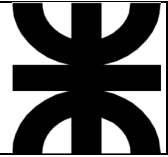


Ingeniería en Sistemas de Información
Paradigmas de Programación
Planificación Ciclo lectivo 2022

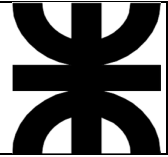
Datos administrativos de la asignatura			
Nivel en la carrera	2°	Duración	cuatrimestral
Bloque curricular:	<input checked="" type="checkbox"/> Programación <input type="checkbox"/> Computación <input type="checkbox"/> Sistemas de Información <input type="checkbox"/> Gestión Ingenieril <input type="checkbox"/> Modelos <input type="checkbox"/> Asignatura Electiva		
Carga horaria presencial semanal:	8 horas	Carga Horaria total:	128 horas
Carga horaria no presencial semanal (si correspondiese)		% horas no presenciales (si correspondiese)	
Propósito			
<p>Que el estudiante sea capaz de comprender, los fundamentos de los paradigmas de programación que dan sustento a los lenguajes de programación actuales, conocer el modelo formal, los mecanismos y las razones filosóficas de constitución, que dieron origen a cada uno de los paradigmas de programación y cómo los mismos se manifiestan dentro de los lenguajes actuales de programación.</p> <p>Ser capaces de aplicar los paradigmas y sus lenguajes asociados en la resolución de problemas de ingeniería en forma concreta.</p>			
Objetivos establecidos en el DC			
<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. 			



Resultados de aprendizaje

Se describen a continuación los Resultados de Aprendizaje (RA) promovidos dentro del desarrollo de cada una de las unidades que conforman la asignatura.

- RA1: 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.
- RA2: 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.
- RA3: 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.
- RA4: 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.
- RA5: 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.

- RA6: Reconocer los elementos constitutivos, mecanismos y formas de implementación dentro de los paradigmas y lenguajes utilizados dentro de un determinado paradigma de programación utilizado, para generar la habilidad de aplicarlos eficientemente en la resolución de diferentes problemas ingenieriles del área de sistemas con los que lleguen a aplicar los estudiantes tanto en el ámbito académico como en el profesional.

Asignaturas correlativas previas

Para cursar debe tener cursada:

- Matemática discreta
- Algoritmos y Estructuras de Datos

Para cursar debe tener aprobada:

- ---

Para rendir debe tener aprobada:

- Matemática discreta
- Algoritmos y Estructuras de Datos

Asignaturas correlativas posteriores

Las asignaturas correlativas posteriores son:

- Gestión de Datos
- Electivas de programación de 3° año

Contenidos Mínimos

(Contenidos Mínimos, según Diseño Curricular- Ordenanza 1150)

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.

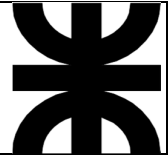
Método y Mensaje.

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.

Programa analítico, Unidades temáticas

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

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.

Carga horaria por unidad:

8 horas totales de clases teóricas más actividades prácticas.

Carga horaria por tipo de formación práctica.

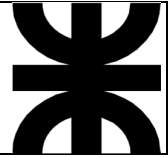
4 horas de actividades prácticas.

Actividades a desarrollar:

Clase Teórica	Clase Práctica
Presentación de la Asignatura, modalidad de cursado y régimen de Regularización. Promoción y Aprobación Directa. Desarrollo completo de la Unidad N° 1. Presentación de los lenguajes de Programación a utilizar.	Presentación de la Parte Práctica. Presentación de las características de los diferentes entornos de desarrollo (IDE). Acceso y ejecución de cada uno de los lenguajes a utilizar dentro del desarrollo de la asignatura.

Las actividades de esta unidad podrán ser dictadas en forma virtual (De ser requerido por el Departamento de Ingeniería en Sistemas de Información). Si son dictadas en forma virtual, será usando los recursos físicos de los docentes y de los estudiantes, con la herramienta Zoom y los entornos de los que dispongan los participantes. (El software utilizado es libre y se proveen los links para su obtención).

Se utilizará como medio de comunicación, la autogestión académica para la publicación de los mensajes generales iniciales para los estudiantes.



Se utilizará el aula virtual para la publicación de la modalidad académica (Programa detallado de la asignatura), el cronograma de clases, el link a las clases virtuales, los apuntes de cada Unidad (repositorio institucional - RIA), las presentaciones de la unidad y todo material relevante para el aprendizaje de los estudiantes. Adicionalmente, se realizará el registro de participación de los estudiantes y de las comunicaciones de los docentes y de los estudiantes a través de los foros, mensajes, entre otros.

Bibliografía Obligatoria:

- [Cátedra Paradigmas de Programación, 2021] Material correspondiente a Unidad 1 elaborado por los docentes de la Cátedra y puesto a disposición de los estudiantes en la UV. Accesible desde la web: <https://ria.utn.edu.ar/xmlui/handle/20.500.12272/5213>.
- [Cátedra de Paradigmas de Programación, 2022] Guía de trabajos prácticos en UV (Moodle de Cada Curso)

Bibliografía optativa y otros materiales a utilizar en la asignatura:

- [Watt, 2004] Watt D. A. (2004). Programming language design concepts. John Wiley & Sons. (Anterior formato: David A. Watt, William Findlay - 2004 - Programming Language Desing Concepts).
- [Sebesta, 2011] Robert Sebesta (2011) - Concepts of Programming Languages
- [Ghezzi, 1996] Ghezzi, Carlo & Mehdi Jazayeri - 1996 - Programming Languages Concepts - 3a Edición

Unidad Nro. 2: Paradigma Imperativo: Secuencial o Estructurado.

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.

