



El futuro digital
es de todos

MinTIC

Hechos

QUE

CONECTAN



CICLO 2

EJE TEMÁTICO 5

**INTERFACES GRÁFICAS
Y BASES DE DATOS**

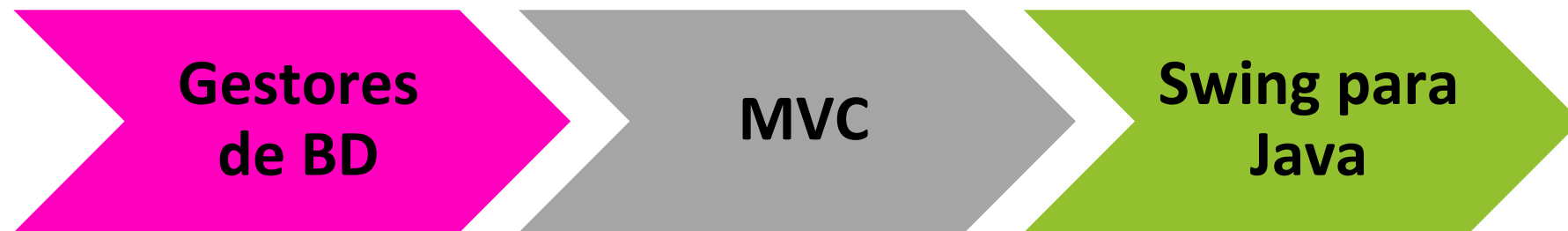
Universidad
Industrial de
Santander



**Misión
TIC 2022**

Interfaces gráficas y base de datos

Ruta de aprendizaje



Introducción

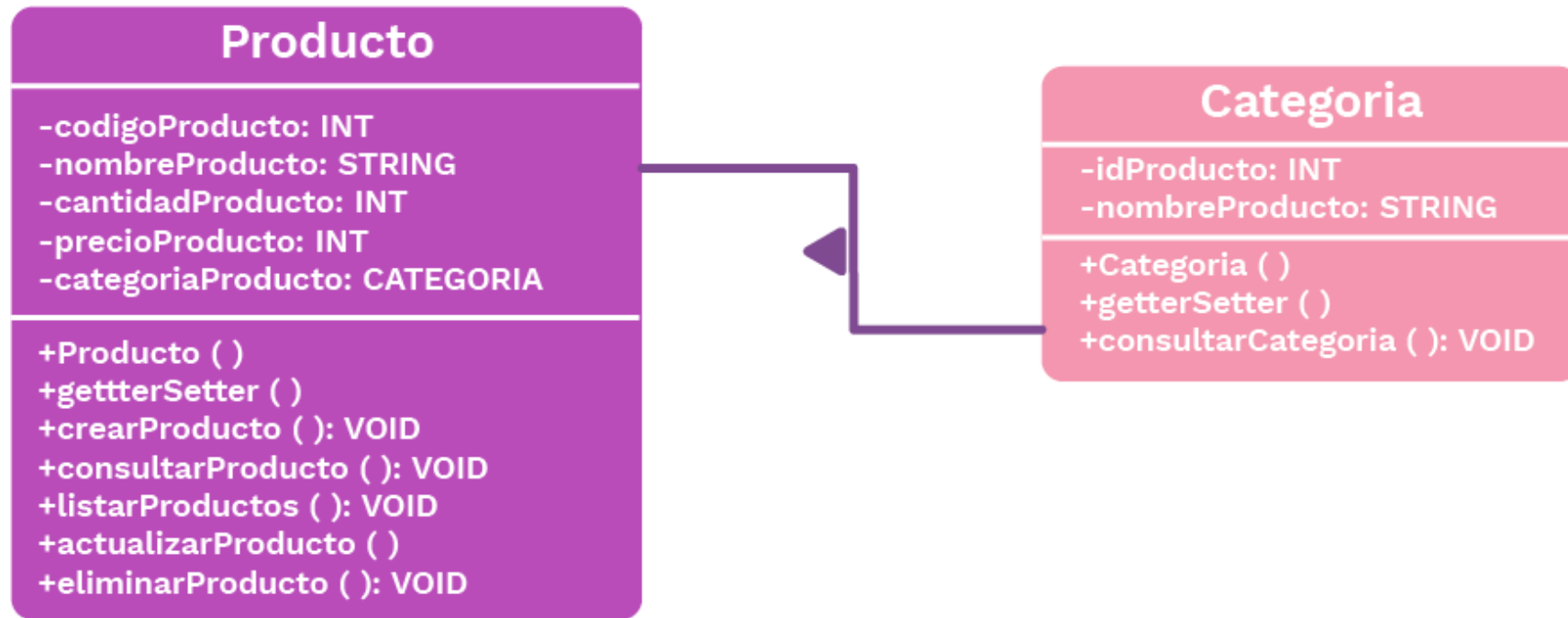
En este recurso encontrará los conceptos y términos que requiere para poder realizar una Java application que tenga una interfaz gráfica y está maneje información a través de un gestor de base de datos. Para esto se revisará **Modelos, JAVA SWING y Conexión a BD.**

