

**Universidad Autónoma de Nuevo León**

**FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS**

**UA:**

**BASE DE DATOS MULTIMEDIA**

**GPO 001**

**Raul Eleazar Hernández Campos 1804813**

**Jair Asael Lozano Fraire 1823161**

índice de Contenido

[Diagrama MySQL 3](#_Toc117199069)

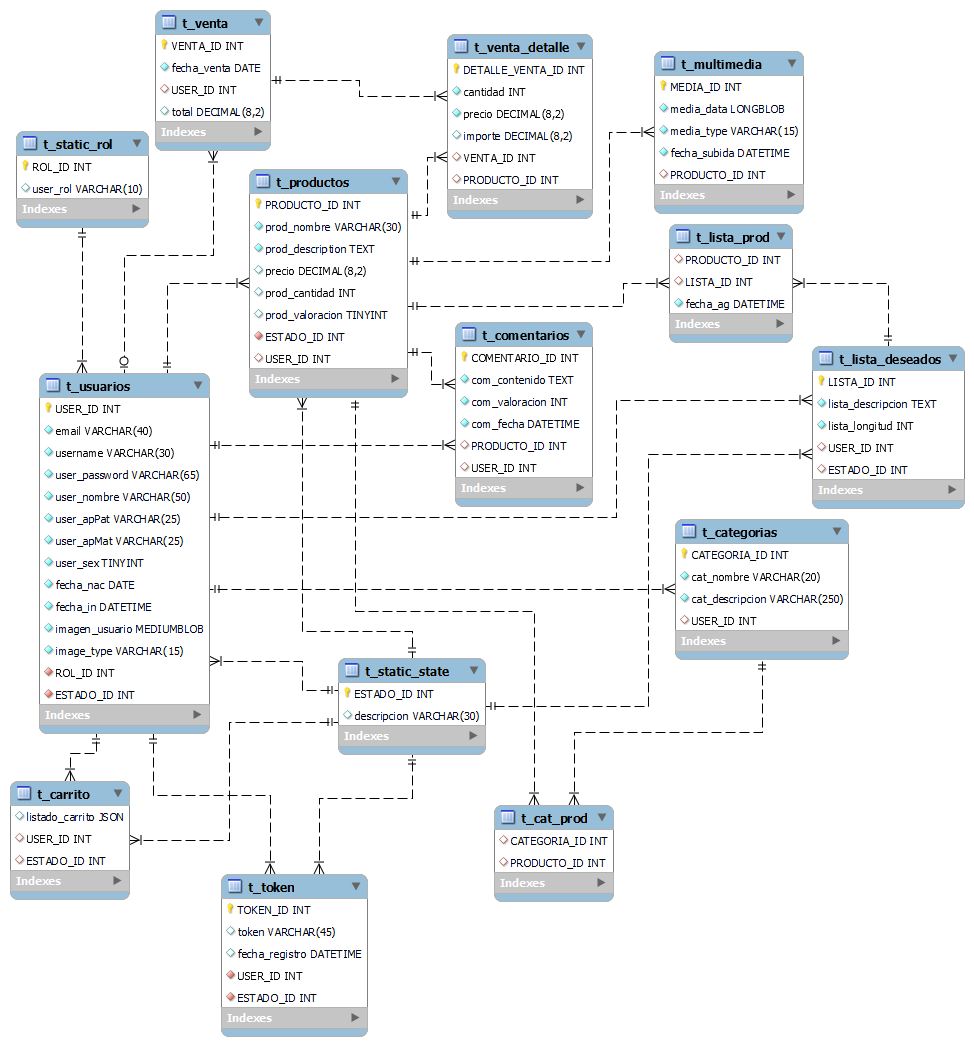
[Creación de tablas 4](#_Toc117199070)

[SP Para la página de usuarios 10](#_Toc117199071)

[Script para la creación del diccionario de datos 13](#_Toc117199072)

[Diccionario de datos obtenido (TABLA DE EXCEL EN WORD) 17](#_Toc117199073)

# Diagrama MySQL

****

# Creación de tablas

drop database if exists db\_cyberkong;

create database db\_cyberkong;

use db\_cyberkong;

create table t\_usuarios(

USER\_ID int auto\_increment, -- PK

email varchar(40) not null unique,

username varchar(30) not null unique,

user\_password varchar(65) not null,

user\_nombre varchar(50) not null,

user\_apPat varchar(25) not null,

user\_apMat varchar(25) not null,

user\_sex tinyint not null,

fecha\_nac date not null,

fecha\_in datetime not null,

imagen\_usuario mediumblob not null,

image\_type varchar(15) not null,

ROL\_ID int not null, -- FK

ESTADO\_ID int not null, -- FK

constraint PK\_usuarios

    primary key(USER\_ID)

);

create table t\_static\_rol(

ROL\_ID int, -- PK

user\_rol varchar(10),

constraint PK\_rol

    primary key(ROL\_ID)

);

create table t\_static\_state(

ESTADO\_ID int, -- PKt\_multimedia

descripcion varchar(30),

constraint PK\_state

    primary key(ESTADO\_ID)

);

create table t\_token(

TOKEN\_ID int auto\_increment, -- PK

token varchar(45) DEFAULT NULL,

fecha\_registro datetime DEFAULT NULL,

USER\_ID int not null, -- FK

ESTADO\_ID int not null, -- FK

constraint PK\_token

    primary key(TOKEN\_ID)

);

create table t\_productos(

PRODUCTO\_ID int auto\_increment,

prod\_nombre varchar(30) not null,

prod\_description text not null,

precio decimal(8,2),

prod\_cantidad int,

prod\_valoracion tinyint,

ESTADO\_ID int not null, -- FK

USER\_ID int, -- FK

constraint PK\_productos

    primary key(PRODUCTO\_ID)

);

create table t\_multimedia(

MEDIA\_ID int auto\_increment,

media\_data mediumblob not null,

media\_type varchar(15) not null,

fecha\_subida datetime not null,

PRODUCTO\_ID int, -- FK

constraint PK\_multimedia

    primary key(MEDIA\_ID)

);

-- tabla asociativa

-- create table t\_media\_productos(

-- MEDIA\_ID int,  -- FK

-- PRODUCTO\_ID int, -- FK

-- );

create table t\_comentarios(

COMENTARIO\_ID int auto\_increment, -- PK

com\_contenido text not null,

com\_valoracion int not null,

com\_fecha datetime not null,

PRODUCTO\_ID int, -- FK

USER\_ID int, -- FK

constraint PK\_comentarios

    primary key (COMENTARIO\_ID)

);

create table t\_lista\_deseados(

LISTA\_ID int auto\_increment, -- PK

lista\_descripcion text not null,

lista\_longitud int not null,

USER\_ID int, -- FK

ESTADO\_ID int, -- FK

constraint PK\_lista\_deseados

    primary key(LISTA\_ID)

);

-- tabla asociativa

create table t\_lista\_prod(

PRODUCTO\_ID int, -- FK

LISTA\_ID int, -- FK

fecha\_ag datetime not null

);

create table t\_carrito(

listado\_carrito json default null,

USER\_ID int, -- FK

ESTADO\_ID int -- FK

);

create table t\_venta(

VENTA\_ID int auto\_increment, -- PK

fecha\_venta date not null,

total decimal(8,2),

USER\_ID int, -- FK

constraint PK\_venta

    primary key(VENTA\_ID)

);

create table t\_venta\_detalle(

DETALLE\_VENTA\_ID int auto\_increment, -- PK

cantidad int not null,

precio decimal(8,2) not null,

importe decimal(8,2),

VENTA\_ID int, -- FK

PRODUCTO\_ID int, -- FK

constraint PK\_detalle\_venta

    primary key(DETALLE\_VENTA\_ID)

);

create table t\_categorias(

CATEGORIA\_ID int auto\_increment, -- PK

cat\_nombre varchar(20) not null,

cat\_descripcion varchar(250) not null,

USER\_ID int, -- FK

constraint PK\_cat

    primary key(CATEGORIA\_ID)

);

-- tabla asociativa

create table t\_cat\_prod(

CATEGORIA\_ID int, -- FK

PRODUCTO\_ID int -- FK

);

