

**UNIVERSIDAD NACIONAL DE ASUNCIÓN**  
**FACULTAD POLITÉCNICA**  
**CIENCIAS INFORMÁTICAS**  
**PLAN 2009**  
**PROGRAMA DE ESTUDIOS**

**I. - IDENTIFICACIÓN**

1. Asignatura	: Paradigmas de la Programación
2. Código	: 4.2
3. Horas semanales	: 5 horas
3.1. Clases teóricas	: 2 horas
3.2. Clases prácticas	: 3 horas
4. Total real de horas disponibles	: 85 horas
4.1. Total de Clases teóricas	: 34 horas
4.2. Total de clases prácticas	: 51 horas

**II. - JUSTIFICACIÓN**

La comprensión profunda de cada paradigma de programación es más importante que un estudio de los diferentes lenguajes. Un paradigma de programación representa un enfoque particular o filosofía para la construcción del software. La asignatura incluye temáticas relacionadas con los paradigmas de programación y da un especial enfoque al paradigma orientado a objetos, de tal manera que se permita a los estudiantes obtener una visión nueva para abordar los problemas de la vida real y acercarse un poco más a la solución de problemas que son estudiados en el campo de la construcción del software. También incluye el paradigma de programación funcional, que al igual que el anterior, se constituye en una forma novedosa de resolver problemas de programación.

En el curso utilizaremos un lenguaje de programación, el lenguaje que representante del paradigma de programación orientada a objetos. La utilización de un lenguaje tan sencillo nos permita reflexionar sobre cuáles son los elementos esenciales de los lenguajes de programación y su semántica. También nos permite ir introduciendo al paradigma orientada a objetos y elementos de programación mediante la extensión del lenguaje.

**III. - OBJETIVOS**

Al elaborar un modelo para resolver un problema mediante programación, existen diferentes enfoques sobre cómo se debe realizar la abstracción de los diferentes elementos de dicho problema. Así, dependiendo de la situación que se desea modelar, cada uno de estos distintos enfoques o paradigmas de programación tiene ventajas y desventajas, que facilitan o entorpecen la construcción de un programa. El objetivo principal del curso es estudiar de manera comparativa los diferentes paradigmas de programación existentes, y conocer los criterios más importantes para seleccionar un lenguaje determinado.

Al terminar el curso, se pretende que el estudiante haya adquirido el conocimiento necesario para:

1. Esbozar la historia de los lenguajes de programación y la manera en que estos han evolucionado para adaptarse a las necesidades de desarrollo de software.
2. Reconocer e identificar los tres principales paradigmas de programación existentes (según el modelo teórico que los sustenta) y sus características principales.
3. Identificar las características de cada lenguaje de programación, su implementación y ambiente de ejecución.
4. Identificar las particularidades de un lenguaje de programación que pueden afectar de una u otra manera la construcción de un programa.
5. Aprovechar las características de los lenguajes y las técnicas de programación orientada a objetos.
6. Emplear técnicas funcionales en la construcción de aplicaciones generales.
7. Comprender la especificación formal de condiciones de salida para la comprobación de resultados de un programa.

**IV. - PRE – REQUISITO**

1. Estructura de Datos.

**V. - CONTENIDO**

**5.1. Unidades programáticas**

1. Características generales de los lenguajes de programación.
2. Paradigma Programación Orientada a Objeto.

**5.2. Desarrollo de las unidades programáticas**

1. Características generales de los lenguajes de programación.
  - 1.1. Introducción a los paradigmas
    - 1.1.1. Definición teórica
    - 1.1.2. Paradigmas de programación
      - 1.1.2.1. El paradigma Imperativo o por Procedimientos
      - 1.1.2.2. El paradigma Funcional

- 1.1.2.3. El paradigma Lógico
- 1.1.2.4. El paradigma Orientado a Objetos
- 1.2. Introducción a los lenguajes de programación
  - 1.2.1. Historia de los lenguajes de programación
  - 1.2.2. Problemas de traducción
  - 1.2.3. Gramáticas y definición formal de lenguajes de programación
  - 1.2.4. Analizadores sintácticos y generación de códigos
- 1.3. Lenguaje de programación y arquitectura
  - 1.3.1. Calidad del software
    - 1.3.1.1. Factores externos
    - 1.3.1.2. Factores internos
  - 1.3.2. Tipos de datos y mecanismos de abstracción
  - 1.3.3. Métodos de encapsulamiento
- 1.4. Especificación y control
  - 1.4.1. Control de secuencia
  - 1.4.2. Control de subprogramas
- 1.5. Herencia y polimorfismo – conceptos generales e implementación.
- 2. Paradigma Programación Orientada a Objeto.
  - 2.1. Tecnología Orientada a objetos – Como se piensa en Objetos
    - 2.1.1. Una perspectiva histórica
    - 2.1.2. Ventajas de este paradigma
  - 2.2. El modelo Orientado a objeto
    - 2.2.1. Objetos
      - 2.2.1.1. Estructura
      - 2.2.1.2. Relaciones
      - 2.2.1.3. Propiedades
      - 2.2.1.4. Métodos
    - 2.2.2. Clases
    - 2.2.3. Herencia
    - 2.2.4. Envío de mensajes
  - 2.3. Características asociadas a la Programación Orientada a Objetos
    - 2.3.1. Abstracción
    - 2.3.2. Encapsulamiento
    - 2.3.3. Ocultamiento
    - 2.3.4. Polimorfismo

## VI. - ESTRATEGIAS METODOLÓGICAS

El curso se basa en la exposición magistral por parte del profesor de cada uno de los contenidos descritos. Además, se harán sesiones de resolución de problemas y prácticas de laboratorio para enfrentar al estudiante de manera directa con las principales dificultades y técnicas utilizadas en programación.

De esta manera también se logra que el estudiante aplique los conceptos expuestos en situaciones específicas.

También habrá proyectos de programación que requieren más tiempo del que se dispone durante las lecciones, donde los estudiantes resuelven en grupo ejercicios de dificultad media o alta, para conocer, estudiar y resolver problemas representativos.

También se harán exposiciones sobre diferentes temas, cuyo estudio no puede hacerse dentro de las limitaciones de tiempo de la clase, pero cuyo conocimiento se considera importante para conseguir cumplir cabalmente el objetivo general del curso.

## VII. - MEDIOS AUXILIARES

1. Pizarra.
2. Proyector multimedia
3. Equipos de laboratorio PC, software de lenguajes de programación y base de datos para las prácticas
4. Ejercitarios.

## VIII. - EVALUACIÓN

1. Diagnóstica, Formativa y Sumativa (exámenes parciales y finales).
2. Participación en clase.
3. Presentación y exposición del trabajo monográfico.
4. Trabajos de investigación y laboratorios

Los exámenes en el curso buscan medir y evaluar la comprensión de cada estudiante del material estudiado durante el curso y del trabajo realizado en los proyectos. Los exámenes deben realizarse y entregarse individualmente, aunque la realización de los proyectos pueden completarse en grupos de dos personas como máximo.

## IX. - BIBLIOGRAFÍA

- Tucker, Allen. Lenguajes de programación, principios y paradigmas/ Allen Tucker, Robert Noonan,
- Meyer, Bertrand. Construcción de software orientado a objetos / Bertrand Meyer : Prentice Hall

- Java Como programar / Deitel : Pearson-Prentice Hall
- <http://www.javapassion.com/>