

Entornos de Desarrollo (ED)**Unidad de Trabajo: UT1****Tarea 02: Prueba varios lenguajes de programación**

Descripción de la tarea:

En esta tarea, realizarás la implementación de un programa simple, el famoso "Hola Mundo", en varios

lenguajes de programación. Esto te permitirá familiarizarte con las herramientas y procedimientos básicos para ejecutar programas en diferentes lenguajes, incluyendo lenguajes compilados e interpretados.

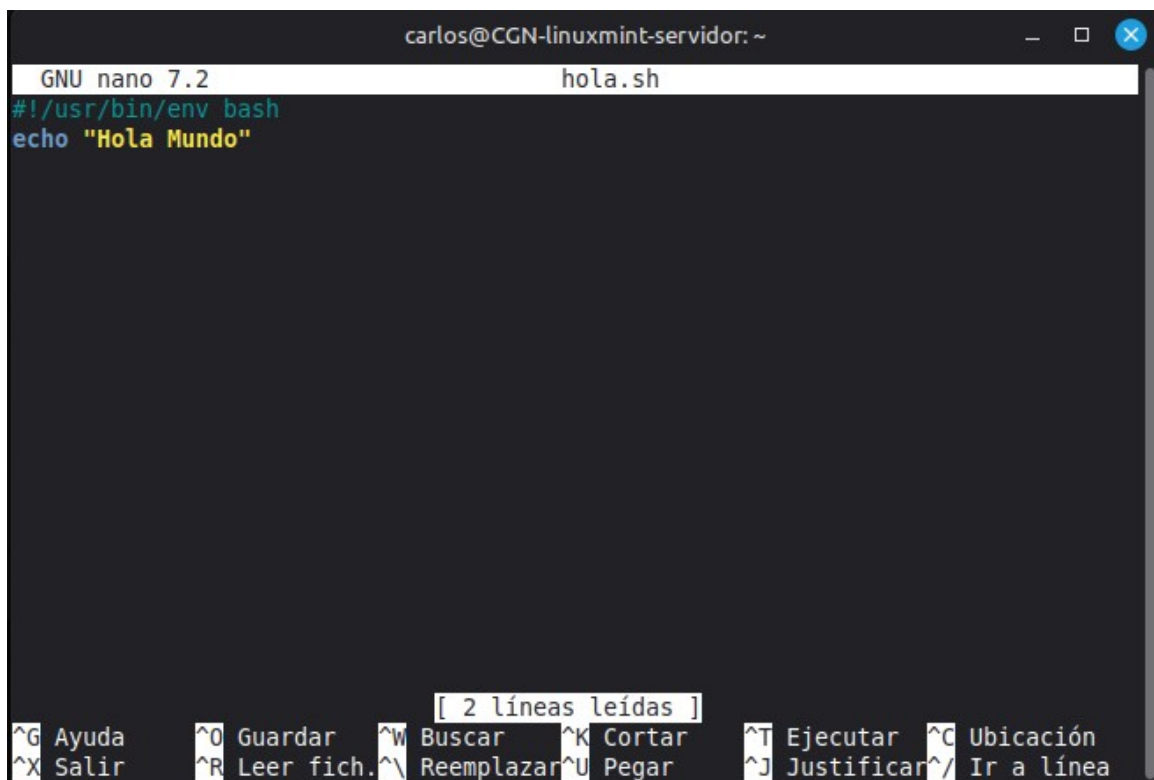
Instrucciones:

1. Lenguajes a utilizar:

- Se te pide que pruebes varios lenguajes de programación, tanto compilados como interpretados.

Entre los lenguajes que deberás utilizar están:

- **Bash**



```
carlos@CGN-linuxmint-servidor: ~  
GNU nano 7.2 hola.sh  
#!/usr/bin/env bash  
echo "Hola Mundo"
```

[2 líneas leídas]

^G Ayuda	^O Guardar	^W Buscar	^K Cortar	^T Ejecutar	^C Ubicación
^X Salir	^R Leer fich.	^\ Reemplazar	^U Pegar	^J Justificar	^_ Ir a línea

