Utilizando Java

¡Comencemos!

¿Qué aprenderás?

- Comprender las formas de trabajar en Java para ejecutar aplicaciones.
- Crear un proyecto para ejecutar bajo la Máquina Virtual de Java instalado.

Introducción

Antes de comenzar debemos instalar Java en nuestro equipo, para eso se recomienda revisar el documento **Instalando Java - Lectura complementaria** ubicado en "Material Complementario".

A continuación, tendremos el primer acercamiento a programar con Java, donde revisaremos cómo trabajar con Java ya sea por terminal (CMD) o utilizando Eclipse para crear nuestro primer código.

¡Vamos con todo!

Formas de trabajar en Java

Al trabajar pequeños trozos de código no existe gran diferencia entre trabajar con un IDE o un editor de texto, sin embargo, en programas un poco más complejos, es mucho más eficiente contar con las herramientas que provee el IDE, optimizando el tiempo de desarrollo.

Compilando por terminal (cmd)

Al crear un programa en Java, este deberá tener el formato de archivo Nombre.Java.

En el terminal, debemos ir a la ruta donde está nuestro programa (usando el comando `cd /"ruta_de_archivo"`) y ejecutar el comando `Javac` y luego el nombre del programa.

```
Javac Nombre.Java
```

Luego, para ejecutarlo, se utiliza el comando `Java`.

```
Java Nombre
```

Y éste deberá realizar las acciones del programa creado.

Utilizando Java en Eclipse IDE

Al ejecutar por primera vez se desplegará la siguiente pantalla, en la cual debemos seleccionar dónde estará la ubicación de nuestro espacio de trabajo (es decir, los programas que vayamos creando), y damos clic a Launch.

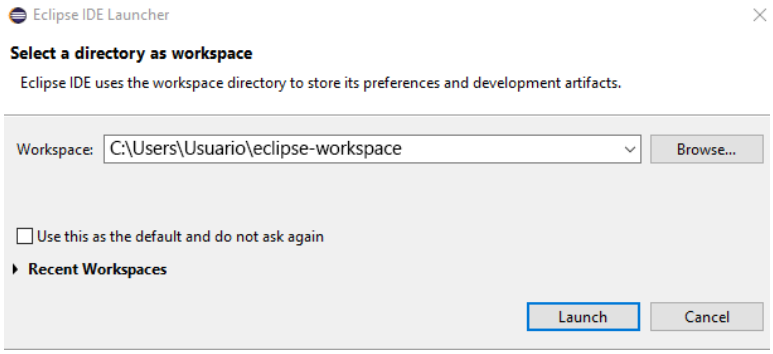

Imagen 1. Seleccionar ubicación del espacio de trabajo.
Fuente: Desafío Latam.

Luego aparecerá la pantalla con las distintas acciones que se pueden realizar con Eclipse, donde crearemos un nuevo proyecto Java.