Modelamiento de datos

Para este último eje temático, será necesario conocer cómo se debe hacer de forma correcta la recopilación de datos, entre esto tenemos entrevistas, encuentros y reuniones con los dueños del producto.



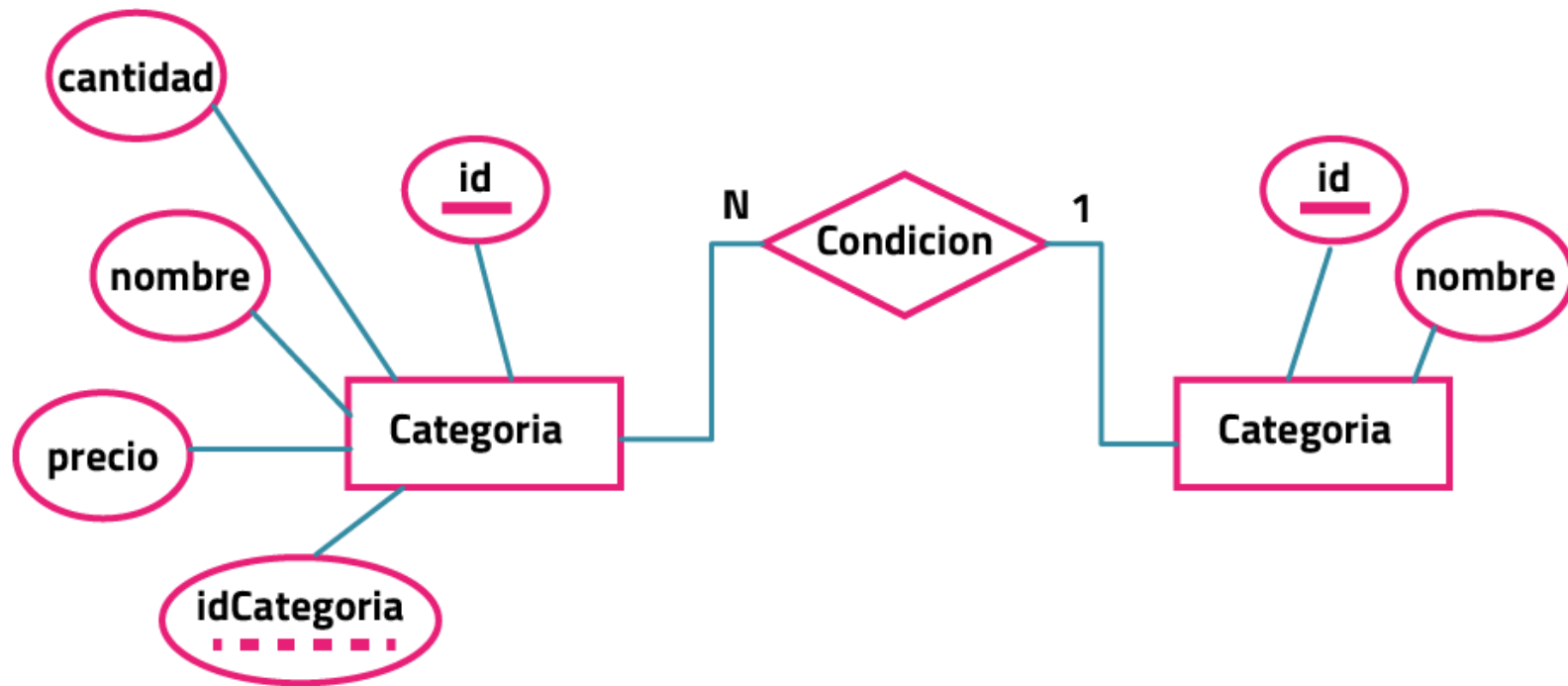
Conocer esta información nos permitirá conocer cómo se estructura la información que vamos a utilizar y como se puede organizar en un diagrama de clases.