The screenshot shows a terminal window titled "Tarea 2 Programacion enlazada [Corriendo] - Oracle VirtualBox". The terminal output shows the installation of Java 17 and the execution of a script named "hola.sh".

```

carlos@CGN-linuxmint-servidor: ~
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/jshell pa
ra proveer /usr/bin/jshell (jshell) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/jstack pa
ra proveer /usr/bin/jstack (jstack) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/jstat par
a proveer /usr/bin/jstat (jstat) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/jstatd pa
ra proveer /usr/bin/jstatd (jstatd) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/serialver
para proveer /usr/bin/serialver (serialver) en modo automático
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/jhsdb par
a proveer /usr/bin/jhsdb (jhsdb) en modo automático
Configurando openjdk-17-jdk:amd64 (17.0.16+8-us1-0ubuntu1~24.04.1) ...
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/jconsole
para proveer /usr/bin/jconsole (jconsole) en modo automático
Procesando disparadores para php8.3-cli (8.3.6-0ubuntu0.24.04.5) ...
Procesando disparadores para libapache2-mod-php8.3 (8.3.6-0ubuntu0.24.04.5) ...
carlos@CGN-linuxmint-servidor:~$ echo "Hola Mundo"
Hola Mundo
carlos@CGN-linuxmint-servidor:~$ nano hola.sh
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.sh
carlos@CGN-linuxmint-servidor:~$ ./hola.sh
Hola Mundo
carlos@CGN-linuxmint-servidor:~$

```

En bash, se debe crear un archivo `.bash` con nano, para la luego darle permisos con `chmod` y finalmente poder ejecutarlo.

■Python

The screenshot shows a terminal window titled "carlos@CGN-linuxmint-servidor: ~". The terminal output shows the creation of a Python script named "hola.py" using the nano editor.

```

GNU nano 7.2 hola.py
#!/usr/bin/env python3
print ("Hola mundo")

```

The bottom of the terminal shows the nano editor's status bar with various keyboard shortcuts:

```

[ 2 líneas leídas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir  ^R Leer fich. ^_ Reemplazar ^U Pegar  ^J Justificar ^_ Ir a línea

```

```

Tarea 2 Programacion enlazada [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

MySQL Workbench

carlos@CGN-linuxmint-servidor:~
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/jhsdb para proveer /usr/bin/jhsdb (jhsdb) en modo automático
Configurando openjdk-17-jdk:amd64 (17.0.16+8-us1-0ubuntu1-24.04.1) ...
update-alternatives: utilizando /usr/lib/jvm/java-17-openjdk-amd64/bin/jconsole para proveer /usr/bin/jconsole (jconsole) en modo automático
Procesando disparadores para php8.3-cli (8.3.6-0ubuntu0.24.04.5) ...
Procesando disparadores para libapache2-mod-php8.3 (8.3.6-0ubuntu0.24.04.5) ...
carlos@CGN-linuxmint-servidor:~$ echo "Hola Mundo"
Hola Mundo
carlos@CGN-linuxmint-servidor:~$ nano hola.sh
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.sh
carlos@CGN-linuxmint-servidor:~$ ./hola.sh
Hola Mundo
carlos@CGN-linuxmint-servidor:~$ python3
Python 3.12.3 (main, Aug 14 2025, 17:47:21) [GCC 13.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hola Mundo")
Hola Mundo
>>>
carlos@CGN-linuxmint-servidor:~$ nano hola.py
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.py
carlos@CGN-linuxmint-servidor:~$ ./hola.py
Hola mundo
carlos@CGN-linuxmint-servidor:~$

```

En python ,es casi lo mismo que bash, crear un archibo .py, darle permisos y finalmente ejecutarlo.

■ PHP

```

carlos@CGN-linuxmint-servidor:~
GNU nano 7.2 hola.php
#!/usr/bin/env php
<?php
echo "Hola mundo\n"
?>

[ 4 líneas leídas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir  ^R Leer fich. ^\ Reemplazar ^U Pegar  ^J Justificar ^_ Ir a línea

```

```

carlos@CGN-linuxmint-servidor: ~
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hola Mundo")
Hola Mundo
>>>
carlos@CGN-linuxmint-servidor:~$ nano hola.py
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.py
carlos@CGN-linuxmint-servidor:~$ ./hola.py
Hola mundo
carlos@CGN-linuxmint-servidor:~$ php -a
Interactive shell

php > echo "Hola mundo\n";
Hola mundo
php > ^Dcarlos@CGN-linuxmint-servidor:~$ nano hola.php
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.php
carlos@CGN-linuxmint-servidor:~$ ./hola.php
>?php
    echo "Hola mundo\n"
?>
carlos@CGN-linuxmint-servidor:~$ nano hola.php
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.php
carlos@CGN-linuxmint-servidor:~$ ./hola.php
Hola mundo
carlos@CGN-linuxmint-servidor:~$

```

En php es lo mismo, sin embargo, en el archivo equivoque en el código y por eso al ejecutarlo se muestra así (>?php), por lo que edite el archivo y lo corregí (<?php).

