

HITO 2, EVALUACIÓN PROCESUAL BASE DE DATOS 2

UNIVERSIDAD PRIVADA FRANZ TAMAYO
DEFENSA HITO 2 – EVALUACION PROCESUAL

Estudiante: Univ. Jhomar Huaycho Quispe

Asignatura: BASE DE DATOS II

Carrera: INGENIERÍA DE SISTEMAS

Paralelo: BDA (1)

Docente: Lic. William Barra Paredes

fecha: 11/09/2022

GITHUB: [https://github.com/JHOMARHUAYCHO/BASE-DE-DATOS-2/tree/main/HITO%202/\(PRACTICA\)%20EVALUACION%20PROCESUAL%20HITO2](https://github.com/JHOMARHUAYCHO/BASE-DE-DATOS-2/tree/main/HITO%202/(PRACTICA)%20EVALUACION%20PROCESUAL%20HITO2)

VIDEO EXPOSICION Y DOCUMENTOS DRIVE:
<https://drive.google.com/drive/folders/15TOKcHETUa7scmjXgcZeGdMvHSpmcbuF?usp=sharing>

PARTE TEORICA

1. ¿A qué se refiere cuando se habla de bases de datos relacionales?

Es aquella que relaciona todos los datos almacenados a través de tablas dentro de una base de datos, y usa el lenguaje SQL para manipular esos datos.

2. ¿A qué se refiere cuando se habla de bases de datos no relacionales?

También llamadas bases de datos nosql, estas no usan el lenguaje SQL, no guardan los datos en tablas sino a través de documentos y son útiles para gestionar información no estructurada.

3. ¿Qué es Mysql y Mariadb?. Explique si existen diferencias o son iguales, etc.

Mysql es un gestor de base de datos relacional, bajo licencia de Oracle.

Mariadb: es un gestor de base de datos, tiene la característica de que es un software libre. Son parecidos ya que mariadb fue creado como reemplazo de mysql, la diferencia más notable es que mariadb es software libre y mysql no lo es.

4. ¿Qué son las funciones de agregación?

Son funciones que ya vienen creadas y solo necesitan ser llamadas, estas funciones nos permiten realizar operaciones en un grupo de datos, ya sea sumar, elegir el número mayor, sacar el promedio, etc.

5. ¿Qué llegaría a ser XAMPP?

Xampp es un paquete (como un kit de herramientas) para nuevos desarrolladores que contiene, un gestor de base de datos, el servidor apache, lenguajes de programación como php.

6. ¿Cuál es la diferencia entre las funciones de agregación y funciones creados por el DBA? Es decir funciones creadas por el usuario.

Las funciones de agregación son aquellas que ya vienen con en el gestor de base de datos solo necesitan ser llamadas.

Las creadas por el usuario son aquellas, como se indica necesitan ser creadas, estas pueden usar las funciones de agregación; y su dificultad con las cual son creadas depende de la habilidad del usuario.

7. ¿Para qué sirve el comando USE?

Indica al gestor, cual base de datos seleccionar para su uso.

8. Que es DML y DDL?

Las sentencias DDL son las sentencias utilizadas para la creación de base de datos, tablas, triggers, etc.

Las sentencias DML se usan para manipular los datos, ejemplo: insertar borrar, modificar datos.

9. ¿Qué cosas características debe de tener una función? Explique sobre el nombre, el return, parametros, etc.

Una función debe tener las siguientes partes

CREATE FUNCTION OR REPLACE NombreDeLaFuncion (paramteros a recibir)

RETURNS INTEGER <---Dato_A_devolver

BEGIN

DECLARE NUMERO INTERGER DEFAULT 0 < -----declaración de datos para usar, si son necesarios

SET NUMERO=2; <----- Procedimiento de la funcion

RETURN NUMERO; <----dato_a_enviar

END;

SELECT NombreDeLaFuncion (parámetros a enviar);

10. ¿Cómo crear, modificar y cómo eliminar una función?

Para crear una función el código es **CREATE FUNCTION**

Para modificar una función es **REPLACE**

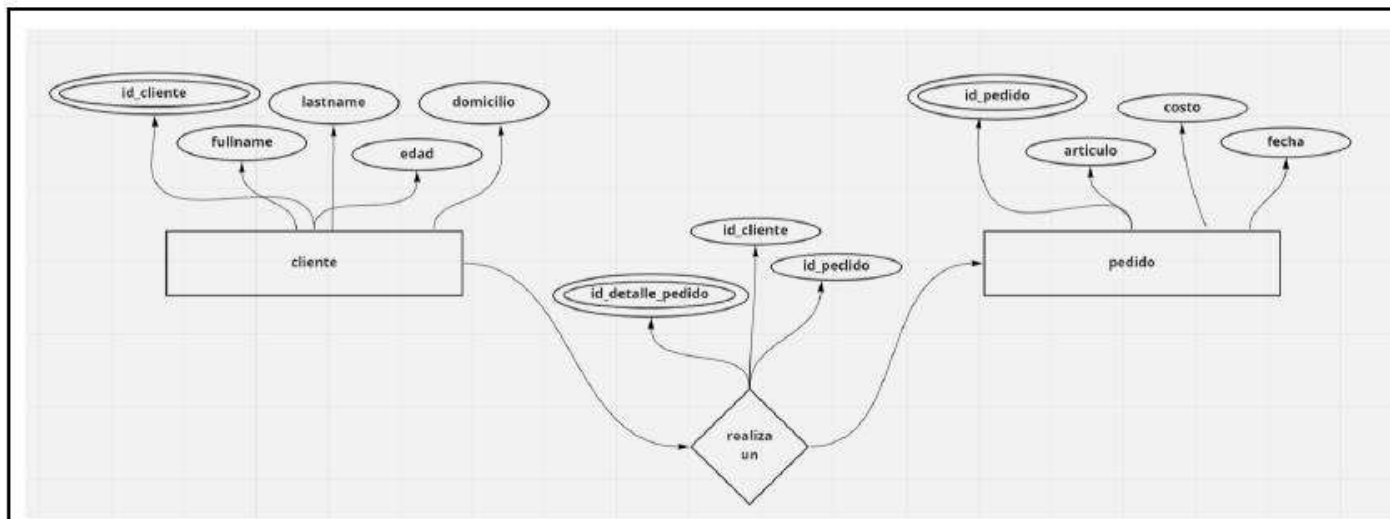
Para eliminar la función es **DROP FUNCTION (nombre de la funcion);**

PARTE PRACTICA

PREGUNTA 11

11. Crear las tablas y 2 registros para cada tabla para el siguiente modelo ER.

○ Diseño.



○ Se sugiere crear una base de datos de nombre POLLOS_COPA y en ella crear las tablas:

- cliente
- detalle pedido
- pedido
- adjuntar código SQL

```

• CREATE DATABASE POLLOS_COPA;
  USE POLLOS_COPA;

  CREATE TABLE CLIENTE
  (
    ID_CLIENTE INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL ,
    FULLNAME VARCHAR(30),
    LASTNAME VARCHAR(30),
    EDAD INTEGER,
    DOMICILIO VARCHAR(40)
  );

  CREATE TABLE PEDIDO
  (
    ID_PEDIDO INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ARTICULO VARCHAR(30),
    COSTO FLOAT,
    FECHA DATE
  );

  CREATE TABLE DETALLE_PEDIDO
  (
    ID_DETALLE_PEDIDO INTEGER AUTO_INCREMENT NOT NULL PRIMARY
    KEY,
    ID_PEDIDO INTEGER NOT NULL,
    ID_CLIENTE INTEGER NOT NULL,
    FOREIGN KEY (ID_PEDIDO) REFERENCES PEDIDO(ID_PEDIDO),
    FOREIGN KEY (ID_CLIENTE) REFERENCES CLIENTE(ID_CLIENTE)
  );

```

```

INSERT INTO CLIENTE(FULLNAME, LASTNAME, EDAD, DOMICILIO)
VALUES ('PEDRO', 'SALAZAR GOMEZ', 19, 'RIO SECO'),
       ('MARIA', 'DELGADO MENDOZA', 20, 'BALLIVIAN');

```

```

INSERT INTO PEDIDO(ARTICULO, COSTO, FECHA)
VALUES ('LECHE', 21.2, '2022-08-02'),
       ('CARNE', 7.4, '2022-09-12');

```

```

INSERT INTO DETALLE_PEDIDO(ID_PEDIDO, ID_CLIENTE)
VALUES (1, 2),
       (2, 1);

```

Capturas de pantalla



ID_CLIENTE	FULLNAME	LASTNAME	EDAD	DOMICILIO
1	PEDRO	SALAZAR GOMEZ	19	RIO SECO
2	MARIA	DELGADO MENDOZA	20	BALLIVIAN

ID_PEDIDO	ARTICULO	COSTO	FECHA
1	1 LECHE	21.2	2022-08-02
2	2 CARNE	7.4	2022-09-12

ID_DETALLE_PEDIDO	ID_PEDIDO	ID_CLIENTE
1	1	2
2	2	1

PREGUNTA 12

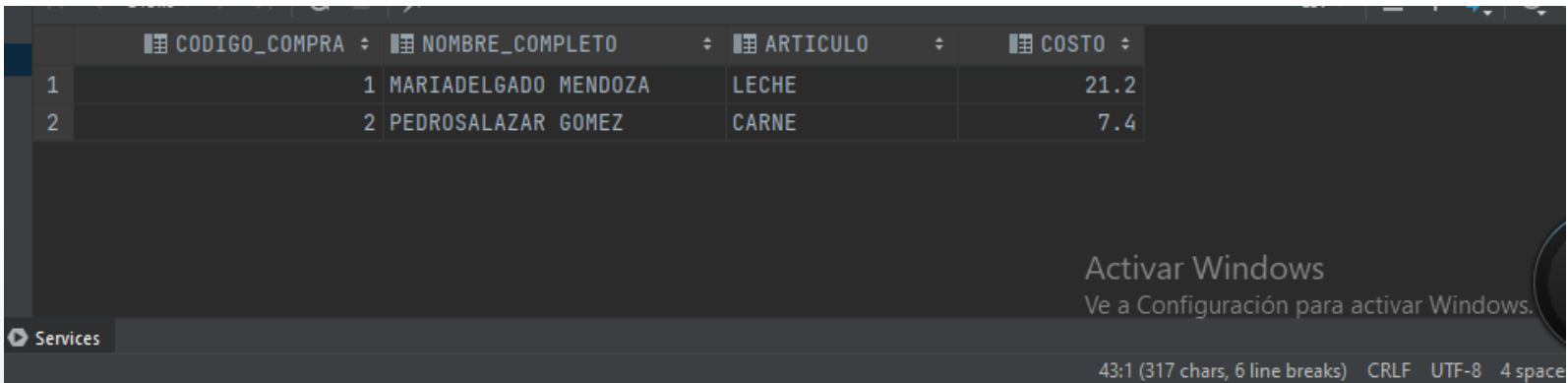
Crear una consulta SQL en base al ejercicio anterior.

- o Debe de utilizar las 3 tablas creadas anteriormente.
- o Para relacionar las tablas utilizar JOINS.
- o Adjuntar el código SQL generado.

RESPUESTA

```
SELECT DETALLE_PEDIDO.ID_DETALLE_PEDIDO AS CODIGO_COMPRA,  
       CONCAT(CLIENTE.FULLNAME, CLIENTE.LASTNAME) AS NOMBRE_COMPLETO,  
       PEDIDO.ARTICULO, PEDIDO.COSTO  
  
FROM DETALLE_PEDIDO  
INNER JOIN CLIENTE ON DETALLE_PEDIDO.ID_CLIENTE = CLIENTE.ID_CLIENTE  
INNER JOIN PEDIDO ON DETALLE_PEDIDO.ID_PEDIDO = PEDIDO.ID_PEDIDO;
```

CAPTURA DE PANTALLAS



	CODIGO_COMPRA	NOMBRE_COMPLETO	ARTICULO	COSTO
1	1	MARIADELGADO MENDOZA	LECHE	21.2
2	2	PEDROSALAZAR GOMEZ	CARNE	7.4

Activar Windows
Ve a Configuración para activar Windows.

Services

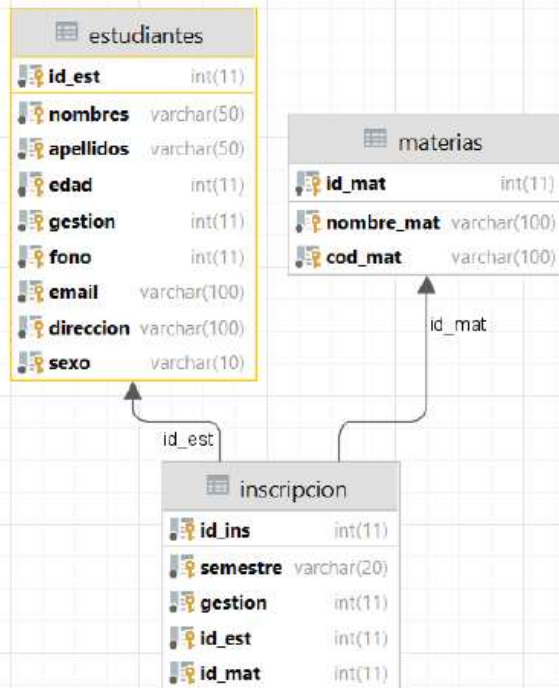
43:1 (317 chars, 6 line breaks) CRLF UTF-8 4 space

PREGUNTA 13

Crear un función que compare dos códigos de materia.

o Recrear la siguiente base de datos:

```
CREATE DATABASE tareaHito2;  
USE tareaHito2;
```



```
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email, direccion, sexo)
VALUES ('Miguel', 'Gonzales Veliz', 20, 2832115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino'),
      ('Sandra', 'Mavir Uria', 25, 2832116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino'),
      ('Joel', 'Aduviri Mondar', 30, 2832117, 'joel@gmail.com', 'Av. 6 de Agosto', 'masculino');
```

```
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Introduccion a la Arquitectura', 'ARQ-101'),
      ('Urbanismo y Diseno', 'ARQ-102'),
      ('Dibujo y Pintura Arquitectonica', 'ARQ-103'),
      ('Matematica discreta', 'ARQ-104'),
      ('Fisica Basica', 'ARQ-105');
```

```
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES (1, 1, '1er Semestre', 2018),
      (1, 2, '2do Semestre', 2018),
      (2, 4, '1er Semestre', 2019),
      (2, 3, '2do Semestre', 2019),
      (3, 3, '2do Semestre', 2020),
      (3, 1, '3er Semestre', 2020),
      (4, 4, '4to Semestre', 2021),
      (5, 5, '5to Semestre', 2021);
```

o Resolver lo siguiente:

- Mostrar los nombres y apellidos de los estudiantes inscritos en la materia ARQ-105, adicionalmente mostrar el nombre de la materia.
- Deberá de crear una función que reciba dos parámetros y esta función deberá ser utilizada en la cláusula WHERE.

CREANDO DE LA FUNCION:

```
CREATE OR REPLACE FUNCTION COMPARA_MATERIAS(CODI_MAT
VARCHAR(100),NOMBRE_MAT VARCHAR(100))
RETURNS BOOLEAN
BEGIN
    DECLARE RESPUESTA BOOLEAN;

    IF CODI_MAT=NOMBRE_MAT
    THEN
        SET RESPUESTA=1;
    END IF;
    RETURN RESPUESTA;
END;
```

LLAMADO DE LA FUNCION:

```
SELECT ESTUDIANTES.id_est, ESTUDIANTES.NOMBRES,ESTUDIANTES.APELLIDOS,
MATERIAS.NOMBRE_MAT,MATERIAS.COD_MAT
FROM inscripcion
INNER JOIN estudiantes ON inscripcion.id_est = estudiantes.id_est
INNER JOIN materias ON inscripcion.id_mat = materias.id_mat
WHERE COMPARA_MATERIAS(materias.cod_mat,'ARQ-105');
```



PREGUNTA 14

14. Crear una función que permita obtener el promedio de las edades del género

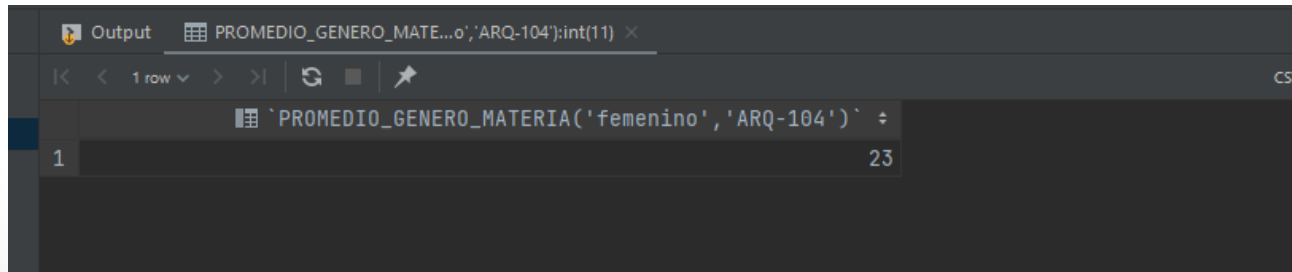
Masculino o femenino de los estudiantes inscritos en la asignatura ARQ-104.

- La función recibe como parámetro el género y el código de materia.

```
CREATE OR REPLACE FUNCTION PROMEDIO_GENERO_MATERIA( GENERO
VARCHAR(11),MATERIA_COD VARCHAR(100))
RETURNS INTEGER
BEGIN
    DECLARE PROMEDIO INTEGER DEFAULT 0;

    SELECT AVG(estudiantes.edad) INTO PROMEDIO
    FROM inscripcion
    INNER JOIN estudiantes ON inscripcion.id_est = estudiantes.id_est
    INNER JOIN materias ON inscripcion.id_mat = materias.id_mat
    WHERE estudiantes.sexo= GENERO AND materias.cod_mat=MATERIA_COD;
    RETURN PROMEDIO;
END;

SELECT PROMEDIO_GENERO_MATERIA('femenino','ARQ-104');
```



PROMEDIO_GENERO_MATERIA('femenino','ARQ-104')	
1	23

PREGUNTA 15

Crear una función que permita concatenar 3 cadenas.

o La función recibe 3 parámetros.

o Si las cadenas fuesen:

- Pepito
- Pep
- 50

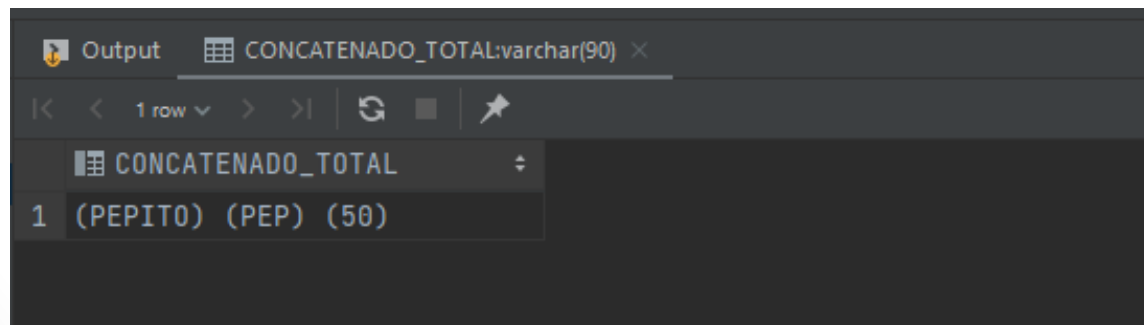
o La salida debería ser: (Pepito), (Pep), (50)

o La función creada utilizarlo en una consulta SQL.

```
CREATE OR REPLACE FUNCTION CONCATENAR_PALABRAS (A1 VARCHAR(30), A2
VARCHAR(30), A3 VARCHAR(30))
RETURNS VARCHAR(90)
BEGIN
    DECLARE CADENA_TOTAL VARCHAR(90) DEFAULT '';
    SET CADENA_TOTAL= CONCAT('(', A1, ')', '(', A2, ')', '(', A3, ')');

    RETURN CADENA_TOTAL ;
END;

SELECT CONCATENAR_PALABRAS('PEPITO', 'PEP', '50') AS CONCATENADO_TOTAL;
```



The screenshot shows a database output window with the title "Output" and a tab labeled "CONCATENADO_TOTAL:varchar(90) X". Below the title bar, there are navigation icons and a dropdown menu showing "1 row". The main area displays a table with one column named "CONCATENADO_TOTAL" and one row containing the value "(PEPITO) (PEP) (50)".

CONCATENADO_TOTAL
(PEPITO) (PEP) (50)

PREGUNTA 16

Crear una función de acuerdo a lo siguiente:

- ❖ Mostrar el nombre, apellidos, edad y el semestre de todos los estudiantes que estén inscritos.
- ❖ Siempre y cuando la suma de las edades del sexo femenino (tambien puede ser masculino) sea par y mayores a cierta edad.
- ❖ Debe de crear una función que sume las edades (recibir como parámetro el sexo, y la edad).
 - Ejemplo: sexo='Masculino' y edad=22
 - Note que la función recibe 2 parámetros.
- ❖ La función creada anteriormente debe utilizarse en la consulta SQL. (Cláusula WHERE).

```

CREATE OR REPLACE FUNCTION EST_SEGUN_EDAD_GENERO(SEXO VARCHAR(11),EDAD
INTEGER)
RETURNS BOOLEAN
BEGIN
    DECLARE SUMA INTEGER DEFAULT 0;
    DECLARE SIoNO BOOLEAN;

    SELECT SUM(ESTUDIANTES.EDAD) INTO SUMA
    FROM estudiantes
    WHERE estudiantes.SEXO=SEXO;

    IF SUMA%2=0 AND SUMA>EDAD
    THEN
        SET SIoNO=1;
    END IF;
    RETURN SIoNO;
END;

```

```

SELECT ESTUDIANTES.NOMBRES,
ESTUDIANTES.APELLIDOS,ESTUDIANTES.EDAD,INSCRIPCION.SEMESTRE
FROM inscripcion
INNER JOIN estudiantes ON inscripcion.id_est = estudiantes.id_est
WHERE EST_SEGUN_EDAD_GENERO('MASCULINO',22);

```

	NOMBRES	APELLIDOS	EDAD	SEMESTRE
1	Miguel	Gonzales Veliz	20	1er Semestre
2	Miguel	Gonzales Veliz	20	2do Semestre
3	Sandra	Mavir Uria	25	1er Semestre
4	Sandra	Mavir Uria	25	2do Semestre
5	Joel	Adubiri Mondar	30	2do Semestre
6	Joel	Adubiri Mondar	30	3er Semestre
7	Andrea	Arias Ballesteros	21	4to Semestre
8	Santos	Montes Valenzuela	24	5to Semestre

Activar Windows
Ve a Configuración para

Services

```
SELECT SUM(EDAD)
FROM estudiantes
GROUP BY(ESTUDIANTES.SEXO);
```

PREGUNTA 17

CREATE OR REPLACE FUNCTION PREGUNTA_17(NOMBRE VARCHAR(100), APELLIDO VARCHAR(100))

Output SUM(EDAD):int(11) x

2 rows

SUM(EDAD)

1	46
2	74

PREGUNTA 17

- Crear una función sobre la tabla estudiantes que compara un nombre y apellidos. (si existe este nombre y apellido mostrar todos los datos del estudiante).
 - La función devuelve un boolean.
 - La función debe recibir 4 parámetros, nombres y apellidos.

Similar al siguiente ejemplo.

```
create function busca_nombres_apellidos(  
    est.nombres,  
    'William',  
    est.apellidos,  
    'Barra Paredes'  
) RETURNS ....
```

- Ejemplo:
- La función debería ser usada en la cláusula WHERE.
- El objetivo es buscar a estudiantes a través de sus nombres y apellidos.

SOLUCION

```
CREATE OR REPLACE FUNCTION BUSCA_NOMBRES_APELLIDOS(nombre
varchar(100),apellidos varchar(100),nombre_c varchar (100),apellido_c
varchar(100))

    RETURNS BOOLEAN
    BEGIN
        DECLARE DECISION BOOLEAN;
        if nombre=nombre_c and apellidos=apellido_c
        then
            set DECISION=1;
        end if;
        RETURN DECISION;

    END;
```

```
select *
from estudiantes
where BUSCA_NOMBRES_APELLIDOS(estudiantes.nombres,
    estudiantes.apellidos,'miguel','gonzales veliz');
```

Output tareahito2.estudiantes x

id_est	nombres	apellidos	edad	gestion	fono	email	direccion	se
1	1 Miguel	Gonzales Veliz	20	<null>	2832115	miguel@gmail.com	Av. 6 de Agosto	mas

Structure Notifications

0 rows

id_est	nombres	apellidos	edad	gestion	fono	email	direccion	se
--------	---------	-----------	------	---------	------	-------	-----------	----

Structure Notifications

Activar Windows
Ve a Configuración para activar Windows.