-- FK Constraints

alter table t\_usuarios

    add constraint FK\_usuarios\_rol

        foreign key(ROL\_ID)

        references t\_static\_rol(ROL\_ID),

    add constraint FK\_usuarios\_state

        foreign key(ESTADO\_ID)

        references t\_static\_state(ESTADO\_ID);

alter table t\_token

    add constraint FK\_token\_user

        foreign key(USER\_ID)

        references t\_usuarios(USER\_ID),

    add constraint FK\_token\_state

        foreign key(ESTADO\_ID)

        references t\_static\_state(ESTADO\_ID);

alter table t\_productos

    add constraint FK\_prod\_state

        foreign key(ESTADO\_ID)

        references t\_static\_state(ESTADO\_ID),

    add constraint FK\_prod\_user

        foreign key(USER\_ID)

        references t\_usuarios(USER\_ID);

alter table t\_multimedia

    add constraint FK\_Multimedia\_Prod

        foreign key(PRODUCTO\_ID)

        references t\_productos(PRODUCTO\_ID);

alter table t\_comentarios

    add constraint FK\_comment\_prod

        foreign key(PRODUCTO\_ID)

        references t\_productos(PRODUCTO\_ID),

    add constraint FK\_comment\_user

        foreign key(USER\_ID)

        references t\_usuarios(USER\_ID);

alter table t\_lista\_deseados

    add constraint FK\_lista\_user

        foreign key(USER\_ID)

        references t\_usuarios(USER\_ID),

    add constraint FK\_lista\_state

        foreign key(ESTADO\_ID)

        references t\_static\_state(ESTADO\_ID);

alter table t\_lista\_prod

    add constraint FK\_prod\_li

        foreign key(PRODUCTO\_ID)

        references t\_productos(PRODUCTO\_ID),

    add constraint FK\_lista\_li

        foreign key(LISTA\_ID)

        references t\_lista\_deseados(LISTA\_ID);

alter table t\_carrito

    add constraint FK\_carrito\_user

        foreign key(USER\_ID)

        references t\_usuarios(USER\_ID),

    add constraint FK\_carrito\_state

        foreign key(ESTADO\_ID)

        references t\_static\_state(ESTADO\_ID);

alter table t\_venta

    add constraint FK\_venta\_comprador

        foreign key(USER\_ID)

        references t\_usuarios(USER\_ID);

alter table t\_venta\_detalle

    add constraint FK\_detalleVenta\_venta

        foreign key(VENTA\_ID)

        references t\_venta(VENTA\_ID),

    add constraint FK\_detalleVenta\_producto

        foreign key(PRODUCTO\_ID)

        references t\_productos(PRODUCTO\_ID);

alter table t\_categorias

    add constraint FK\_cat\_user

        foreign key(USER\_ID)

        references t\_usuarios(USER\_ID);

alter table t\_cat\_prod

    add constraint FK\_categoria

        foreign key(CATEGORIA\_ID)

        references t\_categorias(CATEGORIA\_ID),

    add constraint FK\_producto\_cat

        foreign key(PRODUCTO\_ID)

        references t\_productos(PRODUCTO\_ID);

-- static inserts

insert into t\_static\_state(ESTADO\_ID, descripcion)

values

(1, "USUARIO PUBLICO"),

(2, "USUARIO PRIVADO"),

(3, "USUARIO ELIMINADO"),

(4, "LISTA PUBLICA"),

(5, "LISTA PRIVADA"),

(6, "CARRITO ACTIVO"),

(7, "CARRITO VACIO"),

(8, "PRODUCTO AGOTADO"),

(9, "PRODUCTO COTIZABLE"),

(10, "PRODUCTO DISPONIBLE"),

(11, "PRODUCTO ELIMINADO"),

(12, "TOKEN ACTIVO"),

(13, "TOKEN INACTIVO");

insert into t\_static\_rol(ROL\_ID, user\_rol)

values

(1, "COMPRADOR"),

(2, "VENDEDOR"),

(3, "ADMIN"),

(4, "SUPER\_AD");

# SP Para la página de usuarios

USE `db\_cyberkong`;

DROP procedure IF EXISTS `db\_cyberkong`.`sp\_UsuarioActions`;

;

