



# Planificaciones

7524 - Teoría de la Programación

Docente responsable: FERRIGNO LEANDRO SABASTIAN

## OBJETIVOS

El objetivo de este curso es el de introducir el estudio de los lenguajes de programación en su contexto, no sólo desde el punto de vista de la sintaxis y la semántica, sino también desde la pragmática. Las preguntas que se deberán plantear van por lo tanto desde cómo se puede expresar un determinado concepto en diversos lenguajes de programación, hasta para qué propósitos es útil un lenguaje y para qué propósitos no lo es. La presentación de los diversos conceptos se hacen utilizando un lenguaje multiparadigma: Oz. Los alumnos deberán luego analizar diversos lenguajes de programación utilizando la misma metodología.

## CONTENIDOS MÍNIMOS

### PROGRAMA SINTÉTICO

1. Semántica formal de lenguajes de programación. Teoría de tipos. Análisis y transformación de programas. Análisis comparativo de lenguajes de programación. Metaprogramación y módulos genéricos.
2. Modelo de computación declarativo,
3. Concurrencia declarativa, el modelo de concurrencia guiada por los datos, limitaciones y extensión de la programación declarativa.
4. Concurrencia por pasaje de mensajes. Objetos puerto.
5. Estado explícito. Modelo declarativo con estado explícito. Abstracción de datos.
6. Programación orientada a objetos. Clases como abstracciones de datos completas e incrementales. Relación con los otros modelos. Implementación del sistema de objetos.
7. Implementación de concurrencia en un lenguaje orientado a objetos.

### PROGRAMA ANALÍTICO

Unidad 1:

Introducción de conceptos de programación utilizando OZ: variables, funciones, pasajes de parámetros, listas, funciones sobre listas, evaluación perezosa, funciones de alto orden, concurrencia, flujo de datos, estado explícito, objetos, clases.

Unidad 2:

Modelo declarativo: Cómo definir un lenguaje de programación práctico. La memoria de asignación única. El lenguaje núcleo y su semántica. Manejo de memoria. Recolección de basura. Del lenguaje núcleo a un lenguaje práctico. Excepciones.

Unidad 3:

Concurrencia declarativa: el modelo de concurrencia guiada por los datos. Programación con hilos, técnicas básicas. Flujos (streams). Ejecución perezosa. Funciones de alto orden.

Unidad 4:

Concurrencia por pasaje de mensajes. Objetos puerto. Protocolos simples.

Unidad 5:

Estado Explícito. ¿Qué es el estado? Modelo declarativo con estado explícito. Abstracción de datos. Colecciones con estado.

Unidad 6:

Programación orientada a objetos. Herencia. Clases con abstracción de datos incompleta. Clases como una abstracción de datos completa y como una abstracción incremental. Programación con herencia. Relación con los otros modelos. Implementación del sistema de objetos.

Unidad 7:

Concurrencia con estado compartido. Llaves. Monitores. Transacciones.

### BIBLIOGRAFÍA

Libro de Texto:

Peter Van Roy & Seif Haridi. Concepts, Techniques, and Models of Computer Programming. MIT Press (2004).

Sólo para manejo de memoria y recolección de basura:

Richard Jones & Rafael D. Lins. Garbage Collection: Algorithms for Automatic Dynamic Memory Management. Wiley (1996).

Material de apoyo:

Manuales de referencia de todos los lenguajes que se presenten en el curso.

Material adicional (artículos para discutir y comentar en clase):

Meyer, B., "Principles of Language Design and Evolution", Millennial Perspectives in Computer Science (Proceedings of the 1999 Oxford-Microsoft Symposium in Honour of Sir Tony Hoare), 2000, pp. 229-246.

Meyer, B., "Using Inheritance Well", Object Oriented Software Construction (2nd. ed.), Prentice Hall, 1997, pp. 809-869.

## **RÉGIMEN DE CURSADA**

### **Metodología de enseñanza**

Clases teórico-prácticas (primeras 8 semanas).

Exposición teórica de conceptos fundamentales, presentación de problemas y discusión de su resolución

Resolución por parte de los alumnos y controlada por los docentes auxiliares de problemas correspondientes a las unidades temáticas del programa.

Clases de exposición

Los alumnos exponen en grupos (o individualmente, si hay pocos inscriptos) los lenguajes analizados (últimas 8 semanas), y presentan un informe sobre el tema.

### **Modalidad de Evaluación Parcial**

La evaluación de cada alumno se realiza de la siguiente manera:

a) Evaluación de la exposición y del informe entregado (reemplaza a la evaluación parcial tradicional). Se requiere para firmar la regularidad en la materia.

## CALENDARIO DE CLASES

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<1> 09/03 al 14/03	Introducción a la materias. Presentación general de temas y conceptos a ver durante todo el cuatrimestre.  Primeros pasos en Oz.					
<2> 16/03 al 21/03	Modelo declarativo. Definición de lenguajes de programación: sintaxis y semántica.  Lenguaje Kernel y ejecución en máquina abstracta.  Sistemas de tipos, scoping, abstracción procedural, closures, entre otros temas básicos.					
<3> 23/03 al 28/03	Manejo de memoria, introducción a Garbage Collection, manejo de excepciones, pattern matching.  Practicas de programación funcional Programación de alto orden (high order programming)					
<4> 30/03 al 04/04	Concurrencia, técnicas de programación con concurrencia. Streams Lazy, demand driven execution.					
<5> 06/04 al 11/04	Comunicación entre procesos concurrentes. Puertos, Agentes, Actores. RMI, RPC					
<6> 13/04 al 18/04	Construcción de diferentes tipos de TDA					
<7> 20/04 al 25/04	OOP					
<8> 27/04 al 02/05	OOP + repaso general					
<9> 04/05 al 09/05	Clases de exposición de alumnos					
<10> 11/05 al 16/05	Clases de exposición de alumnos					
<11> 18/05 al 23/05	Clases de exposición de alumnos					
<12> 25/05 al 30/05	Clases de exposición de alumnos					
<13> 01/06 al 06/06	Clases de exposición de alumnos					
<14> 08/06 al 13/06	Clases de exposición de alumnos					
<15> 15/06 al 20/06	Clases de exposición de alumnos					
<16> 22/06 al 27/06	Clases de exposición de alumnos					

## CALENDARIO DE EVALUACIONES

### Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º				
2º				
3º				
4º				