

AÑO 2024

PROYECTO FINAL (G51)

HITO N° 1



Samir Alejandro Casis Valenzuela
ACADEMIA DESAFIO LATAM – GENERACION 51

INDICE

1.- DISEÑO INTERFAZ GRAFICA

2.- NAVEGACION ENTRE VISTAS

3.- DEPENDENCIAS

4.- DISEÑO DE TABLAS Y RELACIONES

5.- DOCUMENTACION

1. DISEÑO INTERFAZ GRÁFICA

PAGINA DE INICIO

LOGO (INICIO)

REGISTRO

INICIAR SESION

CARRUSEL

A-Z	V	PRECIO + o -	V	UBICACION	V
-----	---	--------------	---	-----------	---

IMAGEN PROPIEDAD

TITULO

CARACTERISTICA 1

CARACTERISTICA 2

VER MÁS

IMAGEN PROPIEDAD

TITULO

CARACTERISTICA 1

CARACTERISTICA 2

VER MÁS

IMAGEN PROPIEDAD

TITULO

CARACTERISTICA 1

CARACTERISTICA 2

VER MÁS

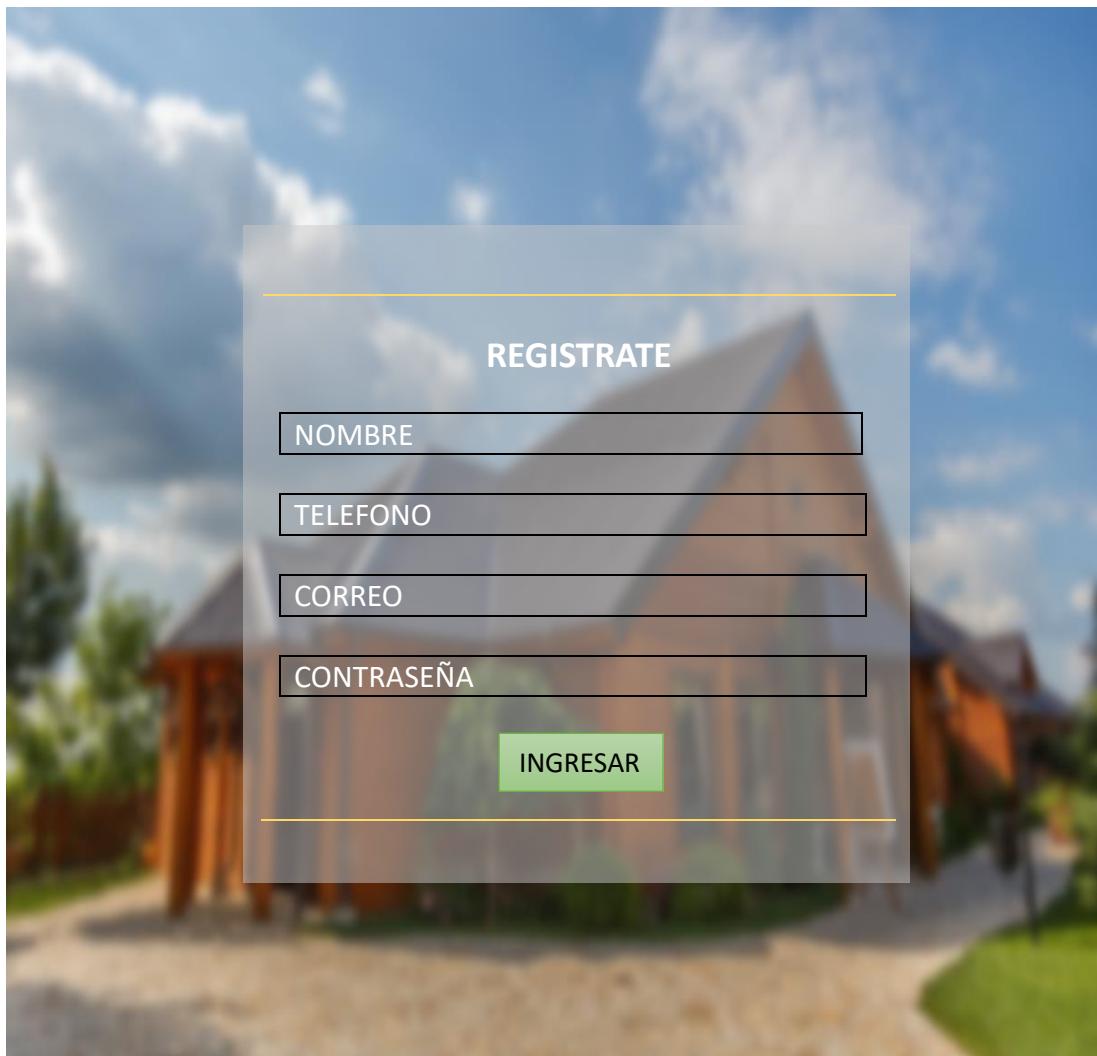
CONTACTO

-TELÉFONO

-CORREO

REGISTRO

LOGO (INICIO)



