



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:*

TISTA GARCÍA EDGAR

*Asignatura:*

PROGRAMACIÓN ORIENTADA A OBJETOS

*Grupo:*

3

*No de Práctica(s):*

PRÁCTICA 1 – ENTORNO Y LENGUAJE DE PROGRAMACIÓN

*Integrante(s):*

CARRILLO CERVANTES IVETTE ALEJANDRA

*No. de Equipo de  
cómputo empleado:*

TRABAJO EN CASA

*No. de Lista o Brigada:*

-

*Semestre:*

2022 - 1

*Fecha de entrega:*

09 SEPTIEMBRE 2021

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## PRÁCTICA #1: Entorno y lenguaje de programación

**OBJETIVO:** Identificar y probar el entorno de ejecución y el lenguaje de programación orientado a objetos a utilizar durante el curso.

**OBJETIVO DE CLASE:** Tener una primera aproximación al lenguaje Java, realizar programas sencillos y familiarizarse con el uso de la consola.

### EJEMPLOS DE LA GUIA

#### Ejemplo 1

Este programa tiene como función principal, además de conocer la sintaxis básica de un programa en Java, imprimir un “Hola Mundo” con ayuda de la función *println* desde la terminal de nuestra computadora, esto se logra al compilar y ejecutar el programa (dentro de *Java Virtual Machine*), usando el comando java seguido del nombre de la clase que contiene el método *main()*. Al compilar y ejecutar desde consola este programa, la salida es la siguiente:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervan  
tes Ivette Alejandra G3 P1 V1\Ejemplos>javac HolaMundo.java  
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervan  
tes Ivette Alejandra G3 P1 V1\Ejemplos>java HolaMundo  
Hola Mundo (:
```

*./HolaMundo.java*

### ACTIVIDADES PARA EL DESARROLLO DE LA PRACTICA

En la entrega número 1 de esta práctica, al realizar alguna ejecución no se imprime ningún acento, para corregir esto se activó la página de códigos 65001 mediante la terminal.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervan  
tes Ivette Alejandra G3 P1 V2>chcp 65001  
Página de códigos activa: 65001
```

*página de códigos 65001*

#### Ejercicio 1

Se compiló y ejecutó este programa; sin embargo, se muestran varios errores en pantalla. El primer error que se encontró se debe a que en el método *main()* falta la palabra reservada *void*, la cual indica que el método no retorna ningún valor, por lo cual se agregó dicha palabra.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervan  
tes Ivette Alejandra G3 P1 V1>javac Ejercicio1.java  
Ejercicio1.java:10: error: invalid method declaration; return type required  
    static public main(String[] Juan) {  
            ^  
1 error
```

*error1*

```
19 static public void main(String[] Juan) {
```

corrección1

El segundo error que se encontró en este programa se debe a que el nombre de la clase está mal escrito, pues recordemos que el nombre de la clase debe de coincidir con el nombre del archivo .java, por lo cual se corrigió el nombre de la clase.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carriillo Cervantes Ivette Alejandra G3 P1 V1>javac Ejercicio1.java
Ejercicio1.java:1: error: class Ejercicio1 is public, should be declared in a file named Ejercicio1.java
public class Ejercicio1{
      ^
1 error
```

error2

```
4 public class Ejercicio1{
```

corrección2

Después, al momento de compilar y ejecutar el programa esta vez sin errores, se observa que la función principal de este, es imprimir varias cadenas de texto mediante la función *println*; no obstante, en una de estas cadenas se imprime la palabra “resultado” junto a unos números previamente inicializados (*a=10*, *b=100*), si lo imprimimos la salida quedaría como: “resultado10100”, pues la función antes descrita *println* toma a el valor de *a* y el valor de *b* como una cadena de caracteres. Ahora, si queremos imprimir el resultado de la suma de ambos números, ponemos modificar esa instrucción agregando paréntesis a la suma de *a* y *b* como se muestra a continuación:

```
System.out.println("resultado "+(a+b));
```

corrección3

También, se llama al *metodo1* de la clase *Ejercicio1* dentro de la función *main*, con el fin de que se ejecuten las instrucciones de este método, esto se logra mediante la siguiente linea de código:

```
30 Ejercicio1.metodo1();
```

