

ENTREGA FINAL

ComidappBD

Introducción:

Las necesidades básicas de las personas han sido un pilar importante para que la tecnología pueda sacar provecho de sus funcionalidades, Comidapp es un software orientado a esos comercios que no pueden tener su medio tecnológico para poder progresar como empresa, este modelo de datos esta basado en un software donde esos comercios de comida callejeros puedan generar ingresos mediante las ventas web, esta idea salió de un proyecto que estoy realizando por fuera de coder y me gustaría sacarle provecho en esta plataforma.

Objetivo:

El objetivo principal de este proyecto es poder diseñar el modelo de datos que permita a los usuarios y clientes de la app poder registrarse y promocionar sus locales mediante esta aplicación, donde se podría respaldar, recuperar y hacer un análisis del rendimiento del mismo local, esto consiste en un administrador del sitio web que confirme o rechace solicitudes de los clientes que quieran unirse a este servicio, el local debería cumplir requisitos mínimos para tener el ingreso y principalmente que este en regla.

Situación problemática:

En el avance tecnológico ya se inventaron ideas y soluciones para problemas de necesidad básica, en este caso ya existe muchos softwares que permiten tener la comida al alcance de tu celular pero por lo menos en Uruguay están dejados de lados los comercios de comida callejera, habiendo muchos comercios que ya están fijos en lugares y habilitados por la Intendencia de Montevideo no pueden contar con este servicio por requisitos que muchos no cumplen, este espacio les permitiría a los comercios poder expandirse al área tecnológico. Esta base de datos esta pensada para este sistema en especial, los requisitos que necesito y lo que creo que seria mas seguro y sencillo para administrar los datos

Modelo Entidad Relación (MER.)

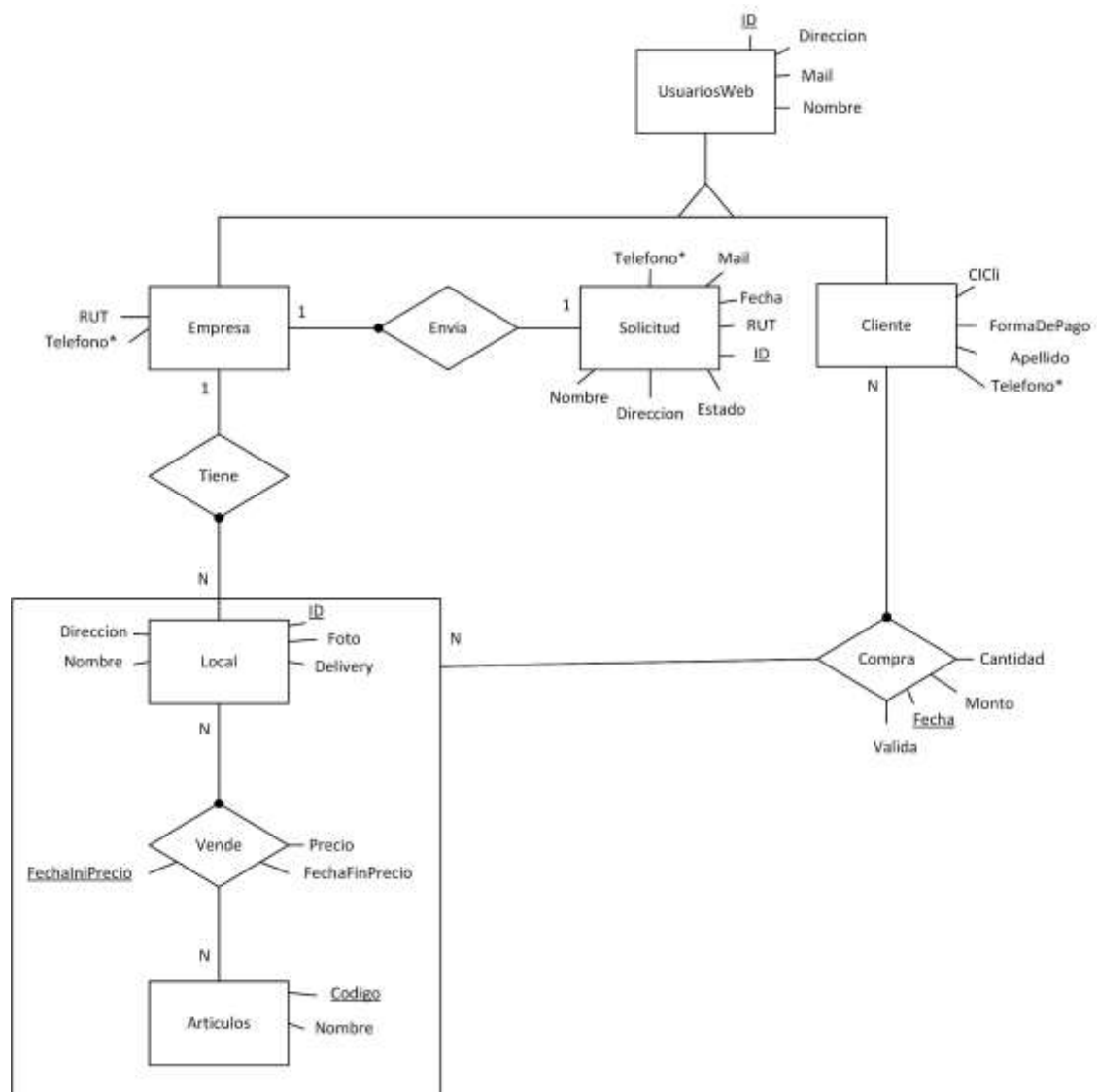
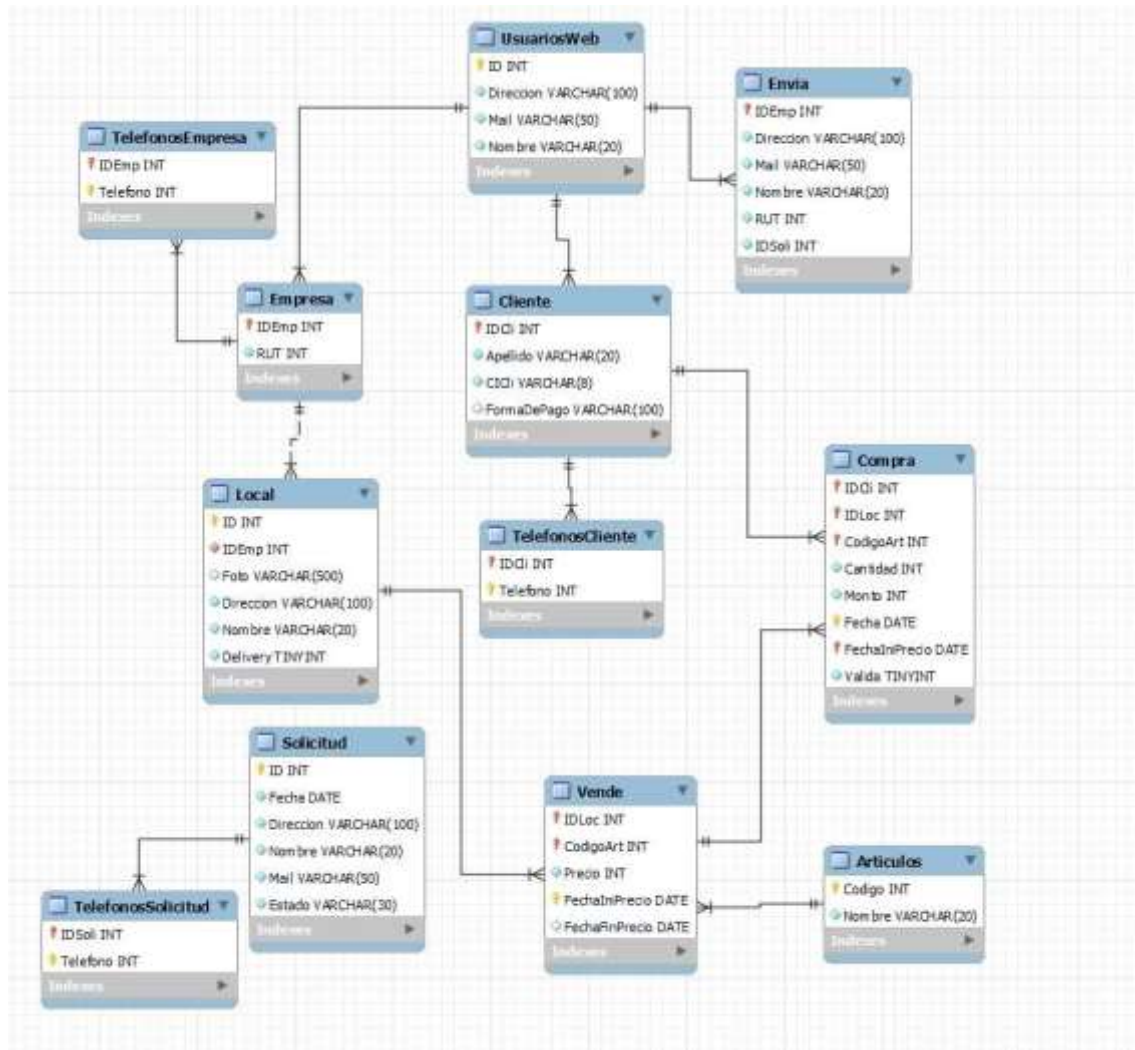