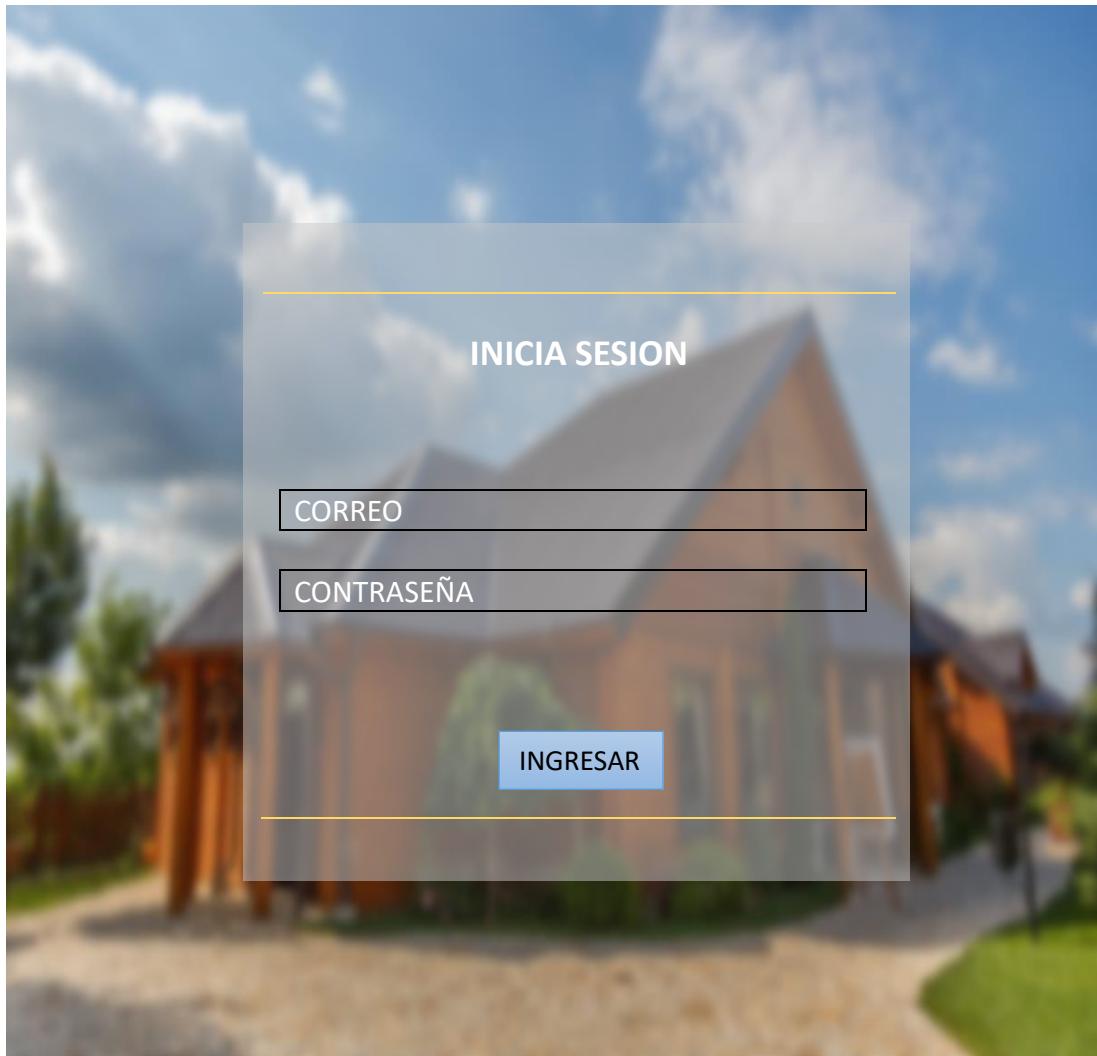
CONTACTO

-TELÉFONO

-CORREO

INICIAR SESION

LOGO (INICIO)



CONTACTO

-TELÉFONO

-CORREO

PERFIL PRIVADO (USUARIO LOGUEADO)

LOGO (INICIO)

