

Atividade Diagnóstica - Aprendizagem Ativa por Projetos

Projeto de Diagnóstico: "Controle de Domótica"

Alunos: Meir, Iago e Lina

Artefatos de Projeto

1. Lista de Casos de Uso:

Usuário

UC01 - Autenticar Usuário

O usuário deve ser capaz de se autenticar (login/logout) no sistema.

Cômodos

UC02 - Cadastrar Cômodo

O usuário deve ser capaz de adicionar um novo cômodo (ex: sala, quarto, cozinha).

UC03 - Editar Cômodo

O usuário deve ser capaz de alterar o nome ou descrição de um cômodo existente.

UC04 - Remover Cômodo

O usuário deve ser capaz de excluir um cômodo.

UC05 - Listar Cômodos

O usuário deve ser capaz de visualizar todos os cômodos cadastrados.

Dispositivos

UC06 - Cadastrar Dispositivo

O usuário deve ser capaz de adicionar um dispositivo a um cômodo.

UC07 - Editar Dispositivo

O usuário deve ser capaz de alterar as informações de um dispositivo (nome, tipo, localização).

UC08 - Remover Dispositivo

O usuário deve ser capaz de excluir um dispositivo.

UC09 - Listar Dispositivos

O usuário deve ser capaz de visualizar todos os dispositivos de um cômodo.

UC10 - Alterar Estado do Dispositivo

O usuário deve ser capaz de ligar e desligar um dispositivo a qualquer momento.

Cenas

UC11 - Criar Cena

O usuário deve ser capaz de criar uma cena definindo uma sequência de ações em dispositivos, com intervalos de tempo entre elas.

UC12 - Editar Cena

O usuário deve ser capaz de modificar uma cena existente (nome, ações e intervalos).

UC13 - Remover Cena

O usuário deve ser capaz de excluir uma cena cadastrada.

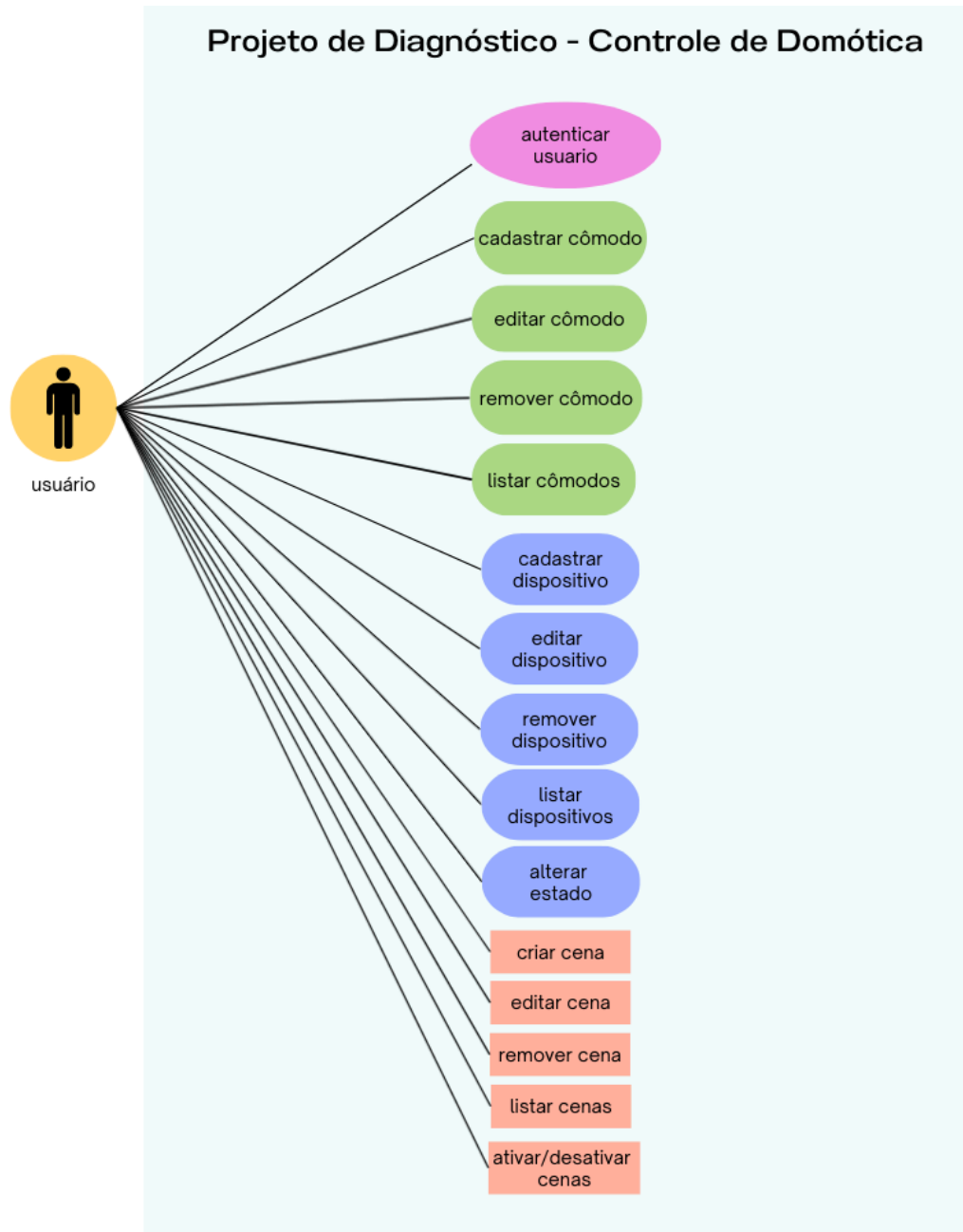
UC14 - Listar Cenas

O usuário deve conseguir visualizar todas as cenas disponíveis.

UC15 - Ativar/Desativar Cena

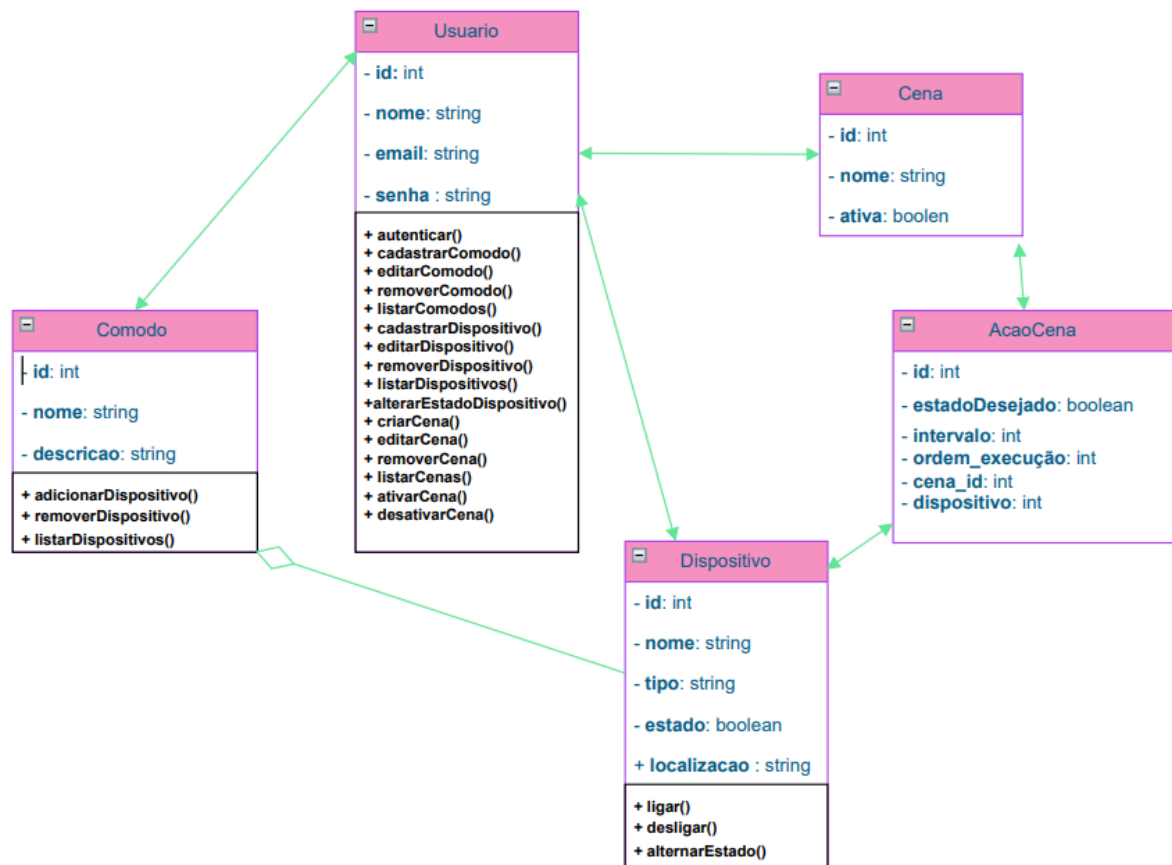
O usuário deve poder habilitar ou desabilitar uma cena cadastrada.