■ JavaScript (Node.js)

```

carlos@CGN-linuxmint-servidor:~$ node
Welcome to Node.js v18.19.1.
Type ".help" for more information.
> console.log('Hola mundo');
Hola mundo
>

```

Java script, ejecutamos el intérprete y imprimimos el comando que queremos que salga.

■ C

```

carlos@CGN-linuxmint-servidor: ~
GNU nano 7.2 hola.c
#include <stdio.h>
int main()
{
    printf("¡Hola, mundo!");
    return 0;
}

[ 6 líneas leídas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir   ^R Leer fich. ^\ Reemplazar ^U Pegar    ^J Justificar ^_ Ir a línea
  
```

```

Tarea 2 Programacion enlazada [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

MySQL Workbench

carlos@CGN-linuxmint-servidor: ~
php > echo "Hola mundo\n";
Hola mundo
php > ^Dcarlos@CGN-linuxmint-servidor:~$ nano hola.php
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.php
carlos@CGN-linuxmint-servidor:~$ ./hola.php
>?php
    echo "Hola mundo\n"
?>
carlos@CGN-linuxmint-servidor:~$ nano hola.php
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.php
carlos@CGN-linuxmint-servidor:~$ ./hola.php
Hola mundo
carlos@CGN-linuxmint-servidor:~$ node
Welcome to Node.js v18.19.1.
Type ".help" for more information.
> console.log('Hola mundo');
Hola mundo
undefined
>
carlos@CGN-linuxmint-servidor:~$ nano hola.c
carlos@CGN-linuxmint-servidor:~$ gcc -o hola hola.c
carlos@CGN-linuxmint-servidor:~$ ./hola
¡Hola, mundo!carlos@CGN-linuxmint-servidor:~$
  
```

Con c, es tan fácil como crear el archivo .c, compilarlo con el comando gcc y ejecutarlo.

■Ensamblador(NASM)

```

carlos@CGN-linuxmint-servidor: ~
GNU nano 7.2 hola.asm
section .data
msg db "¡Hola Mundo!", 0Ah
len equ $ - msg
section .text
global _start
_start:
    mov eax, 04h
    mov ebx, 01h
    mov ecx, msg
    mov edx, len
    int 80h
    mov eax, 01h
    mov ebx, 00h
    int 80h

[ 15 líneas leídas ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^_ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
  
```

```

Tarea 2 Programacion enlazada [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

MySQL Workbench

carlos@CGN-linuxmint-servidor: ~
hola.asm:6: warning: label alone on a line without a colon might be in error [-w
+label-orphan]
hola.asm:14: error: parser: instruction expected
hola.asm:17: error: parser: instruction expected
hola.asm:18: error: parser: instruction expected
carlos@CGN-linuxmint-servidor:~$ ld hola.o -o hola
ld: no se puede encontrar hola.o: No existe el archivo o el directorio
carlos@CGN-linuxmint-servidor:~$ nano hola.asm
carlos@CGN-linuxmint-servidor:~$ nasm -f elf64 hola.asm
hola.asm:5: warning: label alone on a line without a colon might be in error [-w
+label-orphan]
hola.asm:6: warning: label alone on a line without a colon might be in error [-w
+label-orphan]
hola.asm:14: error: parser: instruction expected
carlos@CGN-linuxmint-servidor:~$ nano hola.asm
carlos@CGN-linuxmint-servidor:~$ nasm -f elf64 hola.asm
hola.asm:6: warning: label alone on a line without a colon might be in error [-w
+label-orphan]
carlos@CGN-linuxmint-servidor:~$ ld hola.o -o hola
ld: aviso: no se puede encontrar el símbolo de entrada _start; se usa por defect
o 0000000000401000
carlos@CGN-linuxmint-servidor:~$ ./hola
¡Hola Mundo!
carlos@CGN-linuxmint-servidor:~$
  
