

# Lenguaje de programación I

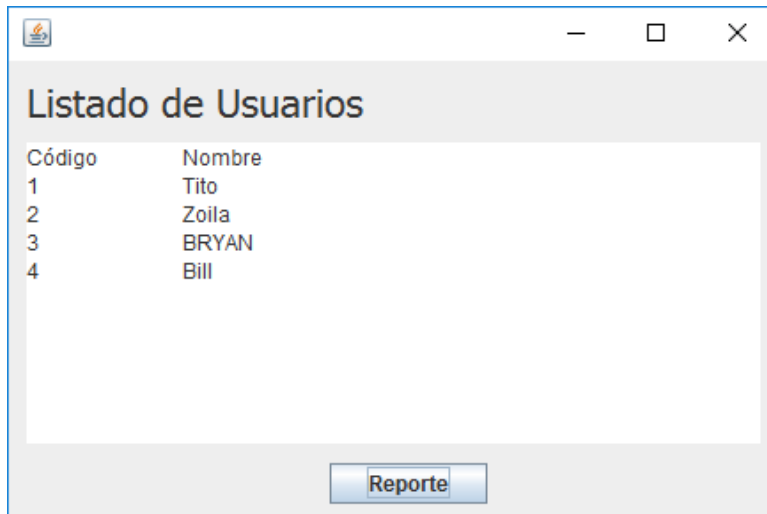
Tema 9: JDBC - Consultas



# Caso

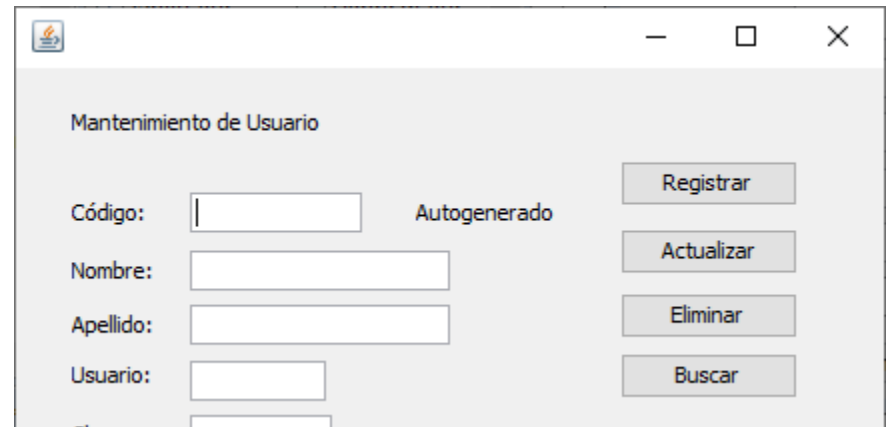


- El administrador de la empresa CiberFarma, necesita un **listado** de los usuarios del sistema. En la GUI de mantenimiento el botón **Buscar** debe devolver los datos del usuario, según el nombre, apellido o código ingresado



Código	Nombre
1	Tito
2	Zoila
3	BRYAN
4	Bill

Reporte



Mantenimiento de Usuario

Código:  Autogenerado

Nombre:

Apellido:

Usuario:

Clave:

Registrar

Actualizar

Eliminar

Buscar

# Contenido

- Las consultas o reportes
- Ejercicio: Listados
- Aplicaciones

## Logros de la Unidad

- Crear aplicaciones de manejo de bases de datos



# Las consultas o reportes

- Estos procesos involucran la ejecución de sentencias **SELECT**, obteniendo un objeto **ResultSet** que representa el resultado de la consulta.
- Ejemplos:

```
select * from tb_productos;
```

Result Grid

Filter Rows:

Export:

	idprod	descripcion	stock	precio	idtipo	estado
▶	P0001	Panadol cj 10	20	1.85	1	1
	P0002	Curitas unidad	100	1.00	3	1
	P0003	Kita tos	80	15.00	2	1
	P0004	Achiz	120	1.00	1	1
	P0005	Jaboncillo cj	120	1.00	3	1
	P0006	Termometro	80	2.80	5	1
	P0007	Panadol jarabe pq	40	10.50	2	1

```
select * from tb_usuarios  
where codigo = 1;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	codigo	nombre	apellido	usuario	clave	fnacim	id_tipo	estado
	1	luisa	perez	juan	perez	2020-06-12	3	0

## Importante.!!!

Para almacenar gran cantidad de información usaremos colecciones (**ArrayList**) de Clase.

# Las consultas o reportes

- Para obtener los datos de cada columna, usaremos métodos de acceso get del conjunto de resultados:
- Ej. Para obtener el código y el precio

rs =

Result Grid						
Filter Rows: <input type="text"/>						
Export: 						
	idprod	descripcion	stock	precio	idtipo	estado
▶	P0001	Panadol cj 10	20	1.85	1	1
	P0002	Curitas unidad	100	1.00	3	1
	P0003	Kita tos	80	15.00	2	1
	P0004	Achiz	120	1.00	1	1
	P0005	Jaboncillo cj	120	1.00	3	1
	P0006	Termometro	80	2.80	5	1
	P0007	Panadol jarabe pq	40	10.50	2	1

`rs.getString(1)`

`rs.getDouble(4)`

# Estructura del Proyecto con BD

## ▼ LPI-semana07-plantilla

### ▼ src

#### ▼ interfaces

> UsuarioInterface.java

#### ▼ mantenimientos

> GestionUsuario.java

#### ▼ model

> Usuario.java

#### ▼ utils

> MySQLConexion.java

#### ▼ vista

> Reporte.java

> Reporte2.java

> ReportexFecha.java

> JRE System Library [JavaSE-1.8]

> jcalendar-1.4.jar

### ▼ bd

bd.sql

### ▼ libs

jcalendar-1.4.jar

mysql-connector-java-5.1.6-bin.jar

1

Listado de métodos para el mantenimiento

3

Implementación de los métodos de mantenimiento

2

**Entidades** utilizadas para transferir los datos desde la base de datos hacia la capa lógica y viceversa

# Aplicaciones. Listado de usuarios



- El proceso de reporte mostrará el listado obtenido de la clase de Gestión

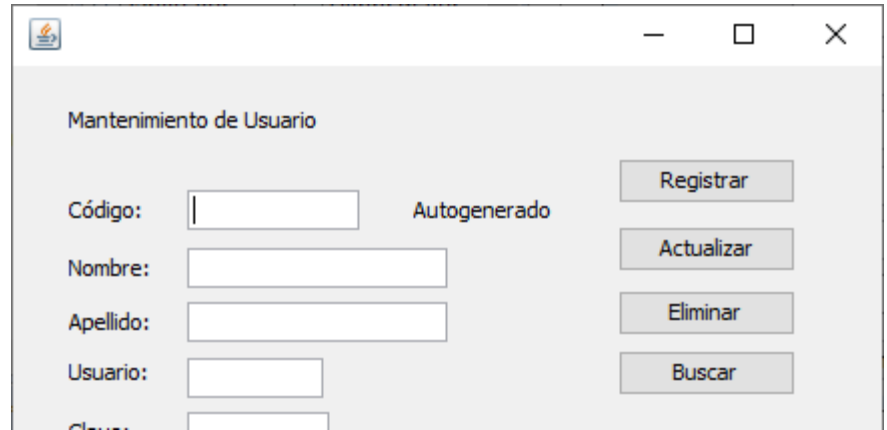
Listado de Usuarios	
Código	Nombre
1	Tito
2	Zoila
3	BRYAN
4	Bill

Reporte

```
void listado() {  
    // Instancia la clase de gestion  
    GestionUsuario gu = new GestionUsuario();  
  
    // Obtiene el listado de usuarios -> ArrayList  
    ArrayList<Usuario> lista = gu.listado();  
  
    // imprime la lista en el área de texto  
    if (lista == null) {  
        txtS.setText("Lista vacía");  
    } else {  
        txtS.setText("Código\tNombre\n");  
        for (Usuario u : lista) {  
            txtS.append(u.getCodigo() + "\t" + u.getNombre() + "\n");  
        }  
    }  
}
```

# Aplicaciones. Consulta de datos

- Al pulsar el botón, se obtiene un objeto con todos los datos del usuario, si es que existe, el cual se debe colocar en los campos respectivos



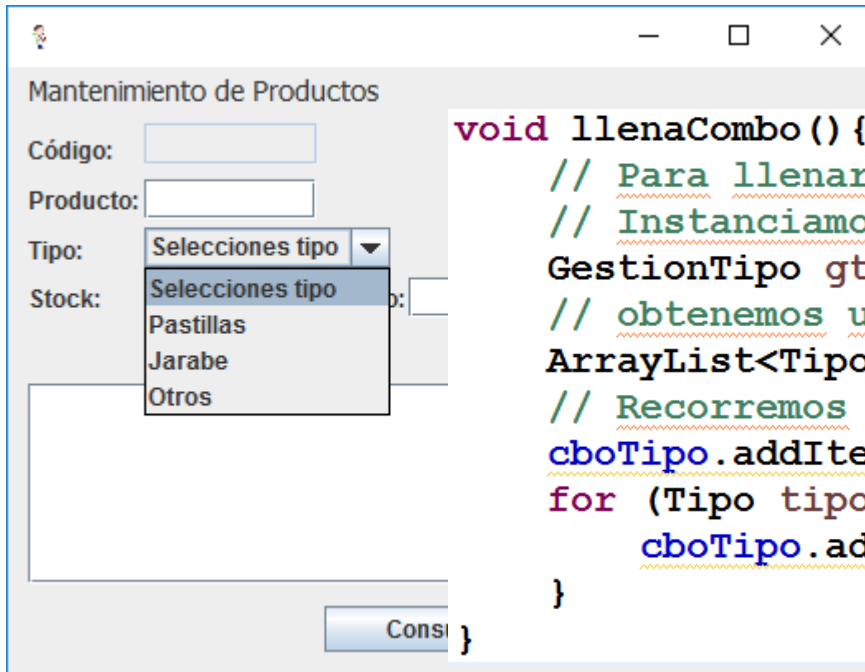
The screenshot shows a web application window titled "Mantenimiento de Usuario". It contains a form with the following fields and buttons:

- Código:** A text input field with a small "Autogenerado" label to its right.
- Nombre:** A text input field.
- Apellido:** A text input field.
- Usuario:** A text input field.
- Clave:** A text input field (partially visible).
- Buttons:** Four buttons are stacked vertically on the right side: "Registrar", "Actualizar", "Eliminar", and "Buscar".



# Aplicaciones. Combo de Categorías

- Realizar listados nos permite emplearlo en diversos elementos como ComboBox.
- Ej. Para llenar el combo de categorías (tipos) de productos en una farmacia:



Mantenimiento de Productos

Código:

Producto:

Tipo:

Stock:

Pastillas

Jarabe

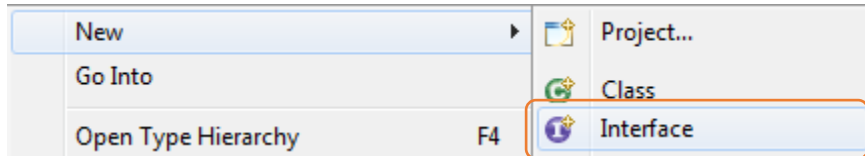
Otros

Conservar

```
void llenaCombo() {  
    // Para llenar un combo de la BD  
    // Instanciamos la clase de Gestión  
    GestionTipo gt = new GestionTipo();  
    // obtenemos una lista con los datos de la tabla  
    ArrayList<Tipo> lista = gt.listado();  
    // Recorremos el arreglo para agregar los elementos  
    cboTipo.addItem("Seleccione tipo");  
    for (Tipo tipo : lista) {  
        cboTipo.addItem(tipo.getDescripcion());  
    }  
}
```

# Paso 1. Interfaces

- La interface, **define** los métodos de mantenimiento a implementar (registrar, actualizar, eliminar, listar, etc.) Ej. Para listado de Usuarios



- Ej. Para listado de Usuarios



```
UsuarioInterface.java X
1 package interfaces;
2
3 public interface UsuarioInterface {
4
5     // lista el contenido de la tabla usuario
6     public ArrayList<Usuario> listado();
7
8     // lista el contenido de la tabla usuario,
9     según el tipo
10    public ArrayList<Usuario> listado(int tipo);
11
12 }
```

## Paso 2. Entidades

- los listados, toman como referencia la tabla o consultas de donde obtendremos la información.
- Ej: Para el listado de toda la tabla **usuarios**:



```
SELECT * FROM tb_usuarios;
```

codigo	nombre	apellido	usuario	clave	facceso	tipo	estado
1	Tito	Siber	U001	10001	2017-06-03	2	1
2	Zoila	Baca	U002	10002	2017-06-03	2	1

Usuario.java

```
package beans;

public class Usuario {
    private int codigo, tipo, estado;
    private String nombre, apellido, usuario, clave, facceso;
```

6

- Ej: Para un listado de las **ventas** mostrando los **productos**:

```
SELECT numvta, fechavta, v.idprod, p.descripcion as 'Producto',
       (v.cantidad * v.preciovta) as 'importe'
FROM tb_ventas v
INNER JOIN tb_productos p ON v.idprod = p.idprod;
```

numvta	fechavta	idprod	Producto	importe
V0001	2017-04-15	P0001	Panadol cj 10	1.50
V0001	2017-04-15	P0004	Achiz	2.00
V0002	2017-04-25	P0002	Curitas unidad	1.00

Reporte1.java

```
1 package beans;
2
3 public class Reporte1 {
4     private String numvta, fechavta, idprod, nomprod;
5     private double importe;
```

# Paso 3. Clase de Control o Gestión



```
public ArrayList<Usuario> listado() {  
    ArrayList<Usuario> lista = new ArrayList<Usuario>();  
    ResultSet rs = null;  
    Connection con = null;  
    PreparedStatement pst = null;  
    try {  
        con = MySQLConexion.getConnection();  
        String sql = "select * from tb_usuarios"; // sentencia sql  
  
        pst = con.prepareStatement(sql);  
        // parámetros según la sentencia  
  
        rs = pst.executeQuery(); // tipo de ejecución  
        // Acciones adicionales en caso de consultas  
        while (rs.next()) { // mientras hay datos en la consulta  
            Usuario u = new Usuario();  
            u.setCodigo(rs.getInt(1));  
            u.setNombre(rs.getString(2));  
            // otros campos  
            lista.add(u);  
        }  
    } catch (Exception e) {  
        System.out.println("Error en la sentencia " + e.getMessage());  
    } finally {  
        try {  
            if (pst != null) pst.close();  
            if (con != null) con.close();  
        } catch (SQLException e) {  
            System.out.println("Error al cerrar ");  
        }  
    }  
    return lista;  
}
```

- El método `next()` del `ResultSet`, hace que el puntero avance al siguiente registro. Si lo consigue devuelve `true`.

	codigo	nombre	apellido	usuario	clave	facceso	tipo	estado
	1	Tito	Siber	U001	10001	2017-05-13	2	1
	2	Zoila	Baca	U002	10002	2017-05-13	2	1



# Referencias

- <https://www.youtube.com/watch?v=uUzQewzCRiU>
- <https://hashblogeando.wordpress.com/2014/02/12/mostrar-datos-de-una-tabla-mysql-con-java-swing/>



# 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: 633-5555

## **SEDE BELLAVISTA**

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

## **SEDE AREQUIPA**

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