desarrollados y trabajados con el docente de teórico, con el fin de conceptualizar términos a utilizar en el desarrollo de las actividades prácticas. Las clases se trabajan con material elaborado en diapositivas que comparte la cátedra, iniciando siempre con revisión de los temas anteriores para integrarlos a cada clase y logrando unificación de contenidos entre todos los cursos.</p> <p>Ejercitación de casos a través de ejercicios prácticos contenidos en material unificado, elaborado por los docentes de la cátedra, además de prácticos que los auxiliares docentes realizan durante el cursado.</p>
<p>Sistema de evaluación</p>	<p>Todos los alumnos son evaluados simultáneamente en dos exámenes parciales en fechas unificadas para todos los cursos (parte práctica), disponiéndose en ambos casos de fechas alternativas para alumnos con problemas laborales o motivos religiosos.</p> <p>Las fechas y alcance de todos los parciales son definidos y anunciados al comenzar el cuatrimestre y publicados en la Web.</p> <p>La preparación de los temas de los parciales está a cargo de los Docentes de cada curso, en forma rotativa, designados al comenzar el año académico. Estos Docentes son también responsables de proponer las soluciones y los criterios específicos de corrección. Todos los alumnos son examinados con los mismos temas y evaluados con criterios de corrección comunes a todos los cursos.</p> <p>Cada examen parcial consiste en dos partes, teórica y práctica, que son corregidos y evaluadas por separado.</p> <p>La modalidad de evaluación de contenidos teóricos será realizada a través de un instrumento con opciones de respuestas de opciones múltiples, realizado íntegramente en máquina, el cual se le asignará un tiempo de 40 minutos para 25 preguntas generadas aleatoriamente, al finalizar se entrega un reporte con la nota obtenida por el alumno.</p> <p>La modalidad de evaluación de contenidos prácticos se realizará en máquina donde dependiendo de los aspectos a evaluar y el paradigma, se solicitará la codificación de la solución propuesta para un determinado problema, o la modificación o ampliación de determinadas partes de una solución previamente planteada. Esta corrección se</p>

	<p>realizará de acuerdo a criterios de corrección previamente informados.</p> <p>En caso de que el instrumento de evaluación del práctico, ya sea parcial o final incluya dos paradigmas diferentes, el alumno para aprobar el mismo, deberá aprobar como mínimo un 50 % de cada paradigma evaluado.</p>																																	
<p>Regularidad: condiciones</p>	<p>Escala de notas de regularidad(*)</p> <table border="1" data-bbox="568 629 1281 1025"> <thead> <tr> <th>NOTAS</th> <th>PORCENTAJE</th> <th>CALIFICACIÓN</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td>No Aprobado</td> </tr> <tr> <td>2</td> <td></td> <td>No Aprobado</td> </tr> <tr> <td>3</td> <td></td> <td>No Aprobado</td> </tr> <tr> <td>4</td> <td>55% a 57%</td> <td>Aprobado</td> </tr> <tr> <td>5</td> <td>58% a 59%</td> <td>Aprobado</td> </tr> <tr> <td>6</td> <td>60% a 68%</td> <td>Aprobado</td> </tr> <tr> <td>7</td> <td>69% a 77%</td> <td>Aprobado</td> </tr> <tr> <td>8</td> <td>78% a 86%</td> <td>Aprobado</td> </tr> <tr> <td>9</td> <td>87% a 95%</td> <td>Aprobado</td> </tr> <tr> <td>10</td> <td>96% a 100%</td> <td>Aprobado</td> </tr> </tbody> </table> <p>(*) Escala acordada en reunión de Docentes Coordinadores</p> <p>Es requisito para regularizar el curso, la regularidad de asistencia y con cinco (5) notas aprobadas; dos correspondientes a los parciales teóricos y las otras dos a los prácticos y una adicional que se corresponderá con la valoración integral de tres prácticos integradores (Uno por cada uno de los Paradigmas Objeto, Lógico y Funcional).</p> <p><u>Alumno Abandono:</u> Se considera en esta situación a todo alumno que no asistió nunca a clases o abandonó el cursado; es decir no concurrió todas las instancias de evaluación.</p> <p><u>Alumno Libre:</u> Quienes no cumplieron las condiciones establecidas para alumno Regular, habiéndose presentado en todas las instancias normales de evaluación como en la instancia de recuperación previstas.</p> <p><u>Alumno Regular:</u> Para alcanzar la condición de Alumno Regular, las cinco notas deben estar aprobadas: En el caso de evaluaciones parciales, tanto teóricas como prácticas deben superar una calificación mínima de cuatro (4) puntos, que corresponde a una evaluación del 55%. La quinta nota surge del promedio de las notas obtenidas por aprobación de los tres trabajos integradores.</p> <p>ACLARACION: El estudiante en condición de regular puede rendir en el plazo de un ciclo lectivo sin control de correlativas aprobadas.</p> <p><u>Instancia de recuperación</u></p>	NOTAS	PORCENTAJE	CALIFICACIÓN	1		No Aprobado	2		No Aprobado	3		No Aprobado	4	55% a 57%	Aprobado	5	58% a 59%	Aprobado	6	60% a 68%	Aprobado	7	69% a 77%	Aprobado	8	78% a 86%	Aprobado	9	87% a 95%	Aprobado	10	96% a 100%	Aprobado
NOTAS	PORCENTAJE	CALIFICACIÓN																																
1		No Aprobado																																
2		No Aprobado																																
3		No Aprobado																																
4	55% a 57%	Aprobado																																
5	58% a 59%	Aprobado																																
6	60% a 68%	Aprobado																																
7	69% a 77%	Aprobado																																
8	78% a 86%	Aprobado																																
9	87% a 95%	Aprobado																																
10	96% a 100%	Aprobado																																

