



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

**CONCEPTOS Y PARADIGMAS DE
LENGUAJES DE PROGRAMACION**

Año 2010

Carrera: **Licenciatura en Informática**
Plan 2003-2007
Licenciatura en Sistemas
Plan 2003-2007

Año: **3°**

Duración: **Semestral**

Profesor: **Prof. Begoña Rodríguez-**
Prof. Viviana Harari

Hs. semanales: **6 hs.**

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).

Programa

- 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.
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, alocaión. 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.
Encapsulamiento y abstracción. Evolución de los tipos. Tipos abstractos. Equivalencias 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

METODOLOGÍA DE ENSEÑANZA

Las actividades se desarrollan bajo la modalidad de teorías y prácticas 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 que deben construir los estudiantes con el objetivo de relacionar los temas y favorecer un adecuado avance 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 distintos horarios (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 disponibles 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 con los estudiantes, en ella se publican el cronograma de toda la cursada, programa, bibliografía, guías de clases teóricas. 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 inicial a la vista de la evaluación de los resultados precedentes y luego se mantiene en comunicación permanente para ajustes puntuales. Así mismo se realiza la revisión continua de los temas desarrollados, ya sea para modificar la metodología utilizada como para introducir o modificar conceptos.

PROPUESTA DE EVALUACION

Para obtener la cursada los alumnos deben haber presentado y aprobado los ejercicios integradores y aprobar dos exámenes parciales (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 tienen dos modalidades. La primera es una evaluación escrita individual con preguntas conceptuales a desarrollar.

La segunda consiste en la elección, presentación escrita y defensa oral de un trabajo de investigación sobre algún lenguaje en forma grupal, complementado con una evaluación escrita reducida e individual.

Las fechas de los exámenes finales están establecidas en el calendario académico de la facultad.

Por otra parte para la autoevaluación del desarrollo de la cursada se realiza una encuesta a los alumnos, consultando sobre su percepción de distintos aspectos. El resultado de estas encuestas es un insumo importante para todos los integrantes de la asignatura a la hora de evaluarlos y pensar en modificaciones a introducir para mejorar.

BIBLIOGRAFIA

- GHEZZI C. – JAZAYERI M.: Programming lenguaje concepts. John Wilwy and Sons. (1998) 3er. Ed.
- SEBESTA: Concepts of Programming lenguajes. Benjamin/Cumming. (2010) 9na. Ed.
- WATT D. Programming Lenguaje Design Concepts (2004).
- PRATT: Programming Lenguajes. Design and Implementation. Prentice Hall (2001) 4ta. Ed.
- SETHI R.: Programming lenguajes: concepts and constructs. Addison – Wesley (1996) 2nd. Ed.
- HOROWITZ: Fundamentals of Programins Lenguajes. Spring- Verlar (1984)



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA
