**Documentación Proyecto Final**

**Luis Fernando Aguilar - Carnet 3190-19-22190**

**Grupo #2**

**CREACIÓN DE TABLAS:**

Table TBL\_CATEGORIA:

Esta tabla tiene como finalidad mostrar las categorías generales para segmentación óptima de los productos.

**Script de creación de tabla Categoría:**

/\*==============================================================\*/

/\* Table: TBL\_CATEGORIA\*/

/\*==============================================================\*/

create table TBL\_CATEGORIA (

ID\_CATEGORIA NUMBER not null,

CATEGORIA VARCHAR2(100) not null,

constraint PK\_TBL\_CATEGORIA primary key (ID\_CATEGORIA)

);

La tabla cuenta con 2 atributos los cuales son *ID\_CATEGORIA* el cuál actúa como llave primaria generado a partir de una secuencia, así como también el atributo *CATEGORIA* el cual contiene la descripción de la categoría.

**Inserts en tabla categoría:**

INSERT INTO TBL\_CATEGORIA (ID\_CATEGORIA,CATEGORIA) VALUES (1,'Aseo de hogar');

INSERT INTO TBL\_CATEGORIA (ID\_CATEGORIA,CATEGORIA) VALUES (2,'Bebidas');

INSERT INTO TBL\_CATEGORIA (ID\_CATEGORIA,CATEGORIA) VALUES (3,'Belleza');

INSERT INTO TBL\_CATEGORIA (ID\_CATEGORIA,CATEGORIA) VALUES (4,'Embutidos');

Table TBL\_GASOLINERA:

La creación de esta tabla se lleva a cabo para mostrar y tener un registro de las gasolineras en las cuales pueda existir la instalación de nuestra tienda de conveniencia.

Dentro de los atributos podemos destacar *ID\_UBICACION* el cual actúa como llave foránea, de la tabla *TBL\_UBICACION* para acceder a la ubicación que tendrá dicha gasolinera.

El atributo *ID\_GASOLINERA* es la llave primaria generada a partir de una secuencia y *GASOLINERA* contiene el nombre de la gasolinera.

**Script de creación de tabla Gasolinera:**

/\*==============================================================\*/

/\* Table: TBL\_GASOLINERA \*/

/\*==============================================================\*/

create table TBL\_GASOLINERA (

ID\_UBICACION NUMBER not null,

ID\_GASOLINERA NUMBER not null,

GASOLINERA VARCHAR2(50) not null,

constraint PK\_TBL\_GASOLINERA primary key (ID\_GASOLINERA)

);

**Inserts en tabla gasolinera:**

INSERT INTO "C##TIENDA"."TBL\_GASOLINERA" (ID\_UBICACION, ID\_GASOLINERA, GASOLINERA) VALUES ('23', '1', 'Capital');

INSERT INTO "C##TIENDA"."TBL\_GASOLINERA" (ID\_UBICACION, ID\_GASOLINERA, GASOLINERA) VALUES ('30', '2', 'Super Mixco');

INSERT INTO "C##TIENDA"."TBL\_GASOLINERA" (ID\_UBICACION, ID\_GASOLINERA, GASOLINERA) VALUES ('37', '3', 'Santa Clara');

INSERT INTO "C##TIENDA"."TBL\_GASOLINERA" (ID\_UBICACION, ID\_GASOLINERA, GASOLINERA) VALUES ('51', '4', 'Km 47 Sumpango');

INSERT INTO "C##TIENDA"."TBL\_GASOLINERA" (ID\_UBICACION, ID\_GASOLINERA, GASOLINERA) VALUES ('84', '5', 'Super Gas Masagua');

Table TBL\_TIENDA:

La tabla tienda esta creada con la finalidad de mostrar nuestras tiendas de conveniencia las cuales se encuentran ubicadas en las distintas gasolineras en todo el país, esto se hace a través de la llave foránea *ID\_GASOLINERA* con la cual accedemos a la tabla *TBL\_GASOLINERA* donde contamos con los valores de nombre de gasolinera y a su vez tenemos la ubicación. Gracias a ello tenemos un registro de tienda con la ubicación de la gasolinera donde se encuentra.

**Script de creación de tabla Tienda:**

/\*==============================================================\*/

/\* Table: TBL\_TIENDA \*/

/\*==============================================================\*/

create table TBL\_TIENDA (

ID\_TIENDA NUMBER not null,

ID\_STATUS INTEGER not null,

ID\_GASOLINERA NUMBER,

TIENDA VARCHAR2(25) not null,

constraint PK\_TBL\_TIENDA primary key (ID\_TIENDA)

);

Dentro de los atributos contamos con *ID\_TIENDA* el cual se obtiene a través de una secuencia, *ID\_STATUS* el cual nos da acceso a la tabla *TBL\_STATUS* donde se cuenta con un catálogo de posibles estados para cada tabla, y el atributo *TIENDA* donde guardamos el nombre de nuestra tienda de conveniencia.