DELIMITER $$

USE `db\_cyberkong`$$

CREATE PROCEDURE `sp\_UsuarioActions`(

in \_action tinyint,

in \_user\_id int,

in \_email varchar(40),

in \_username varchar(30),

in \_userpassword varchar(65),

in \_nombre varchar(50),

in \_apPat varchar(25),

in \_apMat varchar(25),

in \_sexo tinyint,

in \_fechaNacimiento date,

in \_fechaIngreso datetime,

in \_rol\_id int,

in \_state\_id int,

in \_img\_data mediumblob,

in \_img\_type varchar(15)

)

BEGIN

declare image\_id int;

    if \_action = 1 then

        insert into

        t\_usuarios

        (email, username, user\_password, user\_nombre,

        user\_apPat, user\_apMat, user\_sex, fecha\_nac,

        fecha\_in, imagen\_usuario, image\_type, ROL\_ID, ESTADO\_ID)

        value

        (\_email, \_username, \_userpassword, \_nombre,

        \_apPat, \_apMat, \_sexo, \_fechaNacimiento,

        \_fechaIngreso, \_file\_data, \_img\_type, \_rol\_id, \_state\_id);

    elseif \_action = 2 then

        update t\_usuarios set

        email = ifnull(\_email, t\_usuarios.email),

        username = ifnull(\_username, t\_usuarios.username),

        user\_password = ifnull(\_userpassword, t\_usuarios.user\_password),

        user\_nombre = ifnull(\_nombre, t\_usuarios.user\_nombre),

        user\_apPat = ifnull(\_apPat, t\_usuarios.user\_apPat),

        user\_apMat = ifnull(\_apMat, t\_usuarios.user\_apMat),

        user\_sex = ifnull(\_sexo, t\_usuarios.user\_sex),

        fecha\_nac = ifnull(\_fechaNacimiento, t\_usuarios.fecha\_nac),

        ROL\_ID = ifnull(\_rol\_id, t\_usuarios.ROL\_ID),

        image\_usuario = ifnull(\_img\_data, t\_usuarios.image\_usuario),

        image\_type = ifnull(\_img\_type, t\_usuarios.image\_type)

        where USER\_ID = \_user\_id;

    elseif \_action = 3 then

        update t\_usuarios

        set ESTADO\_ID = ifnull(\_state\_id, t\_usuario.ESTADO\_ID)

        where USER\_ID = \_user\_id;

    else

        SELECT "ACTION NOT EXISTS" AS 'ERROR MESSAGE';

    end if;

END$$

DELIMITER ;

USE `db\_cyberkong`;

DROP procedure IF EXISTS `db\_cyberkong`.`sp\_UsuarioReads`;

;

DELIMITER $$

USE `db\_cyberkong`$$

CREATE PROCEDURE `sp\_UsuarioReads`(

in \_action tinyint,

in \_username varchar(30),

in \_password varchar(65),

in \_userId int,

in \_email varchar(40)

)

BEGIN

-- OBTENER USUARIO CON ID

    if \_action = 1 then

        if \_username is not null then

            SELECT

            u.username as "USUARIO",

            u.user\_password as "PASSWORD"

            FROM t\_usuarios as u

            where u.username = \_username;

        end if;

        if \_email is not null then

            SELECT

            u.email as "EMAIL",

            u.user\_password as "PASSWORD"

            FROM t\_usuarios as u

            where u.email = \_email;

        end if;

    elseif \_action = 2 then

        if(\_username is not null) then

            if (exists(select username from t\_usuarios where username = \_username)) then

                select 1 as "EXISTS";

            else

                select 0 as "EXISTS";

            end if;

        elseif(\_email is not null) then

            if (exists(select username from t\_usuarios where email = \_email)) then

                select 1 as "EXISTS";

            else

                select 0 as "EXISTS";

            end if;

        end if;

    elseif \_action = 3 then

        select

        u.email as "EMAIL",

        u.username as "USUARIO",

        concat(u.user\_nombre, u.user\_apPat, u.user\_apMat) as "NOMBRE COMPLETO",

        u.user\_sex "SEXO",

        u.fecha\_nac "FECHA NACIMIENTO",

        m.media\_data "IMAGEN",

        m.media\_type "TIPO DE ARCHIVO"

        FROM t\_usuario as u

        join t\_multimedia as m

        on u.IMAGEN\_ID = m.MEDIA\_ID

        where u.username = \_username;

    end if;

END$$

DELIMITER ;

;

# Script para la creación del diccionario de datos

use db\_cyberkong;

ALTER TABLE t\_usuarios MODIFY COLUMN USER\_ID INT auto\_increment COMMENT 'ID UNICO POR USUARIO';

ALTER TABLE t\_usuarios MODIFY COLUMN email varchar(40) COMMENT 'EMAIL DEL USUARIO CON EL QUE LOGEARA';

ALTER TABLE t\_usuarios MODIFY COLUMN username varchar(30) COMMENT 'USERNAME CON EL QUE SE IDENTIFICA EL USUARIO';

ALTER TABLE t\_usuarios MODIFY COLUMN user\_password varchar(65) COMMENT 'CONTRASEÑA CON LA QUE EL USUARIO LOGEARA';

ALTER TABLE t\_usuarios MODIFY COLUMN user\_nombre varchar(50) COMMENT 'NOMBRE(S) PERSONAL DEL USUARIO';

ALTER TABLE t\_usuarios MODIFY COLUMN user\_apPat varchar(25) COMMENT 'APELLIDO PATERNO DEL USUARIO';

ALTER TABLE t\_usuarios MODIFY COLUMN user\_apMat varchar(25) COMMENT 'APELLIDO MATERNO DEL USUARIO (SI LO TIENE)';

ALTER TABLE t\_usuarios MODIFY COLUMN user\_sex tinyint COMMENT 'SEXO DEL USUARIO';

ALTER TABLE t\_usuarios MODIFY COLUMN fecha\_nac date COMMENT 'FECHA DE NACIMIENTO DEL USUARIO';

ALTER TABLE t\_usuarios MODIFY COLUMN fecha\_in datetime COMMENT 'FECHA DE INGRESO DEL USUARIO';

ALTER TABLE t\_usuarios MODIFY COLUMN imagen\_usuario mediumblob COMMENT 'IMAGEN DEL AVATAR DEL USUARIO';

ALTER TABLE t\_usuarios MODIFY COLUMN image\_type varchar(15) COMMENT 'TIPO DE ARCHIVO QUE SE USARA PARA EL CONTENT/TYPE';

ALTER TABLE t\_usuarios MODIFY COLUMN ROL\_ID int COMMENT 'ROL DE USUARIO, SE DEFINE CON LLAVE FORANEA DE TABLA ROL';

ALTER TABLE t\_usuarios MODIFY COLUMN ESTADO\_ID int COMMENT 'ESTADO DEL USUARIO, SE DEFINE CON LLAVE FORANEA DE TABLA ESTADO';

ALTER TABLE t\_carrito MODIFY COLUMN listado\_carrito JSON COMMENT 'LISTADO DE PRODUCTOS CON DETALLES PARA SU VENTA, EN FORMATO JSON';

ALTER TABLE t\_carrito MODIFY COLUMN USER\_ID INT NOT NULL COMMENT 'LLAVE FORANEA DEL USUARIO DUEÑO DEL CARRITO';

ALTER TABLE t\_carrito MODIFY COLUMN ESTADO\_ID INT COMMENT 'ESTADO EN EL QUE SE ENCUENTRA EL CARRITO';

ALTER TABLE t\_categorias MODIFY COLUMN CATEGORIA\_ID INT auto\_increment COMMENT 'ID UNICO POR CATEGORIA';

ALTER TABLE t\_categorias MODIFY COLUMN cat\_nombre varchar(20) COMMENT 'NOMBRE DE LA CATEGORIA';

ALTER TABLE t\_categorias MODIFY COLUMN cat\_descripcion varchar(250) COMMENT 'DESCRIPCION DE LA CATEGORIA';

ALTER TABLE t\_categorias MODIFY COLUMN USER\_ID int COMMENT 'ID DEL USUARIO QUIEN CREO ESTA CATEGORIA';

ALTER TABLE t\_comentarios MODIFY COLUMN COMENTARIO\_ID INT auto\_increment COMMENT 'ID UNICO DEL COMENTARIO';

ALTER TABLE t\_comentarios MODIFY COLUMN com\_contenido text COMMENT 'CONTENIDO DEL COMENTARIO';

ALTER TABLE t\_comentarios MODIFY COLUMN com\_valoracion int COMMENT 'VALORACION DEL PRODUCTO';