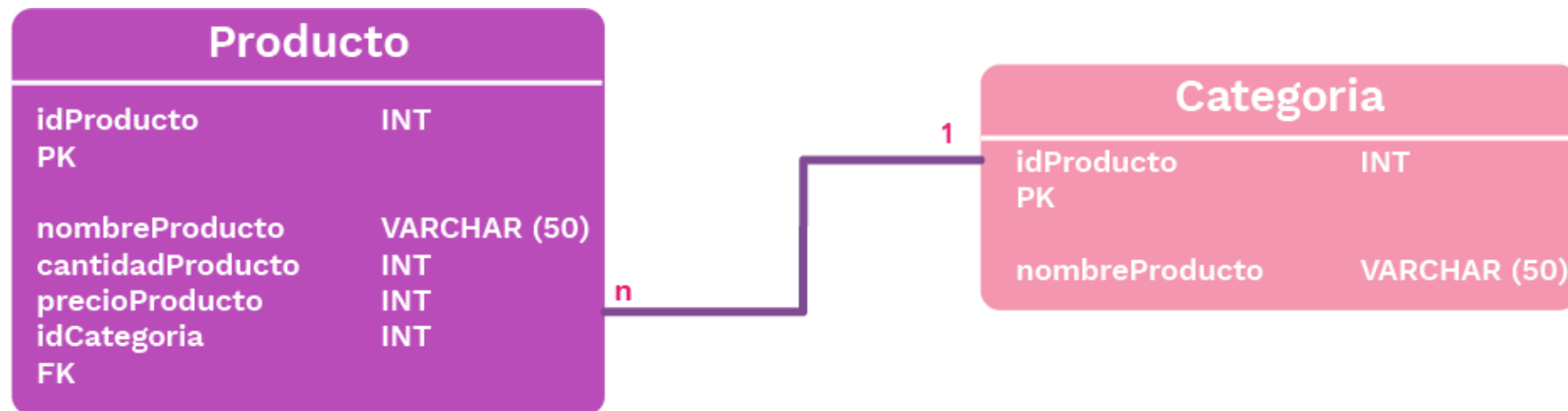
Modelamiento, ER

Posterior a plantear un diagrama de clases que nos permite estructurar nuestro problemas con Clases y relaciones, es necesario analizar cómo organizar la información que se va a mostrar en la base de datos, para ello podemos realizar el modelamiento a través del Modelo Entidad Relación ER.

1.



2.



5. Desarrollo de bases de datos

- Es un sistema de software que no cuenta con interfaz gráfica.
- También es denominado como un complemento para el SGBD.
- Se encarga de la definición y manipulación de los espacios de almacenamiento que implementa las base de datos.



ORACLE



PostgreSQL



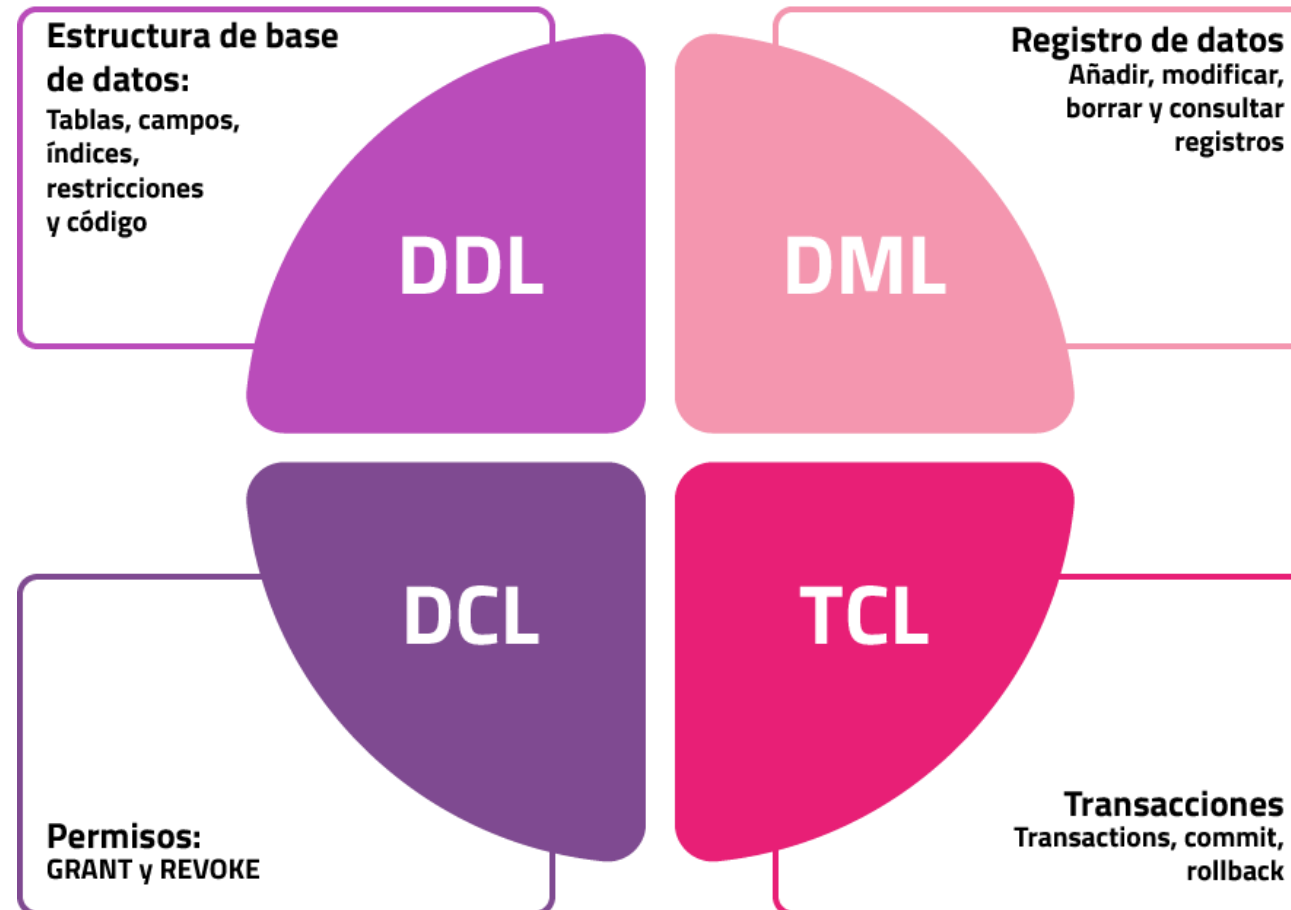
5.1. Interfaces de gestión de BD

- Provee una interfaz de gestión y manipulación para el servicio o motor de base de datos.



**DB Browser for
SQLite**

5.2. Operaciones en bases de datos relacionales



5.3. BD SQLite

Para el desarrollo de este ciclo, utilizaremos SQLite, que es un sistema de gestión de bases de datos liviano y que no demanda mayor dificultad para utilizar debido a que es compatible con las características ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), y que basa su funcionamiento en acceso a una base de datos tipo archivo portable, ideal para aplicaciones que no requieran un proceso independiente para comunicarse con el programa principal.

Para este ciclo, la única herramienta que debemos instalar es DB Browser for SQLite, la cual puede ser descargada en el siguiente enlace:

<https://sqlitebrowser.org/>

5.4. Tipos de datos en BD

En las bases de datos como en las clases, manejaremos información que puede ser datos enteros, datos de tipo texto, flotantes, entre otros, en estos momentos se presenta los tipos de datos que podemos encontrar con SQLITE.



Ejemplos de tipos de datos para creación de tablas o expresiones de cambios de tipo

INT,INTEGER
TINYINT
SMALLINT
MEDIUMINT
BIGINT
UNSIGNED BIG INT
INT2
INT8

CHARACTER(20)
VARCHAR(255)
VARYING CHARACTER(255)
NCHAR(55)
NATIVE CHARACTER(70)
NVARCHAR(100)
TEXT
CLOB

BLOB (Tipo de dato no especificado)

REAL
DOUBLE
DOUBLE PRECISION
FLOAT

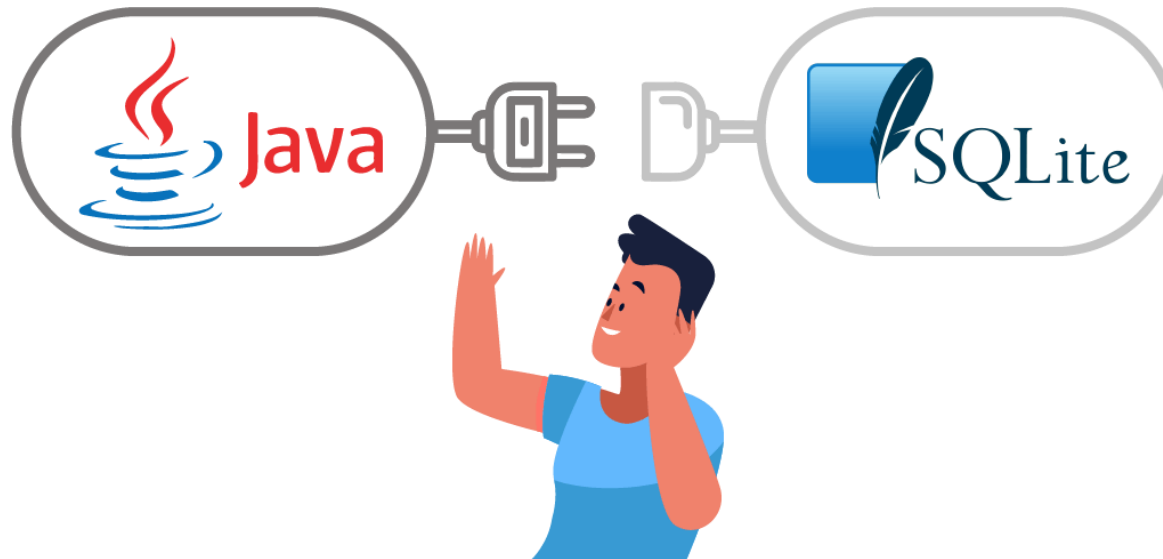
NUMERIC
DECIMAL(10,5)
BOOLEAN
DATE
DATETIME

tomado de <https://www.sqlite.org/datatype3.html>

5.5. Lenguaje de manipulación CRUD

Java Database Connector:

Es un puente que permite conectar la aplicación java con la base de datos



Sigla CRUD	Operación	Comando	Estructura
C reate	Creación de un nuevo registro en base de datos	INSERT	<code>INSERT INTO nombreDeTabla(columna1, columna2, columna3, ...) VALUES (valor1, valor2, valor3, ...);</code>
R ead	Lectura de registros de base de datos	SELECT	<code>SELECT columna1, columna2, ... FROM nombreDeTabla; SELECT * FROM table_name;</code>
U pdate	Actualización de registros de base de datos	UPDATE	<code>UPDATE nombreDeTabla SET columna1= valor1, columna2= valor2, ... WHERE condition;</code>
D elete	Eliminación de registros de base de datos	DELETE	<code>DELETE FROM nombreDeTabla WHERE condicion;</code>

Tomado de <https://www.sqlite.org/datatype3.html>

Revisión de lo aprendido

1. Lo primero que se debe hacer cuando se va a desarrollar un proyecto es:
 - a. Recolección de datos y entrevistas
 - b. Iniciar a codificar con mis propias ideas
 - c. Iniciar a codificar con la idea mencionada por el dueño del product
 - d. Diseñar el diagrama ER

Revisión de lo aprendido

2. El modelo ER es un modelo entidad-relación es una herramienta para el modelo de datos, la cual facilita la representación de entidades de una base de datos.

- a. Verdadero
- b. Falso

Revisión de lo aprendido

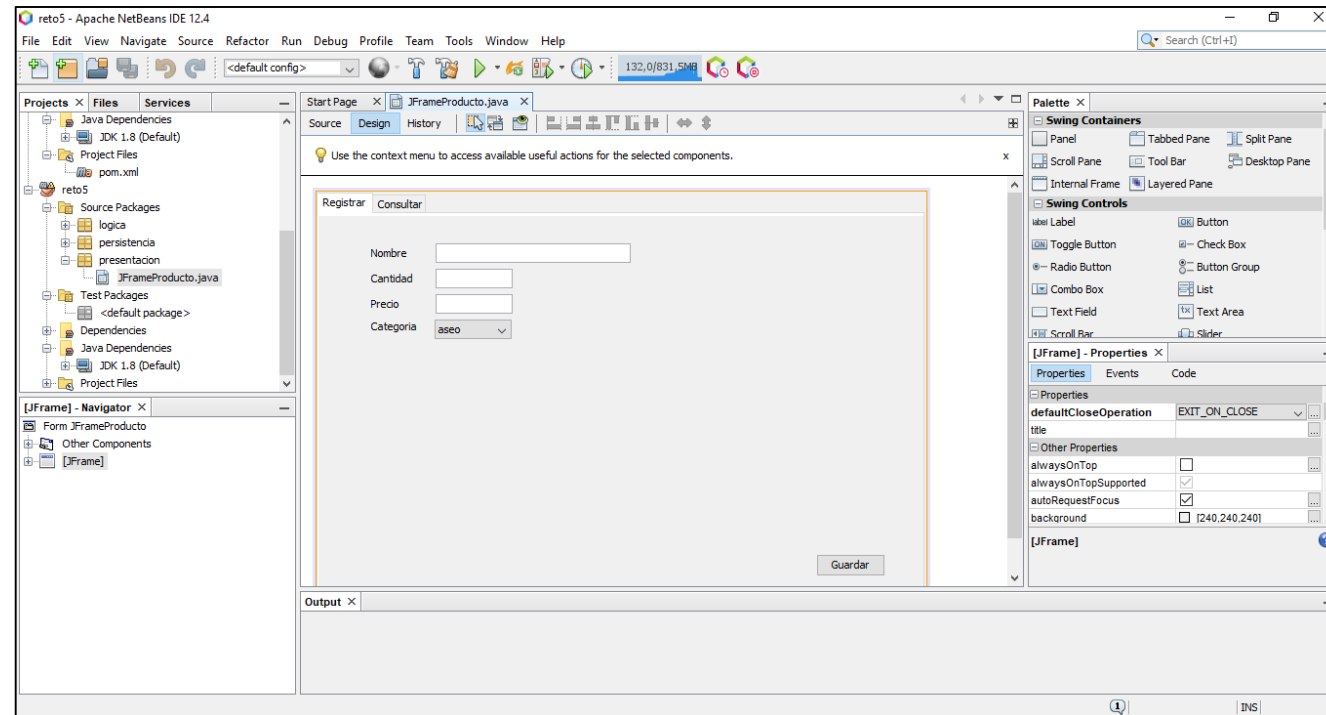
3. Entre las operaciones en bases de datos, las DDL corresponden a:
- a. Estructuras de base
 - b. Registro de datos
 - c. Permisos
 - d. Transacciones

Revisión de lo aprendido

4. CRUD es el acrónimo de crear, leer, actualizar y borrar en bases de datos:
- a. Verdadero
 - b. Falso

5.6. Interfaces gráficas

Las interfaces gráficas (GUI, del inglés Graphical User Interface) ofrecen al usuario botones, menú, barra de herramientas, ventanas, cuadros de diálogo y similares, que conocemos en nuestra interacción como usuarios. Un ejemplo de interfaz gráfica o GUI, podría ser un procesador de texto como lo es Microsoft Word o el bloc de notas.



5.7. Swing

Swing es una biblioteca de clases que nos ofrece la capacidad de construir interfaces gráficas para aplicaciones en escritorio en Java.

Swing nos provee de dos elementos principales para la construcción de interfaces gráficas:

- Contenedores:** Como su nombre lo dice, contienen un conjunto de elementos o componentes.
- Componentes:** Elementos gráficos como botones, cuadros de texto, que se pueden organizar en contenedores.

Existen tres tipos de contenedores de alto nivel: JFrame, JDialog y JApplet.

Ya que estos son de alto nivel, cualquier otro contenedor o componente debe construirse en su interior.

Para usar esta librería debemos importarla de la siguiente manera:

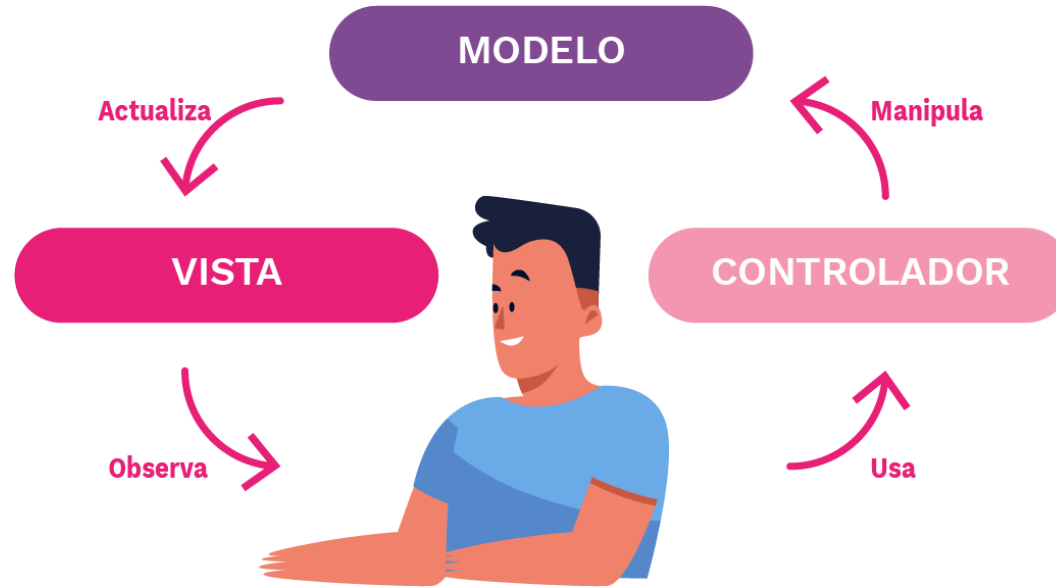
```
import javax.swing.*;
```

5.8. Modelo MVC con swing

El **modelo MVC** es un diseño de software que propone diferenciar el código programado en relación con el rol que cumple en la aplicación.

Se proponen los siguientes roles del código en una aplicación MVC:

- **Modelo:** Es la representación de la lógica de toda la aplicación.
- **Vista:** Es la representación del modelo (información y lógica del negocio)
- **Controlador:** Responde a los eventos realizados por el usuario desde la interfaz



Se tiene de primera mano al usuario, este usuario hace una solicitud al controlador, quien se comunica con el modelo para obtener la información. Una vez procesada la información, el modelo le devuelve esa información procesada al controlador, luego el controlador va a enviar esta información a la vista. Finalmente, el usuario va a tener una respuesta visual de su solicitud.

5.9. Modelo vista controlador – Frameworks

Conjunto de herramientas y utilidades para facilitar el desarrollo de un proyecto.

-Es un framework para mejorar la seguridad y optimizar la operación de desarrollado.

-Contiene un tipo de proyecto Spring MVC que implementa una herramienta que facilita la interacción entre las vistas y los controladores denominada front controller.



- Es un framework que facilita la interacción entre el sistema y la base de datos, realizando un mapeo de los campos de una base de datos relacional, con el fin de manipularlos por medio de objetos.
- Optimiza el flujo de los datos entre los modelos y la base de datos.

5.10. Pruebas Unitarias

- Permite el desarrollo de pruebas específicas a métodos, generando casos de prueba.
- Implementa un framework llamado **junit**.
- Permite encontrar errores específicos o problemas en nuestros algoritmos.



```
public class Calculadora {  
  
    public Calculadora() {  
  
    }  
  
    public int suma(int num1, int num2) {  
        return num1 + num2;  
    }  
  
    public int resta(int num1, int num2) {  
        return num1 - num2;  
    }  
  
    public int multiplicacion(int num1, int num2) {  
        return num1 * num2;  
    }  
}
```

```

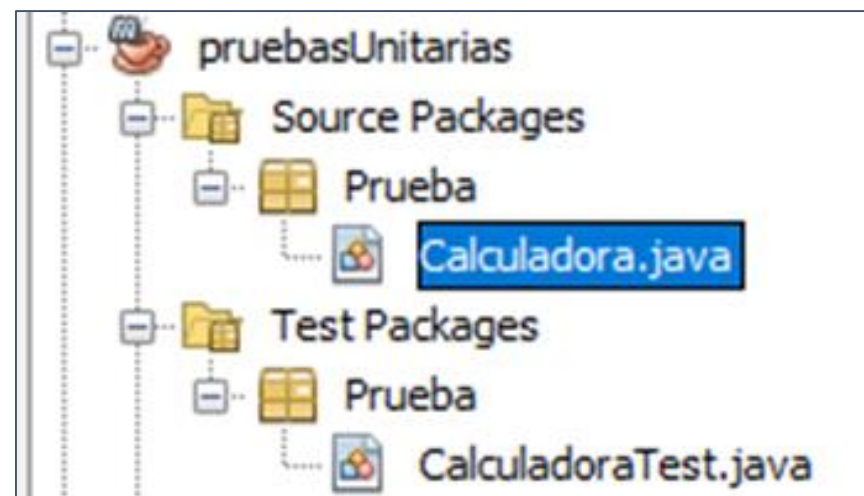
public class CalculadoraTest {

    public CalculadoraTest() {
    }

    @Test
    public void testSuma() {
        Calculadora suma = new Calculadora();
        int resultado = suma.suma(8, 12);
        int esperado = 20;
        assertEquals(esperado, resultado);
    }

    @Test
    public void testResta() {
        Calculadora resta = new Calculadora();
        int resultado = resta.resta(25, 8);
        int esperado = 17;
        assertEquals(esperado, resultado);
    }
}

```



Revisión de lo aprendido

1. Las interfaces gráficas facilitan el trabajo del desarrollador porque tienen predefinidas instrucciones para la vista del desarrollo a realizar, esto es:

- a. Verdadero
- b. Falso

Revisión de lo aprendido

2. El modelo vista controlador propone:

- a. Diferenciar el código programado en relación con el rol que se cumple en la aplicación.
- b. Facilita el Desarrollo
- c. Es un framework
- d. Es un tipo de interfaz gráfica.

Revisión de lo aprendido

3. Los frameworks proponen múltiples ventajas. Una de ellas es:
- a. Herramientas para facilitar el desarrollo de un proyecto: seguridad y interacción con BD
 - b. Estructurar el código de forma más clara
 - c. Mejorar la rapidez del software
 - d. Modelar los diagramas de clase de forma más clara

Revisión de lo aprendido

4. Las pruebas unitarias permiten desarrollar pruebas específicas a métodos, generando casos de prueba.

- a. Verdadero
- b. Falso



El futuro digital
es de todos

MinTIC

Hechos

QUE

CONECTAN ✓

CICLO 2

EJE TEMÁTICO 5

**INTERFACES GRÁFICAS
Y BASES DE DATOS**

Universidad
Industrial de
Santander



Mision
TIC 2022