



Instituto Politécnico Nacional
UPIITA

Práctica 5. Clases

Unidad de Aprendizaje: Programación

Alumno: Brayan Lucero Molina

Boleta: 2024640465

Grupo: 1TV3

Carrera: Ingeniería en Telemática

Fecha: 27 de octubre de 2025

Objetivo

Determinar las clases que conforman un problema y crearlas en un lenguaje de programación.

Introducción

En esta práctica se aplicó el concepto de Programación Orientada a Objetos (POO) mediante la definición de clases, atributos y métodos. Se desarrollaron dos programas que utilizan clases creadas por el alumno: el primero para modelar personas y el segundo para registrar alumnos y un profesor dentro de una lista.

Desarrollo

Programa 1: MPersonas.java — Arreglo de objetos tipo Persona

Descripción: Este programa permite capturar los datos de tres personas (nombre, edad, estatura y teléfono) y mostrarlos en pantalla mediante objetos de la clase `Persona`.

Pasos para construir el código:

1. Se define la clase `Persona` con atributos: nombre, edad, estatura y teléfono.
2. Se crea un constructor para inicializar los valores de cada persona.
3. Se implementa un método `mostrarDatos()` para imprimir la información con el formato indicado.
4. En la clase principal `MainPersonas`, se usa un arreglo de tres objetos `Persona`.
5. Se capturan los datos desde teclado y se almacenan en el arreglo.
6. Finalmente, se recorren los objetos y se muestran los datos de cada persona.

```
1  /**
2   * @file Persona.java
3   * @author Brayan Lucero Molina
4   * @boleta 2024640465
5   * @practica 5
6   * @fecha 26 oct 2025
7   * @descripcion Clase Persona con atributos y metodo para mostrar
   *   datos
8   */
9
10 public class Persona {
11
12     // Atributos
13     String nombre;
14     int edad;
15     double estatura;
16     String telefono;
17 }
```

```

18 // Constructor
19 public Persona(String nombre, int edad, double estatura, String
20 telefono) {
21     this.nombre = nombre;
22     this.edad = edad;
23     this.estatura = estatura;
24     this.telefono = telefono;
25 }
26 // Metodo para mostrar los datos de la persona
27 public void mostrarDatos(int numeroPersona) {
28     int anioNacimiento = 2025 - edad;
29     System.out.println("Persona " + numeroPersona + ":");
30     System.out.println(" Hola !, mi nombre es " + nombre + ",
31         tengo " + edad + " anios, naci en "
32             + anioNacimiento + ", mido " + estatura +
33             " metros y mi numero de contacto es "
34             + telefono + " Saludos !");
35     System.out.println();
36 }
37 }

```

Listing 1: Persona.java

```

1 /**
2  * @file MPersonas.java
3  * @author Brayan Lucero Molina
4  * @boleta 2024640465
5  * @practica 5
6  * @fecha 26 oct 2025
7  * @descripcion Programa que pide datos de 3 personas y los guarda en
8     un arreglo de tipo Persona
9  */
10 import java.util.Scanner;
11
12 public class MPersonas {
13     public static void main(String[] args) {
14         // Crear un arreglo de 3 objetos Persona
15         try (Scanner sc = new Scanner(System.in)) {
16             // Crear un arreglo de 3 objetos Persona
17             Persona[] personas = new Persona[3];
18
19             System.out.println("CAPTURA DE DATOS DE 3 PERSONAS\n");
20
21             // Captura de datos
22             for (int i = 0; i < 3; i++) {
23                 System.out.println("Persona " + (i + 1) + ":");

```

```

24         System.out.print("Nombre: ");
25         String nombre = sc.nextLine();
26
27         System.out.print("Edad: ");
28         int edad = sc.nextInt();
29
30         System.out.print("Estatura (en metros): ");
31         double estatura = sc.nextDouble();
32         sc.nextLine(); // limpiar buffer
33
34         System.out.print("Telefono: ");
35         String telefono = sc.nextLine();
36
37         personas[i] = new Persona(nombre, edad, estatura,
38                                     telefono);
39         System.out.println();
40     }
41
42     // Mostrar resultados
43     System.out.println("=== RESULTADOS ===\n");
44     for (int i = 0; i < personas.length; i++) {
45         personas[i].mostrarDatos(i + 1);
46     }
47 }
48 }

```

Listing 2: MPersonas.java

```

CAPTURA DE DATOS DE 3 PERSONAS

Persona 1:
Nombre: Brayan Lucero Molina
Edad: 22
Estatura (en metros): 1.80
Telefono: 5586120136

```

```

=== RESULTADOS ===
Persona 1:
¡Hola!, mi nombre es Brayan Lucero Molina, tengo 22 años, nací en 2003, mido 1.8 metros y mi número de contacto es 5586120136 ¡Saludos!
Persona 2:
¡Hola!, mi nombre es Purco Hernandez soto, tengo 30 años, nací en 1995, mido 1.9 metros y mi número de contacto es 552014705210 ¡Saludos!
Persona 3:
¡Hola!, mi nombre es Jocelyn Trejo Ponce, tengo 23 años, nací en 2002, mido 1.7 metros y mi número de contacto es 5532148950 ¡Saludos!