2. Diagrama de Caso de Uso:

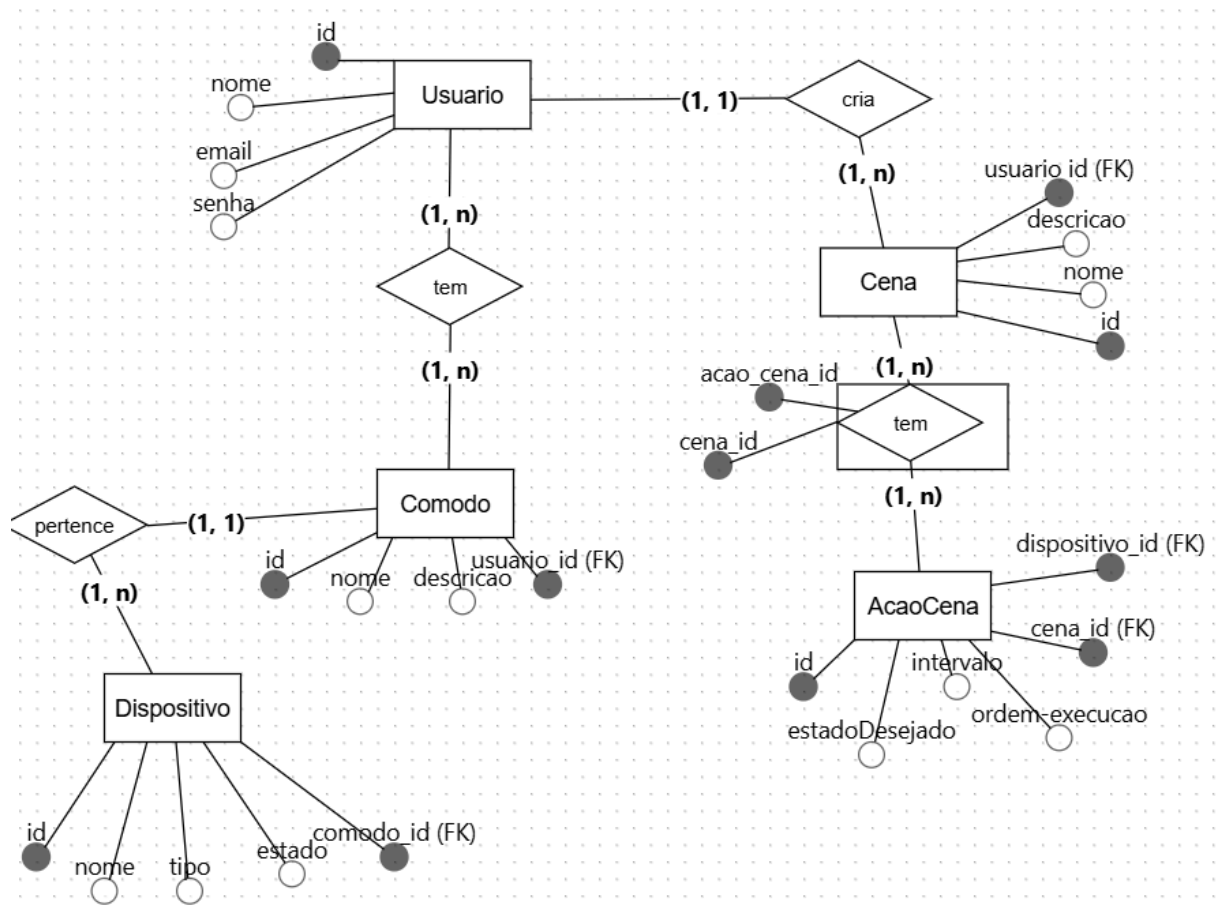


https://www.canva.com/design/DAGxL53laBg/-oR5tjn8-cAfu-XLMavmJA/edit?utm_content=DAGxL53laBg&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

3. **Diagrama de Classes:** Modelar as entidades do sistema (Dispositivo, Cômado, Cena) e seus relacionamentos, atributos e métodos.



4. **Diagrama de Entidade e Relacionamento (DER):** Modelar as entidades do banco de dados relacional (tabelas) e suas chaves primárias e estrangeiras, refletindo as relações entre elas.



5. Documentação de API (Endpoints):

Autenticação e Usuário

Method: POST

- **URL:** /login
- **Descrição:** realiza login do usuário e retorna token de autenticação
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
  "email": "string (obrigatório)",
  "senha": "string (obrigatório)"
}
```
- **Resposta (Response):**

```
{
  "message": "Login realizado com sucesso",
  "usuario": {
    "id": 1,
    "nome": "João",
    "email": "joao@email.com"
  },
  "token": "jwt_token"
}
```
- **Status Codes:**
 - 200 OK (Login realizado)
 - 401 Unauthorized (Senha incorreta)
 - 404 Not Found (Usuário não encontrado)
 - 500 (Internal Server Error)

Method: POST

- **URL:** /logout
- **Descrição:** finaliza sessão do usuário limpando o cookie
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
}
```
- **Resposta (Response):**

```
{
  "message": "Logout realizado"
}
```
- **Status Codes:**

Method: POST

- **URL:** /cadastro
 - **Descrição:** cria um novo usuário
 - **Parâmetros: (Query ou URL):** Nenhum
 - **Corpo da requisição (Body):**

```
{
  "nome": "string",
  "email": "string",
  "senha": "string",
}
```
 - **Resposta (Response):**

```
{
  "message": "Usuário cadastrado com sucesso"
  "usuario": { "id": 1, "nome": "João", "email": "joao@email.com" }
}
```
 - **Status Codes:**
 - 201 Created
 - 400 Bad Request
 - 409 Conflict
 - 500 Internal Server Error
-

Method: GET

- **URL:** /comodo
 - **Descrição:** Lista todos os cômodos do usuário autenticado
 - **Parâmetros: (Query ou URL):** Nenhum
 - **Corpo da requisição (Body):**

```
{
}
```
 - **Resposta (Response):**

```
[
  { "id": 1, "nome": "Sala", "descricao": "Sala principal", "usuario_id": 1 }
]
```
 - **Status Codes:**
 - 200 OK
 - 400 Bad Request
-

Method: GET

- **URL:** /comodo/:id
- **Descrição:** Busca um cômodo específico pelo ID
- **Parâmetros: (Query ou URL):** id (numero)
- **Corpo da requisição (Body):**

```
{
}
```
- **Resposta (Response):**

```
[
```

```
{ "id": 1, "nome": "Sala", "descricao": "Sala principal", "usuario_id": 1 }
]
```

