

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

LENGUAJES DE PROGRAMACIÓN

CÓDIGO: 5696

ÁREA N°: II

CARRERAS

Licenciatura en Ciencias de la Computación

PROFESOR RESPONSABLE:

Dra. María Laura Cobo – Profesora Adjunta con Dedicación Exclusiva

CARGA HORARIA	Teoría 64 hs	Práctica 64 hs	Laboratorio	CANTIDAD DE SEMANAS	16
---------------	-----------------	-------------------	-------------	---------------------	----

CORRELATIVAS

PARA CURSAR LA MATERIA		PARA APROBAR LA MATERIA	
APROBADAS	CURSADAS	APROBADAS	CURSADAS
Lógica para Ciencias de la Computación	Sistemas Operativos y Distribuidos Bases de Datos	Sistemas Operativos y Distribuidos Bases de Datos	

DESCRIPCIÓN

El curso presenta conceptos básicos y avanzados de lenguajes de programación, como así también criterios de diseño y estrategias de implementación.

Durante el curso se utilizan una variedad de lenguajes, tanto pasados como presentes, con énfasis en lenguajes imperativos modernos y con menor profundidad en lenguajes funcionales y dinámicos (o de scripting).

En lugar de hacer foco en las características particulares de lenguajes particulares, se hace énfasis en los conceptos fundamentales y en las diferencias de aproximación entre lenguajes, las razones de esas diferencias y las implicaciones que las mismas tienen sobre la implementación del lenguaje.

Entre los tópicos específicos se pueden mencionar:

- Aspectos de especificación sintáctica y semántica
- Nombres, alcance y ligaduras
- Control de flujo, subrutinas, manejo de excepciones y concurrencia
- Sistema de tipos, mecanismos de abstracción de datos y manejo de almacenamiento
- Paradigmas: imperativo, funcional, basado en lógica y orientado a objetos.

La materia tiene vínculos fuertes con las materias iniciales de programación y las asignaturas *Lógica para Ciencias de la Computación*, *Bases de Datos* y *Sistemas Operativos y Distribuidos*.

Además de los contenidos conceptuales especificados en el programa se espera que los alumnos adquieran y refuercen aptitudes generales y específicas para:

- Construir definiciones usando tanto un lenguaje simbólico como natural.

De la Cobo

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

LENGUAJES DE PROGRAMACIÓN

CÓDIGO: 5696

ÁREA N°: II

- Seleccionar criterios adecuados para comparar el soporte para un determinado concepto o herramienta en diferentes lenguajes o paradigmas de programación
- Interpretar las similitudes y diferencias entre distintos enfoques referidos a un mismo tema y argumentar a favor o en contra de los mismos.
- Aprender nuevos lenguajes en forma autónoma, pudiendo realizar evaluaciones de los mismos en términos del soporte que brindan para los conceptos y paradigmas.

El objetivo del curso no es entrenar a los alumnos para ser expertos en el uso de algún lenguaje de programación, sino en brindarles habilidades para elegir rápidamente entre nuevos lenguajes, que la elección realizada sea apropiada de acuerdo a circunstancias variadas. Que también sean capaces de apreciar las posibles tendencias en el uso de lenguajes.

METODOLOGÍA DE ENSEÑANZA

En las clases teóricas, en general, la metodología de enseñanza es **expositiva, colectiva** y adopta un enfoque **vertical y deductivo** para el desarrollo de los contenidos. Cada contenido conceptual se presenta en general y luego se describen casos particulares. Aunque se promueve la participación del alumno con preguntas que motivan la reflexión y la vinculación de los contenidos de la materia con otros aprendidos previamente, el profesor conduce la clase de acuerdo a un cronograma establecido. Los contenidos son presentados al grupo completo de alumnos y el profesor detiene la explicación cuando surgen dudas individuales. Cada uno de los tópicos inherentes a lenguajes de programación es tratado individualmente y se ilustra tomando como ejemplo lenguajes de programación de diferentes características. Este enfoque permite efectuar un buen análisis comparativo de los diferentes conceptos planteados.

Para algunos temas particulares las clases se desarrollarán con una metodología de tipo taller, donde se proveen textos para generar debates sobre el uso e incorporación de algunos constructores en lenguajes de programación.

