



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

---

## Taller de Lenguajes I

Carrera: Ingeniería en Computación  
Profesor Responsable: Lanzarini, Laura  
Año: 2º  
Duración: Semestral  
Carga Horaria Semanal: 6hs  
Carga Horaria Total: 96hs

---

### Objetivos Generales

Profundizar los conocimientos obtenidos por el alumno en los primeros cursos vinculados con Algoritmos y Programación, permitiéndole desarrollar un estudio teórico-práctico de algún lenguaje de programación procedural (el lenguaje puede variar con los cambios tecnológicos), poniendo énfasis en el análisis formal de las características del lenguaje y su comparación con los que el alumno conociera a ese momento (típicamente Pascal).

### Contenidos Mínimos

Estudio de un lenguaje de programación (como por ejemplo C, que resulta adecuado para la carrera) en el que se desarrollen aplicaciones concretas.

## PROGRAMA

### 1. Introducción

La historia de C. ANSI C. Características de C. La estructura de un programa en C. Identificadores. Alcance de un identificador. Cómo trabaja el compilador C. Análisis de un par de programas sencillos en C. Comentarios. Caracteres especiales. La biblioteca estándar de C. Sentencias printf y scanf.

### 2. Tipos de Datos

Tipos de datos simples. Tipos enteros con una cantidad exacta de bits. Tipos de punto flotante. Conversión de tipos. Conversión de tipos aritméticos. Conversión aritmética usual. Otras conversiones implícitas. Conversión explícita de tipos.

### 3. Expresiones, Operadores y Sentencias

Evaluación de expresiones. Operadores de asignación. Operadores incrementales y decrementales.

Sentencia. Bloque de sentencias. Iteraciones. Sentencias while, for y do-while. Loops anidados. Sentencias de selección. Sentencias *if*, *if-else*, operador condicional y *switch*. Saltos incondicionales. Sentencias *break*, *continue*, *return*.

### 4. Funciones e identificadores



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

Definición, declaración e invocación de funciones. Prototipo de una función. Coerción de tipos. Retorno de una función. El tipo *void*; su uso en declaración de funciones. Funciones recursivas.

Clases de almacenamiento de un identificador. Persistencia automática: *auto* y *register*. Persistencia estática: *static* y *extern*.

### **5. Arreglos**

Arreglos de longitud fija y arreglos de longitud variable. Definición. Acceso a los elementos del arreglo. Inicialización del arreglo. Vector de caracteres. Funciones para cadenas de caracteres. Arreglos y el especificador *static*. Arreglos multidimensionales. Arreglos como argumento de una función.

### **6. Punteros**

Declaración de punteros. Inicialización de punteros. Identificador *null*. Operadores de punteros. Visualización del valor de un puntero. Pasaje de parámetros por referencia. El calificador *const* aplicado a arreglos y a punteros.

Operaciones aritméticas con punteros. Asignación de punteros. Comparación entre punteros.

Arreglos de punteros y punteros a arreglos. Punteros a funciones.

### **7. Cadenas de caracteres**

Definición. Declaración e inicialización. Funciones de manejo de caracteres (librería *ctype.h*). Funciones de conversión (librería *stdlib.h*). Funciones de entrada/salida (*stdio.h*). Funciones de comparación, búsqueda y manipulación de cadenas (*string.h*).

### **8. Estructuras, uniones y enumeraciones**

Estructuras. Definición. Declaración de variables. Inicialización. Operaciones. Acceso a los campos de una estructura. Punteros como miembros de una estructura.

Union. Definición de tipos Union. Operaciones sobre uniones.

Operadores a nivel de bits. Operadores de asignación a nivel de bits. Campos de bits.

Constantes de enumeración. Ejemplos.

### **9. Archivos**

Streams. Dispositivos estándar: *stdin*, *stdout*, *stderr*. Archivos. Definición. Modos de apertura de un archivo. Posición dentro del archivo. Operaciones de alta, baja y modificación de archivos. Acceso secuencial y acceso directo. Manejo de errores de acceso. Archivos binarios.

### **10. Manejo de memoria dinámica**

Alocación de memoria de manera dinámica. Redimensionado y liberación de memoria. Arreglos dinámicos. Ejemplificación creando listas. Matrices dinámicas. Alocación y liberación de memoria.



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

### 11. Directivas para preprocesador

Directiva `#include`. Acceso a los archivos indicados en la directiva `#include`.

Definición y uso de macros. Macros con y sin parámetros. Macros dentro de macros. Alcance y redefinición de macros. Macros predefinidas.

Directiva `#undef`. Compilación condicional. Las directivas `#if` y `#elif`. Directivas `#ifdef` e `#ifndef`.

