



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

**CONCEPTOS Y PARADIGMAS DE
LENGUAJES DE PROGRAMACION**

Año 2015

Carrera:
Licenciatura en Informática Plan 2003/07
Licenciatura en Sistemas Plan 2003 /07

Año: 3°

Duración: **semestral** (1er S)

Carácter: **obligatoria**

Profesor a cargo:

Profesor Adjunto **Lic. Viviana Harari**

Hs. semanales: **6 hs.**

FUNDAMENTACION:

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 lenguajes ya conocidos e incorporar elementos para abordar los nuevos. Como formación complementaria se alienta y evalúa el trabajo colaborativo y la capacidad de presentar y exponer trabajos en forma escrita y oral.

OBJETIVOS GENERALES:

Adquirir la capacidad de evaluar lenguajes de programación desde distintos puntos de vista, ya sea como su diseñador o implementador o como usuario del lenguaje. Introducir, analizar, comparar y evaluar los conceptos subyacentes de los Lenguajes de Programación en los distintos Paradigmas.

CONTENIDOS MINIMOS:

- Sintaxis y semántica.
- Semántica operacional.
- Entidades y ligaduras.
- Sistemas de tipos.
- Encapsulamiento y abstracción.
- Intérpretes y Compiladores.
- Paradigmas de lenguajes (imperativo, orientado a objetos, funcional, lógico).
- Programación basada en scripting.



PROGRAMA ANALITICO

- 1- Lenguajes de Programación como herramientas para producir software. Objetivo del estudio de sus conceptos. Distintos puntos de vista. Capacidad de comunicación. Relación e integración de los lenguajes de programación respecto a otros conceptos.
- 2- Evaluación de los lenguajes de programación a través de las características del software que producen. Principios de diseño de los lenguajes. Perspectiva histórica de los lenguajes de programación motivación, herencia, características, evolución. Desde Fortran hasta Java. Lenguajes funcionales. Lenguajes Orientados a objetos. Abstracción: que papel juega. Necesidad de estandarizar: ¿cuándo y cómo hacerlo?
- 3- Estructura de un lenguaje: sintaxis y semántica.
Sintaxis: Características de las sintaxis. Elementos de las sintaxis. Estructuras sintácticas. Reglas léxicas y sintácticas. Tipos de sintaxis. Sintaxis abstracta y concreta.
Formas de definir la sintaxis. Gramáticas. (Backus Naum Form). Árboles sintácticos y de derivación. Gramáticas recursivas. Subgramáticas. Gramáticas ambiguas. EBNF. Diagramas sintácticos (CONWAY)
Semántica estática.
- 4- Semántica. Tipos de semánticas. Formas de definir la semántica de un lenguaje de programación. Nociones de semántica formal.
Procesamiento de un lenguaje interpretación y traducción. Tipos de traductores. Comparación entre Traductor e Intérprete. Combinación de ambas técnicas. Compiladores. Etapas de Análisis y Síntesis. Optimización.
- 5- Semántica Operacional.
Ligadura. Descriptores. Momentos de ligadura. Estabilidad.
Variables. Arquitectura Von Newman. Atributos. Momentos y estabilidad. Nombre: características. Alcance: visibilidad, reglas. Tipo: definición, clasificación. L-valor: tiempo de vida, asignación. R-valor: constantes, inicialización. Alias
- 6- Unidades. Atributos. Representación en ejecución. Elementos. Unidades recursivas. Unidades genéricas. Alias y sobrecarga. Procesador abstracto: elementos, instrucciones.
Procesamiento de un lenguaje: clasificación. Lenguaje estático. Entidades locales. Rutinas internas. Compilación separada. Lenguajes basados en pila. Unidades recursivas, implementación. Estructura de bloque. Datos semidinámicos y dinámicos. Lenguajes dinámicos.
- 7- Compartir Datos. Ambiente común. Acceso al ambiente no-local. Parámetros. Ventajas. Evaluación de los parámetros reales y ligadura con los parámetros