```

Figura 1: Izquierda: Captura de datos de tres personas. Derecha: Impresión de resultados.

Programa 2: MainLista.java — Lista de alumnos y profesor

Descripción: Este programa implementa tres clases: `Profesor`, `Alumno` y `ListaAlumnos`. Permite capturar la información de un profesor y un número `n` de alumnos, almacenándolos dentro de un arreglo y mostrando la lista final en formato de inscripción.

Clases utilizadas:

- **Profesor:** contiene nombre completo y número de empleado.
- **Alumno:** almacena nombre, apellidos, boleta, fecha de nacimiento, carrera, grupo y correo.
- **ListaAlumnos:** agrupa a los alumnos junto con su profesor y muestra el total de inscritos.

Pasos para construir el código:

1. Se crean las clases `Profesor`, `Alumno` y `ListaAlumnos` con sus atributos y métodos.
2. En `MainLista.java`, se solicita primero la información del profesor.
3. Luego se pide al usuario la cantidad de alumnos a registrar.
4. Por cada alumno se capturan sus datos y se agregan al arreglo.
5. Finalmente, se imprime la lista completa con el formato indicado.

```
1  /**
2  *
3  * @file Profesor.java
4  * @author Brayan Lucero Molina
5  * @boleta 2024640465
6  * @practica 5
7  * @fecha 26 oct 2025
8  * @descripcion Clase Profesor con nombre completo y numero de
9  * empleado
10 */
11 public class Profesor {
12     String nombreCompleto;
13     String numeroEmpleado;
14
15     public Profesor(String nombreCompleto, String numeroEmpleado) {
16         this.nombreCompleto = nombreCompleto;
17         this.numeroEmpleado = numeroEmpleado;
18     }
19
20     public void mostrarDatos() {
21         System.out.println("Profesor: " + nombreCompleto);
```

```

22         System.out.println("No. Empleado: " + numeroEmpleado);
23     }
24 }

```

Listing 3: Profesor.java

```

1  /**
2   * @file Alumno.java
3   * @author Brayan Lucero Molina
4   * @boleta 2024640465
5   * @practica 5
6   * @fecha 26 oct 2025
7   * @descripcion Clase Alumno con los datos de un estudiante
8   */
9
10 public class Alumno {
11     String nombre;
12     String apellidoPaterno;
13     String apellidoMaterno;
14     String boleta;
15     String fechaNacimiento;
16     String carrera;
17     String grupo;
18     String correo;
19
20     public Alumno(String nombre, String apellidoPaterno, String
21         apellidoMaterno,
22         String boleta, String fechaNacimiento, String
23         carrera, String grupo, String correo) {
24         this.nombre = nombre;
25         this.apellidoPaterno = apellidoPaterno;
26         this.apellidoMaterno = apellidoMaterno;
27         this.boleta = boleta;
28         this.fechaNacimiento = fechaNacimiento;
29         this.carrera = carrera;
30         this.grupo = grupo;
31         this.correo = correo;
32     }
33
34     public void mostrarDatos(int numero) {
35         System.out.println(numero + ". " + apellidoPaterno + " " +
36             apellidoMaterno + ", " + nombre +
37             " - " + boleta + " - " + fechaNacimiento +
38             " - " + carrera +
39             " - " + grupo + " - " + correo);
40     }
41 }

```

Listing 4: Alumno.java


```

1 /**
2  * @file ListaAlumnos.java
3  * @author Brayan Lucero Molina
4  * @boleta 2024640465
5  * @practica 5
6  * @fecha 26 oct 2025
7  * @descripcion Clase que almacena y muestra la lista de alumnos
8  */
9
10 public class ListaAlumnos {
11     Alumno[] alumnos;
12     Profesor profesor;
13     int total;
14
15     public ListaAlumnos(Profesor profesor, int total) {
16         this.profesor = profesor;
17         this.total = total;
18         this.alumnos = new Alumno[total];
19     }
20
21     public void agregarAlumno(Alumno alumno, int posicion) {
22         alumnos[posicion] = alumno;
23     }
24
25     public void mostrarLista() {
26         profesor.mostrarDatos();
27         System.out.println("=====");
28         System.out.println("Lista de Alumnos Inscritos");
29         System.out.println("=====");
30         for (int i = 0; i < total; i++) {
31             alumnos[i].mostrarDatos(i + 1);
32         }
33         System.out.println("=====");
34         System.out.println("Total de alumnos inscritos: " + total);
35     }
36 }