llamada al método 1

Finalmente, al volver a compilar y ejecutar el programa, la salida es la siguiente:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carriillo Cervantes Ivette Alejandra G3 P1 V2>java Ejercicio1
Práctica1 2022-1
Resultado: 110
Práctica 1 y ya aprobé el semestre (:
```

./Ejercicio1.java

## Ejercicio2

Este programa tiene como función principal, observar las diferencias entre las diferentes maneras de imprimir datos en pantalla, recordemos que esto se logra a través de las funciones “*println*”, “*print*” y “*printf*”.

### 1ra ejecución con *println*:

Al compilar y ejecutar por primera vez este programa, se observó que se utilizó la función “*println*” para imprimir los datos en pantalla. Usando esta función, la salida en pantalla se ajusta, es decir, no hay necesidad de poner saltos de línea entre una y otra instrucción para imprimir; pero, si se quiere agregar el valor de alguna variable previamente inicializada, se debe de poner un signo “+” fuera de las comillas en donde se encuentra el texto, seguido de la variable que se quiera poner.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V1>javac Ejercicio2.java
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V1>java Ejercicio2
Asignaturas del 3er semestre de computacion
* Programacion Orientada a objetos
* Estructura de datos y algoritmos 2
* Las
* otras
* no
* cuentan
* bueno si
```

*ejecución1 del archivo ./Ejercicio2.java*

### 2da ejecución con *print*:

Al llevar a cabo la segunda ejecución, se observó que se utilizó la función “*print*” para imprimir los datos en pantalla; sin embargo, a diferencia de la función anterior, la salida en pantalla no se ajusta, por lo que, si se quiere imprimir alguna cadena de texto que involucre 2 veces o más la misma función, es necesario escribir un salto de línea “*\n*” para que no quede todo ajustado (cabe recalcar que esta función se usa en el lenguaje de programación Python).

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V1>javac Ejercicio2.java
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V1>java Ejercicio2
Aquí estoy Imprimiendo
Hola
```

*ejecución2 del archivo ./Ejercicio2.java*

### 3ra ejecución con *printf*:

Finalmente, al llevar a cabo la tercera ejecución, se observó que se utilizó la función “*printf*” para imprimir los datos en pantalla, y al igual que la función anterior, la salida en pantalla no se ajusta; no obstante, en esta función, si se requiere agregar una variable previamente inicializada, se pondrá dentro de las comillas (donde va el texto) un especificador de formato según el tipo de la variable que se quiera, y fuera de las comillas se debe poner una coma, seguido de la variable que se quiera agregar concordando con el especificador de formato (cabe recalcar que esta función se usa en el lenguaje de programación C).

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V1>javac Ejercicio2.java
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V1>java Ejercicio2
Un entero: 10
Una cadena: prueba
Dos cadenas: Aprende Java En 21 días
```

*ejecución3 del archivo ./Ejercicio2.java*

### Ejercicio 3

Este programa tiene como función principal llamar a diferentes métodos de diferentes clases desde el método principal *main* de la clase “*Ejercicio3*” y así poder comprender mejor el concepto de método y clase. Durante el proceso de compilación de este programa se generaron los archivos .class de la clase 1 y de la clase 2; sin embargo, el archivo de la clase 3 nunca se crea, esto debido a que no se manda a llamar en ningún momento del programa dicha clase, la impresión de pantalla se ve de la siguiente manera:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V1\Ejercicio 3>javac Ejercicio3.java
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V1\Ejercicio 3>java Ejercicio3
a vale: 40
Hola desde aqui
Ahora a vale: 23
```

*./Ejercicio3/Ejercicio3.java*

Como se mencionó anteriormente, se llamaron a diferentes métodos de diferentes clases; sin embargo, no todas tienen la instrucción para que se puedan imprimir en pantalla, ya que algunos métodos solo retornan un valor, por lo cual al momento de llamar a esos métodos desde el método principal *main*, se agregaron las instrucciones correspondientes para ver el valor de retorno de cada método.

```
27      System.out.println(Clase2.metodo1());
```

*instrucción para imprimir el valor de retorno de un método*

Aparte, se mandó a llamar al método faltante generando así al momento de compilar el programa un archivo .class de la clase 3; además, se agregaron instrucciones extras (igual de impresión) para saber en qué clase y método estamos.