	<p>Se dispondrá de una instancia unificada para recuperación de cada una de evaluaciones parciales, ya sean teóricas o prácticas. Para dar cumplimiento a estas instancias, se dispondrán de dos fechas diferentes: una para las evaluaciones del primer parcial, y otra para las evaluaciones del segundo parcial, ya sean de prácticas o teóricas.</p> <p>El alumno dispondrá de una instancia de recuperación para cada uno de los tres prácticos integradores, que consistirá en la reentrega del trabajo práctico previamente evaluado y con las correcciones efectuadas, al cual se le adicionará alguna funcionalidad. La fecha máxima para dar cumplimiento a la entrega definitiva de los prácticos integradores será la última clase de práctico de la última semana de clase.</p> <p>NOTA: Para cada una de las 5 (cinco) instancias de evaluación, se tomará la mayor nota obtenida, ya sea en la instancia de recuperación o en la inicialmente obtenida.</p>
<p>Promoción: condiciones</p>	<p><u>Alumno con Prácticos Promovidos:</u> En caso de que el alumno obtenga entre los exámenes parciales prácticos, y la nota promedio correspondiente a los prácticos integradores un promedio igual o superior a 8 (ocho) con notas no inferiores a 7 (siete) entre los dos parciales y el promedio de los prácticos integradores con los tres prácticos aprobados, y los exámenes teóricos ambos aprobados, se le da por aprobado el contenido práctico de la materia, y solo debe rendir un examen final teórico.</p> <p><u>Acceso a través de la instancia de Recuperación:</u> Esta condición puede ser alcanzada a través de haber rendido en las fechas estipuladas, o por haber accedido a las instancias de recuperación (con los mismos requisitos de cantidad de exámenes habilitados de recuperación para obtener la regularización de la asignatura).</p> <p><u>Pérdida de promoción de Prácticos</u></p> <p>La vigencia de la promoción de prácticos es de un 1 (un) año lectivo, la cual se contará a partir de la finalización del cursado, o lo que es equivalente a 10 turnos de exámenes, independiente del cuatrimestre de cursado de la asignatura, vencido este plazo conservará la condición de alumno regular y cuyo vencimiento o pérdida de regularidad estará sujeta al cumplimiento establecido en la ordenanza vigente del reglamento de estudios.</p> <p>El alumno pierde la condición de prácticos promovidos, cuando es aplazado en instancia de rendido de la asignatura en el primer examen que se presente, continuando en la condición de alumno regular.</p>
<p>Aprobación Directa: condiciones.</p>	<p><u>Alumno aprobación Directa:</u> Si el alumno alcanzó la promoción de prácticos, y además el promedio de las instancias de evaluación de Teóricos es de 8 (ocho) o superior con ninguna nota inferior a 7 (ocho) obtiene la aprobación directa de la asignatura.</p> <p>ACLARACIONES PARA EL ALUMNO:</p> <ul style="list-style-type: none"> • La calificación será la nota registrada como Nota Final en Autogestión.

