



# Algoritmos y Estructura de Datos

---

Unidad 1: Clases y objetos

Tema 1: Conceptos básicos de POO





# Tema 1: Conceptos básicos de POO

---





# Índice

---

## 1.1 Tema 1: Conceptos básicos de POO

- 1.1.1 Introducción
- 1.1.2 Clase
- 1.1.3 Objeto
- 1.1.4 Declaración y creación de objetos (operador new)
- 1.1.5 Acceso a los miembros de una clase (operador punto)
- 1.1.6 Paquete
- 1.1.7 Especificador de acceso **public**
- 1.1.8 Sentencia **import**
- 1.1.9 Ejemplo





# Capacidades

---

- Identifica los cambios en la forma de hacer un programa.
- Diseña clases y objetos.



# 1.1.1 Introducción

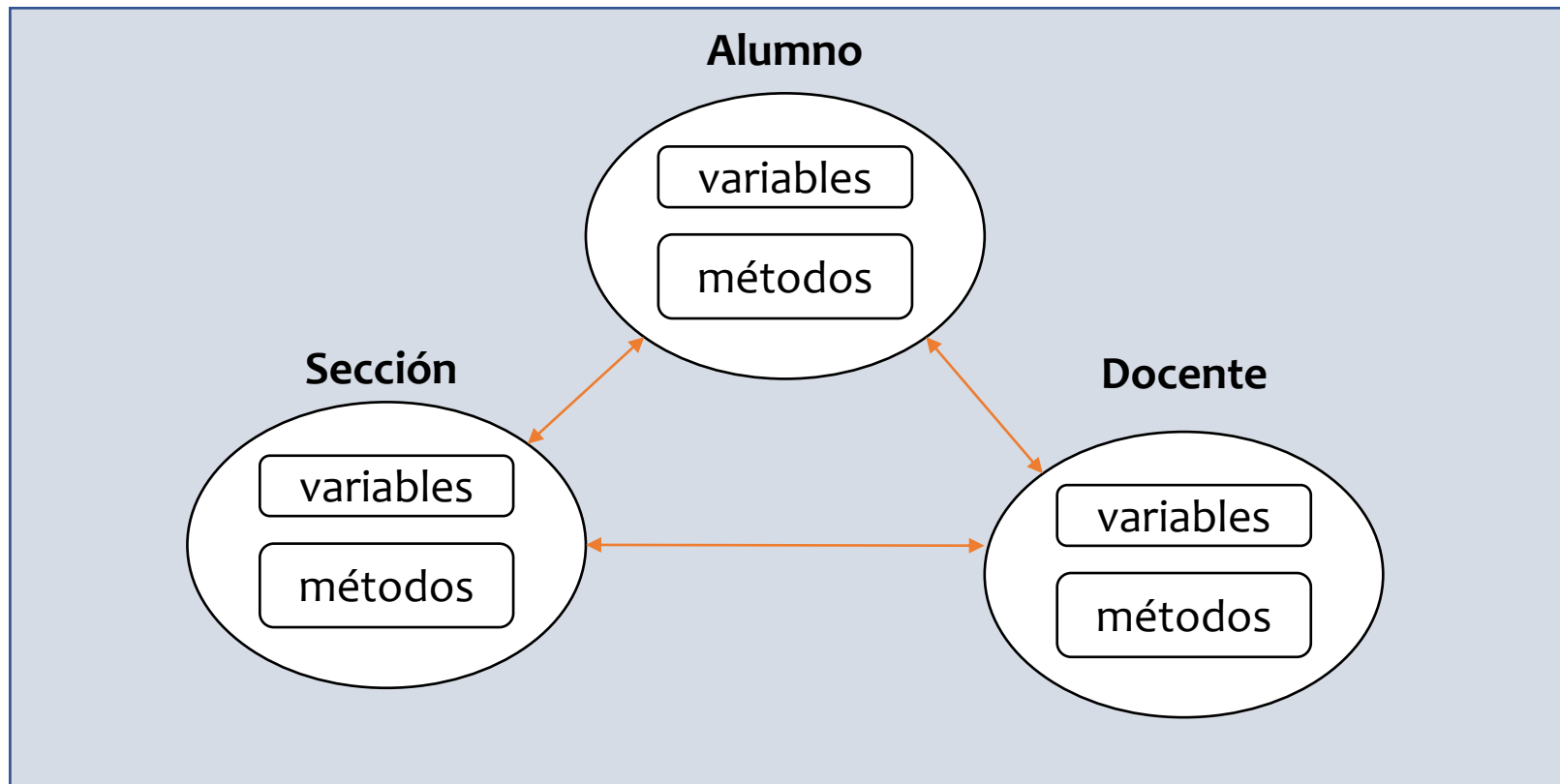
- En el mundo real vivimos rodeados de objetos: personas, sillas, computadoras, tazas, platos, etc.





