

## Proyecto React

En el presente proyecto se desarrollará una aplicación frontend utilizando **ReactJS**, con el objetivo de implementar un sistema de seguridad que permita gestionar usuarios, roles, permisos y otras entidades necesarias para el control de acceso dentro de un entorno de microservicios. Este frontend consumirá los servicios REST proporcionados por un **backend ya desarrollado**, cuyo enlace de acceso se puede encontrar a continuación: [https://github.com/felipebuitragocarmona/ms\\_security](https://github.com/felipebuitragocarmona/ms_security) . La interfaz se construirá siguiendo principios de modularidad y reutilización, permitiendo una gestión clara y eficaz de las entidades involucradas cumpliendo como mínimo con los diseños plasmados en los mockups que anexan en el presente documento.

El sistema contempla la implementación de relaciones entre entidades que reflejan las necesidades del dominio de seguridad. Estas incluyen relaciones **uno a uno** como la que existe entre un usuario y su perfil personal, su dirección física y su firma digital. También se contemplan relaciones **uno a muchos**, como las que vinculan al usuario con sus sesiones activas, sus contraseñas históricas y sus dispositivos registrados. Adicionalmente, se modelarán relaciones **muchos a muchos**, como la que existe entre los usuarios y las preguntas de seguridad (mediante una tabla intermedia de respuestas), entre los usuarios y los roles (mediante una entidad intermedia llamada UserRole), y entre los roles y los permisos (mediante la entidad RolePermission). Todas estas relaciones serán representadas adecuadamente en la capa de presentación, permitiendo una gestión eficiente de cada aspecto del sistema de seguridad.

El proyecto debe realizarse en **grupos de 3 personas**. Cada integrante debe seleccionar un color y cumplir con los siguientes requisitos:

### 1. Autenticación mediante OAuth

Implementar autenticación utilizando OAuth (se puede utilizar Firebase) según el proveedor asignado:

- **Azul:** Microsoft
- **Amarillo:** Google
- **Verde:** Github

### 2. Implementación de CRUD

Cada integrante debe implementar las operaciones CRUD para las entidades definidas en el diagrama de clases. La distribución de trabajo debe garantizar que cada estudiante desarrolle:

- 1 relación **1:1** (uno a uno)
- 1 relación **1:N** (uno a muchos)
- 1 relación **N:N** (muchos a muchos)

### 3. Diseño de Interfaces

Se proporcionan diseños básicos en los mockups anexos, los cuales deben respetarse como **requisito mínimo**. Sin embargo, cada estudiante debe crear componentes adicionales utilizando librerías de diseño según su color asignado:

- **Azul:** Tailwind CSS
- **Amarillo:** Material UI
- **Verde:** Bootstrap

**Importante:** Los componentes creados deben ser **genéricos y reutilizables** para cualquier entidad del sistema. (Componentes de listar, crear y actualizar)

El sistema debe incluir un **selector de librería de diseño** en la parte superior del menú principal. Al cambiar la selección, toda la interfaz del proyecto debe actualizar automáticamente su visualización, aplicando los componentes de la librería elegida (Tailwind CSS, Material UI o Bootstrap) de forma dinámica en todas las vistas.

Para las vistas que no cuenten con mockup, los estudiantes tienen libertad creativa para diseñarlas.

### 4. Seguridad y Navegación

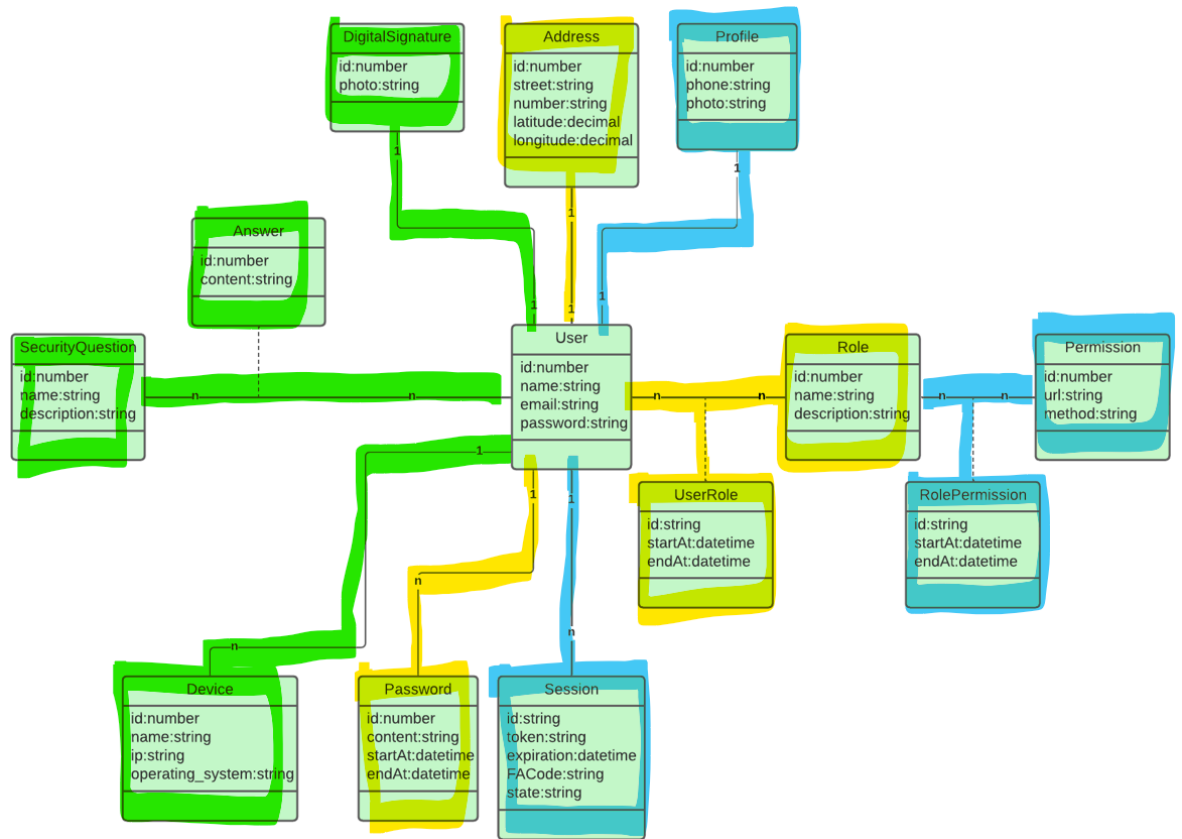
El sistema debe implementar:

- **Interceptores** (para manejo de peticiones HTTP)
- **Guards** (guardianes de rutas para control de acceso)

**Nota:** Los ejemplos de los endpoints del backend los pueden encontrar importando el **json** ([https://github.com/felipebuitragocarmona/ms\\_security/blob/main/postam\\_example\\_s.json](https://github.com/felipebuitragocarmona/ms_security/blob/main/postam_example_s.json)) en POSTMAN

### Fechas

| Grupo               | Grupal (Entregables) | Individual (Nueva Funcionalidad) |
|---------------------|----------------------|----------------------------------|
| Lunes y Miércoles   | 3 de Noviembre       | 5 de Noviembre                   |
| Miércoles y Viernes | 7 de Noviembre       |                                  |




Security

← → ↻

Users Roles Permissions

### FBC - Address

 Street:   
Number:


Create

Security

← → ↻

Users Roles Permissions

### FBC - Signature

 Name: FBC  
Email: fbc@email.com

Security

← → ↻

Users Roles Permissions

### FBC - Signature

Photo:

Update

Security

← → ↻

Users Roles Permissions

### Users


| ID | Name | Email         | Options   |
|----|------|---------------|---|
| 1  | FBC  | fbc@email.com | <a href="#">View</a> <a href="#">Update</a> <a href="#">Delete</a> <a href="#">Profile</a> <a href="#">Address</a> <a href="#">Digital Signature</a> <a href="#">Devices</a> <a href="#">Passwords</a> <a href="#">Sessions</a> |
| 2  | ABC  | abc@email.com | <a href="#">View</a> <a href="#">Update</a> <a href="#">Delete</a> <a href="#">Profile</a> <a href="#">Address</a> <a href="#">Digital Signature</a> <a href="#">Devices</a> <a href="#">Passwords</a> <a href="#">Sessions</a> |

Security

← → ↻

Users Roles Permissions

### FBC - Profile

 Name: FBC  
Email: fbc@email.com  
Phone: +57 300 111 22 33

Security

← → ↻

Users Roles Permissions

### FBC - Profile

Name: FBC  
Email: fbc@email.com  
Phone:

Update

Security

← → ↻

Users Roles Permissions

### Roles

| ID | Name          | Description          | Options  |
|----|---------------|----------------------|--|
| 1  | Administrator | System administrator | <a href="#">View</a> <a href="#">Update</a> <a href="#">Delete</a> <a href="#">Permissions</a> |
| 2  | Visitor       | Temporary visitor    | <a href="#">View</a> <a href="#">Update</a> <a href="#">Delete</a> <a href="#">Permissions</a> |

Security

← → ↻

Users Roles Permissions

### Administrator - Users

| ID | Name | Email         | Options                     |
|----|------|---------------|-----------------------------|
| 1  | FBC  | fbc@email.com | <a href="#">Remove Role</a> |

Add User for Administrator Role

Select user:   
ABC  
XYZ

Security

← → ↻

Users Roles Permissions

### Permissions

| ID | URI      | Method | Options  |
|----|----------|--------|--|
| 1  | /users   | GET    | <a href="#">View</a> <a href="#">Update</a> <a href="#">Delete</a> |
| 2  | /users/? | DELETE | <a href="#">View</a> <a href="#">Update</a> <a href="#">Delete</a> |

Security

← → ↻

Users Roles Permissions

### Administrator - Permissions

| Model       | View                                | Link                                | Create                              | Update                              | Delete                              |
|-------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Users       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| Roles       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Permissions | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Security

← → ↻

Users Roles Permissions

### Permission Create

URI:   
Method:   
POST  
PUT  
DELETE

Create