

USO DE UN ENTORNO DE DESARROLLO INTEGRADO - IDE PARA LA IMPLEMENTACIÓN Y PRUEBA DE SOLUCIONES COMPUTACIONALES

Objetivos

- Reconocer un entorno de desarrollo integrado (IDE) para la implementación y prueba de soluciones computacionales.
- Identificar las etapas para la implementación de una solución computacional en IDE empleando un lenguaje de programación de alto nivel.
- Reconocer las herramientas para prueba de las soluciones computacionales.

Recursos requeridos

- PC con NetBeans instalado

Actividades preliminares al laboratorio

- Lectura de la guía
- Instalación de NetBeans en PC personal (opcional)

Marco teórico

Definiciones Básicas

Algoritmo: un algoritmo es una secuencia ordenada de pasos lógicos que permiten transformar datos de entrada en nuevos datos de salida. Para la descripción de un algoritmo se deben identificar al menos tres elementos: los parámetros de entrada, los parámetros de salida, y el paso a paso que permite manipular los datos de entrada para llegar a la salida. Una forma más fácil de entender que es un algoritmo es asimilarlos a una receta de cocina, donde los ingredientes en nuestro caso son los parámetros de entrada y el producto esperado son los parámetros de salida. Como en una receta de cocina, el paso a paso debe seguir una secuencia lógica para obtener el resultado esperado, esto quiere decir que el orden en que se ejecutan las instrucciones afectará el producto final. La descripción, presentación y diseño de un algoritmo se puede representar mediante un pseudocódigo.

Seudocódigo: el pseudocódigo es una descripción de alto nivel de los pasos lógicos que hacen parte de un algoritmo. En un pseudocódigo se identifican los datos de entrada, los datos de salida, y el paso a paso que permite la transformación de los datos de entrada a la salida. Dado que son descripciones de alto nivel usualmente no se incluyen declaraciones de tipos de variables y algunas funciones o métodos que corresponden a operaciones ampliamente conocidas (ejemplo: la función *length* para retornar la longitud de un vector). Los pseudocódigos emplean palabras reservadas que son ampliamente usadas en diferentes lenguajes de programación, tales como las palabras FOR y WHILE reservadas para ciclos para y mientras, o la palabra IF reservada para secuencias de control condicional. Sin embargo, los pseudocódigos son independientes del lenguaje de programación que se empleará para la implementación de la solución computacional; este debe incluir la descripción suficiente y necesaria para que un programador pueda codificar la solución en cualquier lenguaje de programación. Se recomienda la escritura del pseudocódigo en inglés, esto permite que este sea entendible por cualquier programador alrededor del mundo.

Paradigma de programación: Los paradigmas de programación definen el estilo para diseñar una solución computacional para resolver un problema. Entre los paradigmas de programación de mayor uso se encuentran

la programación imperativa, la programación lógica, la programación funcional y la programación orientada a objetos. Un error común es confundir un paradigma de programación con un lenguaje de programación. Los lenguajes de programación permiten la implementación de programas que siguen uno o más paradigmas.

Lenguaje de programación: define el conjunto de símbolos, instrucciones y reglas gramaticales que permiten implementar una solución computacional, para que este sea interpretado y ejecutado en nuestro computador. Los lenguajes de programación se pueden clasificar como lenguajes de alto y bajo nivel de acuerdo con el nivel de abstracción respecto a la arquitectura del computador. Por otra parte, los lenguajes también se pueden clasificar de acuerdo con el paradigma, aunque un lenguaje puede soportar uno o más modelos de cómputo. Entre otros elementos, los lenguajes de programación definen como declarar, inicializar y manipular variables y arreglos, como establecer estructuras de control como condicionales y ciclos, así como la declaración y uso de funciones.