	<ul style="list-style-type: none"> Se sugiere informar al alumno, que en esta condición puede registrar su nota en examen en el plazo de un ciclo lectivo, sin control de correlativas aprobadas, y después de ello se le exigirán correlativas aprobadas 																																	
<p>Modalidad de examen final</p>	<p>Escala de Notas para Examen Final (*)</p> <table border="1" data-bbox="568 613 1197 1010"> <thead> <tr> <th>NOTA</th> <th>PORCENTAJE</th> <th>CALIFICACIÓN</th> </tr> </thead> <tbody> <tr><td>1</td><td></td><td>Insuficiente</td></tr> <tr><td>2</td><td></td><td>Insuficiente</td></tr> <tr><td>3</td><td></td><td>Insuficiente</td></tr> <tr><td>4</td><td></td><td>Insuficiente</td></tr> <tr><td>5</td><td></td><td>Insuficiente</td></tr> <tr><td>6</td><td>60% a 68%</td><td>Aprobado</td></tr> <tr><td>7</td><td>69% a 77%</td><td>Bueno</td></tr> <tr><td>8</td><td>78% a 86%</td><td>Muy Bueno</td></tr> <tr><td>9</td><td>87% a 95%</td><td>Distinguido</td></tr> <tr><td>10</td><td>96% a 100%</td><td>Sobresaliente</td></tr> </tbody> </table> <p>(*) Escala acordada en reunión de Docentes Coordinadores</p> <p>El examen final se tomará en un único horario y será unificado para toda la cátedra.</p> <p>Horario inicio Examen Práctico: 15:00 Horas Horario Inicio Examen Teórico: 17:00 Horas</p> <p>El examen final para alumnos en condición de Regular, comienza con una instancia práctica en máquina, la misma consta de evaluación de una solución de un problema determinado, codificado y resuelto, el cuál será de carácter eliminatorio.</p> <p>Los alumnos que aprobaron la parte práctica de la evaluación, y los alumnos de prácticos promovidos pasarán a una próxima instancia de evaluación de aspectos teóricos de la asignatura, para lo cual responderán en máquina un cuestionario de opción múltiple de características similares a las evaluaciones de parciales teóricas, pero abarcando la totalidad del programa.</p> <p>De acuerdo al porcentaje obtenido en esta instancia pueden surgir las siguientes alternativas:</p> <ul style="list-style-type: none"> Porcentaje inferior al 60 %: El alumno es desaprobado y finaliza el examen. Porcentaje entre el 60 y el 79 %: El alumno continúa el examen pasando a una instancia oral donde se le dan tres temas para desarrollar en papel y explicarlo en forma satisfactoria para aprobar el examen. Porcentaje igual o superior al 80 %: El examen finaliza con la aprobación del alumno. <p><u>Calificación Final</u></p> <p>Para calificar los exámenes se utilizará el siguiente criterio:</p>	NOTA	PORCENTAJE	CALIFICACIÓN	1		Insuficiente	2		Insuficiente	3		Insuficiente	4		Insuficiente	5		Insuficiente	6	60% a 68%	Aprobado	7	69% a 77%	Bueno	8	78% a 86%	Muy Bueno	9	87% a 95%	Distinguido	10	96% a 100%	Sobresaliente
NOTA	PORCENTAJE	CALIFICACIÓN																																
1		Insuficiente																																
2		Insuficiente																																
3		Insuficiente																																
4		Insuficiente																																
5		Insuficiente																																
6	60% a 68%	Aprobado																																
7	69% a 77%	Bueno																																
8	78% a 86%	Muy Bueno																																
9	87% a 95%	Distinguido																																
10	96% a 100%	Sobresaliente																																