```

Listing 5: ListaAlumnos.java

```

1 /**
2  *
3  * @file MainLista.java
4  * @author Brayan Lucero Molina
5  * @boleta 2024640465
6  * @practica 5
7  * @fecha 26 oct 2025
8  * @descripcion Programa que crea una lista de alumnos con un
   profesor y muestra todos los datos

```

```

9  */
10
11 import java.util.Scanner;
12
13 public class MainLista {
14     public static void main(String[] args) {
15         try (Scanner sc = new Scanner(System.in)) {
16             System.out.println("=== CAPTURA DE DATOS DEL PROFESOR ===");
17             System.out.print("Nombre completo: ");
18             String nombreProfesor = sc.nextLine();
19             System.out.print("Numero de empleado: ");
20             String numEmpleado = sc.nextLine();
21
22             Profesor profesor = new Profesor(nombreProfesor,
23                 numEmpleado);
24
25             System.out.println("\n=== REGISTRO DE ALUMNOS ===");
26             System.out.print("Cuántos alumnos desea registrar? ");
27             int n = sc.nextInt();
28             sc.nextLine(); // limpiar buffer
29
30             ListaAlumnos lista = new ListaAlumnos(profesor, n);
31
32             for (int i = 0; i < n; i++) {
33                 System.out.println("\nAlumno " + (i + 1) + ":");
34                 System.out.print("Nombre: ");
35                 String nombre = sc.nextLine();
36                 System.out.print("Apellido Paterno: ");
37                 String apP = sc.nextLine();
38                 System.out.print("Apellido Materno: ");
39                 String apM = sc.nextLine();
40                 System.out.print("Numero de Boleta: ");
41                 String boleta = sc.nextLine();
42                 System.out.print("Fecha de Nacimiento (dd/mm/aaaa): ");
43                 String fechaNac = sc.nextLine();
44                 System.out.print("Carrera: ");
45                 String carrera = sc.nextLine();
46                 System.out.print("Grupo: ");
47                 String grupo = sc.nextLine();
48                 System.out.print("Correo Electronico: ");
49                 String correo = sc.nextLine();
50
51                 Alumno alumno = new Alumno(nombre, apP, apM, boleta,
52                     fechaNac, carrera, grupo, correo);
53                 lista.agregarAlumno(alumno, i);
54             }
55         }
56     }
57 }

```

```

53
54         System.out.println("\n=== RESULTADOS ===\n");
55         lista.mostrarLista();
56     }
57 }
58 }

```

Listing 6: MainLista.java

```

=== CAPTURA DE DATOS DEL PROFESOR ===
Nombre completo: José luis
Numero de empleado: 123456

```

```

=== REGISTRO DE ALUMNOS ===
Cuantos alumnos desea registrar? 1

Alumno 1:
Nombre: Brayan
Apellido Paterno: Lucero
Apellido Materno: Molina
Numero de Boleta: 2024640465
Fecha de Nacimiento (dd/mm/aaaa): 20/02/2003
Carrera: ing.Telemática
Grupo: 1TV3
Correo Electronico: blucerom2300@alumno.ipn.mx

```

Figura 2: Izquierda: Captura del profesor. Derecha: Lista completa de alumnos inscritos.

Ejecución

La ejecución del programa 2 muestra en pantalla los datos del profesor y alumno registrado.

```
=== RESULTADOS ===  
  
Profesor: Jos? luis  
No. Empleado: 123456  
=====
```

```
Lista de Alumnos Inscritos  
=====  
I. Lucero Molina, Brayon - 2024640465 - 28/02/2003 - Ing.Telemtica - 1TV3 - bluceron2300@alumno.ipn.mx  
=====  
Total de alumnos inscritos: 1
```

Figura 3: Ejecución completa mostrando la lista de alumno inscrito.

Conclusion

En esta práctica se comprendió la importancia de las clases en la programación orientada a objetos. Al separar los atributos y métodos en clases independientes, el código se vuelve más ordenado, legible y facilita el mantenimiento.

El primer programa permitió trabajar con arreglos de objetos, mientras que el segundo mostró cómo interactúan distintas clases entre sí.