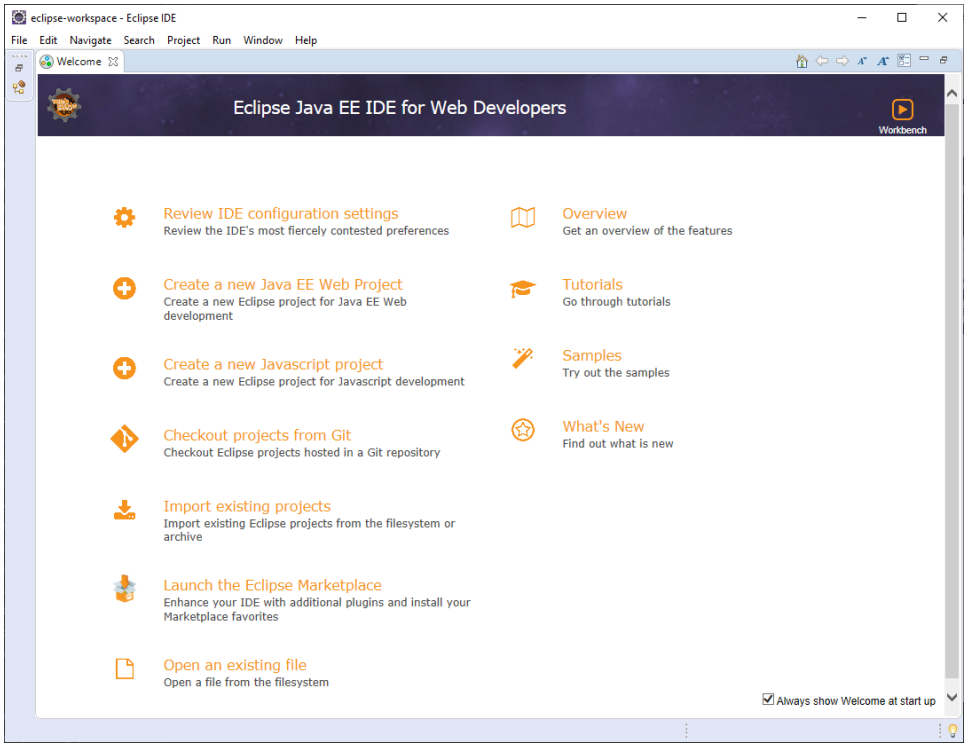

Imagen 2. Crear un nuevo proyecto Java.
Fuente: Desafío Latam.

Creando el primer programa en Java

Ya hemos aprendido algoritmos, específicamente pseudocódigo y diagramas de flujo. Pues, lo aprendido lo llevaremos a la práctica comenzando con un primer acercamiento al lenguaje Java.

Requerimientos previos

1. Versión de Java JRE y JDK 1.8 (Lectura complementaria: Instalando Java).
2. Eclipse IDE instalado.
3. Conocimiento en utilización de Java.

Ejercicio guiado: Nuestro primer proyecto

Para crear nuestro primer programa en Java, se deben seguir los siguientes pasos:

1. Crear un nuevo proyecto en nuestro IDE (Entorno de Desarrollo Integrado) de preferencia. Para nuestro caso será **Eclipse**.
2. Crear y organizar el proyecto en paquetes o package.
3. Crear la clase main o principal, que será la clase que ejecutará la lógica del programa.

Al momento de dar clic al nuevo proyecto de Java, se desplegará una ventana donde debemos escribir el nombre del proyecto, elegir la ubicación y la versión de Java que se utilizará.

Para cada proyecto que creamos, podremos elegir indistintamente qué versión utilizar y no tendremos que estar preocupándonos por la versión que se está ejecutando en el terminal.

