



PROYECTO

FINAL BASE DE DATOS

I



PREPARED FOR

Base de Datos I

Unifranz sede el ALTO - 2021

PREPARED BY

Integrantes del equipo:

1. Kevin Michael Apaza Lara

Nombre del equipo: El caminante

1. Introducción.

Especificar cuál es el objetivo del proyecto.

Ejemplo: *El objetivo del siguiente proyecto es el de poder encontrar fácilmente puntos de cajeros automáticos o bancos cercanos a tu posición o en todo caso sucursales de oficinas cercanas a tu posición.*

Objetivo:

El objetivo propuesto es crear un sistema de información e interfaz gráfica para una empresa (agencia) de pollos, en donde se podrá acceder fácilmente a los datos de los empleados y clientes, sus pedidos y gastos realizados por la empresa (empleados y camiones).

2. Diseño de la base de Datos.

2.1. Contexto de la Base de Datos.

Describir el contexto de la base de datos.

En la situación de un manejo adecuado de datos a la hora de comprar y vender pollos y está al pertenecer a una agencia definimos el nombre de la base de datos en relación a el nombre de su agencia "BD_agencia_de_pollos_Los_valles"

2.2. Análisis y definición de Tablas.

NOMBRE DE LA TABLA	Descripción
camiones	Almacena los camiones existentes en la empresa
mantenimiento_caminones	Esta tabla se relaciona con la tabla camiones para especificar los gastos de los diferentes camiones
empleados	Almacena datos de los empleados de la empresa
conductores	Esta tabla se relaciona con la tabla empleados y camiones para especificar aquellos empleados a cargo de camiones

oficinas	Esta tabla se relaciona con la tabla empleados para especificar aquellos empleados en oficinas.
puesto_ventas	Esta tabla se relaciona con la tabla empleados y camiones para especificar las cargas y recepción de los pollos.
pollos	Esta tabla se relaciona con la tabla camiones para especificar los costos y de donde llegan

2.3. Diseño de la Base de Datos.

2.3.1 Código SQL de las tablas.

Generar el código SQL de las tablas.

Ejemplo:

camiones	<pre> create table camiones (placa_camion varchar(50)primary key not null, tipo_camion varchar(50)not null, --pesado,medio,pequeño marca varchar(50)not null); go insert into camiones (placa_camion, tipo_camion, marca) values ('QWERTY1201','PESADO','Scania'), ('QWERTY1202','PESADO','Volvo'), ('QWERTY1203','PESADO','Volvo'), ('QWERTY1204','MEDIO','Iveco'), ('QWERTY1205','MEDIO','Iveco'), ('QWERTY1206','MEDIO','Isuzu'), ('QWERTY1207','MEDIO','Iveco'), ('QWERTY1208','LIGERO','Isuzu'), ('QWERTY1209','LIGERO','Scania'), ('QWERTY1210','LIGERO','Iveco'); </pre>
-----------------	--

mantenimiento_caminones	<pre> create table mantenimiento_caminones (id_mantenimiento integer identity primary key not null, precio_diesel_mes integer not null, tipo_mantenimiento varchar(100) not null, costo_mantenimiento integer not null, placa_camion varchar(50) not null, foreign key (placa_camion) references camiones(placa_camion)); go insert into mantenimiento_caminones(precio_diesel_mes, tipo_mantenimiento, costo_mantenimiento, placa_camion) values (3000,'Frenos','1500','QWERTY1201'), (2500,'Puertas','2500','QWERTY1202'), (2700,'Normal','0','QWERTY1203'), (2100,'Frenos','1600','QWERTY1204'), (2120,'Normal','0','QWERTY1205'), (2300,'Normal','0','QWERTY1206'), (2000,'Frenos','1000','QWERTY1207'), (1800,'Chaperia','2000','QWERTY1208'), (1900,'Normal','0','QWERTY1209'), (1500,'Normal','0','QWERTY1210'); </pre>
empleados	<pre> create table empleados (ci_empleado varchar(50) primary key not null, salario integer not null, nombre varchar (50)not null, apellido varchar(50)not null, area varchar(50)not null, telefono varchar(50)not null, fecha_contrato varchar(50) not null, fecha_nacimiento varchar(50) not null); go insert into empleados(ci_empleado, salario, nombre, apellido, area, telefono, fecha_contrato, fecha_nacimiento) values ('98765001LP',8500,'Marcos','Fernandez','Administ racion','+591 78910001','2010-10-10','1985'), ('98765002LP',8500,'Maria','Cordoba','Administrac ion','+591 78910001','2010-10-10','1985'), ('98765003LP',5000,'Pedro','Cornel','Ventas','+59 1 78910001','2010-10-10','1985'), ('98765004LP',5000,'Ramiro','Astral','Ventas','+5 91 78910001','2010-10-10','1985'), </pre>

	<pre> ('98765005LP',5000,'Fernando','Quispe ','Ventas','+591 78910001','2010-10-10','1985'), ('98765006LP',5000,'Carlos','Acho','Ventas','+591 78910001','2010-10-10','1985'), ('98765007LP',7000,'Manuel','Obrero','Contabilida d','+591 78910001','2010-10-10','1985'), ('98765008LP',6000,'Jorge','Fernandez ','Registros','+591 78910001','2010-10-10','1985'), ('98765009LP',6000,'Cosmo','Quispe','Registros',' +591 78910001','2010-10-10','1985'), ('98765010LP',5000,'Tereza','Morales','Recepcion' ','+591 78910001','2010-10-10','1985'), ('98765011LP',6500,'Javier','Apaza','Conductor',' +591 78910001','2010-10-10','1985'), ('98765012LP',6500,'Wanda','Quispe','Conductor',' +591 78910001','2010-10-10','1985'), ('98765013LP',6500,'Maria','Cazas','Conductor','+ 591 78910001','2010-10-10','1985'), ('98765014LP',5500,'Pedro','Calle','Conductor','+ 591 78910001','2010-10-10','1985'), ('98765015LP',5500,'Marisol','Quispe','Conductor' ','+591 78910001','2010-10-10','1985'), ('98765016LP',5500,'Mario','Calle','Conductor','+ 591 78910001','2010-10-10','1985'), ('98765017LP',5000,'Agel','Perez','Conductor','+5 91 78910001','2010-10-10','1985'), ('98765018LP',5000,'Angela','Luz','Conductor','+5 91 78910001','2010-10-10','1985'), ('98765019LP',5000,'Jose','Choque','Conductor','+ 591 78910001','2010-10-10','1985'), ('98765020LP',5000,'Jesus','Quispe','Conductor',' +591 78910001','2010-10-10','1985'); </pre>
conductores	<pre> create table conductores (id_conductor integer identity primary key not null, ci_empleado varchar(50) not null, placa_camion varchar(50)not null, foreign key (ci_empleado) references empleados (ci_empleado), foreign key (placa_camion) references camiones (placa_camion)); go insert into conductores(ci_empleado, placa_camion) values ('98765011LP','QWERTY1201'), ('98765012LP','QWERTY1202'), ('98765013LP','QWERTY1203'), </pre>

	<pre> ('98765014LP','QWERTY1204'), ('98765015LP','QWERTY1205'), ('98765016LP','QWERTY1206'), ('98765017LP','QWERTY1207'), ('98765018LP','QWERTY1208'), ('98765019LP','QWERTY1209'), ('98765020LP','QWERTY1210'); </pre>
oficinas	<pre> create table oficinas (id_oficina integer identity primary key not null, ci_empleado varchar(50) not null, foreign key (ci_empleado) references empleados (ci_empleado),); go insert into oficinas(ci_empleado)values ('98765001LP'), ('98765002LP'), ('98765003LP'), ('98765007LP'), ('98765008LP'), ('98765009LP'), ('98765010LP'); </pre>
puesto_ventas	<pre> create table puesto_ventas (id_puesto_ventas integer identity primary key not null, direccion_puesto varchar(100)not null, ci_empleado varchar(50) not null, placa_camion varchar(50)not null, foreign key (ci_empleado) references empleados (ci_empleado), foreign key (placa_camion) references camiones (placa_camion)); go insert into puesto_ventas(direccion_puesto, ci_empleado, placa_camion) values ('Av. Ballivian #530','98765004LP','QWERTY1208'), ('Av. 16 de julio #2','98765005LP','QWERTY1209'), ('Av. La Paz #700','98765006LP','QWERTY1210'); </pre>
pollos	<pre> create table pollos (</pre>

```

id_carga_pollos varchar(50) primary key not
null,
costo integer not null, --medianos, pequeños,
grandes
placa_camion varchar(50)not null,
foreign key (placa_camion) references
camiones (placa_camion)
);

```

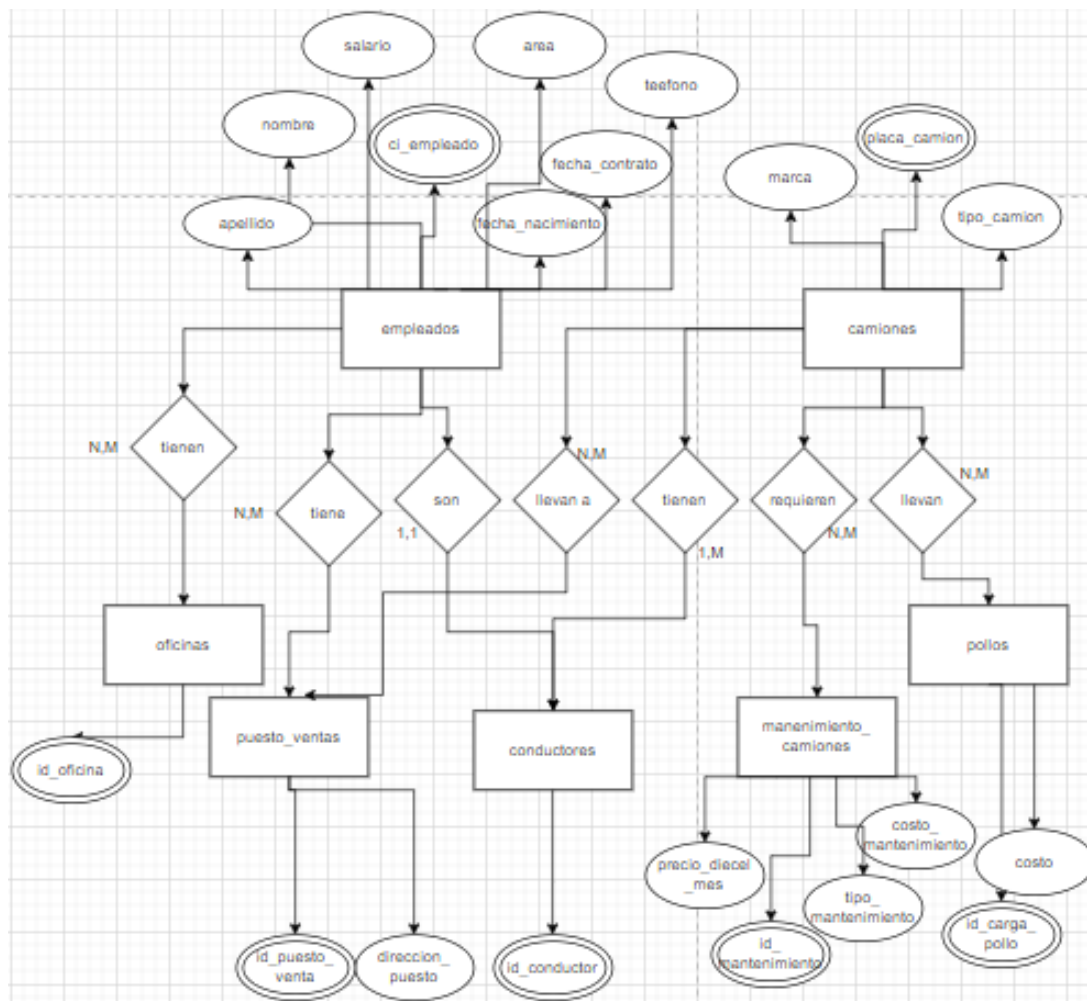
go

```

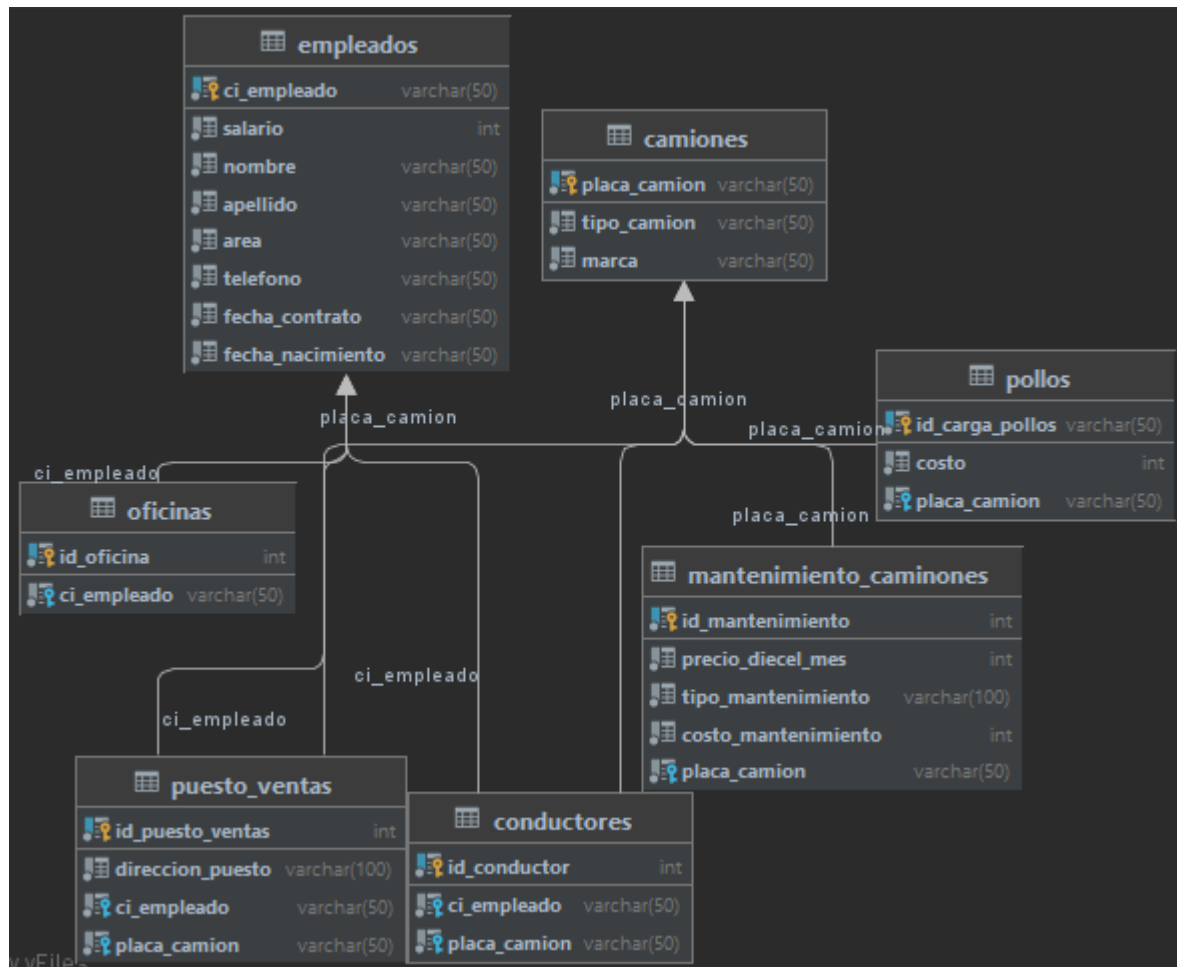
insert into pollos(id_carga_pollos, costo,
placa_camion) values
('A500A1',30000,'QWERTY1201'),
('B500A2',25000,'QWERTY1202'),
('B500A3',27000,'QWERTY1203');

