



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Marco Antonio Martínez Quintana

*Asignatura:* Fundamentos de Programación

*Grupo:* 3

*No de Práctica(s):* 6

*Integrante(s):* Ramírez Pérez Daniela Itzel

*No. de Equipo de cómputo*

*empleado:* No aplica

*No. de Lista o Brigada:* 36

*Semestre:* 1er

*Fecha de entrega:* Viernes 13 de noviembre

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Objetivo:

Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

## Introducción

### Lenguaje de Programación

Serie de instrucciones que pueden ser realizadas por una computadora.

### Tipos de Lenguaje

**Alto Nivel** Cuando las instrucciones pueden ser entendidas fácilmente por un humano.

**Mediano Nivel** Cuando no es totalmente transparente el lenguaje y tiene elementos que tienen que ver con la arquitectura de la máquina a la hora de programar.

**Bajo Nivel** Cuando las instrucciones son cercanas a lo que la máquina puede entender y difícilmente pueden ser comprendidas por una persona que no tenga conocimientos de la máquina en que operarán

## Lenguaje C

Este lenguaje es muy utilizado ya que la forma de dar instrucciones es muy cercana a lo que un humano podría abstraer, por esta razón, C es conocido como un lenguaje de alto nivel, aunque algunos autores lo consideran también como un lenguaje de mediano nivel.

Un aspecto del lenguaje C es combinar características de un lenguaje de alto nivel con uno de bajo nivel, creándose variantes que permiten programar miles de dispositivos electrónicos en el mundo con sus respectivos compiladores.

### Como funciona

Se describe cada una de las instrucciones de acuerdo con las reglas definidas en este lenguaje en un archivo de texto para después ser procesadas en un compilador, que toma como entrada un archivo de texto y tiene como salida un programa ejecutable, se tiene como ventaja que un programa escrito en lenguaje C, siguiendo siempre su estándar, puede correr en cualquier máquina siempre y cuando exista un compilador de C hecho para tal.

Para realizar un programa usando el lenguaje C, es necesario pensar primero en el sistema operativo que corre sobre la máquina y posteriormente, si este sistema cuenta con interfaz gráfica o sólo posee línea de comandos.

Una vez que se han seleccionado estos elementos, se necesita buscar qué opciones de editores y compiladores están disponibles.

## Editores de C

Un programa en C debe ser escrito en un editor de texto para después generar un programa ejecutable en la computadora por medio de un compilador. Tanto el editor de texto como el compilador van de la mano con el sistema operativo y si posee o no interfaz gráfica por lo que son factores que se deben de tomar en cuenta a la hora de elegir el entorno para desarrollar programas en C.

Es importante señalar que no es lo mismo un editor de texto que un procesador de texto. El primero edita un texto plano que puede tener muchas utilidades como guardar una configuración, tener escrito un programa, etc., y será interpretado hasta que se haga una lectura de éste. Un procesador de texto permite dar formato al texto, a la hoja donde está escrito, incrustar imágenes, etc., su salida puede ser un archivo de texto plano que contiene etiquetas que señalan el formato que se le dio al texto o algo un poco más complejo. A continuación, se presentan algunos de los editores más comunes.

## Editor Visual Interface de GNU/Linux (VI)

El editor vi (visual interface) es el editor más común en cualquier distribución de sistemas operativos con núcleo basado en UNIX. Está disponible en línea de comandos y si el sistema operativo tiene entorno gráfico se puede acceder a él desde la terminal.

Para iniciar VI, debe taclearse desde la línea de comandos:

`vi nombre_archivo[.ext]`

Donde “`nombre_archivo`” puede ser el nombre del archivo a editar o el nombre de un archivo nuevo que se creará con VI. Es válido incluir la ruta donde se localiza o localizará el archivo.

## Modo comando

Es el modo por defecto de VI cuando se abre. Las teclas presionadas ejecutan diversas acciones predeterminadas y no se puede editar el texto libremente. Los comandos son sensitivos a las mayúsculas y a las minúsculas. Algunos ejemplos son:

<code>↑ o k</code>	mueve el cursor hacia arriba.
<code>↓ o j</code>	mueve el cursor hacia abajo.
<code>← o h</code>	mueve el cursor hacia la izquierda.
<code>→ o l</code>	mueve el cursor hacia la derecha.
<code>1G</code>	lleva el cursor al comienzo de la primera línea.

G	lleva el cursor al comienzo de la última línea.
x	borra el carácter marcado por el cursor.
dd	borra o corta la línea donde está el cursor.
ndd	donde n es la cantidad de líneas que se borrarán o cortarán después del cursor.
D	borra o corta desde la posición de cursor hasta el final de la línea.
dw	borra o corta desde la posición del cursor hasta el final de una palabra.
yy	copia la línea donde está el cursor.
p	pega un contenido copiado o borrado.
u	deshace el último cambio

### Modo de última línea

Se puede acceder a él desde el modo de última línea. Es muy similar al modo comando, pero los comandos no tendrán efecto hasta que se presiona la tecla Enter además de que se visualizará el comando en la última línea del editor. Es posible cancelar el comando con la tecla Esc. Los comandos de última línea se caracterizan porque inician con /, ? o :. Algunos ejemplos son:

/texto	donde la cadena texto será buscada hacia delante de donde se encuentra el cursor.
?texto	donde la cadena texto será buscada hacia atrás de donde se encuentra el cursor.
:q	para salir de VI sin haber editado el texto desde la última vez que se guardó.
:q!	para salir de VI sin guardar los cambios.
:w	para guardar los cambios sin salir de VI.
:w	archivo para realizar la orden “guardar como”, siendo archivo el nombre donde se guardará el documento.
:wq	guarda los cambios y sale de VI.

### Modo insertar

Este modo permite insertar texto. Las teclas presionadas ya no harán una acción como en el modo comando sino será el contenido que formará el texto del documento. Se puede desplazar con las flechas del teclado y borrar con la tecla de retroceso o de suprimir.

Para ingresar al modo insertar existen varios comandos:

i	pasa al modo insertar poniendo el texto a la izquierda del cursor.
a	pasa al modo insertar poniendo el texto a la derecha del cursor.
A	pasa al modo insertar colocando el texto al final de la línea donde el cursor se encuentra.
I	pasa al modo insertar colocando el texto al principio de la línea donde el cursor se encuentra.

O	coloca una línea arriba de la línea seleccionada por el cursor y pasa al modo insertar.
o	coloca una línea debajo de la seleccionada por el cursor y pasa al modo insertar.

Para salir del modo insertar debe presionarse la tecla Esc. Para verificar que se encuentra en modo insertar es se puede ver -- insertar -- en la última línea del editor.

<b>GNU NANO</b>	<p>Disponible para sistemas operativos basados en UNIX en línea de comandos. Se puede acceder en un entorno gráfico desde la aplicación de terminal.</p> <p>Este editor es mucho más intuitivo que VI, aunque menos potente, proporciona una interfaz que describe los comandos básicos.</p> <p>Para iniciar NANO, debe taclearse desde la línea de comandos:  <code>nano nombre_archivo[.ext]</code></p> <p>Donde “<code>nombre_archivo</code>” puede ser el nombre del archivo a editar o el nombre de un archivo nuevo.</p> <p>Una vez en el editor, en la parte inferior se pueden observar los comandos básicos. Si se presiona la tecla F1 es posible visualizar la ayuda con la lista de todos comandos que existen.</p> <p>Los atajos de teclado pueden corresponder a:</p> <ul style="list-style-type: none"> <li>• ^ que es la tecla Ctrl.</li> <li>• M- que es la tecla Esc o bien Alt.</li> </ul>
<b>GEDIT</b>	<p>Es un editor de entorno gráfico para sistemas basados en UNIX y se encuentra por defecto en el entorno de GNOME.</p> <p>Permite todas las funciones básicas que se pueden encontrar en la mayoría de los editores con las mismas combinaciones de teclas que ya son conocidas por los usuarios.</p> <p>Cuenta con menús en la parte superior que permite acceder a todas las funciones del editor por lo que no requiere conocimientos avanzados para poderlo usar.</p>
<b>Notepad</b>	<p>Es un editor de texto plano de entorno gráfico disponible en todas las ediciones de Windows. Es muy limitado en funcionalidad, la cual es más parecida a un editor de línea de comandos, es conveniente usarlo cuando no existen herramientas adicionales para editar archivos de texto plano.</p>
<b>Notepad++</b>	<p>Es un editor de texto plano diseñado para ejecutarse en entorno gráfico con sistema operativo Windows; es de código libre. Algunas funciones que tiene son:</p>

