

***CODER HOUSE***



Torres Rodrigo Emanuel

# Gestión de Servicios Web



4 de septiembre de 20XX • Tu ciudad, Provincia

Comision N<sup>a</sup>50055

# Definición de Modelo de Negocio:

El modelo de negocio Utilizado es una empresa que se dedica a crear y brindar servicios de creación de Software Back-End

# Objetivo

Cada día en la agencia se reciben muchos datos para ejercer el servicio de desarrollo y gestión de software Back end. El Objetivo será ordenar los principales datos recibidos en la BbDd, para proporcionar información útil para organizar al equipo y Poder tomar decisiones sobre los servicios ofrecidos. Los datos que principalmente van a ordenarse van a ser “Información de los clientes(Id\_cliente,Nombre,Correo,Nº Pedido)”, “Información de trabajos realizados a cada cliente(id\_pedido,Nº pedido, nombre-servicio,descripcion\_pedido,precio,tiempo\_elaboracion, fecha)” y “Cantidad de personas por pedido (id\_Servicio,nombre\_pedido,area\_asignada,personas\_asignadas)” . Con esto se podrá mejorar la gestión de los equipos y pulir los procesos necesarios para la oferta del servicio

# Objetivos secundarios

Se Busca Conseguir los siguientes Objetivos secundarios:

- Organizar las áreas

Organizar la gestión de Clientes

- Organizar la elaboración de Presupuestos

- Generar un mejor flujo de trabajo

- Gestionar los procesos de cada Servicio Ofrecido

- Administrar de manera efectiva los recursos para cumplir con la mayor cantidad de solicitudes posible

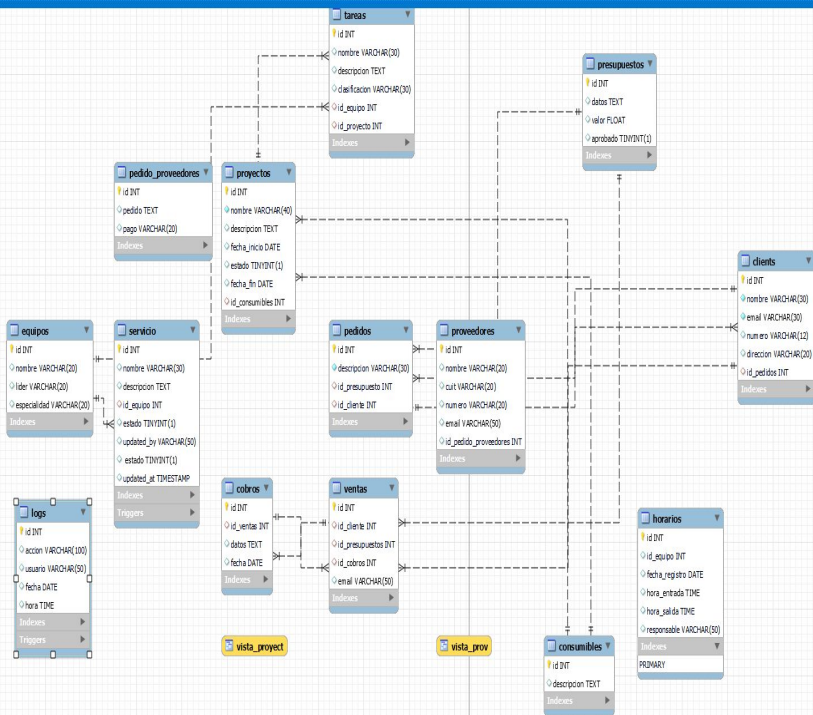
- Tener Registro de los clientes Frecuentes de la Agencia

- Generar una BBDD más compleja que permita recibir de manera ordenada y rápida los datos necesarios

# Necesidad que buscamos cubrir:

La necesidad que se busca Cubrir es la falta de datos para la correcta gestión y el correcto ordenamiento general de las distintas áreas encargadas, de los clientes, de los proveedores y de los proyectos

## Diagrama entidad-Relación:



# Detalle técnico de las tablas:

Tabla	Campo Abreviado	Nombre de Campo Completo	Clave Primaria	Clave Foranea	Tipo de dato
clientes	id_c	id_cliente	PK		INT PRIMARY KEY AUTO_INCREMENT NOT-NULL
	nom_c	nombre_cliente			VARCHAR
	em_c	email_cliente			VARCHAR
	num_c	numero_cliente			VARCHAR
	Dic_c	Direccion_cliente			VARCHAR
pedidos	id_ped_c	id_pedido_cliente	PK		INT PRIMARY KEY AUTO_INCREMENT NOT-NULL
	ped_des	pedido_descripcio			VARCHAR
	id_pres	id_presupuesto		Fk	INT
	dt_pres	datos_presupuest			VARCHAR
	id_cliente			FK	INT
Presupuesto	id_pres	id_presupuesto	PK		INT NOT NULL
	dt_pres	datos_presupuest			TEXT
	ok_c	aprovacion_client			Boolean
	v_pres	valor_presupuest			VARCHAR
Proveedores	Id_prov	Id_proveedores	PK		INT PRIMARY KEY AUTO_INCREMENT NOT-NULL
	nom_prov	nombre_proveedor			VARCHAR
	num_cont_prov	numero_contacto_proveedor			INT
	ct_prov	cuit_proveedor			varchar
	e_prov	email_proveedor			VARCHAR
Pedidos a proveedores	id_ped_prov	id_pedidos_a_proveedores		FK	INT
	ped_re_prov	pedidos_realizados			text
	pa_re	pagos_realizados			varchar



# Detalle técnico de las tablas:

Tareas	Id_T	id_Tareas	PK	BIG INT PRIMARY KEY AUTO_INCREMENT NOT-NUL	
	t_nom	Tarea_nombre			varchar
	t_des	Tarea_descripcion			TEXT
	t_t_asig	Tarea_tiempo_asignado			VARCHAR
	cla_t	Clasificacion_de_tarea			VARCHAR
	id_Eq	Id_Equipo		FK	INT
Equipos de Desarrollo	id_Eq	Id_Equipo	PK	INT PRIMARY KEY AUTO_INCREMENT NOT-NUL	
	nom_Eq	nombre_Equipo			VARCHAR
	l_Eq	Lider_Equipo			VARCHAR
	espe_Eq	especialidad_equipo			VARCHAR
Proyectos de Software	Id_Pr	Id_Proyecto	PK	INT PRIMARY KEY AUTO_INCREMENT NOT-NUL	
	des_Pr	Descripcion_Proy			TEXT
	Ini_Pr	Inicio_Proyecto			DATE
	Est_Pr	Estado_Proyecto			VARCHAR
	F_Fin_Pr	Fecha_Fin_Proyec			DATE
	Id_cons_pr	Id_consumibles_p		FK	INT
	Id_cons_pr	Id_consumibles_p	PK		INT
	Des_cons_pr	Descripcion_Cons			TEXT
Consumibles					

# Detalle técnico de las tablas:

Servicios de Mantenimiento	id_s	id_servicio	PK		INT PRIMARY KEY AUTO_INCREMENT NOT NULL
	desc_s	descripcion_servi			TEXT
	id_eq	id_Equipo			INT
	f_ini_s	F_echa_inicio_ser			DATE
	est_s	estado_servicio			BOOLEAN
	f_fin_s	fin_servicio			DATE
Ventas	id_V	id_ventas	PK		INT PRIMARY KEY AUTO_INCREMENT NOT NULL
	id_c	id_cliente	PK		INT
	em_c	email_cliente			VARCHAR
	id_pres	id_presupuesto		FK	INT
	id_cob	id_cobro		FK	INT
Cobros	id_cob	id_cobro	PK		INT PRIMARY KEY AUTO_INCREMENT NOT NULL
	id_v	id_ventas		FK	INT
	d_cob	Datos_cobro			VARCHAR
	f_cob	Fecha_de_cobro			DATE
Logs	id	id			INT PRIMARY KEY AUTO_INCREMENT NOT NULL
	acot	accion			VARCHAR
	user	usuario			VARCHAR
	fech	fecha			DATE
	tim	hora			TIME
Horarios	id	id	PK		INT PRIMARY KEY AUTO_INCREMENT NOT NULL
	id_eq	id_equipo		FK	INT
	f_d_r	fecha_de_registro			DATE
	ho_e	hora_de_entrada			TIME
	ho_s	hora_de_salida			TIME
	resp	responsable			VARCHAR

# Herramientas técnicas utilizadas:

Google presentaciones: Para realizar el documento y las presentaciones del proyecto

Excel: Para realizar los informes de las tablas

Mysql workbench: Se usó para realizar las bases de datos, las vistas, funciones, etc.