BIENVENIDO #USUARIO

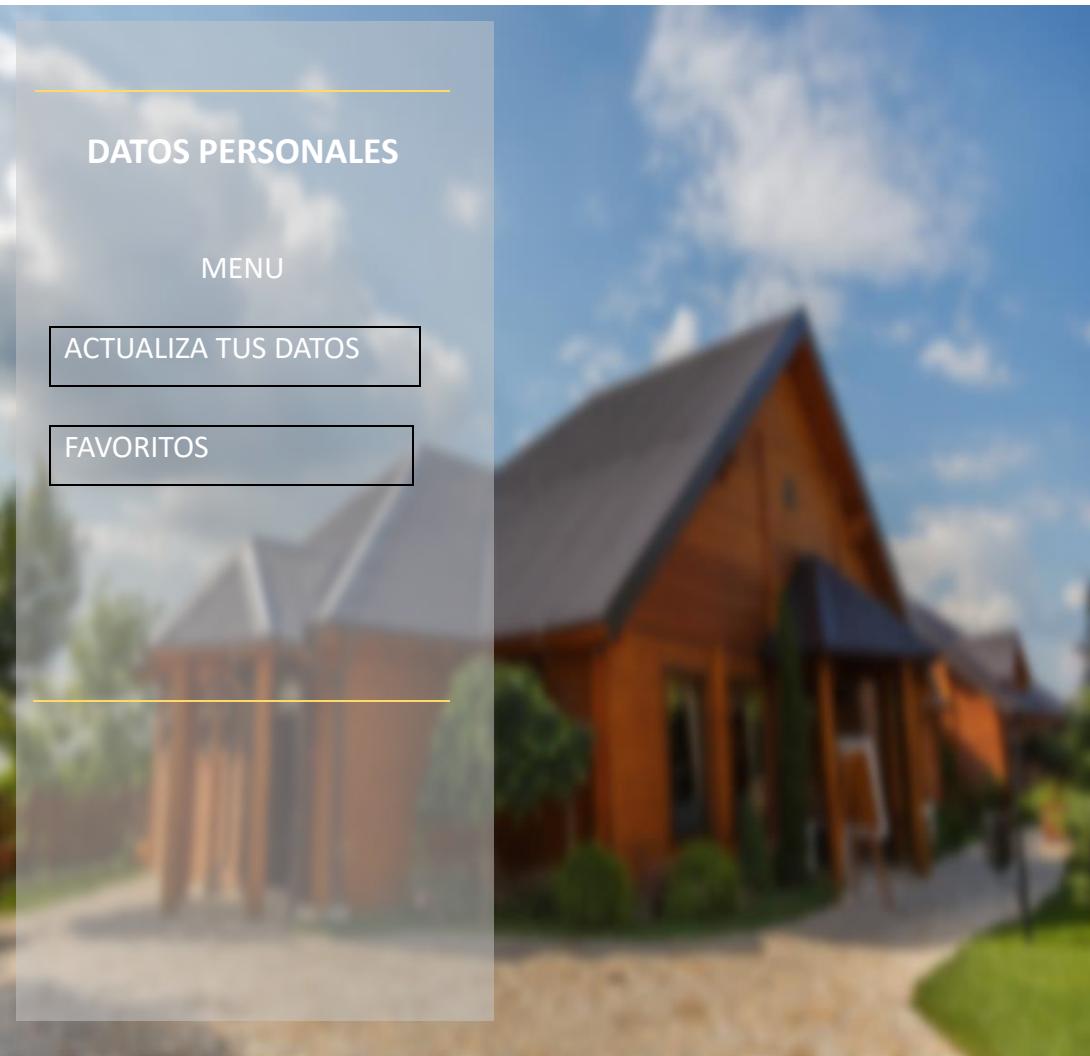
CARRITO 

DATOS PERSONALES

MENU

ACTUALIZA TUS DATOS

FAVORITOS



CONTACTO

-TELÉFONO

-CORREO

FAVORITOS (USUARIO LOGUEADO)

LOGO (INICIO)

BIENVENIDO #USUARIO

CARRITO 

FAVORITOS

IMAGEN PROPIEDAD

TITULO

CARACTERISTICA 1

CARACTERISTICA 2

VER MÁS

IMAGEN PROPIEDAD

TITULO

CARACTERISTICA 1

CARACTERISTICA 2

VER MÁS

CONTACTO

-TELÉFONO

-CORREO

PERFIL PRIVADO (ADMINISTRADOR LOGUEADO)

LOGO (INICIO)