```
3      public static String metodo1(){
4          // Se agregó una impresión en pantalla para saber en que clase se esta
5          System.out.println("\n-> Clase 1, método 1");
```

*instrucciones para saber en que clase y método nos encontramos*

Finalmente, la salida del programa es la siguiente:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V2\Ejercicio 3>javac Ejercicio3.java
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V2\Ejercicio 3>java Ejercicio3
-> Clase Ejercicio 3, método main
a vale: 40
-> Clase 1, método 1
Programación Orientada a objetos
-> Clase 1, método 2
Hola desde aqui
Ahora 'a' vale: 23
-> Clase 2, método 1
¿Puedo usar cosas que se encuentren en otros archivos (clases)?
-> Clase 3, método 8
Saludos desde la clase 3
Aquí adentro a vale 10 XD
```

*./Ejercicio3/Ejercicio3.java*

## Ejercicio 4

Este programa tiene como función principal, dados dos números enteros positivos previamente inicializados, hacer diferentes tipos de operaciones con estos. Este ejercicio se realizó con dos archivos guardados en la carpeta “Ejercicio4”, en el primer archivo se encuentra la clase “Ejercicio4” junto con el método main, en donde se manda a llamar a los diferentes métodos de otra clase llamada “Calculadora” (segundo archivo), estos tienen como parámetro el valor de “a” y el valor de “b” para hacer las operaciones correspondientes a cada método:

```
6      // Método suma
7      public static int suma(int a, int b){
13     // Método resta
14     public static int resta(int a, int b){
20     // Método multiplicación
21     public static int multiplicacion(int a, int b){
27     // Método división
28     public static int division(int a, int b){
34     // Método módulo
35     public static int modulo(int a, int b){
```

*métodos en ./Ejercicio4/Calculadora.java*

Durante la realización de este programa considero que las dificultades que se me presentaron fueron muy básicas, por lo que fue sencillo para mi corregirlas, estas se presentaron al momento de llamar a cualquier método y pasar como parámetros los valores de “a” y de “b” respectivamente, pues pensé que había alguna diferencia en llamar a algún método en Java y llamar a una función en C; sin embargo, después me di cuenta de que prácticamente es lo mismo.

Finalmente, la salida del programa queda de la siguiente manera:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V2\Ejercicio 4>javac Ejercicio4.java
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P1\Carrillo Cervantes Ivette Alejandra G3 P1 V2\Ejercicio 4>java Ejercicio4
- - - - - CALCULADORA - - - - -
El valor de a: 10
El valor de b: 5
La suma entre a+b es: 15
La resta entre a-b es: 5
La multiplicación a*b es: 50
La división a/b es: 2
El módulo a%b es: 0
```

*./Ejercicio4/Ejercicio4.java*

## Ejercicio 5

Para este último ejercicio se ocupó el IDE NetBeans, al principio no sabía como trabajar en este IDE, pero buscando información me di cuenta de que se trabaja por medio de proyectos y no por medio de archivos. Una vez comprendido lo anterior y descargando los paquetes necesarios para ocupar esta aplicación (*los cuales pueden contener definiciones de tipos como las clases e interfaces, dando lugar a estructuras jerárquicas de contenedores alojando los módulos .java con el código fuente*), se creó un nuevo proyecto llamado Ejercicio 5 y dentro de este un archivo de tipo Java main class llamado de la misma manera; a su vez, dentro de esta clase se encuentra el método main el cual tiene la instrucción de imprimir un “Hola Mundo”. Al ejecutar el programa, se vio una de las principales diferencias que tiene este IDE con un programa ejecutado desde la terminal, pues en este caso al ser un archivo Java main class, no se generó algún archivo .class, se muestra en pantalla lo siguiente:

```
Scanning for projects...

-----< com.mycompany:Ejercicio5 >-----
[ ] Building Ejercicio5 1.0-SNAPSHOT
-----[ jar ]-----
[ ] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Ejercicio5 ---
    Hola Mundo (:
-----
BUILD SUCCESS
-----

Total time:  2.032 s
Finished at: 2021-09-09T20:18:43-05:00
-----

Ejercicio5.java
```

Una vez que se verificó que la salida del programa sea correcta, se agregaron algunas instrucciones incorrectas (de variables) las cuales debíamos de corregir, al tratar de ejecutar las instrucciones el programa tiene como salida el siguiente error:

```
BUILD FAILURE
-----
Total time:  2.135 s
Finished at: 2021-09-09T11:24:42-05:00
-----
Failed to execute goal org.codehaus.mojo:exec-maven-plugin:3.0.0:exec (default-cli) on project Ejercicio5: Command execution failed.:
To see the full stack trace of the errors, re-run Maven with the -e switch.
Re-run Maven using the -X switch to enable full debug logging.