ALTER TABLE t\_comentarios MODIFY COLUMN com\_fecha datetime COMMENT 'FECHA EN QUE SE HIZO EL COMENTARIO';

ALTER TABLE t\_comentarios MODIFY COLUMN PRODUCTO\_ID int COMMENT 'PRODUCTO AL QUE SE ESTA HACIENDO EL COMENTARIO';

ALTER TABLE t\_comentarios MODIFY COLUMN USER\_ID int COMMENT 'USUARIO QUIEN ESTA HACIENDO EL COMENTARIO';

ALTER TABLE t\_lista\_deseados MODIFY COLUMN LISTA\_ID int auto\_increment COMMENT 'ID UNICO DE LA LISTA DE DESEOS DEL USUARIO';

ALTER TABLE t\_lista\_deseados MODIFY COLUMN lista\_descripcion text COMMENT 'DESCRIPCION DE LA LISTA';

ALTER TABLE t\_lista\_deseados MODIFY COLUMN lista\_longitud int COMMENT 'CANTIDAD DE ARTICULOS EN LA LISTA';

ALTER TABLE t\_lista\_deseados MODIFY COLUMN USER\_ID int COMMENT 'ID DEL USUARIO DUEÑO DE LA LISTA';

ALTER TABLE t\_lista\_deseados MODIFY COLUMN ESTADO\_ID int COMMENT 'ESTADO DE LA LISTA (PUBLICA O PRIVADA)';

ALTER TABLE t\_lista\_prod MODIFY COLUMN PRODUCTO\_ID int COMMENT 'ID DEL PRODUCTO';

ALTER TABLE t\_lista\_prod MODIFY COLUMN LISTA\_ID int COMMENT 'ID DE LA LISTA DONDE SE MOSTRARA ESTE PRODUCTO';

ALTER TABLE t\_lista\_prod MODIFY COLUMN fecha\_ag datetime COMMENT 'FECHA EN LA QUE SE AGREGO EL PRODUCTO A LA LISTA';

ALTER TABLE t\_multimedia MODIFY COLUMN MEDIA\_ID int auto\_increment COMMENT 'ID UNICO DEL ELEMENTO MULTIMEDIA';

ALTER TABLE t\_multimedia MODIFY COLUMN media\_data longblob COMMENT 'ELEMENTO MULTIMEDIA';

ALTER TABLE t\_multimedia MODIFY COLUMN media\_type varchar(15) COMMENT 'TIPO DE ELEMENTO MULTIMEDIA';

ALTER TABLE t\_multimedia MODIFY COLUMN fecha\_subida datetime COMMENT 'FECHA DE SUBIDA DEL ELEMENTO MULTIMEDIA';

ALTER TABLE t\_multimedia MODIFY COLUMN PRODUCTO\_ID int COMMENT 'ID DEL PRODUCTO AL CUAL PERTENECE ESTE ELEMENTO MULTIMEDIA';

ALTER TABLE t\_productos MODIFY COLUMN PRODUCTO\_ID int auto\_increment COMMENT 'ID UNICO POR PRODUCTO';

ALTER TABLE t\_productos MODIFY COLUMN prod\_nombre varchar(30) COMMENT 'NOMBRE DEL PRODUCTO';

ALTER TABLE t\_productos MODIFY COLUMN prod\_description text COMMENT 'DESCRIPCION DEL PRODUCTO';

ALTER TABLE t\_productos MODIFY COLUMN precio decimal(8,2) COMMENT 'PRECIO DEL PRODUCTO';

ALTER TABLE t\_productos MODIFY COLUMN prod\_cantidad int COMMENT 'CANTIDAD DISPONIBLE DE ESTE PRODUCTO';

ALTER TABLE t\_productos MODIFY COLUMN prod\_valoracion tinyint COMMENT 'VALORACION DEL PRODUCTO (PROMEDIO DE LAS VALORACIONES DE LOS COMENTARIOS DE ESTE PRODUCTO)';

ALTER TABLE t\_productos MODIFY COLUMN ESTADO\_ID int COMMENT 'ESTADO EN QUE SE ENCUENTRA ESTE PRODUCTO';

ALTER TABLE t\_productos MODIFY COLUMN USER\_ID int COMMENT 'ID DE USUARIO DUEÑO DE ESTE PRODUCTO (VENDEDOR)';