En las clases prácticas se adopta un enfoque **participativo, individual, horizontal e inductivo**. Se presentan las características de cuatro paradigmas de programación y el soporte para cada uno de ellos en al menos un lenguaje particular, para luego enunciar principios generales. En este enfoque es posible resaltar claramente las motivaciones que dieron lugar a cada lenguaje de programación particular o a una clase de lenguajes de programación. Cada paradigma se presenta en una clase **expositiva y colectiva**. En el resto de las clases prácticas el alumno debe asumir un rol más activo, los docentes atienden consultas y brindan explicaciones individuales o a grupos reducidos. El ritmo de avance depende de cada alumno o grupo.

La intención de combinar un enfoque vertical y otro horizontal, es que al terminar la materia ambos enfoques se hayan complementado para alcanzar las ventajas de cada uno de ellos.

Tanto en las clases teóricas como prácticas se busca lograr que los alumnos contrasten:

[Handwritten signature]

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

LENGUAJES DE PROGRAMACIÓN

CÓDIGO: 5696

ÁREA N°: II

- Principios Teóricos vs. Detalles Técnicos.
- Semántica vs. Sintaxis.
- Evolución vs. Historia.
- Aprender a Aprender, Evaluar y Comparar.

Las clases están diseñadas para desarrollar las capacidades de comparación y evaluación que permiten seleccionar el lenguaje más adecuado para cada aplicación en particular. Se orientan también a que los alumnos puedan aprender nuevos lenguajes de manera autónoma. La intención no es formar programadores expertos en ningún lenguaje particular sino captar el espíritu de cada lenguaje de programación tratado, reconocer sus capacidades y limitaciones para distintos dominios de aplicación.

MECANISMOS DE EVALUACIÓN

Para aprobar el cursado de la materia, el alumno deberá aprobar un examen parcial escrito o su correspondiente recuperatorio.

Se incluye además el desarrollo de un proyecto donde se evalúa la comprensión de los detalles de traducción e implementación asociados a los temas vistos. Esto resulta particularmente vinculantes a las competencias que se requieren en la materia correlativa Compiladores e Interpretes.

Para aprobar la materia, el alumno rinde un único examen teórico-práctico escrito. Los contenidos de ese examen abarcan todos los contenidos de la materia, a excepción de la traducción detallada de programas que se evalúa a nivel conceptual.

El examen final libre consta de dos instancias escritas, ambas teórico-prácticas y de un trabajo de investigación que consiste, generalmente, en establecer los criterios de comparación y efectivamente comparar algún elemento vinculado a la materia (por ejemplo sistema de tipo de dos lenguajes de programación nuevos para el alumno).

PROGRAMA SINTÉTICO

- Criterios de evaluación de lenguajes de programación.
- Principios de diseño de lenguajes de programación
- Historia de la Computación: Evolución de los Lenguajes de Programación.
- Lenguajes de Programación: Entidades y ligaduras. Atributos. Variables, Constantes, Expresiones.
- Tipos de datos. Tipos de datos recursivos. Sistemas de tipos, niveles de polimorfismo.
- Instrucciones. Unidades.
- Encapsulamiento y abstracción.
- Concurrencia y paralelismo.

Paula

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

LENGUAJES DE PROGRAMACIÓN

CÓDIGO: 5696

ÁREA N°: II

- Procesamiento de lenguajes de programación: Conceptos de intérpretes y compiladores.
- Criterios de Diseño e Implementación de Lenguajes de Programación.
- Representación de datos en memoria. Estrategias de implementación.
- Sintaxis de los lenguajes de programación. Nociones básicas de semántica formal.
- Paradigmas de Programación: Imperativo, Orientado a Objetos, Funcional, Lógico.

PROGRAMA ANALÍTICO

1) Introducción

- Contexto: El proceso de desarrollo de software. Dominios de Aplicación. Metodologías de Software y Lenguajes de Programación. Arquitectura de computadoras y Lenguajes de Programación. Evolución de los Lenguajes de Programación.
- Criterios de Evaluación de Lenguajes de Programación: Metas de diseño: confiabilidad, portabilidad, extensibilidad y eficiencia. Principios: simplicidad, expresividad, seguridad, flexibilidad, robustez, ortogonalidad, eficiencia, reusabilidad, extensibilidad.
- Estructuras básicas: Diseño e Implementación. Intérpretes y Compiladores. Procesadores Abstractos. Estructura de la memoria: la pila, el heap.
- Paradigmas: Definición. Generalidades y fundamentos de cada paradigma

2) Definición de lenguajes de Programación (Sintaxis y Semántica)

- Sintaxis y Semántica. Matemática y Programación. Especificaciones Formales. Beneficios y Limitaciones.
- Sintaxis concreta y abstracta. Notaciones Formales: BNF, Grafos sintácticos. Gramáticas libres de contexto. Parsing.
- Semántica estática. Gramática de atributos. Semántica estática y dinámica. Semántica axiomática. Semántica operacional. Especificación a partir de una máquina virtual.

