

INFORME MER INMOBILIARIA SOFKAU

Actividad 4
C1-2023-QA-BD-04

Entregado A
Juanes Pineda

Presentado
Yeison Ferney Osorio Buitrago

SOFKAU
SAN JOSE DE CUCUTA

TABLE DE CONTENIDO

Mer Inmobiliaria SofkaU	3
Explicación MER	3
Explicación de las relaciones en el Modelo Relacional	4
SQL de la creación de la base de datos	8
Representación de la base de datos en Workbench.....	16
Modelo Relacional Completado.....	17
Ingreso de los bases a la base de datos	17
¿Qué debería cambiar o agregar para incluir la renovación de contratos de alquiler?	23
¿Cómo podría controlar que la misma persona que compra un inmueble con identificador X y se lo entrega a la agencia Inmobiliaria SofkaU, NO pueda alquilar el inmueble con identificador X?	23

Mer Inmobiliaria SofkaU

Requerimientos:

- ✓ Complete únicamente las cardinalidades del modelo entidad relacional adjunto (diagrama1.dia). Anexo en el repositorio
- ✓ Pase el diagrama a workbench donde incluya todas las cardinalidades que especifique en el punto anterior. Anexo en el repositorio
- ✓ Genere el Script aplicando ingeniería hacia adelante. Anexo en el repositorio
- ✓ Corra el Script y genere la base de datos. Anexo en el repositorio
- ✓ Poblar todas las tablas de la base de datos al menos con 5 registros.

después de haber realizado todo lo anterior mencionado responda:

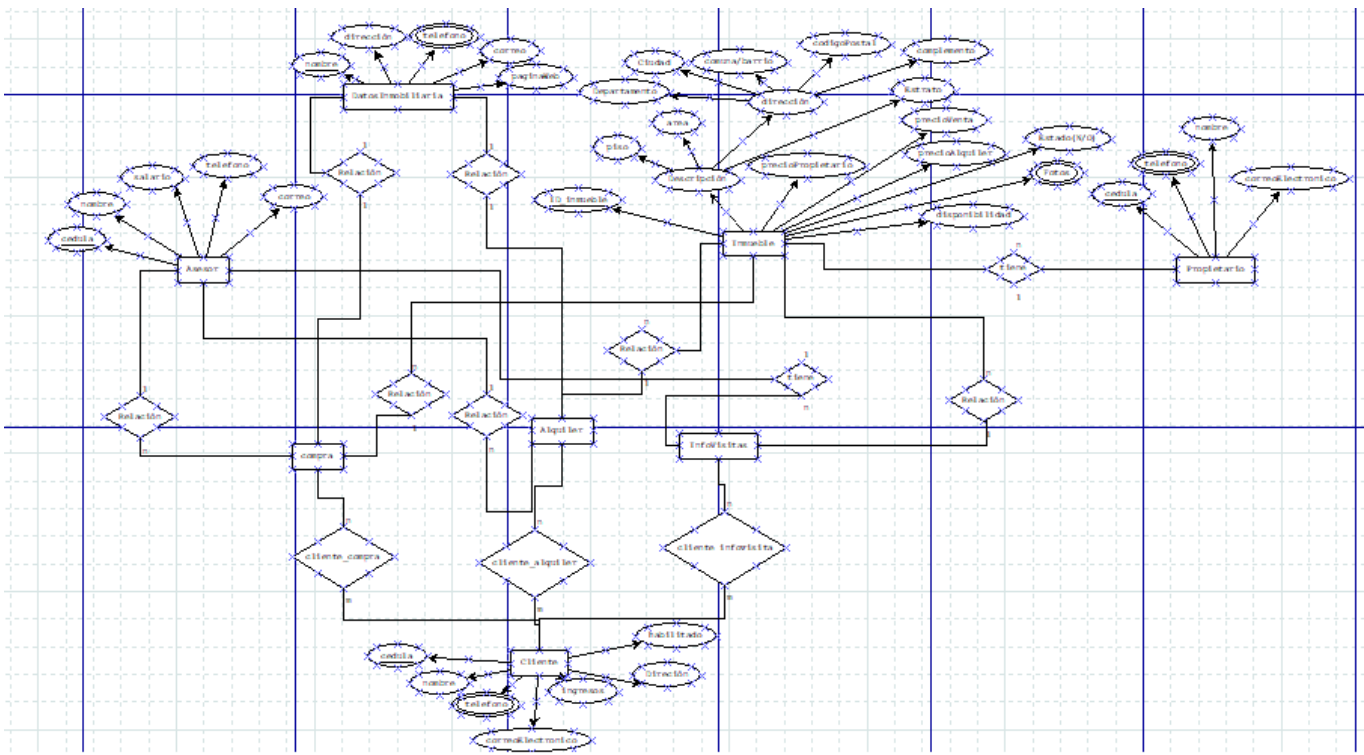
¿Qué debería cambiar o agregar para incluir la renovación de contratos de alquiler?

¿Cómo podría controlar que la misma persona que compra un inmueble con identificador X y se lo entrega a la agencia Inmobiliaria SofkaU, NO pueda alquilar el inmueble con identificador X?

Genere un archivo PDF con la documentación clara que indique el paso a paso de como realizo la solución del taller y las respuestas de las dos preguntas anteriores.

Explicación MER

Primero se analizó el modelo entidad relación realizado en clase para agregarle la cardinalidad a cada relación que fuese acorde a lo solicitado Anexo archivo en el repositorio



Explicación de las relaciones en el Modelo Relacional

Normalización

Primera Forma Normal 1FN

- Para cumplir con la función debimos crear tabla para los valores multivaluados y de esta manera cumplir con los requerimientos.
- No deben existir registros duplicado.
- Todas las tablas tienen llave primaria

Se cumple con la 2FN.