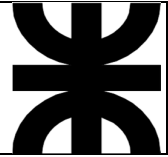
Lenguaje asociado: Python / C++. Entornos Integrados de Desarrollo.

Carga horaria por unidad:

8 horas totales de clases teóricas más actividades prácticas.

Carga horaria por tipo de formación práctica.

4 horas de actividades prácticas.



Actividades a desarrollar:

Clase Teórica	Clase Práctica
Desarrollo completo de la Unidad Temática N° 2.	Repaso de lo visto en la materia Algoritmos y Estructuras de Datos, relacionado al Paradigma Estructurado. Unidad N° 2 Introducción y resolución de problemas con lógica estructurada, utilizando el lenguaje de programación C/C++ o Python.

Las actividades de esta unidad podrán ser dictadas en forma virtual (De ser requerido por el Departamento de Ingeniería en Sistemas de Información). Si son dictadas en forma virtual, será usando los recursos físicos de los docentes y de los estudiantes, con la herramienta Zoom y los entornos de los que dispongan los participantes. (El software utilizado es libre y se proveen los links para su obtención).

Se utilizará como medio de comunicación, la autogestión académica para la publicación de los mensajes generales iniciales para los estudiantes.

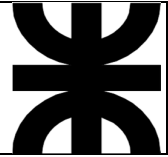
Se utilizará el aula virtual para la publicación de la modalidad académica (Programa detallado de la asignatura), el cronograma de clases, el link a las clases virtuales, los apuntes de cada Unidad (repositorio institucional - RIA), las presentaciones de la unidad y todo material relevante para el aprendizaje de los estudiantes. Adicionalmente, se realizará el registro de participación de los estudiantes y de las comunicaciones de los docentes y de los estudiantes a través de los foros, mensajes, entre otros.

Bibliografía Obligatoria:

- [Cátedra Paradigmas de Programación, 2021] Material correspondiente a Unidad 2 elaborado por los docentes de la Cátedra y puesto a disposición de los estudiantes en la UV. Accesible desde la web: <http://hdl.handle.net/20.500.12272/5214>.
- [Cátedra de Paradigmas de Programación, 2022] Guía de trabajos prácticos en UV (Moodle de Cada Curso)

Bibliografía optativa y otros materiales a utilizar en la asignatura:

- [Deitel, 2013] Harvey M. Deitel y Paul J. Deitel C/C++ Cómo Programar. Editorial: Pearson 2013. Idioma: Español. ISBN: 9786074424621
- [Bishop, 1993] Judy M. Bishop Java – Fundamentos de Programación Publicado por ADDISON WESLEY, 1999 ISBN 10: 8478290222 ISBN 13: 9788478290222
- [Frittelli, 2016] Frittelli, V., et al. (2016). Publicaciones Cátedra de AED. Fichas de Estudio para AED (disponible en Aula Virtual AED). Córdoba: UTN – Facultad Regional Córdoba.



- [Python Software Foundation, 2015]. Python Documentation. Retrieved February 24, 2016, from <https://docs.python.org/3/>.

Unidad N° 3: Paradigma de Programación con Orientación a Objetos.

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.

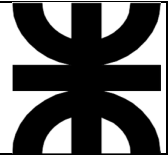
Carga horaria por unidad:

48 horas totales de clases teóricas más las actividades prácticas.

Carga horaria por tipo de formación práctica.

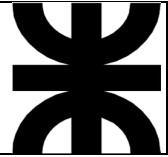
24 horas de actividades prácticas.

Actividades a desarrollar:



Clase Teórica	Clase Práctica
<p>Introducción al paradigma. Conceptos generales. Características y aplicaciones. Introducción al Lenguaje Smalltalk. Características, aspectos y sintaxis. Objetos y Tipos de Mensajes.</p>	<p>Introducción al Lenguaje Smalltalk: Características, aspectos y sintaxis - Presentación entorno Pharo, ventanas, browser, inspecciones. Biblioteca de Smalltalk: clases Boolean, Integer, etc. Evaluación de expresiones. Inspección de clases Smalltalk y ejemplos de clases definidas por el usuario (sin implementar el proceso de creación de clases con los estudiantes). Creación de objetos y paso de mensajes usando las clases de Smalltalk y definidas por el usuario</p>
<p>Sintaxis y definición de una clase simple, con creación e invocación de objetos. Bloques y abstracciones de control</p>	<p>Creación de clase definidas por el usuario. Bloques: ejercicios con bloques.</p>
<p>Relaciones entre clases: Composición. Presentación de diferentes modelos. Presentación de colecciones: mensajes de creación, de inserción y recorrido simple (mensaje do:). (sugerencia OrdererCollection)</p>	<p>Ejercitación con composición y colecciones. Otros mensajes de recorridos en colecciones (detect y select).</p>
<p>Relaciones entre clases: Herencia. Introducción a Polimorfismo. Presentación de diferentes modelos.</p>	<p>Ejercitación con diferentes casos de aplicación usando herencia y polimorfismo y colecciones.</p>
<p>Continuación de Polimorfismo. Presentación de la jerarquía de clases y de los métodos polimórficos de colecciones de Smalltalk.</p>	<p>Ejercitación con diferentes casos de aplicación usando herencia y polimorfismo y colecciones.</p>
<p>Ejercitación con diferentes casos de aplicación usando herencia y polimorfismo y colecciones.</p>	<p>Ejercitación con diferentes casos de aplicación usando herencia y polimorfismo y colecciones.</p>

Las actividades de esta unidad podrán se dictadas en forma virtual (De ser requerido por el Departamento de Ingeniería en Sistemas de Información). Si son dictadas en forma virtual, será usando los recursos físicos de los docentes y de los estudiantes, con la herramienta Zoom y los



entornos de los que dispongan los participantes. (El software utilizado es libre y se proveen los links para su obtención).