3) Entidades, Atributos y Ligaduras:

- Ligadura. Tiempos de Ligadura: ligadura estática y dinámica. Declaraciones. Ambiente de referenciamiento. Alcance y Visibilidad. Estructura general de un programa. Los conceptos son presentados en el contexto de cada uno de los paradigmas, enfatizando las diferencias que pueden llegar a existir.

4) Variables, Constantes y Expresiones:

- Nombres: Atributos de las variables y constantes: nombre, alcance, tipo, tiempo de vida, valor, locación. Representación en memoria. Aproximación desde cada paradigma.
- Expresiones: clasificación, notaciones, control. Efectos colaterales. Transparencia referencial. Variables y expresiones en los lenguajes declarativos. Relevancia y vinculación a los distintos paradigmas. Formas de evaluación y detalles de implementación.
- Clasificación de los lenguajes: Reglas de alcance: alcance estático y dinámico. Acceso al

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

LENGUAJES DE PROGRAMACIÓN

CÓDIGO: 5696

ÁREA N°: II

entorno local y global. Alocación estática, semiestática y semidinámica de memoria. La pila. Alocación dinámica de memoria.

5) Tipo de datos:

- Niveles de definición. Clasificación. Sistemas de tipos. Equivalencia de Tipos. Chequeos. Polimorfismo. Inferencia de Tipos.
- Detalles de implementación: Administración de memoria. Vinculación a variables y expresiones.

6) Instrucciones y unidades:

- Instrucciones: Asignación. Estructuras de Control a nivel instrucción: Secuencia, Transferencia de Control, Condicional e Iteración. Aproximación a los conceptos desde cada paradigma.
- Unidades: Procedimientos y Funciones. Pasaje de Parámetros. Datos y Procedimientos como Parámetros. Pasaje por valor, por referencia, por nombre y por nombre. Aproximación a los conceptos desde cada paradigma. Estructuras de Control a nivel Unidad.
- Implementación de instrucciones y unidades: Administración de la memoria. Asignación. Transferencia de control. Condicional. Iteración. Registro de Activación.

7) Encapsulamiento y abstracción

- Conceptos: Evolución del Concepto de Tipo. Encapsulamiento y Modularización. Abstracción de Datos y Tipos de Datos Abstractos. Genericidad y Tipos Parametrizados Orientación a Objetos y Herencia. Aproximación a los conceptos desde cada paradigma.
- Implementación: Diferencias en la aproximación de los diferentes lenguajes. administración de memoria para lenguajes orientados a objetos. Polimorfismo: genericidad y herencia. Asignación dinámica de código.

8) Concurrencia

- Motivación. Aspectos fundamentales para la programación concurrente (desde la perspectiva de un lenguaje de programación). Implementación de la sincronización. Constructores a nivel lenguaje. Aproximación a los conceptos desde cada paradigma.

BIBLIOGRAFÍA

Bibliografía Básica

- **Michael L. Scott**, Programming Languages Pragmatics. 4^{ta} Edición. Morgan Kaufmann Publishers, diciembre 2015.
- **Sebesta Robert W.**, Concepts of Programming Languages, 11^{ava} Edición. Addison Wesley, febrero 2015.

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

LENGUAJES DE PROGRAMACIÓN

CÓDIGO: 5696

ÁREA N°: II

Bibliografía Adicional

- **Roosta Seyed H.**, Foundations of Programming Languages: Design and Implementation. Brooks/Cole. 2003
- **Mitchell John C.**, Concepts in Programming Languages. Cambridge University Press. 2003.
- **Harper Robert** Practical Foundations for Programming Languages. University Press, 2013.
- **Krishnamuthi Shiram** Programming Languages: Applications and Interpretation. 2007.

AÑO

2019

FIRMA PROFESOR RESPONSABLE

VISADO

COORDINADOR ÁREA

SECRETARIO ACADÉMICO

DIRECTOR
DEPARTAMENTO

Dr. DIEGO C. MARTINEZ
SECRETARIO ACADÉMICO
DPTO. DE CS. E ING. DE LA COMP.
UNIVERSIDAD NACIONAL DEL SUR

Dr. MARCELO A. FALAPPA
DIRECTOR DEPARTAMENTO
DPTO. DE CS. E ING. DE LA COMP.
UNIVERSIDAD NACIONAL DEL SUR