



Universidad Tecnológica Centroamericana

UNITEC

Informe Final de Proyecto

Presentado por:

Marcela Sofía Tovar 12241172

Douglas Omar Rubi Rivera 12011121

Gabriela Sussette Montes Cruz 12241007

Jose Rafael Lobo Guardado 12241127

Presentado a:

Ing. Julio Sandoval

Clase y Sección:

Teoría de Base de Datos 755

Fecha de Entrega:

Tegucigalpa M.D.C 17 de marzo del 2024

Introducción

Los bienes raíces son activos que han estado presentes durante décadas y se han convertido en una parte fundamental de las carteras de inversión.

Entonces el tener una gestión flexible y eficiente de agentes, compradores, vendedores, propiedades en el mercado y propiedades recientemente vendidas es fundamental. La gestión de estos diversos datos, que a menudo tienen relaciones específicas entre sí, requiere una estructura sólida para una gestión adecuada.

Con estas necesidades en mente, nos influenciamos por la estructura y trabajo con la empresa de una destacada agencia inmobiliaria para desarrollar una base de datos que pudiera adaptarse a las complejidades y desafíos que enfrenta dicha empresa. Los bienes raíces son una inversión sólida con un potencial de ingresos pasivos y apreciación a largo plazo.

la combinación de MySQL y NetBeans Java permite que construyamos una sólida base de datos para gestionar información relacionada con bienes raíces y cumplir el objetivo principal de este avance que es mostrar una visión clara y comprensible sobre cómo diseñar e implementar un modelo de base de datos que satisfaga las necesidades operativas de una agencia inmobiliaria.

Esta documentación servirá para mostrar paso a paso como se ha desarrollado el proyecto tanto sus consultas, mantenimiento y funcionalidad del tal.

Especificación

La aplicación consiste en las operaciones de una oficina de bienes raíces. La oficina necesita realizar un seguimiento de los agentes, compradores, vendedores, propiedades en el mercado y propiedades vendidas recientemente. Esta oficina se centra en viviendas más que en bienes raíces comerciales. La administración de esta agencia inmobiliaria no tiene muchos conocimientos de informática. Se le pide que diseñe la base de datos, la complete con datos de muestra (la administración no le permitirá realizar pruebas con datos en vivo debido a preocupaciones de privacidad) y escriba varias consultas SQL para demostrar el sistema. La oficina de bienes raíces está solicitando aproximadamente 50 propuestas, que evaluará a partir del 12 de abril de 2024.

1. Agentes:

- Cada agente de la agencia debe poder ser registrado en el sistema con su ID, nombre y detalles de contacto.
- Se debe mantener un registro de las propiedades que cada agente está a cargo de gestionar.

2. Compradores:

- Se requiere un registro de los compradores interesados en adquirir propiedades.
- Para cada comprador, se deben almacenar su ID, nombre y detalles de contacto.

3. Vendedores:

- Se necesita un registro de los vendedores que desean poner sus propiedades en el mercado.
- Cada vendedor debe tener su propio perfil con su ID, nombre y detalles de contacto.

4. Propiedades en el mercado:

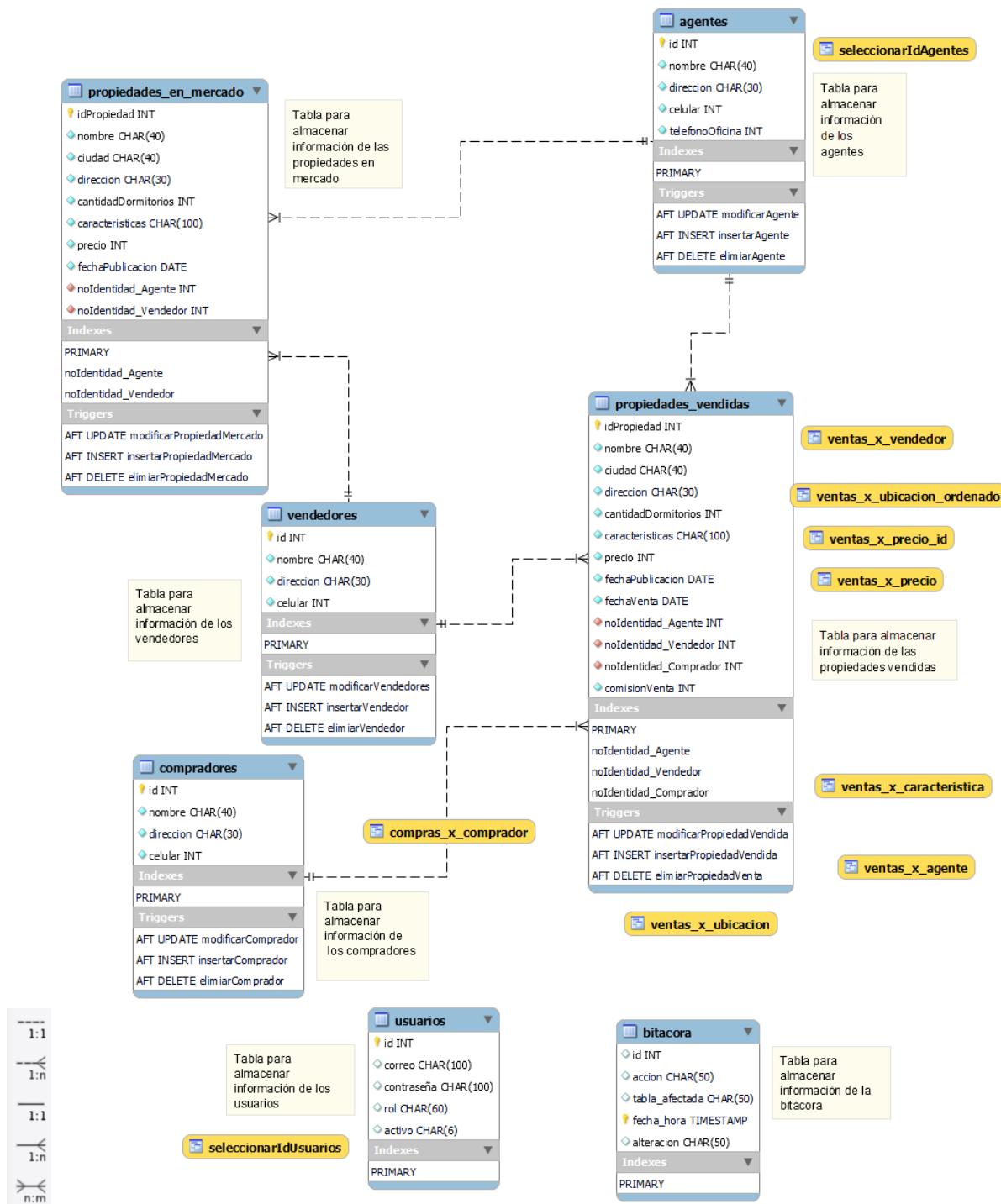
- Deben registrarse todas las propiedades actualmente disponibles para la venta
- Para cada propiedad en el mercado, se deben almacenar detalles como ID, dirección, características (número de dormitorios, baños, tamaño del lote, etc.) y precio de venta/alquiler.
- Cada propiedad debe estar asociada a un vendedor y, opcionalmente, a un agente responsable de su comercialización.

5. Propiedades vendidas recientemente:

- Se requiere un registro de todas las propiedades que han sido vendidas.
- Para cada propiedad vendida, se deben registrar detalles como ID, dirección, características, precio final de venta/alquiler y la fecha de venta/alquiler.
- Cada propiedad vendida debe estar asociada tanto al comprador como al agente que facilitó la transacción.

Documentación y Sistema

- a. Diseño lógico del modelo relacional (DDL) en un DBMS específico, debidamente documentado (comentarios, un punto).