```

2.3.2 Modelo entidad relación de la Base de Datos ER.



2.3.3 Modelo lógico de la Base de Datos.



2.4. Búsquedas, funciones, vistas, etc.

<p>vista: tabla que muestra a aquellos que falta pagar</p>	<pre> create view se_pago as select emp.ci_empleado,emp.nombre,emp.apellido,emp.area,emp.salario,pago= case when emp.salario <=6000 then 'PAGADO' when emp.salario >6000 then 'FALTA PAGAR' else 'indefinido' end from empleados as emp go select * from se_pago; </pre>
---	--

	<table><tr><th>area</th><th>salario</th><th>pago</th></tr><tr><td>Administracion</td><td>8500</td><td>FALTA PAGAR</td></tr><tr><td>Administracion</td><td>8500</td><td>FALTA PAGAR</td></tr><tr><td>Ventas</td><td>5000</td><td>PAGADO</td></tr><tr><td>Ventas</td><td>5000</td><td>PAGADO</td></tr><tr><td>Ventas</td><td>5000</td><td>PAGADO</td></tr><tr><td>Ventas</td><td>5000</td><td>PAGADO</td></tr><tr><td>Contabilidad</td><td>7000</td><td>FALTA PAGAR</td></tr></table>	area	salario	pago	Administracion	8500	FALTA PAGAR	Administracion	8500	FALTA PAGAR	Ventas	5000	PAGADO	Ventas	5000	PAGADO	Ventas	5000	PAGADO	Ventas	5000	PAGADO	Contabilidad	7000	FALTA PAGAR
area	salario	pago																							
Administracion	8500	FALTA PAGAR																							
Administracion	8500	FALTA PAGAR																							
Ventas	5000	PAGADO																							
Ventas	5000	PAGADO																							
Ventas	5000	PAGADO																							
Ventas	5000	PAGADO																							
Contabilidad	7000	FALTA PAGAR																							
vista: tabla que muestra a aquellos camiones que requirieron mantenimiento	<pre>create view se_hizo as select mc.id_mantenimiento,mc.tipo_mantenimiento,mc.placa_camion,se_requirio= case when mc.tipo_mantenimiento != 'Normal' then 'Se requirio' else 'No se requirio' end from mantenimiento_caminones as mc go select * from se_hizo;</pre> <table><tr><th>placa_camion</th><th>se_requirio</th></tr><tr><td>QWERTY1201</td><td>Se requirio</td></tr><tr><td>QWERTY1202</td><td>Se requirio</td></tr><tr><td>QWERTY1203</td><td>No se requirio</td></tr><tr><td>QWERTY1204</td><td>Se requirio</td></tr><tr><td>QWERTY1205</td><td>No se requirio</td></tr></table>	placa_camion	se_requirio	QWERTY1201	Se requirio	QWERTY1202	Se requirio	QWERTY1203	No se requirio	QWERTY1204	Se requirio	QWERTY1205	No se requirio												
placa_camion	se_requirio																								
QWERTY1201	Se requirio																								
QWERTY1202	Se requirio																								
QWERTY1203	No se requirio																								
QWERTY1204	Se requirio																								
QWERTY1205	No se requirio																								
función: muestra la cantidad de mantenimientos realizados	<pre>create or alter function mantenimientos_realizados() returns integer begin declare @resultado integer; set @resultado=(select count(mc.tipo_mantenimiento) from mantenimiento_caminones as mc where mc.tipo_mantenimiento != 'Normal') return @resultado end go print dbo.mantenimientos_realizados();</pre>																								

función: muestra el gasto realizado con la compra de los pollos	<pre> create or alter function gasto_compra() returns integer begin declare @respuesta integer; set @respuesta=(select sum(po.costo) from pollos as po) return @respuesta; end go print dbo.gasto_compra(); </pre>
---	---

3. Usabilidad

3.1. Imágenes acerca del uso del sistema.



VentanaVerRegistros

	ci_employado	salario	nombre	apellido	area	telefono	fecha_contrato	fecha_nacimiento
▶	123CB	5000	MARIA	PEREZ	TRANSPORTE	+501 12345678	11-1-2020	10-10-1985
	123LP	5000	PEDRO	DELGADO	VENTAS	+500 12345678	10-1-2020	1-1-1990
	123SCZ	5000	MARIO	SANCHEZ	TRANSPORTE	+502 12345678	12-1-2020	5-5-1995

CI:

Nombre:

Apellido:

Telefono:

Dirección:

Guardar

Eliminar

3.2. Video acerca del uso del sistema.

<https://youtu.be/FXqW24d7VS4>

4. Conclusión

Se logró crear satisfactoriamente en lo que cabe un sistema factible y funcional en el cual podemos servirnos de la base de datos para tener un mejor control a la hora de las ventas compras y clientes y los gatos de una agencia de pollos.