Se utilizará como medio de comunicación, la autogestión académica para la publicación de los mensajes generales iniciales para los estudiantes.

Se utilizará el aula virtual para la publicación de la modalidad académica (Programa detallado de la asignatura), el cronograma de clases, el link a las clases virtuales, los apuntes de cada Unidad (repositorio institucional - RIA), las presentaciones de la unidad y todo material relevante para el aprendizaje de los estudiantes. Adicionalmente, se realizará el registro de participación de los estudiantes y de las comunicaciones de los docentes y de los estudiantes a través de los foros, mensajes, entre otros.

Bibliografía Obligatoria:

- [Cátedra Paradigmas de Programación, 2021] Material correspondiente a Unidad 3 elaborado por los docentes de la Cátedra y puesto a disposición de los estudiantes en la UV. Accesible desde la web: <http://hdl.handle.net/20.500.12272/5215>.
- [Cátedra de Paradigmas de Programación, 2022] Guía de trabajos prácticos en UV (Moodle de Cada Curso)

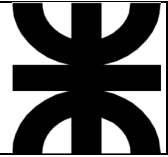
Bibliografía optativa y otros materiales a utilizar en la asignatura:

- [Black 2015] Andrew P. Black Stéphane Ducasse Oscar Nierstrasz Damien Pollet. Pharo por Ejemplo <http://pharobyexample.org/es/PBE1-sp.pdf>
- [Lalonde 1991] Lalonde y J. Pugh; Inside Smalltalk. Vol. I, II: Prentice Hall International.
- [Goldberg 1983] Adele Goldberg and David Robson, Smalltalk 80- The Language its implementation. ISBN 10: 0201113716 ISBN 13: 9780201113716
- [Wirfs-Brock 1986] Wirfs-Brock, R. & Et all, Designing Object-Oriented Software, Prentice Hall. Accesible desde <http://stephane.ducasse.free.fr/FreeBooks/BlueBook/Bluebook.pdf>
- [Prieto 2002] 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 Vázquez.
- [Lombardi 2016] Apunte: Introducción a la orientación a objetos de la Asignatura Paradigmas de Programación UTN – FRBA Autores: Carlos Lombardi – Nicolás Passerini.

Unidad N° 4: Paradigma de Programación Funcional.

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

Carga horaria por unidad:

24 horas totales de clases teóricas más actividades prácticas.

Carga horaria por tipo de formación práctica.

12 horas de actividades prácticas.

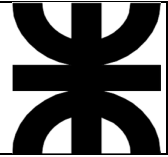
Actividades a desarrollar:

Clase Teórica	Clase Práctica
Introducción, características y conceptos generales. (Funciones, Cálculo Lambda). Introducción a Haskell. Sintaxis. Tipos de datos (tuplas), funciones y expresiones. Expresiones If-Then-Else, Case, guardas.	Introducción a Haskell Evaluación de expresiones simples, composición de funciones. Ejercitación con expresiones If-Then-Else, Case, guardas, definiciones locales y tuplas.
Definiciones locales. Expresiones recursivas. Listas en Haskell, funciones predefinidas. Recorridos recursivos. Listas por comprensión.	Ejercitación con Listas y recursividad. Listas por comprensión.
Continuación de listas. Tipos definidos por el usuario y tipos polimórficos.	Ejercitación con casos de estudio que apliquen todos los contenidos vistos.

Las actividades de esta unidad podrán ser dictadas en forma virtual (De ser requerido por el Departamento de Ingeniería en Sistemas de Información). Si son dictadas en forma virtual, será usando los recursos físicos de los docentes y de los estudiantes, con la herramienta Zoom y los entornos de los que dispongan los participantes. (El software utilizado es libre y se proveen los links para su obtención).

Se utilizará como medio de comunicación, la autogestión académica para la publicación de los mensajes generales iniciales para los estudiantes.

Se utilizará el aula virtual para la publicación de la modalidad académica (Programa detallado de la asignatura), el cronograma de clases, el link a las clases virtuales, los apuntes de cada Unidad



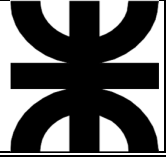
(repositorio institucional - RIA), las presentaciones de la unidad y todo material relevante para el aprendizaje de los estudiantes. Adicionalmente, se realizará el registro de participación de los estudiantes y de las comunicaciones de los docentes y de los estudiantes a través de los foros, mensajes, entre otros.

Bibliografía Obligatoria:

- [Cátedra Paradigmas de Programación, 2021] Material correspondiente a Unidad 3 elaborado por los docentes de la Cátedra y puesto a disposición de los estudiantes en la UV. Accesible desde la web: <http://hdl.handle.net/20.500.12272/5217>.
- [Cátedra de Paradigmas de Programación, 2022] Guía de trabajos prácticos en UV (Moodle de Cada Curso)

Bibliografía optativa y otros materiales a utilizar en la asignatura:

- [Bird 1997] Richard Bird. "Introduction to Functional Programming using Haskell". Prentice Hall International, 2nd Ed. New York, 1997
- [Cardelli 1998] Luca Cardelli. "Basic Polymorphic Typechecking". AT&T Bell Laboratories, Murray Hill, NJ 07974. 1998.
- [Hudak 1999] Paul Hudak. "A Gentle Introduction to Haskell 98". Yale University. Department of Computer Science. John Peterson. 1999.
- [Hudak 2000] Paul Hudak. "The Haskell School of Expression: Learning Functional Programming through Multimedia". Yale University. Cambridge University Press, New York, 2000.
- [Thompson 1999] Simon Thompson. "Haskell: The Craft of Functional Programming". 3rd Edition. ISBN: 9780201882957. 1999.
- [Labra 1998] Jose E. Labra G. "Introducción al lenguaje Haskell". Universidad de Oviedo, Departamento de Informática, Octubre 1998.
- [Normark 2003] Kurt Normark. "Funcional Programming in Scheme". Department of Computer Science, Aalborg University, Denmark. 2003.
- [Kelsey 1998] 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 1984] 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.
- [Varela 2011] Carlos Varela. "Programming Languages". Rensselaer Polytechnic Institute. USA. 2011.
- [Spigariol 2007] Lucas Spigariol. "Paradigma Funcional". Paradigmas de Programación - FRBA - UTN. 2007.
- [Jiménez 2010] Alonso Jiménez, José. "Programación Declarativa: Definición de listas por comprensión". Departamento de Ciencias de la Computación e I. A. Universidad de Sevilla. 2010.



Unidad N° 5: Paradigma de Programación Lógico.

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.

Carga horaria por unidad:

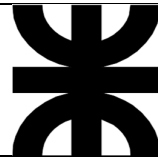
24 horas totales de clases teóricas más actividades prácticas.

Carga horaria por tipo de formación práctica.

12 horas de actividades prácticas.

Actividades a desarrollar:

Clase Teórica	Clase Práctica
Introducción – Lógica Predicados de primer orden. Reglas, hechos, cláusulas de Horn, predicados. Sintaxis y estructura de Prolog.	Entorno y Sintaxis de Prolog. Orden de evaluación. Cláusulas simples.
Motor de Inferencia. Unificación, Backtracking, corte y fallo. Objetos Compuestos.	Ejercitación de unificación, relaciones entre tablas. Objetos Compuestos: Ejercitación de objetos compuestos.
Recursividad y listas. Caso de estudio.	Ejercitación de recursividad y listas. Caso de estudio.



Las actividades de esta unidad podrán ser dictadas en forma virtual (De ser requerido por el Departamento de Ingeniería en Sistemas de Información). Si son dictadas en forma virtual, será usando los recursos físicos de los docentes y de los estudiantes, con la herramienta Zoom y los entornos de los que dispongan los participantes. (El software utilizado es libre y se proveen los links para su obtención).

Se utilizará como medio de comunicación, la autogestión académica para la publicación de los mensajes generales iniciales para los estudiantes.

Se utilizará el aula virtual para la publicación de la modalidad académica (Programa detallado de la asignatura), el cronograma de clases, el link a las clases virtuales, los apuntes de cada Unidad (repositorio institucional - RIA), las presentaciones de la unidad y todo material relevante para el aprendizaje de los estudiantes. Adicionalmente, se realizará el registro de participación de los estudiantes y de las comunicaciones de los docentes y de los estudiantes a través de los foros, mensajes, entre otros.

Bibliografía Obligatoria:

- [Cátedra Paradigmas de Programación, 2021] Material correspondiente a Unidad 3 elaborado por los docentes de la Cátedra y puesto a disposición de los estudiantes en la UV. Accesible desde la web: <http://hdl.handle.net/20.500.12272/5218>.
- [Cátedra de Paradigmas de Programación, 2022] Guía de trabajos prácticos en UV (Moodle de Cada Curso)

Bibliografía optativa y otros materiales a utilizar en la asignatura:

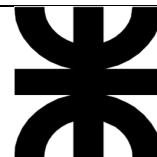
- [Tasistro 1988] Tasistro A., J. Vidart. "Programación Lógica y funcional". EBAI (1988).
- [Pereira 1987] Fernando C. N. Pereira and Stuart M. Shieber. Prolog and Natural Language Analysis. Cambridge University Press, ISBN: 0-9719997-0-4, 1987
- [Merritt 1989] D. Merritt. Building Expert Systems in Prolog. Springer-Verlag, 1989
- [Bratko 1990] I. Bratko. Prolog, Programming for Artificial Intelligence. Addison Wesley, ISBN: 0-201-41606-9, 1990
- [Nilsson 1990] U. Nilsson, J. Maluszynski. Logic, Programming and Prolog. John Wiley & Sons Ltd, 1990
- [Sterling 1994] L. Sterling, E. Shapiro. The Art of Prolog. The MIT Press, ISBN: 0-262- 19338-8, 1994.

Unidad Nro. 6: Elementos constitutivos de Lenguajes y Paradigmas

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.

Carga horaria por unidad:

8 horas totales de clases teóricas más actividades prácticas.

Carga horaria por tipo de formación práctica.

4 horas de actividades prácticas.

Actividades a desarrollar:

Clase Teórica	Clase Práctica
Desarrollo completo de la unidad N° 6	Aplicación sobre casos de estudios integradores.

Las actividades de esta unidad podrán ser dictadas en forma virtual (De ser requerido por el Departamento de Ingeniería en Sistemas de Información). Si son dictadas en forma virtual, será usando los recursos físicos de los docentes y de los estudiantes, con la herramienta Zoom y los entornos de los que dispongan los participantes. (El software utilizado es libre y se proveen los links para su obtención).

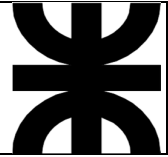
Se utilizará como medio de comunicación, la autogestión académica para la publicación de los mensajes generales iniciales para los estudiantes.

Se utilizará el aula virtual para la publicación de la modalidad académica (Programa detallado de la asignatura), el cronograma de clases, el link a las clases virtuales, los apuntes de cada Unidad (repositorio institucional - RIA), las presentaciones de la unidad y todo material relevante para el aprendizaje de los estudiantes. Adicionalmente, se realizará el registro de participación de los estudiantes y de las comunicaciones de los docentes y de los estudiantes a través de los foros, mensajes, entre otros.