DIAGRAMA ENTIDAD RELACIÓN:



RELACIONES DE LAS TABLAS

Usuarios Web:

- **PRIMARY KEY:** ID

Cliente:

- **PRIMARY KEY:** IDCli
- **FOREIGN KEY:** IDCli refiere a UsuariosWeb.ID

Solicitud:

- **PRIMARY KEY:** ID

Empresa:

- **PRIMARY KEY:** IDEmp
- **FOREIGN KEY:** IDEmp refiere a UsuariosWeb.ID

TelefonosCliente:

- **PRIMARY KEYS:** IDCli, Telefono
- **FOREIGN KEY:** IDCli refiere a Cliente.IDCli

TelefonosSolicitud:

- **PRIMARY KEYS:** IDSoli, Telefono
- **FOREIGN KEY:** IDSoli refiere a Solicitud.ID

TelefonosEmpresa:

- **PRIMARY KEYS:** IDEmp, Telefono
- **FOREIGN KEY:** IDEmp refiere a Empresa.IDEmp

Local:

- **PRIMARY KEY:** ID
- **FOREIGN KEY:** IDEmp refiere a Empresa.IDEmp

Artículos:

- **PRIMARY KEY:**Codigo

Envía:

- **PRIMARY KEY:** IDEmp
- **FOREIGN KEYS:** IDEmp refiere a UsuariosWeb.ID, IDSoli refiere a Solicitud.ID

Vende:

- **PRIMARY KEYS:** IDLoc, CodigoArt, FechalniPrecio
- **FOREIGN KEYS:** IDLoc refiere a Local.ID, CodigoArt refiere a Articulos.Codigo

Compra:

- **PRIMARY KEYS:** IDCli, IDLoc, CodigoArt, Fecha, FechalniPrecio
- **FOREIGN KEYS:** IDCli refiere a Cliente.IDCli, IDLoc, CodigoArt, FechalniPrecio refieren a Vende.IDLoc, Vende.CodigoArt, Vende.FechalniPrecio

Vistas

SUCURSAL QUE VENDE MAS

Proposito:

CREATE VIEW SucursalMasVende

Esta vista esta echa para localizar cual es la sucursal que mas vende, usamos un **join** para vincular el local con las ventas. Este dato nos ayuda a tener una idea de cual es la sucursal que mejor le va en el rubro de la comida callejera dentro de nuestra app

ARTICULO MAS VENDIDO

Propósito:

CAEATE VIEW ArticulosMasVendidos

En esta vista tenemos los artículos mas vendidos dentro de la aplicación donde vinculamos la cantidad de articulo vendido, esto nos ayuda a tener una idea de cual es el articulo que mas se vende, esto nos ayuda a sacar ofertas o saber los gustos de nuestros clientes

Funciones

Propósito:

CREATE FUNCTION TotalEmpresasYClientes()

Necesitamos una función que sume cuantos clientes tenemos y cuantas empresas, esto nos ayuda a tener un control dentro de nuestro sistema, sacamos los datos de las tablas de empresa y clientes.