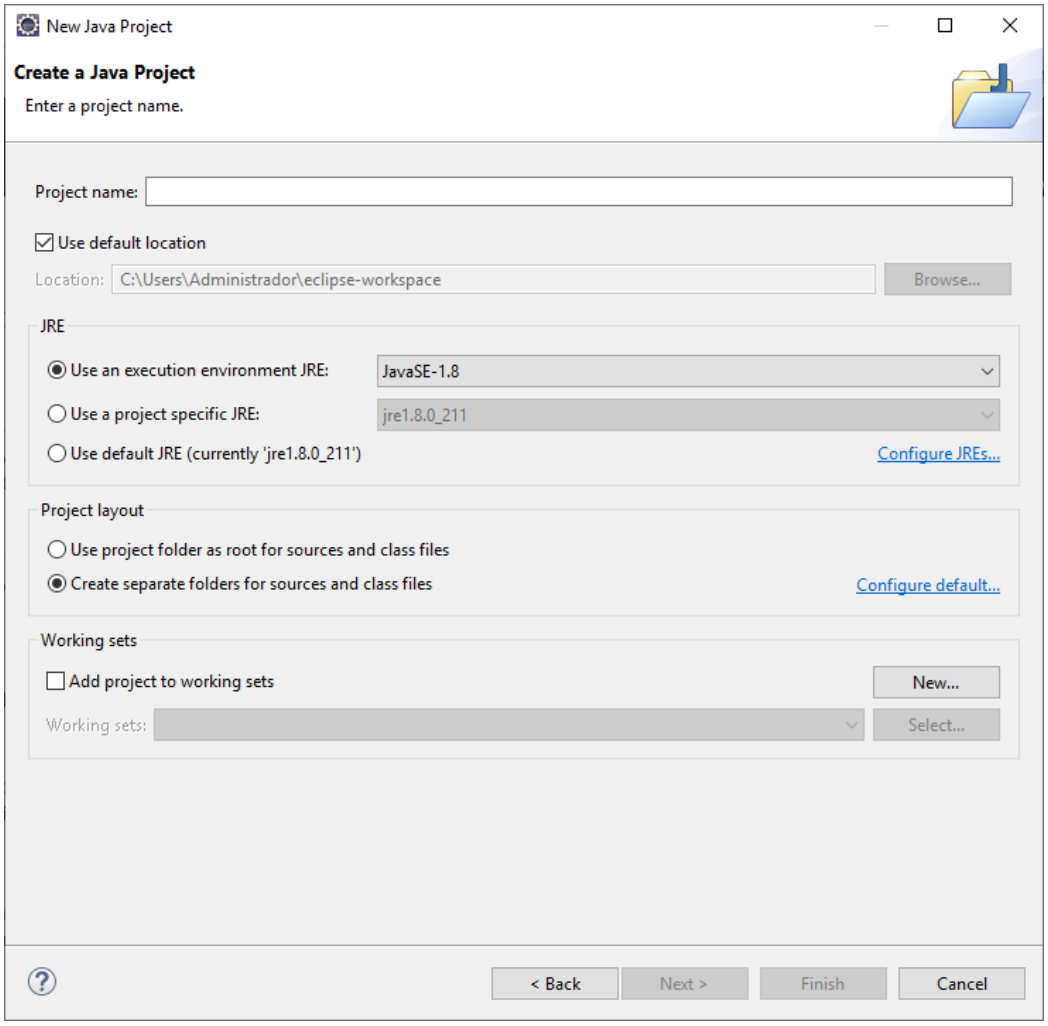

Imagen 3. Creando un nuevo proyecto en eclipse.
Fuente: Desafío Latam.

Al dar clic en el botón finalizar, aparecerá un diálogo que nos preguntará si queremos asociar nuestro proyecto con la perspectiva de Java, la cual aceptaremos.

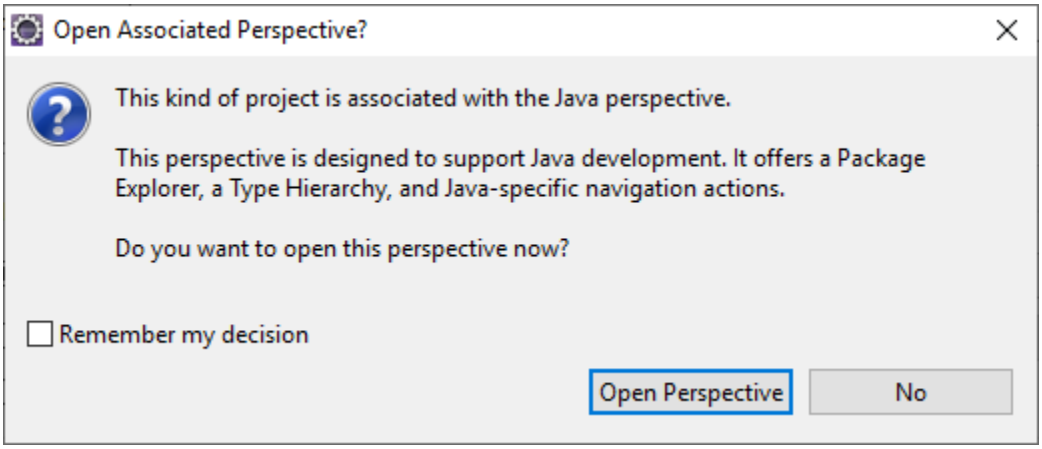

Imagen 4. Asociar proyecto con perspectiva de Java.
Fuente: Desafío Latam.