```

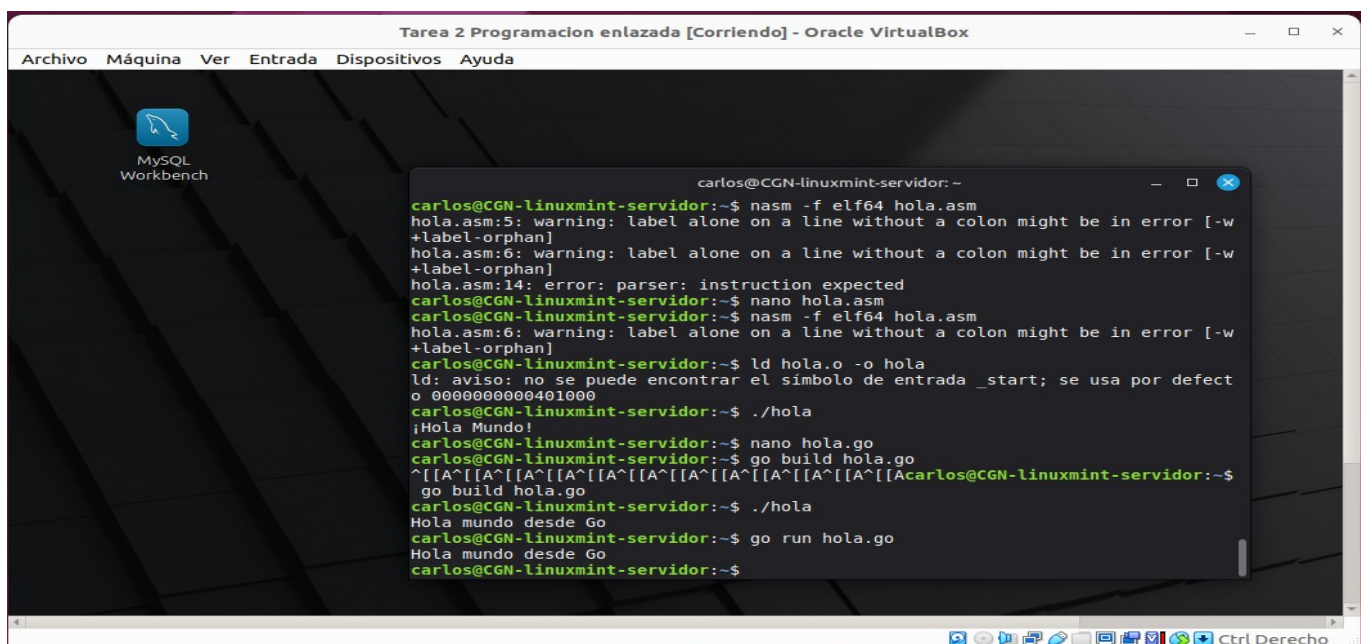
Ensamblador son mas pasos, o más código que escribir. Creamos el archivo .asm y luego hay que ensamblar y enlazar con nasm y ld para finalmente poder ejecutarlo.

■ **Go**

```
carlos@CGN-linuxmint-servidor: ~
GNU nano 7.2 hola.go
package main
import "fmt"
func main () {
    fmt.Println("Hola mundo desde Go")
}
```

[6 líneas leídas]

^G Ayuda	^O Guardar	^W Buscar	^K Cortar	^T Ejecutar	^C Ubicación
^X Salir	^R Leer fich.	^N Reemplazar	^U Pegar	^J Justificar	^_ Ir a línea



Creamos el archivo .go y luego compilamos con go build, pero presione luego ctrl+A, se ahí que salga ^[[A^ ... , pero luego lo ejecute bien, también se puede interpretar con go run.

■ Lisp

```

carlos@CGN-linuxmint-servidor: ~
GNU nano 7.2 hola.lisp
#!/usr/bin/env clisp
(format t "~;Hola, mundo!")

[ 2 líneas leídas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir  ^R Leer fich. ^_ Reemplazar ^U Pegar  ^J Justificar ^_ Ir a línea

```

```

Tarea 2 Programacion enlazada [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

MySQL Workbench

carlos@CGN-linuxmint-servidor: ~
Copyright (c) Bruno Haible, Pierpaolo Bernardi, Sam Steingold 1998
Copyright (c) Bruno Haible, Sam Steingold 1999-2000
Copyright (c) Sam Steingold, Bruno Haible 2001-2018

Teclee :h y pulse Intro para ayuda contextual.

[1]> (format t "~;Hola, mundo!")
¡Hola, mundo!
NIL
[2]>

Adiós.
carlos@CGN-linuxmint-servidor:~$ nano hola.lisp
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.lisp
carlos@CGN-linuxmint-servidor:~$ ./hola.lisp
*** - READ: el flujo de entrada
      #<INPUT BUFFERED FILE-STREAM CHARACTER #P"hola.lisp" @3> termina con una
      cadena
carlos@CGN-linuxmint-servidor:~$ nano hola.lisp
carlos@CGN-linuxmint-servidor:~$ chmod +x hola.lisp
carlos@CGN-linuxmint-servidor:~$ ./hola.lisp
¡Hola, mundo!
carlos@CGN-linuxmint-servidor:~$

```

Este lenguaje, tiene un interprete mas elaborado que los anteriores, sin embargo, se ejecuta igual que bash y python, se le da permisos y se ejecuta.