

# Taller Programación Orientada a Objetos (POO)

## Mini proyecto: Mini Sistema de Login

### ¿Qué clases identificas en el problema?

En el código se pueden identificar **cuatro clases principales**:

1. **Usuario** (clase base o padre)
2. **Administrador** (subclase)
3. **Cliente** (subclase)
4. **SistemaLogin**

### ¿Qué atributos y métodos tendría cada clases?

#### Clase: Usuario

##### **Atributos:**

- nombre: almacena el nombre del usuario.
- contraseña: almacena la contraseña del usuario.

##### **Métodos:**

- `__init__(self, nombre, contraseña)`: inicializa los atributos.
- `mostrar_bienvenida(self)`: muestra un mensaje genérico de bienvenida.

#### Clase: Administrador (*hereda de Usuario*)

##### **Atributos:**

- Hereda todos los atributos de Usuario (nombre, contraseña).

##### **Métodos:**

- `mostrar_bienvenida(self)`: sobrescribe el método del padre (→ **polimorfismo**) para mostrar un mensaje especial indicando que tiene acceso total al sistema.

#### Clase: Cliente (*hereda de Usuario*)

##### **Atributos:**

- Hereda todos los atributos de Usuario (nombre, contraseña).

##### **Métodos:**

- `mostrar_bienvenida(self)`: sobrescribe el método del padre para mostrar un mensaje de bienvenida para clientes.

#### Clase: SistemaLogin

##### **Atributos:**

- `usuarios`: diccionario donde se guardan las instancias de usuarios (clave = nombre, valor = objeto Usuario, Administrador o Cliente).

##### **Métodos:**

- `__init__(self)`: inicializa el diccionario de usuarios vacío.
- `registro(self, nombre, contraseña, tipo)`: permite registrar nuevos usuarios según el tipo (cliente o admin).
- `login(self, nombre, contraseña)`: valida las credenciales e imprime un mensaje usando **polimorfismo**.

#### **¿Qué relaciones existen entre ellas?**

##### **Herencia (generalización)**

Administrador y Cliente **heredan** de la clase Usuario. Esto permite reutilizar atributos y métodos comunes.

##### **Composición**

SistemaLogin **contiene** (agrega) objetos de tipo Usuario, Administrador o Cliente dentro de su diccionario usuarios.

##### **Polimorfismo**

El método `mostrar_bienvenida()` es redefinido en las subclases (Administrador y Cliente) y se ejecuta de forma distinta según el tipo del objeto.

#### **Diagramas UML**

##### Diagrama de clases UML

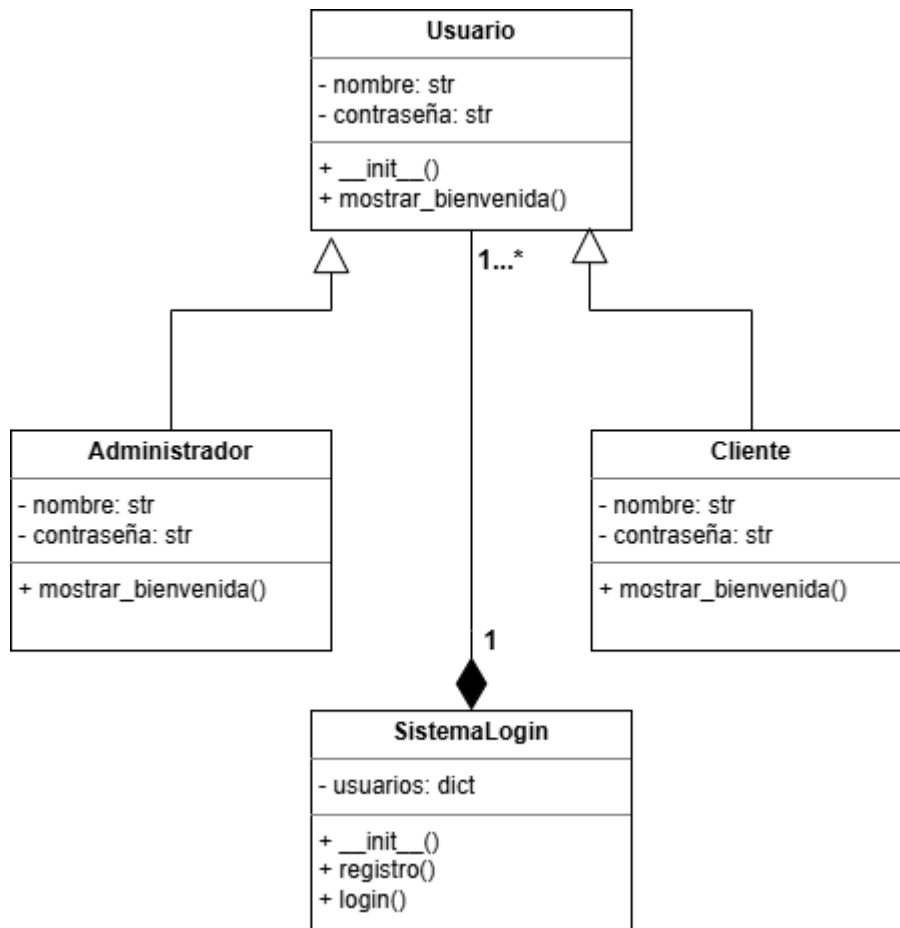


Diagrama de objetos UML

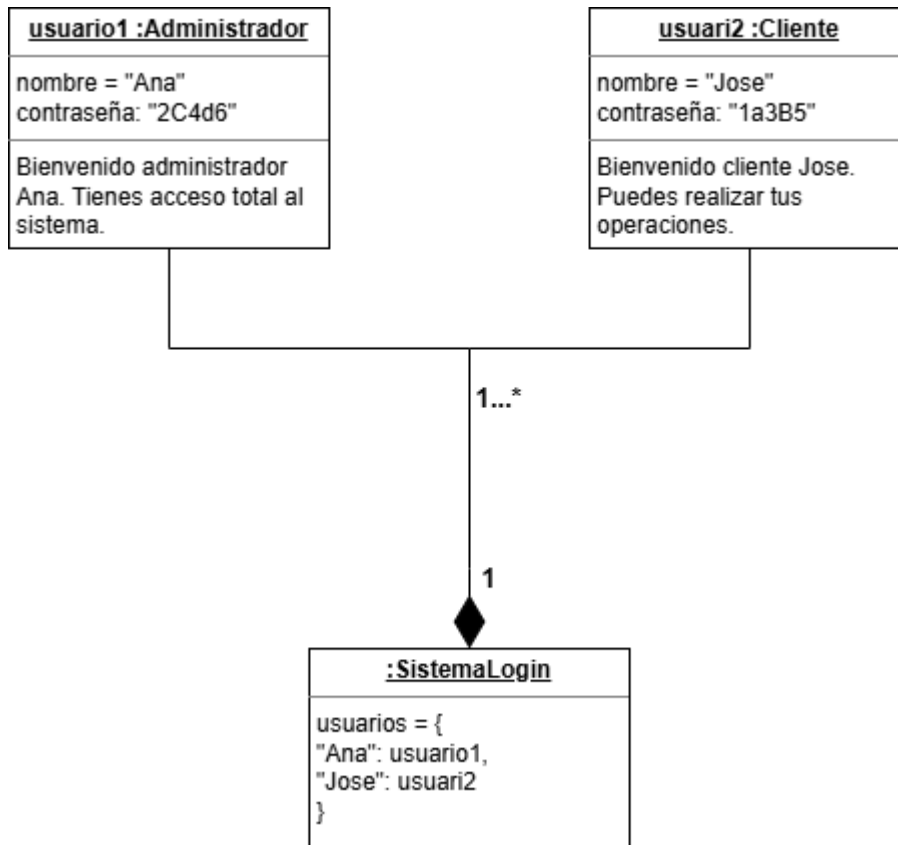