### 12. Librerías

Headers estándar. Uso de headers. Contenido de los headers estándar. Funciones de la librería estándar según su área de aplicación. Aplicaciones formadas por varios archivos.

### 13. Herramientas básicas

Compilación con GCC. Warnings del compilador. Optimización Debugging.

Usando *make* para construir programas en C. Prerrequisitos y comandos. *MakeFile*. Reglas. Comenatrios. Variables. Macros. Funciones. Directivas. Ejecutando *make*.

Debugging de programas C utilizando GDB. Un ejemplo simple de una sesión de debugging.

## BIBLIOGRAFIA

**Título:** El lenguaje de Programación C

**Autores:** Kernighan, Ritchie

**Editorial:** Prentice Hall

**Año de Edición:** 1991

ISBN: 0-13-110362- 8

**Título:** C for Programmers with an Introduction to C11

**Autores:** Paul Deitel, Harvey Deitel

**Editorial:** Prentice Hall

**Año de Edición:** 2013

ISBN: 0133462072, 9780133462074

**Título:** Como Programar en C/C++

**Autores:** Paul Deitel, Harvey Deitel

**Editorial:** Prentice Hall

**Año de Edición:** 2004

ISBN-10: 9702605318

**Título:** Como Programar en C

**Autores:** Antón Fernando



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

**Año de Edición:** 1997  
**ISBN:** 8489245223

### **Descripción de las Actividades Teóricas y Prácticas**

La asignatura tiene como objetivo central capacitar al alumno en la resolución de problemas concretos utilizando el lenguaje de programación C. Por tal motivo, el dictado del curso se realiza completamente en una de las Salas de PC de la Facultad de Informática donde se imparten tanto las clases teóricas como las clases prácticas. El curso posee clases teóricas a cargo del profesor de la asignatura y clases prácticas a cargo de los auxiliares. La carga horaria es de 6 hs semanales repartidas en una clase teórica de 2 hs y dos clases prácticas de 2 hs. cada una.

La interacción entre alumnos, docentes y contenidos se lleva a cabo a través del entorno virtual de enseñanza y aprendizaje WebUNLP. Las teorías son guiadas por transparencias proyectadas con un cañón. Las prácticas se desarrollan en la Sala de PC.

Al inicio del curso se publica el cronograma con las actividades a realizar en cada clase teórica y práctica.

### **Metodología de Enseñanza y de Evaluación**

#### *METODOLOGÍA DE ENSEÑANZA*

Considerando el rol de la asignatura dentro del plan de estudios se reconoce que el saber no se limita a la información y que lo importante es saber – hacer. Por tal motivo, la orientación que se le da a la materia es predominantemente práctica.

Sin embargo, no se deja de prestar atención a conceptos teóricos muy importantes dentro de la disciplina, como el desarrollo modular de programas y el encapsulamiento.

Durante la teoría, se resuelven ejemplos concretos específicos del tema que se está desarrollando. Esto tiene como fin afianzar la explicación teórica permitiendo que el alumno ponga en práctica lo aprendido. Las actividades prácticas se realizan principalmente en grupos de 2 alumnos para fomentar la discusión y el intercambio de ideas.

Antes de finalizar cada teoría, el profesor muestra una solución del ejercicio propuesto sobre la que se analizan soluciones alternativas basadas en lo realizado por los alumnos previamente.

Durante los horarios de práctica, los alumnos resuelven ejercicios propuestos por la cátedra totalmente relacionados con los conceptos vistos en la teoría inmediatamente anterior.

#### *METODOLOGÍA DE EVALUACIÓN*



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

El curso está dividido en dos módulos. Al finalizar cada uno de ellos se toma una evaluación teórico-práctica con un recuperatorio. Al terminar el curso se toma una fecha adicional que podrá ser utilizada para recuperar sólo uno de los módulos.

Los alumnos pueden aprobar la asignatura por promoción o con evaluación final:

**a) Régimen de promoción**

Los alumnos que deseen aprobar la materia por promoción deberán estar inscriptos en la materia bajo esa modalidad y obtener en ambos módulos una calificación individual mayor o igual a 5 puntos y un promedio general mayor o igual a 6 puntos. La nota final del curso es el promedio de las calificaciones obtenidas en los dos módulos.

Quienes aprueben la promoción y se encuentren inscriptos en el curso bajo esta modalidad, tendrán registrada su nota al final del curso.

**b) Con evaluación final**

Los alumnos que aprueben ambos módulos con calificación mayor o igual a 4 (cuatro) tendrán aprobada la cursada de la asignatura.

Quienes aprueben sólo la cursada deberán inscribirse en una mesa de examen final donde a través de un coloquio se definirá su calificación final.