Nombre Alumno/s: Scetti Santiago **DNI:** 43752065
Varela Franco Martín 44280211

Nombre Profesor:

Grupo Laboratorio: "3" **TP:** 2 **Fecha de entrega:** 25/08/24

Introducción:

A lo largo de este informe, se detallarán los conceptos teóricos desarrollados en las clases de teoría junto a la puesta en práctica del trabajo práctico correspondiente. El ensayo busca responder los fundamentos teóricos en relación con los conceptos de *abstracción, encapsulamiento, ocultamiento, visibilidad, objeto, clase, doble encapsulamiento, mensaje, método, protocolo, firma, comportamiento.*

Objeto: las instancias de Persona creadas en el ejecutable representan objetos en la programación orientada a objetos.

```
public class EjecutablePersona {
    public static void main(String[] args) {
        // Definiendo constantes para las personas
        int DNI1 = 44280211;
        String NOMBRE1 = "Franco";
        String APELLIDO1 = "Varela";
        int ANIO_NACIMIENTO1 = 1990;
```

Clase: Es una ***abstracción*** que describe un grupo de instancias con propiedades (atributos) comunes, comportamiento (operaciones) común, relaciones comunes con otros objetos y (lo que es más importante) una semántica común. Por ejemplo: la clase "Alumno", que es una abstracción del mundo real a lo digital. Definimos sólo los atributos y métodos esenciales para el problema, ocultando los detalles innecesarios.

```
public class Persona {
```

Encapsulamiento, ocultamiento y visibilidad: Se utilizó encapsulamiento al hacer privados los atributos de las clases y proporcionar métodos de acceso controlados (getters y setters), lo que protege los datos de modificaciones no deseadas.

```
public class Persona {

    private int nroDni;
    private String nombre;
    private String apellido;
    private int anioNacimiento;
```

Doble encapsulamiento: El acceso a estos atributos está controlado a través de métodos privados (setters) que encapsulan la lógica de modificación. Estos métodos no son accesibles desde fuera de la clase, lo que añade una capa extra de control sobre cómo se modifican los atributos.

```
//Getters
public int getDNI() {
    return nroDni;
}

//Setters
private void setDNI(int p_dni) {
    this.nroDni = p_dni;
}
```

Comportamiento, mensajes y métodos: El comportamiento indica qué sabe hacer el objeto, cuando un objeto envía un mensaje a otro, el objeto receptor responde activando el método asociado a ese mensaje. En el *main* de nuestro EjecutablePersona creamos un objeto llamado p1 de tipo Persona, y enviamos un mensaje para que se active el método que buscamos y poder visualizar el comportamiento del objeto:

```
// Creación de la primera persona
Persona p1 = new Persona(DNI1, NOMBRE1, APELLIDO1, ANIO_NACIMIENTO1);
p1.mostrar();  // Muestra la información de persona1
```

Protocolo: Al conjunto de mensajes que un objeto puede responder se lo denomina *protocolo*. En la clase Persona, el protocolo estaría formado por los siguientes métodos públicos:

getDNI(), getNombre(), getApellido(), getAnioNacimiento(), edad(), nomYApe(), apeYNom(), mostrar()

Firma: La constituye el ***nombre*** del método y ***cantidad, tipo y orden*** de los ***parámetros***.

La firma del constructor de la clase persona:

```
public Persona(int p_dni, String p_nombre, String p_apellido, int p_anio)
```