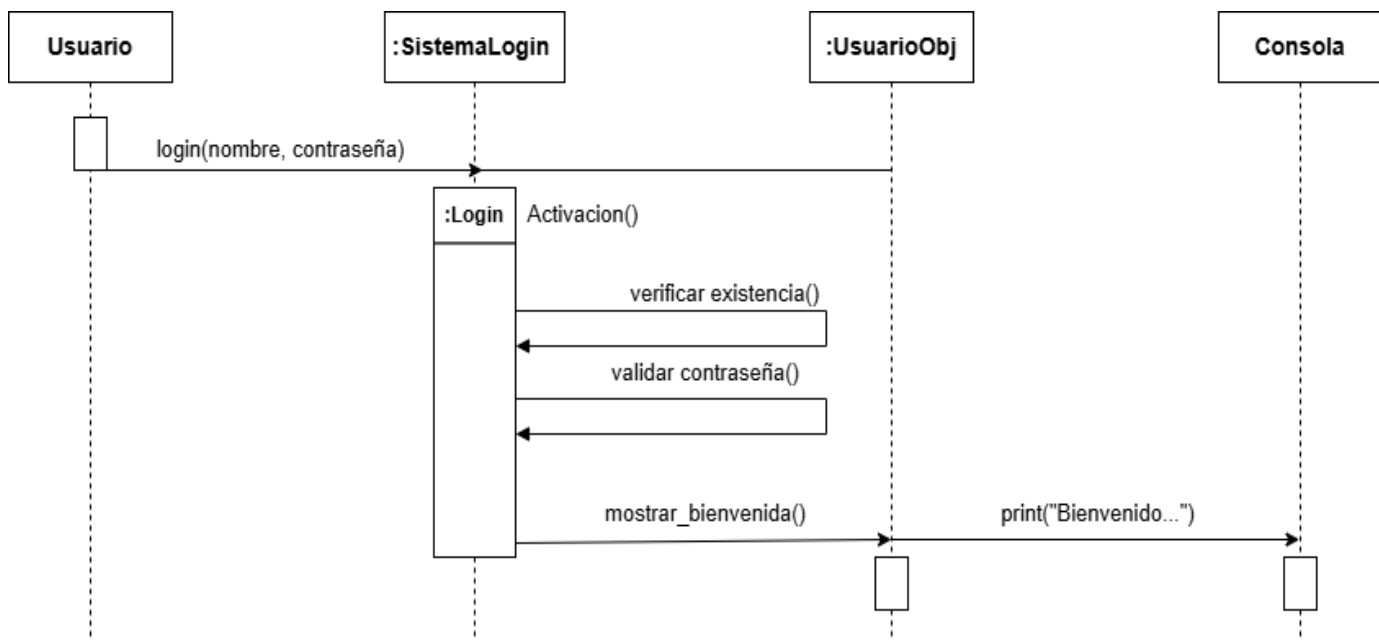
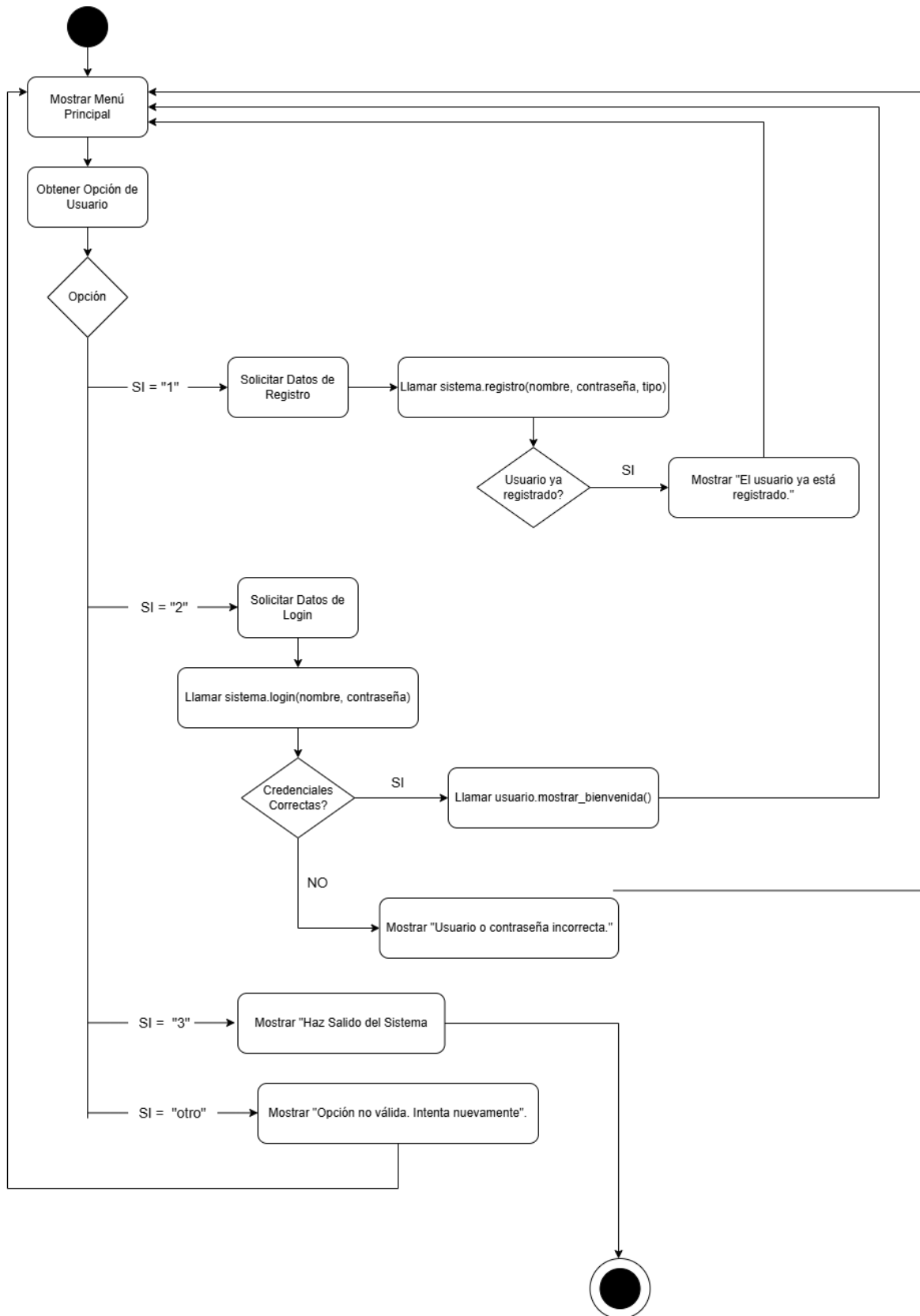


