

# Guía de Trabajos Prácticos

Esta guía contiene los Trabajos Prácticos (TP) obligatorios del curso R1042, incluyendo una guía para su entrega con el formato de presentación y un cronograma con las fechas de entrega.

Durante el transcurso del año se deberán entregar Trabajos Prácticos, que se caracterizan por ser incrementales. Esto significa que en cada TP se irán introduciendo nuevos conocimientos con respecto al TP anterior. A su vez, se deberán utilizar las herramientas SVN, Makefile y Doxygen cuando se lo indique.

## Formato de presentación

- Los archivos fuentes deben tener en todos los casos los comentarios necesarios para clarificar su lectura.
- Cada subrutina/función, debe contar con un encabezado describiendo la operación que realiza, los parámetros que espera como entrada, y los resultados que debe presentar, indicando formato y método de entrega.
- Como encabezado del programa, debe haber un comentario que explique claramente que hace dicho programa, y las instrucciones detalladas (comandos) para su compilación y “linkeo”.
- TODOS los ejercicios son obligatorios.
- **La entrega de TODOS los trabajos prácticos es obligatoria para regularizar la materia.**

# **Trabajo Práctico N°4**

## **- Funciones en C -**

### **Ejercicio 1**

Realizar una función que calcule el factorial de un número entero pasado como argumento. Se deben verificar todas las condiciones correspondientes para que el resultado sea correcto. Utilice el siguiente prototipo de función:

long factorial(int numero)

### **Ejercicio 2**

Realice una función que reciba como argumento un número entero y determine si el mismo es PRIMO. La función debe retornar:

- 1 si es primo.
- 0 si no es primo.

Utilice el siguiente prototipo:

int VerificarPrimo(int numero)

### **Ejercicio 3**

Escriba una función que genere 10 números pseudo aleatorios entre 5 y 20.

*Ayuda: man random, srand y time*

Utilizar los datos entregados por la función time como semilla para las funciones de pseudoaleatoriedad. Haga un comentario en el código explicando por qué se utiliza una semilla y que pasa si esta es constante.

### **Ejercicio 4**

Implemente una función que calcule  $x^y$  sin utilizar la función pow. El prototipo de la función es:

float myPow (float x, int y);