

Sede	Sede Andina
Localidad	Bariloche
Escuela	Escuela de Producción, Tecnología y Medio Ambiente
Carrera	Ingeniería en Computación

Programa analítico de	Programación III	Código Guaraní
	Plan de estudio: ICOMP 2021-V7	B6015
	Ubicación en el plan de estudios	3 año - 1º CUATRIMESTRE

Correlativas vigentes	Para cursar	Cursada aprobada	Sin correlativas vigentes
		Materia aprobada	Matemática I (B6000)
	Para rendir	Materia aprobada	Matemática I (B6000)
Ciclo lectivo: 2023	Régimen de cursada	CUATRIMESTRAL	
	Período de dictado	2º CUATRIMESTRE	
Carga horaria semanal: 6		Carga horaria total: 96	
Horas teóricas totales: 48		Horas prácticas totales: 48	
Carga horaria virtual: 0		Horas de estudio extra clase recomendadas: 8	

Profesor/a	PAOLA VERONICA BRITOS
Equipo de docencia	PAOLA VERONICA BRITOS-PABLO ENRIQUE ARGANARAS

Fundamentación
El conocimiento profundo de los conceptos intrínsecos de los lenguajes de programación permite adquirir herramientas para construir criterios de evaluación de los lenguajes. Esto hace posible consolidar los conocimientos sobre los mismos e incorporar elementos para abordar los nuevos y así usar los mismos adecuadamente para la resolución de problemas. Esta materia posibilita elegir la mejor herramienta para el trabajo aprovechando los puntos fuertes de las distintas plataformas.

Propósitos de la asignatura

Dar a conocer los diferentes paradigmas, su influencia en la forma de ver o modelar un determinado problema, analizando las características de los diferentes paradigmas para la solución en términos de construir e implementar una solución (construcción de un programa) Sentar las bases, en referencia a la solución de problemas a través de la implementación de diferentes lenguajes de programación dentro de diferentes paradigmas, para permitir evolucionar en instancias posteriores de la carrera en el uso eficiente de cada lenguaje. Que el alumno sea capaz discernir entre la aplicación de uno u otro paradigma, en la resolución de un problema, ya que contará con los elementos conceptuales básicos elementales para el análisis y modelado de soluciones a los diferentes problemas. Que los alumnos exploren los distintos niveles de impedancia ante un mismo problema y como la solución cambia entre los distintos paradigmas trabajados.

Contenidos mínimos según plan de estudio

Lógica proposicional y de predicados. Paradigmas y lenguaje de programación. Conceptos fundamentales. Estilos de programación. Evaluación de expresiones. Definición de funciones. Tipos de datos. Programación Funcional. El tipo de datos. Funciones. Intervalos. Operadores. Aplicaciones de las listas. Árboles. Evaluación perezosa. La estrategia de evaluación perezosa. Técnicas de programación funcional perezosa. Fundamentos de la programación lógica. Repaso de la lógica de primer orden. Unificación y resolución. Cláusulas de Horn. Resolución SLD. Programación lógica con cláusulas de Horn. Semántica de los programas lógicos. Representación clausada del conocimiento. Consulta de una base de cláusulas. Espacios de búsqueda. Programación lógica con números, listas y árboles. Control de búsqueda en programas lógicos. Manipulación de términos. Predicados metalógicos. Programación Orientada a Objetos: Componentes, modelado. Integración de paradigmas. Comparación de paradigmas.

Propuesta metodológica

La cursada de la materia se organiza en clases teóricas y actividades prácticas. En las clases teóricas se imparten los conceptos básicos de la materia, induciendo a la participación de los alumnos en la resolución de ejercicios que permite la incorporación activa de los conocimientos. En las clases prácticas los alumnos contarán con el apoyo de las explicaciones del docente donde se describe cada trabajo práctico, y donde pueden realizar las consultas sobre la resolución de estos. Además, las clases de consultas son en sala equipada con computadoras y en aulas tradicionales.

Ajustes para estudiantes con discapacidad

La UNRN desarrolla políticas en torno a accesibilidad académica para estudiantes con discapacidad y acompañamiento de equipos docentes, en consonancia con las normativas nacionales y orientaciones del CIN. Al inicio de la cursada se espera contar con el relevamiento del espacio de APD (Asistencia Pedagógica en discapacidad) sobre los/las estudiantes que cursen la materia con el fin de identificar barreras y definir las configuraciones de apoyo necesarias.

