

**Natalia Rivas**  
**Propiedades**

## **Indice:**

1. [Justificación y objetivo del proyecto](#)
2. [DER \(Diagrama Entidad Relación\)](#)
3. [Vistas](#)
  - a. [Vista 1](#)
  - b. [Vista 2](#)
  - c. [Vista 3](#)
  - d. [Vista 4](#)
  - e. [Vista 5](#)
  - f. [Vista 6](#)
4. [Funciones](#)
  - a. [Funcion 1](#)
  - b. [Funcion 2](#)
5. [Stored Procedures](#)
  - a. [Stored Procedure 1](#)
  - b. [Stored Procedure 2](#)
6. [Triggers](#)
  - a. [Trigger 1](#)
  - b. [Trigger 2](#)

**Justificación del proyecto:****Negocio:**

Natalia Rivas Propiedades es una inmobiliaria nueva en Zona Oeste, que, gracias a su experiencia en otra inmobiliaria, ya tiene clientes fijos para administrar alquileres.

Su responsabilidad es principalmente la administración de alquileres en Tres de Febrero.

**Objetivo:**

Precisan la creación de una base de datos con sus clientes, contratos y propiedades para poder organizarlos y consultarlos de una manera eficaz.

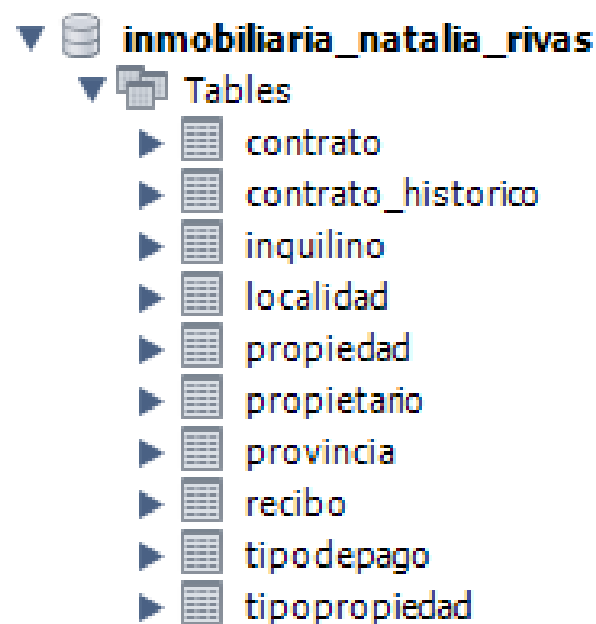
Buscan modernizar su negocio y pasar las cosas del papel a la computadora para ahorrar tiempo verificando información y así, brindar una mejor atención.

## DER (Diagrama Entidad Relación)

Para este modelo se crearon 10 nuevas entidades:

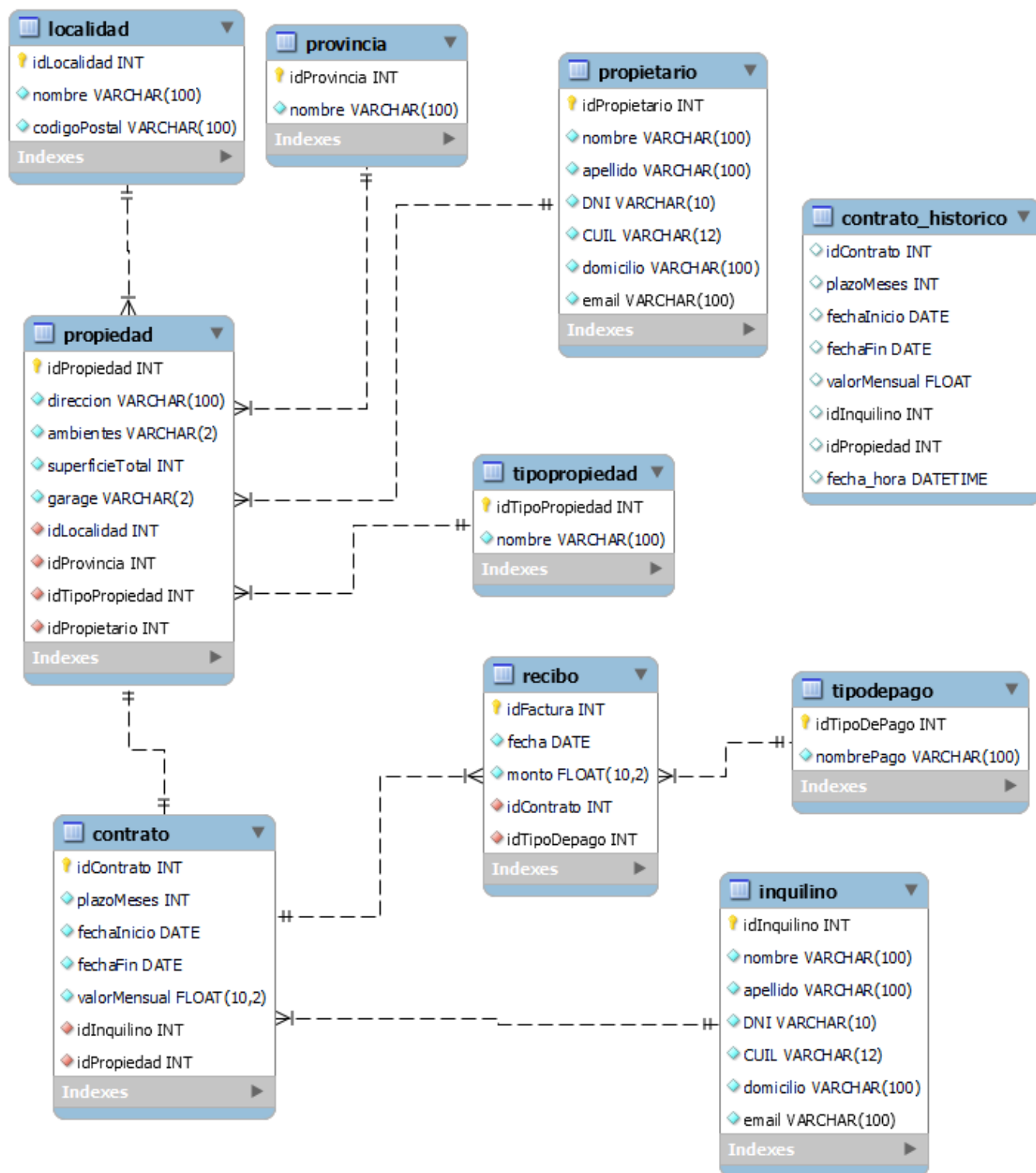
1. Contrato
2. Inquilino
3. Localidad
4. Propiedad
5. Propietario
6. Provincia
7. TipoPropiedad
8. TipoDePago
9. Recibo
10. Contrato\_historico

Se visualizan en el workbench:



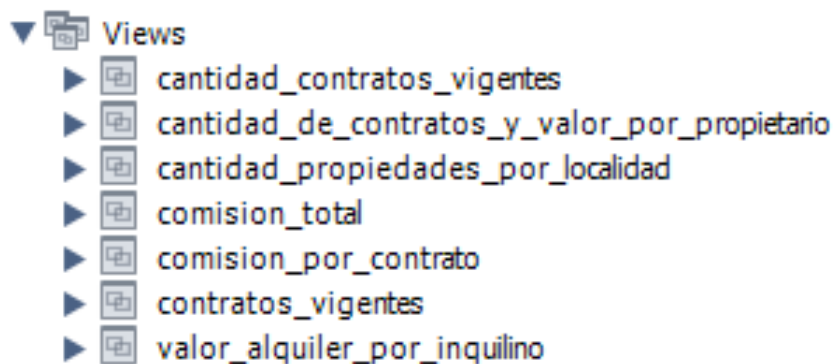


DER generado desde el workbench:



## Vistas:

Fueron creadas 6 vistas:



Estas le permitirán a los usuarios realizar las siguientes consultas:

1. Consultar la cantidad de contratos vigentes.

```
CREATE VIEW cantidad_contratos_vigentes AS
SELECT COUNT(idContrato) AS 'Cantidad Contratos Vigentes'
FROM contrato
WHERE fechaFin > CURRENT_DATE;
```

Resultado:

	Cantidad Contratos Vigentes
▶	16

2. Consultar la cantidad de contratos agrupados por propietario, el valor total que cobra y la cantidad de contratos que tienen a su nombre.

```

CREATE VIEW cantidad_de_contratos_y_valor_por_propietario AS
SELECT CONCAT(pr.nombre, ' ', pr.apellido) AS 'Propietario',
       SUM(c.valorMensual) 'Valor total a cobrar por mes',
       COUNT(c.idContrato) AS 'Cantidad de contratos'
FROM contrato AS c
INNER JOIN propiedad AS p
ON p.idPropiedad = c.idPropiedad
INNER JOIN propietario AS pr
ON pr.idPropietario = p.idPropietario
GROUP BY CONCAT(pr.nombre, ' ', pr.apellido)
ORDER BY CONCAT(pr.nombre, ' ', pr.apellido) ASC;

```

Resultado:

	Propietario	Valor total a cobrar por mes	Cantidad de contratos
►	Annadiana Robertelli	126369.58	1
	Annie Casley	721780.34	2
	Brigid Gerrie	6000.20	1
	Claudine Dowd	194824.00	1
	Clayson Raulstone	36457.13	1
	Deni Haste	35242.00	1
	Dougy Boome	129909.75	1
	Elane O'Cridigan	33041.32	1
	Evangelin Llop	31667.01	2
	Gunther Wynne	587522.88	1
	Kacie Slowgrove	21467.13	1
	Luce Bier	82976.14	1
	Maximilianus Farris	46819.19	1
	Nannie Hammill	43238.27	1
	Newton Dyshart	81652.74	1
	Shelia Mattussevich	32372.66	1
	Tandi Brogi	30506.15	1
	Tracy Rosingdall	48207.37	1
	Valaria Klawi	114541.23	1
	Viviana Emig	178010.70	1

3. Consultar el valor de alquiler que paga cada inquilino, agrupados por el mismo y la dirección completa de la propiedad que alquila.



```

CREATE VIEW valor_alquiler_por_inquilino AS
SELECT CONCAT(i.nombre, ' ', i.apellido) AS 'Inquilino',
       c.valorMensual AS 'Valor a pagar por mes',
       CONCAT(p.direccion, ', ', l.nombre, ', ', pv.nombre) AS 'Propiedad que alquiler'
FROM contrato AS c
INNER JOIN inquilino AS i
ON c.idInquilino = i.idInquilino
INNER JOIN propiedad AS p
ON c.idPropiedad = p.idPropiedad
INNER JOIN localidad AS l
ON p.idLocalidad = l.idLocalidad
INNER JOIN provincia AS pv
ON p.idProvincia = pv.idProvincia
GROUP BY CONCAT(i.nombre, ' ', i.apellido)
ORDER BY CONCAT(i.nombre, ' ', i.apellido) ASC;

```

### Resultado:

	Inquilino	Valor a pagar por mes	Propiedad que alquiler
▶	Alanson Jerams	32372.66	8,941 Melody Point, Caseros, Buenos Aires
	Avery Capstaff	25666.81	2,5 Walton Court, Caseros, Buenos Aires
	Bryanty Kerwin	30506.15	19,6246 Eliot Street, Churruca, Buenos Aires
	Cassie Standbridge	178010.70	10,6 Carey Plaza, Ciudad Jardin , Buenos Aires
	Christoforo Peete	114541.23	9,10652 McCormick Center, Ciudad Jardin , Bue...
	Daniele McNeilley	21467.13	7,32 Straubel Park, Caseros, Buenos Aires
	Donovan Leale	43238.27	5,5 Reinke Alley, Caseros, Buenos Aires
	Gnni Ramberg	48207.37	20,620 Grasskamp Avenue, Caseros, Buenos Ai...
	Hoyt Cherrett	194824.00	13,659 Maryland Court, Ciudadela, Buenos Aires
	Linnie MacCrachen	134257.47	1,875 Little Fleur Park, Caseros, Buenos Aires
	Lucretia Lefley	36457.13	12,215 Fremont Plaza, Ciudadela, Buenos Aires
	Lyssa Petrovykh	126369.58	6,5 Northport Court, Caseros, Buenos Aires
	Morten Quantrell	35242.00	15,2479 Coolidge Crossing, Villa Bosch, Buenos ...
	Parry Chesters	81652.74	11,9 Crownhardt Avenue, Ciudad Jardin , Buen...
	Reiko Ticksall	46819.19	18,169 Eagan Way, Churruca, Buenos Aires
	Teriann Rawdall	33041.32	3,99 Bonner Court, Caseros, Buenos Aires
	Trenna Desborough	82976.14	14,2 South Center, Saenz Peña, Buenos Aires
	Wes Diwell	129909.75	4,31 Delladonna Drive, Caseros, Buenos Aires

4. Consultar la cantidad de propiedades que la inmobiliaria tiene en cada localidad.

```
CREATE VIEW cantidad_propiedades_por_localidad AS
SELECT COUNT(p.idPropiedad) AS 'Cantidad de propiedades',
       l.nombre AS 'Localidad'
FROM propiedad AS p
RIGHT JOIN localidad AS l
ON p.idLocalidad = l.idLocalidad
GROUP BY l.nombre
ORDER BY COUNT(p.idPropiedad) DESC
;
```

Resultado:

	Cantidad de propiedades	Localidad
▶	13	Caseros
	5	Ciudad Jardin
	2	Churruca
	2	Ciudadela
	1	Loma Hermosa
	1	Saenz Peña
	1	Villa Bosch
	1	Villa Raffo
	0	El Libertador
	0	Jose Ingenieros
	0	Martín Coronado
	0	Once de Septie...
	0	Pablo Podesta
	0	Remedios de E...
	0	Santos Lugares

5. Consultar cuanto se cobra por mes de comisión por contrato, que es el 5% del valor del alquiler.

```
CREATE VIEW comision_por_contrato AS
SELECT idContrato,
       FORMAT(valorMensual * 5 /100, 0) AS 'Comision'
FROM contrato
WHERE fechaFin > CURRENT_DATE
GROUP BY idContrato;
```

Resultado:

	idContrato	Comision
▶	1	6,713
	2	1,283
	3	29,376
	4	300
	7	2,162
	10	1,619
	11	5,727
	12	8,901
	13	4,083
	14	1,823
	15	9,741
	16	4,149
	18	29,376
	19	300
	21	1,525
	22	2,410

6. Consultar cuanto se cobra por mes de comisión en total.


```
CREATE VIEW comision_total AS
SELECT COUNT(idContrato) AS 'Total contratos',
       FORMAT(SUM(valorMensual) * 5 /100, 0) AS 'Comision total por mes'
FROM contrato
WHERE fechaFin > CURRENT_DATE;
```

Resultado:

	Total contratos	Comision total por mes
▶	16	109,488

### Funciones:

Fueron creadas 2 funciones:

▼  Functions

- f() traer\_forma\_pago
- f() traer\_monto\_pagado

1. Consultar de que forma fue abonado cada recibo.

```
DELIMITER $$
CREATE FUNCTION traer_forma_pago (p_idTipoDePago INT)
RETURNS VARCHAR(255)
READS SQL DATA
BEGIN
    DECLARE resultado VARCHAR(255);
    SET resultado = (SELECT nombrePago FROM tipoDePago WHERE idTipoDePago = p_idTipoDePago);
    RETURN resultado;
END
$$
```