Diagrama de secuencia UML



## Diagrama de actividad UML



## Código de Mini Sistema de Login Python

```
1  #---- Mini Sistema de Login ----#
2  class Usuario:
3      def __init__(self, nombre, contraseña):
4          self.nombre = nombre
5          self.contraseña = contraseña
6
7      def mostrar_bienvenida(self):
8          print(f"Bienvenido {self.nombre}")
9
10
11  # --- Clases hijas ---
12  class Administrador(Usuario):
13      def mostrar_bienvenida(self):
14          print(f"Bienvenido administrador {self.nombre}. Tienes acceso total al sistema.")
15
16
17  class Cliente(Usuario):
18      def mostrar_bienvenida(self):
19          print(f"Bienvenido cliente {self.nombre}. Puedes realizar tus operaciones.")
20
```

```
21
22  # --- Sistema de Login ---
23  class SistemaLogin:
24      def __init__(self):
25          self.usuarios = {}
26
27      def registro(self, nombre, contraseña, tipo="cliente"):
28          if nombre in self.usuarios:
29              print("El usuario ya está registrado.")
30          elif tipo == "admin":
31              self.usuarios[nombre] = Administrador(nombre, contraseña)
32          else:
33              self.usuarios[nombre] = Cliente(nombre, contraseña)
34              print("Usuario registrado con éxito.")
35
36      def login(self, nombre, contraseña):
37          if nombre in self.usuarios and self.usuarios[nombre].contraseña == contraseña:
38              self.usuarios[nombre].mostrar_bienvenida() # <- polimorfismo
39          else:
40              print("Usuario o contraseña incorrecta.")
41
```

```

41
42
43 # --- PROGRAMA PRINCIPAL ---
44 sistema = SistemaLogin()
45
46 while True:
47     print("\n--- MENÚ PRINCIPAL ---")
48     print("1. Registrar usuario")
49     print("2. Iniciar sesión")
50     print("3. Salir")
51     opcion = input("Elige una opción: ")
52
53     if opcion == "1":
54         nombre = input("Nombre de usuario: ")
55         contraseña = input("Contraseña: ")
56         tipo = input("Tipo de usuario (admin/cliente): ").lower()
57         sistema.registro(nombre, contraseña, tipo)
58
59     elif opcion == "2":
60         nombre = input("Nombre de usuario: ")
61         contraseña = input("Contraseña: ")
62         sistema.login(nombre, contraseña)
63
64     elif opcion == "3":
65         print("Haz Salido del Sistema")
66         break
67
68     else:
69         print("Opción no válida. Intenta nuevamente.")
70

```

### Ejecución del programa

#### Menú Principal

```

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: █

```

Donde se muestran las opciones para elegir, al usuario:

1. Opción (Registrar usuario)
2. Opción (Iniciar sesión)
3. Opción (Salir)

Opción 1

```
Elige una opción: 1
Nombre de usuario: Horacio
Contraseña: 1234567
Tipo de usuario (admin/cliente): cliente
```

Cuando se elige la opción 1 (Registrar usuario) el programa le pedirá al usuario su Nombre y Contraseña, además le preguntará cual es su tipo de usuario (Admin o Cliente).

```
Elige una opción: 1
Nombre de usuario: Horacio
Contraseña: 1234567
Tipo de usuario (admin/cliente): cliente
Usuario registrado con éxito.
```

Después de ingresar los datos, el programa guarda los datos en un diccionario y le da el mensaje al usuario de que se registró con éxito.

```
Elige una opción: 1
Nombre de usuario: Horacio
Contraseña: 1234567
Tipo de usuario (admin/cliente): cliente
Usuario registrado con éxito.

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 
```

Es importante tener en cuenta que después de realizar el registro o el login, se volverá a mostrar el menú principal, ya que es un bucle.

Opción 2



```
Elige una opción: 2
Nombre de usuario: Horacio
Contraseña: 1234567
Bienvenido cliente Horacio. Puedes realizar tus operaciones.
```

En la opción 2 (Inicio sesión) se le pedirá al usuario su nombre y contraseña. Se iniciará sesión con éxito si los datos fueron previamente registrados en la opción 1. Al iniciar sesión con éxito se le mostrara al usuario un mensaje de bienvenida, este mensaje variara dependiendo el usuario es Client o Admin.

```
--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 2
Nombre de usuario: ana
Contraseña: 123
Usuario o contraseña incorrecta.
```

Si se ingresan datos que no fueron previamente registrados por la opción 1. Al usuario le aparecerá un mensaje de usuario o contraseña incorrecta.

Opción 3

```
Elige una opción: 3
Haz Salido del Sistema
PS C:\Users\MI PC\Documents\Taller de POO(SENA)> |
```

En la opción 3 (Salir) es para salir del sistema. Ya que como lo mencione en su momento estamos en un bucle, donde hay que dar una condición para terminarlo. Ingresando esta opción el usuario saldrá del programa y este mismo le mostrara un mensaje de que salió del programa.

```
--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 1
Nombre de usuario: Horacio
Contraseña: 1234567
Tipo de usuario (admin/cliente): cliente
Usuario registrado con éxito.

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 2
Nombre de usuario: Horacio
Contraseña: 1234567
Bienvenido cliente Horacio. Puedes realizar tus operaciones.

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 2
Nombre de usuario: ana
Contraseña: 123
Usuario o contraseña incorrecta.

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 3
Haz Salido del Sistema
PS C:\Users\MI PC\Documents\Taller de POO(SENA)> |
```

## Clases en Java

Una **clase** es la plantilla o plano para crear objetos. En Java, una clase se define usando la palabra clave **class** y suele tener la misma estructura del archivo que la contiene (por ejemplo, la clase Usuario va en el archivo Usuario.java).

**public class:** El modificador **public** indica que la clase es accesible en todo el proyecto.

**Atributos:** Variables que almacenan datos. Se recomienda usar modificadores como **private** (solo accesible dentro de la clase) o **protected** (accesible en la clase y sus subclases).

**Constructor:** Es un bloque de código que se ejecuta al crear una nueva instancia (`new NombreDeLaClase()`). Es esencial para inicializar los atributos del objeto.

### Métodos en Java

Un **método** es un bloque de código que contiene una serie de instrucciones y que se puede ejecutar. Definen la funcionalidad de la clase.

**Modificador de Acceso:** Controla dónde puede ser llamado el método. Los más comunes son: **public**, **private** y **protected**.

**static (Opcional):** Indica que el método pertenece a la **clase** en sí misma, no a una instancia (objeto) específica. El ejemplo más importante es el método principal: `public static void main(...)`.

**Tipo de Retorno:** El tipo de dato que el método devolverá al finalizar su ejecución. Si el método no devuelve nada, se usa la palabra clave **void**.

**Nombre:** El nombre del método (debe ser un verbo o una frase que describa la acción).

Ejemplo (`calcularSuma`)

**Parámetros:** Una lista separada por comas de variables y sus tipos, que el método acepta como entrada. Ejemplo (`int a, int b`)

### Herencia: Compartiendo Base

La **Herencia** se aplicó usando la palabra clave **extends** para crear una jerarquía de clases:

- **Clase Base:** Usuario (Superclase)
- **Clases Heredadas:** Administrador y Cliente (`extends Usuario`)

Esto permitió que Administrador y Cliente **reutilizaran** los atributos comunes (nombre, contraseña) y el constructor del padre mediante **super()**, sin tener que definirlos de nuevo.

### Polimorfismo: Comportamiento Variado

El **Polimorfismo** se aplicó mediante la **Sobreescripción de Métodos** (Method Overriding) en el método `mostrarBienvenida()`:

- Todas las clases tienen el método `mostrarBienvenida()`.
- Las clases hijas (Administrador y Cliente) usan la anotación **@Override** para proporcionar una **implementación única y especializada** del saludo.