Unidades

Nombre de la unidad: 1 - Lógica proposicional y de predicados

Semanas: 1,2

Contenidos: Introducir conceptos básicos de lógica para analizar el lenguaje escrito desde el punto de vista lógico.

Presentación del lenguaje y convenciones básicas para analizar la lógica proposicional.

Construcción del concepto de fórmula bien formada (fbf).

Interpretar el valor de verdad de una proposición compuesta en función de los valores de verdad de sus proposiciones atómicas. Clasificar proposiciones como tautologías, contingencias o contradicciones y analizar la validez de argumentos.

Construcción de un sistema formal y concepto de deducción de una fbf a partir de otras fbf.

Deducir algunas tautologías en el sistema formal.

Introducción y conceptos generales: Programas, paradigmas, lenguajes de programación y programación.

Actividades practicas: Trabajo Práctico Nro. 1: Proposiciones lógicas simples y complejas.

Bibliografía obligatoria: Copi, I. M. (1999). Introducción a la lógica. Eudeba.

Bibliografía complementaria: - Bel, Andrea L. & Del Punta, Jessica A. (2017). Notas Teóricas: Guía de Actividades (1° ed.). Editorial de la Universidad Nacional del Sur. Ediuns.

https://www.matematica.uns.edu.ar/ingresantes/Notas_Teoricas_y_Guia_de_Actividades2018.pdf

- Gutiérrez, C. M. (s. f.). Introducción a la Lógica. Universidad Complutense de Madrid, 44.

Gutiérrez - Introducción a la Lógica.pdf.

- Pons, C., Rosenfeld, R., & Smith, C. P. (2017). Lógica para Informática. Editorial de la Universidad Nacional de La Plata (EDULP). <https://doi.org/10.35537/10915/61426>

- Tapia, N. V. de, Tapia de Bibiloni, A., & Tapia, C. A. (1974). Matemática 1. Estrada.

Nombre de la unidad: 2 - Programación Lógica

Semanas: 3,4,5,6

Contenidos: Introducción a la Programación Lógica. Características. Elementos de la programación lógica: variables, constantes, términos compuestos, listas. Cláusulas y predicados. Reglas y hechos. Dominios de aplicación. 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.

Actividades practicas: Trabajo Práctico: Casos de resolución simple sobre paradigma lógico.
Trabajo Práctico: Casos de resolución compleja sobre paradigma lógico.

Bibliografía obligatoria: Bramer, M. (2005). Logic Programming with Prolog (2005a ed.). Springer.

Carpio Cañada, J., Aranda Corral, G. A., & de la Rosa, J. M. (2011). Programación declarativa. Universidad de Huelva.

Nilsson, U. (2012, mayo 7). Logic, programming and prolog (2ed). Liu.Se.

<https://www.ida.liu.se/~ulfni53/lpp/>

SWISH -- SWI-Prolog for SHaring. (s/f). Swi-prolog.org. Recuperado el 20 de julio de 2023, de <https://swish.swi-prolog.org/>

Bibliografía complementaria: ---

Nombre de la unidad: 3 - Programación Funcional

Semanas: 7,8,9,10

Contenidos: Características. Comparación lenguaje imperativo con lenguaje funcional. Definiciones de funciones. Script. Ventajas y desventajas. Áreas de Aplicación. Conceptos generales: Funciones matemáticas, Funciones de orden superior. Cálculo Lambda, Reglas y estrategias de reducción. Reducción de expresiones y forma normal. Expresión y valor. Transparencia referencial. Evaluación de las expresiones, mecanismo de reducción o simplificación. Orden aplicativo, orden normal (lazy evaluation). Tipos de datos básicos y derivados. Tipos de funciones. Expresiones polimórficas. Currificación. Cálculo Lambda. Dominios de aplicación. Lenguaje Haskell. Introducción. Entorno de Haskell – HUGS. Comentarios. Tipos de datos. Flujo de control. Definición de Funciones. Currificación. Reducción de expresiones. Evaluación. Tuplas. Recursividad. Listas. Tipos definidos por el usuario. Tipos polimórficos. Tipos recursivos. Otras implementaciones del paradigma: LISP, Erlang.

Actividades practicas: Trabajo Práctico: Casos de resolución simple sobre paradigma funcional.

Trabajo Práctico: Casos de resolución compleja sobre paradigma funcional.