# 1.1.1 Introducción

- La POO (Programación Orientada a Objetos) es una metodología de programación que permite crear programas de computadora tomando al objeto como unidad esencial de programación.



objetos en la memoria RAM



# 1.1.1 Introducción

- Los objetos en la POO son entidades o unidades independientes que poseen **atributos** y **operaciones**.
- Los **atributos** son las características del objeto que se representan mediante variables.
- Las **operaciones** son las formas de operar del objeto que se representan mediante métodos.



Persona	
Atributos	Operaciones
nombre edad peso estatura	hablar caminar dormir comer





## 1.1.1 Introducción



CuentaBancaria	
Atributos	Operaciones
titular numero saldo	depositar retirar

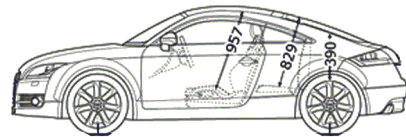




# 1.1.1 Introducción

- En el mundo real, para poder fabricar objetos, se requiere de un plano o un conjunto de planos.
- En forma similar, para poder crear un objeto software, se requiere de una clase o modelo.

Plano



objeto



Clase

```
public class Automovil {  
    .  
    .  
    .  
}
```

objeto





## 1.1.2 Clase

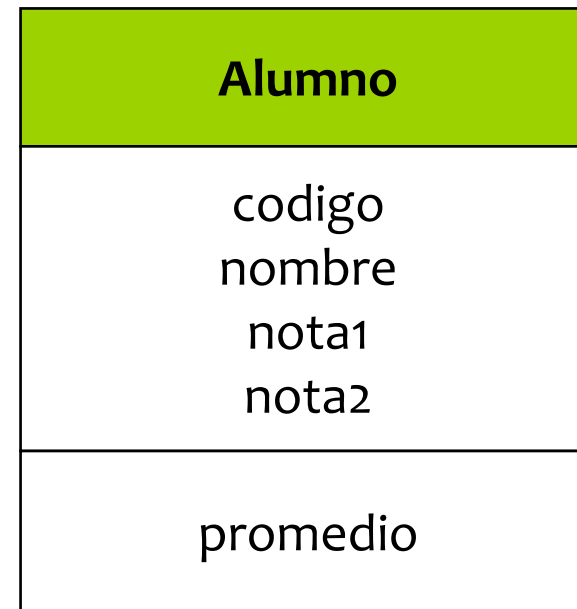
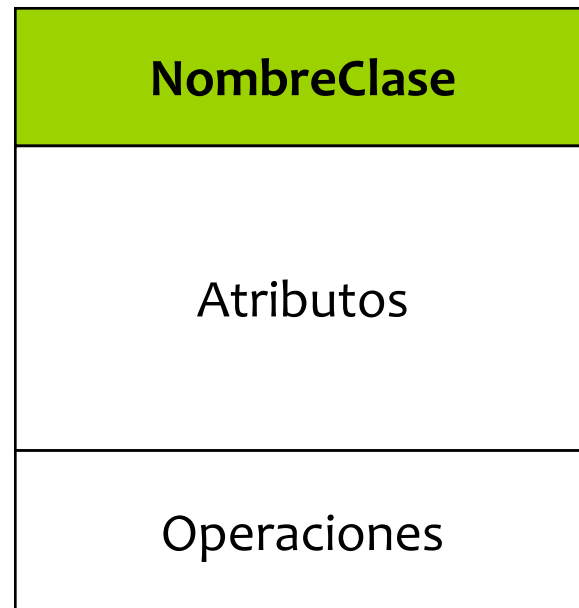
- Una **clase** es una plantilla (más propiamente, un **tipo de dato**) que especifica los *atributos* (mediante variables denominadas *variables miembro*) y las *operaciones* (mediante métodos denominados *métodos miembro*) de un tipo de objeto a crear.

```
public class NombreClase {  
    // Atributos (variables miembro)  
    .  
    .  
    .  
    // Operaciones (métodos miembro)  
    .  
    .  
    .  
}
```