Bibliografía Obligatoria:

- [Cátedra Paradigmas de Programación, 2021] Material correspondiente a Unidad 3 elaborado por los docentes de la Cátedra y puesto a disposición de los estudiantes en la UV. Accesible desde la web: <http://hdl.handle.net/20.500.12272/5219>.
- [Cátedra de Paradigmas de Programación, 2022] Guía de trabajos prácticos en UV (Moodle de Cada Curso)

Bibliografía optativa y otros materiales a utilizar en la asignatura:



- [Sethy 1992] Lenguajes de Programación - Conceptos y Constructores Ravi Sethy Ed. Addison Wesley
- [Universidad de Galileo 2020] Compiladores. Análisis semántico y chequeo de tipos. Universidad Galileo, accesible desde la web; <https://slidetodoc.com/compiladores-analisis-semntico-y-chequeo-de-tipos-resumen/>
- [Campos 2011] Oscar Campos. Diferencias entre Paradigmas de Programación. Accesible desde web <https://www.genbeta.com/desarrollo/diferencias-entre-paradigmas-de-programacion>

Metodología de enseñanza

Los contenidos teóricos 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.

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.

El estudiante realizará aprendizajes significativos a través de la comprensión de los contenidos teóricos y la aplicación de los mismos en la resolución de problemas, asociados a cada paradigma.

En las clases prácticas y en las clases teóricas, en la medida de lo posible, se realizarán actividades prácticas de laboratorios con cada uno de los paradigmas de programación cuya parte práctica sea relevante para el aprendizaje significativo del estudiante.

Los estudiantes, tendrán que realizar y presentar para la regularización de la asignatura, trabajos integradores sobre problemas a resolver en cada Paradigma. Estos prevén para su realización de actividades creativas, autónomas y a su vez colaborativas entre los estudiantes, con interacción con los auxiliares docentes, para la realización de los mismos.

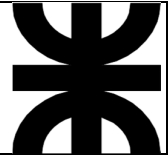
Metodología de evaluación

Evaluaciones

Todos los estudiantes son evaluados en tres grupos de instancias de evaluación:

- Evaluaciones de Parciales Teóricos: Son realizadas al finalizar cada unidad temática a través cuestionarios de múltiple opción en donde cada pregunta tiene asignada una cantidad de puntaje igualitario, se utiliza el aula virtual (plataforma Moodle) para realizarla.

La modalidad de evaluación de contenidos teóricos será realizada a través de cuestionarios con opciones de respuestas de opciones múltiples, realizado íntegramente en máquina, el cual se le asignará un tiempo de 30 minutos para 25 preguntas generadas aleatoriamente, al finalizar se entrega un reporte con la nota obtenida por el estudiante. Se planifica la ejecución para efectuarse durante las clases teóricas.



- Evaluaciones de Parciales Prácticos: Se realizan para las unidades que tienen un alto contenido práctico y del cual depende el aprendizaje significativo del estudiante. En las mismas se plantea una situación problemática a resolver, su modelo de representación gráfico y /o Reglas o hechos y/o bases de conocimientos, según el paradigma que se evalúe. Se le provee al estudiante de un conjunto de datos y los resultados que debe arrojar cada uno de los puntos solicitados. También se le provee una rúbrica de cada instancia de evaluación para que pueda ponderar el impacto que tendrá en la sumatoria de puntaje el realizar en forma correcta el ítem solicitado.

Se realizan simultáneamente en dos exámenes parciales en fechas unificadas para todos los cursos, disponiéndose en ambos casos de fechas alternativas para estudiantes con problemas laborales o motivos religiosos.

Las fechas y alcance de todos los parciales son definidos y anunciados al comenzar el cuatrimestre y publicados en el aula virtual.

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 estudiantes son examinados con los mismos temas y evaluados con criterios de corrección comunes a todos los cursos.

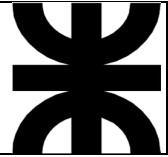
La modalidad de evaluación de contenidos prácticos se realizará en máquina donde dependiendo de los aspectos a evaluar y el paradigma, se solicitará la codificación de la solución propuesta para un determinado problema, o la modificación o ampliación de determinadas partes de una solución previamente planteada. Esta corrección se realizará de acuerdo a criterios de corrección previamente informados.

En caso de que el instrumento de evaluación del práctico, ya sea parcial o final incluya dos paradigmas diferentes, el estudiante para aprobar el mismo, deberá aprobar como mínimo un 50 % de cada paradigma evaluado.

- Evaluación de Trabajos Prácticos Integradores: Similar al punto anterior, la idea central es que el estudiante ejercite y tenga una devolución concreta previa a la realización de la instancia de la evaluación parcial práctica, con los mismos criterios de exigencias que se aplicarán en su corrección. Se diferencia a la anterior en el grado de complejidad y extensión, ya que es un trabajo que se realiza en forma grupal y fuera de la clase, con márgenes de tiempo acordes al grado de la evaluación.

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.

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.



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.

NOTA ACLARATORIA: El vencimiento de cada uno de los prácticos Integradores, junto con su opción de recuperación se consideran como plazos perentorios. Vencido el plazo de recuperación de cada uno de los prácticos integradores y no habiendo obtenido su aprobación, el estudiante, no podrá acceder a las opciones de Regularización de la asignatura,

Criterios Generales:

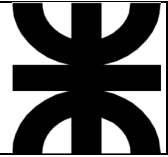
- Los trabajos prácticos integradores deberán ser desarrollados en grupos de hasta 4 (cuatro) estudiantes.
- El medio de entrega de los trabajos prácticos integradores será a través del aula virtual, en el link habilitado específicamente para tal fin.
- Es obligación de cada integrante del grupo subir 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.