Bibliografía obligatoria: Carpio Cañada, J., Aranda Corral, G. A., & de la Rosa, J. M. (2011). Programación declarativa. Universidad de Huelva.
Coding rooms - Haskell online IDE and compiler. (s/f). Coding Rooms. Recuperado el 20 de julio de 2023, de <https://qa-mktg.codingrooms.com/compiler/haskell/>
Seibel, P. (2005). Practical Common Lisp (1a ed.). APress.

Bibliografía complementaria: ---

Nombre de la unidad: 4 - Otros paradigmas

Semanas: 11,12,13

Contenidos: Lenguajes orientados a eventos – reactivos. VHDL, verilog. Codiseño de software y hardware. Orientados a flujos, Introducción a BPMNL Lenguajes Reflectivos, introducción a los lenguajes específicos de la aplicación (DSL) Metaprogramación: macros, metaclases, templates.

Actividades practicas: Trabajo Práctico: Resolución de casos en lenguajes orientados a eventos.

Trabajo Práctico: Resolución de casos en lenguajes orientados a flujos

Bibliografía obligatoria: BPMN 2.0. (s. f.). Omg.org. Recuperado 20 de julio de 2023, de <http://www.omg.org/spec/BPMN/2.0>

Schaumont, P. R. (2012). A practical introduction to hardware/software codesign (2.a ed.). Springer.

von Rosing, M., White, S., Cummins, F., & de Man, H. (2015). Business Process Model and Notation—BPMN. En The Complete Business Process Handbook (pp. 433-457). Elsevier.

Bibliografía complementaria: ---

Nombre de la unidad: 5 - Lenguajes multiparadigmas

Semanas: 14,15

Contenidos: Lenguajes débilmente tipados, integrables, etc. Caso de estudio: programación en un lenguaje que admita diferentes estrategias de programación (Python, por ejemplo).

Actividades practicas: Trabajo Práctico: Resolución de casos en lenguajes multiparadigmas.

Bibliografía obligatoria: Downey, A. B. (2009). Python for software design: How to think like a computer scientist. Cambridge University Press.

Langtangen, H. P. (2007). Python Scripting for Computational Science (H. P. Langtangen, Ed.; 3.a ed.). Springer.

Bibliografía complementaria: ---

Propuesta de evaluación

Con el fin de seguir el desempeño de los alumnos durante la cursada, se establece un mecanismo de evaluación que consiste en 2 exámenes. Quienes aprueben los dos exámenes, el 75% de los trabajos prácticos, y el trabajo final serán aprobados. La escala de calificación es de 1 a 10.

Asignatura posible de ser promocionada sin examen final

Condiciones de regularidad y régimen de promoción

Se establece que la materia es promocionable con la escala Promoción 8 - Promoción. El régimen resuelve las siguientes condiciones: Un alumno cumple con la condición de promoción cuando aprueba el 75% de los TPs, los parciales o sus recuperatorios, el trabajo final y el 75% de asistencias. El promedio de la nota de TPs, parciales o sus recuperatorios y trabajo final deberá ser mayor o igual a 8. Es promediable en cada uno de ellos

Asignatura posible de ser rendida libre

Se establece el régimen para las/los estudiantes que deseen rendir libre de las siguiente manera: Si, es posible rendir libre - Se evaluará el conocimiento teórico y práctico del alumno de forma tal que tenga un manejo adecuado de las herramientas del taller y explique todos los conceptos de la misma forma que los alumnos que han cursado y aprobado la asignatura en forma regular (No se prevén clases de consulta para quienes deseen rendir Libre).

Requisitos de acreditación

Regular: Un alumno cumple con la condición de regular cuando aprueba el 75% de los TPs, los parciales o sus recuperatorios, el trabajo final y el 75% de asistencias. El promedio de la nota de TPs, parciales o sus recuperatorios y trabajo final deberá ser mayor o igual a 4 y menor a 8. Es promediable en cada uno de ellos. Promoción: Un alumno cumple con la condición de promoción cuando aprueba el 75% de los TPs, los parciales o sus recuperatorios, el trabajo final y el 75% de asistencias. El promedio de la nota de TPs, parciales o sus recuperatorios y trabajo final deberá ser mayor o igual a 8. Es promediable en cada uno de ellos. Libre: Se evaluará el conocimiento teórico y práctico del alumno de forma tal que tenga un manejo adecuado de las herramientas del taller y explique todos los conceptos de la misma forma que los alumnos que han cursado y aprobado la asignatura en forma regular (No se prevén clases de consulta para quienes deseen rendir Libre).

Fechas tentativas de evaluación

Semana 8 y 16