	<ul style="list-style-type: none"> • Alumnos promoción Total: Promedio de las cinco notas del cursado. • Alumnos Prácticos Promovidos: Nota obtenida en el examen final teórico. • Alumnos Regulares: Promedio entre la nota obtenida en el examen final teórico y el examen final práctico.
Actividades en laboratorio	Todas las actividades prácticas, se desarrollarán en laboratorio asociado, con la utilización del lenguaje de programación acorde al paradigma.
Horas/año totales de la asignatura (hs. cátedra)	128
Cantidad de horas prácticas totales (hs. cátedra)	64
Cantidad de horas teóricas totales (hs. cátedra)	64
Tipo de formación práctica <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Formación experimental Resolución de problemas de ingeniería Actividades de proyecto y diseño Prácticas supervisadas en los sectores productivos y /o de servicios
Cantidad de horas cátedras afectadas a la formación práctica indicada en el punto anterior	64
Descripción de los prácticos	<p><u>Prácticos Áulicos</u></p> <p>En el desarrollo de la asignatura, para cada uno de los paradigmas se realizarán actividades prácticas de laboratorios en cada uno de los lenguajes de programación.</p> <p><u>Prácticos grupales integradores</u></p> <p>Se deberán desarrollar y presentar en tiempo y forma 3 (tres) prácticos integradores, uno para cada uno de los siguientes paradigmas: orientado a objetos, lógico y funcional.</p> <p>Las fechas de presentación de los mismos estarán estipuladas en el cronograma de actividades propio del cuatrimestre y del ciclo lectivo, el cuál será publicado junto con las fechas de parciales, en oportunidad de dictado del primer día de clase, durante la presentación de la asignatura.</p> <p>Es condición necesaria para regularizar la materia que el alumno apruebe los tres trabajos prácticos integradores en tiempo y forma, o en su instancia de recuperación respectiva.</p>
Criterios generales	Los trabajos prácticos integradores deberán ser desarrollados en grupos de hasta 3 (tres) alumnos.

El medio de entrega de los trabajos prácticos integradores será a través del aula virtual, en el link habilitado específicamente para tal fin.

Es obligación de cada integrante del grupo subir la resolución correspondiente al trabajo práctico integrador en el link específicamente habilitado en el aula virtual para tal fin. Si algún miembro del grupo no lo sube por cualquier motivo, se considerará que ese alumno no realizó ni presentó el trabajo práctico en cuestión.

La entrega de todos los trabajos prácticos integradores tendrá especificada una fecha y hora de vencimiento.

La entrega del trabajo práctico integrador por parte de los integrantes del grupo deberá constar de un único archivo comprimido que contenga lo siguiente: enunciado y resolución correspondiente.

El archivo comprimido deberá ser nombrado de la siguiente manera TP_0X_ApellidoLegajo1_ApellidoLegajo2_ApellidoLegajo3 (X representa el número del trabajo práctico). Todos los integrantes del grupo deberán nombrar y subir el archivo comprimido exactamente igual.

Criterio de Evaluación:

En el enunciado de cada trabajo práctico integrador se describirán los criterios de evaluación específicos que se tendrán en cuenta para la corrección del mismo.

Cada trabajo práctico integrador entregado será calificado con una nota entera comprendida entre 1 y 10.

<p>Cronograma de actividades de la asignatura (contemplando las fechas del calendario 2019 y para cada unidad), dentro de cada cuatrimestre de dictado.</p>	<p>Planificación basada en 15 semanas lectivas, según calendario académico 2019 para el primer semestre.</p>		
	Semana- Fecha	Actividades Teóricas	Actividades Prácticas
	<p>1 18/03/19</p>	<p>Introducción a la Asignatura. y Unidad N° 1.</p>	<p>Ambientación en laboratorio, Características de los diferentes IDE, Acceso y ejecución. Presentación Parte Práctica. Unidad N° 3 Repaso Paradigma Estructurado Paradigma Estructurado, conceptos teóricos generales. Introducción y resolución de problemas con lógica estructurada, utilizando el lenguaje de programación C/C++ o Python.</p>
	<p>2 25/03/19</p>	<p>Unidad N° 2 y Unidad N° 3 Elementos constitutivos de los lenguajes y paradigmas. Tipos de datos, control de flujo, declaraciones. Mecanismos de implementación.</p>	<p>Unidad N°4: Introducción al Lenguaje Smalltalk. Características, aspectos y sintaxis. Presentación entorno Pharo, ventanas, browser, inspecciones. Inspección de clases Smalltalk. Biblioteca de Smalltalk: clases Boolean, Integer, etc. Evaluación de expresiones.</p>
	<p>3 01/04/19 Feriado Martes 2/4</p>	<p>Unidad N°4: POO Introducción al paradigma. Conceptos generales. Características y aplicaciones. Clases: sintaxis y definición en Smalltalk. Objetos y Tipos de Mensajes. Bloques y abstracciones de control. Definición de una clase simple, con creación e invocación de objetos.</p>	<p>Unidad N°4: Creación de clase definidas por el usuario. Bloques: ejercicios con bloques. Presentación Práctico N° 1.</p>
<p>4 08/04/19</p>	<p>Unidad N° 4 Relaciones entre clases: Composición. Presentación de diferentes modelos. Presentación de colecciones: mensajes de creación, de inserción y recorrido simple (mensaje do:).(sugerencia OrderedCollection)</p>	<p>Unidad N°4: Ejercitación con composición y colecciones. Otros mensajes de recorridos en colecciones (detect y select).</p>	
<p>5 15/04/19 Feriados por Semana Santa 18 y 19</p>	<p>Unidad N° 4: POO Relaciones entre clases: Herencia. Introducción a Polimorfismo. Presentación de diferentes modelos.</p>	<p>Unidad N°4: Ejercitación con Herencia y polimorfismo y colecciones.</p>	