Evaluación de cada Resultado de Aprendizaje (evaluación por unidad)

RA1: La evaluación sumativa, referidos a los contenidos de la unidad N° 1, se realiza en el primer cuestionario teórico, con preguntas de múltiple opción sobre los temas trabajados de esta unidad. Se tendrá una instancia de recuperación teórica unificada, con un cuestionario que evaluará RA1, RA2 y RA3 y que en caso de que se repruebe, el estudiante quedará libre.



RA2: La evaluación sumativa, referidos a los contenidos de la unidad N° 2, se realiza en el primer cuestionario teórico, con preguntas de múltiple opción sobre los temas trabajados de esta unidad. Se tendrá una instancia de recuperación teórica unificada, con un cuestionario que evaluara RA1, RA2 y RA3 y que en caso de que se repruebe, el estudiante quedará libre.

RA3: La evaluación sumativa, referidos a los contenidos de la unidad N° 3, se realiza en dos cuestionarios el segundo y tercero, con preguntas de múltiple opción sobre los temas trabajados de esta unidad. Se tendrá una instancia de recuperación teórica unificada, con un cuestionario que evaluara RA1, RA2 y RA3 y que en caso de que se repruebe, el estudiante quedará libre.

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. Se tendrá una instancia de recuperación práctica y que en caso de que se repruebe, el estudiante quedará libre.

El estudiante en forma grupal, deberá resolver y presentar en tiempo y forma el Práctico Integrador N° 1 correspondiente. Se tendrá una instancia de recuperación que en caso de que se repruebe quedará libre.

RA4: La evaluación sumativa, referidos a los contenidos de la unidad N° 4, se realiza en el cuarto cuestionario teórico, con preguntas de múltiple opción sobre los temas trabajados de esta unidad. Se tendrá una instancia de recuperación teórica unificada, con un cuestionario que evaluara RA4, RA5 y RA6 y que en caso de que se repruebe, el estudiante quedará libre.

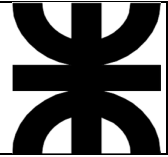
Los contenidos prácticos se evaluarán en máquina donde se solicitará la codificación de la solución propuesta para un determinado problema, o la modificación o ampliación de determinadas partes de una solución previamente planteada. Se tendrá una instancia de recuperación práctica unificada con los contenidos de RA4 y RA5, en caso de que se repruebe, el estudiante quedará libre.

El estudiante en forma grupal, deberá resolver y presentar en tiempo y forma el Práctico Integrador N° 2 correspondiente. Se tendrá una instancia de recuperación que en caso de que se repruebe quedará libre.

RA5: La evaluación sumativa, referidos a los contenidos de la unidad N° 5, se realiza en el quinto cuestionario teórico, con preguntas de múltiple opción sobre los temas trabajados de esta unidad. Se tendrá una instancia de recuperación teórica unificada, con un cuestionario que evaluara RA4, RA5 y RA6 y que en caso de que se repruebe, el estudiante quedará libre.

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. Se tendrá una instancia de recuperación práctica unificada con los contenidos de RA4 y RA5, en caso de que se repruebe, el estudiante quedará libre.

El estudiante en forma grupal, deberá resolver y presentar en tiempo y forma el Práctico Integrador N° 3 correspondiente. Se tendrá una instancia de recuperación que en caso de que se repruebe quedará libre.



RA6: La evaluación sumativa, referidos a los contenidos de la unidad N° 6, se realiza en el primer cuestionario teórico, con preguntas de múltiple opción sobre los temas trabajados de esta unidad. Se tendrá una instancia de recuperación teórica unificada, con un cuestionario que evaluara RA4, RA5 y RA6 y que en caso de que se repruebe, el estudiante quedará libre.

Rúbricas

Para cada instancia de evaluación se le provee al estudiante de la rúbrica correspondiente para que pueda ponderar el impacto que tendrá en la sumatoria de puntaje, al realizar en forma correcta el ítem solicitado.

Condiciones de aprobación

Para la aprobación de las instancias de evaluación se utiliza la siguiente escala de notas de regularidad(*)

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

(*) Escala acordada en reunión de Docentes Coordinadores

Condiciones especiales:

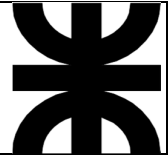
Estudiante Abandono: Se considera en esta situación a todo estudiante que no asistió nunca a clases o abandonó el cursado; es decir no realizó todas las instancias de evaluación.

Estudiante Libre: Quienes no cumplieron las condiciones establecidas para el estudiante Regular, habiéndose presentado en todas las instancias de evaluaciones previstas.

Condición Regular

Es requisito para regularizar el curso, la regularidad de asistencia y/o tener al menos algún registro de participación del cursado y cinco (5) notas aprobadas:

1. Nota promedio de los 3 (tres) primeros cuestionarios teóricos;



2. Nota promedio de los 3 (tres) últimos cuestionarios teóricos;
3. Nota del primer parcial práctico;
4. Nota del segundo parcial práctico;
5. Nota promedio de los tres trabajos prácticos integradores (Uno por cada uno de los Paradigmas Objeto, Lógico y Funcional).

Para alcanzar la condición de Estudiante Regular, las cinco notas deben estar aprobadas: En el caso de evaluaciones parciales, tanto teóricas como prácticas deben superar una calificación mínima de 4 (cuatro) puntos, que corresponde a una evaluación del 55%.

NOTA: En referencia a los Prácticos Integradores, los mismos deben ser cumplimentados y aprobados, en su fecha original de aprobación o de su recuperación. Vencidos estos plazos el estudiante, no podrá acceder a la condición de estudiante regular.