Creación de las tablas:

```
CREATE TABLE IF NOT EXISTS agentes (
    id INT PRIMARY KEY NOT NULL, -- ID único del agente
    nombre CHAR(40) NOT NULL, -- Nombre del agente
    direccion CHAR(30) NOT NULL, -- Dirección del agente
    celular INT NOT NULL, -- Número de celular del agente
    telefonoOficina INT NOT NULL -- Número de teléfono de oficina del agente
);

CREATE TABLE IF NOT EXISTS vendedores (
    id INT PRIMARY KEY NOT NULL, -- ID único del vendedor
    nombre CHAR(40) NOT NULL, -- Nombre del vendedor
    direccion CHAR(30) NOT NULL, -- Dirección del vendedor
    celular INT NOT NULL -- Número de celular del vendedor
);

CREATE TABLE IF NOT EXISTS compradores (
    id INT PRIMARY KEY NOT NULL, -- ID único del comprador
    nombre CHAR(40) NOT NULL, -- Nombre del comprador
    direccion CHAR(30) NOT NULL, -- Dirección del comprador
    celular INT NOT NULL -- Número de celular del comprador
);

CREATE TABLE IF NOT EXISTS propiedades_vendidas (
    idPropiedad INT PRIMARY KEY NOT NULL, -- ID único de la propiedad vendida
    nombre CHAR(40) NOT NULL, -- Nombre de la propiedad
    ciudad CHAR(40) NOT NULL, -- Ciudad de la propiedad
    direccion CHAR(30) NOT NULL, -- Dirección de la propiedad
    cantidadDormitorios INT NOT NULL, -- Cantidad de dormitorios de la propiedad
    caracteristicas CHAR(100) NOT NULL, -- Características de la propiedad
    precio INT NOT NULL, -- Precio de venta de la propiedad
    fechaPublicacion DATE NOT NULL, -- Fecha de publicación de la propiedad
    fechaVenta DATE NOT NULL, -- Fecha de venta de la propiedad
    nolidentidad_Agente INT NOT NULL, -- ID del agente que vendió la propiedad
    nolidentidad_Vendedor INT NOT NULL, -- ID del vendedor de la propiedad
    nolidentidad_Comprador INT NOT NULL, -- ID del comprador de la propiedad
    comisionVenta INT NOT NULL, -- Comisión de venta de la propiedad
    FOREIGN KEY (nolidentidad_Agente) REFERENCES agentes(id) ON DELETE CASCADE ON
    UPDATE CASCADE,
    FOREIGN KEY (nolidentidad_Vendedor) REFERENCES vendedores(id) ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY (nolidentidad_Comprador) REFERENCES compradores(id) ON DELETE
    CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS propiedades_en_mercado (
    idPropiedad INT PRIMARY KEY NOT NULL, -- ID único de la propiedad en mercado
    nombre CHAR(40) NOT NULL, -- Nombre de la propiedad
    ciudad CHAR(40) NOT NULL, -- Ciudad de la propiedad
    direccion CHAR(30) NOT NULL, -- Dirección de la propiedad
    cantidadDormitorios INT NOT NULL, -- Cantidad de dormitorios de la propiedad
    caracteristicas CHAR(100) NOT NULL, -- Características de la propiedad
    precio INT NOT NULL, -- Precio de venta de la propiedad
    fechaPublicacion DATE NOT NULL, -- Fecha de publicación de la propiedad
    nolidentidad_Agente INT NOT NULL, -- ID del agente responsable de la propiedad
    nolidentidad_Vendedor INT NOT NULL, -- ID del vendedor de la propiedad
```

```
    FOREIGN KEY (noldentidad_Agente) REFERENCES agentes(id) ON DELETE CASCADE ON
    UPDATE CASCADE,
    FOREIGN KEY (noldentidad_Vendedor) REFERENCES vendedores(id) ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

- b. Mantenimiento de relaciones base (procedimientos almacenados, deben ser 3 por cada relación base (entidad fuerte), 3 puntos, un punto c/u):**

1. Crear.

Crear Agente

DELIMITER \$\$

```
CREATE PROCEDURE insertarAgente(IN id INT, IN nombre CHAR(10), IN dir CHAR(30), IN cel INT, IN tel INT)
BEGIN
    INSERT INTO agentes VALUES (id, nombre, dir, cel, tel);
END $$
```

DELIMITER ;

Crear Comprador

DELIMITER \$\$

```
CREATE PROCEDURE insertarComprador(IN id INT, IN nombre CHAR(10), IN dir CHAR(30), IN cel INT)
BEGIN
    INSERT INTO agentes VALUES (id, nombre, dir, cel);
END $$
```

DELIMITER ;

Crear Vendedor

DELIMITER \$\$

```
CREATE PROCEDURE insertarVendedor(IN id INT, IN nombre CHAR(10), IN dir CHAR(30), IN cel INT)
BEGIN
    INSERT INTO vendedores VALUES (id, nombre, dir, cel);
END $$
```

DELIMITER ;

Crear Propiedad Vendida

DELIMITER \$\$

```
CREATE PROCEDURE insertarPM(IN idPropiedad INT,
IN nombrePropiedad CHAR(10),
IN ciudad CHAR(20),
IN direccion CHAR(30),
IN cantidadDormitorios INT,
IN caracteristicas CHAR(100),
IN precio INT,
IN fechaPublicacion DATE,
IN fechaVenta DATE,
IN noldentidad_Agente INT,
IN noldentidad_Vendedor INT,
IN noldentidad_Comprador INT,
IN comisionVenta INT)
BEGIN
    INSERT INTO propiedades_vendidas VALUES (idPropiedad,
        nombrePropiedad,
        ciudad,
        direccion,
```

```
cantidadDormitorios,
caracteristicas,
precio,
fechaPublicacion,
fechaVenta,
noldentidad_Agente,
noldentidad_Vendedor,
noldentidad_Comprador,
comisionVenta);
END $$
```

```
DELIMITER ;
```

Crear Propiedad en Venta

```
DELIMITER $$
```

```
CREATE PROCEDURE insertarPV(
    IN idPropiedad INT,
    IN nombrePropiedad CHAR(10),
    IN ciudad CHAR(20),
    IN direccion CHAR(30),
    IN cantidadDormitorios INT,
    IN caracteristicas CHAR(100),
    IN precio INT,
    IN fechaPublicacion DATE,
    IN noldentidad_Agente INT,
    IN noldentidad_Vendedor INT
)
BEGIN
    INSERT INTO propiedades_en_mercado VALUES (
        idPropiedad,
        nombrePropiedad,
        ciudad,
        direccion,
        cantidadDormitorios,
        caracteristicas,
        precio,
        fechaPublicacion,
        noldentidad_Agente,
        noldentidad_Vendedor
    );
END $$
```

2. Modificar.

Modificar Agente

```
DELIMITER $$
```

```
CREATE PROCEDURE modificarAgente(IN id_agente INT, IN atributo CHAR(20), IN
nuevo_valor CHAR(30))
BEGIN
    CASE
        WHEN atributo = 'nombre' THEN
            UPDATE agentes SET nombre = nuevo_valor WHERE id = id_agente;
        WHEN atributo = 'direccion' THEN
            UPDATE agentes SET direccion = nuevo_valor WHERE id =
id_agente;
        WHEN atributo = 'celular' THEN
            UPDATE agentes SET celular = CAST(nuevo_valor AS UNSIGNED)
WHERE id = id_agente;
        WHEN atributo = 'telefonoOficina' THEN
            UPDATE agentes SET telefonoOficina = CAST(nuevo_valor AS
UNSIGNED) WHERE id = id_agente;
        ELSE
            -- Manejo de error si el atributo no es reconocido
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Atributo
desconocido';
    END CASE;
END $$
```

DELIMITER ;

Modificar Comprador

```
DELIMITER $$
```

```
CREATE PROCEDURE modificarComprador(IN id_comprador INT, IN atributo CHAR(20), IN
nuevo_valor CHAR(30))
BEGIN
    CASE
        WHEN atributo = 'nombre' THEN
            UPDATE compradores SET nombre = nuevo_valor WHERE id =
id_comprador;
        WHEN atributo = 'direccion' THEN
            UPDATE compradores SET direccion = nuevo_valor WHERE id =
id_comprador;
        WHEN atributo = 'celular' THEN
            UPDATE compradores SET celular = CAST(nuevo_valor AS
UNSIGNED) WHERE id = id_comprador;
        ELSE
            -- Manejo de error si el atributo no es reconocido
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Atributo
desconocido';
    END CASE;
END $$
```

DELIMITER ;

Modificar Vendedor

