



## Lab 0: “Hola, Mundo!” en el PC

Fecha de presentación: 11 de Marzo

### 1. Objetivo

Este laboratorio corresponde a una experiencia introductoria a los conceptos básicos de la programación en C y a los procesos y elementos involucrados en la compilación y ejecución del programa desarrollado. Además, busca generar una familiarización con el uso de la plataforma GitHub, que se utilizará a lo largo del semestre.

### 2. Descripción de la actividad

Este laboratorio introductorio consta de pequeñas actividades que les ayudarán a preparar su computador para utilizarlo de manera eficiente a lo largo del semestre.

#### 2.1. Descargar e instalar compilador

Dependiendo de su sistema operativo, la instalación del compilador cambiará. Existen diversas herramientas para realizar esto. Es de su responsabilidad escoger la que sea más cómoda y trabajar con ella. Algunas de las lecturas que recomendamos para este proceso son las siguientes:

**Windows:** Recomendamos el uso de TDM-GCC. Algunos links útiles para esto son los siguientes:

- [How to Run C-Program in Command Prompt](#)
- [Tutorial Installing TDM-GCC \(Windows\)](#)

**macOS:** Dado que macOS es un sistema operativo basado en UNIX, el compilador GCC viene por defecto en el terminal por lo cual no requiere software adicional.

**Linux:** Abrir el terminal y ejecutar: `sudo apt install gcc`.



## 2.2. Escribir programa en C

Para escribir su código en C pueden utilizar el editor de texto que sea de su gusto. **Atom** y **Sublime Text** son buenas herramientas para esto. Deben copiar y pegar el siguiente código y guardarlo con un nombre a elección con extensión `.c`. Por ejemplo, `hola_mundo.c`

```
1 | #include <stdio.h>
2 |
3 | int main()
4 | {
5 |     /* Imprime un mensaje de texto. */
6 |     printf("Hola, Mundo!\n")
7 |
8 |     return 0;
9 | }
```

## 2.3. Compilar el programa en C

Una vez realizado el código, es necesario compilarlo y ejecutarlo por terminal utilizando los siguientes comandos:

```
1 | gcc -Wall -Wextra -Wpedantic -o myProgram hola_mundo.c
2 | ./myProgram
```

## 2.4. Subir código al repositorio de GitHub

Ya leída la guía sobre instalación y uso de la plataforma GitHub (se encuentra en el repositorio principal del curso), se procede a subir el código al repositorio asociado a su usuario registrado. A continuación se darán algunas sugerencias e instrucciones sobre el proceso de carga de códigos.

Para subir el código se debe tener la carpeta del repositorio clonada en el computador. En caso contrario la generaremos por primera vez mediante el comando:

```
git clone url_del_repositorio.git
```

Cuando se realicen cambios importantes se sugiere fuertemente enviar un **push** a su repositorio con su respectivo **commit** lo más descriptivo posible sobre la modificación principal del código, de modo que no olvide cuál fue el cambio realizado.

También puede agregar una hoja de texto en formato Markdown (`.md`) para especificar el funcionamiento de su código u otras características que considere relevantes.



### 3. Lectura recomendada

**Tutorial de C:** material de utilidad para iniciarse en el mundo de la programación en C.

**GCC:** información y explicaciones sobre las opciones de la línea de comando de GCC, el compilador recomendado para utilizar en este laboratorio.

### 4. Pauta de Evaluación

#### 4.1. Consideraciones generales

- El laboratorio será evaluado exclusivamente con nota 1.0 (**R**eprobado), 5.5 (**A**probado) y 7.0 (**D**istinguido). En ningún caso habrán notas intermedias.
- No se reciben trabajos después del módulo de presentación. Trabajos no entregados son calificados con nota 1.0 y son considerados dentro del criterio de aprobación del curso.
- Cualquier consulta sobre los criterios de evaluación de cada laboratorio debe ser realizada en las **issues**, donde estará disponible para que sea revisada por todos los alumnos.

#### 4.2. Criterios de aprobación

Se requiere cumplir con todos los puntos mencionados a continuación para poder aprobar. No existen casos excepcionales.

1. Funcionamiento de los requerimientos. El alumno realiza una presentación de su trabajo y se responsabiliza de exponer que su trabajo satisfaga todos los requerimientos mínimos solicitados en la *descripción de la actividad*, los cuales incluyen en este laboratorio:
  - Compilador instalado
  - Programa compilado
  - Programa ejecutándose
  - Código subido a repositorio privado de GitHub
2. Preguntas. Se responde satisfactoriamente a 2 de 3 preguntas aleatorias al momento de la presentación final, las cuales abarcan los siguientes temas:



- Qué es un compilador y cómo funciona en términos generales.
- Qué es GCC, cómo se ejecuta y qué representa cada una de las opciones.
- Qué representa cada línea de código y en qué se traducen en el funcionamiento del programa.
- Qué es GitHub y cuál es su utilidad.

Solo se dispone de una oportunidad para responder estas preguntas. Fallar en este requisito se traduce en la reprobación inmediata de la experiencia de forma inapelable.

### 4.3. Criterios de Distinción

La distinción representa un trabajo adicional que sobresale a los requerimientos mínimos para la aprobación. Agregados adicionales no constituyen por sí mismo una distinción si no representan un verdadero trabajo adicional de comprensión y/o análisis. Estos son analizados en función de lo que presenten sus compañeros. Los distinguidos pueden caer (no exclusivamente) en algunas de las siguientes líneas generales:

- Funcionalidades adicionales sobresalientes.
- Uso aplicado y verificable de funcionalidades adicionales del lenguaje C.
- Creatividad. :D
- Imprimir *Hola, Mundo!* en múltiples idiomas no distingue. Uso excesivo de loops for y while, tampoco.

Las distinciones son discutidas caso a caso por la totalidad del equipo de ayudantes al finalizar la corrección del laboratorio. Serán notificadas públicamente después del módulo de evaluación.