Entorno de desarrollo integrado: un IDE (por sus siglas en inglés *Integrated Development Environment*) es una aplicación o software que facilita la implementación de una solución computacional. Un IDE incluye al menos un editor de código fuente, un compilador y un depurador. El editor de código fuentes es un editor de texto diseñado específicamente para la escritura de códigos de un programa. Los editores integrados en IDE están diseñados para simplificar la escritura de código, con herramientas como la identificación de palabras reservadas, indexación, autocompletar y verificadores de coincidencia de corchetes. Por otra parte, el compilador traduce el código escrito en determinado lenguaje de programación a otro lenguaje usualmente de más bajo nivel, para construir un programa ejecutable. Un depurador es un programa que permite probar y depurar otros programas. La principal tarea de un depurador es correr un programa bajo condiciones controladas evaluando el progreso de las operaciones y el uso de recursos.

Fundamentos del Lenguaje de Programación JAVA

JAVA es un lenguaje de programación orientada a **objetos**, por lo cual los principales actores son objetos que almacenan datos y proporcionada métodos para modificar o acceder a los datos. Cada objeto es instanciado por una **clase**, que define un **tipo** de objeto. La Figura 1 describe los elementos claves para un programa principal en JAVA.

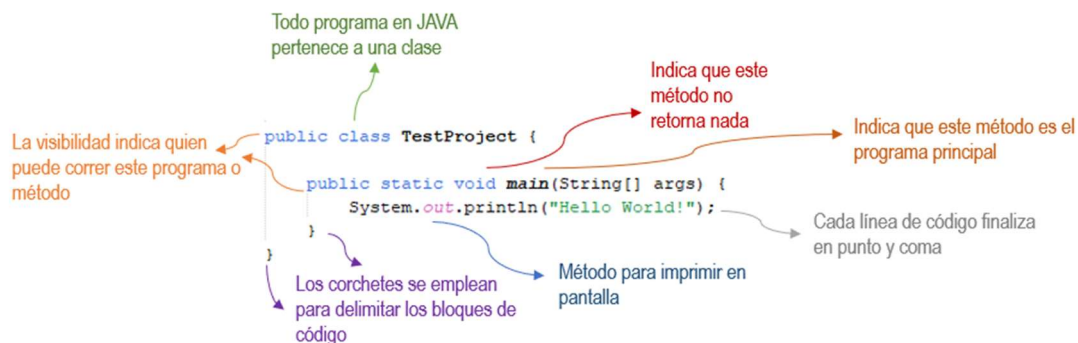


Figura 1. Elementos de un programa "Hello World" en JAVA

JAVA incluye diferentes tipos de datos primitivos (Tabla 1). Una variable declarada como uno de estos tipos de datos almacena directamente en la memoria el valor correspondiente.

boolean	Variable booleana: true o false
char	Carácter 16-bit
byte	Entero complemento dos 8-bit
short	Entero complemento dos 16-bit
int	Entero complemento dos 32-bit
long	Entero complemento dos 64-bit
float	Flotante 32-bit
double	Flotante 64-bit

Tabla 1. Tipos de datos primitivos.

Un tipo especial de dato es la cadena (String), compuesta por una secuencia de caracteres. Las variables de tipo String en JAVA son objetos que permiten hacer operaciones como la concatenación e indexación.

La Tabla 2 presenta la sintaxis para los principales controles de flujo de un programa en JAVA: el condicional IF, y los ciclos FOR y WHILE.

Sintaxis	Ejemplo
<pre>if (boolean_exp) { true_statement; } else { false_statement; }</pre>	<pre>public class TestProject { public static void main(String[] args) { int value = 1; if (value==0){ System.out.println("Values is 0"); } else if (value==1){ System.out.println("Values is 1"); } else { System.out.println("Values is different to 0 and 1"); } } }</pre>
<pre>if (boolean_exp) { true_statement; } else if(second_boolean_exp){ second_true_statement; } else { false_statement; }</pre>	

<pre>while (boolean_exp) { loop_statement; }</pre>	<pre>public class TestProject { public static void main(String[] args) { int value = 10; while (value>=0){ System.out.println("Hello world"); value--; } } }</pre>
<pre>for (int;cond;inc){ loop_statement; }</pre>	<pre>public class TestProject { public static void main(String[] args) { for (int value=10; value>=0; value--){ System.out.println("Hello world"); } } }</pre>

Tabla 2. Sintaxis para ciclos y condicionales.