```
CREATE PROCEDURE modificarvendedor(IN id_vendedor INT, IN atributo CHAR(20), IN
nuevo_valor CHAR(30))
BEGIN
    CASE
```

```

        WHEN atributo = 'nombre' THEN
            UPDATE vendedores SET nombre = nuevo_valor WHERE id =
        id_vendedor;
        WHEN atributo = 'direccion' THEN
            UPDATE vendedores SET direccion = nuevo_valor WHERE id =
        id_vendedor;
        WHEN atributo = 'celular' THEN
            UPDATE vendedores SET celular = CAST(nuevo_valor AS UNSIGNED)
WHERE id = id_vendedor;
    ELSE
        -- Manejo de error si el atributo no es reconocido
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Atributo
desconocido';
    END CASE;
END $$

DELIMITER ;

```

Modificar Propiedad Vendida

```

DELIMITER $$

CREATE PROCEDURE modificarPropiedadEnMercado(IN id_propiedad INT, IN atributo
CHAR(20), IN nuevo_valor VARCHAR(100))
BEGIN
    CASE atributo
        WHEN 'nombre' THEN
            UPDATE propiedades_en_mercado SET nombre = nuevo_valor WHERE idPropiedad =
        id_propiedad;
        WHEN 'ciudad' THEN
            UPDATE propiedades_en_mercado SET ciudad = nuevo_valor WHERE idPropiedad =
        id_propiedad;
        WHEN 'direccion' THEN
            UPDATE propiedades_en_mercado SET direccion = nuevo_valor WHERE idPropiedad =
        id_propiedad;
        WHEN 'cantidadDormitorios' THEN
            UPDATE propiedades_en_mercado SET cantidadDormitorios = CAST(nuevo_valor AS
UNSIGNED) WHERE idPropiedad = id_propiedad;
        WHEN 'caracteristicas' THEN
            UPDATE propiedades_en_mercado SET caracteristicas = nuevo_valor WHERE
idPropiedad = id_propiedad;
        WHEN 'precio' THEN
            UPDATE propiedades_en_mercado SET precio = CAST(nuevo_valor AS UNSIGNED)
WHERE idPropiedad = id_propiedad;
        WHEN 'fechaPublicacion' THEN
            UPDATE propiedades_en_mercado SET fechaPublicacion = nuevo_valor WHERE
idPropiedad = id_propiedad;
        WHEN 'noldentidad_Agente' THEN
            UPDATE propiedades_en_mercado SET noldentidad_Agente = CAST(nuevo_valor AS
UNSIGNED) WHERE idPropiedad = id_propiedad;
        WHEN 'noldentidad_Vendedor' THEN
            UPDATE propiedades_en_mercado SET noldentidad_Vendedor = CAST(nuevo_valor AS
UNSIGNED) WHERE idPropiedad = id_propiedad;
    ELSE
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Atributo desconocido';
    END CASE;
END $$

DELIMITER ;

```

Modificar Propiedad en Venta

DELIMITER \$\$

```
CREATE PROCEDURE modificarPropiedadVendida(IN id_propiedad INT, IN atributo CHAR(20),
IN nuevo_valor VARCHAR(100))
BEGIN
    CASE atributo
        WHEN 'nombre' THEN
            UPDATE propiedades_vendidas SET nombre = nuevo_valor WHERE idPropiedad =
id_propiedad;
        WHEN 'ciudad' THEN
            UPDATE propiedades_vendidas SET ciudad = nuevo_valor WHERE idPropiedad =
id_propiedad;
        WHEN 'direccion' THEN
            UPDATE propiedades_vendidas SET direccion = nuevo_valor WHERE idPropiedad =
id_propiedad;
        WHEN 'cantidadDormitorios' THEN
            UPDATE propiedades_vendidas SET cantidadDormitorios = CAST(nuevo_valor AS
UNSIGNED) WHERE idPropiedad = id_propiedad;
        WHEN 'caracteristicas' THEN
            UPDATE propiedades_vendidas SET caracteristicas = nuevo_valor WHERE idPropiedad =
id_propiedad;
        WHEN 'precio' THEN
            UPDATE propiedades_vendidas SET precio = CAST(nuevo_valor AS UNSIGNED) WHERE
idPropiedad = id_propiedad;
        WHEN 'fechaPublicacion' THEN
            UPDATE propiedades_vendidas SET fechaPublicacion = nuevo_valor WHERE
idPropiedad = id_propiedad;
        WHEN 'fechaVenta' THEN
            UPDATE propiedades_vendidas SET fechaVenta = nuevo_valor WHERE idPropiedad =
id_propiedad;
        WHEN 'nolidentidad_Agente' THEN
            UPDATE propiedades_vendidas SET nolidentidad_Agente = CAST(nuevo_valor AS
UNSIGNED) WHERE idPropiedad = id_propiedad;
        WHEN 'nolidentidad_Vendedor' THEN
            UPDATE propiedades_vendidas SET nolidentidad_Vendedor = CAST(nuevo_valor AS
UNSIGNED) WHERE idPropiedad = id_propiedad;
        WHEN 'nolidentidad_Comprador' THEN
            UPDATE propiedades_vendidas SET nolidentidad_Comprador = CAST(nuevo_valor AS
UNSIGNED) WHERE idPropiedad = id_propiedad;
        WHEN 'comisionVenta' THEN
            UPDATE propiedades_vendidas SET comisionVenta = CAST(nuevo_valor AS UNSIGNED)
WHERE idPropiedad = id_propiedad;
        ELSE
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Atributo desconocido';
    END CASE;
END $$
```

DELIMITER ;

3. Eliminar.

Eliminar Agente

```
DELIMITER $$  
CREATE TRIGGER eliminarAgente AFTER DELETE ON agentes for each row  
BEGIN  
    DECLARE id_usuario_insertante INT;  
    SELECT u.id INTO id_usuario_insertante  
    FROM usuario u  
    WHERE u.activo = 'TRUE';  
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Eliminar agente','agentes',  
    NOW(),OLD.id);  
END $$
```

Eliminar Comprador

```
DELIMITER $$  
CREATE TRIGGER eliminarComprador AFTER DELETE ON compradores for each row  
BEGIN  
    DECLARE id_usuario_insertante INT;  
    SELECT u.id INTO id_usuario_insertante  
    FROM usuario u  
    WHERE u.activo = 'TRUE';  
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Eliminar comprador','compradores',  
    NOW(),OLD.id);  
END $$
```

Eliminar Vendedor

```
DELIMITER $$  
CREATE TRIGGER eliminarVendedor AFTER DELETE ON vendedores for each row  
BEGIN  
    DECLARE id_usuario_insertante INT;  
    SELECT u.id INTO id_usuario_insertante  
    FROM usuario u  
    WHERE u.activo = 'TRUE';  
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Eliminar vendedor','vendedores',  
    NOW(),OLD.id);  
END $$
```

Eliminar Propiedad Vendida

```
DELIMITER $$  
  
CREATE PROCEDURE eliminarPM(IN id_pm INT)  
BEGIN  
    DELETE FROM propiedades_en_mercado WHERE idPropiedad = id_pm;  
END $$
```

```
DELIMITER ;
```

Eliminar Propiedad en Venta

```
DELIMITER $$  
  
CREATE PROCEDURE eliminarPV(IN id_pv INT)  
BEGIN  
    DELETE FROM propiedades_vendidas WHERE idPropiedad = id_pv;  
END $$
```

```
DELIMITER ;
```

- c. Bitácora (disparadores, deben ser 3 por cada relación base (entidad fuerte), 3 puntos, un punto c/u):

1. Crear

Crear Agente

```
DELIMITER $$  
CREATE TRIGGER insertarAgente AFTER INSERT ON agentes for each row  
BEGIN  
    DECLARE id_usuario_insertante INT;  
    SELECT u.id INTO id_usuario_insertante  
    FROM usuario u  
    WHERE u.activo = 'TRUE';  
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Insertar agente','agentes',  
    NOW(),NEW.id);  
END $$
```

Crear Comprador

```
DELIMITER $$  
CREATE TRIGGER insertarComprador AFTER INSERT ON compradores for each row  
BEGIN  
    DECLARE id_usuario_insertante INT;  
    SELECT u.id INTO id_usuario_insertante  
    FROM usuario u  
    WHERE u.activo = 'TRUE';  
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Insertar comprador','compradores',  
    NOW(),NEW.id);  
END $$
```

Crear Vendedor

```
DELIMITER $$  
CREATE TRIGGER insertarVendedor AFTER INSERT ON vendedores for each row  
BEGIN  
    DECLARE id_usuario_insertante INT;  
    SELECT u.id INTO id_usuario_insertante  
    FROM usuario u  
    WHERE u.activo = 'TRUE';  
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Insertar vendedor','vendedores',  
    NOW(),NEW.id);  
END $$
```

Crear Propiedad Vendida

```
DELIMITER $$  
CREATE TRIGGER insertarPropiedadVendida AFTER INSERT ON propiedades_vendidas for each  
row  
BEGIN  
    DECLARE id_usuario_insertante INT;  
    SELECT u.id INTO id_usuario_insertante  
    FROM usuario u  
    WHERE u.activo = 'TRUE';  
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Insertar  
propiedad_vendida','propiedades_vendidas', NOW(),NEW.idPropiedad);  
END $$
```

Crear Propiedad en Venta

```
DELIMITER $$  
CREATE TRIGGER insertarPropiedadMercado AFTER INSERT ON propiedades_en_mercado for  
each row  
BEGIN  
    DECLARE id_usuario_insertante INT;  
    SELECT u.id INTO id_usuario_insertante  
    FROM usuario u
```

```

        WHERE u.activo = 'TRUE';
        INSERT INTO bitacora VALUES(id_usuario_insertante,'Insertar
propiedades_en_mercado','propiedades_en_mercado', NOW(),NEW.idPropiedad);
END $$
```

2. Modificar.

Modificar Agente

```

DELIMITER $$
CREATE TRIGGER modificarAgente AFTER UPDATE ON agentes for each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Agente modificado','agentes',
NOW(),NEW.id);
END $$
```

Modificar Comprador

```

DELIMITER $$
CREATE TRIGGER modificarComprador AFTER UPDATE ON compradores for each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Comprador
modificado','compradores', NOW(),NEW.id);
END $$
```

Modificar Vendedor

```

DELIMITER $$
CREATE TRIGGER modificarVendedores AFTER UPDATE ON vendedores for each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Vendedor modificado','vendedores',
NOW(),NEW.id);
END $$
```

Modificar Propiedad Vendida

```

DELIMITER $$
CREATE TRIGGER modificarPropiedadVendida AFTER UPDATE ON propiedades_vendidas for
each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Propiedad vendida
modificada','propiedades_vendidas', NOW(),NEW.idPropiedad);
END $$
```

Modificar Propiedad en Venta

```
DELIMITER $$
```

```

CREATE TRIGGER modificarPropiedadMercado AFTER UPDATE ON propiedades_en_mercado
for each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Propiedad en mercado
modificada','propiedades_en_mercado', NOW(),NEW.idPropiedad);
END $$
```

3. Eliminar.

Eliminar Agente

```

DELIMITER $$
CREATE TRIGGER eliminarAgente AFTER DELETE ON agentes for each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Eliminar agente','agentes',
NOW(),OLD.id);
END $$
```

Eliminar Comprador

```

DELIMITER $$
CREATE TRIGGER eliminarComprador AFTER DELETE ON compradores for each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Eliminar comprador','compradores',
NOW(),OLD.id);
END $$
```

Eliminar Vendedor

```

DELIMITER $$
CREATE TRIGGER eliminarVendedor AFTER DELETE ON vendedores for each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Eliminar vendedor','vendedores',
NOW(),OLD.id);
END $$
```

Eliminar Propiedad Vendida

```

DELIMITER $$
CREATE TRIGGER eliminarPropiedadMercado AFTER DELETE ON propiedades_en_mercado for
each row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Eliminar propiedad en
mercado','propiedades_en_mercado', NOW(),OLD.idPropiedad);
```

```
END $$
```

Eliminar Propiedad en Venta

```
DELIMITER $$
```

```
CREATE TRIGGER eliminarPropiedadVenta AFTER DELETE ON propiedades_vendidas for each
row
BEGIN
    DECLARE id_usuario_insertante INT;
    SELECT u.id INTO id_usuario_insertante
    FROM usuario u
    WHERE u.activo = 'TRUE';
    INSERT INTO bitacora VALUES(id_usuario_insertante,'Eliminar propiedad
vendida','propiedades_vendidas', NOW(),OLD.idPropiedad);
END $$
```

d. Reportes (vistas, 8 puntos, un punto c/u)

1. Cantidad de ventas por agente.

```
CREATE VIEW ventas_x_agente AS
select a.id AS id,a.nombre AS nombre,count(p.idPropiedad) AS
Cantidad_de_propiedades_vendidas
from (agentes a join propiedades_vendidas p on((a.id = p.noldentidad_Agente)))
group by a.id;
```

2. Cantidad de ventas por vendedor.

```
CREATE VIEW ventas_x_vendedor AS
select a.id AS id,a.nombre AS nombre,count(p.idPropiedad) AS
Cantidad_de_propiedades_vendidas
from (vendedores a join propiedades_vendidas p on((a.id = p.noldentidad_Vendedor)))
group by a.id;
```

3. Cantidad de compras por comprador.

```
CREATE VIEW compras_x_comprador AS
select c.id AS id,c.nombre AS nombre,count(p.idPropiedad) AS Cant_Propiedades_Compradas
from (compradores c join propiedades_vendidas p on((c.id = p.noldentidad_Comprador)))
group by c.id;
```

4. Ventas por ubicación.

Método 1(Muestra las ventas totales por dirección)

```
CREATE VIEW ventas_x_ubicacion AS select p.ciudad AS ciudad,p.direccion AS
direccion,count(0) AS total_ventas
from propiedades_vendidas p group
by p.direccion;
```

Método 2(Muestra las id de las direcciones)

```
CREATE VIEW ventas_x_ubicacion_ordenado AS select p.direccion AS direccion,p.ciudad AS
ciudad,p.idPropiedad AS idPropiedad
from propiedades_vendidas p
order by p.direccion;
```

5. Ventas por precio de propiedad.

Método 1

```
CREATE VIEW ventas_x_precio AS select p.precio AS precio,count(p.idPropiedad) AS
Cant_de_casas
```

```
from propiedades_vendidas p  
group by p.precio;
```

Método 2

```
CREATE VIEW ventas_x_precio_id AS select p.precio AS precio,p.idPropiedad AS IdPropiedad  
from propiedades_vendidas p  
order by p.precio;
```

6. Ventas de propiedades por características (por ejemplo, cantidad de habitaciones, si tiene piscina, etc.).

```
CREATE VIEW ventas_x_caracteristica AS select p.cantidadDormitorios AS  
cantidadDormitorios,p.idPropiedad AS idPropiedad,(case when (p.caracteristicas like '%Tiene  
piscina%') then 'SI' else 'NO' end) AS TienePiscina  
from propiedades_vendidas p  
order by p.cantidadDormitorios;
```

7. Agente que vendió la mayor cantidad de propiedades en el año por valor total.

```
DELIMITER $$  
CREATE PROCEDURE agenteVendioMayor(fechaEmpezar Int, fechaTerminar Int)  
BEGIN  
    SELECT a.nombre, a.id,count(p.idPropiedad) as propiedades_vendidas,Sum(p.precio)  
    as ValorTotal  
    FROM agentes a  
    JOIN propiedades_vendidas p ON a.id = p.noldentidad_Agente  
    WHERE year(p.fechaVenta) BETWEEN fechaEmpezar AND fechaTerminar  
    GROUP BY a.id, a.nombre  
    ORDER BY COUNT(p.idPropiedad) DESC, SUM(p.precio) DESC  
    LIMIT 1;  
END$$  
DELIMITER ;
```

8. Para cada agente, calcule el precio de venta promedio de las propiedades vendidas en el año y el tiempo promedio que la propiedad estuvo en el mercado.

```
DELIMITER $$  
CREATE PROCEDURE promedioVentas(IN anio int)  
BEGIN  
    SELECT a.id, a.nombre, COALESCE(AVG(p.precio), 0) as promedioPrecio,  
    AVG(TIMESTAMPDIFF(MINUTE, p.fechaPublicacion, p.fechaVenta)) AS  
    tiempoPromedioEnMercado  
    FROM agentes a JOIN propiedades_vendidas p ON a.id = p.noldentidad_Agente  
    WHERE YEAR(p.fechaVenta) = anio  
    GROUP BY a.id;  
END$$  
DELIMITER ;
```

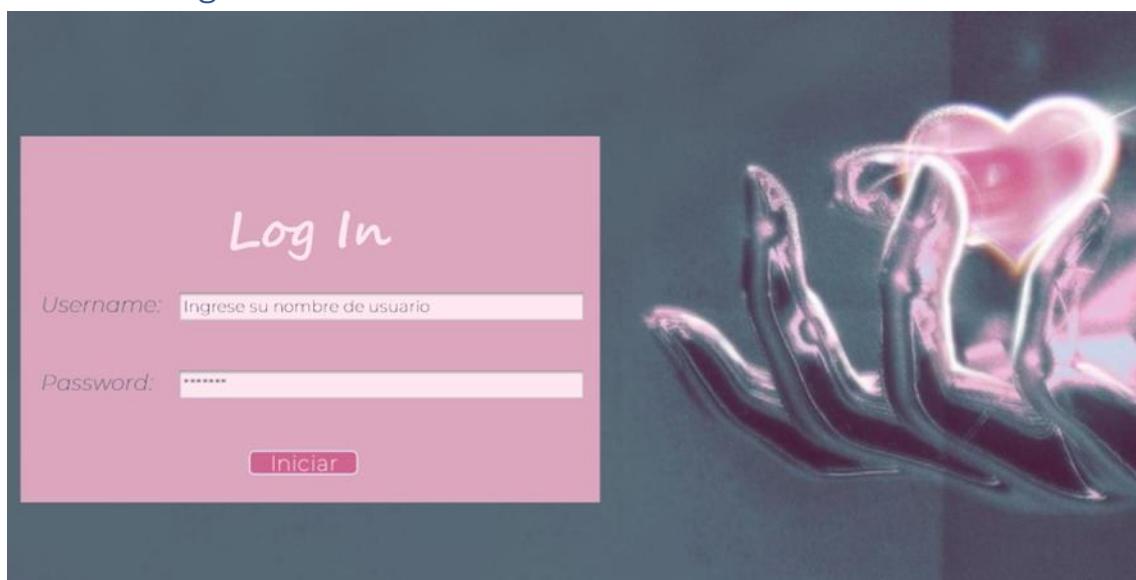
Anexos

Proyecto en Github

https://github.com/JRafaelLobo/TeoriaDeBaseDeDatos1_Proyecto.git

Manual de Usuario

Ventana Log-in



La ventana de iniciar sesión le permite a los usuarios entrar a sus respectivas ventanas. Según el tipo de usuario que es cada uno se abrirá una ventana de administrador, agente, vendedor o comprador. Para realizar esto el programa pide que el usuario ingrese su username y contraseña, en caso que uno o ambos estén incorrectos, el programa enviara un mensaje alertando al usuario.

Ventana Administrador



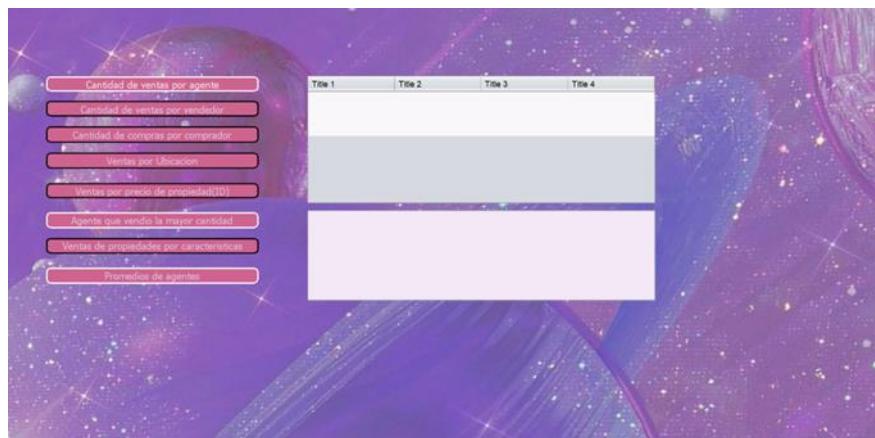
La “Ventana administrador” es aquella que se le muestra solo a los empleados tipo administrador. Cuenta con las funciones de Ver Reportes, Ver bitácora, Mantenimiento (Agentes, Compradores, Vendedores, Propiedades Vendidas, y Propiedades en Venta), y permite visualizar las Propiedades en Venta, las Propiedades Vendidas, los Agentes, Vendedores y Compradores que han sido creados. Las funcionalidades de los administradores serán explicadas a detalle a continuación:

Reportes

La ventana reportes se accede por medio del botón “Ver Reportes” en la ventana que se le muestra a los administradores.



La ventana reportes muestra los datos seleccionados de reportes de los datos del programa por medio de una tabla, estos son:



- Cantidad de ventas por agente: muestra el id, nombre y cantidad de ventas de todos los agentes existentes.

Cantidad de ventas por agente		
id	nombre	Cantidad Vendida
1	hardin	2
2	Mario	4
9	Karia	1

1). ID: 1 | NOMBRE: hardin | CANTIDAD VENDIDAS: 2
2). ID: 2 | NOMBRE: Mario | CANTIDAD VENDIDAS: 4
3). ID: 9 | NOMBRE: Karia | CANTIDAD VENDIDAS: 1

- Cantidad de ventas por vendedor: muestra el id, nombre y cantidad de ventas de todos los vendedores existentes.

Cantidad de ventas por vendedor		
id	nombre	Cantidad Vendida
3	Piper	1
7	Rosa	6

1). ID: 3 | NOMBRE: Piper | CANTIDAD VENIDAS: 1
2). ID: 7 | NOMBRE: Rosa | CANTIDAD VENIDAS: 6

- Cantidad de compras por comprador: muestra el id, nombre y cantidad de compras de todos los compradores existentes.

Cantidad de ventas por vendedor		
id	nombre	Cantidad Vendida
3	Piper	1
7	Rosa	6

1). ID: 3 | NOMBRE: Piper | CANTIDAD VENIDAS: 1
2). ID: 7 | NOMBRE: Rosa | CANTIDAD VENIDAS: 6

- Ventas por ubicación: muestra cuantas ventas se han registrado por cada ubicación existente de una propiedad. Además del id y dirección de cada propiedad por ubicación.

Ventas por Ubicacion		
ciudad	direccion	idPropiedad
knowles	carter	5
lisa	jennie	4
namjoon	jimin	6
myleon	minnie	9
b	nicki	3
Jacaleapa	nose	1
seoul	nose	2

1). Ciudad: Jacaleapa | Dirección: nose | Total_Ventas: 2
2). Ciudad: b | Dirección: nicki | Total_Ventas: 1
3). Ciudad: lisa | Dirección: jennie | Total_Ventas: 1
4). Ciudad: knowles | Dirección: carter | Total_Ventas: 1
5). Ciudad: namjoon | Dirección: jimin | Total_Ventas: 1
6). Ciudad: myleon | Dirección: minnie | Total_Ventas: 1

- Ventas por precio de propiedad: muestra el id y el precio de cada propiedad. Además de la cantidad de propiedades que tienen el mismo precio.

Ventas por precio de propiedad	
Precio	IDPropiedad
1289	6
1890	2
2302	1
8900	4
8976	3
23478	5
331000	9

1) Precio: 2302 | CANTIDAD DE CASAS: 1
2) Precio: 1890 | CANTIDAD DE CASAS: 1
3) Precio: 8976 | CANTIDAD DE CASAS: 1
4) Precio: 8900 | CANTIDAD DE CASAS: 1
5) Precio: 23478 | CANTIDAD DE CASAS: 1
6) Precio: 1289 | CANTIDAD DE CASAS: 1

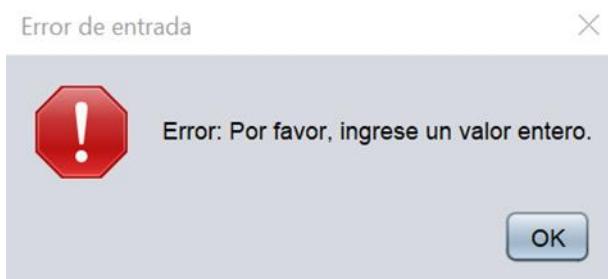
- Agente que vendió la mayor cantidad: muestra el agente que vendió más propiedades dentro de un periodo solicitado por el usuario.

Especifica su id, la cantidad de propiedades que este vendió, el periodo que se tomo en cuenta y el valor total de estas.

Agente que vendio la mayor cantidad				
Id	Nombre	Propiedades Vendidas	Valor Total	2001 - 2022
2	Mario	3	355767	Agente del Año

En el año por valor total: ID: 2 NOMBRE: Mario Propiedades Vendidas: 3 Valor vendido 355767

En caso de que los años ingresados por el usuario sean inválidos, no se mostraran los datos y se enviará un mensaje de error.



- Ventas de propiedades por características: muestra el id de la propiedad, su cantidad de dormitorios, y si tiene piscina o no.

Ventas de propiedades por características		
idPropiedad	cantidadDormitorios	TienePiscina
1	1	NO
2	2	NO
3	4	NO
4	4	NO
9	5	NO
6	7	NO
5	12	NO
18	34	NO

CANTIDAD DE DORMITORIOS: 1 ID: 1 Piscina: NO
CANTIDAD DE DORMITORIOS: 2 ID: 2 Piscina: NO
CANTIDAD DE DORMITORIOS: 3 ID: 3 Piscina: NO
CANTIDAD DE DORMITORIOS: 4 ID: 4 Piscina: NO
CANTIDAD DE DORMITORIOS: 5 ID: 9 Piscina: NO
CANTIDAD DE DORMITORIOS: 7 ID: 6 Piscina: NO

- Promedios de agentes: muestra el id, nombre, precio promedio de ventas, y tiempo promedio de cada agente activo en un año seleccionado por el usuario.



En caso de que el año ingresado por el usuario sea invalido, no se mostraran los datos y se enviara un mensaje de error.



Bitácora

La ventana “Bitácora” se ingresa por medio del botón “Ver Bitácora” en la interfaz gráfica de Usuario administrador.



Esta ventana muestra el id, la acción realizada, la tabla afectada, el tiempo en que se realizó la acción, y el objeto modificado por cada acción realizada en el programa.

Id: Buscar

ID	Acción	Tabla afectada	Timestamp	Objeto modificado
0	Insertar agente	agentes	2024-03-17 19:03:46	8012003
12241127	Eliminar agente	agentes	2024-03-17 21:20:08	8012003
12241172	Insertar agente	agentes	2024-03-17 21:51:00	8011974
12241007	Insertar agente	agentes	2024-03-17 22:08:12	1
12241007	Insertar agente	agentes	2024-03-17 22:08:35	2
12241007	Insertar agente	agentes	2024-03-17 22:09:10	9
12241007	Agente modificado	agentes	2024-03-17 22:09:29	1
12241007	Eliminar agente	agentes	2024-03-17 22:09:34	8011974
12241007	Insertar comprador	compradores	2024-03-17 22:10:14	1
12241007	Insertar comprador	compradores	2024-03-17 22:10:53	2
12241007	Insertar comprador	compradores	2024-03-17 22:11:40	3
12241007	Comprador modificado	compradores	2024-03-17 22:12:30	3
12241007	Insertar comprador	compradores	2024-03-17 22:12:46	4
12241007	Eliminar comprador	compradores	2024-03-17 22:12:50	4
12241007	Insertar vendedor	vendedores	2024-03-17 22:13:34	7
12241007	Insertar vendedor	vendedores	2024-03-17 22:14:52	3
12241007	Vendedor modificado	vendedores	2024-03-17 22:15:06	3
12241007	Insertar vendedor	vendedores	2024-03-17 22:15:30	1
12241007	Eliminar vendedor	vendedores	2024-03-17 22:15:35	1
12241007	Insertar propiedades_en_me...	propiedades_en_mercado	2024-03-17 22:16:48	1
12241007	Insertar propiedades_en_me...	propiedades_en_mercado	2024-03-17 22:19:26	6

Además ingresando el id de un usuario y presionando buscar, muestra todas las acciones realizadas por ese usuario.

Id: 12241127 Buscar

ID	Acción	Tabla afectada	Timestamp	Objeto modificado
12241127	Eliminar agente	agentes	2024-03-17 21:20:08	8012003
12241127	Insertar propiedad_vendida	propiedades_vendidas	2024-03-17 22:25:28	3
12241127	Insertar propiedad_vendida	propiedades_vendidas	2024-03-17 22:28:28	5
12241127	Insertar propiedad_vendida	propiedades_vendidas	2024-03-17 22:29:44	6
12241127	Insertar propiedad_vendida	propiedades_vendidas	2024-03-17 22:30:46	4
12241127	Propiedad_vendida modificada	propiedades_vendidas	2024-03-17 22:31:02	1
12241127	Eliminar propiedad_en_mercado	propiedades_en_mercado	2024-03-17 22:32:31	1
12241127	Eliminar propiedad_en_mercado	propiedades_en_mercado	2024-03-17 22:32:39	6
12241127	Insertar propiedades_en_mer...	propiedades_en_mercado	2024-03-17 22:34:35	1
12241127	Insertar propiedades_en_mer...	propiedades_en_mercado	2024-03-17 22:35:29	2
12241127	Propiedad_en_mercado modifi...	propiedades_en_mercado	2024-03-17 22:35:44	1

En caso que el id de ese usuario no exista o de que no haya realizado ningún cambio la tabla se mostrara vacía.

Id: 1 Buscar

ID	Acción	Tabla afectada	Timestamp	Objeto modificado
1				

Mantenimiento Propiedades Vendidas

Se accede por medio de la pestaña “Mantenimiento” en la ventana administrador en el botón “Propiedades Vendidas” en la barra de botones “Administrar”.



Esta ventana nos permite crear, buscar, modificar y eliminar propiedades que han sido vendidas.



Crear Propiedad Vendida

Crear agente requiere que el administrador ingrese el id, nombre, ciudad, dirección, cantidad de dormitorios, características, precio, fecha de publicación, fecha de venta, id agente asignado, id del vendedor, y id del comprador de la nueva propiedad vendida. Además, que se debe seleccionar una imagen para representar la propiedad.

Crear Propiedad Vendida

Id:	<input type="text"/>
Nombre:	<input type="text"/>
Ciudad:	<input type="text"/>
Dirección:	<input type="text"/>
Dormitorios:	<input type="text"/>
Características:	<input type="text"/>
Precio:	<input type="text"/>
Fecha Publicación:	<input type="text"/>
Fecha Venta:	<input type="text"/>
ID Agente:	<input type="text" value="2"/>
ID Vendedor:	<input type="text" value="3"/>
ID Comprador:	<input type="text" value="1"/>
Comisión:	<input type="text"/>



En caso que uno de los datos sea invalido o no se haya seleccionado una imagen se enviará un mensaje del error y no se permitirá crear la propiedad. Si la creación es exitosa se enviará un mensaje de confirmación al usuario.



Modificar Propiedad vendida

Modificar Propiedad Vendida modifica una propiedad vendida utilizando su id. El usuario selecciona el id de la propiedad que quiere modificar, el atributo que quiere cambiar, y su nuevo valor.



En caso que el nuevo valor ingresado sea invalido se enviara un mensaje de error al usuario y no se realizara la modificación.



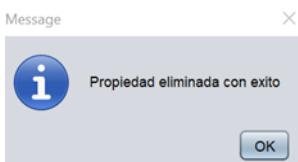
Si la modificación es exitosa se enviará un mensaje de confirmación al usuario



Eliminar Agente

Se selecciona el id que se quiere borrar y este se elimina al presionar eliminar.

Al finalizar se envía un mensaje confirmando la eliminación.



Buscar Propiedades Vendidas

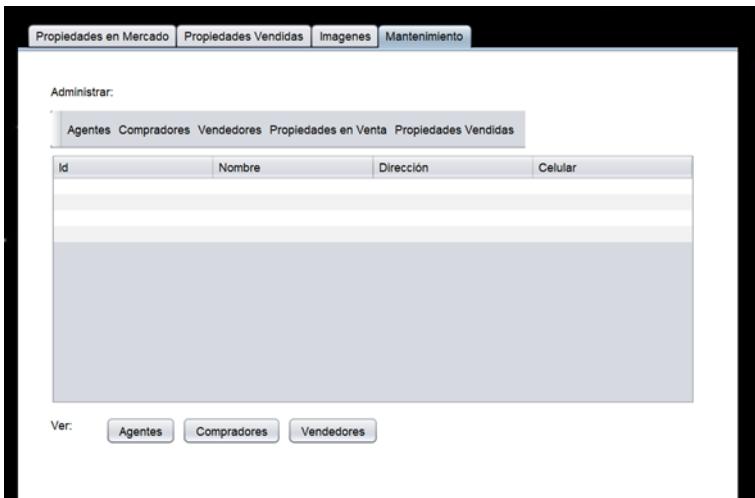
