



## MANUAL DE USUARIO

Nombres:

Eduardo Alexander Reyes Gonzalez

Carnet: 202010904

Randy Marcelino Sum Coyoy

Carnet: 2253625030901

Fecha: 29 de diciembre de 2023



## Manual de Usuario -XSQL

### Introducción

Bienvenido XSQL proyecto de compiladores, un sistema de administración de bases de datos realiza en python con la herramienta PLY aqui encontraras informacion sobre instalación, navegación por la interfaz, creación y manipulación de bases de datos, así como la ejecución de consultas SQL

### Instalación

Puedes instalar python en tu sistema desde [python.org](https://python.org)  
Descarga el código fuente desde el repositorio en [github](https://github.com)

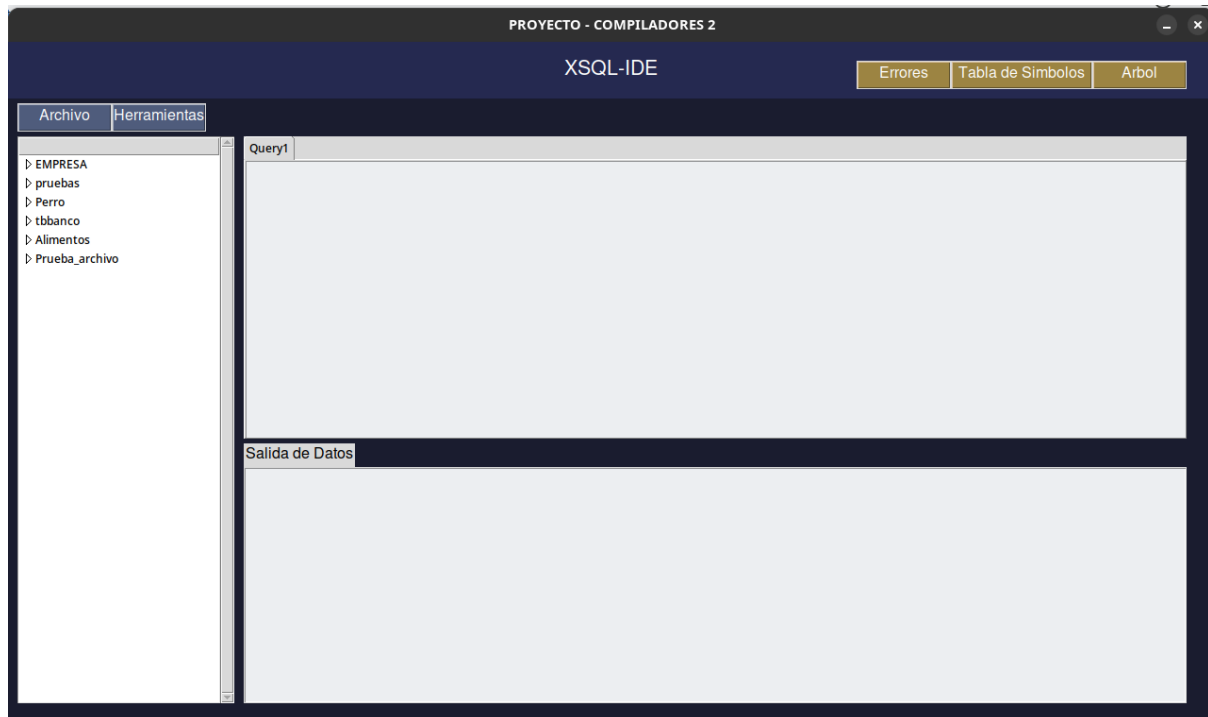
### Iniciar XSQL

#### Ejecutar el IDE

Abre tu gestor de código favorito, inicia una nueva terminal y ejecuta el comando:  
`python interfaz.py`



## Interfaz del IDE

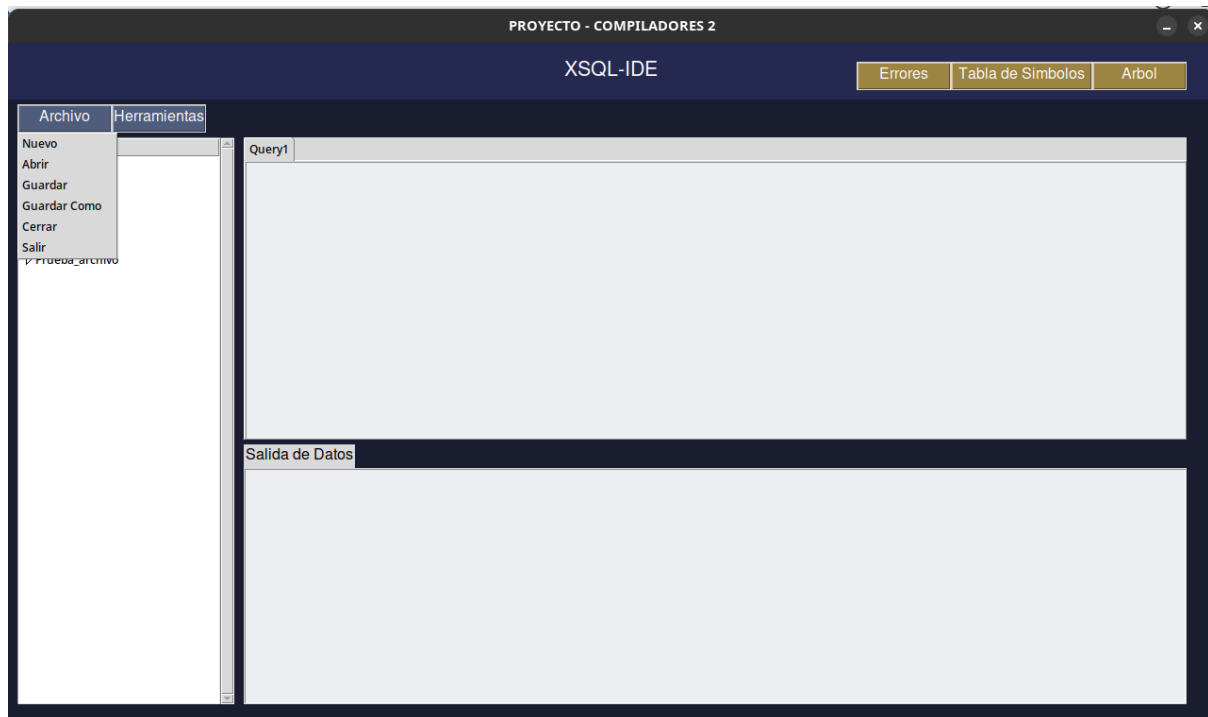


XSQL presenta una interfaz amigable con menús desplegables para acceder a las distintas funciones del sistema, tiene un editor de código que permite ingresar las instrucciones para realizar las distintas operaciones, tiene una salida de datos que muestra la información sobre los procesos que se están ejecutando.

## Menús principales

### Archivo

Permite gestionar archivos(Nuevo, Abrir, Guardar, Guardar Como, Cerrar y Salir) de un nuevo archivo de base de datos



## Herramientas

Proporciona las distintas funciones para gestionar las bases de datos, como crear base de datos, eliminar, crear DUMP

## Bases de datos

A un costado podemos ver las bases de datos que están guardadas en nuestra aplicación, podemos ver sus distintas tablas también

## Errores

La aplicación crea una tabla cuando algún procedimiento contiene errores

## Tabla de símbolos

La aplicación muestra una tabla de símbolos con los tokens utilizados

## Árbol

La aplicación muestra un árbol de la gramática.



## Creación y manejo de bases de datos

El lenguaje XSQL es un lenguaje de bases de datos. Este lenguaje se divide en 4 partes: DML, DDL, DCL y SSL.

Para la ejecución de las distintas instrucciones nos situamos en el editor de código y podremos realizar las distintas ejecuciones, es importante que después de escribir una instrucción selecciones en herramientas -> SQL -> Ejecutar QUERY.

### Tipos de datos

Entre los distintos tipos de datos que XSQL acepta son los siguientes

Grupo	Tipo de Dato	Descripción	Ejemplo
Numéricos exactos	INT	Tipo de dato que maneja valores numéricos enteros. De -2.147.483.648 a 2.147.483.647	1, 50, 1000
	BIT	Tipo de datos entero que puede aceptar los valores 1, 0 o NULL	1 0
	DECIMAL	Tipo de dato que maneja valores numéricos decimales o enteros. Rango entre -2.147.483.648 a 2.147.483.647	5, 5.5, 12.5
Fecha y hora	DATE	Tipo de dato para almacenar fechas en formato dd-mm-yyyy	26-05-2005
	DATETIME	Tipo de dato para - almacenar la fecha y la - hora en formato dd-mmyyyy hh:mm:ss	26-05-2005 22:50
Cadenas de caracteres	NCHAR(n)	Dato carácter de longitud fija, con n caracteres, n debe estar entre 1 y 4,000	"Hola" "Adios"
	NVARCHAR	Dato carácter de longitud variable, el valor máximo de n puede ser 2,000,000	"HolaMundo"

### Expresiones Aritméticas:

Tiene operaciones aritméticas básicas como suma, resta multiplicación y división, que se pueden realizar con los distintos tipos de datos.



	Bit	Int/Decimal	Date/Datetime	Nchar/Nvarchar
<b>Suma(+)</b>				
<b>Bit</b>	OR	Suma	Error	Concatenar
<b>Int/Decimal</b>	Suma	Suma	Error	Concatenar
<b>Date/Datetime</b>	Error	Error	Error	Error
<b>Nchar/Nvarchar</b>	Concatenar	Concatenar	Error	Concatenar
<b>Resta(-)</b>				
<b>Bit</b>	Error	Resta	Error	Error
<b>Int/Decimal</b>	Resta	Resta	Error	Error
<b>Date/Datetime</b>	Error	Error	Error	Error
<b>Nchar/Nvarchar</b>	Error	Error	Error	Error
<b>Multiplica (*)</b>				
<b>Bit</b>	AND	Multi	Error	Error
<b>Int/Decimal</b>	Multi	Error	Error	Conca
<b>Date/Datetime</b>	Error	Error	Error	Conca
<b>Nchar/Nvarchar</b>	Error	Error	Concatena	Error
<b>División (/)</b>				
<b>Bit</b>	Error	Div	Error	Error
<b>Int/Decimal</b>	Div	Div	Error	Error
<b>Date/Datetime</b>	Error	Error	Error	Concatena
<b>Nchar/Nvarchar</b>	Error	Error	Conca	Error

Expresiones  
Relacionales

Nombre	Símbolo	Descripción
Igual	==	Esta operación comprueba si dos expresiones tienen el mismo valor, de ser así retorna 1 (verdadero) de lo contrario retorna 0 (falso).
Diferente	!=	Esta operación comprueba si dos expresiones tienen distinto valor, de ser así retorna 1 (verdadero) de lo contrario



		retorna 0(falso).
Menor que	<	Esta operación comprueba si la primera expresión es menor a la segunda expresión, de ser así retorna 1 (verdadero) de lo contrario retorna 0(falso).
Mayor que	>	Esta operación comprueba si la primera expresión es mayor a la segunda expresión, de ser así retorna 1 (verdadero) de lo contrario retorna 0(falso).
Menor igual	<=	Esta operación comprueba si la primera expresión es menor o igual a la segunda expresión, de ser así retorna 1 (verdadero) de lo contrario retorna 0(falso).
Mayor igual	>=	Esta operación comprueba si la primera expresión es mayor o igual a la segunda expresión, de ser así retorna 1 (verdadero) de lo contrario retorna 0(falso).

## Expresiones Lógicas

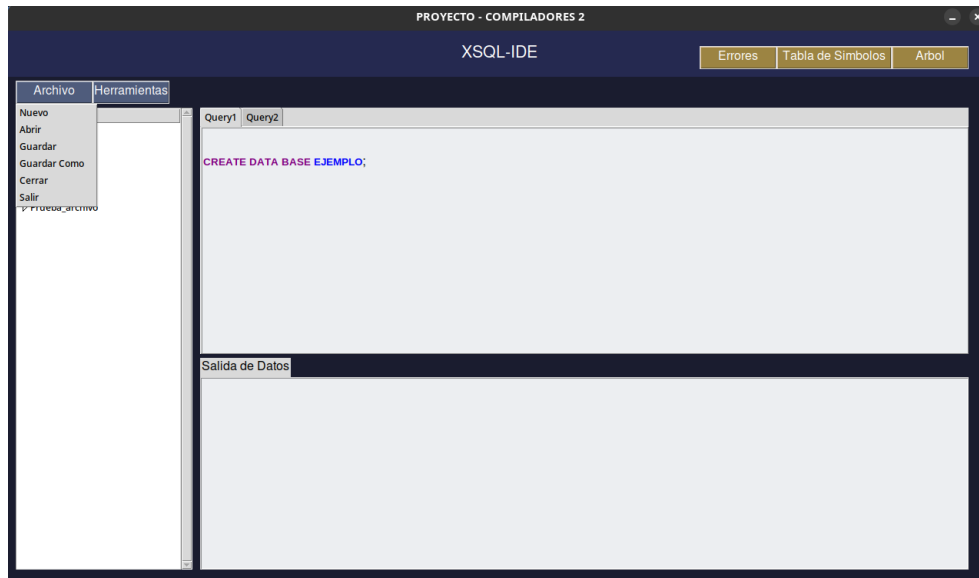
Nombre	Símbolo	Descripción
AND	&&	Este operador comprueba que tanto la primera expresión como la segunda expresión tengan valor verdadero. Si esto se cumple retorna 1 (verdadero)
OR		Este operador comprueba que la primera expresión o la segunda expresión tengan valor verdadero. Si esto se cumple retorna 1 (verdadero) de lo contrario retorna 0(falso).
NOT	!	Este operador cambia el valor de la expresión de la que viene acompañada.

## Create

### Create Data Base

Permite la creación de una base de datos  
 la instrucción a utilizar es la siguiente:

```
CREATE DATA BASE Prueba_archivo;
```



### Create Table

Para la creación de una tabla utilizamos el comando

El nombre de la tabla no debe contener caracteres especiales (+, -, \*, /, %, \$, %, (, ), ", ! ?), se puede utilizar \_ en el nombre

Debemos tener seleccionada una base de datos con la instrucción USE

```
USE DATABASE;
```

```
CREATE TABLE "nombre_tabla";
```

Esto nos permite crear una tabla dentro de nuestra base de datos

Podemos agregarle distintos atributos con sus respectivos tipos de datos, por ejemplo nombre tipo NVARCHAR(10) y así.

Podemos utilizar palabras como:

NULL si especificamos que puede aceptar el valor nulo en este campo.

Primary Key: Se utiliza para especificar si el campo será llave principal de la tabla

Foreign Key: Se utiliza para especificar si el campo será llave foránea. Seguido del nombre de la tabla a la que hace referencia.

Un ejemplo de una instrucción create table es el siguiente:

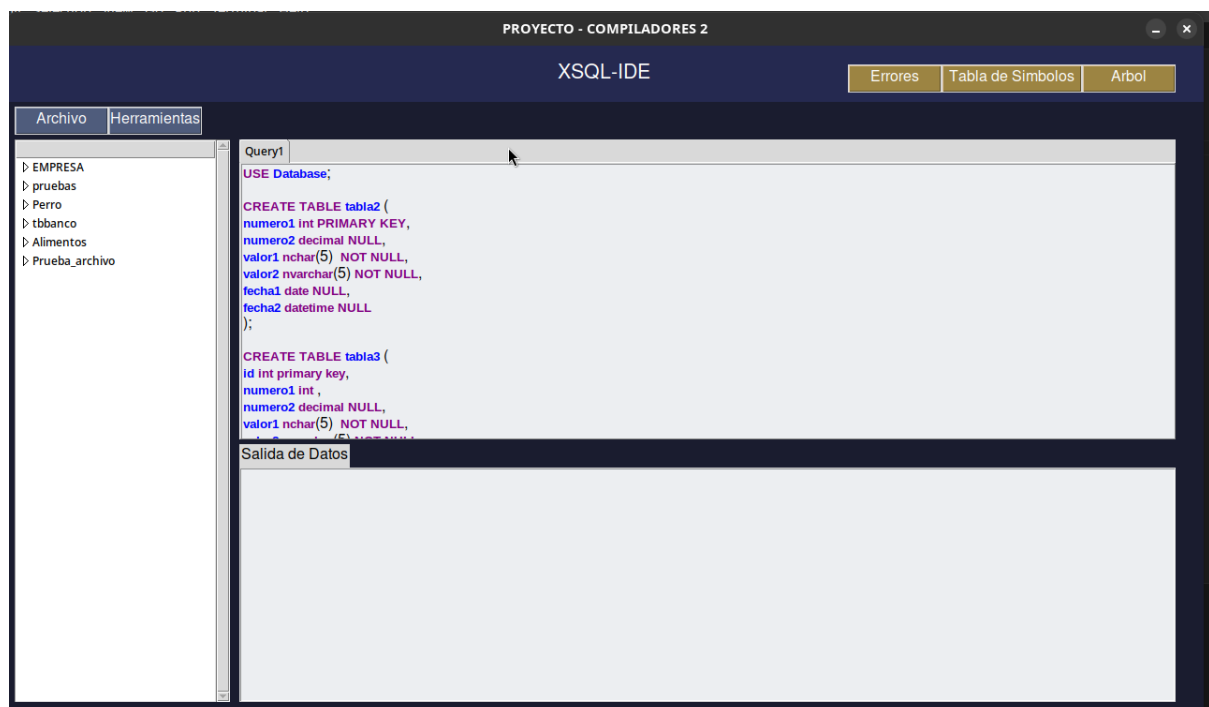
```
CREATE TABLE tabla1 (  
  numero1 int PRIMARY KEY,  
  numero2 decimal NULL,  
  valor1 nchar(5) NOT NULL,  
  valor2 nvarchar(5) NOT NULL,  
  fecha1 date NULL,
```





```
fecha2 datetime NULL
);

CREATE TABLE tabla2 (
id int primary key,
numero1 int ,
numero2 decimal NULL,
valor1 nchar(5) NOT NULL,
valor2 nvarchar(5) NOT NULL,
fecha1 date NULL,
fecha2 datetime NULL,
FOREIGN KEY(numero1) REFERENCE tabla2(numero1)
);
```



### Create Function

Una función es un objeto dentro del motor que permite ejecutar sentencias XSQL y retornar un valor, pueden ser llamadas dentro de otras sentencias XSQL como por ejemplo dentro de un SELECT

ejemplo:



```
CREATE FUNCTION funcion1
RETURN INT
AS
BEGIN
    DECLARE @VALOR1 int;
    SET @VALOR1 = 5;
    RETURN @VALOR1;

END;
```

### Create Procedure

Un procedimiento es un objeto dentro del motor en el que se permite ejecutar sentencias XSQL, consultas, actualizaciones, inserciones, eliminación, pero no retorna ningún valor, pero puede recibir parámetros.

la sintaxis es la siguiente:

```
CREATE PROCEDURE procedure4
AS
BEGIN
    DECLARE @VALOR1 INT;
    SET @VALOR1= @var1;
END;

CREATE PROCEDURE procedure2 (@var1 INT)
AS
BEGIN
    DECLARE @VALOR1 INT;
    SET @VALOR1= @var1;
END;
```

### Alter

Esta sentencia nos permite hacer modificaciones a las estructuras de tablas, procedimientos y funciones.



Alter Table:

Permite modificar la estructura de la tabla, permite agregar o eliminar columnas.

Ejemplo

```
ALTER TABLE tabla2 ADD COLUMN tasa decimal;
```

esta instrucción nos permite crear una nueva columna llamada tasa con un tipo de dato "decimal" en la tabla "tabla2"

```
ALTER TABLE tabla2 DROP COLUMN tasa;
```

Esta instrucción nos permite borrar la columna tasa dentro de la tabla "tabla2"

Alter Function

Esta instrucción nos permite modificar una función, modifica el contenido de la función, cambios en parámetros y resultados.

```
ALTER FUNCTION funcion1  
RETURN INT  
AS  
BEGIN  
    DECLARE @VALOR1 DECIMAL;  
    SET @VALOR1 = 15.0;  
    RETURN @VALOR1;  
END;
```

Alter Procedure

Permite modificar la estructura de un procedimiento, modifica el contenido del procedimiento.

```
ALTER PROCEDURE procedure4  
AS
```



```
BEGIN
    DECLARE @VALOR1 INT;
    SET @VALOR1= @var1;
END;
```

## Drop

Nos permite eliminar una base de datos, una tabla, una función y un procedimiento.

### Drop Data Base

```
DROP DATA BASE Ejemplo;
```

### Drop Table

```
DROP DATA TABLE tabla1;
```

### Drop Function

```
DROP FUNCTION funcion1;
```

### Drop Procedure

```
DROP PROCEDURE procedure1;
```

## Truncate

Esta instrucción permite borrar todos los registros almacenados sin borrar el esqueleto de la misma.

### Truncate Table

Permite eliminar todos los registros dentro de la tabla seleccionada sin borrar los parámetros ni tipos de parámetros que contiene, borra únicamente los registros.

```
TRUNCATE TABLE tabla1;
```

## Funciones

Son objetos XSQL que retornar un valor escalar, hay funciones definidas por el usuario y funciones propias del sistema de Base de Datos.



### Funciones definidas por el usuario

Las funciones definidas por el usuario en XSQL serán únicamente de los tipos Numéricos exactos, no permite otro tipo de dato para las funciones

```
CREATE FUNCTION suma(@val1 AS int,@val2 AS int)
RETURNS int
AS
BEGIN
Declare @valor int
Set @valor = @val1 + @val2
RETURN @valor
END
```

### Funciones del sistema

El sistema cuenta con funciones propias como la siguientes:

Nombre	Descripción	Parámetros	Valor retorno
CONCATENA	Concatena dos valores	(NCHAR, NCHAR) (NVARCHAR, NVARCHAR)	NVARCHAR
SUBSTRAER	Extrae una porción de una cadena de caracteres	(texto, inicio, longitud) Nvarchar, int, int ó Nchar, int, int	nvarchar
CONTAR	Devuelve el número de filas de una tabla que cumplen con cierto criterios		Int
SUMA	Devuelve la suma de una columna numérica	(nombre de la columna o número de la columna)	Decimal
CAS	Convierte una columna o variable a un valor definido	(Variable as Valor)	Según el valor seleccionado en el CAS

### Ejemplos:

#### CONCATENAR

```
SELECT CONCATENAR('HOLA','MUNDO')
```

Resultado

```
HOLAMUNDO
```

#### SUBSTRAER



**SELECT SUBSTR('HOLAMUNDO',1,4)**

Resultado

**HOLA**

HOY

**SELECT HOY()**

Resultado

**13-7-2023 23:15**

CONTAR

**SELECT CONTAR(\*) FROM TABLA WHERE CAMPO1 = 1**

Resultado

**100**

SUMA

**SELECT SUMA(IVA) FROM TABLA WHERE CAMPO1 = 1**

Resultado

**25**

**CAST EJEMPLO:**

SQL	Valor Salida
Select CAST(@BIT AS INT)	1 o 0 (Entero)
Select CAST(@NCHAR AS INT) o Select CAST(@NVARCHAR AS INT)	Valor ASCII del texto si es un valor, y si es una cadena es la sumatoria de los ASCII de la cadena
Select CAST(@INT AS VARCHAR)	Si el valor esta entre 1 y 255 imprimir el valor ASCII. Si es mayor a 255 devolver NULL
Select CAST(@BIT A VARCHAR)	Devuelve 0 y 1 según el valor

