



# Planificaciones

7540 - Algoritmos y Programación I

Docente responsable: AZCURRA DIEGO ANDRES

## OBJETIVOS

Aprender a diseñar e implementar programas sencillos. Introducir la noción de recursos (tiempo y espacio) consumidos por los programas. Programación iterativa y recursiva. Uso de objetos. Estructuras de datos simples.

## CONTENIDOS MÍNIMOS

-

## PROGRAMA SINTÉTICO

Componentes de un programa. Estructuras de control. Nociones de tiempo y espacio asociados a un programa. Algoritmos de búsqueda. Tratamiento de archivos y excepciones. Nociones introductorias de POO. Estructuras de datos dinámicas. Recursividad. Algoritmos de ordenamiento.

## PROGRAMA ANALÍTICO

Unidad 1. Variables y programas. Estado y computación. Componentes de programas sencillos. Tipos simples: números, cadenas, booleanos. Funciones y procedimientos.

Unidad 2. Alternativas. Ciclos definidos e indefinidos. Tratamiento de secuencias. Pre y postcondiciones. Nociones de tiempo y espacio consumido.

Unidad 3. Búsquedas y algoritmos relacionados. Comparación de algoritmos.

Unidad 4. Entrada/Salida. Excepciones. Archivos. Apareos.

Unidad 5. Nociones introductorias de programación orientada a objetos. Uso de objetos dados. Construcción de objetos sencillos.

Unidad 6. Estructuras de datos: pilas, colas, listas

Unidad 7. Recursividad y división y conquista.

Unidad 8. Métodos de ordenamiento: Métodos sencillos. Métodos recursivos. Comparación de algoritmos.

## BIBLIOGRAFÍA

- Apunte del curso: <http://www.algoritmos7540-rw.tk/material>

- John Zelle: Python Programming: An Introduction to Computer Science, Franklin, Beedle & Associates, 2003.

- Allen B. Downey: Think Python: An Introduction to Software Design", 2008.  
<http://www.greenteapress.com/thinkpython/>

- Mark Lutz: Learning Python (3rd Edition), OReilly Media Inc., 2007.

- Simon Thompson: How to program it, 1996. Recuperado el 19 de agosto de 2008 de <http://www.ling.gu.se/lager/kurser/Programming/HowToProgIt.html>

- Josh Cogliati: Non-Programmer's Tutorial for Python, Wikibooks.  
Recuperado el 29 de agosto de 2008 de [https://en.wikibooks.org/wiki/Non-Programmer%27s\\_Tutorial\\_for\\_Python\\_2.6](https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_2.6)

## RÉGIMEN DE CURSADA

### Metodología de enseñanza

Se darán dos clases semanales teóricas y dos clases semanales prácticas. Al menos una de las clases semanales prácticas se da en laboratorio (según disponibilidad de laboratorios). Hay ejercitación en laboratorio desde la primera semana de clase. Los alumnos arman grupos que son supervisados por un docente auxiliar.

### Modalidad de Evaluación Parcial

Se toman cuatro pequeños parciales distribuidos a lo largo de todo el curso. Todos ellos cuentan con dos recuperaciones. Para regularizar la cursada se debe aprobar todos los trabajos especiales de laboratorio y los cuatro parciales.

## 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 materia Construcción de programas. Análisis, especificación y diseño. Debugging. Ejemplos de programas sencillos.					
<2> 16/03 al 21/03	Procedimientos y funciones. Abstracción. Definición e invocación en programas. Parámetros, valores de retorno y scope de variables.	Ejercicio EJ1 "área de un triángulo"				
<3> 23/03 al 28/03	Expresiones booleanas. Operadores lógicos. Estructuras condicionales. Ciclos. Ciclos indefinidos. Uso de centinelas.					
<4> 30/03 al 04/04	Secuencias: Cadenas, tuplas y listas. Secuencias. Listas y mutabilidad.	Ejercicio EJ2 "pedir entero"				
<5> 06/04 al 11/04	Buscar en listas. Pasaje de parámetros mutables a funciones. Algoritmos de búsqueda (lineal, binaria). Pasaje de funciones por parámetro. Map y filter.		Presentación TP1			
<6> 13/04 al 18/04	Diccionarios. Contratos: pre y post condiciones. Invariantes.			Primer parcialito: Estructuras de control, Cadenas, Listas, Tuplas		
<7> 20/04 al 25/04	Manejo de archivos Excepciones. Manejo de errores.		Entrega TP1			
<8> 27/04 al 02/05	Recorridos de archivos: apareo y corte de control.		Presentación TP2			

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
	Objetos. Propiedades. Creación de objetos. Abstracción.					
<9> 04/05 al 09/05	Listas enlazadas.			Segundo parcialito: Dicionarios, Archivos, Objetos, Excepciones		
<10> 11/05 al 16/05	Pilas y colas		Entrega TP2			
<11> 18/05 al 23/05	Pila de ejecución. Alcance de variables. Recursión. Métodos de ordenamiento sencillos.		Presentación TP3			
<12> 25/05 al 30/05	Pila de ejecución. Alcance de variables. Recursión. División y conquista. Ordenamientos recursivos.			Tercer parcialito: Listas Enlazadas, Pilas, Colas		
<13> 01/06 al 06/06	Lenguaje de programación C.					
<14> 08/06 al 13/06	Lenguaje de programación C. Cierre de la materia. Encuestas.		Entrega TP3			
<15> 15/06 al 20/06				Cuarto parcialito: Recursividad, Ordenamiento, Lenguaje C		
<16> 22/06 al 27/06				1er Recuperatorio. Parcialitos 1 y 2 1er Recuperatorio. Parcialitos 3 y 4 2do Recuperatorio. Parcialitos 1, 2, 3 y 4		

## CALENDARIO DE EVALUACIONES

### Evaluación Parcial

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