

## Setters & Getters

Los Setters & Getters son métodos de acceso lo que indica que son siempre declarados públicos, y nos sirven para dos cosas:

Setters: Del Inglés Set, que **significa establecer**, pues nos sirve para asignar un valor inicial a un atributo, pero de forma explícita, además el Setter nunca retorna nada (Siempre es void), y nos permite dar acceso público a ciertos atributos que deseemos que se puedan modificar.

Getters: Del Inglés Get, que **significa obtener**, pues nos sirve para obtener (recuperar o acceder) el valor ya asignado a un atributo y utilizarlo para cierto método.

Entonces veamos en un ejemplo, si tenemos una Clase de nombre Estudiante, cómo serían sus Setters & sus Getters?

```
public class Estudiante
{
    private String nombre;
    private String apellido;
    private String correo;

    public Estudiante(String nombre,String apellido,String correo){
        this.nombre=nombre;
        this.apellido=apellido;
        this.correo=correo;}

    public String getNombre()
    {
        return nombre;
    }
    public void setNombre(String nombre)
    {
        this.nombre = nombre;
    }

    public String getApellido()
    {
        return apellido;
    }
    public void setApellido(String apellido)
    {
        this.apellido = apellido;
    }

    public String getCorreo()
    {
        return correo;
    }
    public void setCorreo(String correo)
    {
        this.correo = correo;
    }
}
```

```

    }
}

public class Prueba
{
    public static void main(String args[])
    {
        Estudiante estudiante1 = new
Estudiante("Mari", "Pascar", "maria.pascar@epn.edu");
        // Ingreso los datos con los Setter a la nueva instancia de
        // Estudiante, estos datos pueden ingresarse desde teclado
        // o a través de una GUI
        estudiante1.setNombre("María");
        estudiante1.setApellido("Paucar");
        estudiante1.setCorreo("maria.paucar@epn.edu");

        // Para obtener los datos uso los Getter
        // y los podemos imprimir en Consola o a una GUI
        System.out.println(estudiante1.getNombre());
        System.out.println(estudiante1.getApellido());
        System.out.println(estudiante1.getCorreo());
    }
}

```

## Ventajas

Aquellos métodos y atributos que definamos como public (públicos) serán visibles desde cualquier otra clase. Por el contrario, los métodos y atributos que definamos como private (privados) estarán encapsulados y solo podrán ser invocados y manipulados dentro de la misma clase.

**Los sets y gets son la forma de acceder a atributos de una clase.** Generalmente, se usan con los atributos privados, ya que a los públicos se puede acceder directamente sin tener que acudir a ellos.

La protección de las variables miembro que utiliza métodos de acceso permite a los programadores modificar fácilmente el código interno sin cambiar la forma en que se utiliza la clase por otros.

## Convención

La convención aceptada es la de nombrar métodos get y set para la variable de miembro asociada (por ejemplo, "getName" y "setName", asociado a la variable "nombre"), con el prefijo get o set seguido del nombre del atributo **con la primera letra de cada palabra interna en mayúsculas y no se permiten caracteres especiales.**

No todas las variables miembro pueden tener métodos de acceso asociados. Estos métodos se escriben solamente para los valores a los que se deben acceder externamente.