Actividades preliminares al laboratorio

Instalación de NetBeans (PC personal - opcional)

Las salas de cómputo cuentan con PC que tienen instalado NetBeans. Esta actividad es opcional y está dirigida a instalar el IDE NetBeans en sus computadoras personales.

Para la instalación de NetBeans, descargue la versión más actualizada desde la página web: <https://netbeans.apache.org/front/main/download/>. Recuerde descargar el instalador que corresponde al sistema operativo de su computadora personal. NetBeans es soportado por Windows, Linux y MacOS. Ejecute el instalador siguiendo los pasos que este le va indicando.

Exploración de interfaz grafica

Una vez instalado NetBeans, vamos a explorar la interfaz gráfica. Por defecto, este IDE incluye cuatro ventanas y el menú principal. En la Figura 2 se presenta la interfaz gráfica por defecto de NetBeans. En la ventana 1 encontraremos el directorio de proyectos y archivos disponibles. La ventana 2 está destinada a el editor de código; en esta parte de la interfaz es donde estaremos escribiendo nuestro programa. La ventana 3 despliega información referente al código que estamos implementando; en el caso de una clase, en esta parte del IDE se mostrará los atributos y métodos de la clase que estemos programando. La ventana 4 corresponde a la consola de salida; esta consola mostrará información relacionada con la compilación del programa y las salidas que se muestren en pantalla. El menú principal (sección 5 en la Figura 2) incluye diferentes herramientas que nos ayudaran en la programación de nuestras soluciones computacionales.

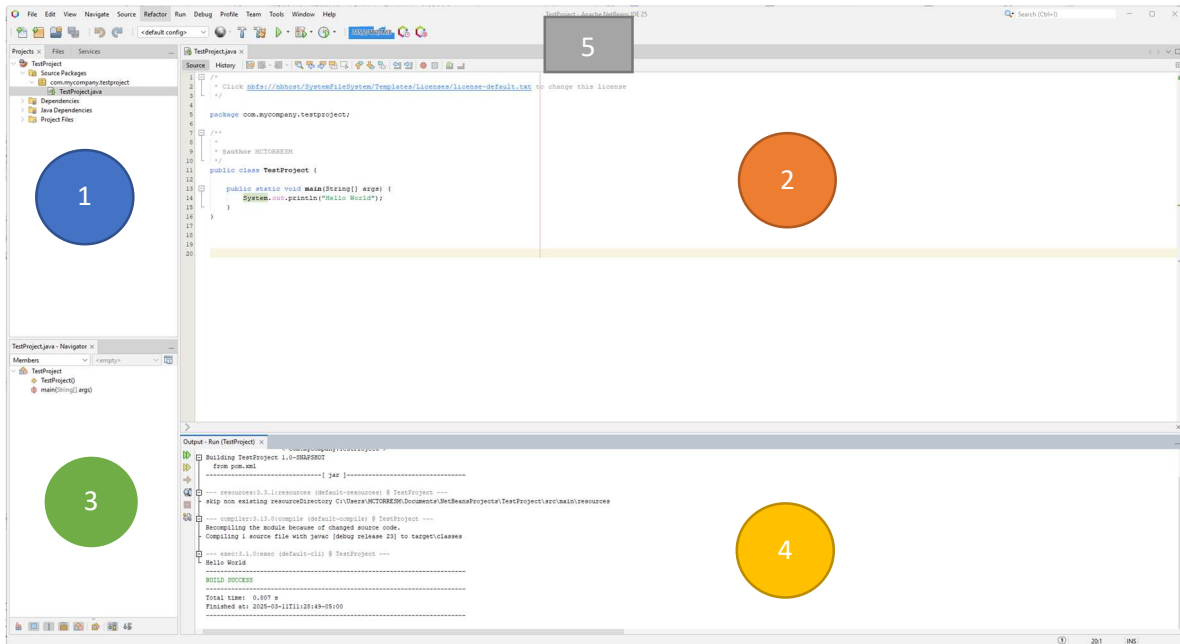


Figura 2. Interfaz gráfica por defecto de NetBeans.