Draw.io: Se usa para realizar el diagrama entidad relacion base para todo el proyecto

ONLINE CHARTLE: Para la realización del gráfico de ventas

# Vistas de la DB

Vista 'vista\_clients':

- Objetivo:

- Proporcionar una vista más legible o simplificada de la información de clientes.
- Utiliza alias para cambiar los nombres de las columnas para hacerlos más descriptivos y amigables.

Vista 'vista\_horarios':

Objetivo:

- Ofrecer una vista más enfocada en los datos de horarios, extrayendo solo las columnas relevantes.
- Proporciona un conjunto de datos que se centra en la información específica de los horarios de los equipos.

# Vista de la DB

Vista 'vista\_presu':

- Objetivo:

- Simplificar la presentación de datos de presupuestos, mostrando solo las columnas necesarias.
- Proporciona una vista centrada en los datos clave relacionados con los presupuestos.

Vista 'vista\_proyect':

- Objetivo:

- Suministrar una vista más clara de la información de proyectos, extrayendo solo las columnas relevantes.
- Facilita la consulta y presentación de datos específicos sobre proyectos.



Vista 'vista\_prov':

- Objetivo:

- Presentar una vista más simple y centrada en los datos de proveedores, utilizando alias para nombres de columnas.
- Proporciona una vista fácilmente comprensible de la información de los proveedores.

# Vistas de la DB

vista\_presu:

Result Grid   Filter Rows: <input type="text"/>			
	d_pres	v_pres	ok_pres
▶	Presupuesto1	1000	1
	Presupuesto2	1500	0
	Presupuesto3	1200	1
	Presupuesto4	800	0
	Presupuesto5	2000	1
	Presupuesto6	1800	0
	Presupuesto7	900	1
	Presupuesto8	1100	0
	Presupuesto9	1300	1
	Presupuesto10	1600	0

# Vistas de la DB

vista\_proyect:

	nom_proy	des_proy	f_proy	est_proy	f_fin_proy
►	Proyecto1	Descripción1	2023-01-01	1	2023-03-31
	Proyecto2	Descripción2	2023-02-01	0	2023-04-30
	Proyecto3	Descripción3	2023-03-01	1	2023-05-31
	Proyecto4	Descripción4	2023-04-01	0	2023-06-30
	Proyecto5	Descripción5	2023-05-01	1	2023-07-31
	Proyecto6	Descripción6	2023-06-01	0	2023-08-31
	Proyecto7	Descripción6	2023-07-01	1	2023-09-30
	Proyecto8	Descripción8	2023-08-01	0	2023-10-31
	Proyecto9	Descripción9	2023-09-01	1	2023-11-30
	Proyecto10	Descripción10	2023-10-01	0	2023-12-31

# Vistas de la DB

vista\_clientes:

Result Grid   Filter Rows:  Export:  Wrap Cell Content:

	nom_c	em_c	num_c	Dir_c	id_ped_c
►	Cliente1	cliente1@email.com	1234567890	Nueva Direccion	1
	Cliente2	cliente2@email.com	2345678901	Calle B, Ciudad Y	2
	Cliente3	cliente3@email.com	3456789012	Calle C, Ciudad Z	3
	Cliente4	cliente4@email.com	4567890123	Calle D, Ciudad W	4
	Cliente5	cliente5@email.com	5678901234	Calle E, Ciudad V	5
	Cliente6	cliente6@email.com	6789012345	Calle F, Ciudad U	6
	Cliente7	cliente7@email.com	7890123456	Calle G, Ciudad T	7
	Cliente8	cliente8@email.com	8901234567	Calle H, Ciudad S	8
	Cliente9	cliente9@email.com	9012345678	Calle I, Ciudad R	9
	Cliente10	cliente10@email.com	0123456789	Calle J, Ciudad Q	10



# Vistas de la DB

vista\_prov:

	nom_prov	cu_prov	num_prov	em_prov
►	Proveedor1	CUIT1	1234567890	proveedor1@email.com
	Proveedor2	CUIT2	2345678901	proveedor2@email.com
	Proveedor3	CUIT3	3456789012	proveedor3@email.com
	Proveedor4	CUIT4	4567890123	proveedor4@email.com
	Proveedor5	CUIT5	5678901234	proveedor5@email.com
	Proveedor6	CUIT6	6789012345	proveedor6@email.com
	Proveedor7	CUIT7	7890123456	proveedor7@email.com
	Proveedor8	CUIT8	8901234567	proveedor8@email.com
	Proveedor9	CUIT9	9012345678	proveedor9@email.com
	Proveedor10	CUIT10	0123456789	proveedor10@email.com

# Triggers de la DB

## Trigger Before 'before\_actualizar\_servicio':

- Descripción General: Este trigger se activa antes de realizar una actualización en la tabla 'servicio'.
- Objetivo/Beneficio: Asigna el usuario actual que realiza la actualización, la marca de tiempo actual y actualiza el campo 'estado'.
- Datos/Tablas Manipulados: Se trabaja con la fila que está siendo actualizada en la tabla 'servicio'.

## Trigger After 'after\_actualizar\_servicio':

- Descripción General: Se activa después de actualizar un registro en la tabla 'servicio'.
- Objetivo/Beneficio: Registra en la tabla 'logs' la acción de actualización, el usuario, la fecha y la hora.
- Datos/Tablas Manipulados: Se insertan datos en la tabla 'logs'

# Triggers de la DB

## Trigger Before 'before\_insertar\_producto':

- Descripción General: Se activa antes de insertar un nuevo registro en la tabla 'logs'.
- Objetivo/Beneficio: Asigna el usuario actual, la fecha y la hora al insertar un nuevo producto en la tabla 'logs'.
- Datos/Tablas Manipulados: Se trabaja con la fila que está siendo insertada en la tabla 'logs'.

## Trigger After 'after\_insertar\_producto':

- Descripción General: Se activa después de insertar un nuevo registro en la tabla 'logs'.
- Objetivo/Beneficio: Registra en la tabla 'logs' la acción de inserción de un nuevo producto, el usuario, la fecha y la hora.
- Datos/Tablas Manipulados: Se insertan datos en la tabla 'logs'.

# Funciones de la DB y su objetivo

- Función: insertar\_cliente
  - Descripción: Inserta un nuevo cliente en la tabla 'clients'.
  - Parámetros:
    - - p\_nombre: Nombre del cliente (VARCHAR).
    - - p\_email: Dirección de correo electrónico del cliente (VARCHAR).
    - - p\_numero: Número de teléfono del cliente (VARCHAR).
    - - p\_direccion: Dirección del cliente (VARCHAR).
    - - p\_id\_pedidos: ID de pedidos asociado al cliente (INT).
  - Retorna: El nuevo ID del cliente insertado.

# Funciones de la DB y sus objetivos

Función: insertar\_proveedor

Descripción: Inserta un nuevo proveedor en la tabla 'proveedores'.

Parámetros:

--p\_nombre: Nombre del proveedor (VARCHAR).

--p\_cuit: Número de CUIT del proveedor (VARCHAR).

--p\_numero: Número de teléfono del proveedor (VARCHAR).

--p\_email: Dirección de correo electrónico del proveedor (VARCHAR).

--p\_id\_pedido\_proveedores: ID de pedidos asociado al proveedor (INT).

--Retorna: El nuevo ID del proveedor insertado.

# Stored Procedure de la DB

## OrdenarTabla

Descripción: Este procedimiento ordena una tabla según un campo y un criterio de ordenamiento dados.

Parámetros:

- tabla\_nombre: Nombre de la tabla que se va a ordenar (VARCHAR).
- campo\_orden: Campo por el cual se va a ordenar la tabla (VARCHAR).
- ordenamiento: Criterio de ordenamiento (ASC o DESC) (VARCHAR).
- Objetivo/Beneficio: Proporciona una manera flexible de ordenar tablas según diferentes criterios y direcciones.
- Datos/Tablas: Puede manipular cualquier tabla que se pase como argumento.

# Stored Procedure de la DB

## ActualizarRegistros

Descripción: Este procedimiento actualiza registros en una tabla según una condición dada.

Parámetros:

- tabla\_nombre: Nombre de la tabla que se va a actualizar (VARCHAR).
- campo\_actualizar: Campo que se va a actualizar en la tabla (VARCHAR).
- nuevo\_valor: Nuevo valor que se asignará al campo (VARCHAR).
- condicion: Condición que determina qué registros actualizar (VARCHAR).
- Objetivo/Beneficio: Proporciona una forma dinámica de actualizar registros en una tabla basada en condiciones específicas.
- Datos/Tablas: Puede manipular cualquier tabla que se pase como argumento.

## Uso de DCL:

En este archivo sql me dediqué a la creación de diferentes usuarios, con contraseña, dándole diferentes permisos por sobre todas las tablas para poder Administrar, leer y editar diferentes tablas. Para así permitir la visualización y dinamizar el uso de los datos dentro de la empresa



# Vista de código DCL:

```
-- Crear usuarios con diferentes permisos

-- Usuario 1
CREATE USER 'roDRIADM'@'localhost' IDENTIFIED BY '14-07-2023';
-- Conceder permisos de lectura, escritura y eliminación sobre todas las tablas
GRANT SELECT, INSERT, DELETE ON proyect_schema.* TO 'roDRIADM'@'localhost';

-- Usuario 2
CREATE USER 'user121'@'localhost' IDENTIFIED BY 'password1';
GRANT SELECT ON proyect_schema.* TO 'user121'@'localhost';

-- Usuario 3
CREATE USER 'user12'@'localhost' IDENTIFIED BY 'password2';
GRANT INSERT ON proyect_schema.* TO 'user12'@'localhost';

-- Usuario 4
CREATE USER 'user3'@'localhost' IDENTIFIED BY 'password3';
GRANT UPDATE ON proyect_schema.* TO 'user3'@'localhost';

-- Usuario 5
CREATE USER 'user4'@'localhost' IDENTIFIED BY 'password4';
GRANT DELETE ON proyect_schema.* TO 'user4'@'localhost';
```

## Vista de código DCL:

```
-- Usuario 5
CREATE USER 'user4'@'localhost' IDENTIFIED BY 'password4';
GRANT DELETE ON proyect_schema.* TO 'user4'@'localhost';

-- Usuario 6
CREATE USER 'user56'@'localhost' IDENTIFIED BY 'password5';
GRANT SELECT, INSERT ON proyect_schema.* TO 'user56'@'localhost';

-- Usuario 7
CREATE USER 'user68'@'localhost' IDENTIFIED BY 'password6';
GRANT SELECT, UPDATE ON proyect_schema.* TO 'user68'@'localhost';

-- Usuario 8
CREATE USER 'user73'@'localhost' IDENTIFIED BY 'password7';
GRANT SELECT, DELETE ON proyect_schema.* TO 'user73'@'localhost';

-- Usuario 9
CREATE USER 'user8'@'localhost' IDENTIFIED BY 'password8';
GRANT ALL PRIVILEGES ON proyect_schema.* TO 'user8'@'localhost';
-- Modifico la contraseña de algunos usuarios

ALTER USER 'user73'@'localhost' IDENTIFIED BY 'NEW-PASS--WORD';
ALTER USER 'user8'@'localhost' IDENTIFIED BY 'NEW-PASS--21';
ALTER USER 'user68'@'localhost' IDENTIFIED BY 'USER68-NEW-PASS';
```

# Vista de código DCL:

```
-- RENOMBRO ALGUNOS USUARIOS

RENAME USER 'user56'@'localhost' TO 'newuser56'@'localhost';
RENAME USER 'user73'@'localhost' TO 'user73new'@'localhost';
RENAME USER 'user12'@'localhost' TO 'newnameuser12'@'localhost';

-- Borro algunos usuarios
DROP USER 'user121'@'localhost';

use mysql;

SHOW GRANTS FOR 'newnameuser12'@'localhost';

-----Tarea solicitada en

-- CREO AL USUARIO QUE SOLO TIENE PERMISO DE LECTURA -----
CREATE USER 'user43'@'localhost' IDENTIFIED BY '45asr64r';
-- entrego el permiso de lectura -----
GRANT SELECT ON proyect_schema.* TO 'user43'@'localhost';

-- CREO AL USUARIO QUE SOLAMENTE TIENE PERMISO DE LECTURA Y MODIFICACION DE DATOS -----
CREATE USER 'user23'@'localhost' IDENTIFIED BY 'ASR4455E2';
-- entrego los permisos de lectura(Select) , de insercion (insert) y modificacion(update)
GRANT SELECT, UPDATE, INSERT ON proyect_schema.* TO 'user23'@'localhost';
```

## Uso de TCL:

Use este lenguaje en mi proyecto para tener un control en el flujo de los datos que se agregan o modifican en la Base de Datos con un guardado de la carga de clientes y proveedores para comprobación de datos y luego de revisarlos un commit para definir el cambio en estos datos

