

# I/O Files

Pablo R. Ramis

Universidad Nacional de Rosario, Instituto Politécnico, Dto. de Informática,  
prramis@ips.edu.ar,  
WWW home page: <http://informatica.ips.edu.ar>

**Resumen** Verán una serie de ejercicios que van aumentando progresivamente en la dificultad.

Si bien la cantidad es acotada, cada uno puede ser implementado de varias maneras, archivos texto, binarios, escritura de strings, o estructuras. Las variantes las pueden ir generando ustedes en la medida que lo deseen.

## 1. Ejercicios

### 1.1. Hola Mundo

Escribe un programa en C que cree un archivo de texto llamado saludo.txt y escriba el texto "Hola, mundo!".<sup>en</sup> él. Si el archivo ya existe, que la línea se agregue.

### 1.2. Leer desde un archivo

Escribe un programa que lea el contenido de un archivo llamado datos.txt y lo muestre en pantalla. Si el archivo no existe, debe mostrar un mensaje de error.

### 1.3. Contar líneas

Crea un programa que cuente el número de líneas en un archivo de texto llamado texto.txt y muestre el resultado en pantalla.

### 1.4. Añadir texto

Escribe un programa que abra el archivo notas.txt y le añada (sin sobrescribir) una nueva línea con el texto "Nueva nota: 100".

### 1.5. Copiar texto

Escribe un programa que copie el contenido de un archivo origen.txt a otro archivo copia.txt.

### **1.6. Contar palabras**

Crea un programa que lea el contenido de un archivo `texto.txt` (puede ser el archivo anterior) y cuente la cantidad total de palabras en el archivo. Una palabra se define como una secuencia de caracteres alfanuméricos separados por espacios, saltos de línea o signos de puntuación.

### **1.7. Ordenar nombres y escribir en un archivo**

Crea un programa que lea una lista de nombres desde un archivo `nombres.txt`, los ordene alfabéticamente y luego los escriba en un archivo nuevo llamado `nombres_ordenados.txt`.

Usa un array de strings para almacenar los nombres.

Implementa un algoritmo de ordenamiento como `qsort` o el método de la burbuja para ordenar los nombres.

### **1.8. Buscar y reemplazar palabras**

Crea un programa que abra un archivo `documento.txt` y reemplace todas las ocurrencias de una palabra específica (por ejemplo, "vieja") por otra palabra (por ejemplo, "nueva"). El resultado debe guardarse en un archivo nuevo llamado `documento_modificado.txt`.

### **1.9. Almacenar y leer estructuras en un archivo binario**

Crea un programa que guarde una lista de estudiantes en un archivo binario `estudiantes.dat`. Cada estudiante tiene un nombre, una edad y un promedio de calificaciones. Luego, el programa debe leer y mostrar todos los estudiantes guardados.

### **1.10. Extraer y calcular datos de un archivo CSV**

Crea un programa que lea un archivo `ventas.csv` donde cada línea contiene el nombre de un producto y su precio de venta. El programa debe calcular el total de todas las ventas e imprimirlo.