ACLARACION: El estudiante en condición de regular puede rendir en el plazo de un ciclo lectivo sin control de correlativas aprobadas.

Instancia de recuperación

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.

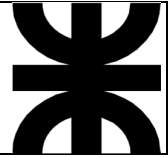
El estudiante dispondrá de una instancia de recuperación para cada uno de los tres trabajos prácticos integradores, que consistirá en la reentrega del trabajo práctico previamente evaluado y con las correcciones efectuadas, al cual se le adicionará alguna funcionalidad. La fecha máxima para dar cumplimiento a la entrega definitiva de los prácticos integradores estará definida en el cronograma de actividades.

NOTA: Para cada una de las 5 (cinco) instancias de evaluación, se tomará la mayor nota obtenida, ya sea en la instancia normal de evaluación o en la de recuperación.

Condición Promoción Práctica

En caso de que el estudiante obtenga entre las evaluaciones parciales prácticas, y la nota promedio correspondiente a los trabajos 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.

Acceso a través de la instancia de Recuperación: Esta condición puede ser alcanzada por haber aprobado en las instancias de evaluaciones normales o en las de recuperación (con los mismos



requisitos de cantidad de exámenes habilitados de recuperación para obtener la regularización de la asignatura).

Pérdida de promoción de Prácticos: 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.

El estudiante pierde la condición de promoción práctica, cuando es aplazado en la primera instancia de examen final que se presente de la asignatura, continuando en la condición de alumno regular.

Condición de Aprobación Directa

Si el estudiante 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 (siete) obtiene la aprobación directa de la asignatura.

ACLARACIONES PARA EL ESTUDIANTE:

- La calificación será el promedio de las 5 (cinco) notas, registrada como Nota Final en Autogestión.
- Puede registrar su nota en un examen final dentro del plazo de un ciclo lectivo, sin control de correlativas aprobadas, y después de ello se le exigirán correlativas aprobadas.

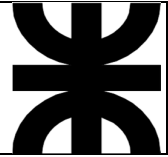
Modalidad de examen final

Para la aprobación del examen final se utiliza la siguiente escala de Notas para Examen Final (*)

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

(*)Escala acordada en reunión de Docentes Coordinadores

El examen final se tomará en un único horario y será unificado para toda la cátedra.



Horario inicio Examen Práctico y Teórico: 15:00 Horas

Examen Final Práctico

El examen final para estudiantes 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.

Examen Final Teórico

Los estudiantes que aprobaron el examen práctico previo, y los estudiantes con **Promoción Práctica** rendirán un coloquio oral que consiste en 3 (tres) preguntas que deberá responder en forma correcta y surgirá de manera aleatoria según las unidades temáticas de la materia, cada uno es de carácter eliminatorio. Es decir para aprobar el examen final, el estudiante, deberá responder en forma correcta las tres preguntas que le hubieran tocado.

Calificación Final

Para calificar los exámenes se utilizará el siguiente criterio:

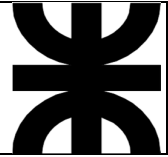
- Aprobación directa: Promedio de las cinco notas del cursado.
- Promoción Práctica: Nota obtenida en el examen final teórico.
- Regular: Promedio entre la nota obtenida en el examen final teórico y el examen final práctico.

Cronograma de clases/trabajos prácticos/exámenes (tentativo)

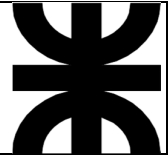
Primer Cuatrimestre

Cronograma de clases

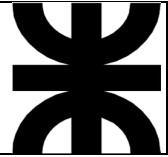
Semana- Fecha	Actividades Teóricas Docente Adjunto	Actividades Prácticas Docente JTP y Auxiliar
1 14/03/22	[Actividad Virtual] Introducción a la Asignatura. Unidad N° 1 y 2.	[Actividad Virtual] Características de los diferentes IDE, Acceso y ejecución. Presentación Parte Práctica. Repaso Paradigma Estructurado: conceptos teóricos generales. Unidad N° 2 Introducción y resolución de problemas con lógica estructurada, utilizando el lenguaje de programación C/C++ o Python.
	[Actividad Virtual]	[Actividad Virtual]



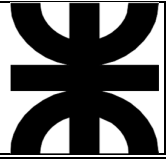
2 21/03/22	Unidad N° 3: POO Introducción al paradigma. Conceptos generales. Características y aplicaciones. Introducción al Lenguaje Smalltalk. Características, aspectos y sintaxis. Objetos y Tipos de Mensajes.	Unidad N° 3: POO Introducción al Lenguaje Smalltalk: Características, aspectos y sintaxis - Presentación entorno Pharo, ventanas, browser, inspecciones. Biblioteca de Smalltalk: clases Boolean, Integer, etc. Evaluación de expresiones. Inspección de clases Smalltalk y ejemplos de clases definidas por el usuario (sin implementar el proceso de creación de clases con los estudiantes). Creación de objetos y paso de mensajes usando las clases de Smalltalk y definidas por el usuario.
3 28/03/22	[Actividad Virtual] Unidad N°3: Sintaxis y definición de una clase simple, con creación e invocación de objetos. Bloques y abstracciones de control	[Actividad Virtual] Unidad N°3: Creación de clase definidas por el usuario. Bloques: ejercicios con bloques.
4 04/04/22	[Actividad Virtual] Unidad N° 3: Relaciones entre clases: Composición. Presentación de diferentes modelos. Presentación de colecciones: mensajes de creación, de inserción y recorrido simple (mensaje do:). (sugerencia OrdererCollection)	[Actividad Virtual] Unidad N°3: Ejercitación con composición y colecciones. Otros mensajes de recorridos en colecciones (detect y select).
5 11/04/22	[Actividad Virtual] Unidad N° 3: Relaciones entre clases: Herencia. Introducción a Polimorfismo. Presentación de diferentes modelos.	[Actividad Virtual] Unidad N° 3: Ejercitación con Herencia y polimorfismo y colecciones.
6 18/04/22	[Actividad Virtual] Unidad N° 3: Continuación de Polimorfismo. Presentación de la jerarquía de clases y de los métodos polimórficos de colecciones de Smalltalk.	[Actividad Virtual] Unidad N° 3: Ejercitación con Herencia, Polimorfismo y Colecciones.
7 25/04/22	[Actividad Virtual] Unidad N° 3: Casos de estudios. Repaso para el parcial práctico	[Actividad Virtual] Unidad N° 3: Ejercitación con Herencia, Polimorfismo y Colecciones.
8 02/05/22	[Actividad Presencial] Unidad N° 4: Paradigma funcional	[Actividad Presencial] Unidad N° 4: Introducción a Haskell