	6 22/04/19	Unidad N° 4: POO Continuación de Polimorfismo. Presentación de la jerarquía de clases y de los métodos polimórficos de colecciones de Smalltalk.	Unidad N°4: Ejercitación con Herencia, Polimorfismo y Colecciones. Entrega Práctico N° 1
	7 29/04/19 Feriado miércoles 1/05	Primer Parcial - Parte teórica Repaso para parcial práctico	Unidad N°4: Ejercitación con Herencia, polimorfismo y Colecciones. Devolución Práctico N° 1
	Sábado 04/05/19	Primer parcial práctico	
	8 06/05/19	Unidad N° 5: Paradigma funcional Introducción, características y conceptos generales. (Funciones, Cálculo Lambda). Introducción a Haskell. Sintaxis. Tipos de datos, funciones y expresiones. Expresiones If-Then-Else, Case, guardas, definiciones locales.	Unidad N° 5: Introducción a Haskell Evaluación de expresiones simples, composición de funciones. Presentación de Práctico N° 2
	9 13/05/19	Unidad N° 5: Expresiones recursivas y listas.	Unidad N° 5: Ejercitación con expresiones If-Then-Else, Case, guardas, definiciones locales. Ejercitación con Listas y recursividad.
	10 20/05/19	Unidad N° 5: Tuplas. Tipos definidos por el usuario y tipos polimorfismos.	Unidad N° 5: Ejercitación con listas.
	11 27/05/19	Unidad N° 6: Paradigma Lógico Introducción – Lógica Predicados de primer orden. Reglas, hechos, cláusulas de Horn, predicados. Sintaxis y estructura de Prolog.	Unidad N° 6: Entorno y Sintaxis de Prolog. Orden de evaluación. Cláusulas simples. Entrega del Práctico N° 2. Presentación de Práctico N° 3.
	12 03/06/19	Unidad N° 6: Motor de Inferencia. Unificación, Backtracking, corte y fallo. Introducción a recursividad y a listas. Objetos Compuestos.	Unidad N° 6: Ejercitación de unificación, relaciones entre tablas. Objetos Compuestos: Ejercitación de objetos compuestos y listas. Devolución de Práctico N° 2.
	13 10/06/19	Unidad N° 6: Continuación de recursividad y de listas. Ejercitación y consultas para el parcial.	Unidad N° 6: Ejercitación de recursividad y listas. Caso de estudio: Repaso para el parcial. Entrega del Práctico N° 3. Antes de la hora de clase Devolución en clase de Práctico N° 3, con Autocorrección en el aula.
	Sábado 15/06/19	Segundo parcial práctico	
	14 17/06/19 Feriados	Segundo Parcial teórico	Ejercitación para recuperatorio parte práctica. Cierre de notas de Trabajos