Mi primer programa en NetBeans

Configurando un proyecto: para empezar a trabajar con Java en NetBeans es necesario configurar un proyecto. En el menú **File** seleccione la opción **New Project**. En la ventana emergente seleccione la opción de **Java Application** (ver Figura 3). Una vez seleccionado vamos a la siguiente pantalla con el botón **Next**. En la nueva ventana vamos a asignar el nombre de nuestro proyecto y finalizar la configuración. Una vez creado el proyecto, en la ventana principal se mostrarán los directorios automáticamente creados para nuestro primer proyecto.

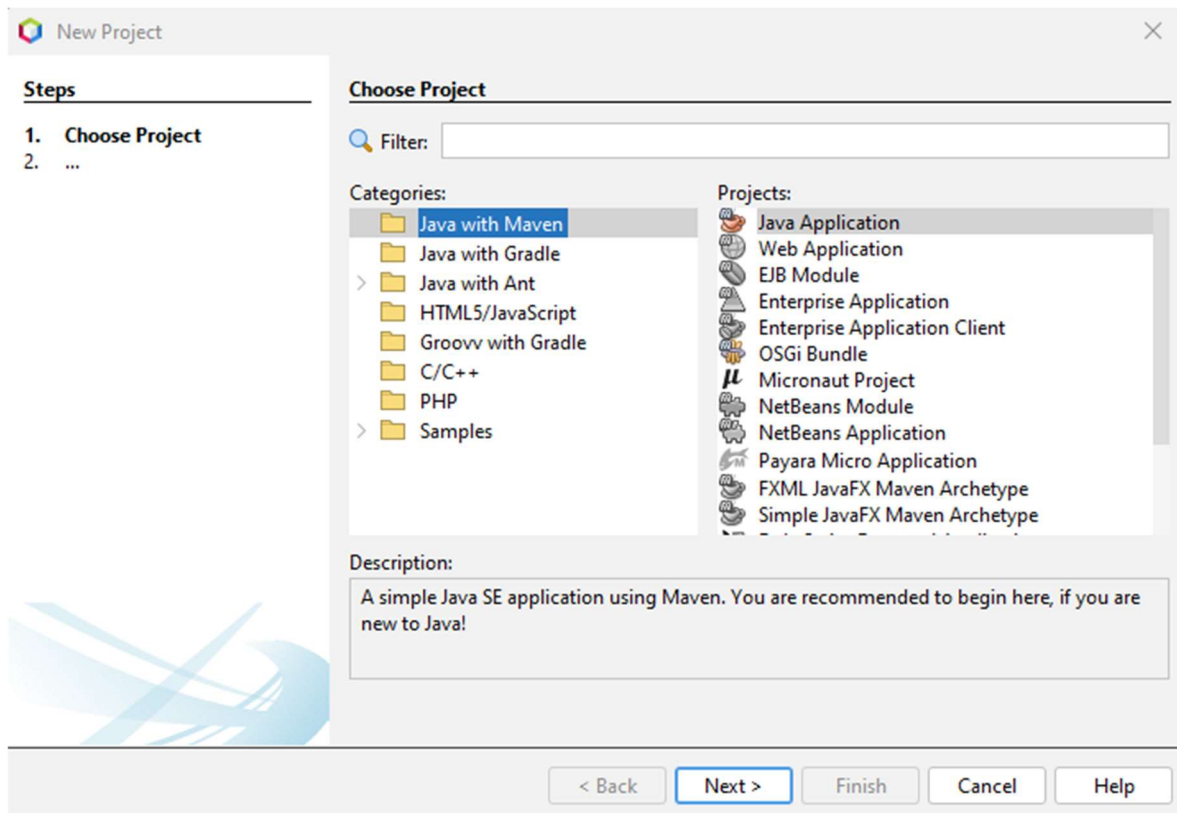


Figura 3. Configuración de un proyecto en NetBeans.

