



Proyecto final:

Desarrollo Full Stack JavaScript

INTEGRANTES:

- Rodrigo González
- Leslie Reyes
- Matias Cuadros

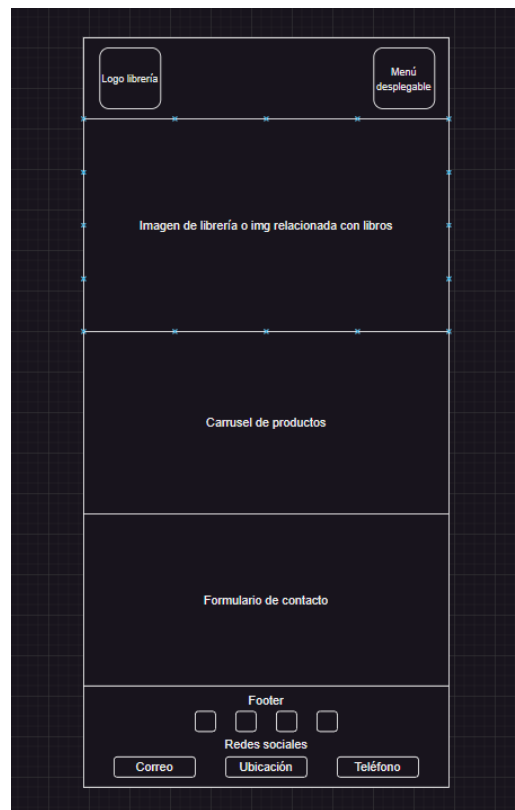
SANTIAGO - CHILE

2023

Hito 1

1. Diseño de la interfaz gráfica:

- Landing page:



- Galería de libros:



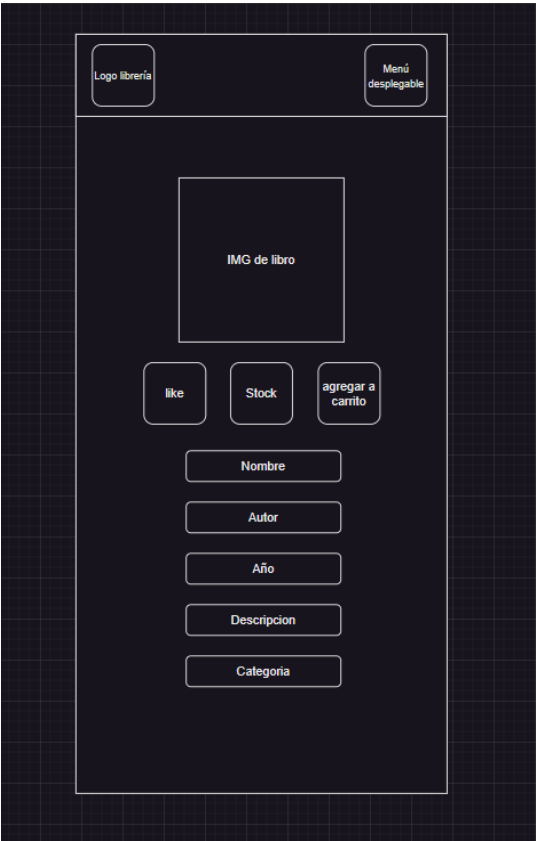
- Login:

The login form is a vertical rectangle with a dark blue background and a light blue grid. It features a header with a 'Logo librería' on the left and a 'Menú desplegable' on the right. The main content area is a rounded rectangle containing two input fields: 'Correo / Nombre usuario' and 'Contraseña'. Below these is a link that says '¿No tienes cuenta? Regístrate'. The footer contains four social media icons, the text 'Redes sociales', and three buttons: 'Correo', 'Ubicación', and 'Teléfono'.

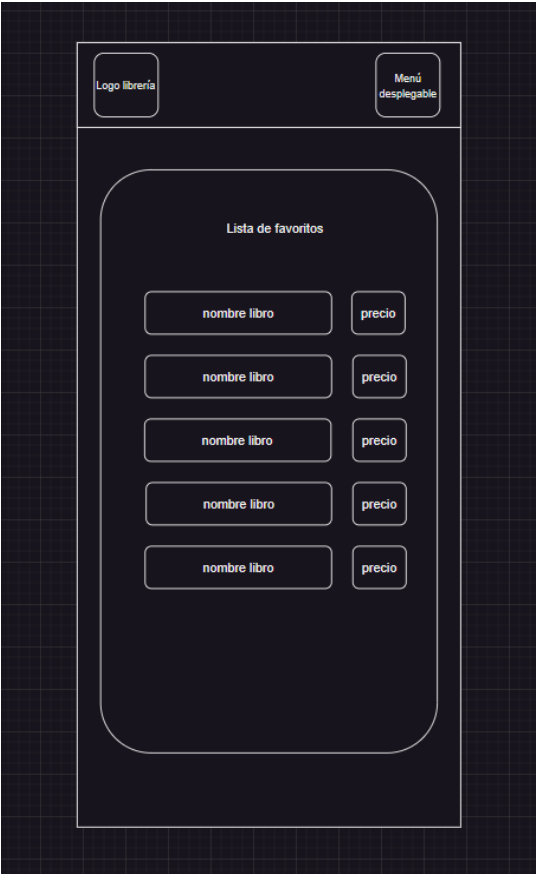
- Registro:

The registration form is a vertical rectangle with a dark blue background and a light blue grid. It features a header with a 'Logo librería' on the left and a 'Menú desplegable' on the right. The main content area is a rounded rectangle containing seven input fields: 'Nombre', 'Apellido', 'Nombre de usuario', 'Correo', 'Contraseña', 'Repita contraseña', and 'Fecha Nacimiento'. Below these are two buttons: 'limpiar' and 'confirmar registro'. The footer contains four social media icons, the text 'Redes sociales', and three buttons: 'Correo', 'Ubicación', and 'Teléfono'.

- Descripción libro:



- Favoritos:



- **Editar perfil:**

Logo librería

Menú desplegable

Nombre

Apellido

Nombre de usuario

contraseña

Fecha Nacimiento

limpiar

editar

Footer

Redes sociales

Correo

Ubicación

Teléfono

- **Carrito de compras:**

Logo librería

Menú desplegable

nombre libro

+

-

10

total

nombre libro

+

-

10

total

nombre libro

+

-

10

total

nombre libro

+

-

10

total

subtotal

\$2342

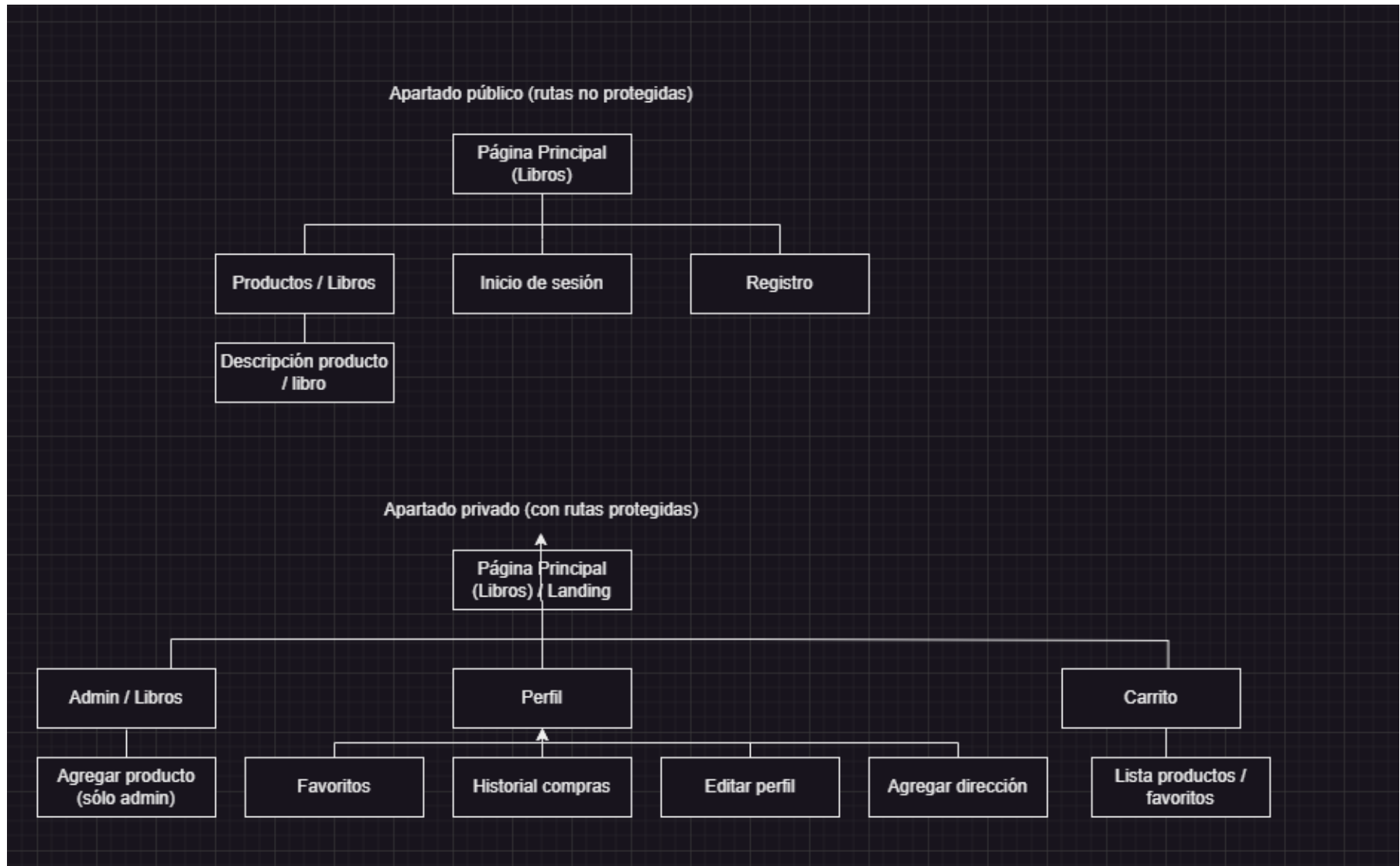
total

\$2342

libro

libro

2. Definición de la navegación



3. Listado de dependencias a utilizar en el proyecto

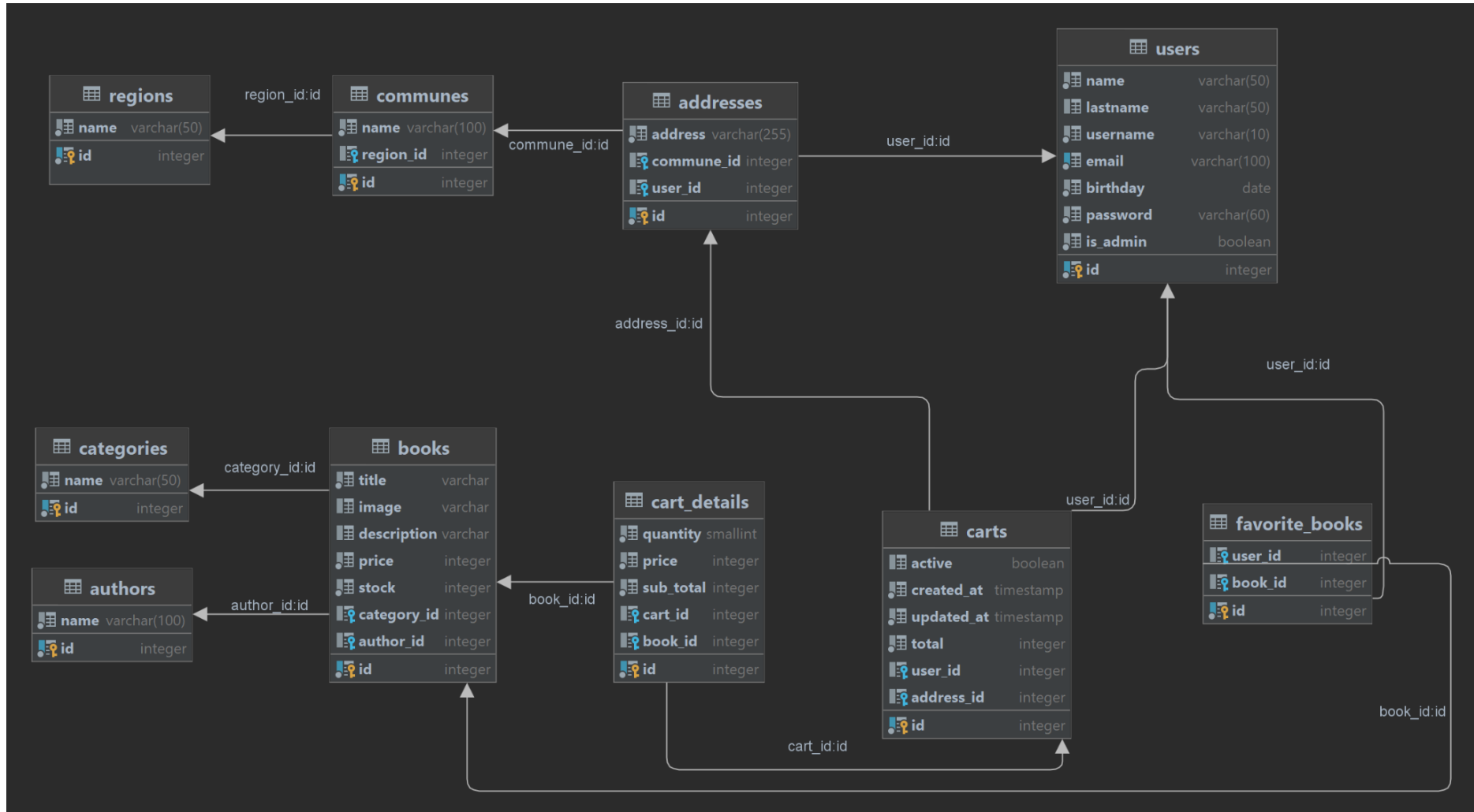
Dependencias Frontend

```
{
  "dependencies": {
    "bootstrap": "^5.2.3",
    "react": "^18.2.0",
    "formik": "^2.4.2",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.14.1",
    "react-icons": "^4.10.1",
    "react-multi-carousel": "^2.8.4",
    "react-toastify": "^9.1.3",
    "sass": "^1.63.6",
    "material-ui": "^5.13.7"
  },
  "devDependencies": {
    "nodemon": "^2.0.22"
  }
}
```

Dependencias Backend

```
{
  "dependencies": {
    "cors": "^2.8.5",
    "dotenv": "^16.1.3",
    "express": "^4.18.2",
    "pg": "^8.11.0",
    "pg-format": "^1.0.4",
    "jest": "^28.1.3",
    "supertest": "^6.2.4",
    "morgan": "^1.10.0"
  },
  "devDependencies": {
    "nodemon": "^2.0.22"
  }
}
```