---

Resultado:

	idFactura	fecha	monto	Forma de pago
▶	1	2022-08-19	7676.43	efectivo
	2	2022-11-05	190943.17	efectivo
	3	2023-05-17	24129.32	efectivo
	4	2022-10-02	117342.62	efectivo
	5	2023-03-18	77732.85	efectivo
	6	2022-12-31	21116.64	cuenta bancaria
	7	2023-03-30	74531.82	cuenta bancaria
	8	2023-04-13	26991.27	cuenta bancaria
	9	2022-10-28	153923.80	cuenta bancaria
	10	2022-07-25	152790.30	cuenta bancaria
	11	2022-07-15	79263.03	cuenta bancaria
	12	2022-09-04	93575.43	efectivo
	13	2022-07-23	180974.53	efectivo
	14	2023-05-03	28840.59	cheque
	15	2023-06-02	66980.40	cheque

2. Consultar si fue pagado el monto total, parcial o extra del alquiler mensual.

```

DELIMITER $$
CREATE FUNCTION traer_monto_pagado(p_id_factura INT)
RETURNS VARCHAR(255)
READS SQL DATA
BEGIN

DECLARE p_monto FLOAT;
DECLARE p_idcontrato INT;
DECLARE p_valorMensual FLOAT;

SET p_monto = (select monto from recibo where idfactura = p_id_factura);
SET p_idcontrato = (select idcontrato from recibo where idfactura = p_id_factura);
SET p_valorMensual = (select valorMensual from contrato where idcontrato= p_idcontrato) ;
CASE
    WHEN p_monto = p_valorMensual THEN RETURN 'MONTO TOTAL';
    WHEN p_monto < p_valorMensual THEN RETURN 'MONTO PARCIAL';
    WHEN p_monto > p_valorMensual THEN RETURN 'MONTO EXTRA';
    ELSE RETURN 'NO TIENE RECIBO';
END CASE;
END
$$

```

---

#### Resultado:

	idFactura	fecha	monto	idContrato	idTipoDepago	Monto Pagado
▶	1	2022-08-19	7676.43	1	1	MONTO PARCIAL
	2	2022-11-05	190943.17	3	1	MONTO PARCIAL
	3	2023-05-17	24129.32	4	1	MONTO EXTRA
	4	2022-10-02	117342.62	6	1	MONTO PARCIAL
	5	2023-03-18	77732.85	8	1	MONTO PARCIAL
	6	2022-12-31	21116.64	2	2	MONTO PARCIAL
	7	2023-03-30	74531.82	5	2	MONTO EXTRA
	8	2023-04-13	26991.27	7	2	MONTO PARCIAL
	9	2022-10-28	153923.80	9	2	MONTO EXTRA
	10	2022-07-25	152790.30	1	2	MONTO EXTRA
	11	2022-07-15	79263.03	2	2	MONTO EXTRA
	12	2022-09-04	93575.43	11	1	MONTO PARCIAL
	13	2022-07-23	180974.53	12	1	MONTO EXTRA
	14	2023-05-03	28840.59	13	3	MONTO PARCIAL
	15	2023-06-02	66980.40	14	3	MONTO EXTRA

#### Stored Procedure:

Fueron creadas 2 stored procedures:

## ▼ Stored Procedures

sp\_contratos\_a\_vencer

sp\_traer\_total\_comision\_por\_mes

1. Consultar la cantidad de dinero que entró en la inmobiliaria como comisión en el mes actual.

```
DELIMITER //
```

```
CREATE PROCEDURE sp_traer_total_comision_por_mes (OUT total FLOAT)
```

```
BEGIN
```

```
    SELECT SUM(valorMensual) * 5 /100 INTO total FROM contrato WHERE fechaFin >= CURRENT_DATE;
```

```
END//
```

```
set @total = 0;
```

```
CALL sp_traer_total_comision_por_mes(@total);
```

```
SELECT ROUND(@total, 2) AS 'Total Comisiones Por Mes';
```

---

Resultado:

	Total Comisiones Por Mes
▶	82983.02

2. Consultar todos los contratos a vencer desde la fecha ingresada hasta los próximos 3 meses.

```
DELIMITER $$
```

```
CREATE PROCEDURE sp_contratos_a_vencer(IN p_fechaFin DATE)
```

```
BEGIN
```

```
    IF p_fechaFin IS NULL THEN
```

```
        SELECT 'Debe ingresar un valor';
```

```
    ELSE
```

```
        IF (SELECT COUNT(*) FROM contrato
```

```
            WHERE fechaFin BETWEEN p_fechaFin AND DATE_ADD(fechaFin, INTERVAL +3 MONTH)) = 0 THEN
```

```
            SELECT 'No hay contratos a vencer';
```

```
        ELSE
```

```
            SELECT * FROM contrato
```

```
            WHERE fechaFin BETWEEN p_fechaFin AND DATE_ADD(p_fechaFin, INTERVAL +3 MONTH);
```

```
        END IF;
```

```
    END IF;
```

```
END $$
```

```
CALL sp_contratos_a_vencer(null);
```

```
CALL sp_contratos_a_vencer('2023-06-07');
```

```
CALL sp_contratos_a_vencer('2028-06-07');
```

Resultado:



	Debe ingresar un valor						
▶	Debe ingresar un valor						

	idContrato	plazoMeses	fechaInicio	fechaFin	valor	idInquilino	idPropiedad
▶	15	36	2020-09-01	2023-08-31	194824.00	13	13
	21	36	2020-09-01	2023-08-31	30506.15	19	19
	22	36	2020-07-01	2023-06-30	48207.37	20	20

	No hay contratos a vencer						
▶	No hay contratos a vencer						

Triggers:

**Fueron creados 2 triggers:**

 trigger\_contrato\_historico  
 comprobar\_plazoMeses

1. Tener un histórico de contratos luego de que se insertan.

Tabla:

```
CREATE TABLE IF NOT EXISTS contrato_historico (
  idContrato INT,
  plazoMeses INT,
  fechaInicio DATE,
  fechaFin DATE,
  valorMensual FLOAT,
  idInquilino INT,
  idPropiedad INT,
  fecha_hora DATETIME
);
```

---

Trigger:

```
DELIMITER //
CREATE TRIGGER trigger_contrato_historico
AFTER INSERT ON contrato
FOR EACH ROW
BEGIN
  INSERT INTO contrato_historico (idContrato, plazoMeses, fechaInicio, fechaFin, valorMensual, idInquilino, idPropiedad, fecha_hora)
  VALUES (NEW.idContrato, NEW.plazoMeses, NEW.fechaInicio, NEW.fechaFin, NEW.valorMensual, NEW.idInquilino, NEW.idPropiedad, NOW());
END//
```

---

Inserción de nuevos datos:

```
INSERT INTO contrato VALUES(NULL, 36, '20200401', '20230331', 56841, 18, 2);
```

Resultado:

	idContrato	plazoMeses	fechaInicio	fechaFin	valorMensual	idInquilino	idPropiedad	fecha_hora
▶	26	36	2020-04-01	2023-03-31	56841	18	2	2023-07-07 21:27:18

2. Evitar que se actualicen plazos negativos en la tabla contrato mostrando un mensaje de error.



```
DELIMITER //
```

```
CREATE TRIGGER comprobar_plazoMeses
```

```
BEFORE UPDATE ON contrato
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE mensaje_error VARCHAR(100);
```

```
    SET mensaje_error = CONCAT ('El plazo no puede ser menor a 0');
```

```
    IF NEW.plazoMeses <= 0 THEN
```

```
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = mensaje_error;
```


```
    END IF;
```

```
END//
```

Update:

```
UPDATE contrato SET plazoMeses = -3 WHERE idContrato = 2;*/
```

Resultado:

	585 21:14:18 UPDATE contrato SET plazoMeses = -3 WHERE idContrato = 2	Error Code: 1644. El plazo no puede ser menor a 0
---	---	---