**Inserts en tabla tienda:**

INSERT INTO "C##TIENDA"."TBL\_TIENDA" (ID\_TIENDA, ID\_STATUS, ID\_GASOLINERA, TIENDA) VALUES ('1', '1', '1', 'Select Capital');

INSERT INTO "C##TIENDA"."TBL\_TIENDA" (ID\_TIENDA, ID\_STATUS, ID\_GASOLINERA, TIENDA) VALUES ('2', '1', '1', 'Aceitera Capital');

INSERT INTO "C##TIENDA"."TBL\_TIENDA" (ID\_TIENDA, ID\_STATUS, ID\_GASOLINERA, TIENDA) VALUES ('3', '1', '2', 'Super 24 Mixco');

INSERT INTO "C##TIENDA"."TBL\_TIENDA" (ID\_TIENDA, ID\_STATUS, ID\_GASOLINERA, TIENDA) VALUES ('4', '1', '5', 'Select Masagua');

Table TBL\_TIENDA\_PRODUCTO:

Esta tabla tiene como finalidad ser una tabla “pivote” la cual crea una relación directa entre *TBL\_TIENDA* y *TBL\_PRODUCTO.*

Dicha tabla se maneja únicamente a nivel base de datos ya que la forma de insert de datos se hace a través de un Trigger el cuál se dispara al momento de crear un nuevo registro de tienda o producto con una existencia en 0. Esta estructura es necesaria para ingresar stock para la venta.

**Script de creación de tabla Tienda Producto:**

/\*==============================================================\*/

/\* Table: TBL\_TIENDA\_PRODUCTO \*/

/\*==============================================================\*/

create table TBL\_TIENDA\_PRODUCTO (

ID\_TIENDA NUMBER not null,

ID\_PRODUCTO NUMBER not null,

EXISTENCIA NUMBER not null,

constraint PK\_TBL\_TIENDA\_PRODUCTO primary key (ID\_PRODUCTO, ID\_TIENDA)

);

**Atributos:**

Llave primaria compuesta por *ID\_TIENDA* de la tabla *TBL\_TIENDA* y por *ID\_PRODUCTO* de la tabla *TBL\_PRODUCTO*.

*EXISTENCIA* registra el total en stock de cada producto para cada tienda.

**SECUENCIAS Y TRIGGERS**

---SECUENCIAS---

--TBL\_GASOLINERA--

CREATE SEQUENCE SEC\_TBL\_GASOLINERA

MINVALUE 1

MAXVALUE 99999999999999999999999999

START WITH 1

INCREMENT BY 1

CACHE 20;

--TIENDA--

CREATE SEQUENCE SEC\_TBL\_TIENDA

MINVALUE 1

MAXVALUE 99999999999999999999999999

START WITH 1

INCREMENT BY 1

CACHE 20;

--CATEGORIA--

CREATE SEQUENCE SEC\_TBL\_CATEGORIA

MINVALUE 1

MAXVALUE 99999999999999999999999999

START WITH 1

INCREMENT BY 1

CACHE 20;

--TRIGGERS--

--TBL\_GASOLINERA

create or replace TRIGGER TGR\_GASOLINERA

BEFORE INSERT ON TBL\_GASOLINERA REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW

BEGIN

SELECT SEC\_TBL\_GASOLINERA.NEXTVAL INTO :New.ID\_GASOLINERA FROM DUAL;

END;

--TIENDA

create or replace TRIGGER TGR\_TIENDA

BEFORE INSERT ON TBL\_TIENDA REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW

BEGIN

SELECT SEC\_TBL\_TIENDA.NEXTVAL INTO :New.ID\_TIENDA FROM DUAL;

END;

--T\_CATEGORIA

create or replace TRIGGER TGR\_CATEGORIA

BEFORE INSERT ON TBL\_CATEGORIA REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW

BEGIN

SELECT SEC\_TBL\_CATEGORIA.NEXTVAL INTO :New.ID\_CATEGORIA FROM DUAL;

END;

**VISTAS**

VIEW\_TIENDA:

Vista que proporciona la estructura para consulta personalizada de tiendas:

create or replace view VIEW\_TIENDA as

select

t.id\_tienda,

t.tienda,

u.ubicacion,

g.gasolinera,

s.status

from

tbl\_tienda t,

tbl\_gasolinera g,

tbl\_status s,

tbl\_ubicacion u

where

t.id\_gasolinera = g.id\_gasolinera

and t.id\_status = s.id\_status

and g.id\_ubicacion = u.id\_ubicacion;

**SPRING**

Para la creación del CRUD desde Spring, creamos los diferentes servicios para cada tabla según corresponda.

**Servicios tbl\_categoria:**

*Método Get*

Texto

Descripción generada automáticamente

Con este método listamos los registros de la tabla, haciendo una consulta general devolviendo los resultados en un JSON.

*Método Post*

Texto

Descripción generada automáticamente

Con este método ingresamos nuevos registros a la tabla, así como también podemos actualizar datos de registros ya creados.

*Método Delete*

Texto

Descripción generada automáticamente

Dicho método elimina un registro de la tabla, para ello recibe como parámetro el identificador.

**Servicios tbl\_gasolinera:**

*Método Get*

Texto

Descripción generada automáticamente

Con este método listamos los registros de la tabla, haciendo una consulta general devolviendo los resultados en un JSON.

*Método Post*

Texto

Descripción generada automáticamente

Con este método ingresamos nuevos registros a la tabla, así como también podemos actualizar datos de registros ya creados.

*Método Delete*

Texto

Descripción generada automáticamente

Dicho método elimina un registro de la tabla, para ello recibe como parámetro el identificador.

**Servicios tbl\_tienda:**

Texto

Descripción generada automáticamente*Método Get*

Con este método listamos los registros de la tabla, haciendo una consulta general devolviendo los resultados en un JSON.

Texto

Descripción generada automáticamente*Método Post*

Con este método ingresamos nuevos registros a la tabla, así como también podemos actualizar datos de registros ya creados.

*Método Delete*

En el caso de la tabla tienda el método *delete* es utilizado más de forma de eliminación lógica, cambiando de status el registro de activo a inactivo.

**Servicios view\_tienda:**

*Método Get*

Texto

Descripción generada automáticamente

Con este método listamos los registros de la vista, haciendo una consulta general devolviendo los resultados en un JSON.