

TALLER No. 7

TALLER CASO DE ESTUDIANTE RESTAURANTE EL BUEN COMER

OBJETIVO:

- Crear Modelo relacional
- Crear Modelo físico
- Crear Objetos en la base de datos.

EVIDENCIA(S) A ENTREGAR:
EV1 Desarrollar caso de estudio de acuerdo con:

Se debe entregar un documento llamado “CasoEstudioRestaurante.sql”, que contenga:

- Creación modelo físico
- Creación de objetos de la base de datos
- Informe de desarrollo del caso de estudio.

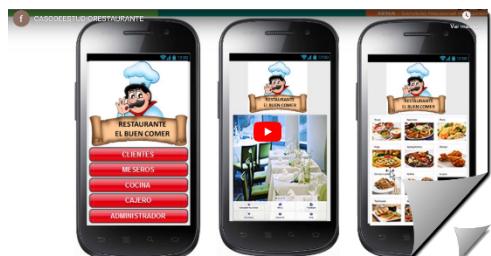
CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	JOSE FERNANDO GALINDO S.	Instructor	Teleinformática	28/05/2020

CONTROL DE CAMBIOS (diligenciar únicamente si realizan ajustes al taller)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

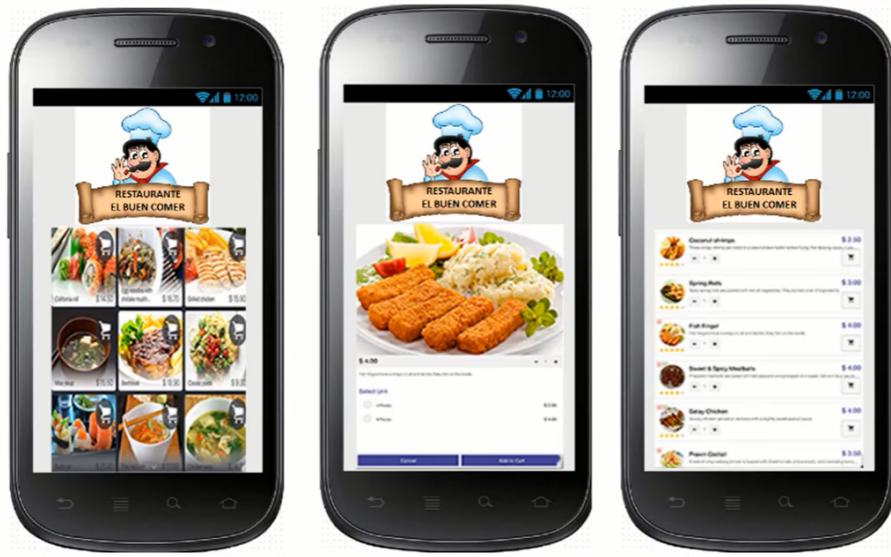
PRESENTACIÓN



Como parte final de la actividad de aprendizaje, se presenta un caso de estudio donde se plantea una problemática con el restaurante “[El buen comer](#)”, el video orienta al aprendiz a identificar los requisitos funcionales y los objetivos a cumplir.

El departamento de análisis y diseño entrega los diagramas de robustez del funcionamiento que debe tener el aplicativo a desarrollar. Su labor es desarrollar el modelo relacional, el modelo físico en una base de datos MYSQL donde se debe construir tablas, relaciones, procedimientos CRUD, TRIGGERS y vistas que den respuesta a los diseñados y a los requisitos funcionales identificados.

CASO DE ESTUDIO RESTAURANTE EL BUEN COMER



QUE PERMITA PRESTAR SERVICIOS COMO: ATENCION A CLIENTES, MESEROS, COCINA Y FACTURACION.

CASO DE ESTUDIO RESTAURANTE EL BUEN COMER



EL MESERO ESCOGE LA MESA PARA ATENDER AL CLIENTE.

CASO DE ESTUDIO RESTAURANTE EL BUEN COMER



EL CLIENTE OBSERVA POR CATEGORIA EL PLATO QUE DESEA DESDE SU TABLETA .

CASO DE ESTUDIO RESTAURANTE EL BUEN COMER



EL CLIENTE OBSERVA LOS PLATOS POR LA CATEGORIA ESCOGIDA DESDE SU TABLETA CON LOS PRECIOS.

CASO DE ESTUDIO RESTAURANTE EL BUEN COMER



LOS INTEGRANTES DE LA COCINA, VERAN POR MEDIO DE UN MONITOR DE 40 PULGADAS, LOS PEDIDOS SOLICITADOS POR MESA Y CUANDO LOS PRODUCTOS ESTEN, CAMBIARAN EL ESTADO A LISTO PARA QUE EL MESERO LOS ENTREGUE A LOS CLIENTES.

CASO DE ESTUDIO RESTAURANTE EL BUEN COMER

EL CLIENTE SOLICITA AL MESERO LA CUENTA Y ESTE PEDIRA LA FACTURA AL CAJERO PARA SU ELABORACION Y UNA VEZ REALIZADA, EL MESERO IRA POR ELLA PARA ENTREGARLA AL CLIENTE PARA SU RESPECTIVO PAGO.

CASO DE ESTUDIO RESTAURANTE EL BUEN COMER

EL MESERO PREGUNTA AL CLIENTE QUE MEDIO DE PAGO VA A UTILIZAR PARA CANCELAR.

MAPA DE PROCESOS

CREANDO UN PROCEDIMIENTO CRUD

Los procedimientos CRUD son utilizados para evitar que los programadores utilicen instrucciones DML (insert, Update, delete), quemados dentro de la programación. Aparte de esto, se pueden configurar para que determinen los derechos que se tienen para así evitar inserciones, actualizaciones o borradas de registros que no estén autorizadas para los usuarios.

En el presente taller, se enseña a crear y usar procedimientos CRUD y utilizar rutina de permisos para las tablas.

```
DELIMITER $  
CREATE OR REPLACE PROCEDURE PREPS(IN POPER VARCHAR[1],IN PID INTEGER,IN PNOM VARCHAR[30],IN ESTADO INTEGER)  
SP: BEGIN  
-- DECLARAR HAY ENTERO;  
IF(POPER IN('Q','U','D')) THEN  
    SELECT COUNT(*) INTO @B FROM EPS WHERE IDEPS=PID;  
    IF(@B=0) THEN  
        SELECT 'ID NO EXISTE' AS MSG;  
        LEAVE SP;  
    END IF;  
    END IF;  
    END IF;  
  
    IF(POPER='A') THEN  
        SELECT * FROM EPS;  
    END IF;  
    IF(POPER='Q') THEN  
        SELECT * FROM EPS WHERE IDEPS=PID;  
    END IF;  
  
    IF(POPER='I') THEN  
        INSERT INTO EPS(NOMBRE,ESTADOEPS) VALUES(PNOM,ESTADO);  
    END IF;  
    IF(POPER='U' ) THEN  
        IF(PNOM IS NULL AND ESTADO IS NOT NULL) THEN  
            UPDATE EPS SET ESTADOEPS=ESTADO WHERE IDEPS=PID;  
        END IF;  
        IF(PNOM IS NOT NULL AND ESTADO IS NULL) THEN  
            UPDATE EPS SET NOMBRE=PNOM WHERE IDEPS=PID;  
        END IF;  
        IF(PNOM IS NOT NULL AND ESTADO IS NOT NULL) THEN  
            UPDATE EPS SET NOMBRE=PNOM, ESTADOEPS=ESTADO WHERE IDEPS=PID;  
        END IF;  
    END IF;  
    IF(POPER='D') THEN  
        DELETE FROM EPS WHERE IDEPS=PID;  
    END IF;  
    SELECT 'OK' AS MSG;  
END;  
$  
DELIMITER ;
```



En el código que se presenta se muestra la manera de cómo realizar un procedimiento CRUD; este contiene cuatro parámetros: la operación, el id de la EP, el nombre de la EPS y el estado de la EPS.

La primera parte vemos que, para las operaciones de actualización, borrado y consulta, se valida que el id de la EPS exista; si no existe se debe dar un mensaje de id no existente y salir del procedimiento.

Procedemos a crear la programación necesaria para insertar, borrar y actualizar.

Para la actualización tenemos tres opciones: actualizar el nombre de la EPS, actualizar el estado de la EPS o ambas.

La primera opción es “A” que me lista todas la EPS disponibles en la tabla.

```
MariaDB [salud]> CALL PREPS('A',NULL,NULL,NULL);
+-----+-----+-----+
| ideps | nombre          | estadoeps |
+-----+-----+-----+
|     1 | SALUDCOOP        |         4   |
|     2 | CAFESALUD        |         1   |
|     3 | NUEVA EPS         |         1   |
|     4 | ISS               |         5   |
|     5 | CAPRECOM         |         4   |
|     6 | COOMEVA          |         1   |
|     7 | COMPENSAR         |         1   |
|     8 | COLMEDICA         |         1   |
|     9 | HUMANA VIVIR      |         2   |
|    10 | FAMISANAR        |         1   |
|    11 | CRUZ BLANCA       |         3   |
|    12 | SOLSALUD          |         1   |
|    13 | SALUD TOTAL        |         5   |
|    14 | SANITAS           |         3   |
|    18 | SU EPZZZZZ        |         1   |
+-----+-----+-----+
15 rows in set (0.00 sec)

+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.10 sec)

Query OK, 0 rows affected (0.12 sec)
```



Consulta la EPS mediante un Id con la opción “Q”.

```
MariaDB [salud]> CALL PREPS('Q',18,NULL,NULL);
+-----+-----+-----+
| ideps | nombre      | estadoeps |
+-----+-----+-----+
|    18 | SU EPZZZZZ |          1 |
+-----+-----+-----+
1 row in set (0.00 sec)

+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.04 sec)
```



Para insertar una nueva EPS utilizamos la opción “I”.

```
MariaDB [salud]> CALL PREPS('I',NULL,'MIEPS',1);
+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)

MariaDB [salud]> CALL PREPS('Q',19,NULL,NULL);
+-----+-----+-----+
| ideps | nombre      | estadoeps |
+-----+-----+-----+
|    19 | MIEPS      |          1 |
+-----+-----+-----+
1 row in set (0.00 sec)

+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.04 sec)
```



Para actualizar con la opción “U” y solamente el nombre de ésta.

```
MariaDB [salud]> CALL PREPS('U',19,'SUEPS',NULL);
+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)

MariaDB [salud]> CALL PREPS('Q',19,NULL,NULL);
+-----+-----+-----+
| ideps | nombre | estadoeps |
+-----+-----+-----+
|    19 | SUEPS |          1 |
+-----+-----+-----+
1 row in set (0.00 sec)

+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.04 sec)
```



Para actualizar con la opción “U” y solamente el estado de la EPS.

```
MariaDB [salud]> CALL PREPS('U',19,NULL,2);
+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)

MariaDB [salud]> CALL PREPS('Q',19,NULL,NULL);
+-----+-----+-----+
| ideps | nombre | estadoeps |
+-----+-----+-----+
|    19 | SUEPS |          2 |
+-----+-----+-----+
1 row in set (0.00 sec)

+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.04 sec)
```



Para actualizar con la opción “U”, el nombre y el estado de la EPS.

```
MariaDB [salud]> CALL PREPS('U',19,'LAEPS',1);
+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)

MariaDB [salud]> CALL PREPS('Q',19,NULL,NULL);
+-----+-----+-----+
| ideps | nombre | estadoeps |
+-----+-----+-----+
| 19   | LAEPS  |      1 |
+-----+-----+-----+
1 row in set (0.00 sec)

+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)
```



Verificamos al actualizar si el código existe.

```
Query OK, 0 rows affected (0.03 sec)

MariaDB [salud]> CALL PREPS('U',29,NULL,2);
+-----+
| MSG |
+-----+
| ID NO EXISTE |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```



Para borrar una EPS con la opción D".

```
MariaDB [salud]> CALL PREPS('D',19,NULL,NULL);
+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)

MariaDB [salud]> CALL PREPS('Q',19,NULL,NULL);
+-----+
| MSG |
+-----+
| ID NO EXISTE |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```



Borran una EPS con un id desconocido.

```
MariaDB [salud]> CALL PREPS('D',19,NULL,NULL);
+-----+
| MSG |
+-----+
| ID NO EXISTE |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```



SEGURIDAD EN LOS PROCEDIMIENTOS CRUD.

Para implementar debemos crear la tabla “PERMISOXTABLAXUSUARIO”

```
MariaDB [salud]> CREATE TABLE PERMISOXTABLAXUSUARIO(
-> USUARIO VARCHAR(30),
-> TABLA VARCHAR(30),
-> INSERTAR INTEGER(1) DEFAULT 0,
-> LISTAR INTEGER(1) DEFAULT 0,
-> ACTUALIZAR INTEGER(1) DEFAULT 0,
-> BORRAR INTEGER(1) DEFAULT 0,
-> CONSULTAR INTEGER(1) DEFAULT 0
-> );
Query OK, 0 rows affected (0.03 sec)

MariaDB [salud]> DESC PERMISOXTABLAXUSUARIO;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| USUARIO | varchar(30) | YES | NULL | NULL |
| TABLA | varchar(30) | YES | NULL | NULL |
| INSERTAR | int(1) | YES | 0 |
| LISTAR | int(1) | YES | 0 |
| ACTUALIZAR | int(1) | YES | 0 |
| BORRAR | int(1) | YES | 0 |
| CONSULTAR | int(1) | YES | 0 |
+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)
```



Creamos la función “TIENEpermiso”, donde le pasaremos la operación y la tabla y el retornará 0 si no está autorizado o 1 si está autorizado para INSERTAR, ACTUALIZAR, BORRAR o LISTAR.

```
MariaDB [salud]> DELIMITER $
MariaDB [salud]> CREATE OR REPLACE FUNCTION TIENEpermiso(
-> POPER VARCHAR(1),
-> PTABLA VARCHAR(30)) RETURNS INTEGER
-> BEGIN
->   DECLARE PERMI INTEGER;
->   SELECT CASE
->     WHEN POPER='I' THEN INSERTAR
->     WHEN POPER='U' THEN ACTUALIZAR
->     WHEN POPER='D' THEN BORRAR
->     WHEN POPER='A' THEN LISTAR
->   END PERMISO INTO PERMI
->   FROM PERMISOXTABLAXUSUARIO
->   WHERE USUARIO=USER() AND TABLA=PTABLA;
->   RETURN PERMI;
->
-> END;
-> $
Query OK, 0 rows affected (0.01 sec)

MariaDB [salud]> DELIMITER ;
```



Modificamos el procedimiento CRUD y adicionamos las siguientes líneas:

```
IF(TIENEPERMISO(POPER,'EPS')=0) THEN
    SELECT 'NO TIENE PERMISO' AS MSG;
    LEAVE SP;
END IF;
```

```
MariaDB [salud]> DELIMITER $
MariaDB [salud]> CREATE OR REPLACE PROCEDURE PREPS(
-> IN POPER VARCHAR(1),
-> IN PID INTEGER,
-> IN PNOM VARCHAR(30),
-> PESTADO INTEGER)
-> SP: BEGIN
-> -- DECLARE HAY INTEGER;
-> IF(POPER IN('Q','U','D')) THEN
->     SELECT COUNT(*) INTO @B FROM EPS WHERE IDEPS=PID;
->     IF(@B=0) THEN
->         SELECT 'ID NO EXISTE' AS MSG;
->         LEAVE SP;
->     END IF;
-> END IF;
-> IF(TIENEPERMISO(POPER,'EPS')=0) THEN
->     SELECT 'NO TIENE PERMISO' AS MSG;
->     LEAVE SP;
-> END IF;
-> IF(POPER='A') THEN
->     SELECT * FROM EPS;
-> END IF;
-> IF(POPER='Q') THEN
->     SELECT * FROM EPS WHERE IDEPS=PID;
-> END IF;
-> IF(POPER='I') THEN
->     INSERT INTO EPS(NOMBRE,ESTADOEPS) VALUES(PNOM,PESTADO);
-> END IF;
-> IF(POPER='U' ) THEN
->     IF(PNOM IS NULL AND PESTADO IS NOT NULL) THEN
->         UPDATE EPS SET ESTADOEPS=PESTADO WHERE IDEPS=PID;
->     END IF;
->     IF(PNOM IS NOT NULL AND PESTADO IS NULL) THEN
->         UPDATE EPS SET NOMBRE=PNOM WHERE IDEPS=PID;
->     END IF;
->     IF(PNOM IS NOT NULL AND PESTADO IS NOT NULL) THEN
->         UPDATE EPS SET NOMBRE=PNOM, ESTADOEPS=PESTADO WHERE IDEPS=PID;
->     END IF;
->     INSERT INTO LOGAUDITORIA VALUES(USER(),NOW(),'U','EPS',PID)  ;
-> END IF;
-> IF(POPER='D') THEN
->     DELETE FROM EPS WHERE IDEPS=PID;
-> END IF;
-> SELECT 'OK' AS MSG;
-> END;
-> $
```

Query OK, 0 rows affected (0.01 sec)

```
MariaDB [salud]> DELIMITER ;
```



Se necesita ejecutar consulta en tiempo de ejecución (dinámica), se debe construir el siguiente procedimiento:

```
MariaDB [salud]> DELIMITER $
MariaDB [salud]> CREATE OR REPLACE PROCEDURE executeImmediate(
-> _query MEDIUMTEXT
-> )
-> MODIFIES SQL DATA
-> SQL SECURITY DEFINER
-> BEGIN
->
-> SET @sqlText = _query;
-> PREPARE stmt FROM @sqlText;
-> EXECUTE stmt;
-> DEALLOCATE PREPARE stmt;
->
-> COMMIT;END
-> $
Query OK, 0 rows affected (0.01 sec)

MariaDB [salud]> DELIMITER ;
```



Para crear o modificar los permisos a un usuario sobre una tabla en una opción en especial, debemos crear el procedimiento “PRPERMISOXTABLAXUSUARIO” que crea el registro si no existe y lo actualiza.

```
MariaDB [salud]> DELIMITER $
MariaDB [salud]> CREATE OR REPLACE PROCEDURE PRPERMISOXTABLAXUSUARIO(
-> IN PUSUARIO VARCHAR(30),
-> IN PTABLA VARCHAR(30),
-> IN PPRIVILEGIO VARCHAR(15),
-> IN PESTADO INTEGER)
-> BEGIN
->     DECLARE AUX TEXT;
->     SELECT COUNT(*) INTO @B
->     FROM PERMISOXTABLAXUSUARIO
->     WHERE USUARIO=PUSUARIO AND TABLA=PTABLA;
->
->     IF(@B=0) THEN
->         INSERT INTO PERMISOXTABLAXUSUARIO(USUARIO,TABLA) VALUES(PUSUARIO,PTABLA);
->     END IF;
->
->     SET AUX=
->     CONCAT('UPDATE PERMISOXTABLAXUSUARIO SET ','.',PPRIVILEGIO,'=',PESTADO,
->     ' WHERE USUARIO=''',PUSUARIO,''' AND TABLA=''',PTABLA,'''');
->     -- SELECT AUX;
->     CALL executeImmediate(AUX);
-> END;
-> $
Query OK, 0 rows affected (0.01 sec)
```



No se tiene permiso para listar todas la EPS

```
MariaDB [salud]> SELECT * FROM PERMISOXTABLAXUSUARIO;
+-----+-----+-----+-----+-----+-----+-----+
| USUARIO | TABLA | INSERTAR | LISTAR | ACTUALIZAR | BORRAR | CONSULTAR |
+-----+-----+-----+-----+-----+-----+-----+
| root@localhost | EPS | 0 | 0 | 1 | 0 | 0 |
| root@localhost | PERSONA | 0 | 0 | 1 | 0 | 0 |
| root@localhost | TIPOIDENTIFICACION | 0 | 1 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

MariaDB [salud]> CALL PREPS('A',NULL,NULL,NULL);
+-----+
| MSG |
+-----+
| NO TIENE PERMISO |
+-----+
1 row in set (0.03 sec)

Query OK, 0 rows affected (0.07 sec)
```



Un ejemplo de su utilización, dando permiso para listar todas las EPS

```
MariaDB [salud]> CALL PRPERMISOXTABLAXUSUARIO('root@localhost','EPS','LISTAR',1);
Query OK, 0 rows affected (0.02 sec)

MariaDB [salud]> SELECT * FROM PERMISOXTABLAXUSUARIO;
+-----+-----+-----+-----+-----+-----+-----+
| USUARIO | TABLA | INSERTAR | LISTAR | ACTUALIZAR | BORRAR | CONSULTAR |
+-----+-----+-----+-----+-----+-----+-----+
| root@localhost | EPS | 0 | 1 | 1 | 0 | 0 |
| root@localhost | PERSONA | 0 | 0 | 1 | 0 | 0 |
| root@localhost | TIPOIDENTIFICACION | 0 | 1 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

MariaDB [salud]> CALL PREPS('A',NULL,NULL,NULL);
+-----+-----+
| ideps | nombre | estadoeps |
+-----+-----+
| 1 | SALUDCOOP | 4 |
| 2 | CAFESALUD | 1 |
| 3 | NUEVA EPS | 1 |
| 4 | ISS | 5 |
| 5 | CAPRECOM | 4 |
| 6 | COOMEVA | 1 |
| 7 | COMPENSAR | 1 |
| 8 | COLMEDICA | 1 |
| 9 | HUMANA VIVIR | 2 |
| 10 | FAMISANAR | 1 |
| 11 | CRUZ BLANCA | 3 |
| 12 | SOLSALUD | 1 |
| 13 | SALUD TOTAL | 5 |
| 14 | SANITAS | 3 |
| 20 | BUENISIMA | 2 |
| 23 | LA BUENISIMA | 1 |
+-----+-----+
16 rows in set (0.00 sec)

+-----+
| MSG |
+-----+
| OK |
+-----+
1 row in set (0.12 sec)

Query OK, 0 rows affected (0.14 sec)
```



AUDITORIA DE LOS PROCEDIMIENTOS CRUD.

Para esto crearemos la tabla “LOGAUDITORIA”

```
MariaDB [salud]> CREATE TABLE LOGAUDITORIA(
-> USUARIO VARCHAR(30),
-> FECHA TIMESTAMP,
-> OPERACION VARCHAR(1),
-> TABLA VARCHAR(30),
-> ID INTEGER);
Query OK, 0 rows affected (0.04 sec)
```



Se debe insertar esta línea de código:

```
MariaDB [salud]> DELIMITER $
MariaDB [salud]> CREATE OR REPLACE PROCEDURE PREPS(
-> IN POPER VARCHAR(1),
-> IN PID INTEGER,
-> IN PNOM VARCHAR(30),
-> PESTADO INTEGER)
-> SP: BEGIN
-> -- DECLARE HAY INTEGER;
-> IF(POPER IN('Q','U','D')) THEN
->   SELECT COUNT(*) INTO @B FROM EPS WHERE IDEPS=PID;
->   IF(@B=0) THEN
->     SELECT 'ID NO EXISTE' AS MSG;
->     LEAVE SP;
->   END IF;
-> END IF;
-> IF(TIENEPERMISO(POPER,'EPS')=0) THEN
->   SELECT 'NO TIENE PERMISO' AS MSG;
->   LEAVE SP;
-> END IF;
-> IF(POPER='A') THEN
->   SELECT * FROM EPS;
-> END IF;
-> IF(POPER='Q') THEN
->   SELECT * FROM EPS WHERE IDEPS=PID;
-> END IF;
-> IF(POPER='I') THEN
->   INSERT INTO EPS(NOMBRE,ESTADOEPS) VALUES(PNOM,PESTADO);
-> END IF;
-> IF(POPER='U' ) THEN
->   IF(PNOM IS NULL AND PESTADO IS NOT NULL) THEN
->     UPDATE EPS SET ESTADOEPS=PESTADO WHERE IDEPS=PID;
->   END IF;
->   IF(PNOM IS NOT NULL AND PESTADO IS NULL) THEN
->     UPDATE EPS SET NOMBRE=PNOM WHERE IDEPS=PID;
->   END IF;
->   IF(PNOM IS NOT NULL AND PESTADO IS NOT NULL) THEN
->     UPDATE EPS SET NOMBRE=PNOM, ESTADOEPS=PESTADO WHERE IDEPS=PID;
->   END IF;
->   INSERT INTO LOGAUDITORIA VALUES(USER(),NOW(),'U','EPS',PID) ;
-> END IF;
-> IF(POPER='D') THEN
->   DELETE FROM EPS WHERE IDEPS=PID;
-> END IF;
-> SELECT 'OK' AS MSG;
-> END;
-> $
```

Query OK, 0 rows affected (0.01 sec)

ADMINISTRACION DE USUARIOS

Creamos un usuario con los mismos privilegios que el usuario root. Para esta otorgamos todos los privilegios para todas las base de datos existentes.

```
MariaDB [salud]> CREATE USER 'YO'@'localhost' IDENTIFIED BY 'YO1234';
Query OK, 0 rows affected (0.00 sec)

MariaDB [salud]> GRANT ALL PRIVILEGES ON *.* TO 'YO'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Ingresamos con el usuario YO, verificamos el usuario y vemos los privilegios otorgados.

```
# mysql --user=YO --password=YO1234
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 23
Server version: 10.1.37-MariaDB mariadb.org binary distribution

copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SELECT USER();
+-----+
| USER() |
+-----+
| YO@localhost |
+-----+
1 row in set (0.00 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'YO'@'localhost';
+-----+
| Grants for YO@localhost
|   |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'YO'@'localhost' IDENTIFIED BY PASSWORD '*1E94EB84A625B8B0F591555EB37C7C2EE
EA22' |
+-----+
1 row in set (0.00 sec)
```

Continuando con el taller, vamos a crear un usuario para la señora Zoila Vaca que tendra acceso a los objetos de la base de datos HR y solo podra realizar SELECT, INSERT y UPDATE; los demas privilegios los tiene negado, lo mismo ingresar a otra base de datos diferente a HR.

```
MariaDB [(none)]> CREATE USER ZVACA IDENTIFIED BY 'prueba1234';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT SELECT,INSERT,UPDATE ON HR.* TO 'ZVACA'@'localhost' IDENTIFIED BY 'prueba1234';
Query OK, 0 rows affected (0.00 sec)
```

Ingresamos a MySql con el usuario ZVACA y la clave prueba1234, ingresamos a la base de datos HR, verificamos el usuario y borramos el departamento cuyo ID es el 280.

```
# mysql --user=ZVACA --password=prueba1234
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 101
Server version: 10.1.37-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE HR
Database changed
MariaDB [HR]> INSERT INTO DEPARTMENTS VALUES(280,'ASUNTOS SIN IMPORTANCIA',NULL,NULL);
Query OK, 1 row affected (0.01 sec)

MariaDB [HR]> UPDATE DEPARTMENTS SET DEPARTMENT_NAME='ASUNTOS CULINARIOS' WHERE DEPARTMENT_ID=280;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  changed: 1  Warnings: 0

MariaDB [HR]> SELECT USER();
+-----+
| USER() |
+-----+
| ZVACA@localhost |
+-----+
1 row in set (0.00 sec)

MariaDB [HR]> DELETE FROM DEPARTMENTS WHERE DEPARTMENT_ID=280;
ERROR 1142 (42000): DELETE command denied to user 'ZVACA'@'localhost' for table 'departments'
MariaDB [HR]>
```

Le revocamos los permisos de INSERT y UPDATE al usuario ZVACA desde el usuario root.

```
MariaDB [HR]> REVOKE INSERT,UPDATE,DELETE ON HR.* FROM ZVACA@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Verificamos que permisos tiene el usuario.

```
MariaDB [HR]> REVOKE INSERT,UPDATE,DELETE ON HR.* FROM ZVACA@'localhost';
Query OK, 0 rows affected (0.00 sec)

MariaDB [HR]> SHOW GRANTS FOR ZVACA@'localhost';
+-----+
| Grants for ZVACA@localhost
+-----+
| GRANT USAGE ON *.* TO 'ZVACA'@'localhost' IDENTIFIED BY PASSWORD '*6EABC9EFD6A71EB363D4E5AEED54E5977DC24F4A'
| GRANT SELECT ON `hr`.* TO 'ZVACA'@'localhost'
| -----
+-----+
2 rows in set (0.00 sec)
```

Ingrese con el usuario ZVACA y use la base de datos SALUD y realizar un select sobre la tabla DEPARTMENTS. Explique que pasa.

```
MariaDB [(none)]> USE SALUD
ERROR 1044 (42000): Access denied for user 'ZVACA'@'localhost' to database 'salud'
MariaDB [(none)]> USE HR
Database changed
MariaDB [HR]> SELECT * FROM DEPARTMENTS;
+-----+-----+-----+-----+
| department_id | department_name | manager_id | location_id |
+-----+-----+-----+-----+
|      10 | Administration |        200 |       1700 |
|      20 | Marketing     |        201 |       1800 |
|      30 | Purchasing    |       114 |       1700 |
|      40 | Human Resources |      203 |       2400 |
|      50 | Shipping      |       121 |       1500 |
|      60 | IT            |       103 |       1400 |
|      70 | Public Relations |      204 |       2700 |
|      80 | Sales          |       145 |       2500 |
|      90 | Executive     |       100 |       1700 |
|     100 | Finance        |       108 |       1700 |
|     110 | Accounting    |       205 |       1700 |
|     120 | Treasury       |        NULL |       1700 |
|     130 | Corporate Tax  |        NULL |       1700 |
|     140 | Control And Credit |      NULL |       1700 |
|     150 | Shareholder Services |      NULL |       1700 |
|     160 | Benefits       |        NULL |       1700 |
|     170 | Manufacturing  |        NULL |       1700 |
|     180 | Construction   |        NULL |       1700 |
|     190 | Contracting   |        NULL |       1700 |
|     200 | Operations     |        NULL |       1700 |
|     210 | IT Support     |        NULL |       1700 |
|     220 | NOC            |        NULL |       1700 |
|     230 | IT Helpdesk    |        NULL |       1700 |
|     240 | Government Sales |      NULL |       1700 |
|     250 | Retail Sales    |        NULL |       1700 |
|     260 | Recruiting     |        NULL |       1700 |
|     270 | Payroll         |        NULL |       1700 |
|     280 | ASUNTOS CULINARIOS |      NULL |        NULL |
+-----+-----+-----+-----+
28 rows in set (0.00 sec)
```

Ingrese un nuevo departamento llamado JUGUETERIA

Actualiza el departamento 280 por ASUNTOS JUDICIALES.

Borre el departamento 280.

Explique si lo logro, sino lo puede hacer explique el por qué'

```
MariaDB [HR]> SELECT USER();
+-----+
| USER() |
+-----+
| ZVACA@localhost |
+-----+
1 row in set (0.00 sec)

MariaDB [HR]> INSERT INTO DEPARTMENTS VALUES(281,'JUGUETERIA',NULL,NULL);
ERROR 1142 (42000): INSERT command denied to user 'ZVACA'@'localhost' for table 'departments'
MariaDB [HR]> UPDATE DEPARTMENTS SET DEPARTMENT_NAME='ASUNTOS JUDICIALES' WHERE DEPARTMENT_ID=280;
ERROR 1142 (42000): UPDATE command denied to user 'ZVACA'@'localhost' for table 'departments'
MariaDB [HR]> DELETE FROM DEPARTMENTS WHERE DEPARTMENT_ID=280;
ERROR 1142 (42000): DELETE command denied to user 'ZVACA'@'localhost' for table 'departments'
```

LLAMADO DE LOS PROCEDIMIENTOS CRUD

```
1 <?php
2 $mysqli = new mysqli("", "root", "", "salud");
3 if (!$mysqli->multi_query("CALL preps('I',NULL,'YYYYYYYY',1)")) {
4     echo "Falló CALL: (" . $mysqli->errno . ") " . $mysqli->error;
5 }
```

MariaDB [salud]> SELECT * FROM EPS;		
ideps	nombre	estadoeps
1	SALUDCOOP	4
2	CAFESALUD	1
3	NUEVA EPS	1
4	ISS	5
5	CAPRECOM	4
6	COOMEVA	1
7	COMPENSAR	1
8	COLMEDICA	1
9	HUMANA VIVIR	2
10	FAMISANAR	1
11	CRUZ BLANCA	3
12	SOLSALUD	1
13	SALUD TOTAL	5
14	SANITAS	3
16	LA HERMOSA	1
17	xxx	1
18	YYYYYYYY	1
17 rows in set (0.00 sec)		

```

1 <?php
2 $f = mysqli_connect("", "root", "", "salud");
3 $sentencia="CALL preps('A',16,NULL,NULL)";
4 $a=mysqli_query($f,$sentencia);
5 echo "<table border><th>ID</th><th>NOMBRE</th><th>ESTADO</th></tr>";
6 while($e=mysqli_fetch_array($a,MYSQLI_NUM)){
7 echo "<tr><td>".$e[0]."</td>";
8 echo "<td>".$e[1]."</td>";
9 echo "<td>".$e[2]."</td>";
10 echo "</tr>";
11 }
12 echo "</table>";
13 ?>

```

ID	NOMBRE	ESTADO
1	SALUDCOOP	4
2	CAFESALUD	1
3	NUEVA EPS	1
4	ISS	5
5	CAPRECOM	4
6	COOMEVA	1
7	COMPENSAR	1
8	COLMEDICA	1
9	HUMANA VIVIR	2
10	FAMISANAR	1
11	CRUZ BLANCA	3
12	SOLSALUD	1
13	SALUD TOTAL	5
14	SANITAS	3
16	LA HERMOSA	1
17	xxx	1
18	YYYYYYYYYY	1

```

1 <?php
2 error_reporting(0);
3 $f = mysqli_connect("", "root", "", "salud");
4 $sentencia="CALL preps('Q',26,NULL,NULL)";
5 $a=mysqli_query($f,$sentencia);
6 echo "<table border><th>ID</th><th>NOMBRE</th><th>ESTADO</th></tr>";
7 while($e=mysqli_fetch_array($a,MYSQLI_NUM)){
8 echo "<tr><td>".$e[0]."</td>";
9 echo "<td>".$e[1]."</td>";
10 echo "<td>".$e[2]."</td>";
11 echo "</tr>";
12 }
13 echo "</table>";
14 ?>

```

ID	NOMBRE	ESTADO
ID NO EXISTE		

```

1 <?php
2 error_reporting(0);
3 $f = mysqli_connect("", "root", "", "salud");
4 $sentencia="CALL preps('D',18,NULL,NULL)";
5 $a=mysqli_query($f,$sentencia);
6 echo "<table border><th>ID</th><th>NOMBRE</th><th>ESTADO</th></tr>";
7 while($e=mysqli_fetch_array($a,MYSQLI_NUM)){
8 echo "<tr><td>".$e[0]."</td>";
9 echo "<td>".$e[1]."</td>";
10 echo "<td>".$e[2]."</td>";
11 echo "</tr>";
12 }
13 echo "</table>";
14 ?>

```

ID	NOMBRE	ESTADO
NO TIENE PERMISO		

```

1  <?php
2   error_reporting(0);
3   $f = mysqli_connect("", "root", "", "salud");
4   $sentencia="SELECT * FROM PERMISOXTABLAXUSUARIO";
5   $a=mysqli_query($f,$sentencia);
6   echo "<br><br><table border><th>USUARIO</th><th>TABLA</th><th>INSERTAR</th><th>LISTAR</th><th>
7   ACTUALIZAR</th><th>BORRAR</th><th>CONSULTAR</th></tr>";
8   while($e=mysqli_fetch_array($a,MYSQLI_NUM)){
9     echo "<tr><td>".$e[0]."</td>";
10    echo "<td>".$e[1]."</td>";
11    echo "<td>".$e[2]."</td>";
12    echo "<td>".$e[3]."</td>";
13    echo "<td>".$e[4]."</td>";
14    echo "<td>".$e[5]."</td>";
15    echo "<td>".$e[6]."</td>";
16    echo "</tr>";
17  }
18  echo "</table>";
?>
```

USUARIO	TABLA	INSERTAR	LISTAR	ACTUALIZAR	BORRAR	CONSULTAR
root@localhost	EPS	1	1	0	0	1

```

1  <?php
2   error_reporting(0);
3   $f = mysqli_connect("", "root", "", "salud");
4   $sentencia="CALL PRPERMISOXTABLAXUSUARIO('root@localhost','EPS','BORRAR',1)";
5   $a=mysqli_query($f,$sentencia);
6   /*
7   echo "<table border><th>ID</th><th>NOMBRE</th><th>ESTADO</th></tr>";
8   while($e=mysqli_fetch_array($a,MYSQLI_NUM)){
9     echo "<tr><td>".$e[0]."</td>";
10    echo "<td>".$e[1]."</td>";
11    echo "<td>".$e[2]."</td>";
12    echo "</tr>";
13  }
14  echo "</table>";
15  */
16  ?>
```

USUARIO	TABLA	INSERTAR	LISTAR	ACTUALIZAR	BORRAR	CONSULTAR
root@localhost	EPS	1	1	0	1	1

```

error_reporting(0);
$f = mysqli_connect("", "root", "", "salud");
$sentencia="CALL PREPS('D',18,NULL,NULL)";
$a=mysqli_query($f,$sentencia);
/*
```

```
MariaDB [salud]> SELECT * FROM EPS;
+-----+-----+-----+
| ideps | nombre          | estadoeps |
+-----+-----+-----+
|     1 | SALUDCOOP        |         4   |
|     2 | CAFESALUD        |         1   |
|     3 | NUEVA EPS         |         1   |
|     4 | ISS               |         5   |
|     5 | CAPRECOM          |         4   |
|     6 | COOMEVA          |         1   |
|     7 | COMPENSAR         |         1   |
|     8 | COLMEDICA         |         1   |
|     9 | HUMANA VIVIR      |         2   |
|    10 | FAMISANAR         |         1   |
|    11 | CRUZ BLANCA       |         3   |
|    12 | SOLSALUD          |         1   |
|    13 | SALUD TOTAL        |         5   |
|    14 | SANITAS           |         3   |
|    16 | LA HERMOSA         |         1   |
|    17 | xxx               |         1   |
+-----+-----+-----+
16 rows in set (0.00 sec)
```

MANEJO DE EXCEPCIONES

Los errores que ocurren al ejecutar sentencias en MYSQL se pueden atrapar mediante las excepciones o manejadores de errores.

Un ejemplo de esto sería:

SIGNAL SQLSTATE '45001'

```
SET MESSAGE_TEXT='ID NO EXISTE',
MYSQL_ERRNO=1001;
```

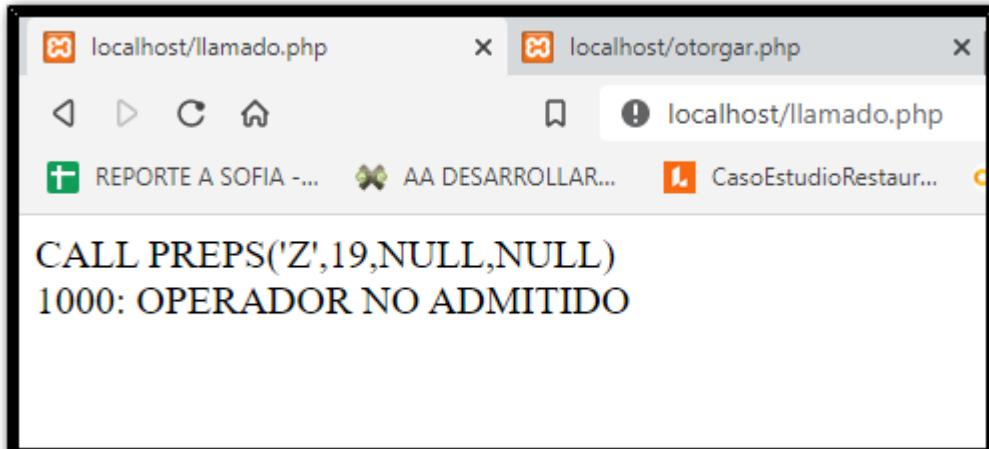
```
IF(POPER IN('Q','U','D')) THEN
    SELECT COUNT(*) INTO @B FROM EPS WHERE IDEPS=PID;
    IF(@B=0) THEN
        -- SELECT 'ID NO EXISTE' AS MSG;
        INSERT INTO LOGAUDITORIA VALUES(USER(),NOW(),'N','EPS/ACCESO DENEGADO',PID) ;
        SIGNAL SQLSTATE '45001'
        SET MESSAGE_TEXT='ID NO EXISTE',
        MYSQL_ERRNO=1001;
        LEAVE SP;
    END IF;
END IF;
```

CALL PREPS('A',19,NULL,NULL)

ID	NOMBRE	ESTADO
1	SALUDCOOP	4
2	CAFESALUD	1
3	NUEVA EPS	1
4	ISS	5
5	CAPRECOM	4
6	COOMEVA	1
7	COMPENSAR	1
8	COLMEDICA	1
9	HUMANA VIVIR	2
10	FAMISANAR	1
11	CRUZ BLANCA	3
12	SOLSALUD	1
13	SALUD TOTAL	5
17	LA PODEROSA	1
19	LA ESTRELLITA	1