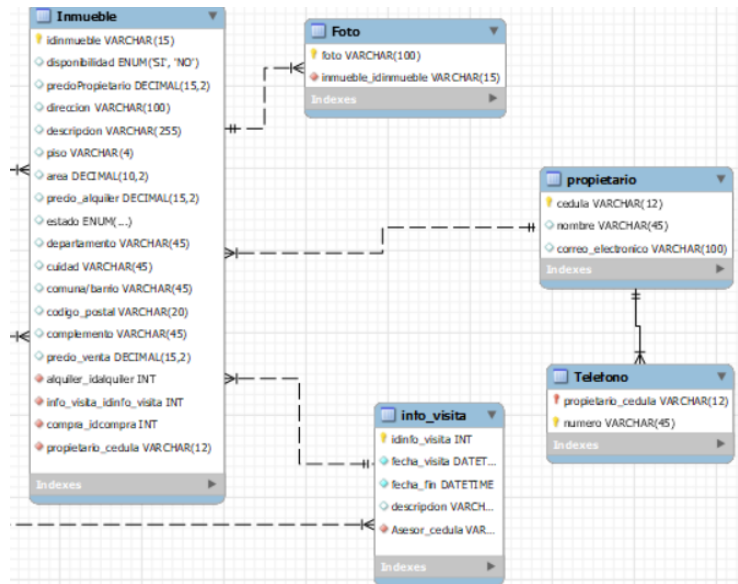
- Cumple con 1FN
- Todos los valores de las columnas dependen únicamente de la llave primaria de la tabla.
- Las tablas tienen una única llave primaria que identifica a la tabla y sus atributos dependen de ella. Esto significa que si se conoce la llave primaria, se pueden determinar todos los demás valores en la tabla.

Se cumple con la 3FN.

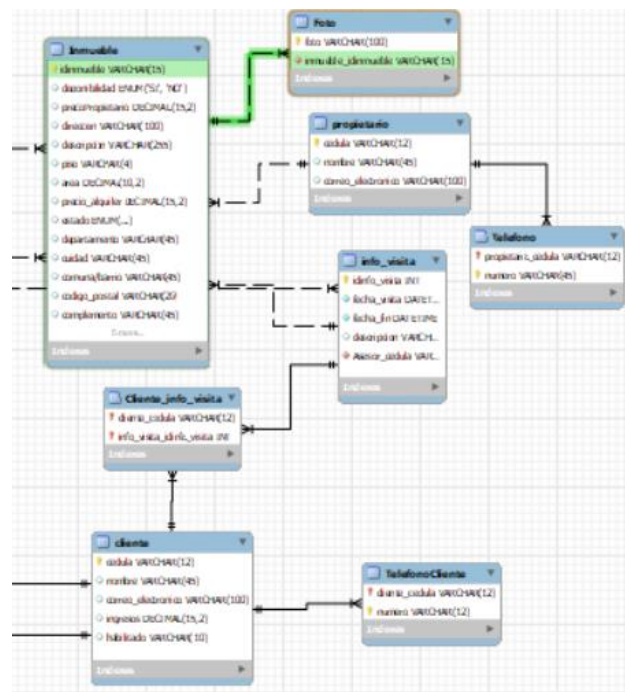
Estar en 2FN

- Cada atributo que no está incluido en la clave primaria no depende de la clave primaria. Esto significa que un atributo no debe depender de otro atributo que no forme parte de la clave primaria.

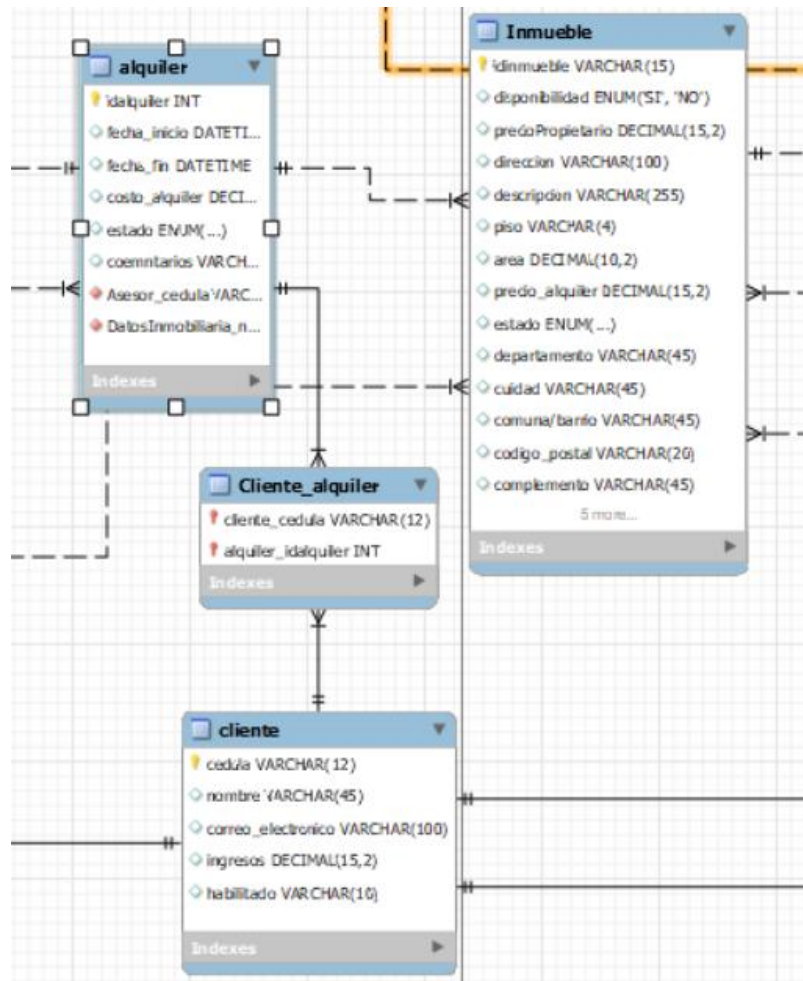
Paso a seguir definir los datos adicionales que iba a tener el modelo relacional realizado en workbench, representación de la base de datos echa en workbench



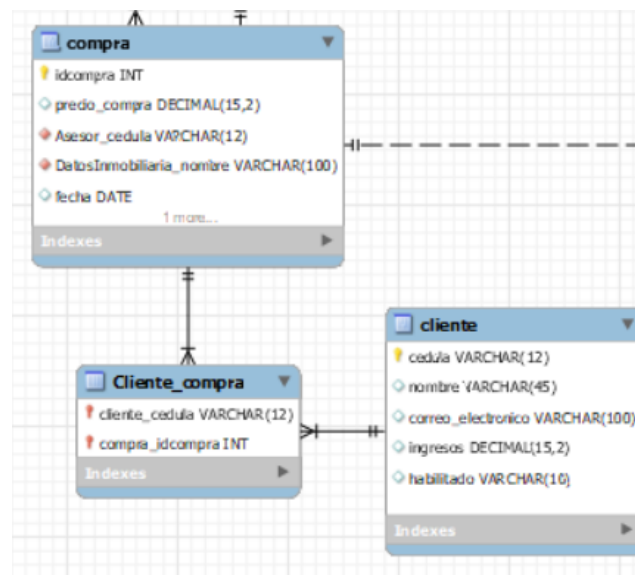
Relación propietario inmueble un propietario puede tener muchos inmuebles y un inmueble pertenece a un propietario en este caso tomando como premisa que el propietario es único dueño. La tabla foto tiene ellas fotos guardas de las propiedades en la tabla teléfono.



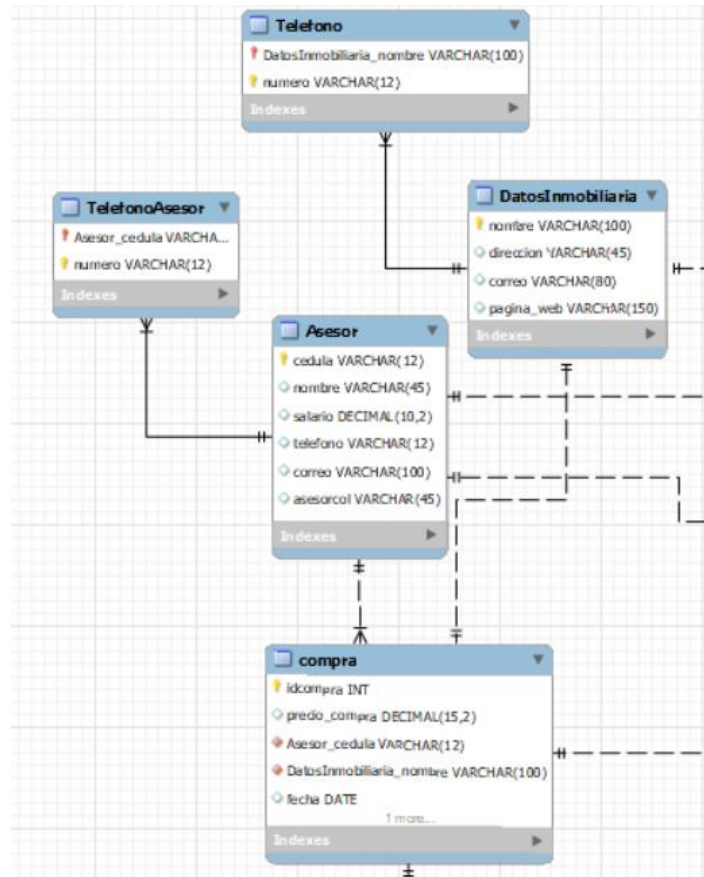
Relación cliente e info_visita nos genera una nueva tabla intermedia la cual lleva el id de las dos tablas relacionas a esta tabla se llama cliente_infor_visita donde un cliente puede visitar muchos inmuebles y un inmueble puede ser visitado por muchos clientes las tablas adicionales de clientes es Telefono_cliente



Relación entre tabla cliente e inmueble y alquiler donde esta relación nos crea una tabla intermedia entre el cliente_alquiler donde un cliente puede alquilar uno o muchos inmuebles y un alquiler pertenece a un cliente



Relación entre un cliente y una compra donde el cliente tendrá sus atributos relacionados y una compra tendrá los datos de la compra donde un cliente puede comprar 1 o muchos inmuebles y una compra puede ser realizada por muchos accionistas o participantes tomando como premisa que una compra puede ser comprada por varios clientes como en compra compartida, donde dicha relación nos crea una tabla intermedia por ser muchos a muchos que se llama cliente_compra.



Relación entre la compra que hace el cliente y es asesorada por un vendedor y que al momento de realizar la compra se registran los datosInmobiliaria donde la cual tenemos una tabla con toda la información de la empresa.

SQL de la creación de la base de datos

```
14 ● CREATE SCHEMA IF NOT EXISTS `Inmobiliaria` DEFAULT CHARACTER SET utf8 ;
15 ● USE `Inmobiliaria` ;
16 -- -----
17 -- Table `Inmobiliaria`.`cliente`
18 -- -----
19 ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`cliente` (
20     `cedula` VARCHAR(12) NOT NULL,
21     `nombre` VARCHAR(45) NULL,
22     `correo_electronico` VARCHAR(100) NULL,
23     `ingresos` DECIMAL(15,2) NULL,
24     `habilitado` VARCHAR(10) NULL,
25     PRIMARY KEY (`cedula`))
26 ENGINE = InnoDB;
27
28 -- -----
29 -- Table `Inmobiliaria`.`Asesor`
30 -- -----
31 ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`Asesor` (
32     `cedula` VARCHAR(12) NOT NULL,
33     `nombre` VARCHAR(45) NULL,
34     `salario` DECIMAL(10,2) NULL,
35     `correo` VARCHAR(100) NULL,
36     PRIMARY KEY (`cedula`))
37 ENGINE = InnoDB;
38
39 -- -----
40 -- Table `Inmobiliaria`.`DatosInmobiliaria`
41 -- -----
42 ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`DatosInmobiliaria` (
43     `nombre` VARCHAR(100) NOT NULL,
44     `direccion` VARCHAR(45) NULL,
45     `correo` VARCHAR(80) NULL,
46     `pagina_web` VARCHAR(150) NULL,
47     PRIMARY KEY (`nombre`))
48 ENGINE = InnoDB;
```



```

51  -----
52  -- Table `Inmobiliaria`.`alquiler`
53  -----
54  ● CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`alquiler` (
55      `idalquiler` INT NOT NULL,
56      `fecha_inicio` DATETIME NULL,
57      `fecha_fin` DATETIME NULL,
58      `costo_alquiler` DECIMAL(18,2) NULL,
59      `estado` ENUM('vigente', 'finalizado', 'renovado') NULL,
60      `comentarios` VARCHAR(255) NULL,
61      `Asesor_cedula` VARCHAR(12) NOT NULL,
62      `DatosInmobiliaria_nombre` VARCHAR(100) NOT NULL,
63      PRIMARY KEY (`idalquiler`),
64      INDEX `fk_alquiler_Asesor1_idx` (`Asesor_cedula` ASC) VISIBLE,
65      INDEX `fk_alquiler_DatosInmobiliaria1_idx` (`DatosInmobiliaria_nombre` ASC) VISIBLE,
66      CONSTRAINT `fk_alquiler_Asesor1`
67          FOREIGN KEY (`Asesor_cedula`)
68          REFERENCES `Inmobiliaria`.`Asesor` (`cedula`)
69          ON DELETE NO ACTION
70          ON UPDATE NO ACTION,
71      CONSTRAINT `fk_alquiler_DatosInmobiliaria1`
72          FOREIGN KEY (`DatosInmobiliaria_nombre`)
73          REFERENCES `Inmobiliaria`.`DatosInmobiliaria` (`nombre`)
74          ON DELETE NO ACTION
75          ON UPDATE NO ACTION)
76  ENGINE = InnoDB;