ALTER TABLE t\_cat\_prod MODIFY COLUMN CATEGORIA\_ID int COMMENT 'ID DE CATEGORIA QUE REPRESENTA LA CATEGORIA ASOCIADA';

ALTER TABLE t\_cat\_prod MODIFY COLUMN PRODUCTO\_ID int COMMENT 'ID DE PRODUCTO QUE REPRESENTA EL PRODUCTO AL QUE SE LE VA ASOCIAR LA CATEGORIA';

ALTER TABLE t\_static\_rol MODIFY COLUMN ROL\_ID int COMMENT 'ID UNICO DEL ROL';

ALTER TABLE t\_static\_rol MODIFY COLUMN user\_rol varchar(10) COMMENT 'NOMBRE/DESCRIPCION QUE LLEVARA EL ROL (EJEMP. "ADMIN")';

ALTER TABLE t\_static\_state MODIFY COLUMN ESTADO\_ID int COMMENT 'ID UNICO DEL ESTADO';

ALTER TABLE t\_static\_state MODIFY COLUMN descripcion varchar(30) COMMENT 'DESCRIPCION DEL ESTADO (EJEM. "DISPONIBLE")';

ALTER TABLE t\_token MODIFY COLUMN TOKEN\_ID int auto\_increment COMMENT 'ID UNICO DEL TOKEN';

ALTER TABLE t\_token MODIFY COLUMN token varchar(45) COMMENT 'TOKEN DE AUTENTIFICACION';

ALTER TABLE t\_token MODIFY COLUMN fecha\_registro datetime COMMENT 'FECHA EN QUE SE CREO EL TOKEN';

ALTER TABLE t\_token MODIFY COLUMN USER\_ID int COMMENT 'ID DEL USUARIO AL QUE PERTENECE EL TOKEN';

ALTER TABLE t\_token MODIFY COLUMN ESTADO\_ID int COMMENT 'ESTADO DEL TOKEN';

ALTER TABLE t\_venta MODIFY COLUMN VENTA\_ID int auto\_increment COMMENT 'ID UNICO DE VENTA';

ALTER TABLE t\_venta MODIFY COLUMN fecha\_venta date COMMENT 'FECHA EN QUE SE CREO ESTA VENTA';

ALTER TABLE t\_venta MODIFY COLUMN USER\_ID int COMMENT 'USUARIO PARA OBTENER EL CARRITO AL QUE PERTENECE ESTA VENTA';

ALTER TABLE t\_venta\_detalle MODIFY COLUMN DETALLE\_VENTA\_ID int COMMENT 'ID UNICO DE DETALLE VENTA';

ALTER TABLE t\_venta\_detalle MODIFY COLUMN cantidad int COMMENT 'CANTIDAD DE PRODUCTOS';

ALTER TABLE t\_venta\_detalle MODIFY COLUMN precio int COMMENT 'PRECIO UNITARIO DEL PRODUCTO';

ALTER TABLE t\_venta\_detalle MODIFY COLUMN importe int COMMENT 'PRECIO TOTAL POR TODOS LOS PRODUCTOS';

ALTER TABLE t\_venta\_detalle MODIFY COLUMN VENTA\_ID int COMMENT 'VENTA A LA QUE PERTENECE';

ALTER TABLE t\_venta\_detalle MODIFY COLUMN PRODUCTO\_ID int COMMENT 'ID DEL PRODUCTO';

SELECT distinct

t.table\_schema AS db\_name,

t.table\_name,

(case when t.table\_type = 'BASE TABLE' then 'table'

when t.table\_type = 'VIEW' then 'view'

else t.table\_type

end) as table\_type,

c.column\_name,

c.column\_type,

c.column\_default,

c.column\_key,

c.is\_nullable,

c.extra,

c.column\_comment

from information\_schema.tables as t

inner join information\_schema.columns as c

on t.table\_name = c.table\_name

and t.table\_schema = c.table\_schema

where t.table\_type in ('BASE TABLE')

and t.table\_schema = 'db\_cyberkong'

order by

t.table\_schema,

t.table\_name,

c.ordinal\_position;

# Diccionario de datos obtenido (TABLA DE EXCEL EN WORD)