## Código de Mini Sistema de Login JAVA

```
1 // Archivo Único: SistemaLogin.java
2
3 import java.util.HashMap;
4 import java.util.Scanner;
5
6 // Clase principal del sistema que contiene el mapa de usuarios y la lógica de negocio
7 public class SistemaLogin {
8     // Se usa HashMap<String, Usuario> como equivalente al diccionario de Python
9     private HashMap<String, Usuario> usuarios;
10
11     // Constructor
12     public SistemaLogin() {
13         this.usuarios = new HashMap<>();
14     }
15
16     // Método de Registro
17     public void registro(String nombre, String contrasena, String tipo) {
18         if (usuarios.containsKey(nombre)) {
19             System.out.println(x:"El usuario ya está registrado.");
20         } else if (tipo.equalsIgnoreCase(anotherString:"admin")) {
21             usuarios.put(nombre, new Administrador(nombre, contrasena));
22             System.out.println(x:"Administrador registrado con éxito.");
23         } else {
24             // Si el tipo no es "admin", por defecto es Cliente
25             usuarios.put(nombre, new Cliente(nombre, contrasena));
26             System.out.println(x:"Usuario registrado con éxito.");
27         }
28     }
29 }
```

```

29
30 // Método de Login
31 public void login(String nombre, String contrasena) {
32     if (usuarios.containsKey(nombre)) {
33         Usuario usuario = usuarios.get(nombre);
34         // Compara la contraseña usando .equals()
35         if (usuario.getContrasena().equals(contrasena)) {
36             // Polimorfismo: llama a la versión correcta de mostrarBienvenida
37             usuario.mostrarBienvenida();
38         } else {
39             System.out.println(x:"Usuario o contraseña incorrecta.");
40         }
41     } else {
42         System.out.println(x:"Usuario o contraseña incorrecta.");
43     }
44 }
45

```

```

45
46 // --- CLASES INTERNAS (Para mantener la estructura en un solo archivo) ---
47
48 // Clase Base: Usuario (no-pública para permitir que esté en el mismo archivo)
49 static class Usuario {
50     protected String nombre;
51     protected String contrasena;
52
53     // Constructor
54     public Usuario(String nombre, String contrasena) {
55         this.nombre = nombre;
56         this.contrasena = contrasena;
57     }
58
59     // Getter para la contraseña
60     public String getContrasena() {
61         return contrasena;
62     }
63
64     // Método polimórfico
65     public void mostrarBienvenida() {
66         System.out.println("Bienvenido " + nombre);
67     }
68 }
69
70 // Clase Hija: Administrador
71 static class Administrador extends Usuario {
72     // Constructor
73     public Administrador(String nombre, String contrasena) {
74         super(nombre, contrasena);
75     }
76

```

```
76
77     // Sobrescritura (Polimorfismo)
78     @Override
79     public void mostrarBienvenida() {
80         System.out.println("Bienvenido administrador " + nombre + ". Tienes acceso total al sistema.");
81     }
82 }
83
84 // Clase Hija: Cliente
85 static class Cliente extends Usuario {
86     // Constructor
87     public Cliente(String nombre, String contrasena) {
88         super(nombre, contrasena);
89     }
90
91     // Sobrescritura (Polimorfismo)
92     @Override
93     public void mostrarBienvenida() {
94         System.out.println("Bienvenido cliente " + nombre + ". Puedes realizar tus operaciones.");
95     }
96 }
97
```

```

98
99 // --- PROGRAMA PRINCIPAL (Método main) ---
Run | Debug
100 public static void main(String[] args) {
101     Scanner scanner = new Scanner(System.in);
102     SistemaLogin sistema = new SistemaLogin();
103     String opcion;
104
105     while (true) {
106         System.out.println(x:"\n--- MENÚ PRINCIPAL ---");
107         System.out.println(x:"1. Registrar usuario");
108         System.out.println(x:"2. Iniciar sesión");
109         System.out.println(x:"3. Salir");
110         System.out.print(s:"Elige una opción: ");
111         opcion = scanner.nextLine();
112
113         switch (opcion) {
114             case "1":
115                 System.out.print(s:"Nombre de usuario: ");
116                 String nombreReg = scanner.nextLine();
117                 System.out.print(s:"Contraseña: ");
118                 String contrasenaReg = scanner.nextLine();
119                 System.out.print(s:"Tipo de usuario (admin/cliente): ");
120                 String tipoReg = scanner.nextLine().toLowerCase();
121                 sistema.registro(nombreReg, contrasenaReg, tipoReg);
122                 break;
123             case "2":
124                 System.out.print(s:"Nombre de usuario: ");
125                 String nombreLog = scanner.nextLine();
126                 System.out.print(s:"Contraseña: ");
127                 String contrasenaLog = scanner.nextLine();
128                 sistema.login(nombreLog, contrasenaLog);
129                 break;
130             case "3":
131                 System.out.println(x:"Haz Salido del Sistema");
132                 scanner.close();
133                 return;
134             default:
135                 System.out.println(x:"Opción no válida. Intenta nuevamente.");
136                 break;
137         }
138     }
139 }
140 }