Se ingresa el id de una propiedad vendida, si esta existe se muestran sus datos en una tabla al presionar buscar.

Propiedades Vendidas

ID:	9												Buscar
id	nombre	ciudad	direccion	cant_dormitorios	caracteristicas	precio	fechaPublicacion	fechaVenta	noldentido	noldentido	noldentido	comision...	
9	idle	mlyeon	minnle	5	no	331000	2022-02-02	2022-03-03	2	7	2	456	

Mantenimiento Propiedades en Venta

Se accede por medio de la pestaña “Mantenimiento” en la ventana administrador en el botón “Propiedades en Venta” en la barra de botones “Administrar”.



Esta ventana nos permite crear, buscar, modificar y eliminar propiedades que están actualmente en el mercado

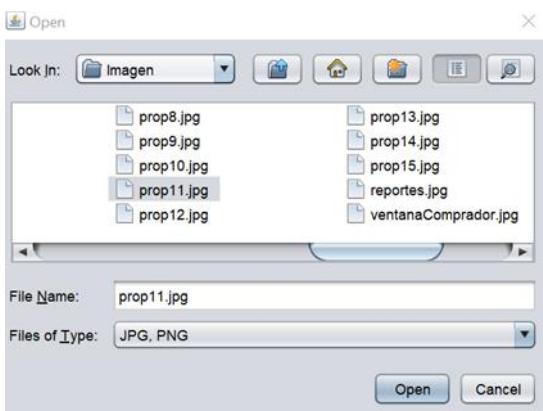


Crear Propiedad en Venta

Crear agente requiere que el administrador ingrese el id, nombre, ciudad, dirección, cantidad de dormitorios, características, precio, fecha, id agente asignado, id del vendedor de la propiedad. Además, que se debe seleccionar una imagen para representar la propiedad.

Crear Propiedad en Venta

Id:	
Nombre:	
Ciudad:	
Dirección:	
Dormitorios:	
Características:	
Precio:	
Fecha:	
ID Agente:	2
ID Vendedor:	3



En caso que uno de los datos sea invalido o no se haya seleccionado una imagen se enviará un mensaje del error y no se permitirá crear la propiedad. Si la creación es exitosa se enviará un mensaje de confirmación al usuario.



Modificar Propiedad en Venta

Modificar Propiedad Vendida modifica una propiedad vendida utilizando su id. El usuario selecciona el id de la propiedad que quiere modificar, el atributo que quiere cambiar, y su nuevo valor.

Modificar Propiedad en Venta

Id:	<input type="text" value="2"/>
Atributo:	<input type="text" value="nombre"/>
Modificación:	<input type="text" value=""/>
<input type="button" value="Modificar"/>	

En caso que el nuevo valor ingresado sea invalido se enviara un mensaje de error al usuario y no se realizara la modificación.



Si la modificación es exitosa se enviará un mensaje de confirmación al usuario



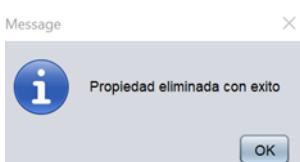
Eliminar Agente

Se selecciona el id que se quiere borrar y este se elimina al presionar eliminar.

Eliminar Propiedad en Venta

ID:	<input type="text" value="2"/>
<input type="button" value="Eliminar"/>	

Al finalizar se envía un mensaje confirmando la eliminación.



Buscar Propiedades Vendidas

Se ingresa el id de una propiedad en venta, si esta existe se muestran sus datos en una tabla al presionar buscar.

Propiedades en Venta									
ID:	4	Buscar							
id	nombre	ciudad	direccion	cant_d...	caracte...	precio	fechaPub	agente	vendedor
4	sailor	tegux	f	2	no	12	2024-1...	12241345	122422...

Mantenimiento Agentes

Se accede por medio de la pestaña “Mantenimiento” en la ventana administrador en el botón “Agentes” en la barra de botones “Administrar”.

Propiedades en Mercado	Propiedades Vendidas	Imagenes	Mantenimiento																
Administrar:																			
Agentes Compradores Vendedores Propiedades en Venta Propiedades Vendidas																			
<table border="1"><thead><tr><th>Id</th><th>Nombre</th><th>Dirección</th><th>Celular</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></tbody></table>				Id	Nombre	Dirección	Celular												
Id	Nombre	Dirección	Celular																
Ver:	Agentes	Compradores	Vendedores																

Esta ventana nos permite crear, modificar y eliminar agentes.

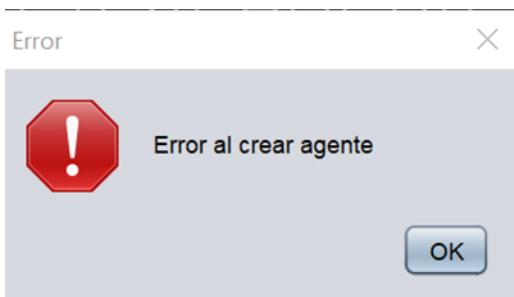


Crear Agente

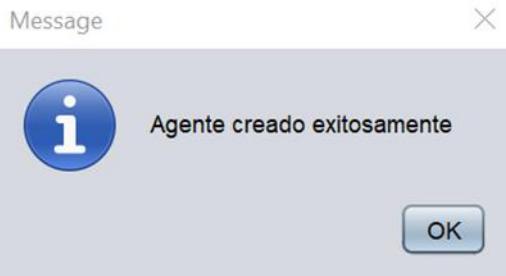
Crear agente requiere que el administrador ingrese el id, nombre, dirección, celular y teléfono de Oficina del nuevo agente.



En caso que uno de los datos sea invalido se enviará un mensaje del error y no se permitirá crear el agente.



Si la creación es exitosa se enviará un mensaje de confirmación al usuario.



Modificar Agente

Modificar agente modifica un agente utilizando su id. El usuario selecciona el id del agente a modificar y el atributo que quiere cambiar con su nuevo valor.



En caso que el nuevo valor ingresado sea invalido se enviara un mensaje de error al usuario y no se realizara la modificación.



Si la modificación es exitosa se enviará un mensaje de confirmación al usuario

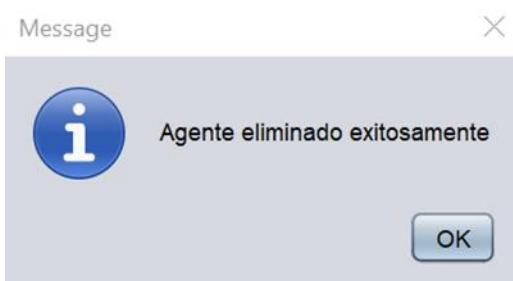


Eliminar Agente

Se selecciona el id que se quiere borrar y este se elimina al presionar eliminar.



Al finalizar se envía un mensaje confirmando la eliminación.



Buscar Agente

Se ingresa el id de un agente, al presionar buscar, si el agente existe, se mostraran sus datos ordenados en una tabla.

Resultados de la búsqueda				
Id	Nombre	Dirección	Celular	Telefono Oficina
2	Mario	nose	12345678	998022

Mantenimiento Compradores

Se accede por medio de la pestaña “Mantenimiento” en la ventana administrador en el botón “Compradores” en la barra de botones “Administrar”.

A screenshot of the "Mantenimiento" tab in the "Administrador" window. The main area displays a table with columns: Id, Nombre, Dirección, and Celular. Above the table, there are buttons for "Agentes", "Compradores", "Vendedores", and "Propiedades en Venta". At the bottom, there is a footer with buttons for "Ver:" and "Agentes", "Compradores", "Vendedores".

Esta ventana nos permite crear, modificar y eliminar Compradores.

