



Universidad de Costa Rica  
Facultad de Ingeniería  
Escuela de Ingeniería Eléctrica  
**IE-0117 Programación Bajo Plataformas Abiertas**



Ing. Marco Villalta Fallas - I Ciclo 2018

---

## Laboratorio # 6

Arreglos, compiladores, herramientas y automatización de proyectos

---

### Instrucciones Generales:

Los laboratorios se deben de realizar de manera individual.

El laboratorio debe de entregarse antes del 14 de mayo a las 23:59.

Utilice capturas de pantalla para demostrar funcionalidad (en caso de ser necesario).

Entregue un archivo comprimido que incluya un directorio llamado **informe** con los archivos necesarios para generar el PDF del informe (.tex, imágenes, código, entre otros) y un directorio llamado **src** con los archivos de código fuente que sean la solución a un problema determinado.

En la calificación de este laboratorio se tomará en cuenta el uso del control de versiones git y la documentación en Doxygen

## 1. Ejercicios con arreglos en C

Nota: haga un archivo diferente para cada ejercicio.

1. **Segundo elemento mayor:** Un programa que encuentre el segundo número más grande del arreglo
2. **Imprimir elementos únicos:** Un programa que recorra un arreglo, e imprima solo los elementos únicos del arreglo.
3. **Búsqueda de un elemento:** Un programa que recorra un arreglo y retorne todos los índices del arreglo en los que se coincide con el criterio de búsqueda.

## 2. Compiladores

Conteste las siguientes preguntas de forma clara y concisa.

1. En su forma más general, en una sola palabra, ¿qué es un compilador?
2. Por medio de un diagrama, represente la principal diferencia entre un interprete y un compilador.
3. Explique brevemente con una o más razones porque un lenguaje compilado puede ser más eficiente que uno interpretado.
4. ¿A que se le conoce como cross-compiler?
5. Enumere los 5 elementos que componen la estructura típica de un compilador.
6. Describa brevemente la función que realiza, en el proceso de compilación, cada uno de los elementos mencionados en el punto anterior.

### 3. Herramientas adicionales para programación en C

#### Depuración: gdb

1. ¿Cuál es la bandera del compilador gcc que se utiliza para generar un código que tenga los componentes necesarios para depurar (*debugging*)?
2. ¿Cuál es la línea de comandos en terminal necesaria para compilar con gcc un código fuente con el nombre `fuentes.c` en un ejecutable llamado `bin` capaz de ser depurado con gdb?
3. Describa brevemente para que sirven los siguientes comandos en gdb:

- |                      |                           |
|----------------------|---------------------------|
| ■ <code>run</code>   | ■ <code>continue</code>   |
| ■ <code>break</code> | ■ <code>info break</code> |
| ■ <code>quit</code>  | ■ <code>print</code>      |

4. Realice la depuración del programa `factorial.c` proporcionado por el profesor a través de la herramienta gdb.

**Nota:** Si el programa funcionara correctamente, debería de imprimir en consola el factorial del número proporcionado como argumento. Muestre a través de capturas de pantalla el proceso realizado utilizando **gdb** para depurarlo.

### 4. Automatización de proyectos: make

1. Realice una estructura de directorios y archivos como la que se muestra a continuación:

```
.
|-- build
|-- include
|   |-- functions.h
|-- src
|   |-- functions.c
|   |-- main.c
3 directories, 4 files
```

2. Descomponga la función `factorial.c` en tres archivos. Uno debe de ser el `main.c` encargado de llamar la función `long long factorial (int N)`, la cual se encuentra definida en `functions.h` e implementada en el archivo `functions.c`, para calcular el factorial de un número ingresado como argumento en línea de comandos.
3. Automatice el proceso de compilación realizando un Makefile para el ejercicio. Los archivos objeto (ejecutables) deben de quedar en el directorio `build`, para no pollutar los directorios de código fuente.
4. Comprima por aparte los ejercicios realizados de Makefile e incluyalos juntos con los archivos `.tex` utilizados para generar su informe final del laboratorio