	<ul style="list-style-type: none"> <li>• Autocompletado,</li> <li>• Corrector Ortográfico</li> <li>• coloreado sintáctico</li> <li>• edición múltiple de archivos</li> <li>• resaltado de paréntesis, etc.</li> </ul> <p>Soporta gran variedad de lenguajes de programación y permite instalar aditamentos para aumentar su funcionalidad.</p>
<b>GitHub Atom</b>	<p>Es otro editor de texto reciente con varias funciones además de soporte para distintos lenguajes de programación, está hecho para correr en entornos gráficos en sistemas operativos Linux, Windows y Mac OS X.</p> <p>Es comparado con Notepad++ con Atom diseñado para usuarios experimentados mientras que Notepad++ es tanto para iniciantes como avanzados.</p>

## Compiladores

Una vez codificado un programa en C en algún editor de texto, éste debe ser leído por un programa que produzca un archivo ejecutable. A este programa se le conoce como compilador y depende totalmente del hardware de la computadora y el sistema operativo que corre sobre ella porque es un traductor que transforma ese lenguaje universal a un programa ejecutable que sólo puede correr ese equipo.

El programa en C, debe ser escrito de manera arbitraria respetando una serie de reglas para que el compilador pueda entenderlas y realizar su función.

Los errores provocan que el compilador no pueda generar el programa ejecutable y muestra en la línea de comandos de qué error se trata y en qué línea pudo haberse producido. Es importante señalar que un solo error puede desencadenar muchos otros y al corregirlo los demás dejarán de ser errores.

<b>GCC (GNU Compiler Collection)</b>	<p>Es un conjunto de compiladores de uso libre para sistemas operativos basados en UNIX.</p> <p>El compilador trabaja en línea de comandos.</p> <p>Existe también una versión modificada que puede correr y crear programas para plataformas Windows en un paquete llamado MinGW (Minimalist GNU for Windows).</p>
--------------------------------------	--

	<p>Al compilar un programa en C el compilador genera diversos archivos intermedios que corresponden a las distintas fases que realiza.</p> <p><b>Otras opciones:</b></p> <p>Suponiendo que se tiene un programa escrito en C y se le llamó calculadora.c la manera de compilarlo es localizándose mediante la línea de comandos en la ruta donde el archivo se encuentra y ejecutando el comando:</p> <p><code>gcc calculadora.c</code></p> <p>Esto creará un archivo a.out (en Windows a.exe) que es el programa ejecutable resultado de la compilación.</p> <p>Si se desea que la salida tenga un nombre en particular, debe definirse por medio del parámetro</p> <p>-o de gcc, por ejemplo, para que se llame calculadora.out (en Windows calculadora.exe):</p> <p><code>gcc calculadora.c -o calculadora.out</code></p> <p>Para realizar un programa más complejo, se necesitan bibliotecas que se instalaron en el equipo previamente y se definió su uso en el programa escrito en C pero al momento de compilar es necesario indicar a GCC que se está usando bibliotecas que no se encuentran en su repertorio de bibliotecas estándar. Para ello es necesario utilizar el parámetro -l seguido inmediatamente por el nombre de la biblioteca, sin dejar espacio alguno:</p> <p><code>gcc calculadora.c -o calculadora -lnombre_libreria</code></p>
<b>LCC</b>	<p>Es un compilador diseñado para ejecutarse en sistemas operativos Windows, sean de 64 bits o 32 bits.</p> <p>A diferencia de GCC, la compilación consiste en dos pasos, el primero genera un archivo objeto y el segundo a partir de éste genera el programa ejecutable.</p> <p>Para poder hacer uso de LCC debe haber sido instalado previamente y agregado al PATH del sistema.</p> <p>Existen opciones adicionales para usuarios avanzados a la hora de invocar al compilador.</p> <p>Si se tiene un programa llamado calculadora.c se debe establecer primero la ruta donde se encuentra el archivo y luego generar el archivo objeto calculadora.obj con el siguiente comando:</p> <p><code>lcc calculadora.c</code></p> <p>Después se tiene que generar el programa ejecutable calculadora.exe por medio de:</p> <p><code>lcclnk calculadora.obj</code></p>