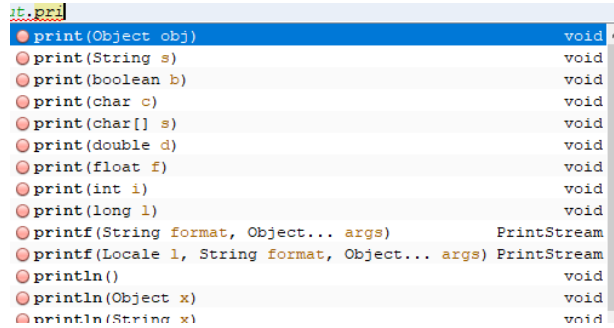
For more information about the errors and possible solutions, please read the following articles:
[Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException
```

error

11

Dentro del programa se observó que a lado de las líneas en donde pusimos los datos incorrectos en un principio, se encuentran unos foquitos con una señal de advertencia, la cual indica que algo en esa línea está mal definido, o bien, mal estructurado. Al presionar dichas figuras, te aparecen opciones las cuales te indican que es lo que probablemente este mal y lo puedas corregir antes de ejecutar. En cuanto a la generación de código automático, al momento de escribir alguna función dentro del programa, te aparece una lista de opciones las cuales puedes ocupar durante el desarrollo de tu programa; por ejemplo, al iniciar a escribir la función `println`, nos muestra lo mencionado anteriormente:





*generación de código automático*

Finalmente, corregimos las instrucciones incorrectas que se dieron a un principio de la práctica, de la siguiente manera:

*// Int a = 23; <- el tipo de dato int se escribe toto en minúsculas*  
 int a = 23;

*// float 23 = 3.14; <- una variable no puede ser un número, además agregamos la f*  
 float var23 = 3.14f;

*//double Alumn@ = 9.811956; <- no puede llevar @ una variable*  
 double Alumno = 9.811956;

*//char \_Var = "Hola"; <- Solo se puede ingresar un carácter*  
 char \_Var = 'A';

*//int null = 24; <- null es una palabra reservada*  
 int nulle = 24;

*//int Null = 10.6; <- Debe de ser de tipo entero, (la letra "n" va en mayúscula, por lo tanto, no es una palabra reservada)*  
 int Null = 10;

*//string cadena = "hola"; <- La palabra String, comienza con mayúscula*  
 String cadena = "hola";

La salida del programa completo, agregando instrucciones para que se imprima en pantalla el valor de cada variable, es la siguiente:

```

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Ejercicio5 ---
Hola Mundo (:
TIPOS PRIMITIVOS
Esto es un entero: 23
Esto es un flotante: 3.14
Esto es un double: 9.811956
Esto es un caracter: A
Esto es un entero: 24
Esto es un entero: 10
Esto es una cadena: hola

-----
BUILD SUCCESS
-----
Total time: 1.349 s
Finished at: 2021-09-09T11:51:58-05:00
-----

```

*Ejercicio5.java*



## Conclusiones

Se cumplieron los objetivos de esta práctica ya que tuvimos una mejor aproximación al lenguaje Java, vimos su sintaxis, así como sus funciones y conceptos principales. Además, identificamos y probamos el entorno de ejecución del lenguaje Java mediante el Símbolo del sistema, al igual que mediante un IDE, en este caso fue NetBeans. Se pudo observar que la principal diferencia entre la ejecución en ambos casos fue la forma de ejecución y los paquetes que tiene NetBeans para poderse ocupar.

Considero que cada programa si contribuyó al tema visto durante las últimas clases, ya que pudimos ver de una manera más práctica como se implementan los programas en Java, en el ejercicio 1 se pudo observar algunos de los principales errores que cometemos al trabajar con Java (personalmente he usado muy poco este lenguaje, pero si he tenido el error de llamar diferente la clase y el archivo); en el ejercicio 2 se observó las diferencias entre las funciones que se tienen para imprimir datos en pantalla y me percaté que 2 de esas funciones ya las había ocupado en lenguaje C y en lenguaje Python; en el ejercicio 3, se hizo uso de métodos; en el ejercicio 4 considero que fue una buena práctica de métodos, así como de llamar a un método que cuenta con parámetros; mientras que en el ejercicio 5, como se menciono anteriormente se hizo uso de un IDE y además de corregir algunas instrucciones que estaban mal, con este ejercicio se vio las principales diferencias entre un IDE y la ejecución del programa mediante el Símbolo del Sistema.

Creo que esta práctica es fundamental para el inicio del curso, pues se mencionan aspectos muy importantes que se deben de tener en cuenta para iniciar a programar con lenguaje JAVA; considero que los ejercicios estan muy bien planteados para esta práctica. (: