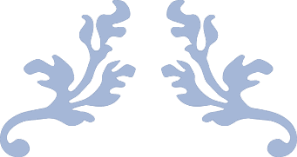


PROYECTO FINAL



FINAL

FINAL



**PRESENTADO POR:**

1. Denilson Urías López Alfaro 148620
2. Miguel Abraham Herrera Crespín 209420
3. Sergio ADONAY LOPEZ MEJIA 186720
4. Jonathan Francisco Mortaya Cazun 188120

**GRUPO DE LABORATORIO:**

**SIS11A**

**DOCENTE ENCARGADO:**

**Tec: Raisa Fabiola Ramires Reyes.**

**FECHA DE PRESENTACIÓN**

01/06/2021

# Index

[Index 2](#_Toc73312764)

[INTRODUCUCCION. 3](#_Toc73312765)

[Descripción del proyecto 4](#_Toc73312766)

[OBJETIVOS 5](#_Toc73312767)

[Diseño de base de datos: Diagrama del modelo 6](#_Toc73312768)

[SCRIP DE LA BASE DE DATOS 7](#_Toc73312769)

[INSERCION DE DATOS 10](#_Toc73312770)

[PROCEDIMIENTOS ALMACENADOS (PARTE l) 11](#_Toc73312771)

[VISTAS 13](#_Toc73312772)

[DICCIONARIO DE DATOS. 21](#_Toc73312773)

[Guía básica del usuario. 31](#_Toc73312774)

[Bibliografía 45](#_Toc73312775)

[(Galdame, 2021) 45](#_Toc73312776)

# INTRODUCUCCION.

El termino e-commerce es una palabra que se ha sobremencionado últimamente en las bocas de las grandes, medianas y pequeñas empresas en los últimos años, pues tienen un negocio local con un potencial para llegar más allá de las fronteras y a cualquier rincón del país. El minisúper GoFood no es la excepción puesta se ha creado un sistema de aplicación Web que permitirá a los clientes a comprar desde la comodidad de su casa. El hecho de ser aplicación Web facilita enormemente el acceso a los clientes y por esta razón el minisúper tendrá muchos más ingresos que tener una aplicación de escritorio donde se tiene la necesidad de instalarse. El sistema contara con un sistema de carretilla donde se guardarán todos los productos que el usuario agregue, con un sistema de categorías donde el usuario podrá encontrar lo que busque lo más rápido posible, todo esto será posible mientras el cliente se logeé para poder efectuar la compra.

La aplicación contará con un sistema de Login con tres niveles de acceso (usuario, Empleado, Administrador), donde cada uno tendrá funciones y objetivos diferentes para el uso de la plataforma. Toda la información personal se guardará en una base de datos segura y confiable. El mantenimiento de la aplicación se lleva acabo por el administrador, este se encargará de gestionar los productos en las categorías, así como agregar secciones y ofertas. Los pagos del cliente son completamente seguros, no habrá preocupación alguna.

En cuanto a los pedidos, contaran con “estados” donde “Enviando” significara que el pedido solo se ha registrado en el sistema, mientras que “Entregado” significara que el pedido ha sido entregado satisfactoriamente a su destino. Todos estos pedidos serán enviados junto con su respectivo Ticket especificando el monto de la compra junto con el impuesto IVA del 13%.

La aplicación tendrá una serie de reportes donde se listarán: pedidos filtrados por día, por estado y un informe con los detalles de cada pedido filtrado por código,

Todo esto será completamente responsive donde no habrá limitaciones para el uso de esta y así poder llegar a todas las personas.

# Descripción del proyecto

Un emprendedor posee un minisúper llamado GoFood que sirve productos a domicilio y requiere de un sistema para llevar el control de los pedidos que le realizan sus clientes, por lo cual, solicita una aplicación web que cumpla con lo requerido.

# OBJETIVOS

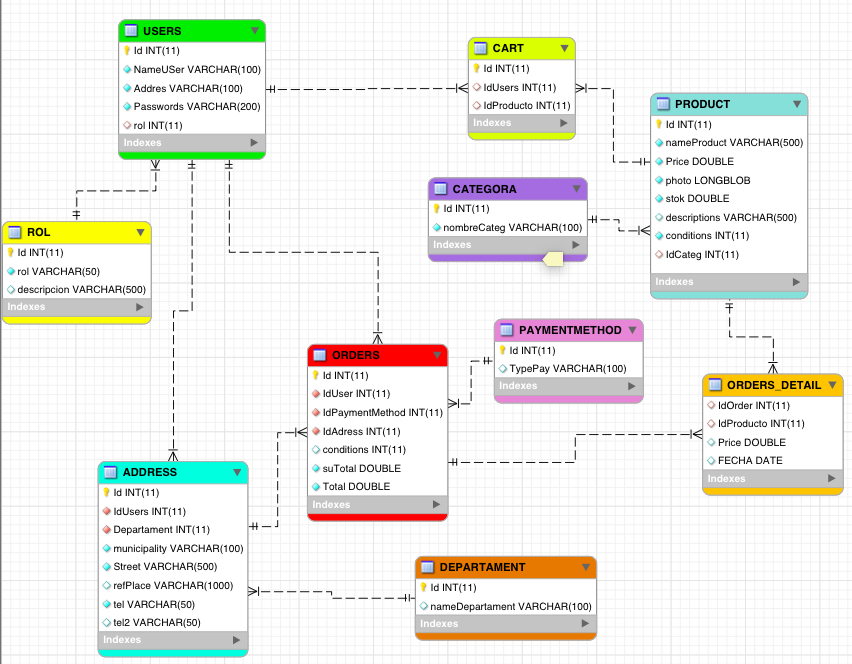
GENERAL :

* Crear una aplicación web para gestionar el servicio de envio de productos a domicilio para el minisuper GoFood.

ESPECIFICOS:

* + Crear una base de datos óptima para el funcionamiento de la aplicación.
  + Crear una pantalla de logo que cuente con validación de acceso.
  + Crear pantalla index con sus header, body y footer.
  + Recolectar los requerimientos sobre el manejo de distribución de pedidos a domicilio.
  + Diseñar interfaces amigables para asegurar una buena experiencia de usuario.
  + Generación de informes parametrizados en TCPPDF
  + Generación de gráficos.

# Diseño de base de datos: Diagrama del modelo



# SCRIP DE LA BASE DE DATOS

CREATE DATABASE GoFooD;

use GoFooD;

CREATE TABLE DEPARTAMENT(

Id int primary key auto\_increment,

nameDepartament varchar(100)

);

CREATE TABLE PAYMENTMETHOD(

Id int primary key auto\_increment,

TypePay varchar(100)

);

CREATE TABLE ROL(

Id int primary key auto\_increment,

rol varchar(50) not null,

descripcion varchar(500)

);

CREATE TABLE USERS(

Id int primary key auto\_increment,

NameUSer varchar(100) not null,

Addres varchar(100) not null,

Passwords varchar(200) not null,

rol int,

FOREIGN KEY (rol) REFERENCES ROL(Id)

);

CREATE TABLE CATEGORA(

Id int primary key auto\_increment,

nombreCateg varchar(100) not null

);

CREATE TABLE PRODUCT(

Id int primary key auto\_increment,

nameProduct varchar(500) not null,

Price double not null,

photo longblob not null,

stok double not null,

descriptions varchar(500),

conditions int not null, */\*EL CAMPO ES BINARIO (0 SI ESTA EN OFERTA Y 1 SI NO LO ESTA)\*/*

IdCateg int,

foreign key (IdCateg) references CATEGORA(Id)

);

CREATE TABLE ADDRESS(

Id int primary key auto\_increment,

IdUsers int not null,

Departament int not null,

municipality varchar(100) not null,

Street varchar(500) not null,

refPlace varchar(1000),

tel varchar(50) not null,

tel2 varchar(50),

FOREIGN KEY (IdUsers) REFERENCES USERS(Id),

FOREIGN KEY (Departament) REFERENCES DEPARTAMENT(Id)

);

CREATE TABLE CART(

Id int primary key auto\_increment,

IdUsers INT,

IdProducto INT,

foreign key(IdUsers) references USERS(Id),

foreign key(IdProducto) references PRODUCT(Id)

);

CREATE TABLE ORDERS(

Id int primary key auto\_increment,

IdUser int NOT NULL,

IdPaymentMethod int not null,

IdAdress int not null,

conditions int, */\*campo binario condicion si esta en proceso de entrega o no\*/*

suTotal double not null,

Total double not null,

fecha DATE,

eliminadoLogic int,

foreign key(IdUser) references USERS(Id),

foreign key(IdPaymentMethod) references PAYMENTMETHOD(Id),

foreign key(IdAdress) references ADDRESS(Id)

);

CREATE TABLE ORDERS\_DETAIL(

IdOrder INT,

IdProducto INT,

Price DOUBLE,

FECHA date,

foreign key(IdOrder) references ORDERS(Id),

foreign key(IdProducto) references PRODUCT(Id)

);

## INSERCION DE DATOS

INSERT INTO ROL (rol,descripcion) VALUES (*'USUARIO'*,*'Tiene derecho solo a poder comprar en este sitio y ver sus compras crud de su carrito de compras'*);

INSERT INTO ROL (rol,descripcion) VALUES (*'EMPLEADO'*,*'No definido'*);

INSERT INTO ROL (rol,descripcion) VALUES (*'ADMINISTRADOR'*,*'Tiene derecho a casi todas las opciones de esta pagina web'*);

INSERT INTO USERS (NAMEUSER,ADDRES,PASSWORDS,ROL) VALUES (*'EMPLEADO 1'*,*'EMPLEADO1@GMAIL.COM'*,*'EMPLEADO'*,2);

INSERT INTO USERS (NAMEUSER,ADDRES,PASSWORDS,ROL) VALUES (*'EMPLEADO 2'*,*'EMPLEADO2@GMAIL.COM'*,*'EMPLEADO2'*,2);

INSERT INTO USERS (NAMEUSER,ADDRES,PASSWORDS,ROL) VALUES (*'ADMINISTRADOR'*,*'SOYADMIN@GMAIL.COM'*,*'ADMIN'*,3);

INSERT INTO PAYMENTMETHOD (TypePay) VALUES (*'CREDITO'*);

INSERT INTO PAYMENTMETHOD (TypePay) VALUES (*'CONTADO'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'AHUCHAPAN'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'CABAÑAS'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'CHALATENANGO'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'CUCATLAN'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'LA LIVERTAD'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'LA PAZ'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'LA UNION'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'MORAZAN'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'SAN MIGUEL'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'SAN SALVADOR'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'SAN VICENTE'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'SANTA ANA'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'SONSONATE'*);

INSERT INTO DEPARTAMENT (nameDepartament) VALUES (*'USULUTAN'*);

INSERT INTO CATEGORA (nombreCateg) VALUES (*'HIGIENE'*);

INSERT INTO CATEGORA (nombreCateg) VALUES (*'FRUTAS Y VERDURAS'*);

INSERT INTO CATEGORA (nombreCateg) VALUES (*'CARNE Y EMBUTIDOS'*);

INSERT INTO CATEGORA (nombreCateg) VALUES (*'GRANOS BASICOS'*);

INSERT INTO CATEGORA (nombreCateg) VALUES (*'BEBIDAS'*);

INSERT INTO CATEGORA (nombreCateg) VALUES (*'OTROS'*);

## PROCEDIMIENTOS ALMACENADOS

DELIMITER ;

USE *`gofood`*;

DROP procedure IF EXISTS *`InsertUSers`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`InsertUSers`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`InsertUSers`*(

in nombre varchar(100),

in correo varchar(100),

in contra varchar(200) ,

in rol int

)

BEGIN

INSERT INTO USERS (NameUser,Addres,Passwords,rol) VALUES (nombre,correo,contra,rol);

END$$

DELIMITER ;

;

*/\*---------------------------------\*/*

DELIMITER ;

USE *`gofood`*;

DROP procedure IF EXISTS *`insertDireccion`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`insertDireccion`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`insertDireccion`*(

in usuario int,

in Depa int,

in municipio varchar(200),

in calle varchar(200),

in ref varchar(200),

in tel varchar(50),

in te2 varchar(50)

)

BEGIN

INSERT INTO address(IdUsers,Departament,municipality,Street,refPlace,tel,tel2)

VALUES (usuario,Depa,municipio,calle,ref,tel,te2);

END$$

DELIMITER ;

;

*/\*-----------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`sp\_insertProduct`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`sp\_insertProduct`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`sp\_insertProduct`*(

in nombre varchar(100),

in precio double,

in photo longblob,

in stok double,

in descripcion varchar(500),

in condicion int,

in categoria int

)

BEGIN

INSERT INTO PRODUCT (nameProduct,price,photo,stok,descriptions,conditions,IdCateg) VALUES (nombre,precio,photo,stok,descripcion,condicion,categoria);

END$$

DELIMITER ;

;

*/\*---------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`sp\_SelecIdUsers`*;

DELIMITER $$

USE *`gofood`*$$

CREATE PROCEDURE *`sp\_SelecIdUsers`*(

in Usuario varchar(100),

in contra varchar(100)

)

BEGIN

select Id FROM USERS WHERE NameUser = Usuario AND Passwords = contra;

END$$

*/\*------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_USERS`*;

DELIMITER $$

USE *`gofood`*$$

CREATE PROCEDURE *`SP\_USERS`* (

in nombres varchar(100),

in correo varchar(100),

in contra varchar(100)

)

BEGIN

INSERT INTO USERS (NAMEUSER,ADDRES,PASSWORDS,ROL) VALUES (nombres,correo,contra,1);

END$$

*/\*------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_ADDRESS`*;

DELIMITER $$

USE *`gofood`*$$

CREATE PROCEDURE *`SP\_ADDRESS`* (

in nombres varchar(100),

in correo varchar(100),

in contra varchar(100)

)

BEGIN

INSERT INTO USERS (NAMEUSER,ADDRES,PASSWORDS,ROL) VALUES (nombres,correo,contra,1);

END$$

*/\*------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`sp\_showcart`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`sp\_showcart`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`sp\_showcart`*(

in usuario int

)

BEGIN

select c.Id,

p.photo,

p.nameProduct,

ca.nombreCateg,

p.price,

p.descriptions,

p.conditions

from cart c

inner join product p

on p.id = c.idproducto

inner join categora ca

on ca.id = p.Idcateg

where c.idUsers = usuario;

END$$

DELIMITER ;

;

*/\*---------------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`updateProduct`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`updateProduct`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`updateProduct`*(

in codigo int,

in nombre varchar(500),

in precio double,

in foto longblob,

in cantidad int,

in descripcion varchar(500),

in condicion int,

in categoria int

)

BEGIN

UPDATE PRODUCT SET nameProduct = nombre,

price = precio,

photo = foto,

stok = cantidad,

descriptions = descripcion,

conditions = condicion,

Idcateg = categoria where Id = codigo;

END$$

DELIMITER ;

;

*/\*---------------------------------------------------------------------------------------------------------------\*/*

*/\*+---------------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

CREATE OR REPLACE VIEW *`view\_product`* AS

select p.id,photo,nameProduct,c.nombreCateg,descriptions,stok,price,if(conditions=1,*'DISPONIBLE'*,*'NO DISPONIBLE'*) AS CONDICION from product p inner join categora c on p.IdCateg = c.Id order by p.Id asc;

*/\*-----------------------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`showFactura`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`showFactura`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`showFactura`*(

in usuario int

)

BEGIN

SELECT D.IDORDER,

P.NAMEPRODUCT,

P.DESCRIPTIONS,

D.PRICE,

max(FECHA)

FROM ORDERS\_DETAIL D

INNER JOIN ORDERS O

ON D.IDORDER = O.ID

INNER JOIN PRODUCT P

ON P.ID = D.IDPRODUCTO WHERE O.IDUSER = 3;

END$$

DELIMITER ;

;

*/\*----------------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_FACTURA\_DETALLE`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_FACTURA\_DETALLE`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_FACTURA\_DETALLE`*(

in NumOrden int

)

BEGIN

SELECT IDORDER,

P.NAMEPRODUCT,

P.DESCRIPTIONS,

P.PRICE,

D.FECHA,

SUTOTAL,

FORMAT(TOTAL,2) AS TOTAL

FROM ORDERS\_DETAIL D

INNER JOIN PRODUCT P

ON P.ID = D.IDPRODUCTO

INNER JOIN ORDERS O

ON O.ID = D.IDORDER

WHERE IDORDER = NumOrden;

END$$

DELIMITER ;

;

*/\*--------------------------------------------------------------------------------------\*/*

*/\*------------------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_CARRITO`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_CARRITO`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_CARRITO`*(

in USUARIO int,

in PRODUCTO int

)

BEGIN

INSERT INTO CART (IDUSERS,IDPRODUCTO) VALUES (USUARIO,PRODUCTO);

SELECT @STOK := stok FROM PRODUCT WHERE ID = PRODUCTO;

UPDATE PRODUCT SET STOK = (@STOK-1) WHERE ID = PRODUCTO;

END$$

DELIMITER ;

;

*/\*------------------------------------------------------------------------------------------------------------------\*/*

*/\*------------------------------------------------------------------------------------------------------------------\*/*

*/\*------------------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_PEDIDOS`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_PEDIDOS`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_PEDIDOS`*(

in usuario int,

in metodPa int

)

BEGIN

DECLARE GASTO DOUBLE;

DECLARE DIRECCION INT;

DECLARE ESTADO INT;

DECLARE BORRAR INT;

DECLARE FECH DATE;

SET FECH = CURDATE();

SET BORRAR = 1;

SET ESTADO = 1;

SET GASTO = (SELECT SUM(P.PRICE) FROM CART C INNER JOIN PRODUCT P ON C.IDPRODUCTO=P.ID WHERE C.IDUSERS=usuario);

SET DIRECCION = (SELECT ID FROM ADDRESS WHERE IDUSERS = USUARIO);

INSERT INTO ORDERS (IDUSER,IDPAYMENTMETHOD,IdAdress,CONDITIONS,SUTOTAL,TOTAL,FECHA,ELIMINADOLOGIC) VALUES (usuario,metodPa,DIRECCION,ESTADO,GASTO,GASTO\*1.13,CURDATE(),1);

END$$

DELIMITER ;

;

*/\*------------------------------------------------------------------------------------------------------------------\*/*

*/\*CALL SP\_PEDIDOS(2,1);\*/*

*/\*SELECT ROUND(TOTAL,2) FROM ORDERS; PARA ESCOGER DOS DECIMALES EN LA CONSULTA\*/*

SELECT \* FROM ORDERS;

*/\*-------------------------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_DETALLE`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_DETALLE`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_DETALLE`*(

IN USUARIO INT

)

BEGIN

SELECT @ID := MAX(ID) FROM ORDERS WHERE IDUSER =USUARIO;

INSERT INTO ORDERS\_DETAIL (IDORDER,IDPRODUCTO,PRICE,FECHA)

SELECT @ID,P.ID,P.PRICE,CURDATE()

FROM CART C

INNER JOIN PRODUCT P

ON P.ID = C.IDPRODUCTO WHERE C.IDUSERS = USUARIO;

END$$

DELIMITER ;

;

CALL SP\_DETALLE(2);

SELECT \* FROM CART;

*/\*---------------------------------------------------------------------------------------------------------------------------------------\*/*

SELECT

O.ID,

P.NAMEPRODUCT,

P.PRICE,

IF (O.CONDITIONS=1,*"ENVIADO"*,*"ENTREGADO"*) AS ESTADO

FROM ORDERS\_DETAIL D

INNER JOIN ORDERS O

ON O.ID = D.IDORDER

INNER JOIN PRODUCT P

ON P.ID = D.IDPRODUCTO

WHERE O.IDUSER = 1 ORDER BY O.ID ASC;

USE *`gofood`*;

CREATE OR REPLACE VIEW *`view\_orders`* AS

SELECT

O.ID,

U.NAMEUSER,

DE.NAMEDEPARTAMENT,

A.MUNICIPALITY,

STREET,

REFPLACE,

TEL,

TEL2,

IF (O.CONDITIONS=1,*"ENVIADO"*,*"ENTREGADO"*) AS ESTADO,

IF (O.IDPAYMENTMETHOD=1,*"CREDITO"*,*"CONTADO"*) AS METODO

FROM ORDERS O

INNER JOIN USERS U

ON U.ID = O.IDUSER

INNER JOIN ADDRESS A

ON A.ID = O.IDADRESS

INNER JOIN DEPARTAMENT DE

ON A.ID = DE.ID;

*/\*-------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_PRODUCTO\_ENTREGADO`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_PRODUCTO\_ENTREGADO`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_PRODUCTO\_ENTREGADO`*(

in ORDEN int

)

BEGIN

UPDATE ORDERS SET CONDITIONS = 2 WHERE ID = ORDEN;

END$$

DELIMITER ;

;

*/\*---------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_FILTRADO\_FECHA`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_FILTRADO\_FECHA`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_FILTRADO\_FECHA`*(

in FECHA VARCHAR(100)

)

BEGIN

select

O.ID,

U.NAMEUSER,

IF (O.CONDITIONS=1,*"ENVIADO"*,*"ENTREGADO"*) AS ESTADO,

SUTOTAL,

FORMAT(TOTAL,2) AS TOTAL,

OD.FECHA

from ORDERS O

INNER JOIN ORDERS\_DETAIL OD

ON O.ID = OD.IDORDER

INNER JOIN USERS U

ON U.ID = O.IDUSER

WHERE OD.FECHA = FECHA GROUP BY O.ID;

END$$

DELIMITER ;

;

*/\*------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_FILTRADO\_ESTADO`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_FILTRADO\_ESTADO`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_FILTRADO\_ESTADO`*(

in ESTADO VARCHAR(100)

)

BEGIN

select

O.ID,

U.NAMEUSER,

IF (O.CONDITIONS=1,*"ENVIADO"*,*"ENTREGADO"*) AS ESTADO,

SUTOTAL,

FORMAT(TOTAL,2) AS TOTAL,

OD.FECHA

from ORDERS O

INNER JOIN ORDERS\_DETAIL OD

ON O.ID = OD.IDORDER

INNER JOIN USERS U

ON U.ID = O.IDUSER

WHERE O.CONDITIONS = ESTADO GROUP BY O.ID;

END$$

DELIMITER ;

;

*/\*--------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_COMPRAS\_CLIENTES`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_COMPRAS\_CLIENTES`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_COMPRAS\_CLIENTES`*(

in usuario int

)

BEGIN

SELECT

O.ID,

U.NAMEUSER,

DE.NAMEDEPARTAMENT,

A.MUNICIPALITY,

STREET,

REFPLACE,

TEL,

TEL2,

IF (O.CONDITIONS=1,*"ENVIADO"*,*"ENTREGADO"*) AS ESTADO,

IF (O.IDPAYMENTMETHOD=1,*"CREDITO"*,*"CONTADO"*) AS METODO

FROM ORDERS O

INNER JOIN USERS U

ON U.ID = O.IDUSER

INNER JOIN ADDRESS A

ON A.ID = O.IDADRESS

INNER JOIN DEPARTAMENT DE

ON A.ID = DE.ID

WHERE O.IDUSER = USUARIO AND ELIMINADOLOGIC = 1;

END$$

DELIMITER ;

;

SELECT \* FROM ORDERS;

*/\*--------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_ELIMINADO\_LOGICO`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_ELIMINADO\_LOGICO`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_ELIMINADO\_LOGICO`*(

in ORDEN int

)

BEGIN

UPDATE ORDERS SET ELIMINADOLOGIC = 2 WHERE ID = ORDEN;

END$$

DELIMITER ;

;

*/\*--------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_SUMA\_CARRITO`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_SUMA\_CARRITO`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_SUMA\_CARRITO`*(

in USUARIO int

)

BEGIN

SELECT SUM(PRICE) AS SUMA FROM CART C INNER JOIN PRODUCT P ON P.ID = C.IDPRODUCTO WHERE IDUSERS = USUARIO;

END$$

DELIMITER ;

;

*/\*-----------------------------------------------------------------------------------------------------------------\*/*

USE *`gofood`*;

DROP procedure IF EXISTS *`SP\_ELIMIAR\_CARRITO`*;

USE *`gofood`*;

DROP procedure IF EXISTS *`gofood`*.*`SP\_ELIMIAR\_CARRITO`*;

;

DELIMITER $$

USE *`gofood`*$$

CREATE DEFINER=*`root`*@*`localhost`* PROCEDURE *`SP\_ELIMIAR\_CARRITO`*(

in FILA int

)

BEGIN

SELECT @PROD := IDPRODUCTO FROM CART WHERE ID = FILA;

SELECT @STOK := stok FROM PRODUCT WHERE ID = @PROD;

UPDATE PRODUCT SET STOK = (@STOK+1) WHERE ID = @PROD;

DELETE FROM CART WHERE ID = FILA;

END$$

DELIMITER ;

;

# **Vistas**

use gofood;

USE *`gofood`*;

CREATE OR REPLACE VIEW *`view\_higiene`* AS

select p.id,photo,nameProduct,c.nombreCateg,descriptions,stok,price from product p inner join categora c on p.IdCateg = c.Id where IdCateg = 1 AND CONDITIONS = 1 AND STOK > 0;

USE *`gofood`*;

CREATE OR REPLACE VIEW *`view\_frutas`* AS

select p.id,photo,nameProduct,c.nombreCateg,descriptions,stok,price from product p inner join categora c on p.IdCateg = c.Id where IdCateg = 2 AND CONDITIONS = 1 AND STOK > 0;

USE *`gofood`*;

CREATE OR REPLACE VIEW *`view\_carne`* AS

select p.id,photo,nameProduct,c.nombreCateg,descriptions,stok,price from product p inner join categora c on p.IdCateg = c.Id where IdCateg = 3 AND CONDITIONS = 1 AND STOK > 0;

USE *`gofood`*;

CREATE OR REPLACE VIEW *`view\_granos`* AS

select p.id,photo,nameProduct,c.nombreCateg,descriptions,stok,price from product p inner join categora c on p.IdCateg = c.Id where IdCateg = 4 AND CONDITIONS = 1 AND STOK > 0;

USE *`gofood`*;

CREATE OR REPLACE VIEW *`view\_bebidas`* AS

select p.id,photo,nameProduct,c.nombreCateg,descriptions,stok,price from product p inner join categora c on p.IdCateg = c.Id where IdCateg = 5 AND CONDITIONS = 1 AND STOK > 0;

USE *`gofood`*;

CREATE OR REPLACE VIEW *`view\_otros`* AS

select p.id,photo,nameProduct,c.nombreCateg,descriptions,stok,price from product p inner join categora c on p.IdCateg = c.Id where IdCateg = 6 AND CONDITIONS = 1 AND STOK > 0;

# DICCIONARIO DE DATOS.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** DEPARTAMENT | | | | | |
| **Descripción:** Tabla donde se almacenara  el departamento donde vive el cliente. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato** | **Descripción:** |
| P | Id | Claves del departamento | Obligatoria, única y auto incrementable. | Numérico | Clave única del departamento y se va incrementando automáticamente  de uno en uno. |
|  | nameDepartament | Nombre del departamento . | Entrada obligatoria | Carácter | Nombre del departamento donde vive el cliente (La libertad, sonsonate). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** PAYMENTMETHOD | | | | | |
| **Descripción:** Tabla donde se almacenara  el tipo de pago que va a realizar el cliente. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato** | **Descripción:** |
| P | Id | Clave del tipo de pago | Obligatoria, única y auto incrementable. | Numérico | Clave única del tipo de pago se va incrementando automáticamente  de uno en uno. |
|  | TypePay | Nombre del tipo de pago. | Entrada obligatoria | Carácter | Nombre del tipo de pago que se va a efectuar por parte del cliente (tarjeta, efectivo) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** ROL | | | | | |
| **Descripción:** Tabla donde se almacenara  el rol del usuario. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato :** | **Descripción:** |
| P | Id | Clave del rol | Obligatoria, única y auto incrementable. | Numérico | Clave única del rol y se va incrementando automáticamente  de uno en uno. |
|  | rol | Nombre del rol. | Entrada obligatoria | Carácter | Nombre del rol que va a ejecutar dentro de la aplicación (USUARIO,EMPLEADO y ADMINISTRADOR). |
|  | descripcion | Describir que va hacer. | Entrada obligatoria | Carácter | Describir cual es su rol en la aplicación y que accesos va a tener. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** USER | | | | | |
| **Descripción:** Tabla donde se almacenara los datos del usuario. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato :** | **Descripción:** |
| P | Id | Clave del usuario | Obligatoria, única y auto incrementable. | Numérico | Clave única del usuario se va incrementando automáticamente  de uno en uno. |
| F | rol | Foreing key de rol. | Entrada obligatoria | Numerico. | id de la tabla rol. |
|  | NameUser | Nombre del usuario. | Entrada obligatoria | Carácter | Nombre del usuario que se va a registrar . |
|  | Addres | Direccion del usuario | Entrada obligatoria. | Carácter. | Una direccion de donde vive el usuario. |
|  | Passwords | Contraseña del usuario. | Entrada obligatoria y unica. | Carácter. | Contraseña unica departe del usuario y pediante ella tendra el acceso a la aplicación. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** CATEGORA | | | | | |
| **Descripción:** Tabla donde se almacenara la cateoria de los productos. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato** | **Descripción:** |
| P | Id | Clave del la cateoria | Obligatoria, única y auto incrementable. | Numérico | Clave única de la cateoria y se va incrementando automáticamente  de uno en uno. |
|  | nombreCateg | Nombre de la cateoria . | Entrada obligatoria | Carácter | Nombre de las cateorias para agrupar los productos segun su cateoria (HIGIENE, FRUTAS Y VERDURAS y BEBIDAS) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** PRODUCT | | | | | |
| **Descripción:** Tabla donde se almacenara los productos. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato** | **Descripción:** |
| P | Id | Clave del producto | Obligatoria, única y auto incrementable. | Numérico | Clave única de los productos y se va incrementando automáticamente  de uno en uno. |
| F | IdCateg | Foreing key de categora. | Entrada obligatoria. | Numérico. | Id de la tabla categora. |
|  | NameProduct | Nombrel producto. | Entrada obligatoria | Carácter | Nombre del producto y según su nombre se va agregar por cateoria (zanahoria, naranja, carne, pollo, etc..). |
|  | Price | Precio del producto | Entrada obligatoria | Numérico. | Precio de cada producto ingrsado. |
|  | photo | Foto del producto. | Entrada obligatoria | Longblob ó tipo foto. | Foto del cada producto. |
|  | stok | Cantidad de producto. | Entrada obligatoria | Numérico. | Se contara la cantidad de productos existentes (papel Higienico = 10.) |
|  | descriptions | Descripcion del producto. | Entrada obligatoria | Carácter. | Breve descriccion del producto (huevos: carton de 12 unidades medianas de la empresa el campo feliz). |
|  | conditions |  | Entrada obligatoria | Binario. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** ADDRESS | | | | | |
| **Descripción:** Tabla donde se almacenara la direccion del cliente. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato** | **Descripción:** |
| P | Id | Clave del producto | Obligatoria, única y auto incrementable. | Numérico | Clave única de la direccion del cliente y se va incrementando automáticamente  de uno en uno. |
| F | IdUsers | Foreing key de user . | Entrada obligatoria. | Numérico. | Id de la tabla categora. |
| F | Departament | Foreing key de departament. | Entrada obligatoria | Numérico | Id de la tabla departament. |
|  | municipality | Municipo. | Entrada obligatoria | Carácter. | Nombre del municipio donde vive el cliente. |
|  | Street | Calle. | Entrada obligatoria | Carácter. | Numero de calle, nombre de la calle donde vive el cliente. |
|  | refPlace | Referencia. | No obligatorio. | Carácter. | Referencia donde vive el cliente  (Casa de 2 plantas, blanca) |
|  | Tel | Telefono. | Entrada obligatoria | Carácter. | Numero de telefono del cliente. |
|  | Tel2 | Telefono2 | No obligatorio. | Carácter. | Un segundo numero de telefono del cliente. |

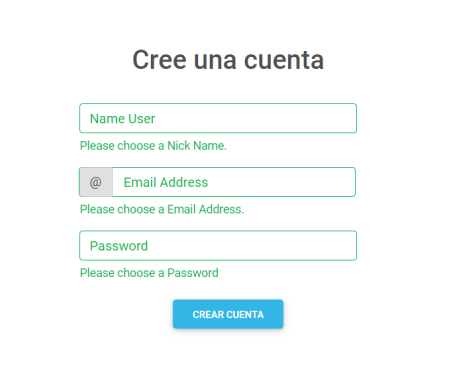
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** CART | | | | | |
| **Descripción:** Tabla que servira como carrito de compras. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato** | **Descripción:** |
| P | Id | Clave del carrito. | Obligatoria, única y auto incrementable. | Numérico | Clave única de cart y se va incrementando automáticamente  de uno en uno. |
| F | IdUsers | Foreing key de users. | Entrada obligatoria | Numérico. | Id de la tabla user. |
| F | IdProducto | Foreing key de product. | Entrada obligatoria | Numérico. | Id de la tabla product. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** ORDERS | | | | | |
| **Descripción:** Tabla donde se almacenara las ordenes de los clientes. | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato** | **Descripción:** |
| P | Id | Clave de la orden | Obligatoria, única y auto incrementable. | Numérico | Clave única de la orden del cliente y se va incrementando automáticamente  de uno en uno. |
| F | IdUsers | Foreing key de user . | Entrada obligatoria. | Numérico. | Id de la tabla categora. |
| F | IdPaymentMethod | Foreing key de paymenthod. | Entrada obligatoria | Numérico | Id de la tabla paymenthod. |
| F | IdAdress | Foreing key de address. | Entrada obligatoria | Numérico. | Id de la tabla address. |
|  | conditions | Condicion. | Entrada obligatoria. | Numérico. | Condicion de la orden si se ha entegado o enviado (1 enviado 2 entregado). |
|  | suTotal | subTotal. | Entrada obligatoria. | Numérico. | Sub total de todas las compras del cliente. |
|  | Total. | Telefono. | Entrada obligatoria | Numérico. | Total final del cliente . |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre del archivo:** ORDERS\_DETAIL | | | | | |
| **Descripción:** Tabla donde se guardara detalle de ordenes . | | | | | |
| **Clave:** | **Campo:** | **Dominio:** | **Validación:** | **Tipo de dato** | **Descripción:** |
| F | IdOrder | Foreing key de orders. | Entrada obligatoria. | Numérico | Id de la tabla orders. |
| F | IdProducto | Foreing key de product. | Entrada obligatoria | Numérico. | Id de la tabla product. |
|  | Price | Precio. | Entrada obligatoria | Numérico. | Precio de la orden generada por el cliente. |
|  | Fecha | Fecha de orden | Entrada obligatoria | Date. | Fecha de cuando se efectuo la orden. |

# Guía básica del usuario.

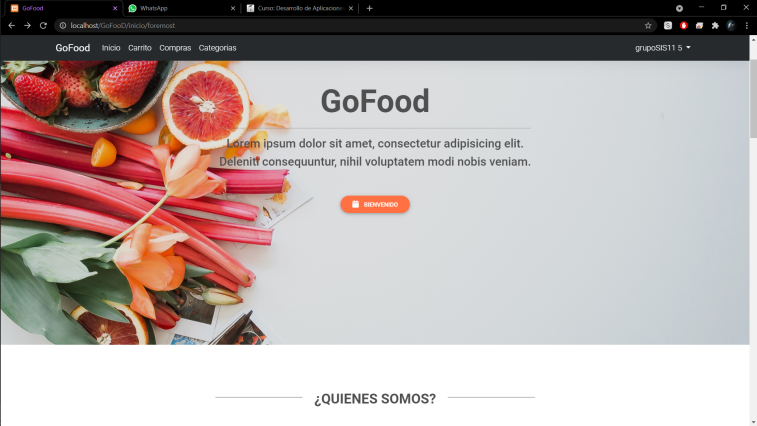
La aplicación web contará con un login con diferentes accesos  donde se requerirá un correo y una contraseña y con ello se te dará un acceso a la aplicación y en la parte inferior muestra un link para registrarse si es que no cuenta con una cuenta.

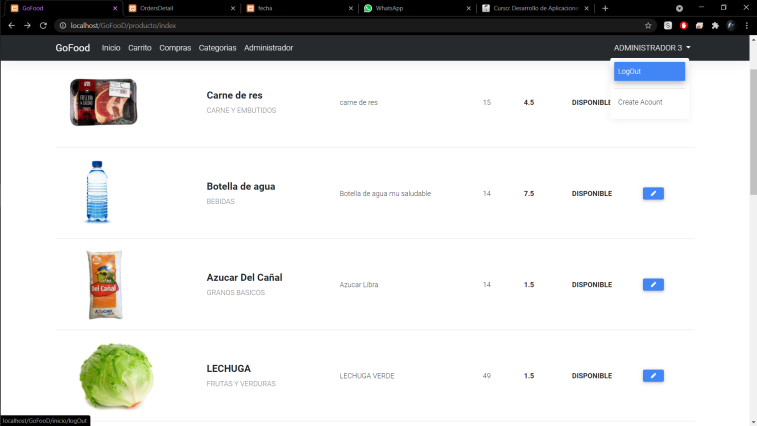


Una vez das click en Registrarte aquí  te mostrará un formulario donde se te pide un nombre de usuario, un correo y una contraseña todos serán requeridos obligatoriamente.



Después de darle al botón (CREAR CUENTA) te mostrará un segundo formulario para terminar de complementar los datos donde se te requerirá datos de dirección y números de teléfono donde el campo de teléfono 2 no es obligatorio una vez enviados estos datos te redirigirá al formulario del login.

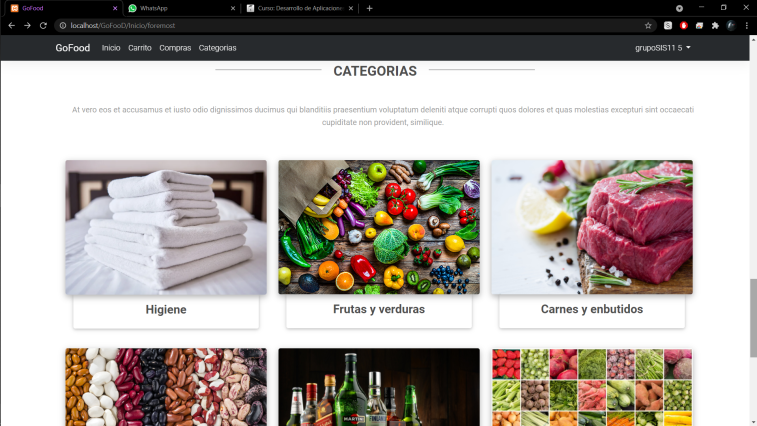
Cuando el rol es  de usuario se te muestra un inicio agradable, en la parte de arriba se muestra un menú donde **Carrito**  es tu carrito de compras, **Compra** muestra tus compras realizadas, **Categorías**  muestra los productos que puedes comprar, en la parte derecha  muestra el nombre del usuario, en la parte de abajo se describe quiénes somos y más abajo se encuentran las categorías de los productos y al final un footer donde está la información como redes sociales y cómo encontrarnos.



A la derecha del menú, a la par del nombre de usuario se despliegan 2 botones (LogOut, Crear cuenta).

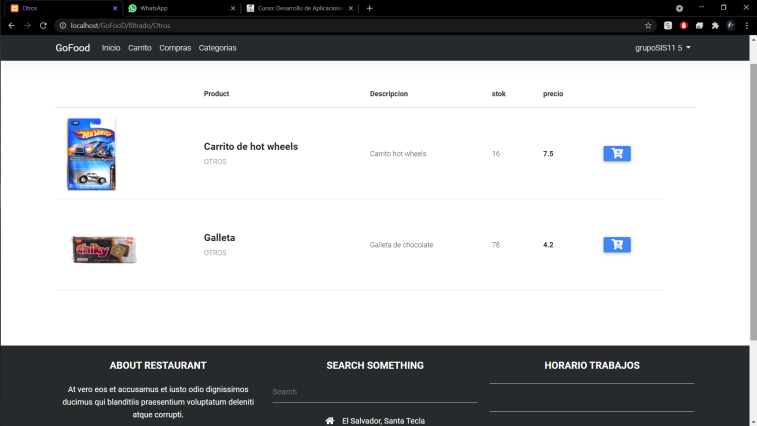
El primero al darle click te cerrará la sesión y te dirigira hacia el login.

El segundo te dirigirá hacia los formularios que están al principio en los cuales se te solicita los datos para crear tu cuenta.



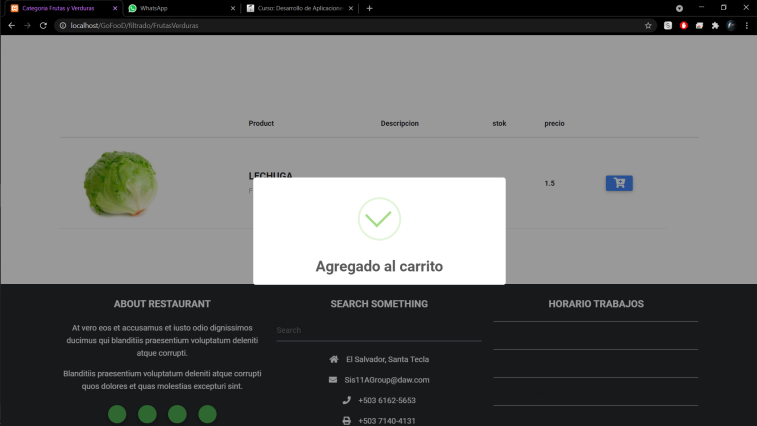


Al darle click al botón (**Categorías**) te dirige a la parte de abajo del inicio donde te muestra diferentes categorías y dentro de ellas los productos que puedes comprar.

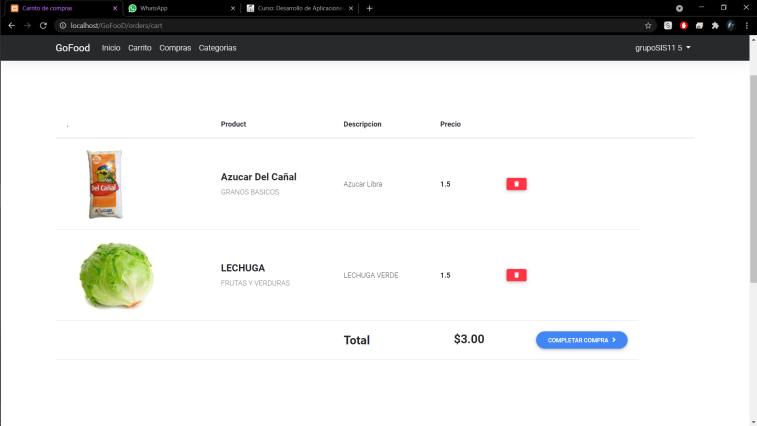


Una ves seleccionado una categoría me muestra una pantalla donde puedes ver una foto del producto, **nombre** el producto y abajo del nombre la categoría a la que pertenece, una breve **descripción** del producto, **Stock** que es la cantidad de ese producto existente, **precio** y un **botón con el cual agregas** ese producto en tu carrito de compras.



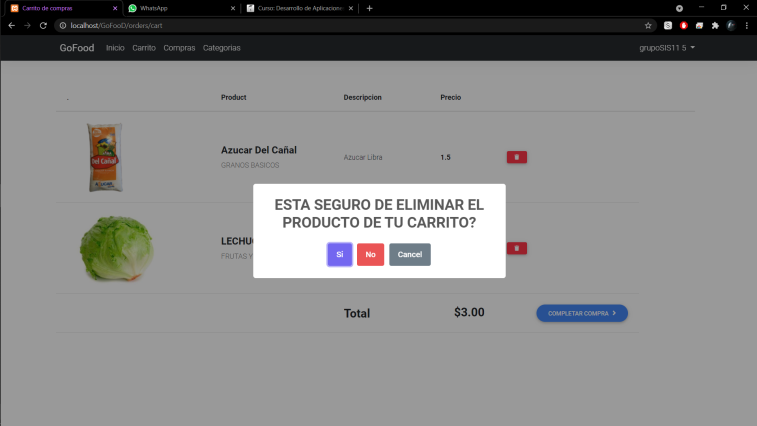


Al dar click en el botón del carrito, mostrar una alerta (**Agregado al carrito**) la cual te confirma que ese producto ha sido agregado a tu carrito de compra.

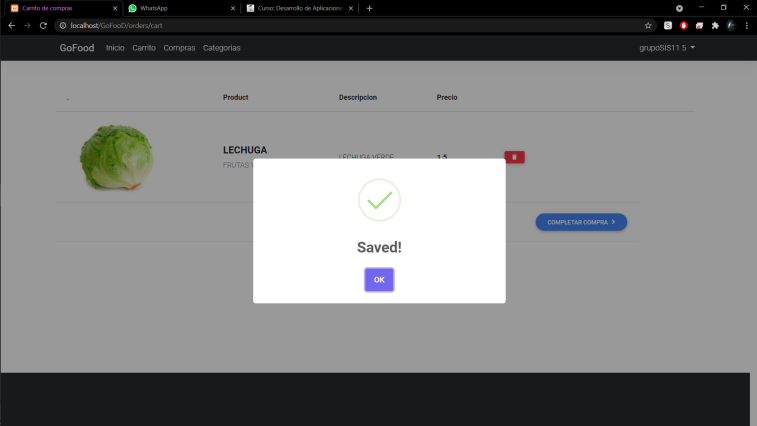


En el botón (**Carrito**) del menú al darle click nos muestra la siguiente pantalla la cual contiene los datos del producto que seleccionamos en la pantalla de categoría además muestra un botón para **eliminar** producto, en la parte de abajo muestra el **total** de todos los productos que hemos seleccionado y un botón(**Completar compra** ) para completar dicha compra, además si seleccionamos 2 del mismo producto se muestra un solo producto pero en el precio se le suma el otro producto (azúcar costo por unidad es de $1.5 si seleccionamos 2 productos de azúcar en el precio del producto mostrará $3.00).





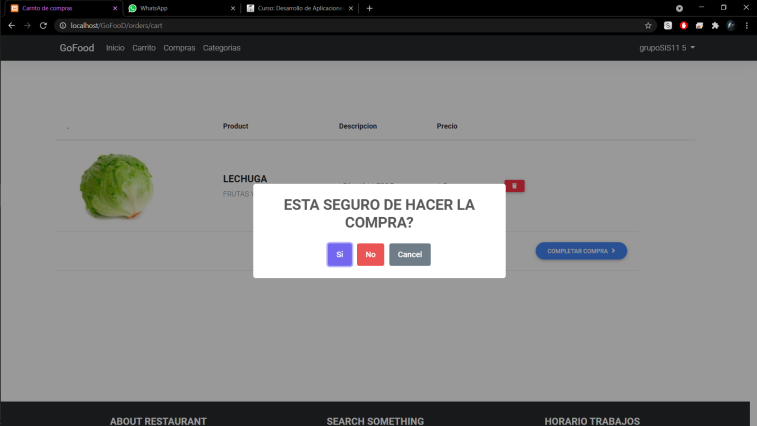
Al darle click al botón de eliminar producto en la pantalla de carrito se te muestra la siguiente alerta.



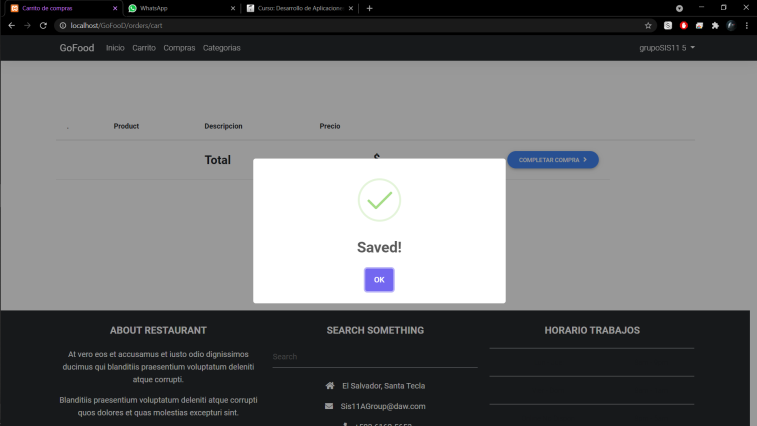
Si la respuesta es si, automaticamente se borra el

producto (Se eliminó la azucar) posteriormente

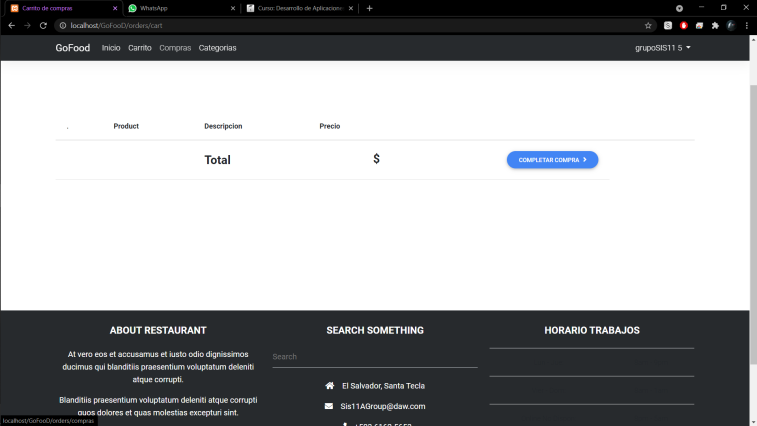
muestra una alertar confirmando que dicho cambio fue guardado .

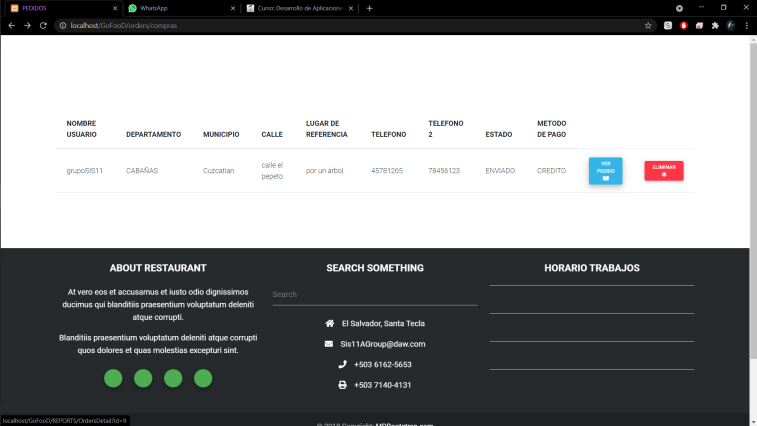


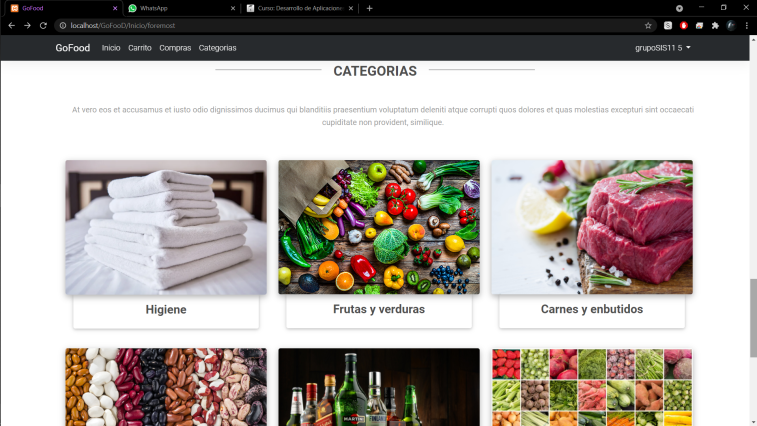
Cuando le das click al botón (**Completar compra**) te muestra una alerta preguntando si estas seguro de realizar la compra.



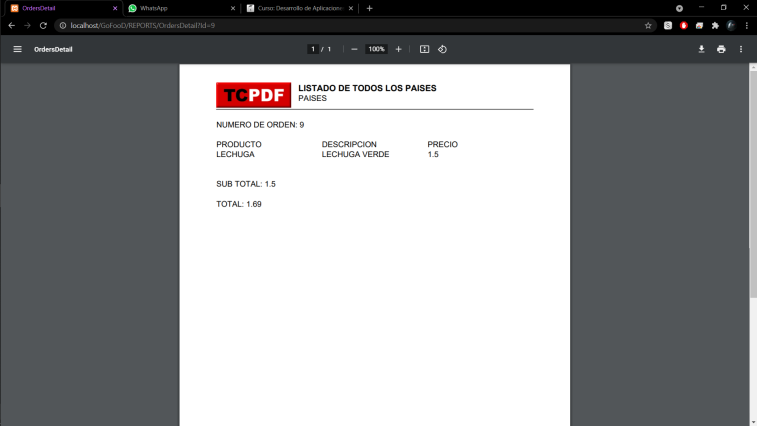
Si la respuesta es que si quieres realizar la compra, se mostrará una alerta confirmando que has realizado la compra.

Luego de realizar la compra dichos producto se eliminarán de la pantalla carrito, ya que tu has realizado la compra con éxito y si quieres otros productos repite los pasos anteriores para no volver a comprar productos que ya adquiriste con anterioridad.

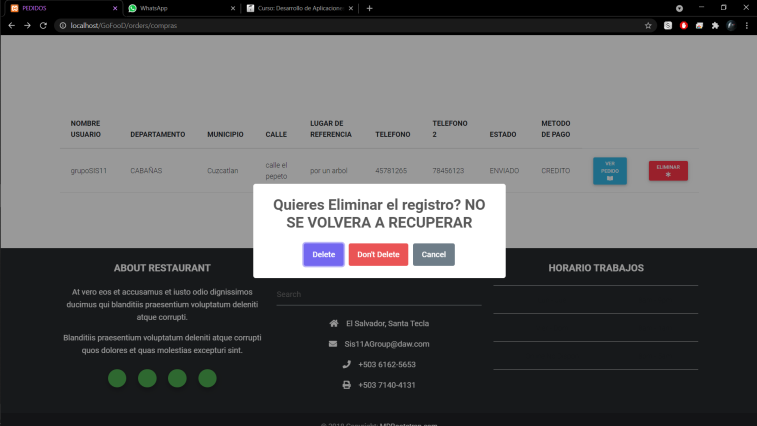


Al dar click al boton **(Compras)** se muestra una pantalla la cual te muestra el detalle de tu compra y contiene 2 botones uno para **eliminar** dicho registro y otro para que te **muestre el ticket** de tu compra.

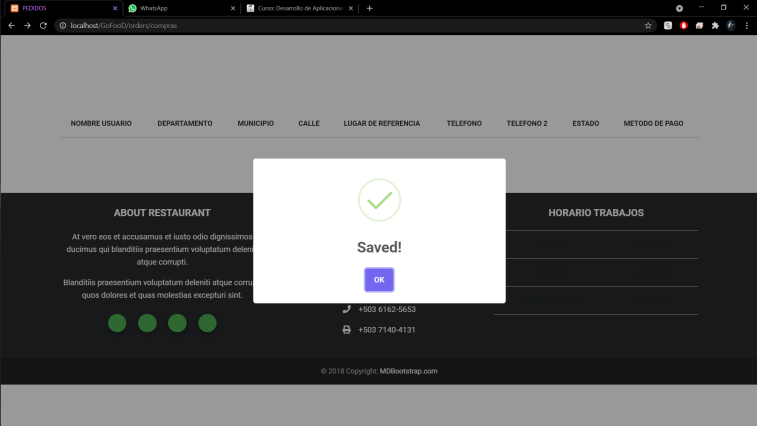


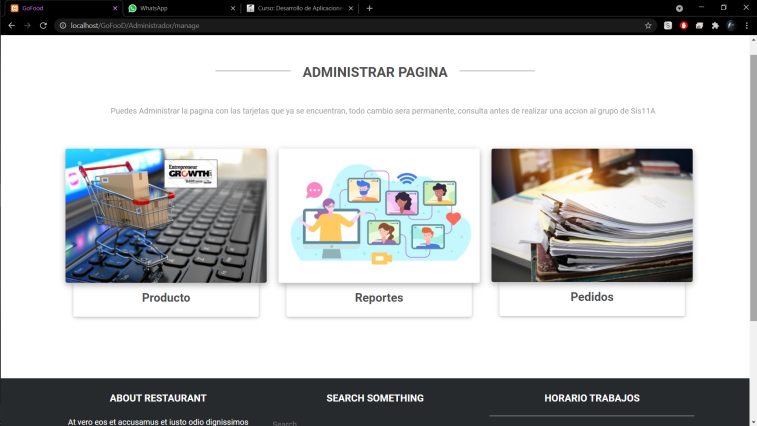


Cuando le das click al botón (**Ver pedido**) te muestra un pdf el cual es tu ticket de compra en el cual especifica tu detalle de compra.

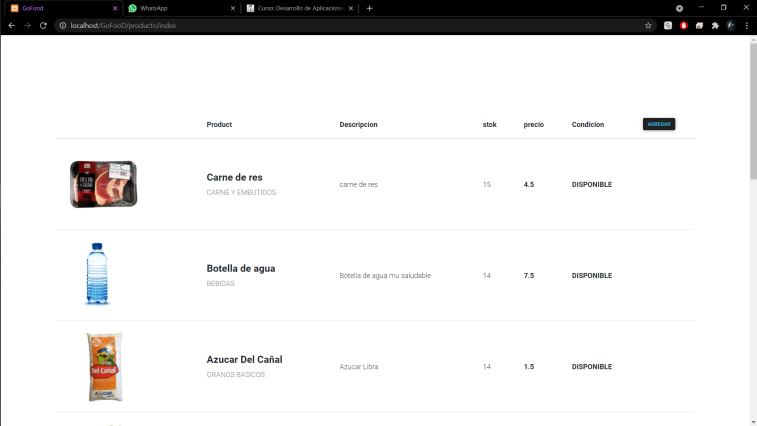


Cuando le da click al botón(Eliminar) te muestra la suiguente alerta, y una ves eliminado no podras ver mas ese detalle de tu compra.

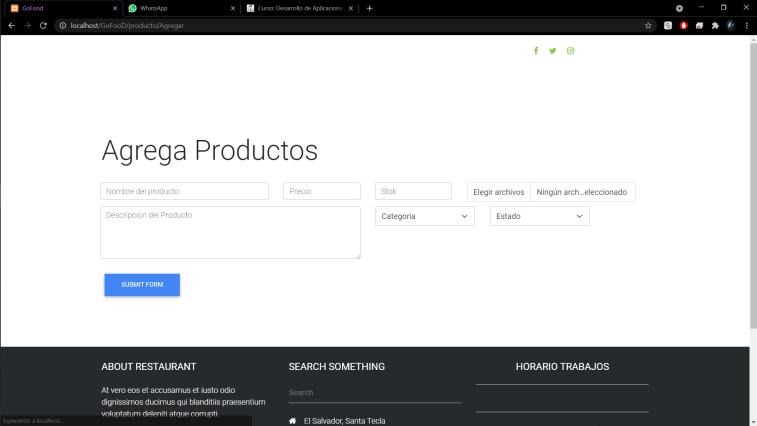
Una ves le des click al botón de eliminar, te muestra una alerta que significa que la operación fue concretada con éxito, en la tabla de tu detalle pedido se actualiza y ese datos desaparece para siempre.

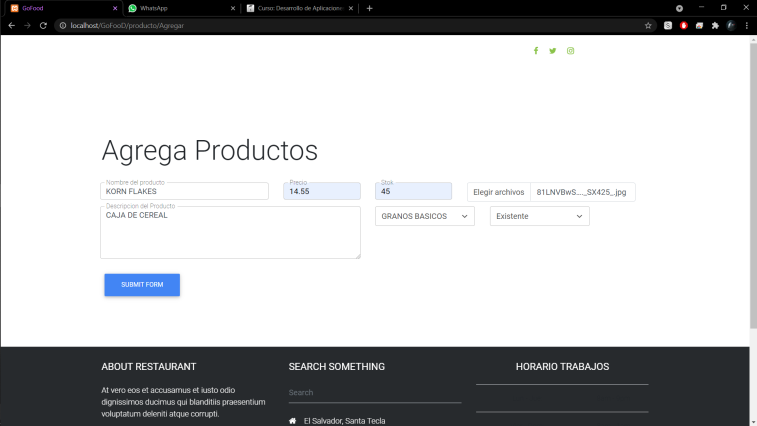
Rol como empleado….

Una ves ingresado con un correo y una contraseña que fueron validadas y su rol es empleado, al dar click en iniciar sesion se te muestra la siguiente pantalla en la cual tu menu estara limitado y se le mostrara unicamente (Producto, reportes y Pedidos).

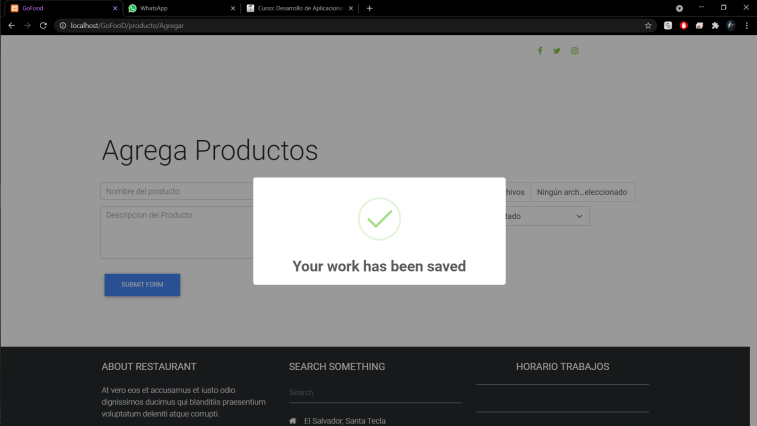
Cuando el empleado le de click al botón producto se le mostrará una pantalla donde podrá ver los productos existentes y además de ello un botón donde podrá agregaras mas productos pero estará limitado a solo agregar productos el empleado no podrá modificar ni borrar productos y tampoco un empleado puede comprar productos dentro de la aplicación una ves registrado como empleado.



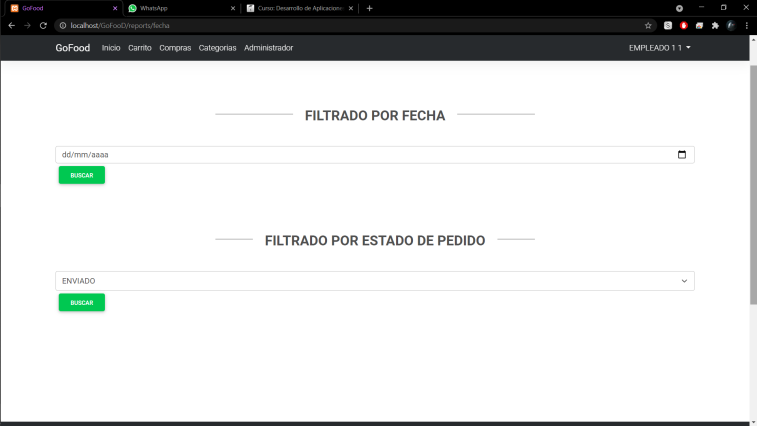
Una ves de click en el botón agregar, se mostrar una pantalla en la cual podra agregar el producto.

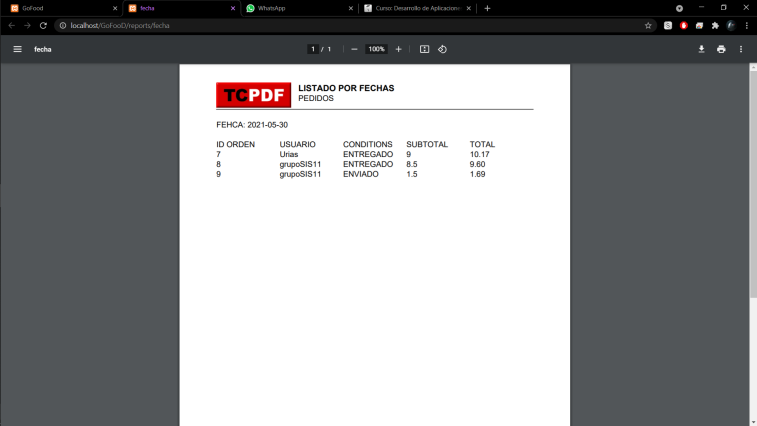


En dicho formulario se piden datos como nombre del producto, precio, cantidad, foto, descripción, categoría y su estado, en categorías es un select que muestra únicamente las categorías que existen en la base de datos, todos los campos son requeridos obligatoriamente.

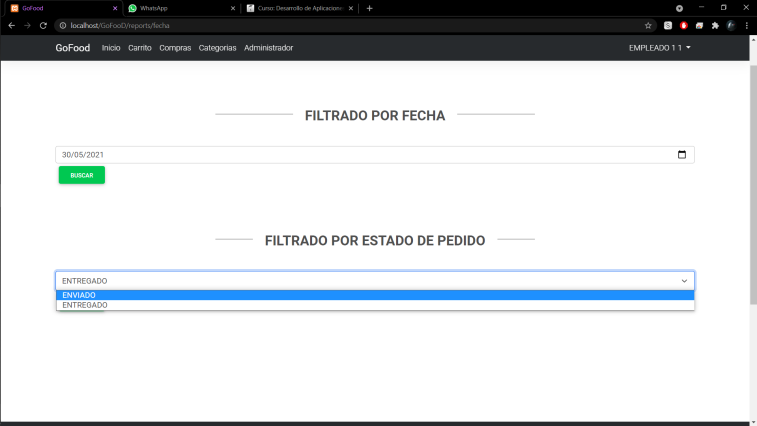


Al dar click en el botón agregar, mostrar la siguiente alerta la cual confirma que el producto fue ingresado con éxito, los campos se reinician y se borran los datos que digito el empleado, una ves el producto haya sido ingresado el empleado no podrá modificarlo.

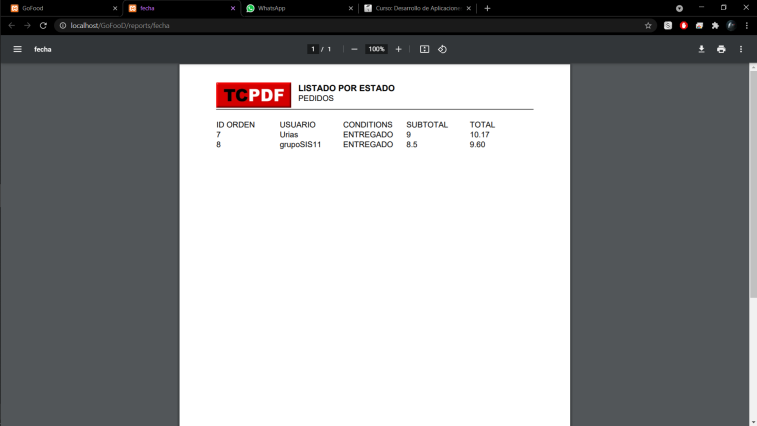
 Cuando el empleado de click al botón de Reportes de le muestra la siguiente pantalla, en la cual mediante un select podrá generar reportes parametrizados ya sea por días o por estado de pedidos.

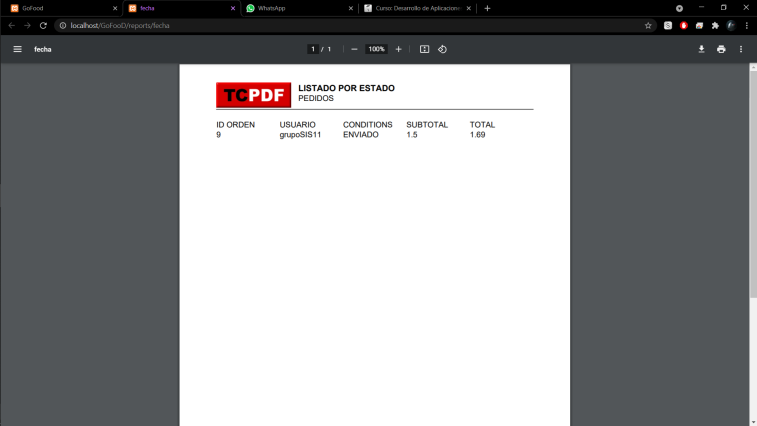


Cuando el empleado le de click al botón de generar reporte por fecha se abrirá una nueva pestaña en la cual se mostrará un pdf con los datos de los usuarios que realizaron pedidos en la fecha elegida por el empleado.

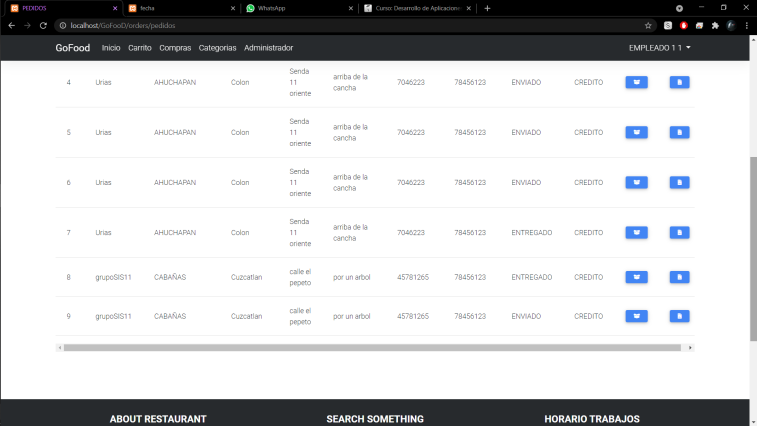


En el select de filtrado por estado de pedido, se mostrará únicamente 2 parámetros los cuales son enviados y entregados que son los 2 estados de los pedidos realizados por el usuario.

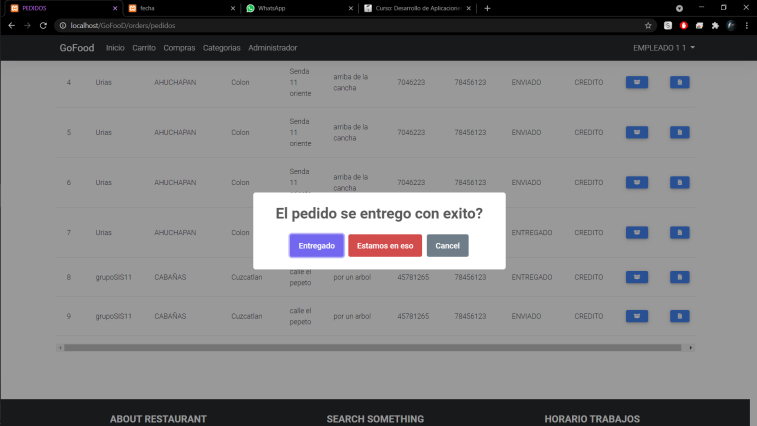
Cuando el empleado seleccione **Entregados**  en el estado de pedidos y le de click al botón de generar reportes se le abrirá otra pagina la cual le mostrará mediante un pdf los usuarios cuyo estado de pedido sea.

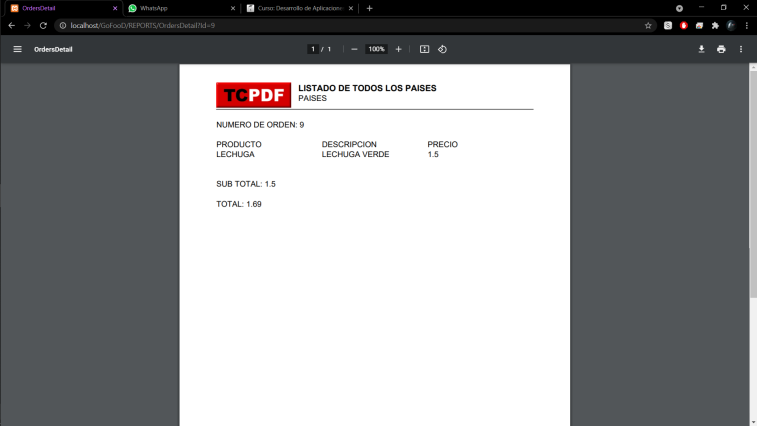


Cuando el empleado seleccione **Enviado** en el estado de pedidos y le de click al botón de generar reportes se le abrirá otra pagina la cual le mostrará mediante un pdf los usuarios cuyo estado sea Enviado.