<p>Propuesta para la atención de consultas y mail de contacto.</p>	<p>Las actividades de consulta previa a cada instancia de evaluación ya sea de parciales o de entrega de prácticos, se realizarán en espacio destinado a tal fin dentro del horario de cursado de la asignatura.</p> <p>Uso de email y del foro del aula virtual, todo el año. No existe un cronograma que limite o restrinja las consultas de alumnos.</p>
<p>Plan de integración con otras asignaturas</p>	<p>La asignatura Paradigmas de Programación, se encuentra en el segundo nivel de la implementación de la carrera. Con respecto a la ubicación dentro del área, previamente ya se han dictado los contenidos de otras asignaturas del área tales como: Matemática Discreta, Algoritmos y estructura de datos y Sintaxis y semántica de los Lenguajes, y luego de Paradigmas de Programación solo resta de dictado de asignaturas del área a Gestión de Datos.</p> <p>Por lo tanto, esta asignatura tiene por misión el de sentar sólidas bases en lo que respecta a los lenguajes y paradigmas de programación e integrar los conocimientos ya vistos en las otras materias del área. En Matemática discreta, se han abordado métodos inductivos, deductivos y recursivos en la resolución de situaciones problemáticas y demostraciones matemáticas y la de comprender los conceptos y procedimientos necesarios para resolver relaciones de recurrencia. En Sintaxis y semántica del Lenguajes, se han visto conceptos relacionados con informática teórica y máquinas abstractas. En Algoritmos y estructura de datos, conceptos de algoritmos y resolución de problemas. Todos los conocimientos descriptos de las 3 asignaturas anteriores resultan fundamentales para el desarrollo de Paradigmas de Programación.</p> <p>Posteriormente en el 3er y 4to nivel de la carrera, los alumnos disponen de asignaturas electivas en el área programación, las cuales tienen por objetivo el de brindar a los alumnos el aprendizaje y utilización de lenguajes de programación utilizados en el ámbito comercial, pudiendo optar por la rama de programación visual, con tecnologías de .Net o a través del uso de Java. Por cualquier de las ramas que el alumno opte, deberá contar con conocimientos de los Paradigmas, los lenguajes, mecanismos y formas de implementación vistos en Paradigmas de Programación.</p>
<p>Bibliografía Obligatoria</p>	<ul style="list-style-type: none"> • El Material de estudio para cada una de las unidades temáticas se encuentra editado, compilado y publicado en el sitio de la Universidad Virtual, el cual se encuentra disponible para cada alumno inscripto al cursado de la asignatura. • También están disponibles las filminas para el dictado de teóricos y prácticos, conjuntamente con guía de prácticos de cada unidad.
<p>Bibliografía Complementaria</p>	<ul style="list-style-type: none"> • David A. Watt, William Findlay - 2004 - Programming Language Desing Concepts • Robert Sebesta - 2011 - Concepts of Programming Languages • Ghezzi, Carlo - Mehdi Jazayeri - 1996 - Programming Languages Concepts - 3a Edición • Apunte de la Cátedra P.P.R -UTN FRC Tymoschuk/Serra/Castillo Ed. Educa • Piensa en Java Bruce Heckel Ed. PEARSON Prentice Hall • Lenguajes de Programación - Conceptos y Constructores Ravi Sethy Ed. Addison Wesley • Compiladores. Análisis semántico y chequeo de tipos. Universidad Galileo • Diferencias entre Paradigmas de Programación. Oscar Campos • C/C++ Cómo Programar de Harvey M. Deitel y Paul J. Deitel. ISBN: 970-

	<p>2605318.</p> <ul style="list-style-type: none"> • Java – Fundamentos de Programación Judy M. Bishop – Addison Wesley- • Inside Smalltalk. Vol. I, II: Lalonde y J. Pugh; Prentice Hall International; 1991. • Adele Goldberg and David Robson , Smalltalk 80- The Language. • Wirfs- Brock, R. y otros, Designing Object-Oriented Software, Prentice Hall • Tomas Khum – “Estructura de las Revoluciones Científicas” . • Apunte: Introducción a la orientación a objetos de la Asignatura Paradigmas de Programación UTN – FRBA Autores: Carlos Lombardi – Nicolás Passerini. • Tasistro A., J. Vidart. "Programación Lógica y funcional". EBAI (1988). • Tutorial Prolog Universidad de Oviedo http://www.fdi.ucm.es/profesor/evah/IAIC/prolog/PPPProlog.pdf • Apunte de la asignatura Paradigmas de Programación UTN-FRC disponible en el Sitio: Labsys.frc.utn.edu.ar, Sitios de las Cátedra, Paradigmas de Programación. • Richard Bird. “Introduction to Functional Programming using Haskell”. Prentice Hall International, 2nd Ed. New York, 1997 • Luca Cardelli. “Basic Polymorphic Typechecking”. AT&T Bell Laboratories, Murray Hill, NJ 07974. 1998. • Paul Hudak. “A Gentle Introduction to Haskell 98”. Yale University. Department of Computer Science. John Peterson. 1999. • Paul Hudak. “The Haskell School of Expression: Learning Functional Programming through Multimedia”. Yale University. Cambridge University Press, New York, 2000. • Simon Thompson. “Haskell: The Craft of Functional Programming”.3rd Edition. ISBN: 9780201882957. 1999. • Jose E. Labra G. “Introducción al lenguaje Haskell”. Universidad de Oviedo, Departamento de Informática, Octubre 1998. • Kurt Normark. “Funcional Programming in Scheme”. Department of Computer Science, Aalborg University, Denmark. 2003. • Richard Kelsey, William Clinger and Jonathan Rees. "Revised Report on the Algorithmic Language Scheme". Higher-Order and Symbolic Computation, Vol. 11, No. 1, August 1998, pp. 7--105. • Barendregt, Hendrik Pieter. “The Lambda Calculus: Its Syntax and Semantics, Studies in Logic and the Foundations of Mathematics”. 103 (Revised ed.), North Holland, Amsterdam. Corrections. 1984. • Ken Slonneger y Barry L. Kurtz. “Formal Syntax and Semantics of Programming Languages: A Laboratory-Based Approach”. Addison-Wesley, Reading, Massachusetts. 1995. • Carlos Varela. “Programming Languages”. Rensselaer Polytechnic Institute. USA. 2011.
--	--

Distribución de docentes por curso y cuatrimestre	Curso	C	Turno	Días y horas	Docente	Auxiliar	Ayudante
	2K1	2	Mañana	Mar 4,5,6,7 Práctico Jue 1,2,3,4 Teórico	CORSO, Cynthia	STEFFOLANI, Felipe	POLLIOTTO, Martín
	2K2	2	Mañana	Mie 1 Jue 1,2,3 Teóricos Vie 1,2,3,4 Práctico	LIGORRIA, Laura	COLACIOPPO Nicolás	SERRANO, Diego
	2K3	2	Mañana	Lun 1,2,3,4 Práctico Mie 4,5,6,7 Teórico	CORSO, Cynthia	STEFFOLANI, Felipe	FRIAS, Pablo
	2K4 (contrac.)	1	Mañana	Mie 1,2,3,4 Teórico Jue 4,5,6,7 Práctico	MARCISZACK, Marcelo / GUZMÁN, Analía	CASTILLO, Julio	POLLIOTTO, Martín
	2K5	2	Tarde	Lun 3,4,5,6 Teórico Vie 1,2,3,4 Práctico	LIGORRIA, Laura	SERRA, Silvio	POLLIOTTO, Martín
	2K6	2	Tarde	Mar 0,1,2,3 Teórico Mie 3,4,5,6 Práctico	LIGORRIA, Laura	COLACIOPPO Nicolás	POLLIOTTO, Martín
	2K7 (contrac.)	1	Tarde	Mie 0,1,2,3 Teórico Jue 3,4,5,6 Práctica	CORSO, Cynthia	COLACIOPPO Nicolás	SERRANO, Diego
	2K8	2	Noche	Mar 0,1,2,3 Teórico Vie 3,4,5,6 Práctico	GUZMÁN, Analía	LIGORRIA, Karina	ROMANI, Germán
	2K9	2	Noche	Lun 3,4,5,6 Teórico Jue 0,1,2,3 Práctico	MARCISZACK, Marcelo / LIGORRIA, Laura	LIGORRIA, Karina	ROMANI, Germán
	2K10 (contrac.)	1	Noche	Mar 3,4,5,6 Teórico Mie 3,4,5,6 Práctico	LIGORRIA, Laura	CASTILLO, Julio	ROMANI, Germán
	2K11	2	Mañana	Mar 4,5,6,7 Teórico Jue 1,2,3,4 Práctico	GUZMÁN, Analía	FRIAS, Pablo	POLLIOTTO, Martín



Firma:

Aclaración: Marcelo Martín Marciszack