Revisando el espacio de trabajo

Del lado izquierdo tenemos el explorador de paquetes, es decir, donde tendremos nuestros ficheros a utilizar en el programa y estará el código Java.

En la zona media, podremos ver el contenido de cada uno de los ficheros donde escribiremos el código.

En la zona inferior se mostrarán los errores y cuando se ejecuta el programa. Debemos agregar la consola para ver la salida que obtendremos de nuestras ejecuciones.

Para ello debemos ir al menú superior, Window -> Show View -> Console.

Continuemos

Ya creamos la base de nuestro proyecto, ahora debemos crear el fichero donde escribiremos el código como tal.

1. En la raíz del proyecto, haremos clic derecho sobre src -> New -> Package, al cual le colocaremos el nombre: cl.desafiolatam.

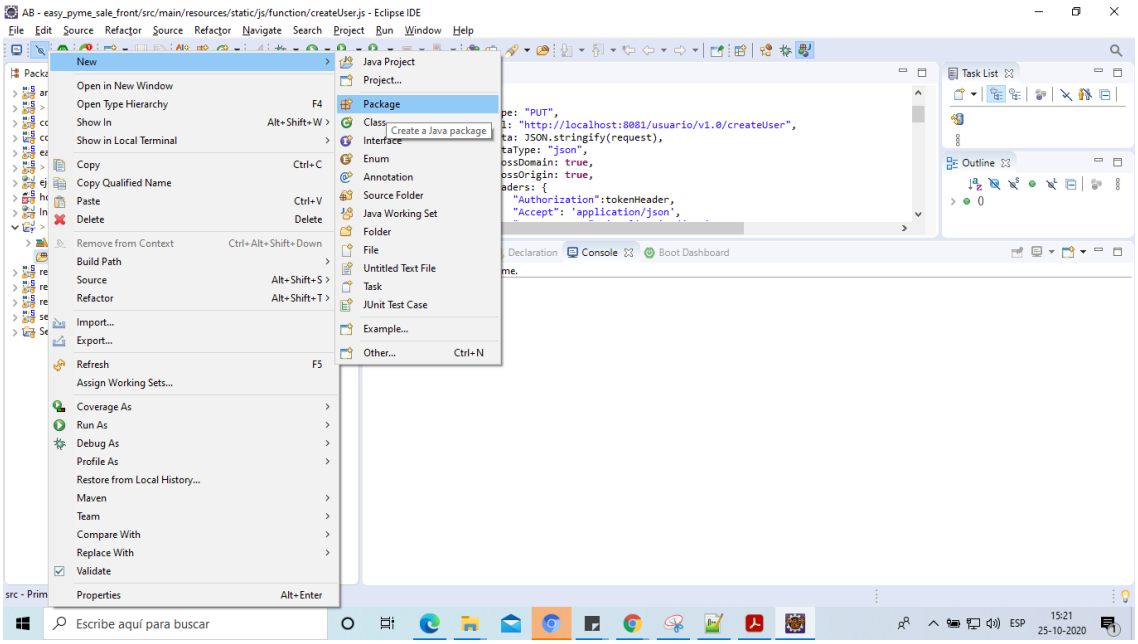

Imagen 5. Creación de un nuevo package (paquete) en eclipse.
Fuente: Desafío Latam.

