



Programa de la Asignatura:

# Lenguajes de Programación II



Código: 749

Carrera: **Ingeniería en Computación**  
Unidad Académica: **Secretaría Académica**  
Departamento: **Ingeniería**  
Formación Experimental: **00 %**

Plan: **2008**  
Curso: **Segundo Año – Primer cuatrimestre**  
Carga horaria total: **60 hs.**  
Formación teórica: **50 %**

Carácter: **Obligatoria**  
Carga horaria semanal: **4 hs.**  
Formación práctica: **50 %**

## Materias Correlativas Obligatorias

- **Lenguajes de Programación I (cód. 745)**
- -----
- -----

## Cuerpo Docente

Ismael, Sebastián  
Wachenhauzer, Rosa

## Índice

- Fundamentación pág. 2
- Encuadre y articulación de la asignatura pág. 2
  - Encuadre dentro del Plan de Estudios pág. 2
  - Articulación Horizontal pág. 2
  - Articulación Vertical pág. 2
- Objetivos pág. 2
  - Objetivo General pág. 2
  - Objetivos Específicos pág. 2
- Contenidos mínimos pág. 2
- Programa analítico pág. 3
- Bibliografía básica pág. 3
- Bibliografía de consulta pág. 3
- Metodología del aprendizaje pág. 3
  - Desarrollo de la asignatura pág. 3
  - Dinámica del dictado de las clases pág. 4
  - Trabajos prácticos pág. 4
- Metodología de evaluación pág. 4
- Planificación pág. 4
- Información de versiones pág. 5

AÑO ACADÉMICO 2013

ÚLTIMA REVISIÓN 19/06/2013

Firma Docente

Firma Coordinador

## 1. FUNDAMENTACION

La asignatura está incluida entre las materias que brindan a los alumnos los conocimientos necesarios para poder desarrollar soluciones programáticas, utilizando un lenguaje de programación para representar resoluciones a los problemas planteados. En particular, tiene el rol de profundizar los aspectos específicos de la sintaxis y semántica de los lenguajes en cuanto soportan los conceptos aprendidos.

Siguiendo con el enfoque de las materias anteriores de programación (Estructura de Datos I y Lenguaje de Programación I), se aplica el Currículum Invertido propuesto por Bertrand Meyer.

## 2. ENCUADRE Y ARTICULACIÓN DE LA ASIGNATURA

### Encuadre dentro del Plan de Estudios

La asignatura está ubicada en la currícula de la carrera en la parte inicial de la misma que corresponde al "Tecnologías Básicas".

En esa etapa, los alumnos continúan con la adquisición de experiencia en el área de la programación.

Requieren los conceptos de lógica, estructuras de control de flujo, nociones de programación orientada a objetos, modularidad y contrato.

Esta materia es correlativa de Lenguaje de Programación I.

### Articulación Horizontal

En cuanto a la articulación horizontal se ha planeado que esta asignatura se curse en simultáneo con Estructuras de Datos II, de modo que esa materia aporta una perspectiva más profunda sobre las estructuras de datos y a su vez se complementa con los aportes de esta materia en cuanto a herramientas del lenguaje.

### Articulación Vertical

Esta asignatura se articula verticalmente con Lenguaje de Programación I que brinda los conocimientos básicos del funcionamiento de un lenguaje de programación, y los conceptos básicos de Programación Orientada a Objetos.

Por otra parte, a esta materia le seguirá en un curso superior la asignatura Lenguaje de Programación III que avanza en el desarrollo de conceptos de programación de bajo nivel, en particular referencias y punteros.

## 3. OBJETIVOS

### Objetivo General

La cátedra se ha fijado como "objetivo cognoscitivo" de esta materia,

***"Proveer herramientas modernas y específicas de los lenguajes para plasmar los conceptos aprendidos en las materias de programación anteriores".***

### Objetivos Específicos

- Formalizar conocimientos de Herencia
- Formalizar conocimientos de Polimorfismo
- Desarrollar conocimientos de entrada/salida y archivos

## 4. CONTENIDOS MÍNIMOS

Programación Orientada a Objetos: Herencia y Polimorfismo. Clases Abstractas. Interfaces. Redefinición y Sobrecarga. Ligadura Tardía. Excepciones. Entrada/Salida. Archivos de texto. Archivos binarios. Pruebas. Patrones de Diseño.

## 5. PROGRAMA ANALÍTICO

**Unidad 1:** Programación Orientada a Objetos: Herencia. Herencia de comportamiento. Herencia de tipos.

**Unidad 2:** Métodos abstractos. Clases Abstractas. Redefinición y Sobrecarga.

**Unidad 3:** Polimorfismo de clases y de métodos. Ligadura tardía (dynamic binding).

**Unidad 4:** Interfaces. Herencia múltiple.

**Unidad 5:** Herencia y contrato. Identificación de tipos en tiempo de ejecución.

**Unidad 6:** Pruebas. Pruebas unitarias. JUnit.

**Unidad 7:** Entrada/Salida. Flujos (*streams*). Archivos texto. Archivos binarios

**Unidad 8:** Ruptura de contrato. Excepciones.

**Unidad 9:** Patrones de Diseño. Reutilización. Mecanismos de reutilización. Clasificación de patrones. Ejemplos de patrones.