ADMINISTRADOR

CARRITO 

DATOS PERSONALES

MENU

ACTUALIZA TUS DATOS

AGREGAR PROPIEDADES

ACTUALIZAR PROPIEDAD

CONTACTO

-TELÉFONO

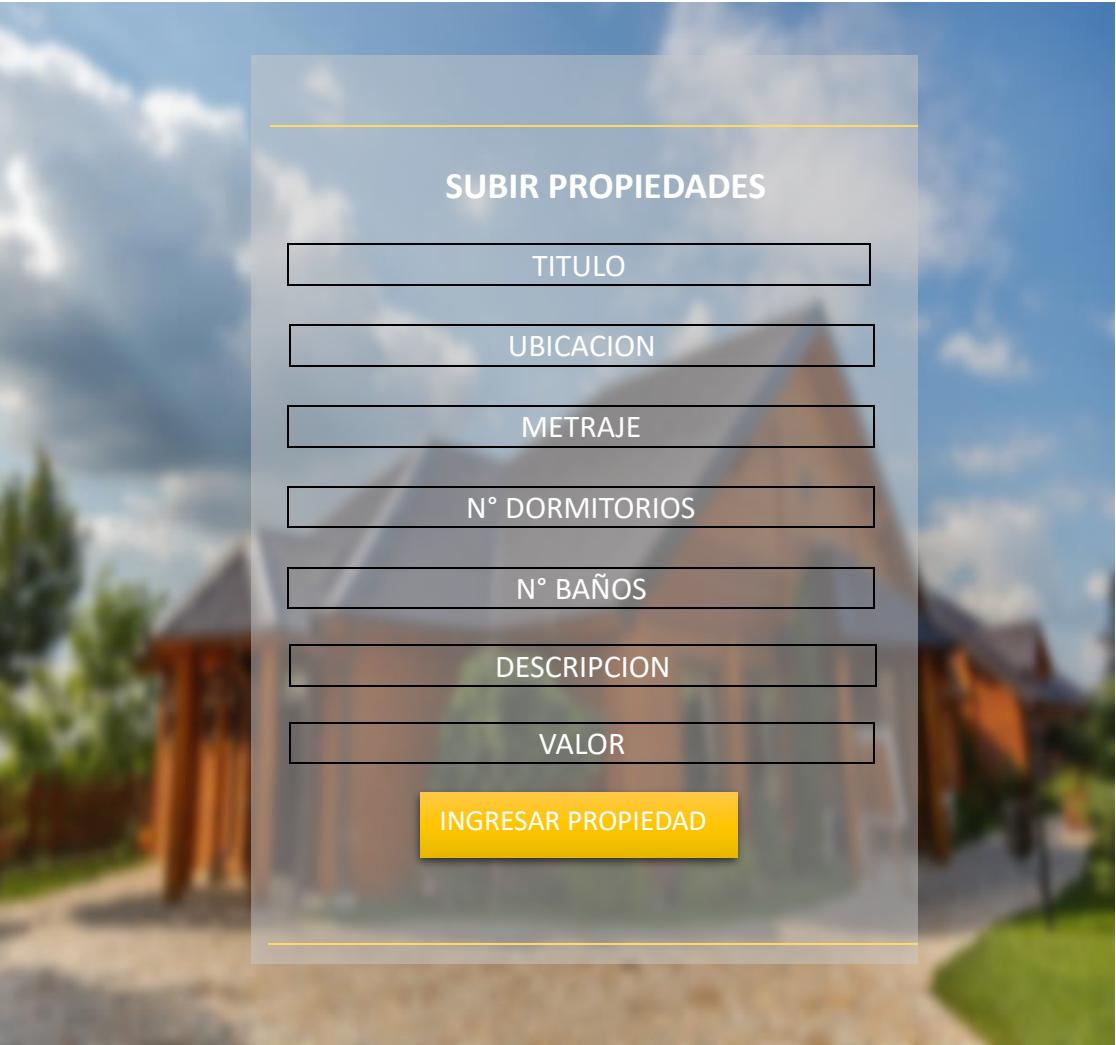
-CORREO

AGREGAR PROPIEDADES (ADMINISTRADOR LOGUEADO)

LOGO (INICIO)

BIENVENIDO #USUARIO //
ADMINISTRADOR

CARRITO 



SUBIR PROPIEDADES

TITULO

UBICACION

METRAJE

N° DORMITORIOS

N° BAÑOS

DESCRIPCION

VALOR

INGRESAR PROPIEDAD

CONTACTO

-TELÉFONO

-CORREO

ACTUALIZAR PROPIEDADES (ADMINISTRADOR LOGUEADO)

LOGO (INICIO)

BIENVENIDO ADMINISTRADOR

CARRITO 

ACTUALIZAR PROPIEDADES

TITULO

UBICACION

METRAJE

N° DORMITORIOS

N° BAÑOS

DESCRIPCION

VALOR

ACTUALIZAR PROPIEDAD

CONTACTO

-TELÉFONO

-CORREO

DETALLE PROPIEDADES (ADMINISTRADOR O USUARIO LOGUEADO)

LOGO (INICIO) ADMINISTRADOR CARRITO

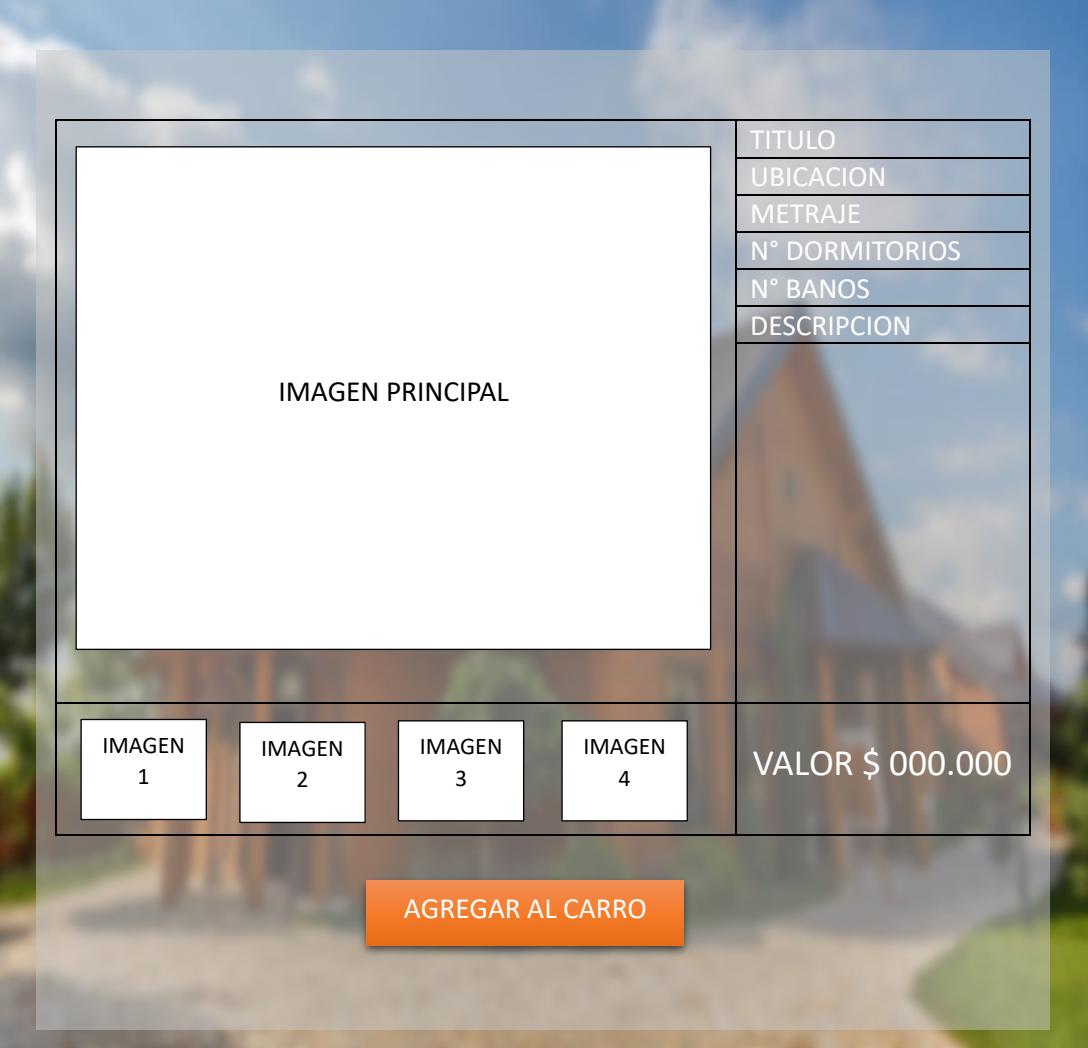


IMAGEN PRINCIPAL

IMAGEN 1	IMAGEN 2	IMAGEN 3	IMAGEN 4	VALOR \$ 000.000
----------	----------	----------	----------	------------------

AGREGAR AL CARRO

CONTACTO

-TELÉFONO

-CORREO

CARRO DE COMPRAS

LOGO (INICIO)

CARRITO 

DETALLE

PRODUCTO
1

DESCRIPCION

\$000.000

ELIMINAR

PRODUCTO
2

DESCRIPCION

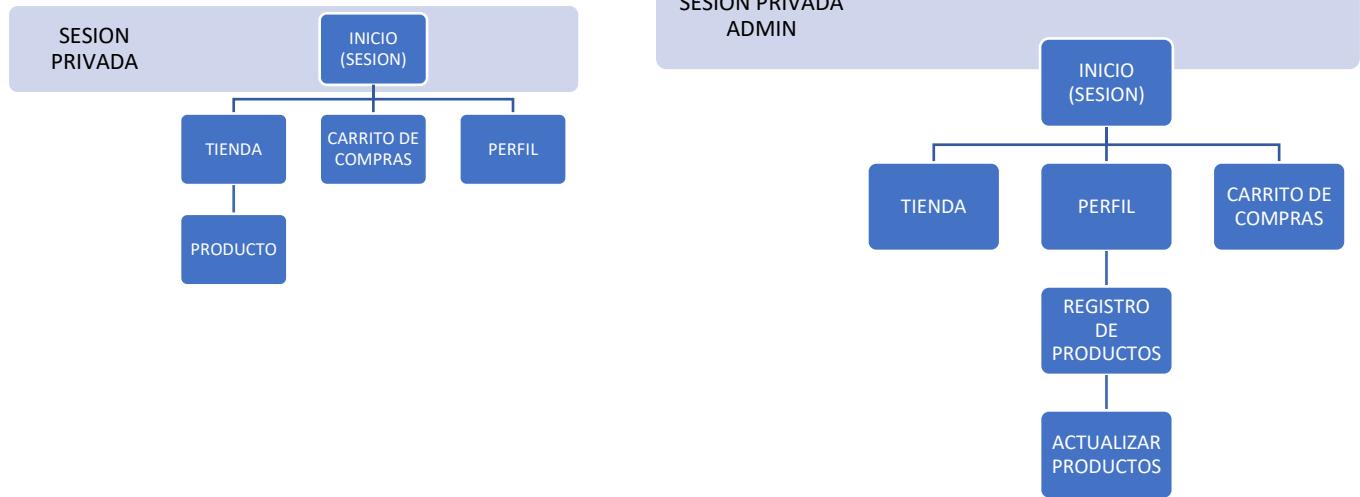
\$000.000

ELIMINAR

TOTAL: \$000.000

TERMINAR COMPRA

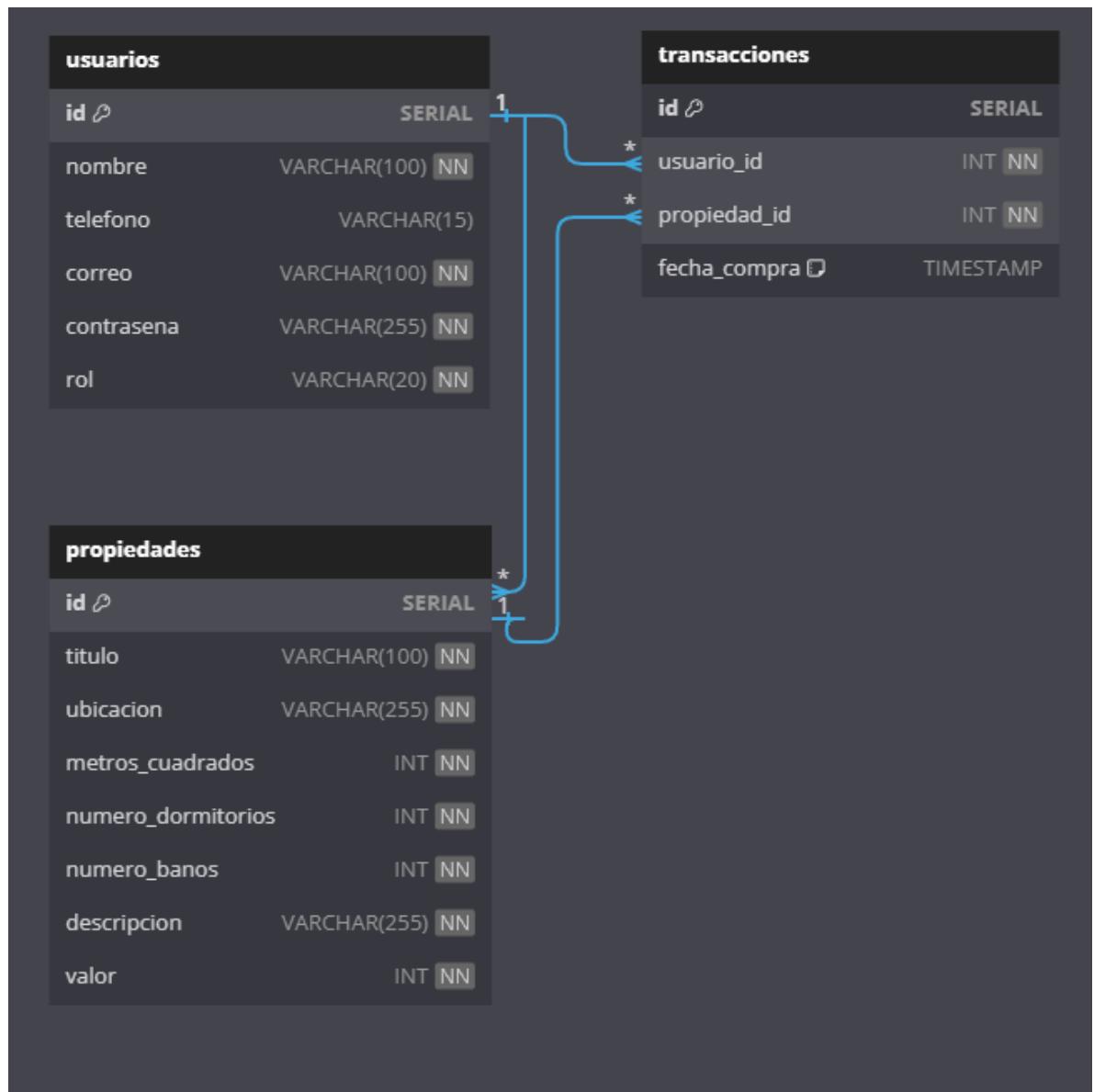
2. NAVEGACION ENTRE VISTAS



3. DEPENDENCIAS

- FRONTEND
 - BACKEND
- | | |
|----------------------|---------------|
| 1-) REACT | 1-) NODE |
| 2-) REACT ROUTER DOM | 2-) EXPRESS |
| 3-) AXIOS | 3-) JWT |
| 4-) REACT BOOTSTRAP | 4-) BCRYPT |
| 5-) CSS | 5-) POSTGRES |
| 6-) VITE | 6-) PG |
| 7-) SWEET ALERT 2 | 7-) PG-FORMAT |
| | 8-) SUPERTEST |
| | 9-) JEST |
| | 10-) CORS |
| | 11-) DOTENV |
| | 12-) SWAGGER |

4. DISEÑO DE TABLAS Y RELACIONES



DICIONARIO DE DATOS

TABLA	USUARIOS			
Columna	Tipo de Dato	Valor	Restricción	Descripción
id	SERIAL	Autoincremental	PRIMARY KEY	Identificador único del usuario.
nombre	VARCHAR(100)	-	NOT NULL	Nombre del usuario.
telefono	VARCHAR(15)	-		Número de teléfono del usuario.
correo	VARCHAR(100)	-	UNIQUE, NOT NULL	Correo electrónico único del usuario.
contrasena	VARCHAR(255)	-	NOT NULL	Contraseña hasheada del usuario.
rol	VARCHAR(20)	administrador, usuario	NOT NULL	Rol del usuario (administrador o usuario).

TABLA	PROPIEDADES			
Columna	Tipo de Dato	Valor	Restricción	Descripción
id	SERIAL	Autoincremental	PRIMARY KEY	Identificador único de la propiedad.
titulo	VARCHAR(100)	-	NOT NULL	Nombre de la propiedad
ubicacion	VARCHAR(255)	-	NOT NULL	Ubicación de la propiedad.
metros_cuadrados	INT	-	NOT NULL	Metros cuadrados de la propiedad.