	Introducción, características y conceptos generales. (Funciones, Cálculo Lambda). Introducción a Haskell. Sintaxis. Tipos de datos (tuplas), funciones y expresiones. Expresiones If-Then-Else, Case, guardas.	Evaluación de expresiones simples, composición de funciones. Ejercitación con expresiones If-Then-Else, Case, guardas, definiciones locales y tuplas.
9 09/05/22	[Actividad Presencial] Unidad N° 4: Definiciones locales. Expresiones recursivas. Listas en Haskell, funciones predefinidas. Recorridos recursivos. Listas por comprensión.	[Actividad Presencial] Unidad N° 4: Ejercitación con Listas y recursividad. Listas por comprensión.
10 16/05/22	[Actividad Presencial] Unidad N° 4: Continuación de listas. Tipos definidos por el usuario y tipos polimórficos.	[Actividad Presencial] Unidad N° 4: Ejercitación con Listas y Tuplas: ejercicios simples.
11 23/05/22	[Actividad Presencial] Unidad N° 5: Paradigma Lógico Introducción – Lógica Predicados de primer orden. Reglas, hechos, cláusulas de Horn, predicados. Sintaxis y estructura de Prolog.	[Actividad Presencial] Unidad N° 5: Entorno y Sintaxis de Prolog. Orden de evaluación. Cláusulas simples.
12 30/05/22	[Actividad Presencial] Unidad N° 5: Motor de Inferencia. Unificación, Backtracking, corte y fallo. Objetos Compuestos.	[Actividad Presencial] Unidad N° 5: Ejercitación de unificación, relaciones entre tablas. Objetos Compuestos: Ejercitación de objetos compuestos.
13 06/06/22	[Actividad Presencial] Unidad N° 5: Recursividad y listas.	[Actividad Presencial] Unidad N° 5: Ejercitación de recursividad y listas. Caso de estudio:
14 13/06/22	[Actividad Presencial] Unidad N° 6: Elementos constitutivos de los lenguajes y los paradigmas.	[Actividad Presencial] Unidad N° 4, 5 y 6: Ejercitación de recursividad y listas. Repaso para el parcial.
15 20/06/22	[Actividad Presencial] Casos de estudio de repaso para recuperatorios y finales.	[Actividad Presencial] Consulta y/o devolución de TPS.
16 27/06/22	[Actividad Presencial] Casos de estudio de repaso para recuperatorios y finales.	[Actividad Presencial] Repaso para recuperatorio del parcial práctico.



Cronograma de instancias de evaluaciones parciales		
Semana- Fecha	Cuestionarios Teóricos	Trabajos Prácticos Integradores
3 28/03/22	Evaluación del Cuestionario 1. Unidades 1 y 2.	Presentación Práctico N° 1.
5 11/04/22	Evaluación del cuestionario 2: Unidad 3- Parte de Introducción al Paradigma Orientado a Objeto hasta relaciones de Clases (Composición).	
6 18/04/22		Entrega Práctico N° 1
7 25/04/22	Evaluación del cuestionario 3: Unidad 3- Colecciones, Herencia y Polimorfismo.	Devolución Práctico N° 1
Sábado 30/04/22	Primer parcial práctico	
8 02/05/22		Presentación de Práctico N° 2
9 09/05/22		Entrega recuperatorio Práctico N°1
10 16/05/22		Devolución recuperatorio Práctico N°1
11 23/05/22	Evaluación del cuestionario 4: Unidad 4- Paradigma Funcional (completo).	Entrega del Práctico N° 2. Presentación de Práctico N° 3.
12 30/05/22		Devolución de Práctico N° 2.
14 13/06/22	Evaluación del cuestionario 5: Unidad 5- Paradigma Lógico (completo).	Entrega del Práctico N° 3. Entrega recuperatorio Práctico N°2
Sábado 18/06/22	Segundo Parcial práctico	
15	Evaluación del cuestionario 6:	Devolución Práctico N° 3



20/06/22	Unidad 6- Elementos constitutivos de los lenguajes y los paradigmas (completo).	
16 27/06/22	Instancias de Recuperación –Teórica (1 etapa y 2 etapa)	Entrega recuperatorio Práctico Nº3
Sábado 02/07/21	Instancias de Recuperación – Prácticas (1er y 2do Parcial)	
Atención y orientación de estudiantes		
<p>En la UV de cada curso están publicados los mails de contacto de cada uno de los docentes y planta docente de cada curso.</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 estudiantes.</p>		

Marcelo M. Marciszack
 Director de Cátedra
 Paradigmas de Programación