## 6. BIBLIOGRAFÍA BÁSICA

- MEYER , BERTRAND, Construcción de Software Orientada a Objetos, Prentice-Hall, 1985, 2da. Edición 1997.

## 7. BIBLIOGRAFÍA DE CONSULTA

- MEYER, BERTRAND, Touch of class, Springer, 2009.
- ARNOLD, GOSLING, HOLMES, El lenguaje de programación Java, Addison-Wesley, 3ra. Edición 2002.
- ECKEL, BRUCE, Piensa en Java, Prentice Hall, 2007.
- GAMMA, HELM, JOHNSON, VLISSIDES, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.

## 8. METODOLOGÍA DEL APRENDIZAJE

### 8.a DESARROLLO DE LA ASIGNATURA

Inicialmente, cada unidad temática se aborda presentado al alumno una perspectiva teórica de los conceptos básicos involucrados para luego exponer ejemplos prácticos que permiten desarrollar los aspectos avanzados.

Posteriormente, se realiza la aplicación de los conceptos adquiridos en ejercicios prácticos que los alumnos desarrollan con la asistencia de los docentes.

Finalmente, se plantean problemas, de creciente nivel de complejidad y carácter integrador, cuya resolución involucra la aplicación del total de temas vistos en la materia.

En todo momento, se busca instaurar un ámbito crítico y reflexivo tanto de los conocimientos previamente adquiridos como también de los presentados en ese momento.

### 8.b DINÁMICA DEL DICTADO DE LAS CLASES

Para poder llevar adelante los objetivos de la cátedra, se hace un gran énfasis hacia la carga horaria de las aproximaciones a problemas, dando tanto peso a éstas como a la teoría.

El Profesor a cargo del curso se ocupará en forma personal y semanal del dictado de aquellos temas con un fuerte contenido teórico y que significan conceptos básicos y poco volátiles en la especialidad. Procederá a describir técnicas, características y pondrá ejemplos. Éste generará un ámbito de reflexión y discusión de los temas presentados, para que mediante la intervención de los alumnos, se puedan aclarar aquellos aspectos que el docente puede captar a través de las consultas recibidas, como los que han resultado de más compleja comprensión. También deberá discutir las distintas soluciones que se presentan en muchos casos, y mostrar ventajas y desventajas.

Habrà una parte de la clase dedicada a la aplicación de los conceptos vistos durante la parte teórica. Se desarrollarán problemas con creciente nivel de dificultad.

### 8.c TRABAJOS PRÁCTICOS

Habrà dos trabajos prácticos durante el cuatrimestre, los cuales tendrán como objetivo resolver un problema algorítmico novedoso, para el cual los alumnos deberán profundizar sobre los conceptos vistos.

Los mismos serán de consigna variante según el año en curso, lo cual ayuda a darle flexibilidad a la cátedra.

## 9. METODOLOGÍA DE EVALUACIÓN

### 9.a NORMAS DE EVALUACIÓN.

- El criterio es que la evaluación del alumno es permanente.
- Se tomarán dos exámenes parciales teórico/prácticos pudiendo acceder a un recuperatorio.
- Las notas de los parciales representan los resultados de la evaluación teórico/práctica.
- Los exámenes parciales y sus recuperatorios serán escritos.

### 9.b RÉGIMEN DE APROBACIÓN DE LA MATERIA.

- Para la aprobación de la materia los alumnos deberán tener los dos parciales aprobados, teniendo la posibilidad de recuperar cada uno de ellos en una única oportunidad adicional, en la fecha acordada con los docentes.
- Además los alumnos deberán aprobar los trabajos prácticos, como condición para la aprobación de la materia.
- Los alumnos que obtengan una nota inferior a cuatro puntos se les asignará la nota insuficiente y deberán recursar la materia.

## 10. PLANIFICACIÓN

CALENDARIO DE CLASES Y EVALUACIONES	
Semana 1	Presentación: docentes y cátedra. Metodologías de dictado y evaluación. Algunos conceptos asociados a la calidad. Repaso de OOP.
Semana 2	Unidad 1

Semana 3	Unidad 2
Semana 4	Unidad 3
Semana 5	Unidad 4
Semana 6	Unidad 5
Semana 7	Parcial 1
Semana 8	Unidad 6
Semana 9	Unidad 7
Semana 10	Unidad 8
Semana 11	Unidad 9
Semana 12	Repaso
Semana 13	Parcial 2
Semana 14	Recuperatorio Parcial 1 - Entrega notas Parcial 2
Semana 15	Recuperatorio Parcial 2 - Entrega notas Recuperatorio Parcial 1
Semana 16	Entrega notas Recuperatorio Parcial 2
Del al de	FINAL

Información de Versiones	
Nombre del Documento:	Ficha Académica de la asignatura Lenguajes de Programación II
Nombre del Archivo	Lenguajes de Programación II
Documento origen:	
Elaborado por:	Sebastián Ismael
Revisado por:	Diego Fontdevila
Aprobado por:	Diego Fontdevila
Fecha de Elaboración:	01-02-2013
Fecha de Revisión:	18-06-2013
Fecha de aprobación	19-06-2013
Versión:	1.0