

Nombre Alumno/s: Reyes, Kevin Josue

DNI: 4593556

Nombre Profesor: Aldo Mettini

Grupo Laboratorio: 3

TP: 2

Fecha de entrega: 21-08-2025

En la serie del tp 2 se pusieron en práctica conceptos del paradigma de objetos como la abstracción porque debimos crear clases de objetos de la vida real, pero solo tomando elementos necesarios para el dominio del problema por ejemplo en el caso de la clase Persona:

```
public class Persona {  
    //Atributos de la clase  
    private int nroDni;  
    private String nombre;  
    private String apellido;  
    private int anioNacimiento;
```

Se ven que los atributos son dni, nombre, apellido, año de nacimiento. A una persona no la componen sólo esos elementos pero para la resolución del problema que nos propone el práctico sólo son necesarios esos elementos como atributos.

También se vio el encapsulamiento y doble encapsulamiento. En el caso de el encapsulamiento vemos que sus atributos y setter son privados es decir que solamente la clase puede utilizar sus funciones si un objeto externo lo quiere usar no podría:

```
public class Persona {  
    //Atributos de la clase  
    private int nroDni;  
    private String nombre;  
    private String apellido;  
    private int anioNacimiento;
```

```
//Setters
```

```
private void setDNI(int p_dni) {  
    nroDni = p_dni;  
}  
private void setNombre (String p_nombre) {  
    nombre = p_nombre;  
}  
private void setApellido (String p_apellido) {  
    apellido = p_apellido;  
}  
private void setAnioNacimiento (int p_anio) {  
    anioNacimiento = p_anio;  
}
```

Pero tampoco todos los métodos de la clase acceden a los atributos libremente, si no que lo hacen por medio de los accessor específicamente de los setters en el caso del método constructor:

```
//Metodo especial: Constructor
public Persona (int p_dni, String p_nombre, String p_apellido, int p_anio) {
    setDNI(p_dni);
    setNombre (p_nombre);
    setApellido (p_apellido);
    setAnioNacimiento (p_anio);
}
```

Y en el caso que quieran acceder a los atributos para mostrarlos lo hacen mediante los accessor getters que son de acceso público:

```
public int edad() { //Retorna la edad de la persona
    Calendar fechaHoy = new GregorianCalendar();
    int anioHoy = fechaHoy.get(Calendar.YEAR); //Obtiene el año actual
    return (anioHoy - getAnioNacimiento());
}

public String nomYApe() { //Retorna nombre y apellido de la persona
    return getNombre() + ", " + getApellido();
}

public String apeYNom() { //Retorna apellido y nombre de la persona
    return getApellido() + ", " + getNombre();
}

public void mostrar() {
    System.out.println("Nombre y Apellido: " + nomYApe());
    System.out.println("DNI: " + getDNI() + "\t" + "Edad: " + edad() + " años");
}
```

En esos métodos que se usan para mostrar la información por pantalla son usados los getters para obtener los atributos.

También se trabajó con métodos sobrecargados, en el caso de la clase Laboratorio tiene dos constructores uno en cual recibe por parámetros todos los estados con los cual se instancian sus atributos, y en el caso del otro sólo recibe estados para los atributos nombre, domicilio y teléfono.

```
//Metodos constructores
public Laboratorio (String p_nombre, String p_domicilio, String p_telefono,
                    int p_compraMin, int p_diaEnt) {
    setNombre (p_nombre);
    setDomicilio (p_domicilio);
    setTelefono (p_telefono);
    setCompraMinima (p_compraMin);
    setDiaEntrega (p_diaEnt);
}

public Laboratorio (String p_nombre, String p_domicilio, String p_telefono) {
    setNombre (p_nombre);
    setDomicilio (p_domicilio);
    setTelefono (p_telefono);
}
```