<b>TCC</b>	Es un compilador de uso libre diseñado para ejecutarse en sistemas operativos Windows.
	Para poder hacer uso de TCC debe haber sido instalado previamente y agregado al PATH. Si se tiene un programa llamado calculadora.c se tiene que establecer primero la ruta donde se encuentra el archivo. El siguiente comando permite generar el programa ejecutable y ejecutarlo: <div style="text-align: center;"><code>tcc -run calculadora.c</code></div>

## IDE

Una IDE significa entorno de desarrollo integrado por sus siglas en inglés y combina un editor de textos con un compilador además de varias herramientas que facilitan la programación haciendo todo lo mencionado en esta práctica invisible para el programador.

La siguiente lista presenta varios compiladores e IDE para varios sistemas operativos y arquitecturas. Algunos compiladores soportan otros lenguajes además de C.

<ul style="list-style-type: none"> <li>• GCC</li> <li>• EMX/RXNT</li> <li>• Embarcadero (Borland) □ LCC para Windows de 32 o 64 bits Ch</li> <li>• Microsoft Visual C/C++</li> <li>• Code::Blocks,</li> <li>• MinGW para Windows de 32 o 64 bits</li> </ul>	<ul style="list-style-type: none"> <li>• Cygwin</li> <li>• Miracle C</li> <li>• Dev-C++</li> <li>• Orange C</li> <li>• DJGPP</li> <li>• Pelles C</li> <li>• TCC</li> <li>• Watcom</li> </ul>
---	--

## Ejecución

La ejecución es la etapa que sigue después de haber compilado el programa. Una vez compilado el programa, se puede distribuir para equipos que ejecuten el mismo sistema operativo y tengan la misma plataforma de hardware (tipo de procesador, set de instrucciones y arquitectura en general).

Los pasos para realizar la ejecución dependen del sistema operativo y del entorno.

En Windows se puede ejecutar el programa haciendo doble clic sobre el programa ya compilado, pero recomienda exhaustivamente que se haga desde símbolo de sistema porque, aunque el programa finalice su ejecución, los resultados continuarán siendo visibles en la consola.

Como el programa realizado es para línea de comandos, si se ejecuta en entorno gráfico en Windows el programa sólo se abrirá y se cerrará sin poder ver los resultados de la ejecución, aunque el programa haga sus funciones.



Teniendo un programa llamado calculadora.exe debe teclearse en símbolo de sistema:

`calculadora.exe`

En este caso se localiza previamente en la ruta donde se encuentra el ejecutable.

Si el programa realizado necesita tener una entrada de información por medio de argumentos, éstos se colocan así:

`calculadora argumento1 argumento2`

En un inicio, cualquier programa escrito en C sólo funcionará en modo línea de comandos, es difícil realizar un programa con interfaz gráfica universal y que respete ANSI C, ya que el entorno depende del sistema operativo y las herramientas que provee.

Muchos errores no se reflejarán en el compilador porque el programa está correctamente escrito de acuerdo con lo que ANSI C señala, pero lo que se programó puede ser erróneo y tener resultados distintos a los deseados. Por ello, en la fase de ejecución deben hacerse diversas pruebas para verificar que el programa hace lo que debería.

Es muy común que se construya un programa conforme a lo que ANSI C determina y se pruebe en un equipo personal que, por lo general, es de altas prestaciones. A la hora de instalar ese programa en un equipo especializado que tiene prestaciones limitadas, el programa puede no funcionar ya que no se optimizó y aunque el sistema operativo haga lo posible por ejecutar el programa, será con un rendimiento pobre o simplemente el sistema se colgará.

## Actividad

```
Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\twins>gcc
gcc: fatal error: no input files
compilation terminated.

C:\Users\twins>cd Desktop

C:\Users\twins\Desktop>cd Lenguaje c

C:\Users\twins\Desktop\Lenguaje C>cd ejemplos

C:\Users\twins\Desktop\Lenguaje C\Ejemplos>gcc holaMundo.c -o holaMundo.exe

C:\Users\twins\Desktop\Lenguaje C\Ejemplos>holaMundo.exe
Hola Mundo en C!!!
```