- formales. Clase de parámetros: Datos y Subprogramas. Modos de pasaje de parámetros datos. Pasaje de Rutinas como parámetros.
- 8- Sistema de tipos. Tipos predefinidos, tipos definidos por el usuario, tipos estructurados, tipos abstractos. Implementación de datos, su representación. Sistema de tipos: lenguajes seguros y fuertemente tipados. Seguridad en el manejo de tipos. Encapsulamiento y abstracción. Evolución de los tipos. Tipos Abstractos. Equivalencia de tipos.
 - 9- Abstracción de Control. Estructuras de control: Definición de estructuras de control a nivel de sentencia y a nivel de unidad. Tipos de estructuras de control a nivel de sentencia. Diferencia entre sentencia de asignación y expresión. Evolución de las sentencias de selección e iteración en diferentes lenguajes, desde PL/1 a Java.
 - 10- Abstracción de control a nivel de unidad. Excepciones: Definición. Modelos de Terminación y Reasunción. Distintos modelos de manejo de excepciones. Comparación.
 - 11- Paradigma funcional. Características. Comparación lenguaje imperativo con lenguaje funcional. Definiciones de funciones. Script. 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.
 - 12- Paradigma Orientado a Objetos. Características. Elementos básicos de la programación orientada a objetos: objetos, mensajes, métodos, clases. Conceptos de generalización, especificación y herencia. Diferentes tipos de herencia. Lenguajes híbridos, características principales. Dominios de aplicación.
 - 13- Paradigma lógico. Características. Elementos de la programación lógica: variables, constantes, términos compuestos, listas. Cláusulas y predicados. Reglas y hechos. Dominios de aplicación.
 - 14- Programación basada en scripting. Definición. Introducción histórica. Características. Tipos. Dominios de interés. Los lenguajes de scripting y la WWW. Aspectos innovadores.

METODOLOGIA DE ENSEÑANZA

Las actividades se desarrollan bajo la modalidad de teorías y practicas distribuidas equitativamente de acuerdo con el Plan de Estudios vigente.

Los contenidos del programa se presentan y analizan en las clases teóricas y se



consolidan con los trabajos prácticos. Existe una estrecha relación entre la teoría y práctica. La teoría toma sentido y se fortalece con las prácticas, a su vez la práctica fundamenta sus procedimientos en la teoría.

Los alumnos desarrollan actividades individuales y grupales que debieran conducirlos a un seguimiento y avance gradual sobre los temas presentados. En las prácticas, se presentan dos ejercicios integradores, cuya entrega es obligatoria, en dónde se plantean diferentes actividades de investigación y desarrollo que deben llevar a cabo los estudiantes para poder realizar la entrega. Las mismas están relacionadas con la aplicación, en diferentes lenguajes de programación, de los conceptos abordados en la materia. Esto permite que el alumno pueda relacionar los temas permitiéndole avanzar en la maduración de los conceptos.

Se pretende fomentar el trabajo en grupo y la adquisición de experiencia en la comunicación escrita y oral de sus trabajos.

Se ofrecen tres horarios de práctica distribuidos en la semana en distintas franjas (mañana, tarde y noche) a fin de cubrir todas las posibilidades que faciliten la asistencia.

Los materiales didácticos son los tradicionales, haciendo uso del cañón y PC disponible en las aulas para apoyar el desarrollo de los temas a través de aplicaciones de presentación.

Se pone a disposición de los alumnos una plataforma web con la que se mantiene una comunicación dinámica. En ella se publican el cronograma de toda la cursada, programa, bibliografía, guías de clases teóricas y trabajos prácticos. También se utiliza este medio para comunicar novedades, recordar fechas y publicar resultados de las correcciones. Los alumnos pueden comunicarse con los profesores y el jefe a través de sus cuentas de e-mail

El equipo docente mantiene reuniones al comienzo de cada ciclo lectivo para delinear la planificación anual a la vista de la evaluación de los resultados precedentes y luego se mantiene en comunicación permanente para ajustes puntuales. Asimismo se realiza la revisión continua de los temas desarrollados, ya sea para modificar la metodología utilizada como para introducir o modificar conceptos.

