

**Ingeniería del Software.**

**Aplicación de Base de Datos.**

**Efrén Jiménez Delgado.**

**Proyecto Final Súper Familia Sánchez.**

**Roque Chacón Corrales .**

**Kevin Arias Arce.**

**Primer cuatrimestre ,2017.**

Contenido

[ Resumen Ejecutivo 3](#_Toc480827087)

[ Objetivo General 4](#_Toc480827088)

[ Objetivos Específicos 4](#_Toc480827089)

[ Introducción 5](#_Toc480827090)

[ Descripción del problema 6](#_Toc480827091)

[ Desarrollo 7](#_Toc480827092)

[ Conclusiones 8](#_Toc480827093)

[ Recomendaciones 9](#_Toc480827094)

[ Bibliografía 10](#_Toc480827095)

[ Anexos. 10](#_Toc480827096)

* Resumen Ejecutivo

El presente documento contiene la solución a un problema de una micro-empresa, y a su vez el paso a paso de manera más correcta de ejecutarlo.

En si el proyecto se basa en la creación, optimización de una base de datos para tener un control total de la empresa, un plus será la creación de triggers y funciones dentro de la base de datos, a lo cual le podemos llamar lógica de negocios.

Puede afirmarse, que una buena base de datos puede llevar al éxito a una determinada empresa.

* Objetivo General

Como objetivo general nos hemos trazado la creación de una base de Datos, cumpliendo con todas las normas de la misma.

* Objetivos Específicos
* Diseñar la estructura de la base de datos necesaria para el súper, definiendo sus restricciones y relaciones.
* Crear los modelos de desarrollo de los objetos de la BD.
* Crear diferentes funciones.
* Crear Trigers.

# Introducción

Se deberá crear una propuesta de base de datos aplicada en la vida real, en el cual se pretenderá solventar con un modelo de base de datos.

Este debe contemplara toda la capa lógica de negocios en base de datos, las restricciones, procesos y validaciones para mantener la integridad de la base de datos.

También se realizará los modelos respectivos, así como la creación de Triggers y funciones.

* Descripción del problema (Historia de la empresa y solución a construir).

La micro-empresa está ubicada en el sector de San Carlos, surge de la carencia de un súper en la comunidad, y la visión de una familia, El súper familia Sánchez abrió las puertas hace 3 años, por inconvenientes se tuvo que cerrar, algunos de estos eran la falta de proveedores, un deficiente control en la entrada y salida de productos no se llevaba un control del mismo, lo cual a su vez termino con una mala toma de decisiones, pero hace un año volvió a abrir sus puertas y retomar su funcionabilidad, brindando a una comunidad, un servicio de calidad, con el pasar de días, meses y el súper ha tenido una aceptación muy buena, es ahí donde la familia se plantea hacer una mejoría, lo primero será surtir de más productos su empresa, también mantener sus dos cajas siempre abiertas y en total disponibilidad, también contar con una base de datos que le ayude a llevar un control de entradas y salidas de productos así como un historial de las compras de los consumidores para mejor la calidad y ser líderes en dicho campo. EL súper está en proceso de crecimiento y la familia sabe que es hora de tomar decisiones acertadas y ser objetivos, para alcanzar el éxito que siempre han anhelado.

# Desarrollo

Concepto “Base de Datos” Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para tomar decisiones acertadas a futuro.

Lo primero será, analizar la problemática y brindar una solución acertada y contundente al problema.

Y a su vez crear algo eficaz y eficiente teniendo en cuanta la visión que tiene la empresa y el crecimiento planteado de la familia Sánchez.

Para ello seguiremos las reglas de la normalización de una base de datos y construiremos modelos normalizados.

Lo segundo empezaremos con la construcción del Modelo Relacional,

cliente(id, nombre, apellido1, apellido2, telefono, montoLimite,direccion)

clienteDebe(id, id\_cliente, montoDebe, fechaLimiteCancelacion)

estadoCuenta(id, id\_cliente, saldoAnterior, abono, saldo\_actual, fecha\_hora)

caja(codigo, montoIncial)

detalleVenta(id, descripcion, precio, cantidad, impuesto, codigo\_caja)

entrada(id, descripcion, monto fecha\_hora)

salida(id, descripcion, monto fecha\_hora)

usuario(id, usuario, clave, nombre, apellido,genero ,telefono, idPuestos)

puesto(id,puesto)

producto(codigo, descripcion, costoBase, precioVenta, cantidad, enStock)

ticket(id, nombre, localidad, direccion, propietario, telefono, mensajefinal, ultimalinea)

ventaContado(id, idTicket, monto, pagaCon, suVuelto, codigoCaja)

ventaCredito(id, idTicket, monto, idCliente,idUsuario,codigoCaja

venta(id, fecha\_hora, tipoPago, idUsuario, codigo\_caja, idDetalleVenta)

proveedor(codigo, nombre, direccion, telefono, fechaPedido,compañia)

Seguidamente la creación de los esquemas y tablas, acá unos ejemplos de los mismos;

-------Esquemas-----

CREATE SCHEMA admin;