numero_dormitorios	INT	-	NOT NULL	Número de dormitorios en la propiedad.
numero_banos	INT	-	NOT NULL	Número de baños en la propiedad.
descripcion	VARCHAR(255)	-	NOT NULL	Descripción en detalle de la propiedad
valor	INTEGER	-	NOT NULL	Valor de la propiedad en UF (número entero).

TABLA	VENTAS			
Columna	Tipo de Dato	Valor	Restricción	Descripción
id	SERIAL	Autoincremental	PRIMARY KEY	Identificador único de la transacción.
usuario_id	INT	-	NOT NULL, FOREIGN KEY (usuarios.id)	Identificador del usuario que realiza la compra.
propiedad_id	INT	-	NOT NULL, FOREIGN KEY (propiedades.id)	Identificador de la propiedad comprada.
fecha_compra	TIMESTAMP	-	DEFAULT CURRENT_TIMESTAMP	Fecha y hora de la compra.

5. DOCUMENTACION

CONFIGURACION DE SWAGGER

```
● ● ●
1 import swaggerJSDoc from 'swagger-jsdoc'
2 import swaggerUi from 'swagger-ui-express'
3
4 const options = {
5   definition: {
6     openapi: '3.1.0',
7     info: {
8       title: 'API Propiedades',
9       version: '1.0.0',
10      description: 'API de gestión de propiedades',
11    },
12    servers: [
13      {
14        url: 'http://localhost:5200/api/v1',
15        description: 'Servidor de prueba',
16      },
17    ],
18  },
19  apis: ['src/server/routes/docs/*.js'],
20 }
21
22 const specs = swaggerJSDoc(options)
23
24 export default (app) => {
25   app.use(
26     '/api/v1/docs',
27     swaggerUi.serve,
28     swaggerUi.setup(specs, {
29       explorer: true,
30       customCssUrl:
31         'https://cdn.jsdelivr.net/npm/swagger-ui-themes@3.0.0/themes/3.x/theme-flattop.css',
32     })
33   )
34 }
```

The screenshot shows the Swagger UI interface for the 'Propiedades' API. At the top, there's a navigation bar with the 'Swagger' logo, a search bar, and an 'Explore' button. Below the header, the title 'API Propiedades' is displayed with version '1.0.0' and specification 'OAS 3.1'. A subtitle 'API de gestión de propiedades' follows. In the main content area, under the 'Servers' section, the URL 'http://localhost:5200/api/v1 - Servidor de prueba' is listed. The main focus is the 'Propiedades' resource, which is described as 'La API de gestión de propiedades'. Below this, five operations are listed: a GET method for '/propiedades' (Retorna la lista de todas las propiedades), a POST method for '/propiedades' (Crear una nueva propiedad), a GET method for '/propiedades/{id}' (Obtener una propiedad por id), a PUT method for '/propiedades/{id}' (Actualizar una propiedad), and a DELETE method for '/propiedades/{id}' (Eliminar una propiedad). Each operation has a detailed description and a collapse/expand arrow. At the bottom left, there's a 'Schemas' section with a 'Property' entry and an 'Expand all object' link.

COMPONENTES EN SWAGGER

```
● ● ●
1  /**
2   * @swagger
3   * components:
4   *   schemas:
5   *     Property:
6   *       type: object
7   *       required:
8   *         - titulo
9   *         - ubicacion
10  *         - metros_cuadrados
11  *         - numero_dormitorios
12  *         - numero_banos
13  *         - descripcion
14  *         - valor
15  *       properties:
16  *         id:
17  *           type: string
18  *           description: El id autogenerado de la propiedad
19  *         titulo:
20  *           type: string
21  *           description: El título de la propiedad
22  *         ubicacion:
23  *           type: string
24  *           description: La ubicación de la propiedad
25  *         metros_cuadrados:
26  *           type: number
27  *           description: El tamaño de la propiedad en metros cuadrados
28  *         numero_dormitorios:
29  *           type: number
30  *           description: El número de dormitorios
31  *         numero_banos:
32  *           type: number
33  *           description: El número de baños
34  *         descripcion:
35  *           type: string
36  *           description: Una descripción de la propiedad
37  *         valor:
38  *           type: number
39  *           description: El valor de la propiedad
40  *         example:
41  *           id: 00123
42  *           titulo: "Hermosa Casa Familiar"
43  *           ubicacion: "Calle Transversal 1001"
44  *           metros_cuadrados: 150
45  *           numero_dormitorios: 3
46  *           numero_banos: 2
47  *           descripcion: "Una hermosa casa familiar ubicada en el centro de la ciudad."
48  *           valor: 3000
49  */
```

SWAGGER ESQUEMA

Schemas >

Property ^ Collapse all object

- id ^ Collapse all string**
El id autogenerado de la propiedad
- titulo* ^ Collapse all string**
El título de la propiedad
- ubicacion* ^ Collapse all string**
La ubicación de la propiedad
- metros_cuadrados* ^ Collapse all number**
El tamaño de la propiedad en metros cuadrados
- numero_dormitorios* ^ Collapse all number**
El número de dormitorios
- numero_banos* ^ Collapse all number**
El número de baños
- descripcion* ^ Collapse all string**
Una descripción de la propiedad
- valor* ^ Collapse all number**
El valor de la propiedad

Example

```
{"id":123,"titulo":"Hermosa Casa Familiar","ubicacion":"Calle Transversal 1001","metros_cuadrados":150,"numero_dormitorios":3,"numero_banos":2,"descripcion":"Una hermosa casa familiar ubicada en el centro de la ciudad.", "valor":3000}
```

SWAGGER DOCUMENTACION: GET / GET :ID

```
● ● ●

1  /**
2   * @swagger
3   * tags:
4   *   name: Propiedades
5   *   description: La API de gestión de propiedades
6   */
7
8  /**
9   * @swagger
10  * /propiedades:
11  *   get:
12  *     summary: Retorna la lista de todas las propiedades
13  *     tags: [Propiedades]
14  *     responses:
15  *       200:
16  *         description: La lista de las propiedades
17  *         content:
18  *           application/json:
19  *             schema:
20  *               type: array
21  *               items:
22  *                 $ref: '#/components/schemas/Property'
23  */
24
25 /**
26 * @swagger
27 * /propiedades/{id}:
28 *   get:
29 *     summary: Obtener una propiedad por id
30 *     tags: [Propiedades]
31 *     parameters:
32 *       - in: path
33 *         name: id
34 *         schema:
35 *           type: string
36 *           required: true
37 *           description: El id de la propiedad
38 *     responses:
39 *       200:
40 *         description: La descripción de la propiedad por id
41 *         content:
42 *           application/json:
43 *             schema:
44 *               $ref: '#/components/schemas/Property'
45 *       404:
46 *         description: La propiedad no fue encontrada
47 */
```

Propiedades La API de gestión de propiedades

^

GET /propiedades Retorna la lista de todas las propiedades

Parameters

No parameters

Responses

Code Description Links

200	La lista de las propiedades	No links
-----	-----------------------------	----------

Media type application/json ▾
Controls Accept header.