Cuando el empleado le de click al botón de **Pedidos** se le mostrar la siguiente pantalla la cual contendrá el detalle pedido del cliente y aunque el cliente haya borrado el detalle pedido al empleado siempre se le mostrará, dicha pantalla contiene 2 botones, el primero es para cambiar el estado del detalle pedido y el otro es para ver el ticket del cliente

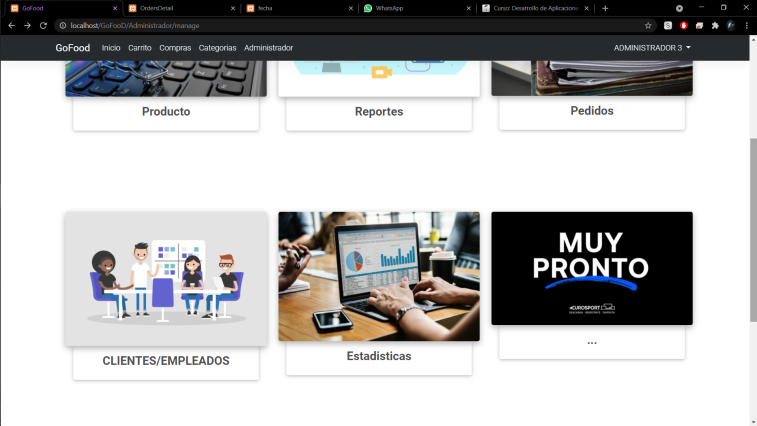
 Cuando el empleado quiera cambiar el estado del pedido se le mostrará la siguiente alerta en la cual le pregunta si el pedido se entregó o está en curso(Enviando).



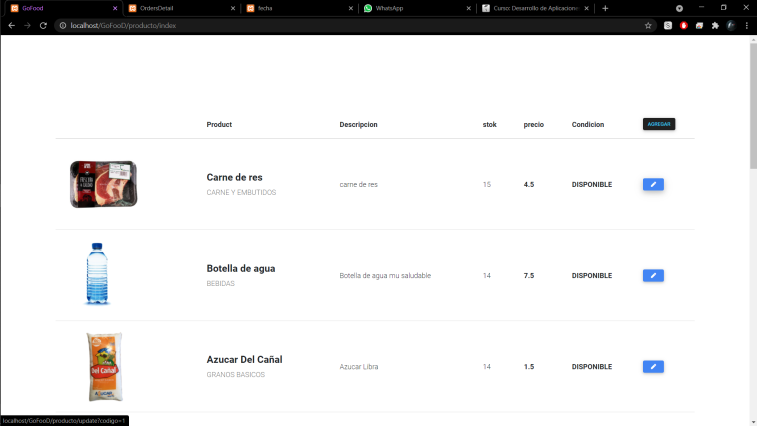
Si el empleado selecciona el botón de ver ticket del pedido se le abrirá otra pestaña la cual le mostrará un pdf el cual contiene los datos del pedido.

ADMINISTRADOR.

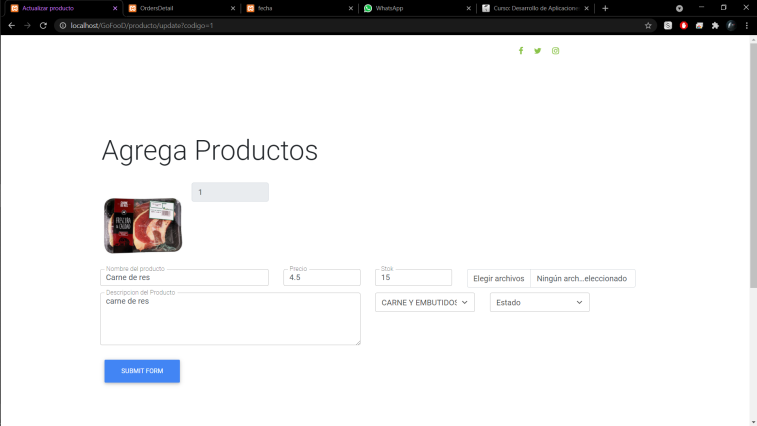
Si al ingresar su correo y contraseña se verifican los datos en la base de datos y su rol es administrador el tendrá acceso a todo lo anterior y otras cosas que a continuación se muestran.

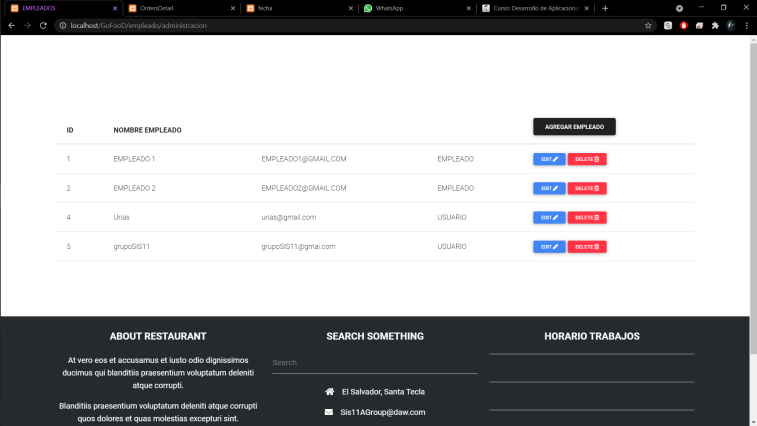


Una ves registrado como administrador, su menú es más amplio y se le muestran Clientes-empleados y estadísticas que a clientes y a empleados nos se les mostraba.

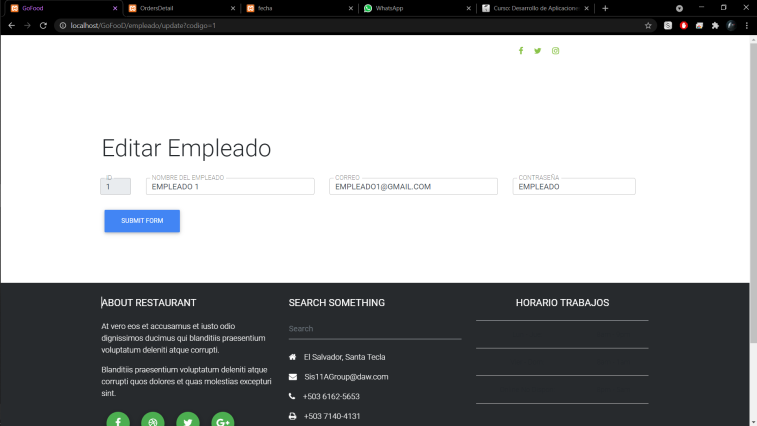


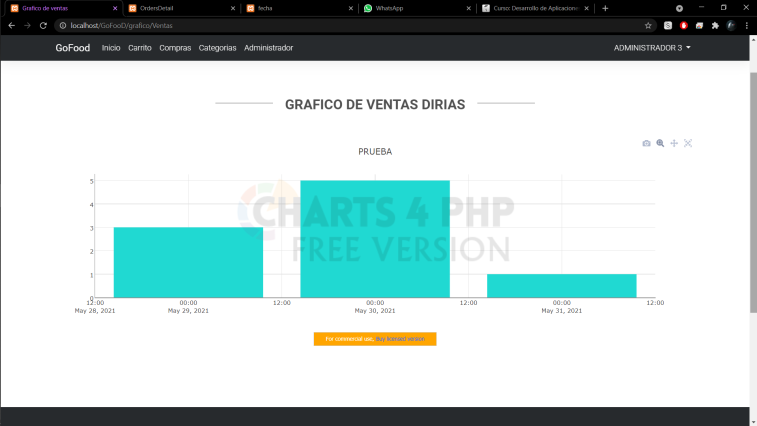
 Cuando el administrador da click en el botón producto se le muestra la siguiente pantalla, es casi la misma que se le muestra al empleado pero al  administrador se le muestra un botón para editar los productos.

Cuando el administrador le de click al boton de ditar producto se le mostrar la siguiente pantalla donde tiene los permisos para editar los productos ya ingresados por parte del empleado o de el mismo y el administrador en el Stock puede agregar mas productos en ves de agregar mas productos de los mismo el solamente le incrementa la cantidad de existencia.



Si el empleado da click al botón (**Clientes-Empleados**) se le muestra la siguiente pantalla en la cual se puede ver todos los clientes y empleados registrados y el administrador tiene los permisos para eliminar, editar y agregar nuevas cuentas de los empleados y clientes.

 Cuando el administrador da click en el botón editar se muestra la siguiente pantalla en la cual el administrador tiene los derechos de editar nombres, correos y contraseñas.



El administrador al darle click al botón de **Estadísticas** se le abrirá otra pagina la cual le va a mostrar una grafica de línea, en la cual se plasman las ventas realizadas por dia.

# Bibliografía

# (Galdame, 2021)

Galdame, P. R. (30 de 05 de 2021). *Blogdelproferenan*. Obtenido de Blogdelproferenan: http://blogdelproferenan.blogspot.com/2017/06/diccionario-de-datos.html

mid42. (30 de 05 de 2021). *mid42*. Obtenido de mind42: https://mind42.com/public/3d332a89-333d-44e1-acfb-4a3ebfc3e52c

kfc. (30 de 04 de 2021). *kfc.com.sv*. Obtenido de kfc: https://www.kfc.com.sv/

walmart. (30 de 04 de 2021). *walmart*. Obtenido de walmart: https://centroamerica.walmart.com/1/account/info?countryId=4&formatId=16&utm\_source=google\_ads&utm\_medium=cpc&utm\_campaign=ecommerce&utm\_term=&utm\_content=search&gclid=Cj0KCQjw78yFBhCZARIsAOxgSx1Om1qA01\_P1C0\_qDQ\_SLarF1eOqqMDyPQcv0P-ylxk2DMoX6DCPmYaAshlEAL