CREATE SCHEMA usuarios;

CREATE SCHEMA clientes;

----Cambio Esquema-----

SET search\_path = clientes;

---Tablas Esquema Cliente---

CREATE TABLE cliente

(

id serial NOT NULL,

nombre VARCHAR NOT NULL,

apellido1 VARCHAR NOT NULL,

apellido2 VARCHAR NOT NULL,

telefono VARCHAR NOT NULL,

montoLimite DOUBLE PRECISION NOT NULL,

direccion VARCHAR NOT NULL

);

ALTER TABLE cliente ADD CONSTRAINT pk\_cliente PRIMARY KEY (id);

CREATE TABLE clienteDebe

(

id serial NOT NULL,

idCliente INT NOT NULL,

montoDebe DOUBLE PRECISION NOT NULL,

fechaLimiteCancelar TIMESTAMP NOT NULL

);

Luego de crear las tablas vamos a construir unas vistas para mostrarle al admin diferentes movimientos para que tenga una acertada toma de decisiones.

set search\_path = clientes;

CREATE VIEW vista\_debe\_cliente AS

SELECT cliente.id, nombre, apellido1, apellido2, telefono, montolimite, direccion,clientedebe.montodebe

FROM clientes.cliente INNER JOIN clientes.clientedebe ON clientedebe.idcliente = clientes.cliente.id

ORDER BY clientedebe.montodebe;

También crearemos índices, un índice es una estructura de disco asociada con una tabla o una vista que acelera la recuperación de filas de la tabla o de la vista.

Hay diferentes tipos de índices

Agrupado

* Los índices agrupados ordenan y almacenan las filas de los datos de la tabla o vista de acuerdo con los valores de la clave del índice.

No agrupado

* Los índices no agrupados tienen una estructura separada de las filas de datos.

set search\_path = usuarios;

CREATE INDEX index\_estado\_cuenta ON usuarios.estadocuenta (idcliente,saldoanterior,abono,saldoactual);

CREATE UNIQUE INDEX index\_venta\_contado ON usuarios.ventacontado(idticket);

Tambien crearemos un diccionario de datos y Usuarios

CREATE USER administrador;

ALTER ROLE administrador WITH SUPERUSER CREATEROLE;

ALTER ROLE administrador WITH PASSWORD '12345';

Dicionario de Datos

SELECT \* FROM pg\_stat\_user\_indexes; --Mostrar Indices

SELECT proname FROM pg\_proc WHERE prolang = 11859 ORDER BY proname --Mostrar Funciones

SELECT tgname FROM pg\_trigger WHERE tgisinternal = false ORDER BY tgname; --Mostrar Tiggers

SELECT table\_schema AS schemas FROM information\_schema.tables WHERE table\_name !~~ '%pg%' AND table\_schema !~~ '%schema%' GROUP BY schemas;--Mostrar esquemas

SELECT table\_name FROM information\_schema.columns WHERE table\_name !~~ '%pg%' AND table\_name !~~ '%sql%' AND is\_nullable = 'NO' GROUP BY table\_name; -- Mostrar tablas

SELECT table\_name, column\_name FROM information\_schema.columns WHERE table\_name !~~ '%pg%' AND table\_name !~~ '%sql%' AND is\_nullable = 'NO'

GROUP BY table\_name, column\_name ORDER BY table\_name, column\_name; -- Mostrar tablas y atributos

Por último, haremos una serie de funciones para un mayor aprovechamiento de nuestra base de datos los cuales nos devolverán un dato el cual lo podremos utilizar para una toma de decisión.

set search\_path = clientes;

--------------------funciones Cliente-------------------------------

CREATE OR REPLACE FUNCTION insertarCliente(nombre VARCHAR, apellido1 VARCHAR, apellido2 VARCHAR, telefono VARCHAR, montolimite DOUBLE PRECISION, direccion VARCHAR) RETURNS INTEGER AS

$clienteTrigger$

DECLARE ID INTEGER;

BEGIN

INSERT INTO clientes.cliente("nombre", "apellido1", "apellido2", "telefono", "montolimite", "direccion")

VALUES (nombre, apellido1, apellido2, telefono, montolimite, direccion);

ID = (SELECT currval('clientes.cliente\_id\_seq'));

RETURN ID;

END

$clienteTrigger$ language plpgsql;

# Conclusiones

**Un buen planteamiento y diseño inicial de la base de datos soluciona muchos de los problemas a una empresa, así como a su vez llevará a una buena toma de decisiones ya que se podrá contar con un historial, y así tener una idea de cuando invertir y cuando mantener el negocio rodando no más.**

# Recomendaciones

Darle mantenimiento a la base de datos en periodos cortos, así como respaldar los datos.

**Optimizar los índices.** Tener una buena relación de índices entre tablas es básico para las búsquedas relacionales funcionen correctamente.

Siempre mantener una integridad en los datos es indispensable.

# Bibliografía

Web:

https://rua.ua.es/dspace/bitstream/10045/2990/1/ApuntesBD1.pdf

# Anexos.