Una vez completado la configuración del proyecto, nos aparecerá el código base para “Hello world” (ver Figura 4).

```
package com.mycompany.testproject2;

/**
 *
 * @author MCTORRESM
 */
public class TestProject2 {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Figura 4. Hello World

Compilación y prueba: Una vez finalizada la codificación, vamos al menú superior donde encontraremos las opciones para correr el programa. La primera vez que corremos el programa nos solicitará seleccionar el archivo que contiene la clase principal.

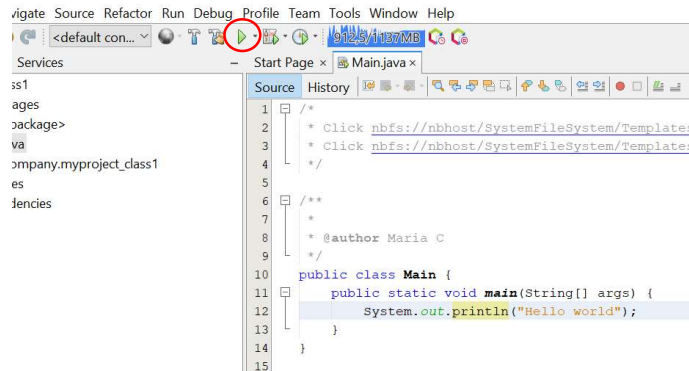


Figura 5. Corriendo el programa.

En la consola podremos revisar si existió algún tipo de error de compilación, así como las salidas del programa.

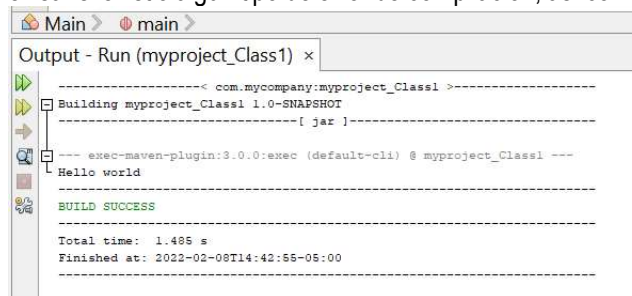


Figura 6. Consola de NetBeans.

Actividades a desarrollar durante el laboratorio

Vamos a realizar diferentes pruebas para conocer un poco más del lenguaje de programación JAVA:

- Declaración de variables y tipos de variables
- Casting
- Condicionales
- Ciclos FOR y WHILE
- Arreglos
- Métodos
- Lectura de datos en consola

Nota: Una vez finalizada la práctica, copia de los códigos realizados en clase estarán disponibles en el Moodle.

Actividades a desarrollar por los estudiantes:

Parte 1. Diseño e implementación de un programa en Java – Ejercicio practico

Vamos a diseñar un programa que reciba dos arreglos de enteros (A y B) y realice algunas operaciones básicas. El programa debe permitir:

1. Recibir por consola el tamaño de los arreglos; ambos arreglos tendrán la misma longitud y debe ser menor o igual a 12; si se ingresa un valor más alto, se debe pedir nuevamente el tamaño del arreglo.
2. Para el arreglo A, se debe pedir por consola los valores del arreglo.
3. Para el arreglo B, se debe generar de forma aleatoria los valores enteros del arreglo en un rango del 1 a 20.
4. Realizar la suma de los arreglos $A + B$, mostrando en la consola el arreglo resultante.
5. Realizar el calculo del producto punto entre los arreglos A y B, mostrando en la consola el resultado.
6. Calcular la suma de los enteros pares del arreglo A.
7. Calcular el promedio de los valores del arreglo B.
8. Construir un arreglo C el cual es la unión de los elementos de A y B, considerando que $A = [a_0, a_1, \dots, a_{n-1}]$ y $B = [b_0, b_1, \dots, b_{n-1}]$, el arreglo C debe estar organizado tal que $C = [a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}]$
9. Determinar el valor máximo en el arreglo C y su posición en el arreglo.

Parte 2. Diseño e implementación de un programa en Python – Ejercicio practico

Repita el ejercicio de la parte 1, implementando la solución en Python. Se recomienda el uso de Visual Studio Code para esta segunda parte.

Entrega de actividades parte 1 y parte 2: los programas funcionales tanto en JAVA como en Python deben ser sustentados durante la sesión del 24 de abril.