```
IF(POPER NOT IN('Q','U','D','A','I')) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT='OPERADOR NO ADMITIDO',
    MYSQL_ERRNO=1000;
    LEAVE SP;
END IF;
```

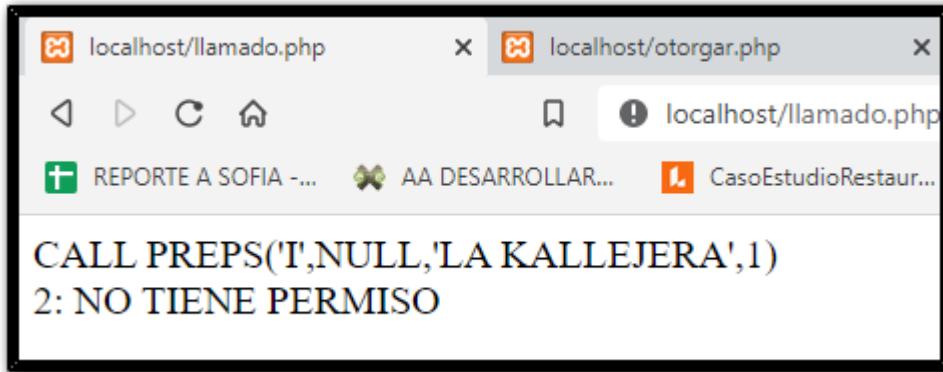




CALL PREPS('D',19,NULL,NULL)

ID	NOMBRE	ESTADO
1	SALUDCOOP	4
2	CAFESALUD	1
3	NUEVA EPS	1
4	ISS	5
5	CAPRECOM	4
6	COOMEVA	1
7	COMPENSAR	1
8	COLMEDICA	1
9	HUMANA VIVIR	2
10	FAMISANAR	1
11	CRUZ BLANCA	3
12	SOLSALUD	1
13	SALUD TOTAL	5
17	LA PODEROSA	1





CALL PREPS('T',NULL,'LA KALLEJERA',1)

ID	NOMBRE	ESTADO
23	LA KALLEJERA	1

CALL PREPS('U',23,'LA PREVISORA',1)

ID	NOMBRE	ESTADO
23	LA PREVISORA	1

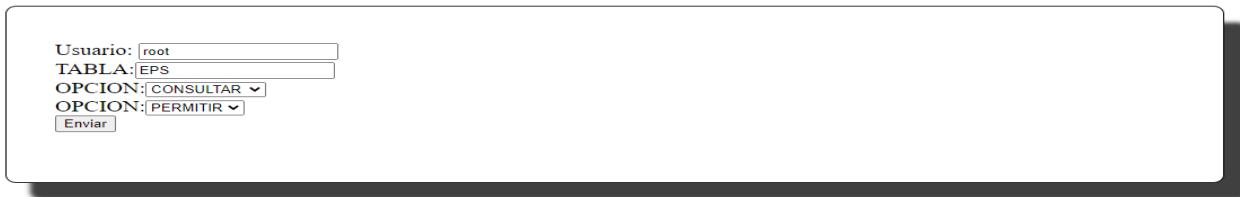
Creamos un archivo llamado “otorgar.php”

```

1  <link href="https://egesussoft.com/analitica/EXE1.css" rel="stylesheet" /><meta charset="utf-8">
2  <center>
3      <table id=tabla>
4          <tr>
5              <td>
6      <form action=ingresa.php>
7  Usuario:
8  <input type=text name="usua"><br>
9  TABLA:<input type=text name="tabla"><br>
10 OPCION:<select name="opcion">
11     <option value=INSERTAR>INSERTAR</option>
12     <option value=LISTAR>LISTAR</option>
13     <option value=ACTUALIZAR>ACTUALIZAR</option>
14     <option value=BORRAR>BORRAR</option>
15     <option value=CONSULTAR>CONSULTAR</option>
16 </select><br>
17 OPCION:<select name="permiso">
18     <option value=0>NEGAR</option>
19     <option value=1>PERMITIR</option>
20 </select><br>
21 <input type="submit" value="Enviar">
22 </form>
23         </td>
24     </tr>
25

```

Ejecutamos “<http://localhost/otorgar.php>”



The form contains the following fields:

- Usuario: root
- TABLA: EPS
- OPCION: CONSULTAR
- OPCION: PERMITIR
- Enviar

Creamos un archivo llamado “ingresa.php”.

```
1 <?php
2     error_reporting(0);
3     $f = mysqli_connect("", "root", "", "salud");
4
5     $sentencia="CALL PRPERMISOXTABLAXUSUARIO('".$_GET["usua"]."','@localhost'";
6     $sentencia=$sentencia."'','".$_GET["tabla"]."','".$_GET["opcion"]."','".$_GET["permiso"]."')";
7     echo $sentencia;
8     $a=mysqli_query($f,$sentencia);
9     ?>
10    <script type="text/javascript">
11        history.back();
12    </script>
```

La lista de errores generados desde MYSQL, se pueden ver en:

<https://manuales.guebs.com/mysql-5.0/error-handling.html>

Recomendamos ver el siguiente video: [AQUÍ](#)

RETO POR REALIZAR

Para desarrollar este reto se debe tener en cuenta los siguientes pasos:

- Identifique la entidades y relaciones de acuerdo con los modelos de diseño entregados.
- Crear el modelo físico de acuerdo con el modelo relación diseñado
- No se permiten realizar sentencias DML directas, estas se deben realizar mediante procedimientos fabricados para cada tabla de referencia que permita hacer matriz CRUD (Create-Read-Update-Delete).
- Todas las transacciones realizadas se deben registrar en una tabla llamada “LOGAUDITORIA”, esta acción se debe realizar mediante TRIGGERS.
- Esta tabla debe ser consultada mediante una vista de usuario que permita realizar consultas por medio de criterios de fecha, usuario, acción y tablas.
- Desarrolle tabla de ROL que permita controlar el ingreso a los diferente módulos del sistema.

Realizado por el instructor José Fernando
Galindo Suárez
jgalindos@sena.edu.co 2020