Crear Comprador

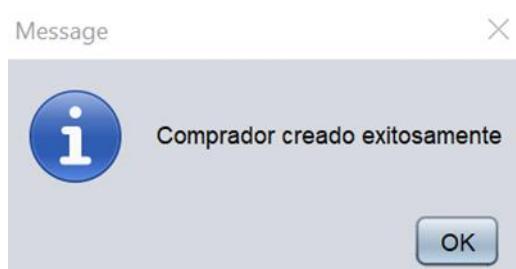
Crear comprador requiere que el administrador ingrese el id, nombre, dirección y celular del comprador que se quiere crear.



En caso que uno de los datos sea invalido se enviará un mensaje del error y no se permitirá crear el agente.



Si la creación es exitosa se enviará un mensaje de confirmación al usuario.



Modificar Comprador

Modificar comprador modifica un comprador existente requiriendo su id. El usuario selecciona el id del comprador que desea modificar y el atributo que quiere cambiar junto con su nuevo valor.



En caso que el nuevo valor ingresado sea invalido se enviara un mensaje de error al usuario y no se realizara la modificación.



Si la modificación es exitosa se enviará un mensaje de confirmación al usuario

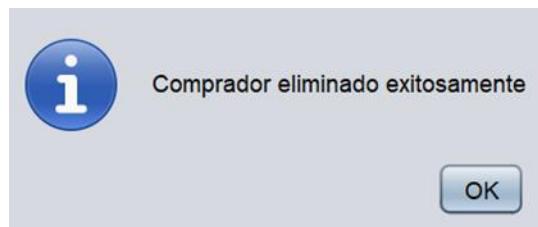


Eliminar Agente

Se selecciona el id que se quiere borrar y este se elimina al presionar eliminar.



Al finalizar se envía un mensaje confirmando la eliminación.



Buscar Comprador

Se ingresa el id de un comprador, al presionar buscar, si el comprador existe, se mostrarán sus datos en una tabla.

Administrador Comprador

ID:

ID	Nombre	Direccion	Celular
16	Mario	Alameda	31401222

Mantenimiento Vendedores

Se accede por medio de la pestaña “Mantenimiento” en la ventana administrador en el botón “Vendedores” en la barra de botones “Administrar”.

Propiedades en Mercado Propiedades Vendidas Imagenes Mantenimiento

Administrar:

Agentes Compradores Vendedores Propiedades en Venta Propiedades Vendidas

ID	Nombre	Dirección	Celular

Ver:

Esta ventana nos permite crear, modificar y eliminar Compradores.

Crear Vendedor

Crear Vendedor requiere que el administrador ingrese el id, nombre, dirección y celular del vendedor que se quiere crear.



A screenshot of a mobile application interface titled "Crear Vendedor". It features a header with a small star icon. Below the title are four input fields: "Id:" (with placeholder "0"), "Nombre:" (with placeholder "Nombre"), "Dirección:" (with placeholder "Dirección"), and "Celular:" (with placeholder "Celular"). At the bottom right is a pink "Crear" button.

Modificar Vendedor

El usuario selecciona el id del vendedor que desea modificar y el atributo que quiere cambiar junto con su nuevo valor.



A screenshot of a mobile application interface titled "Modificar Vendedor". It has three input fields: "Id:" with value "3", "Atributo" with value "nombre", and "Modificación:" (empty). At the bottom right is a purple "Modificar" button.

En caso que el nuevo valor ingresado sea invalido se enviara un mensaje de error al usuario y no se realizara la modificación.



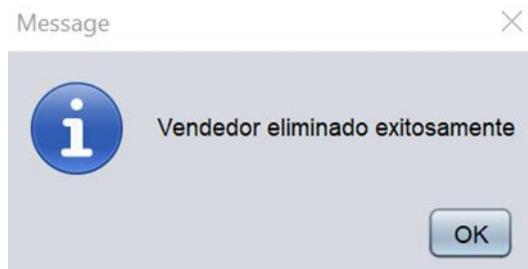
Si la modificación es exitosa se enviará un mensaje de confirmación al usuario



Eliminar Agente

Se selecciona el id que se quiere borrar y este se elimina al presionar eliminar.

Al finalizar se envía un mensaje confirmando la eliminación.



Buscar Vendedor

Se busca el id de un vendedor, si este existe en la base de datos, se muestra su id, nombre, dirección, y celular.

Administrador Vendedores				
ID: 12242234 Buscar				
ID	Nombre	Dirección	Celular	
12242234	Alejandro Sanchez	h	1	

Propiedades en Mercado

La pestaña muestra los datos de las propiedades en venta con la opción de actualizar si se hacen modificaciones.

Propiedades en Mercado Propiedades Vendidas Imágenes Mantenimiento									
Actualizar									
nombre	ciudad	direccion	cantidad ...	caracteri...	precio	fechaPub...	Agente	Vendedor	
2	f(x)	krystal	luna	5	piscina	10009922	2009-09-09	2	
3	grimes	elon	musk	4	no	9	2021-01-01	9	
4	sailor	tegux	f	2	no	12	2024-12-12	12241345	

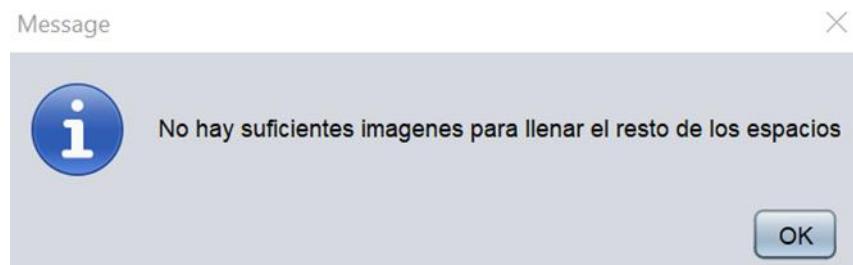
Propiedades Vendidas

La pestaña muestra los datos de las propiedades vendidas con la opción de actualizar si se hacen modificaciones.

id	nombre	ciudad	direc...	cant...	carac...	precio	fecha...	fecha...	nolde...	nolde...	comis...
5	beyo...	knowl...	carter	12	piscina	23478	2009-...	2009-...	2	7	3
6	bts	namj...	jimin	7	piscina	1289	2016-...	2017-...	2	7	3
9	idle	mlyeon	minnie	5	no	331000	2022-...	2022-...	2	7	2

Imágenes de Propiedades

Permite cargar las imágenes de las propiedades creadas. Si no existen suficientes propiedades para llenar los espacios se envía un mensaje de advertencia al usuario.



Al presionar cargar se pueden ver las imágenes de las propiedades

Vendedor

Vendedor permite cargar las imágenes de las propiedades vendidas en la pestaña “Propiedades Vendidas”

Vendedor

Propiedades Vendidas Mis propiedades Propiedades en Venta Propiedades Disponibles

Cargar imágenes

Propiedad 7 id

Propiedad 8 id

Y permite ver sus imágenes de propiedades en venta

Vendedor

Propiedades Vendidas Mis propiedades Propiedades en Venta Propiedades Disponibles

cargar imágenes

3 76 4

65 892

Comprador

La pestaña “Propiedades en el mercado” carga las propiedades en venta.



La pestaña “Propiedades compradas” carga las propiedades que el comprador a comprado.



Agente

Permite ver las imágenes de las propiedades asignadas en “Propiedades del mercado asignadas” para las propiedades en venta y “Propiedades vendidas” para las propiedades vendidas.