EVALUACION

Para la aprobación de la materia, los alumnos deben realizar la presentación de dos ejercicios de investigación, integradores de las temáticas abordadas y, aprobar dos exámenes parciales, divididos en parte práctica y parte teórica (con tres fechas cada uno).

Las fechas de cada uno de los parciales se establecen de acuerdo a la duración del semestre fijado por el Calendario Académico de la Facultad y son publicados al principio del ciclo lectivo y se organizan de forma tal de no entorpecer el normal desarrollo de las cursadas

Para la evaluación final de la asignatura se cuenta con dos modalidades. La primera es una evaluación escrita individual con preguntas conceptuales a desarrollar.

La segunda consiste en el desarrollo, con presentación escrita y oral, de un trabajo de investigación sobre un tema específico aplicado a uno o varios lenguajes de programación. Este último será realizado en forma grupal.



BIBLIOGRAFIA OBLIGATORIA

- GHEZZI C. – JAZAYERI M.: Programming language concepts. John Wiley and Sons. (1998) 3er. Ed
- SEBESTA: Concepts of Programming languages. Benjamin/Cumming. (2010) 9a. Ed.
- LOUDEN K.C.: Programming languages: principles and practices (2011)
- PRATT: Programming Languages. Design and Implementation. Prentice Hall (2001) 4ta. Ed.
- SETHI R.: Programming languages: concepts and constructs. Addison – Wesley (1996) 2nd. Ed.

BIBLIOGRAFIA COMPLEMENTARIA

- FRIEDAMAN D.: Essentials of programming languages (2008)
- HOROWITZ: Fundamentals of Programming Languages. Spring- Verlar (1984)
- SCOTT M.L.: programming language pragmatics (3er. Ed.) (2009)
- TURCKER A.: Programming languages (2006)
- WATT D. Programming Language Design Concepts (2004)



CRONOGRAMA CLASES Y EVALUACIONES

Clase	Contenido/ Actividad	Evaluaciones previstas
1	PRESENTACION DE LA ASIGNATURA: pautas Introducción, evaluación, historia de los lenguajes	
2	ESTRUCTURA DE UN LENGUAJE: sintaxis y semántica. Sintaxis.	
3	SEMANTICAS. Procesamiento de un lenguaje: interpretación y compilación	
4	SEMANTICA OPERACIONAL: entidades y ligaduras. Variables	
5	SEMANTICA OPERACIONAL: unidades. Procesamiento de lenguajes estáticos	
6	SEMANTICA OPERACIONAL: unidades recursivas, cadena estática. Acceso al ambiente no local	
7	SEMANTICA OPERACIONAL: comportamientos mas dinámicos. Cadena dinámica	Ejercicio integrador. Presentación y debate
8	SEMANTICA OPERACIONAL: conclusiones, diferenciación y reconocimiento de atributos, repaso.	
9	COMPARTIR DATOS: casos. Parámetros	Primer parcial primer fecha
10	SISTEMA DE TIPOS: características, tipos predefinidos y tipos definidos por el usuario.	
11	SISTEMA DE TIPOS: tipos compuestos y tipos abstractos.	Primer parcial 2da fecha
12	SISTEMA DE TIPOS: lenguajes seguros, equivalencia y compatibilidad.	
13	SISTEMA DE TIPOS: implementación de los tipos	
14	ABSTRACCION DE CONTROL: a nivel de sentncia. Asignaciones y expresiones.	
15	ABASTRACCION DE CONTROL: a nivel de unidad. Excepciones	Ejercicio integrador. Presentación y debate
16	PARADIGMA OO	Segundo parcial primer fecha
17	PARADIGMA FUNCIONAL	Distribución temas trabajo final: pautas.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

		Encuesta
18	PARADIGMA LOGICO	Segundo parcial, segunda fecha
19	TRABAJOS FINALES: seguimiento grado de avance, ajuste de pautas.	Recuperación : tercer fecha del primer y segundo parcial y recuperación general
20		Presentación Trabajo Final
21		Presentación Trabajo Final

Comunicación con la asignatura, a través de:

<https://catedras.info.unlp.edu.ar>
vharari@info.unlp.edu.ar
dalila_r@info.unlp.edu.ar

Prof, Harari Viviana