Propósito:

CREATE FUNCTION FormasDePagosCuenta()

Esta otra función salió para la necesidad de saber una estadística de si los clientes pagan con tarjeta o con efectivo para sacar promociones para la gente que usa tarjetas o efectivo y tener un promedio, se cuenta de las tablas de clientes las formas de pago.

Stored Procedures

Actualizar estado de solicitud

Propósito:

CREATE PROCEDURE ActualizarEstadoSolicitud()

Este SP se creo con el fin de poder actualizar el estado de una solicitud para corregir errores si una función del backend no llega a funciona o falla algún servicio.

Actualizar estado de delivery

Propósito:

CREATE PROCEDURE ActualizarDelivery()

Bueno por la elección que decidí crear este SP fue para que el administrador de la base pueda modificar el estado de servicio de delivery con el id del local.

TRIGGERS

Propósito:

CREATE TRIGGER CheckDeliveryAntesDe

Este trigger lo vamos a utilizar para chequear si el local tiene delivery cuando hacemos la compra, ni bien esta función esta elaborada con un desarrollo aparte, también me gustaría que la base de datos pueda definir si el local tiene o no delivery, también que el administrador pueda ver si agrega o no el servicio cuando este local cuente con el mismo.

Propósito:

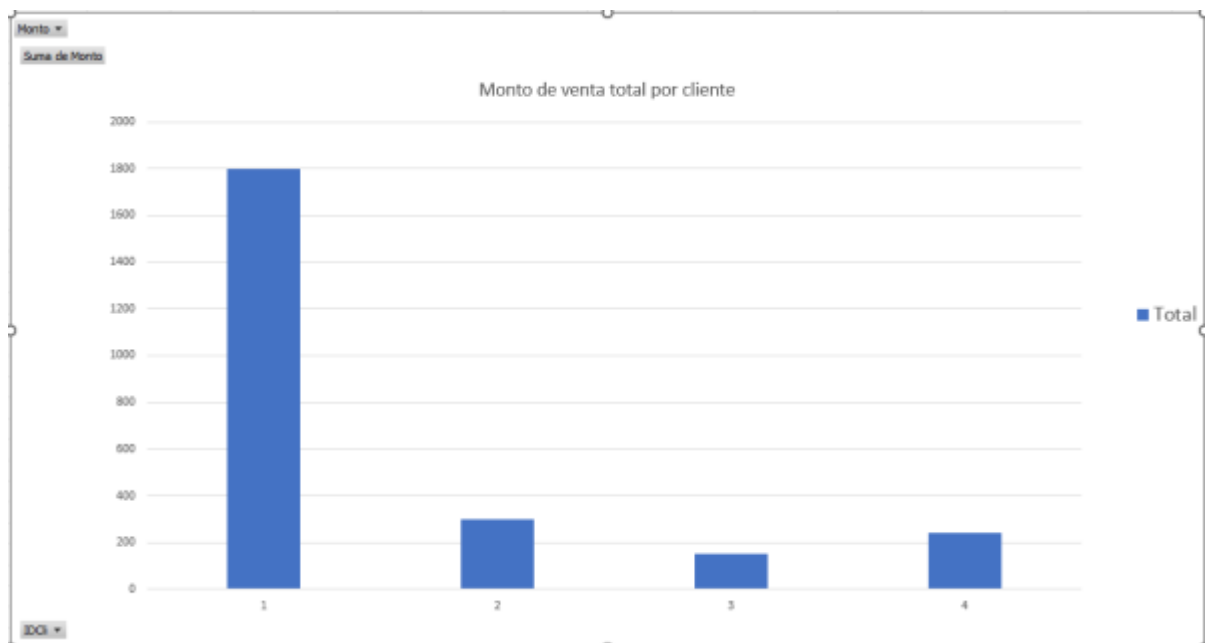
CREATE TRIGGER CheckDireccionEnviaAntesDe

Este trigger nos va ayudar a saber si la dirección donde se hace el envio tiene dirección, en caso de que no tenga o tenga dato nulo no se podría hacer un envio lanzando un error 45000.

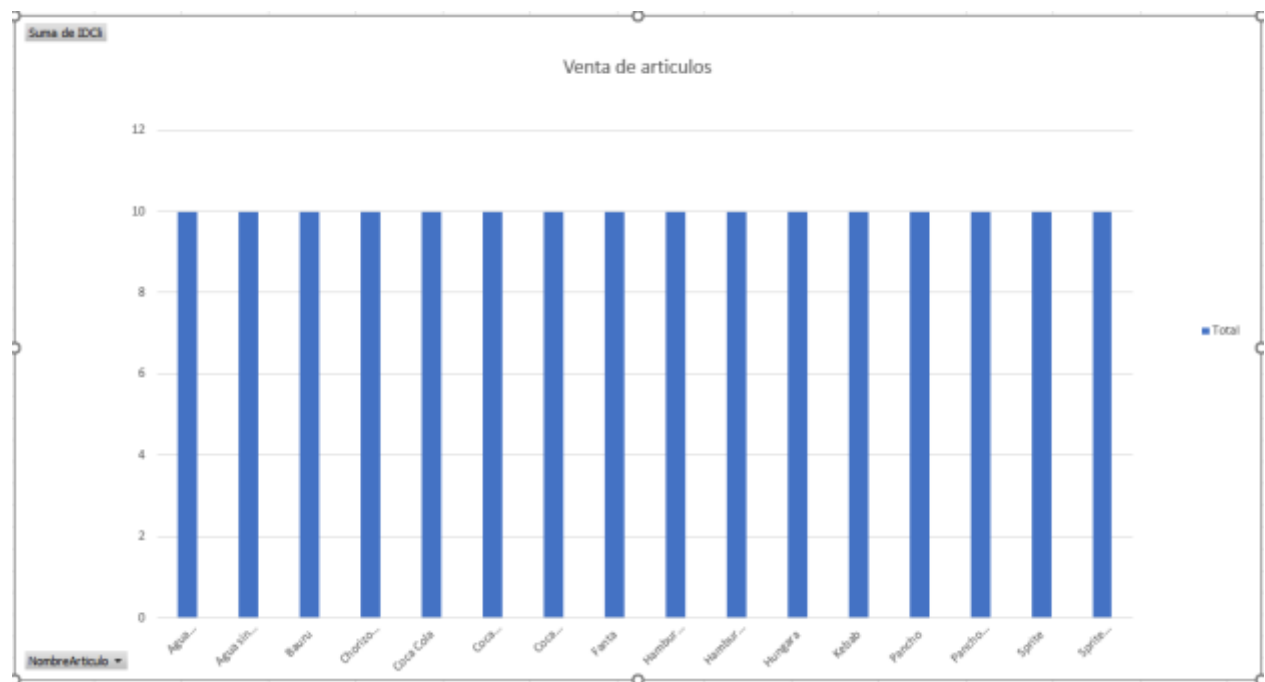
Analítica de la base de datos:

El análisis de datos se basan para el área de marketing y financiero, usando grafica para evaluar el articulo mas vendido y el monto por cliente.

Monto de venta total por cliente:



Ventas de articulo:



Link del repositorio:

[https://github.com/Rodridls22/ENTREGA-CODER-
/tree/main/ENTREGA%20FINAL](https://github.com/Rodridls22/ENTREGA-CODER-/tree/main/ENTREGA%20FINAL)