



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS



Universidad Autónoma de Nuevo León

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

UA:

BASE DE DATOS MULTIMEDIA

GPO 001

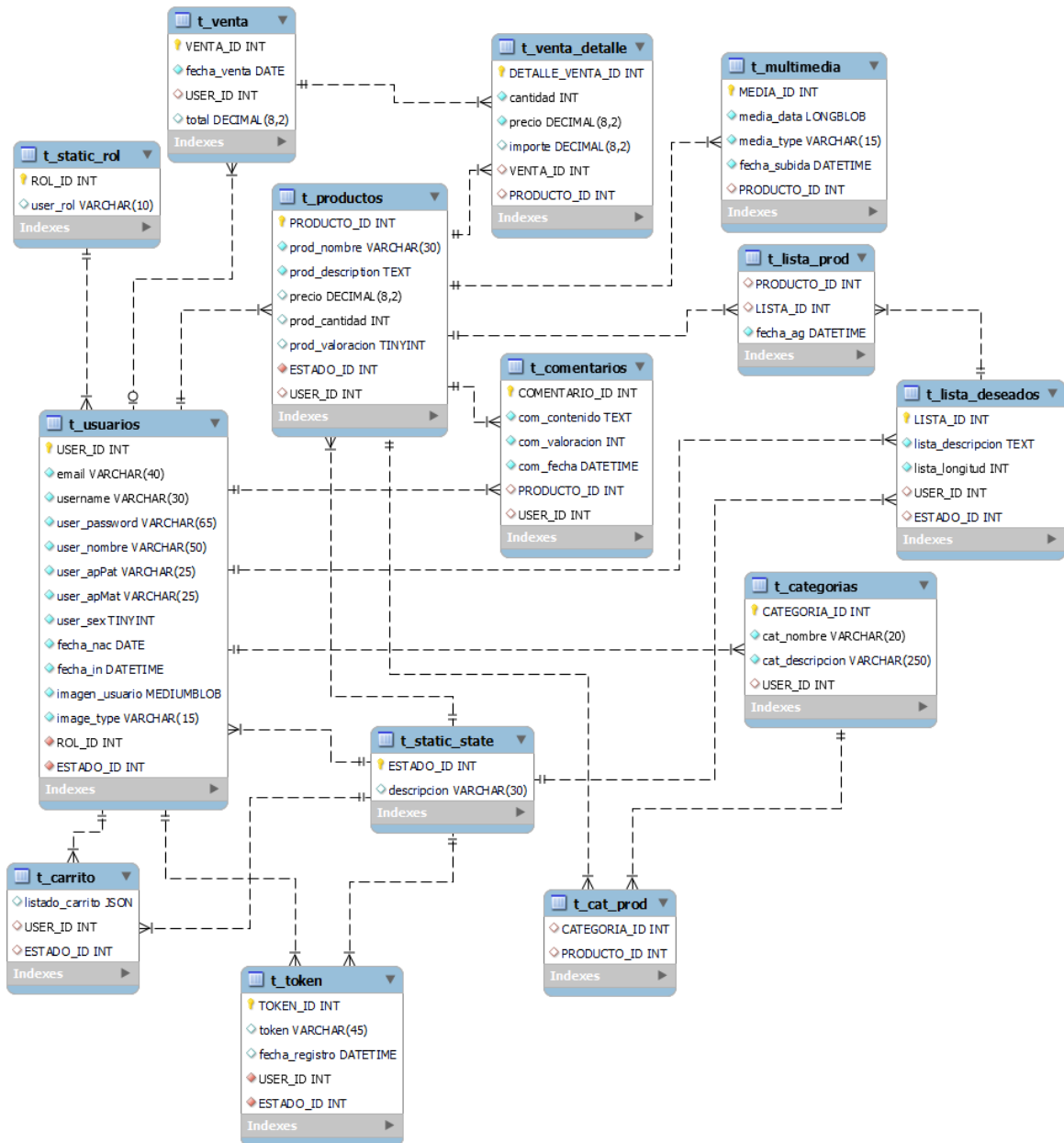
Raul Eleazar Hernández Campos 1804813

Jair Azael Lozano Fraire 1823161

Índice de Contenido

Diagrama MySQL.....	3
Creación de tablas	4
SP Para la página de usuarios.....	10
Script para la creación del diccionario de datos	13
Diccionario de datos obtenido (TABLA DE EXCEL EN WORD)	17

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)

USER_ID	int	MUL	YES		ID DEL USUAR
COMENTARIO	int	PRI	NO	auto_increme	ID UNICO DEL
com_contenid	text		YES		CONTENIDO D
com_valoraci	int		YES		VALORACION
com_fecha	datetime		YES		FECHA EN QUI
PRODUCTO_Id	int	MUL	YES		PRODUCTO AL
USER_ID	int	MUL	YES		USUARIO QUI
LISTA_ID	int	PRI	NO	auto_increme	ID UNICO DE L
lista_descripci	text		YES		DESCRIPCION
lista_longitud	int		YES		CANTIDAD DE
USER_ID	int	MUL	YES		ID DEL USUAR
ESTADO_ID	int	MUL	YES		ESTADO DE LA
PRODUCTO_Id	int	MUL	YES		ID DEL PRODU
LISTA_ID	int	MUL	YES		ID DE LA LISTA
fecha_ag	datetime		YES		FECHA EN LA (
MEDIA_ID	int	PRI	NO	auto_increme	ID UNICO DEL
media_data	longblob		YES		ELEMENTO M
media_type	varchar(15)		YES		TIPO DE ELEM
fecha_subida	datetime		YES		FECHA DE SUE
PRODUCTO_Id	int	MUL	YES		ID DEL PRODU
PRODUCTO_Id	int	PRI	NO	auto_increme	ID UNICO POR