## 1.1.2 Clase

- Diagrama UML de una clase
- UML = Lenguaje Unificado de Modelado





## 1.1.2 Clase

- Todo nombre de clase debe comenzar con letra mayúscula.

```
public class Alumno {  
    // Atributos  
    int codigo, nota1, nota2;  
    String nombre;  
    // Operaciones  
    double promedio() {  
        return (nota1 + nota2) / 2.0;  
    }  
}
```

## 1.1.3 Objeto

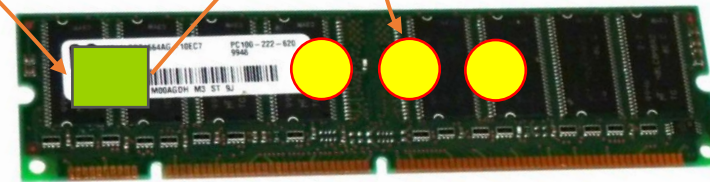
- Un objeto es una instancia o ejemplar creado en base a una clase y ubicado en la memoria del computador.
- Para cada objeto se creará una copia de cada una de las variables miembro.
- Los métodos miembro son compartidos por todos los objetos de la clase.
- Todo nombre de objeto debe comenzar con letra minúscula.
- La clase se carga a la memoria RAM la primera vez que se menciona en el código, en adelante será utilizada desde la RAM para crear objetos (se convierte en una fábrica de objetos).

```
public class Alumno {  
    // Atributos públicos  
    public int codigo, nota1, nota2;  
    public String nombre;  
    // Operaciones públicas  
    public double promedio() {  
        return (nota1 + nota2) / 2.0;  
    }  
}
```

Clase

la clase se  
carga a la  
memoria

se crean  
objetos de la  
clase



objeto en la memoria RAM



# 1.1.4 Declaración y creación de objetos

## Forma 1

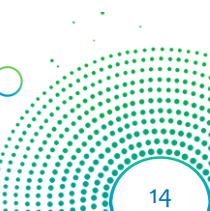
```
// Declarar una variable referencia
NombreClase nombreObjeto;

// Crear el objeto usando el operador new
nombreObjeto = new NombreClase();
```

## Ejemplo

```
// Declarar una variable referencia
Alumno a;

// Crear el objeto usando el operador new
a = new Alumno();
```