```

```

78  -----
79  -- Table `Inmobiliaria`.`info_visita`
80  -----
81  ● CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`info_visita` (
82      `idinfo_visita` INT NOT NULL,
83      `fecha_visita` DATETIME NOT NULL,
84      `fecha_fin` DATETIME NOT NULL,
85      `descripcion` VARCHAR(255) NULL,
86      `Asesor_cedula` VARCHAR(12) NOT NULL,
87      PRIMARY KEY (`idinfo_visita`),
88      INDEX `fk_info_visita_Asesor1_idx` (`Asesor_cedula` ASC) VISIBLE,
89      CONSTRAINT `fk_info_visita_Asesor1`
90          FOREIGN KEY (`Asesor_cedula`)
91          REFERENCES `Inmobiliaria`.`Asesor` (`cedula`)
92          ON DELETE NO ACTION
93          ON UPDATE NO ACTION)
94  ENGINE = InnoDB;
95

```

```

97  -----
98  -- Table `Inmobiliaria`.`compra`
99  -----
100 ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`compra` (
101      `idcompra` INT NOT NULL,
102      `precio_compra` DECIMAL(15,2) NULL,
103      `Asesor_cedula` VARCHAR(12) NOT NULL,
104      `DatosInmobiliaria_nombre` VARCHAR(100) NOT NULL,
105      `fecha` DATE NULL,
106      `descripcion` VARCHAR(255) NULL,
107      PRIMARY KEY (`idcompra`),
108      INDEX `fk_compra_Asesor1_idx` (`Asesor_cedula` ASC) VISIBLE,
109      INDEX `fk_compra_DatosInmobiliaria1_idx` (`DatosInmobiliaria_nombre` ASC) VISIBLE,
110      CONSTRAINT `fk_compra_Asesor1`
111          FOREIGN KEY (`Asesor_cedula`)
112          REFERENCES `Inmobiliaria`.`Asesor` (`cedula`)
113          ON DELETE NO ACTION
114          ON UPDATE NO ACTION,
115      CONSTRAINT `fk_compra_DatosInmobiliaria1`
116          FOREIGN KEY (`DatosInmobiliaria_nombre`)
117          REFERENCES `Inmobiliaria`.`DatosInmobiliaria` (`nombre`)
118          ON DELETE NO ACTION
119          ON UPDATE NO ACTION)
120  ENGINE = InnoDB;
121
122
123  -----
124  -- Table `Inmobiliaria`.`propietario`
125  -----
126 ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`propietario` (
127      `cedula` VARCHAR(12) NOT NULL,
128      `nombre` VARCHAR(45) NULL,
129      `correo_electronico` VARCHAR(100) NULL,
130      PRIMARY KEY (`cedula`))
131  ENGINE = InnoDB;

```

```

134  -----
135  -- Table `Inmobiliaria`.`Inmueble`
136  -----
137  ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`Inmueble` (
138      `idinmueble` VARCHAR(15) NOT NULL,
139      `disponibilidad` ENUM('SI', 'NO') NULL,
140      `precioPropietario` DECIMAL(15,2) NULL,
141      `direccion` VARCHAR(100) NULL,
142      `descripcion` VARCHAR(255) NULL,
143      `piso` VARCHAR(4) NULL,
144      `area` DECIMAL(10,2) NULL,
145      `precio_alquiler` DECIMAL(15,2) NULL,
146      `estado` ENUM('nuevo', 'segunda', 'remodelado') NULL,
147      `departamento` VARCHAR(45) NULL,
148      `ciudad` VARCHAR(45) NULL,
149      `comuna/barrio` VARCHAR(45) NULL,
150      `codigo_postal` VARCHAR(20) NULL,
151      `complemento` VARCHAR(45) NULL,
152      `precio_venta` DECIMAL(15,2) NULL,
153      `alquiler_idalquiler` INT NOT NULL,
154      `info_visita_idinfo_visita` INT NOT NULL,
155      `compra_idcompra` INT NOT NULL,
156      `propietario_cedula` VARCHAR(12) NOT NULL,
157      PRIMARY KEY (`idinmueble`),
158      INDEX `fk_Inmueble_alquiler1_idx` (`alquiler_idalquiler` ASC) VISIBLE,
159      INDEX `fk_Inmueble_info_visita1_idx` (`info_visita_idinfo_visita` ASC) VISIBLE,
160      INDEX `fk_Inmueble_compra1_idx` (`compra_idcompra` ASC) VISIBLE,
161      INDEX `fk_Inmueble_propietario1_idx` (`propietario_cedula` ASC) VISIBLE,
162      CONSTRAINT `fk_Inmueble_alquiler1`
163          FOREIGN KEY (`alquiler_idalquiler`)
164          REFERENCES `Inmobiliaria`.`alquiler` (`idalquiler`)

165          ON DELETE NO ACTION
166          ON UPDATE NO ACTION,
167      CONSTRAINT `fk_Inmueble_info_visita1`
168          FOREIGN KEY (`info_visita_idinfo_visita`)
169          REFERENCES `Inmobiliaria`.`info_visita` (`idinfo_visita`)
170          ON DELETE NO ACTION
171          ON UPDATE NO ACTION,
172      CONSTRAINT `fk_Inmueble_compra1`
173          FOREIGN KEY (`compra_idcompra`)
174          REFERENCES `Inmobiliaria`.`compra` (`idcompra`)
175          ON DELETE NO ACTION
176          ON UPDATE NO ACTION,
177      CONSTRAINT `fk_Inmueble_propietario1`
178          FOREIGN KEY (`propietario_cedula`)
179          REFERENCES `Inmobiliaria`.`propietario` (`cedula`)
180          ON DELETE NO ACTION
181          ON UPDATE NO ACTION)
182  ENGINE = InnoDB;

```

```

185      -----
186      -- Table `Inmobiliaria`.`TelefonoAsesor`
187      -----
188  ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`TelefonoAsesor` (
189      `cedula_asesor` VARCHAR(12) NOT NULL,
190      `numero` VARCHAR(12) NOT NULL,
191      INDEX `fk_TelefonoAsesor_Asesor1_idx` (`cedula_asesor` ASC) VISIBLE,
192      PRIMARY KEY (`cedula_asesor`, `numero`),
193      CONSTRAINT `fk_cedula_asesor`
194          FOREIGN KEY (`cedula_asesor`)
195          REFERENCES `Inmobiliaria`.`Asesor` (`cedula`)
196          ON DELETE NO ACTION
197          ON UPDATE NO ACTION)
198      ENGINE = InnoDB;
199
200
201      -----
202      -- Table `Inmobiliaria`.`TelefonoCliente`
203      -----
204  ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`TelefonoCliente` (
205      `cliente_cedula` VARCHAR(12) NOT NULL,
206      `numero` VARCHAR(12) NOT NULL,
207      PRIMARY KEY (`cliente_cedula`, `numero`),
208      INDEX `fk_Telefono_cliente1_idx` (`cliente_cedula` ASC) VISIBLE,
209      CONSTRAINT `fk_Telefono_cliente1`
210          FOREIGN KEY (`cliente_cedula`)
211          REFERENCES `Inmobiliaria`.`cliente` (`cedula`)
212          ON DELETE NO ACTION
213          ON UPDATE NO ACTION)
214      ENGINE = InnoDB;

```

```

218 -- Table `Inmobiliaria`.`Foto`
219 -----
220 ● ⊖ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`Foto` (
221     `foto` VARCHAR(100) NOT NULL,
222     `inmueble_idinmueble` VARCHAR(15) NOT NULL,
223     PRIMARY KEY (`foto`),
224     INDEX `fk_Foto_inmueble1_idx` (`inmueble_idinmueble` ASC) VISIBLE,
225     CONSTRAINT `fk_Foto_inmueble1`
226         FOREIGN KEY (`inmueble_idinmueble`)
227         REFERENCES `Inmobiliaria`.`Inmueble` (`idinmueble`)
228         ON DELETE NO ACTION
229         ON UPDATE NO ACTION)
230 ENGINE = InnoDB;
231
232
233 -----
234 -- Table `Inmobiliaria`.`Telefono`
235 -----
236 ● ⊖ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`Telefono` (
237     `propietario_cedula` VARCHAR(12) NOT NULL,
238     `numero` VARCHAR(45) NOT NULL,
239     INDEX `fk_Telefono_propietario1_idx` (`propietario_cedula` ASC) VISIBLE,
240     PRIMARY KEY (`propietario_cedula`, `numero`),
241     CONSTRAINT `fk_Telefono_propietario1`
242         FOREIGN KEY (`propietario_cedula`)
243         REFERENCES `Inmobiliaria`.`propietario` (`cedula`)
244         ON DELETE NO ACTION
245         ON UPDATE NO ACTION)
246 ENGINE = InnoDB;

```



```

249  -----
250  -- Table `Inmobiliaria`.`Telefono`
251  -----
252  ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`Telefono` (
253      `propietario_cedula` VARCHAR(12) NOT NULL,
254      `numero` VARCHAR(45) NOT NULL,
255      INDEX `fk_Telefono_propietario1_idx` (`propietario_cedula` ASC) VISIBLE,
256      PRIMARY KEY (`propietario_cedula`, `numero`),
257      CONSTRAINT `fk_Telefono_propietario1`
258          FOREIGN KEY (`propietario_cedula`)
259          REFERENCES `Inmobiliaria`.`propietario` (`cedula`)
260          ON DELETE NO ACTION
261          ON UPDATE NO ACTION)
262  ENGINE = InnoDB;
263  -----
264  -- Table `Inmobiliaria`.`Cliente_compra`
265  -----
266  ● ○ CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`Cliente_compra` (
267      `cliente_cedula` VARCHAR(12) NOT NULL,
268      `compra_idcompra` INT NOT NULL,
269      INDEX `fk_Cliente_compra_cliente1_idx` (`cliente_cedula` ASC) VISIBLE,
270      INDEX `fk_Cliente_compra_compra1_idx` (`compra_idcompra` ASC) VISIBLE,
271      PRIMARY KEY (`cliente_cedula`, `compra_idcompra`),
272      CONSTRAINT `fk_Cliente_compra_cliente1`
273          FOREIGN KEY (`cliente_cedula`)
274          REFERENCES `Inmobiliaria`.`cliente` (`cedula`)
275          ON DELETE NO ACTION
276          ON UPDATE NO ACTION,
277      CONSTRAINT `fk_Cliente_compra_compra1`
278          FOREIGN KEY (`compra_idcompra`)
279          REFERENCES `Inmobiliaria`.`compra` (`idcompra`)
280          ON DELETE NO ACTION
281          ON UPDATE NO ACTION)
282  ENGINE = InnoDB;