```

### Ejecución del programa

La ejecución o salida del programa es igual, ya que lo único que cambia es el lenguaje de programación.

## Menú Principal

```
--- MENÚ PRINCIPAL ---  
1. Registrar usuario  
2. Iniciar sesión  
3. Salir  
Elige una opción: █
```

Donde se muestran las opciones para elegir, al usuario:

1. Opción (Registrar usuario)
2. Opción (Iniciar sesión)
3. Opción (Salir)

### Opción 1

```
Elige una opción: 1  
Nombre de usuario: Horacio  
Contraseña: 1234567  
Tipo de usuario (admin/cliente): cliente █
```

Cuando se elige la opción 1 (Registrar usuario) el programa le pedirá al usuario su Nombre y Contraseña, además le preguntará cual es su tipo de usuario (Admin o Cliente).

```
Elige una opción: 1  
Nombre de usuario: Horacio  
Contraseña: 1234567  
Tipo de usuario (admin/cliente): cliente  
Usuario registrado con éxito.
```

Después de ingresar los datos, el programa guarda los datos en un diccionario y le da el mensaje al usuario de que se registró con éxito.



```
Elige una opción: 1
Nombre de usuario: Horacio
Contraseña: 1234567
Tipo de usuario (admin/cliente): cliente
Usuario registrado con éxito.

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: █
```

Es importante tener en cuenta que después de realizar el registro o el login, se volverá a mostrar el menú principal, ya que es un bucle.

Opción 2

```
Elige una opción: 2
Nombre de usuario: Horacio
Contraseña: 1234567
Bienvenido cliente Horacio. Puedes realizar tus operaciones.
```

En la opción 2 (Inicio sesión) se le pedirá al usuario su nombre y contraseña. Se iniciará sesión con éxito si los datos fueron previamente registrados en la opción 1. Al iniciar sesión con éxito se le mostrara al usuario un mensaje de bienvenida, este mensaje variara dependiendo el usuario es Client o Admin.

```
--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 2
Nombre de usuario: ana
Contraseña: 123
Usuario o contraseña incorrecta.
```

Si se ingresan datos que no fueron previamente registrados por la opción 1. Al usuario le aparecerá un mensaje de usuario o contraseña incorrecta.

Opción 3

```
Elige una opción: 3
Haz Salido del Sistema
PS C:\Users\MI PC\Documents\Taller de POO(SENA)> |
```

En la opción 3 (Salir) es para salir del sistema. Ya que como lo mencione en su momento estamos en un bucle, donde hay que dar una condición para terminarlo. Ingresando esta opción el usuario saldrá del programa y este mismo le mostrara un mensaje de que salió del programa.

```
--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 1
Nombre de usuario: Horacio
Contraseña: 1234567
Tipo de usuario (admin/cliente): cliente
Usuario registrado con éxito.

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 2
Nombre de usuario: Horacio
Contraseña: 1234567
Bienvenido cliente Horacio. Puedes realizar tus operaciones.

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 2
Nombre de usuario: ana
Contraseña: 123
Usuario o contraseña incorrecta.

--- MENÚ PRINCIPAL ---
1. Registrar usuario
2. Iniciar sesión
3. Salir
Elige una opción: 3
Haz Salido del Sistema
PS C:\Users\MI PC\Documents\Taller de POO(SENA)> |
```