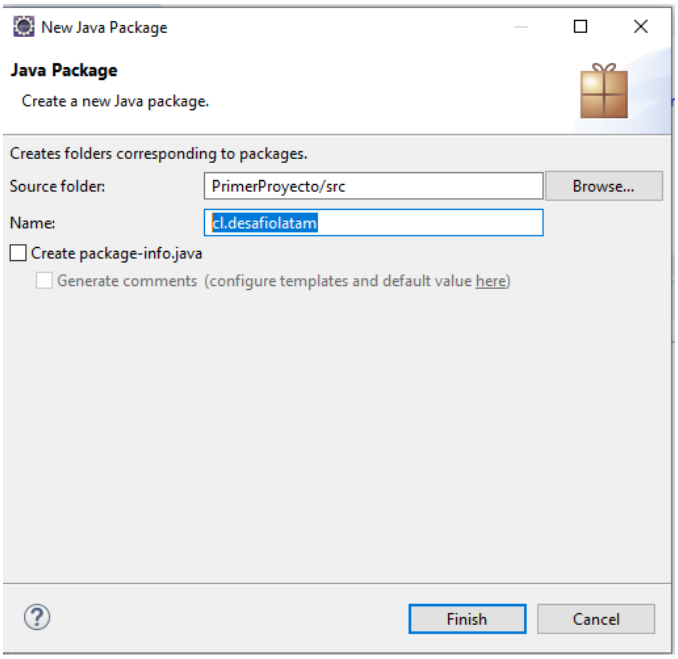

Imagen 6. Creación de un nuevo package (paquete) en eclipse.
Fuente: Desafío Latam.

2. Haremos clic derecho sobre el package cl.desafiolatam -> New -> class, a la cual le pondremos el nombre MiPrimerPrograma, donde aparecerán varias opciones. De momento marcaremos public static void main (String[] args) y finalizar.