```

```

285  -----
286  -- Table `Inmobiliaria`.`Cliente_alquiler`
287  -----
288  ● CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`Cliente_alquiler` (
289      `cliente_cedula` VARCHAR(12) NOT NULL,
290      `alquiler_idalquiler` INT NOT NULL,
291      PRIMARY KEY (`cliente_cedula`, `alquiler_idalquiler`),
292      INDEX `fk_Cliente_alquiler_cliente1_idx` (`cliente_cedula` ASC) VISIBLE,
293      INDEX `fk_Cliente_alquiler_alquiler1_idx` (`alquiler_idalquiler` ASC) VISIBLE,
294      CONSTRAINT `fk_Cliente_alquiler_cliente1`
295          FOREIGN KEY (`cliente_cedula`)
296          REFERENCES `Inmobiliaria`.`cliente` (`cedula`)
297          ON DELETE NO ACTION
298          ON UPDATE NO ACTION,
299      CONSTRAINT `fk_Cliente_alquiler_alquiler1`
300          FOREIGN KEY (`alquiler_idalquiler`)
301          REFERENCES `Inmobiliaria`.`alquiler` (`idalquiler`)
302          ON DELETE NO ACTION
303          ON UPDATE NO ACTION)
304  ENGINE = InnoDB;
305  ~~~

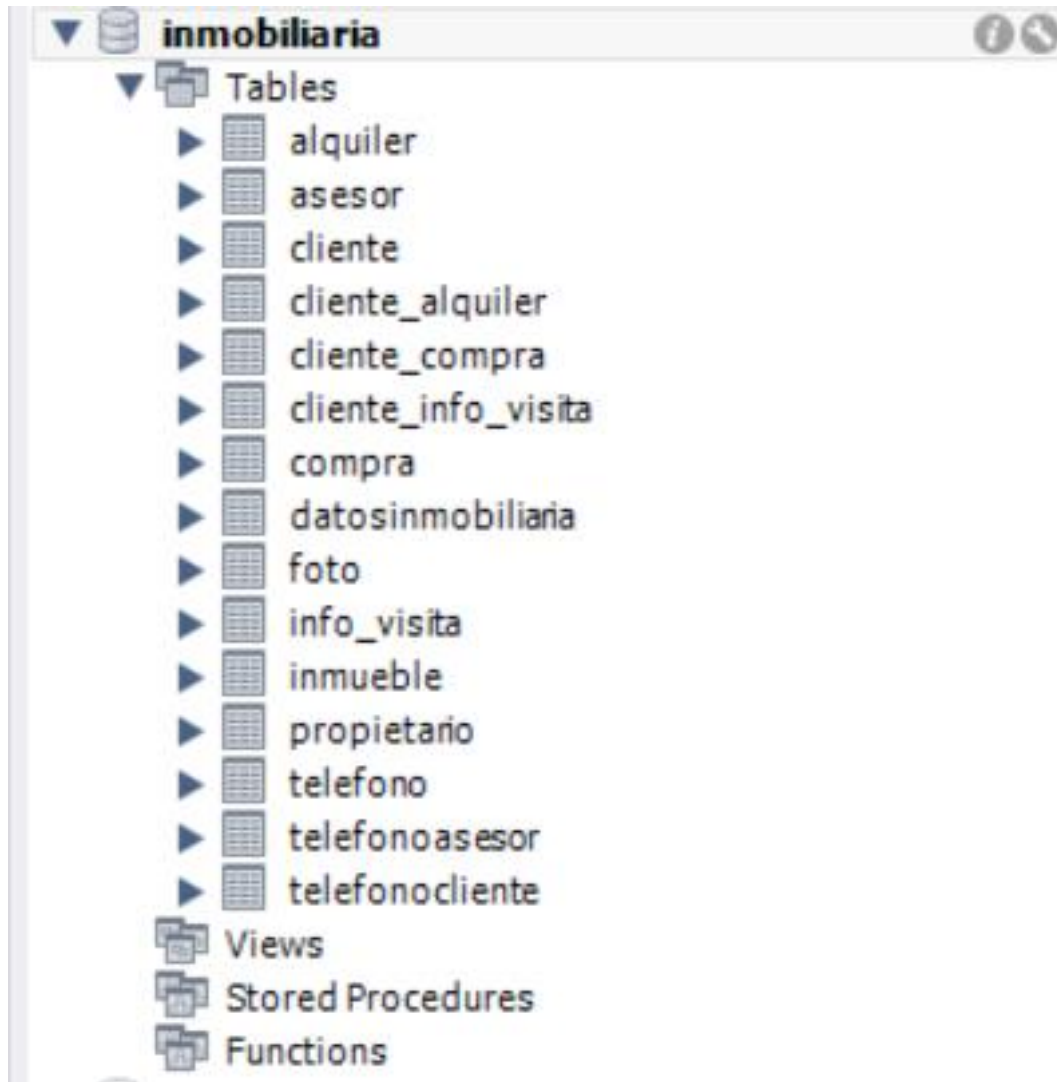
```

```

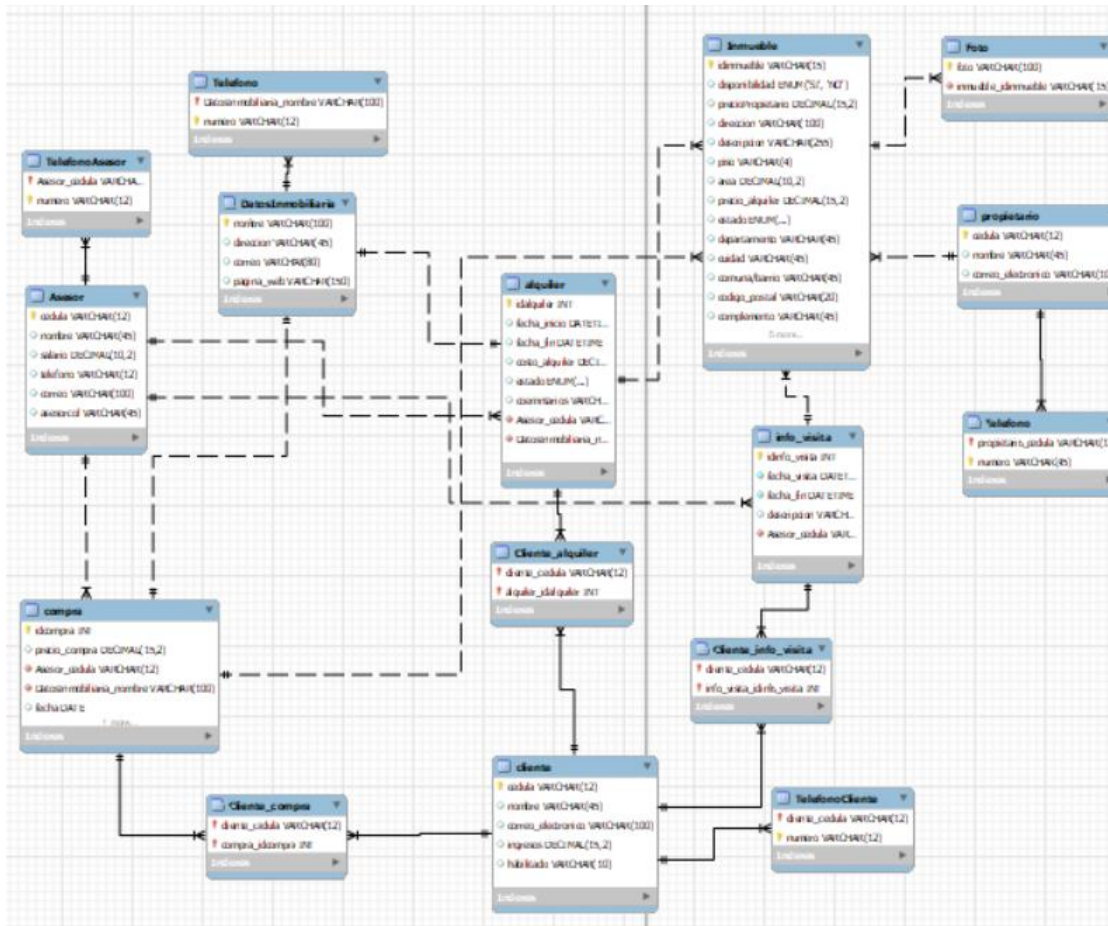
307  -----
308  -- Table `Inmobiliaria`.`Cliente_info_visita`
309  -----
310  ● CREATE TABLE IF NOT EXISTS `Inmobiliaria`.`Cliente_info_visita` (
311      `cliente_cedula` VARCHAR(12) NOT NULL,
312      `info_visita_idinfo_visita` INT NOT NULL,
313      INDEX `fk_Cliente_info_visita_cliente1_idx` (`cliente_cedula` ASC) VISIBLE,
314      INDEX `fk_Cliente_info_visita_info_visita1_idx` (`info_visita_idinfo_visita` ASC) VISIBLE,
315      PRIMARY KEY (`cliente_cedula`, `info_visita_idinfo_visita`),
316      CONSTRAINT `fk_Cliente_info_visita_cliente1`
317          FOREIGN KEY (`cliente_cedula`)
318          REFERENCES `Inmobiliaria`.`cliente` (`cedula`)
319          ON DELETE NO ACTION
320          ON UPDATE NO ACTION,
321      CONSTRAINT `fk_Cliente_info_visita_info_visita1`
322          FOREIGN KEY (`info_visita_idinfo_visita`)
323          REFERENCES `Inmobiliaria`.`info_visita` (`idinfo_visita`)
324          ON DELETE NO ACTION
325          ON UPDATE NO ACTION)
326  ENGINE = InnoDB;
327
328
329  ● SET SQL_MODE=@OLD_SQL_MODE;
330  ● SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
331  ● SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

Representación de la base de datos en Workbench



Modelo Relacional Completado



Tomando en cuenta las relaciones anteriormente descritas la base de datos quedaría así.

Ingreso de los bases a la base de datos

Tabla info_visita

1 • `SELECT * FROM inmobiliaria.info_visita;`

idinfo_visita	fecha_visita	fecha_fin	descripcion	Asesor_cedula
1	01-01-202...	01-01-2...	visita de a...	1010101010
2	02-01-202...	02-01-2...	visita para...	2020202020
3	03-01-202...	03-01-2...	Reunión d...	4040404040
4	04-01-202...	04-01-2...	Reunión d...	2020202020
5	05-01-202...	05-01-2...	Reunión d...	3030303030
6	NULL	NULL	NULL	NULL

Tabla alquiler

Limit to 2000 rows

```
1 • SELECT * FROM inmobiliaria.alquiler;
```

idalquiler	fecha_inicio	fecha_fin	costo_alquiler	estado	coemntarios	Asesor_cedula	DatosInmobiliaria_nombre
1	2023-02-10 12:30:15	2023-02-10 12:30:15	50000.00	vigente	MUY BUEN ESTADO	1010101010	Inmo1
2	2020-01-01 00:00:00	2020-01-01 23:59:59	1000000.00	finalizado	PERFECTO ESTADO	2020202020	Inmo1
3	2021-03-01 10:30:00	2021-03-01 12:15:00	1542500.00	vigente	NUEVO	3030303030	Inmo1
4	2022-05-15 08:00:00	2022-05-15 17:00:00	500000.00	finalizado	PARA REMODELAR	4040404040	Inmo1
5	2022-12-31 00:00:00	2022-12-31 23:59:59	600000.00	vigente	SOLO EL LOTE	5050505050	Inmo1
6	2023-02-10 09:00:00	2023-02-10 17:30:00	800000.00	vigente	NUEVO	1010101010	Inmo1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Datos inmobiliaria

Limit to 2000 rows

```
1 • SELECT * FROM inmobiliaria.datosinmobiliaria;
```

nombre	direccion	correo	pagina_web
Inmo1	Dirección 1	inmo1@example.com	www.inmo1.com
Inmo1	Dirección 2	inmo2@example.com	www.inmo2.com
Inmo1	Dirección 3	inmo3@example.com	www.inmo3.com
Inmo4	Dirección 4	inmo4@example.com	www.inmo4.com
Inmo5	Dirección 5	inmo5@example.com	www.inmo5.com
*	NULL	NULL	NULL

Datos Asesor

Limit to 2000 rows

```
1 • SELECT * FROM inmobiliaria.asesor;
```

cedula	nombre	salario	correo
1010101010	Luis Garcia	5000.00	luis@example.com
2020202020	Miguel Hernandez	6000.00	miguel@example.com
3030303030	Karla Rodriguez	7000.00	karla@example.com
4040404040	Luz Maria	800.00	luz@example.com
5050505050	Manuel Perez	9000.00	manuel@example.com
*	NULL	NULL	NULL

Datos Telefono Asesor

1 • `SELECT * FROM inmobiliaria.telefonoasesor;`

cedula_asesor	numero
1010101010	3138514587
1020	165151
2020202020	32015485
3030303030	31057841
4040404040	31087984
NULL	NULL

Datos Propietario

1 • `SELECT * FROM inmobiliaria.propietario;`

cedula	nombre	correo_electronico
123-456-789	Juan Pérez	juanperez@email.com
234-567-890	Maria García	mariagarcia@email.com
345-678-901	Carlos Sánchez	carlossanchez@email.com
456-789-012	Ana Rodríguez	anarodriguez@email.com
567-890-123	Luis Martínez	luismartinez@email.com

Datos Telefono Propietario

1 • `SELECT * FROM inmobiliaria.telefono;`

propietario_cedula	numero
123-456-789	147252752
234-567-890	2578414724
345-678-901	87147141
456-789-012	31315414
567-890-123	3169154
NULL	NULL

Datos Cliente

Limit to 2000 rows

```
1 • SELECT * FROM inmobiliaria.cliente;
```

Result Grid

cedula	nombre	correo_electronico	ingresos	habilitado
0123456789	Pedro Martinez	pedro@example.com	60000.00	no
1098765432	Maria Rodriguez	maria@example.com	40000.00	si
1234567890	Juan Perez	juan@example.com	50000.00	si
6789012345	Jorge Martinez	jorge@example.com	45000.00	no
9876543210	Sofia Gomez	sofia@example.com	55000.00	si
NULL	NULL	NULL	NULL	NULL

Datos Telefono cliente

Limit to 2000 rows

```
1 • SELECT * FROM inmobiliaria.telefonocliente;
```

Result Grid

cliente_cedula	numero
0123456789	35145...
1098765432	1581451
1234567890	6126961
6789012345	6169144
9876543210	3251481
NULL	5811848
NULL	NULL

Datos Cliente Alquiler

Limit to 2000 rows

```
1 • SELECT * FROM inmobiliaria.cliente_alquiler;
```

Result Grid

cliente_cedula	alquiler_idalquiler
0123456789	1
1098765432	2
1234567890	3
6789012345	4
9876543210	5

Datos Inmueble

```
1 • SELECT * FROM inmobiliaria.inmueble;
```

	idinmueble	disponibilidad	precioPropietario	direccion	descripcion	piso	area	precio_alquiler	estado	departamento	cuidad	comuna/barrio
	1	SI	100000.00	CALLE 23	CASA GRANDE	1	50.00	20000.00	nuevo	NORTE DE SANTANDER	CUCUTA	COMUNA 6
	2	SI	20000.00	Avenida de los Su...	Casa de campo encantadora co...	2	48.00	1000000.00	nuevo	antioquia	medellin	13
	3	NO	50000.00	Plaza de los Dese...	Espectacular loft con techos alt...	3	90.00	800000.00	nuevo	cundinamarca	bogota	1
	4	SI	7000.00	Paseo de la Ilusió...	Casa unifamiliar con jardín y pis...	2	100.00	700000.00	nuevo	norte de santander	cucuta	5
▶	A01	SI	100000.00	Calle de las Estrell...	Hermoso apartamento de 1 hab...	1	50.00	600000.00	nuevo	no aplica	medellin	6
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

codigo_postal	complemento	precio_venta	alquiler_idalquiler	info_visita_idinfo_visita	compra_idcompra	propietario_cedula
540010	COMPLE1	1200000.00	1	1	1	123-456-789
541000	complemento 1	500000.00	1	1	1	456-789-012
541000	complemento 1	80000.00	1	1	1	456-789-012
541000	complemento 1	700000.00	1	1	1	456-789-012
541000	complemento 1	20000000.00	1	1	1	123-456-789
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Datos Foto

```
1 • SELECT * FROM inmobiliaria.foto;
```

	foto	inmueble_idinmueble
▶	ssdvsvdsvdcv	1
	dsdcscdcasdc	2
	scfsvdcvscdc	2
	sdsdcvscdsd	3
	sdvsvdsvdvd	4
▶	NULL	NULL

Datos Compra

1 • `SELECT * FROM inmobiliaria.compra;`

	idcompra	precio_compra	Asesor_cedula	DatosInmobiliaria_nombre	fecha	descripcion
▶	1	200000.00	1010101010	Inmo1	2023-01-01	se compra la casa grande
	2	50000.00	1020	Inmo1	2023-01-01	se compra la casa grande
	3	500000.00	2020202020	Inmo1	2023-01-01	se compra la casa grande
	4	100000.00	3030303030	Inmo1	2023-01-01	se compra la casa grande
	5	350000.00	4040404040	Inmo1	2023-01-01	se compra la casa grande
*	NULL	NULL	NULL	NULL	NULL	NULL

Datos Cliente Info Vista

1 • `SELECT * FROM inmobiliaria.cliente_info_visita;`

	cliente_cedula	info_visita_idinfo_visita
	0123456789	1
	1098765432	2
	1234567890	3
	6789012345	4
	9876543210	5
▶*	NULL	

Datos Cliente compra

1 • `SELECT * FROM inmobiliaria.cliente_compra;`

	cliente_cedula	compra_idcompra
▶	0123456789	1
	1098765432	2
	1234567890	3
	6789012345	4
	9876543210	5
*	NULL	NULL

¿Qué debería cambiar o agregar para incluir la renovación de contratos de alquiler?

Para incluir la renovación de contratos de alquiler, se podría agregar una columna o campo en la base de datos que registre la fecha de finalización del contrato de alquiler actual y la fecha en que se renueva.

¿Cómo podría controlar que la misma persona que compra un inmueble con identificador X y se lo entrega a la agencia Inmobiliaria SofkaU, NO pueda alquilar el inmueble con identificador X?

Para controlar que la misma persona no pueda alquilar el mismo inmueble con igual identificador X, se podría implementar que la verificación de duplicados en la base de datos. Cada vez que se agrega un nuevo contrato de alquiler, el sistema podría comparar los datos del propietario y del inquilino con los registros existentes en la base de datos y notificar si se encuentra una coincidencia. También se podría implementar una verificación adicional en el momento de renovar un contrato de alquiler, para asegurarse de que el propietario y el inquilino sean diferentes de los registrados en cualquier contrato anterior para el mismo inmueble.

Archivos adicionales en el repositorio.