4. Diseño de tablas de la base de datos y sus relaciones



5. Diseño de contrato de datos de API REST

```
"Registro de usuario"
POST /api/v1/register
request:
payload: {
  name: String;
  email: String;
  username: String;
  dateOfBirth: String;
  password: String
};
response: {
  ok: boolean;
  message: string
}
```

```
"Inicio de sesión de usuario"
POST /api/v1/login
request:
payload: {
  login: String;
  password: String
};
response: {
  ok: boolean;
  token: string
}
```

"Edición de usuario"

PUT /api/v1/user

request:

```
headers: {  
  Authorization: Bearer token;  
};
```

```
payload: {  
  name: String;  
  username: String;  
  password: String  
};
```

```
response: {  
  ok: boolean;  
  message: string  
}
```

"Agregar dirección de usuario"

GET/api/v1/region //tabla específica

request:

```
response: {  
  ok: boolean;  
  region: []  
}
```

GET/api/v1/commune //tabla específica

request:

```
response: {  
  ok: boolean;  
  commune: []  
}
```

POST /api/v1/user/address

request:

```
headers: {  
  Authorization: Bearer_token  
};
```

```
body: {  
  address: string;  
  idCommune: Number  
};
```

```
response: {  
  ok: boolean;  
  message: string  
}
```

```
"Obtener una galería de libros (productos), con filtrado, ordenación y paginación"
GET /api/v1/books
request:
queries: {
  sort ? : Number;
  limit ? : Number;
  page = 1;
  category: Number; //corresponde al id de la categoría/autor
  author: Number;
};
response: {
  ok: boolean;
  books: []
}
```

```
"Obtener el detalle de un libro (producto) específico:"
GET /api/v1/books/{bookId}
request:
params: {
  id: Number
};
response: {
  ok: boolean;
  book: {}
}
```

```
"Agregar un nuevo libro (producto)"
POST /api/v1/books
request:
headers: {
  Authorization: Bearer token;
};
payload: {
  title: String;
  image: String;
  description: String;
  price: number;
  stock: number;
  author: Number; //corresponde al id de la categoría/autor
  category: Number;
};
response: {
  ok: boolean;
  message: string
}
```

```
"Actualizar la información de un libro (producto)"
PUT /api/v1/books/{bookId}
request:
headers: {
  Authorization: Bearer token;
};
payload: {
  title: String;
  image: String;
  description: String;
  price: number;
  stock: number;
  author: Number; //corresponde al id de la categoría/autor
  category: Number;
};
response: {
  ok: boolean;
  message: string
}
```

```
"Agregar un libro a la lista de favoritos de un usuario"
POST /api/v1/user/favorites
request:
headers: {
  Authorization: Bearer_token
};
body: {
  book: Number; //id
};
response: {
  ok: boolean;
  message: string
}
```

```
"Obtener favoritos de un usuario"
GET /api/v1/user/favorites
request:
headers: {
  Authorization: Bearer_token
};
response: {
  books: []
}
```

```
"Carro actual, sin finalizar"
GET /api/v1/user/cart
request:
headers: {
  Authorization: Bearer_token
};
response: {
  ok: boolean;
  cart_detail: []
}
```

```
"Finalizar compra, simular proceso de pago"
POST /api/v1/user/purchase
request:
headers: {
  Authorization: Bearer_token
};
body: {
  idAddress: Number
};
response: {
  ok: boolean;
  message: string
}
```

```
"Agregar el detalle de la compra realizada"
POST /api/v1/user/cartdetail
request:
headers: {
  Authorization: Bearer_token
};
payload: {
  book_id: number;
  quantity: number
};
response: {
  ok: boolean;
  message: string
}
```

```
"Ver histórico de compras"
GET /api/v1/user/allcarts
request:
headers: {
  Authorization: Bearer_token
};
response: {
  ok: boolean;
  carts: []
}
```

```
"Obtener el detalle de una compra en específico"
GET /api/v1/user/cart/:id
request:
headers: {
  Authorization: Bearer_token
};
params: {
  id: Number
};
response: {
  ok: boolean;
  cart: {}
  cart_detail: []
}
```