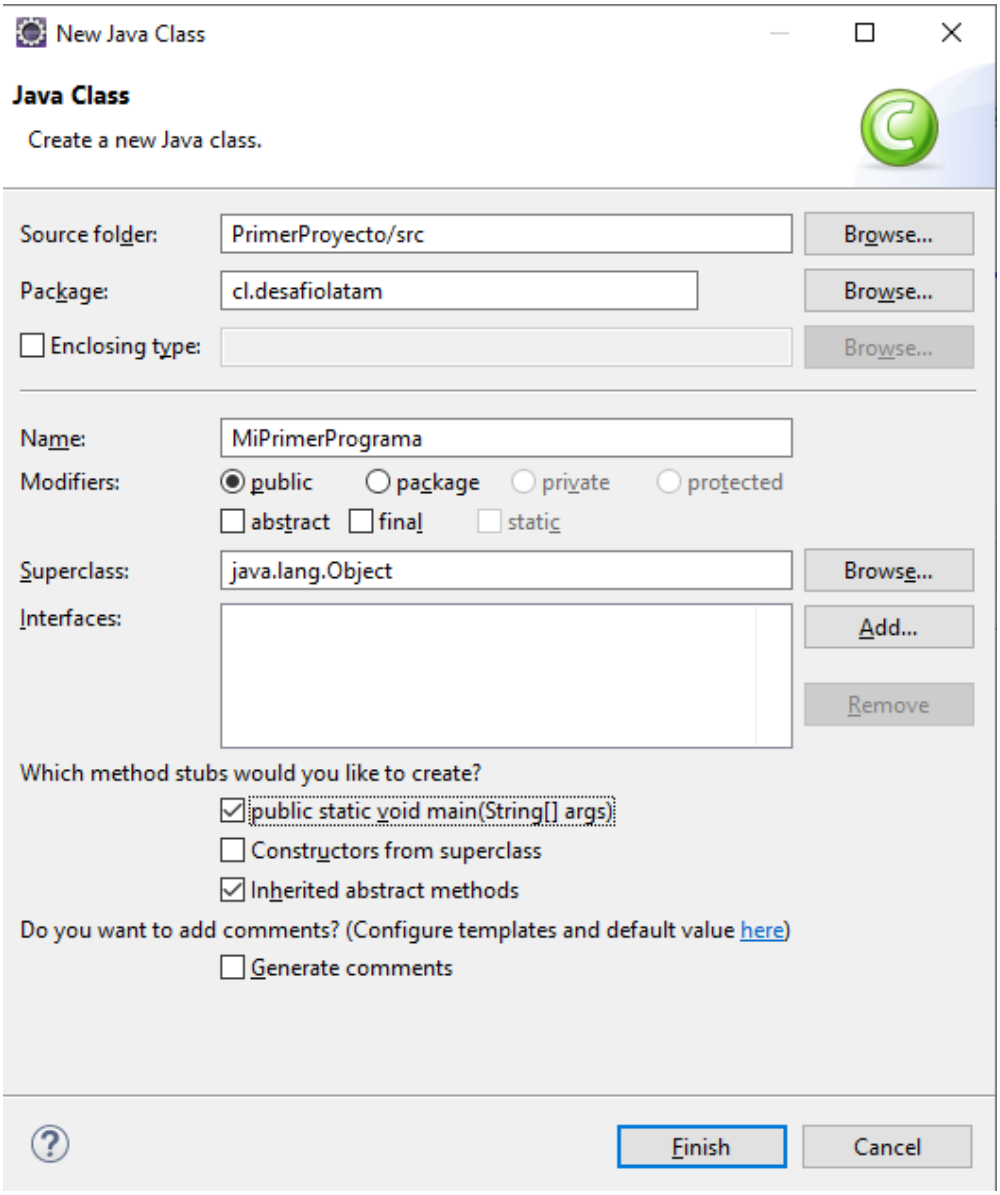

Imagen 7. Creación de clase main (Clase principal) en eclipse.
Fuente: Desafío Latam.

¡Hola, Mundo! En Java

Ahora que tenemos nuestra clase creada, crearemos nuestro primer "Hola Mundo!"

Para hacer ello, usaremos el método System.out.printf que muestra por pantalla lo que necesitemos:

```
package cl.desafiolatam;

public class MiPrimerPrograma {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		System.out.printf("Hola Mundo!\n");
	}
}
```

Ahora para ejecutarlo, hacemos clic derecho sobre el paquete, y seleccionamos **Run As Java Application**.

En la zona inferior vemos que aparece una consola, donde dice "Hola Mundo!", qué fue lo que escribimos.

Al final del mensaje se colocó `\n` que corresponde a un salto de línea. Más adelante profundizaremos más en cuanto a los formatos de printf.

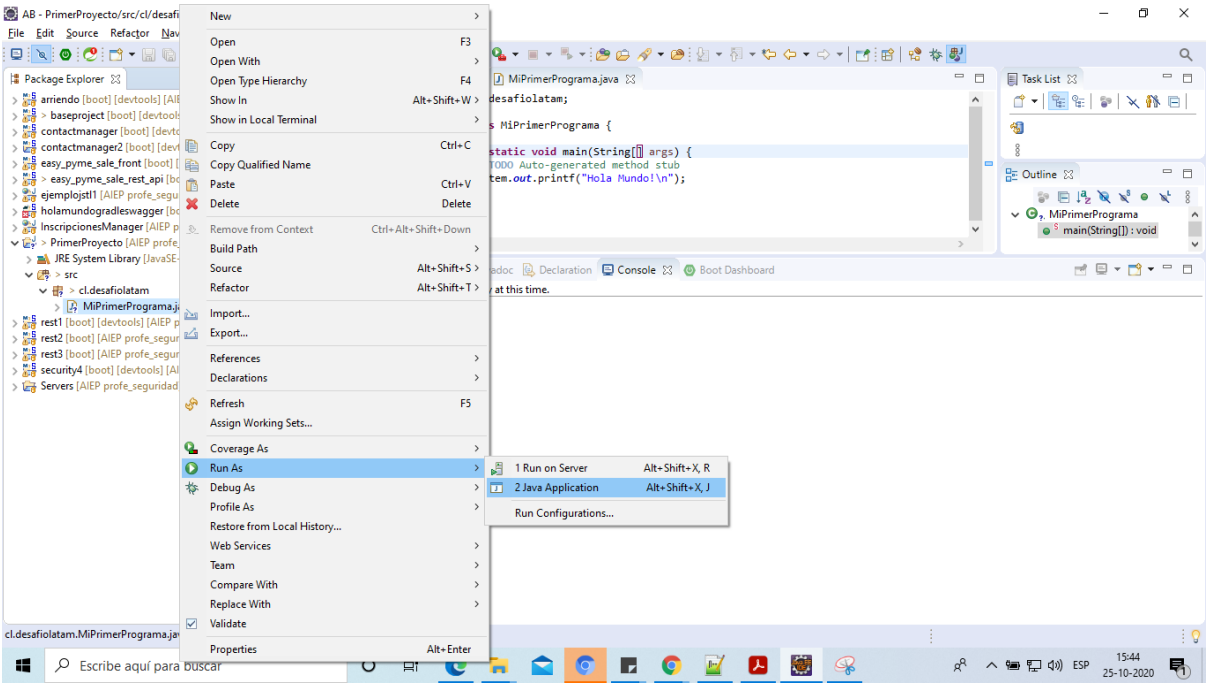

Imagen 8. Ejecutar programa Java en Eclipse IDE.
Fuente: Desafío Latam.

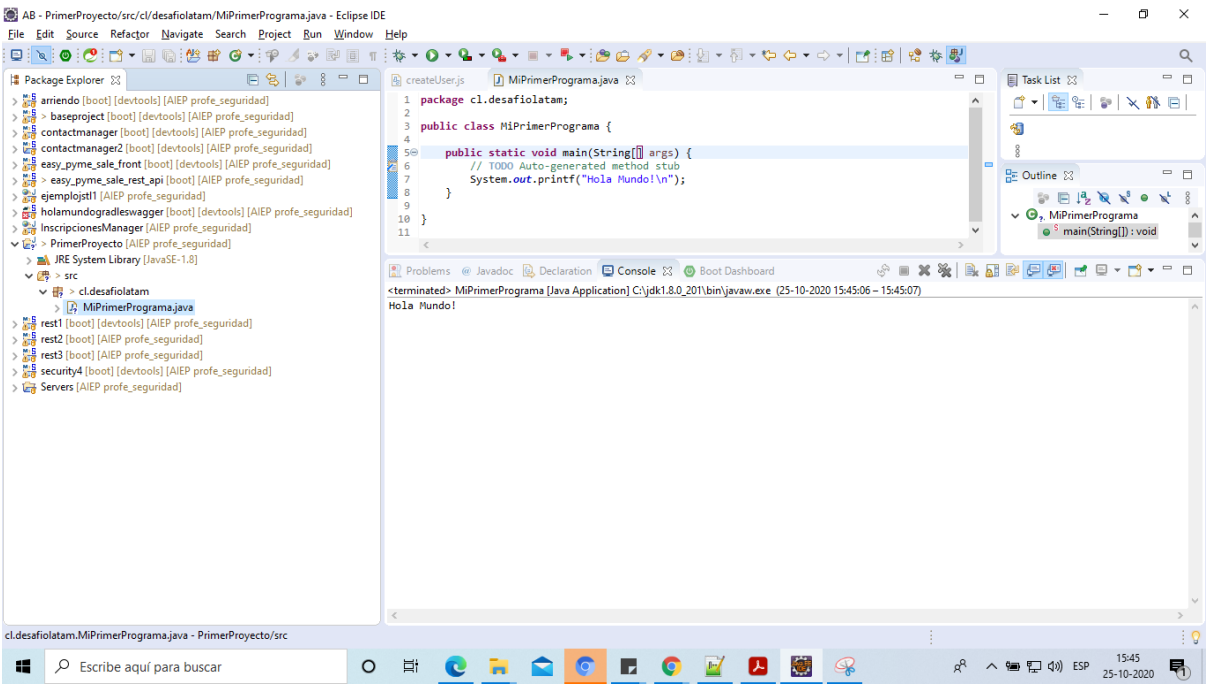

Imagen 9. Ejecución de nuestro primer programa en Java Hola Mundo!.
Fuente: Desafío Latam.

Java es un lenguaje que se debe aprender como si fuera cualquier otro lenguaje, por ejemplo, el Inglés, Francés, etc. Al ser un "lenguaje" tiene ciertas formas gramaticales que debemos aprender en base a la práctica constante. Se recomienda leer la documentación web complementaria, específicamente los siguientes apartados:

- Introducción a Java.
- Conceptos básicos en Java.
- Operadores Java.
- Sentencias de control.

Resumen

Para crear código en Java, debemos crear nuestra clase principal (más adelante profundizaremos en qué es una clase), la cual contendrá la función principal llamada main.