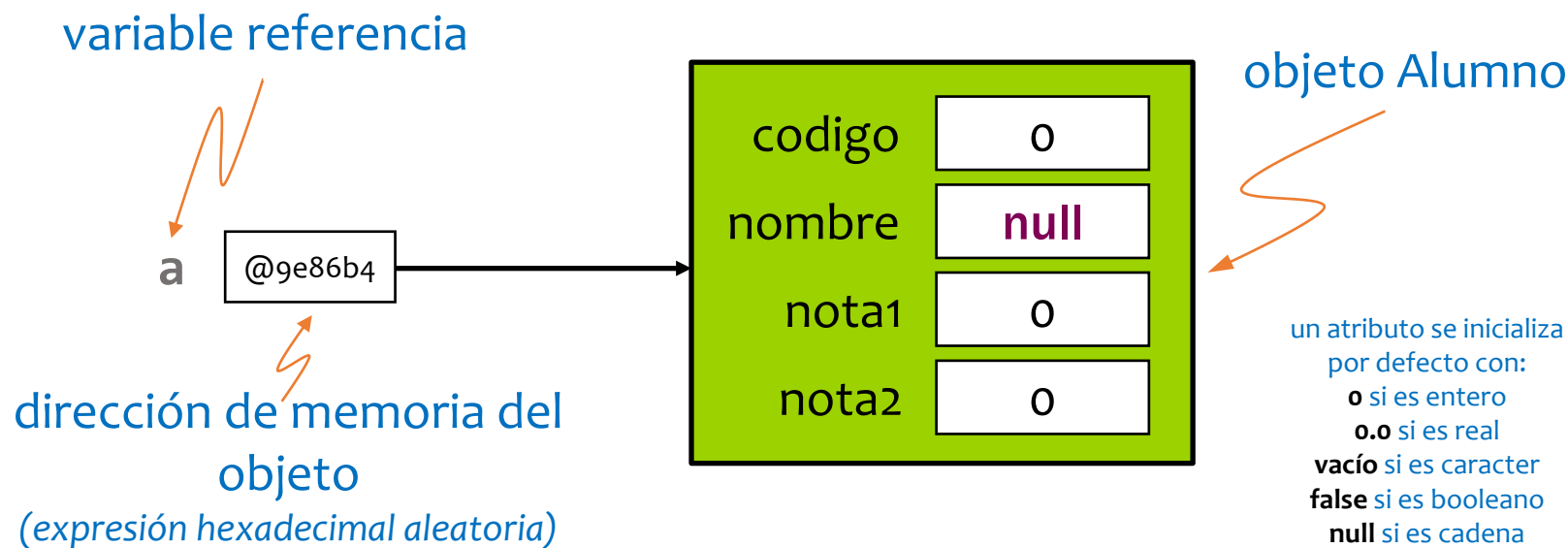


# 1.1.4 Declaración y creación de objetos

```
Alumno a;
```

a ? ↗ Si es global, se inicializa automáticamente con el valor **null**. Si es local, no contiene nada.

```
a = new Alumno();
```



Si mandamos a imprimir directamente el objeto **a** Java muestra su dirección de memoria



## 1.1.4 Declaración y creación de objetos

### Forma 2

```
// Declarar una variable referencia y crear el objeto con new  
NombreClase nombreObjeto = new NombreClase();
```