# Vista de TCL:

```
-- Desactivo el autocommitt
SELECT @@autocommit;
SET AUTOCOMMIT = 0;

-- agrego a la tabla clientes un lote de clientes y agrego un savepoint--- Trabajo
-- Establecer el punto de guardado inicial
SAVEPOINT inicio;

-- Generar clientes con datos aleatorios
INSERT INTO clients (id, nombre, email, numero, direccion)
VALUES
    (11, 'Cliente11', 'cliente1@example.com', '123456789', 'Calle A, Ciudad'),
    (12, 'Cliente12', 'cliente2@example.com', '987654321', 'Calle B, Ciudad');
-- Puedes agregar más filas con datos aleatorios aquí...

-- Establecer el punto de guardado a la mitad del proceso
SAVEPOINT mitad;

-- Continuar generando clientes con datos aleatorios
INSERT INTO clients (id, nombre, email, numero, direccion)
VALUES
    (13, 'Cliente13', 'cliente3@example.com', '555555555', 'Calle C, Ciudad'),
    (14, 'Cliente14', 'cliente4@example.com', '111111111', 'Calle D, Ciudad');
```

# Vista de TCL:

```
SAVEPOINT mitad;
```

```
-- Continuar generando clientes con datos aleatorios
```

```
INSERT INTO clients (id, nombre, email, numero, direccion)
```

```
VALUES
```

```
(13, 'Cliente13', 'cliente3@example.com', '55555555', 'Calle C, Ciudad'),
```

```
(14, 'Cliente14', 'cliente4@example.com', '11111111', 'Calle D, Ciudad');
```

```
-- Establecer el punto de guardado al final del proceso
```

```
SAVEPOINT fin;
```

```
-- Elimino dos registros de la tabla clientes
```

```
START TRANSACTION;
```

```
DELETE FROM clients
```

```
WHERE id = 2 AND id = 3;
```

```
-- genero una transaccion
```

```
START TRANSACTION;
```

```
UPDATE clients SET
```

```
nombre = 'gerardo'
```

```
WHERE id =1;
```

# Vista de TCL:

```
-- Verifico los cambios realizados en la tabla clientes
SELECT * FROM clients;

-- ROLLBACK TO mitad;

-- COMMIT;

-- INICIO UNA TRANSACCION EN LA SEGUNDA TABLA----- Trabajo segunda Tabla
-- Establecer el punto de guardado inicial
SAVEPOINT inicio2;

-- Generar proveedores con datos aleatorios
INSERT INTO proveedores (id, nombre, cuit, numero, email)
VALUES
    (11, 'Proveedor11', '12345678901', '987654321', 'proveedor11@example.com'),
    (12, 'Proveedor12', '98765432109', '143456789', 'proveedor12@example.com'),
    (13, 'Proveedor13', '98765432459', '593456789', 'proveedor13@example.com'),
    (14, 'Proveedor14', '98765432789', '223456789', 'proveedor14@example.com');
-- Puedes agregar más filas con datos aleatorios aquí...

-- Establecer el punto de guardado a la mitad del proceso
SAVEPOINT mitad2;
```

# Vista TCL:

```
-- Establecer el punto de guardado a la mitad del proceso
```

```
SAVEPOINT mitad2;
```

```
-- Continuar generando proveedores con datos aleatorios
```

```
INSERT INTO proveedores (id, nombre, cuit, numero, email)
```

```
VALUES
```

```
(15, 'Proveedor15', '5555555555', '11111111', 'proveedor15@example.com'),
```

```
(16, 'Proveedor16', '1111111111', '555555546', 'proveedor16@example.com'),
```

```
(17, 'Proveedor17', '1111111123', '555555555', 'proveedor17@example.com'),
```

```
(18, 'Proveedor18', '98765444109', '123456778', 'proveedor18@example.com');
```

```
-- Puedes agregar más filas con datos aleatorios aquí...
```

```
-- Establecer el punto de guardado al final del proceso
```

```
SAVEPOINT fin2;
```

```
RELEASE SAVEPOINT mitad2;
```



# Backup de la base de datos

Se realizo un Backup del proyecto final de la base de datos, en este backup se guardaron las tablas, triggers, store procedures y funciones de el project\_schema.

LAS TABLAS INCLUIDAS DEL PROYECTO SON:

TABLA CLIENTES, TABLA COBROS, TABLA CONSUMIBLES, TABLA EQUIPOS, TABLA EQUIPOS, TABLA HORARIOS, TABLA HORARIOS, TABLA LOGS, TABLA PEDIDO\_PROVEEDORES, TABLA PEDIDOS, TABLA PRESUPUESTOS, TABLA PROVEEDORES

TABLA PROYECTOS, TABLA SERVICIO, TABLA TAREAS, TABLA VENTAS

# Gráfico de ventas a lo largo del tiempo

Ventas anuales

# ventas a conseguir # ventas conseguidas