Cada clase que eventualmente creemos debe estar contenida dentro de un paquete (que vendría a ser como una carpeta de Java).

Todo el código que vayamos a escribir de ahora en adelante siempre estará dentro del main, que es la función que llamará al ejecutar el programa.

Observación con el nombre de las clases y paquetes

Si notaron, al definir el nombre de la clase se escribió la primera letra con mayúscula y, cada vez que empezaba una palabra nueva, vuelve a ser mayúscula: MiPrimerPrograma.

En el caso del paquete, debe ser todo con minúscula: cl.desafiolatam.

Esto se debe a que existe una convención para escribir el código en Java, no es mandatorio seguirlo estrictamente, pero se considera una buena práctica seguirla.

Algoritmos, diagramas de flujos e implementación en Java

Para controlar el flujo y la lógica en cualquier programa en Java, debemos implementar algunos de los conceptos básicos que existen en cualquier lenguaje de programación, estas son las variables, operadores y estructuras de control (condiciones). Por ejemplo, se requiere crear un programa que calcule la raíz cuadrada de un número:

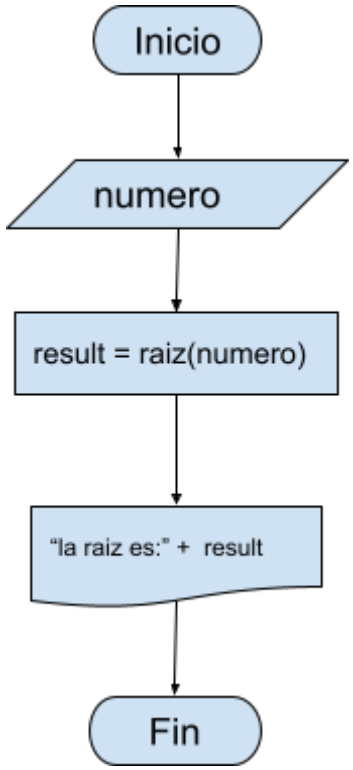

Imagen 10. Diagrama de flujo del cálculo de la raíz cuadrada de un número.
Fuente: Desafío Latam.

En el diagrama de flujo expuesto en la imagen 5, nótese que la operación de cálculo se representa mediante la palabra textual en español "raíz", esto es debido a que el pseudocódigo o los algoritmos representados mediante diagrama de flujo, deben ser autoexplicativos y tener lógica, pero no representar la implementación en un lenguaje en particular. Además notar el símbolo para impresión "la raíz es: + result". Este es el símbolo que representa una impresión por pantalla o bien en papel.

La implementación del flujo anterior en Java sería la siguiente:

```
package cl.desafiolatam;

import java.util.Scanner;

public class MiPrimerPrograma {
	public static void main(String[] args) {
		System.out.printf("Ingrese un número: "); /*Se imprimirá
Ingrese un número: por pantalla */
		Scanner sc = new Scanner(System.in); /* La clase Scanner se
utiliza para leer por consola un valor ingresado por el usuario */
		long numero = sc.nextLong(); /* numero, recibe el valor
ingresado por el usuario. nextLong devuelve un tipo de dato primitivo long
*/
		double result = Math.sqrt(numero); /* result, recibe el
resultado del método sqrt de la clase Math, la cual calcula la raíz
cuadrada de un número */
		System.out.printf("La raíz cuadrada es: %f", result); /*
finalmente, se imprime el resultado por pantalla con System.out */
	}

}
```

La diferencia al traspasar el diagrama de flujo al código en lenguaje Java es precisamente el lenguaje. Debemos saber, por ejemplo, que debemos ocupar el objeto `Scanner` para leer una variable por pantalla; `long` y `double` son tipos de datos primitivos; la clase `Math` nos ayuda a realizar operaciones matemáticas complejas, etc.

Lo importante es la lógica del flujo y del programa más que el lenguaje o la técnica, ya que debemos considerar qué documentación para lo que queramos hacer hay mucha, solo debemos saber cómo implementarla.