### Ejemplo

```
// Declarar una variable referencia y crear el objeto con new  
Alumno a = new Alumno();
```

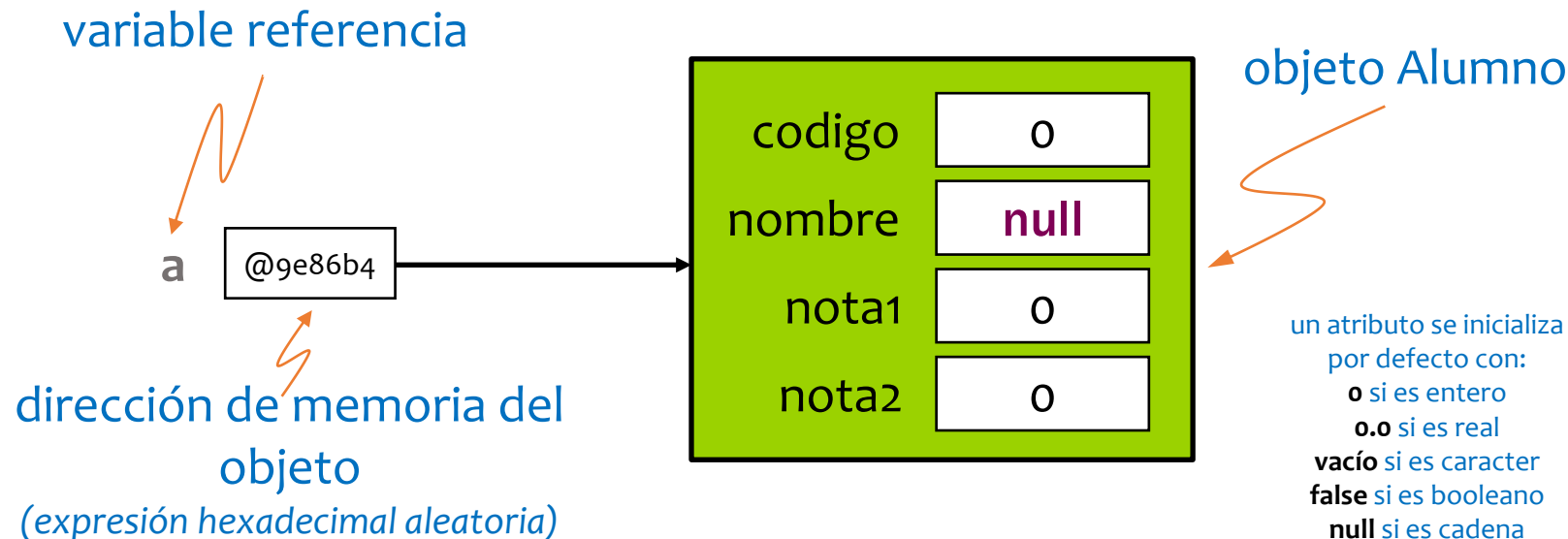






## 1.1.4 Declaración y creación de objetos

```
Alumno a = new Alumno();
```





# 1.1.5 Acceso a los miembros de una clase

- El acceso a las variables y a los métodos de una clase, para un objeto en particular, se hace mediante la variable referencia (nombre del objeto) y el operador punto (.)

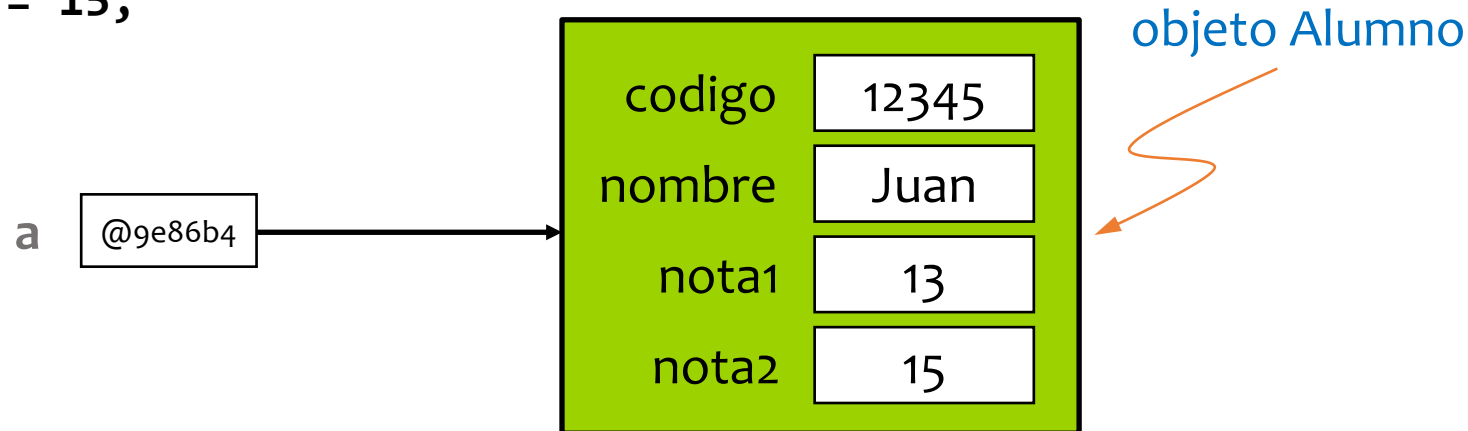
// Asigna valores a los atributos del objeto a

a.codigo = 12345;

a.nombre = "Juan";

a.nota1 = 13;