Example Value | Schema

```
[ { "id": 123, "titulo": "Hermosa Casa Familiar", "ubicacion": "Calle Transversal 1001", "metros_cuadrados": 150, "numero_dormitorios": 3, "numero_banos": 2, "descripcion": "Una hermosa casa familiar ubicada en el centro de la ciudad.", "valor": 3000 } ]
```

GET /propiedades/{id} Obtener una propiedad por id

Parameters

Name Description

id * required string (path)	El id de la propiedad id
--	-----------------------------

Responses

Code Description Links

200	La descripción de la propiedad por id	No links
-----	---------------------------------------	----------

Media type application/json ▾
Controls Accept header.

Example Value | Schema

```
{ "id": 123, "titulo": "Hermosa Casa Familiar", "ubicacion": "Calle Transversal 1001", "metros_cuadrados": 150, "numero_dormitorios": 3, "numero_banos": 2, "descripcion": "Una hermosa casa familiar ubicada en el centro de la ciudad.", "valor": 3000 }
```

404	La propiedad no fue encontrada	No links
-----	--------------------------------	----------

SWAGGER DOCUMENTACION: POST

```
1  /**
2   * @swagger
3   * /propiedades:
4   *   post:
5   *     summary: Crear una nueva propiedad
6   *     tags: [Propiedades]
7   *     requestBody:
8   *       required: true
9   *       content:
10  *         application/json:
11  *           schema:
12  *             type: object
13  *             properties:
14  *               property:
15  *                 $ref: '#/components/schemas/Property'
16  *             responses:
17  *               '201':
18  *                 description: Éxito
19  *                 content:
20  *                   application/json:
21  *                     schema:
22  *                       type: object
23  *                       properties:
24  *                         property:
25  *                           $ref: '#/components/schemas/Property'
26  *               '400':
27  *                 description: Error al crear la propiedad
28  */
```

POST /propiedades Crear una nueva propiedad ^

Parameters Try it out

No parameters

Request body required application/json ▼

Example Value | Schema

```
{  
  "property": {  
    "id": 123,  
    "titulo": "Hermosa Casa Familiar",  
    "ubicacion": "Calle Transversal 1001",  
    "metros_cuadrados": 150,  
    "numero_dormitorios": 3,  
    "numero_banos": 2,  
    "descripcion": "Una hermosa casa familiar ubicada en el centro de la ciudad.",  
    "valor": 3000  
  }  
}
```

Responses

Code	Description	Links
201	Éxito	No links
	Media type application/json ▼ Controls Accept header.	
	Example Value Schema	
	<pre>{ "property": { "id": 123, "titulo": "Hermosa Casa Familiar", "ubicacion": "Calle Transversal 1001", "metros_cuadrados": 150, "numero_dormitorios": 3, "numero_banos": 2, "descripcion": "Una hermosa casa familiar ubicada en el centro de la ciudad.", "valor": 3000 } }</pre>	
400	Error al crear la propiedad	No links

SWAGGER DOCUMENTACION: PUT

```
1  /**
2   * @swagger
3   * /propiedades/{id}:
4   *   put:
5   *     summary: Actualizar una propiedad
6   *     tags: [Propiedades]
7   *     parameters:
8   *       - in: path
9   *         name: id
10  *        required: true
11  *        schema:
12  *          type: string
13  *          description: El id de la propiedad
14  *        requestBody:
15  *          required: true
16  *          content:
17  *            application/json:
18  *              schema:
19  *                type: object
20  *                properties:
21  *                  property:
22  *                    $ref: '#/components/schemas/Property'
23  *      responses:
24  *        200:
25  *          description: Éxito
26  *          content:
27  *            application/json:
28  *              schema:
29  *                type: object
30  *                properties:
31  *                  property:
32  *                    $ref: '#/components/schemas/Property'
33  *        400:
34  *          description: Error al actualizar la propiedad
35  */
```

PUT /propiedades/{id} Actualizar una propiedad

Parameters

Name Description

id * required
string
(path)
El id de la propiedad
id

Request body required

application/json

Example Value | Schema

```
{
  "property": {
    "id": 123,
    "titulo": "Hermosa Casa Familiar",
    "ubicacion": "Calle Transversal 1001",
    "metros_cuadrados": 150,
    "numero_dormitorios": 3,
    "numero_banos": 2,
    "descripcion": "Una hermosa casa familiar ubicada en el centro de la ciudad.",
    "valor": 3000
  }
}
```

Responses

Code	Description	Links
200	Éxito	No links
	Media type application/json	
	Controls Accept header.	
	Example Value Schema	
	{ "property": { "id": 123, "titulo": "Hermosa Casa Familiar", "ubicacion": "Calle Transversal 1001", "metros_cuadrados": 150, "numero_dormitorios": 3, "numero_banos": 2, "descripcion": "Una hermosa casa familiar ubicada en el centro de la ciudad.", "valor": 3000 } }	
400	Error al actualizar la propiedad	No links

SWAGGER DOCUMENTACION: DELETE

```
 1  /**
 2   * @swagger
 3   * /propiedades/{id}:
 4   *   delete:
 5   *     summary: Eliminar una propiedad
 6   *     tags: [Propiedades]
 7   *     parameters:
 8   *       - in: path
 9   *         name: id
10   *         required: true
11   *         schema:
12   *           type: string
13   *           description: El id de la propiedad
14   *     responses:
15   *       204:
16   *         description: Exito
17   *       400:
18   *         description: Error al eliminar una propiedad
19   */
```

DELETE /propiedades/{id} Eliminar una propiedad

Parameters

Name	Description
id * required string (path)	El id de la propiedad <input type="text" value="id"/>

Responses

Code	Description	Links
204	Exito	No links
400	Error al eliminar una propiedad	No links