



El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

## CICLO II: Programación Básica en Java

Misión  
TIC 2022





El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

# Sesión 8: Introducción a Java

Programación Orientada a Objetos (POO)





# Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Reconocer el uso permitido y no permitido de los métodos y variables con modificadores de acceso public y private
2. Definir y construir un programa que muestre los usos de los modificadores public y private
3. Explicar la declaración de main como static y aplicarlo en un contexto práctico.
4. Explicar la aplicación conjunta de las palabras claves public y static en un método.

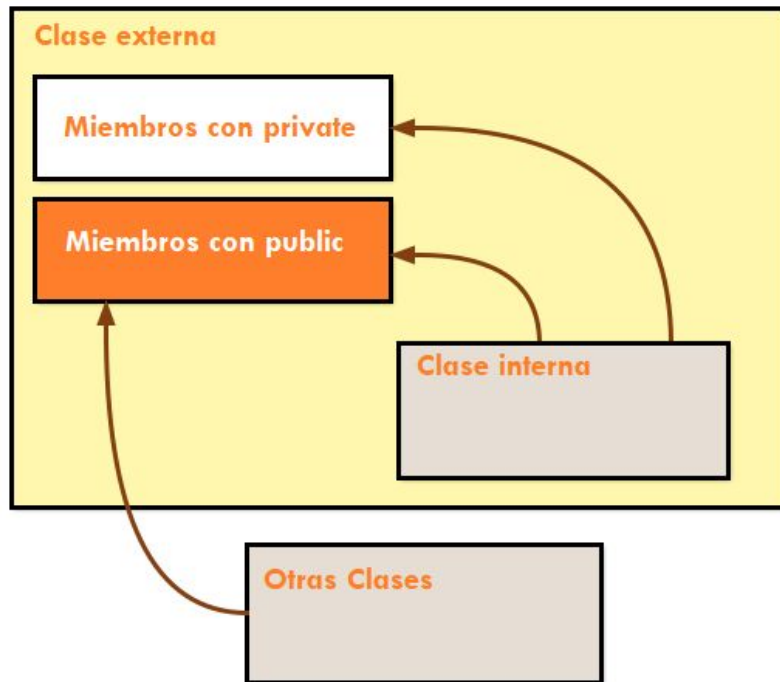


# Modificadores de acceso java: public, private

- Con los modificadores de acceso se puede aplicar el encapsulamiento. Buscando la forma de controlar el acceso a los datos de un objeto o instancia.
- Los modificadores se usan para dar seguridad a las aplicaciones limitando el acceso a diferentes atributos, métodos, constructores asegurando una “ruta” específica para acceder a la información.
- Con modificadores de acceso se asegura que un valor no será modificado de forma incorrecta. Para lograrlo se utiliza el acceso a los atributos con los métodos get y set.



# Modificadores de acceso java: public, private





# Modificadores de acceso java: public, private

Visibilidad	Public	Protected	Default	Private
Desde la misma Clase	SI	SI	SI	SI
Desde cualquier Clase del mismo Paquete	SI	SI	SI	NO
Desde una SubClase del mismo Paquete	SI	SI	SI	NO
Desde una SubClase fuera del mismo Paquete	SI	SI, a través de la herencia	NO	NO
Desde cualquier Clase fuera del Paquete	SI	NO	NO	NO



# Modificadores de acceso java: private

- Este modificador sólo permite acceder desde la misma clase.
- En el caso de un método solo se podrá llamar de forma interna.
- Para los atributos, se crearán métodos públicos GET y SET.



# Modificadores de acceso java: private

```
public class Ejemplo1
{
    // Atributo y sus metodos
    private int atributo;

    public int getatributo()
    {
        return atributo;
    }

    public void setAtributo(int atributo)
    {
        this.atributo = atributo;
    }

    // Metodo
    private int metodo(){
        //codigo
    }
}
```





# Modificadores de acceso java: public

- Es el más permisivo.
- Cualquier clase tendrá acceso a este modificador sin importar el paquete o de dónde provenga.



# Modificadores de acceso java: public

```
public class Ejemplo2
{
    public static int atributo = 0;

    public static void metodo()
    {
        System.out.println("Metodo Publico");
    }
}
```

```
package otro.paquete;
import app.ejemplo2.Ejemplo2;
```

```
public class OtraExterna
{
    public static void main(String[] args)
    {
        System.out.println(Ejemplo2.atributo);
        Ejemplo2.metodo();
    }
}
```



# ¿Por qué el método main se declara como static?

Al ejecutarse la máquina virtual de Java (JVM) con el comando java, ésta intenta llamar al método main de la clase que se le especifica; Hasta este momento no se ha creado ningún objeto de esa clase. Al declararse main como static, la JVM puede llamar a main sin necesidad de crear una instancia de la clase.

Cuando se ejecuta la aplicación, se especifica el nombre de la clase como un argumento para el comando java:

```
java NombreClase argumento1 argumento2 ...
```

La JVM carga la clase especificada por NombreClase y utiliza el nombre de esa clase para llamar al método main.



El futuro digital  
es de todos

MinTIC

**UN** UNIVERSIDAD  
**DEL NORTE**

Vigilada Mineducación

**¡GRACIAS**  
**POR SER PARTE DE**  
**ESTA EXPERIENCIA**  
**DE APRENDIZAJE!**



**Misión**  
**TIC 2022**