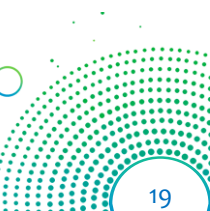
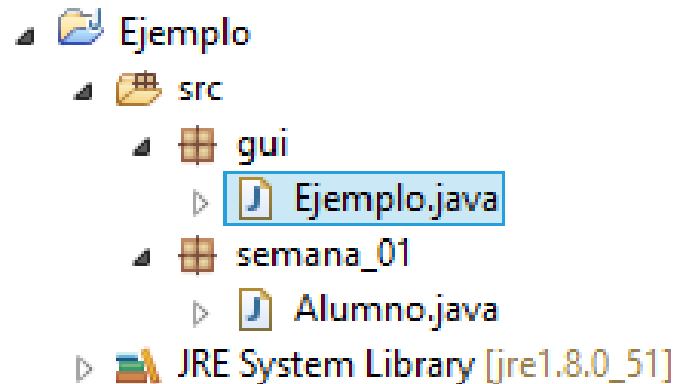
a.nota2 = 15;





## 1.1.6 Paquete

- Un paquete es un identificador que congrega a una o varias clases.
- Por cada paquete, Java crea una carpeta con el mismo nombre del paquete.
- Se usan paquetes para evitar conflictos entre clases con el mismo nombre.
- Para crear un paquete se usa la siguiente sentencia como la primera de todas dentro de la clase: **package nombrePaquete;**
- Por ejemplo, para implementar la clase Alumno en el paquete **semana\_01** debe anteponerse al inicio de dicha clase **package semana\_01;**
- Todo nombre de paquete debe comenzar con letra minúscula.





## 1.1.7 Especificador de acceso *public*

- Permite el acceso a la clase desde cualquier lugar del programa.
- Los atributos y operaciones también tienen que ser declarados como public.

```
package semana_01;

public class Alumno {
    // Atributos públicos
    public int codigo, nota1, nota2;
    public String nombre;
    // Operaciones públicas
    public double promedio() {
        return (nota1 + nota2) / 2.0;
    }
}
```

semana\_01  
Alumno.java





## 1.1.8 Sentencia *import*

- La sentencia **import** permite acceder a los atributos y operaciones **public** de las clases de otro paquete que hayan sido declaradas como **public**.
- La sentencia import tiene la siguiente sintaxis:

**import** nombrePaquete.NombreClase;

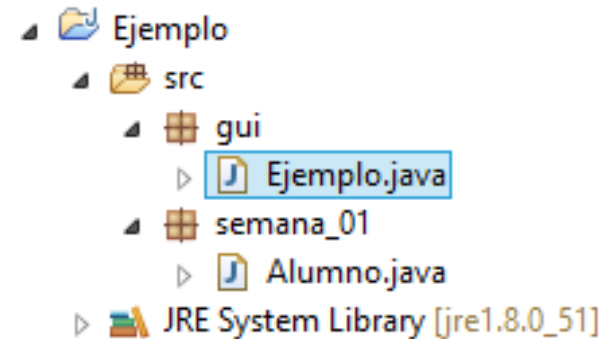
- Por ejemplo, para poder utilizar la clase **Alumno** en la clase **Ejemplo** se debe colocar la siguiente sentencia:

**import** semana\_01.Alumno;

```
package gui;

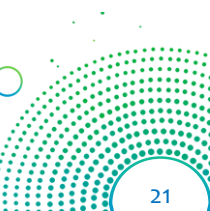
import semana_01.Alumno;
...

public class Ejemplo extends JFrame implements ActionListener {
    ...
}
```