También podemos llamar a las funciones que creamos de la siguiente manera

```
USE Prueba_archivo;  
DECLARE @VAL INT;  
DECLARE @VAL2 int;  
DECLARE @VAL3 int;  
SET @VAL = funcion1();  
set @VAL2 = funcion2(1,2);
```



```
set @VAL3 = funcion2(@VALOR1=1,@VALOR2=2);
```

## Procedimientos

El objeto XSQL procedimientos almacenados es un conjunto de instrucciones que se guardará en una base de datos para ser utilizada posteriormente.

```
USE Prueba_archivo;  
  
exec procedure1;  
  
exec procedure2 5;  
exec procedure2 @var1 = 6;
```

## If

```
DECLARE @VALOR AS INT;  
set @VALOR = 1;  
  
IF @VALOR >= 1 THEN  
SET @VALOR = 10;  
ELSE  
SET @VALOR = 5;  
END IF;  
  
@VALOR +IF(@VALOR >5,100,200)
```

## Case

### Insert

Esta instrucción nos permite insertar nuevos registros en una tabla. Permite agregar datos en una o más columnas de una fila.

```
INSERT INTO table (column1, column2, ...)  
VALUES (value1, value2, ...);
```

### Ejemplo



```
INSERT INTO Empleados (Nombre, Salario, Departamento)
VALUES ('Carlos', 60000, 'Ventas');
```

## Select

Esta instrucción nos permite recuperar datos de una o más tablas en una base de datos. Se utiliza para realizar consultas y extraer información específica.

```
SELECT column1, column2, ...
FROM table
WHERE condition;
```

## Ejemplo

```
SELECT Nombre, Salario
FROM Empleados
WHERE Departamento = 'Ventas' AND Salario > 5000;
```

## Ejemplo de salida:

Nombre	Salario
Carlos	5500

## Update

Se utiliza para modificar los datos existentes en una tabla. Permite cambiar los valores de una o más columnas en una o varias filas de la tabla basandose en una condicion especificada.

```
UPDATE table
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

## Ejemplo

```
UPDATE Empleados
SET Salario = Salario * 1.1
```





```
WHERE Departamento = 'Ventas';
```

Nombre	Salario
Carlos	6050

## Delete

Esta instrucción nos permite eliminar registros de una tabla que cumplen con una condición en específico.

```
DELETE FROM table  
WHERE condition;
```

Ejemplo:

```
DELETE FROM Empleados  
WHERE Nombre = 'Carlos';
```

## Editor de texto

### Crear nuevo Query

Nos permite ejecutar la instrucción que tenemos en el editor de texto, esta revisa primero si tiene una sintaxis correcta de acuerdo a el lenguaje XSQL, y realiza distintas validaciones semánticas.

Tenemos que tomar en cuenta que no permite la creación de dos bases de datos con el mismo nombre, así como dentro de una base de datos la creación de dos tablas con mismo nombre, o procedimientos y funciones.

## Herramientas de SQL

### Exportar datos

Permite el export de la base de datos seleccionada.

### Importar datos

Permite cargar datos de un archivo xml de una base de datos



## Salida de datos

El IDE nos permite ver la salida de información sobre los distintos procedimientos que se apliquen, así como cuando ejecutamos una instrucción SELECT, nos permite visualizar los datos que solicitamos.

### Ejemplo

The screenshot shows a database IDE interface. At the top, there is a tab labeled 'Query1'. Below it, the SQL query is entered: `Use Alimentos;` followed by `select * from products where product_no >1;`. Below the query editor, there is a section titled 'Salida de Datos'. It indicates that the database being used is 'Alimentos' and the table is 'products'. Below this, a table of results is displayed with three columns: 'product\_no', 'name', and 'id\_producto'. The results show 9 rows of data, with 'product\_no' ranging from 2 to 10, 'name' as 'Manza' followed by a number, and 'id\_producto' as 1.

product_no	name	id_producto
2	Manza8	1
3	Manza9	1
4	Manza10	1
5	Manza11	1
6	Manza12	1
7	Manza13	1
8	Manza14	1
9	Manza15	1
10	Manza16	1

## Cierre de la aplicación

Puede pulsar la x que está en una esquina o en ->Archivo -> Salir.

## Agradecimiento

Gracias por utilizar XSQL.