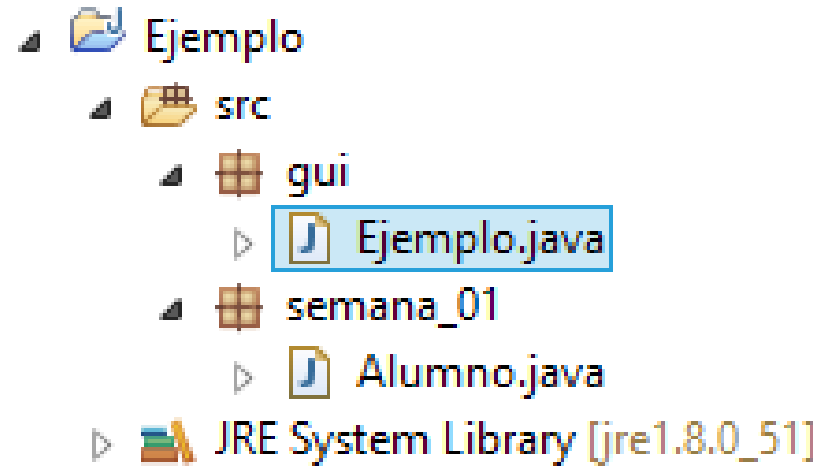


- No se requiere importar aquellas clases que forman parte del mismo paquete.
- Para importar todas las clases implementadas dentro de un mismo paquete se puede utilizar el símbolo asterisco. Así:

**import** nombrePaquete.\*;



## 1.1.9 Ejemplo



- Implementa la clase **Alumno** en el *package* **semana\_01**
- Coloca en la clase **Ejemplo**:  
`import semana_01.Alumno;`
- A la pulsación del botón Procesar:
  - Declara y crea un objeto Alumno.
  - Ingresa datos fijos.
  - Visualiza todos sus datos.



# Conclusiones

---

- Un objeto es una entidad independiente que posee **atributos** y **operaciones**.
- Los **atributos** son las características del objeto que se representan mediante variables.
- Las **operaciones** son las formas de operar del objeto que se representan mediante métodos.
- Un objeto es un ejemplar creado en base a una clase.

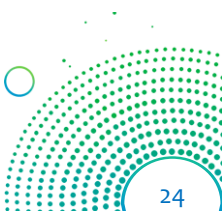




# Referencias bibliográficas

---

- **Joyanes Aguilar Luis.** Fundamentos de programación: algoritmos, estructuras de datos y objetos. Madrid, España: McGraw-Hill (005.1 JOYA/A 2021)
- **Lewis John.** Estructuras de datos con Java: diseño de estructuras y algoritmos. Madrid, Pearson Educación (005.73 LEWI/E 2021)
- **Deitel Harvey.** Cómo programar en Java. México, D.F.: Pearson Educación (005.133J DEIT 2021)





# GRACIAS



## **SEDE MIRAFLORES**

Calle Díez Canseco Cdra 2 / Pasaje Tello  
Miraflores – Lima  
Teléfono: 633-5555

## **SEDE INDEPENDENCIA**

Av. Carlos Izaguirre 233  
Independencia – Lima  
Teléfono: 633-5555

## **SEDE BREÑA**

Av. Brasil 714 – 792  
(CC La Rambla – Piso 3)  
Breña – Lima  
Teléfono: 633-5555

## **SEDE TRUJILLO**

Calle Borgoño 361  
Trujillo  
Teléfono: (044) 60-2000

## **SEDE SAN JUAN DE LURIGANCHO**

Av. Próceres de la Independencia 3023-3043  
San Juan de Lurigancho – Lima  
Teléfono: 633-5555

## **SEDE LIMA CENTRO**

Av. Uruguay 514  
Cercado – Lima  
Teléfono: 419-2900

## **SEDE BELLAVISTA**

Av. Mariscal Oscar R. Benavides 3866 – 4070  
(CC Mall Aventura Plaza)  
Bellavista – Callao  
Teléfono: 633-5555

## **SEDE AREQUIPA**

Av. Porongoche 500  
(CC Mall Aventura Plaza)  
Paucaarpata - Arequipa  
Teléfono: (054) 60-3535