- **Status Codes:**

- 200 OK

- 400 Bad Request

- 404 Cômodo não encontrado

Cômodos

Method: POST

- **URL:** /comodo

- **Descrição:** Cria um novo cômodo

- **Parâmetros: (Query ou URL):** id (numero)

- **Corpo da requisição (Body):**

```
{
  "nome": "string",
  "descricao": "string"
}
```

- **Resposta (Response):**

```
{
  "id": 2, "nome": "Quarto", "descricao": "Quarto casal", "usuario_id": 1
}
```

- **Status Codes:**

- 201 Sucess

- 400 Bad Request

Method: PUT

- **URL:** /comodo/:id/

- **Descrição:** Atualiza um cômodo do usuário

- **Parâmetros: (Query ou URL):** id (numero)

- **Corpo da requisição (Body):**

```
{
  "nome": "string",
  "descricao": "string"
}
```

- **Resposta (Response):**

```
{
}
```

- **Status Codes:**

- 200 Sucess

- 400 Bad Request

- 404 Cômodo não encontrado

Method: DELETE

- **URL:** /comodo/:id/

- **Descrição:** Remove um cômodo do usuário

- **Parâmetros: (Query ou URL):** id (numero)

- **Corpo da requisição (Body):**

```
{
}
```


- **Resposta (Response):**
 { "mensagem": "Cômodo removido com sucesso" }
 - **Status Codes:**
 - 200 Sucess
 - 400 Bad Request
 - 404 Cômodo não encontrado
-

Method: GET

- **URL:** /nome-do-comodo?comodo_id=1
- **Descrição:** Lista todos os dispositivos dos cômodos do usuário autenticado.
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**
 {
 }
 }
- **Resposta (Response):**
 {
 }
 }
- **Status Codes:**
 - 200 Sucess
 - 400 Bad Request
 - 404 Cômodo não encontrado

Dispositivos

Method: GET

- **URL:** /dispositivo
 - **Descrição:** Lista todos os dispositivos dos cômodos do usuário autenticado.
 - **Parâmetros: (Query ou URL):** Nenhum
 - **Corpo da requisição (Body):**
 {
 }
 }
 - **Resposta (Response):**
 {
 }
 }
 - **Status Codes:**
 - 200 Sucess
 - 400 Bad Request
-

Method: GET

- **URL:** /dispositivo-comodo?comodo_id=1
- **Descrição:** Lista dispositivos de um cômodo específico
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**
 {
 }
 }
- **Resposta (Response):**
 {
 }

- ```
}

```
- **Status Codes:**
    - 200 Sucess
    - 400 Bad Request
- 

**Method: GET**

- **URL:** /dispositivo/:id
  - **Descrição:** Busca um dispositivo pelo ID do usuário
  - **Parâmetros: (Query ou URL):** Nenhum
  - **Corpo da requisição (Body):**

```
{
}
```
  - **Resposta (Response):**

```
{
}
```
  - **Status Codes:**
    - 200 Sucess
    - 400 Bad Request
    - 404 Not Found
- 

**Method: POST**

- **URL:** /dispositivo
  - **Descrição:** Adiciona novo dispositivo em um cômodo do usuário
  - **Parâmetros: (Query ou URL):** Nenhum
  - **Corpo da requisição (Body):**

```
{
 "nome": "string",
 "tipo": "string",
 "estado": true,
 "comodo_id": 1
}
```
  - **Resposta (Response):**

```
{
}
```
  - **Status Codes:**
    - 201 Sucess
    - 400 Bad Request
    - 403 Acesso negado (não pode adicionar dispositivo a um cômodo que não é do usuário)
- 

**Method: PUT**

- **URL:** /dispositivo/:id
- **Descrição:** Atualiza informações de um dispositivo
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
 "nome": "string",
 "tipo": "string",
 "estado": false,

```

- ```
        "comodo_id": 1
      }
    }
  }
}

```
- **Resposta (Response):**

```
{
}
```
 - **Status Codes:**
 - 200 Sucess
 - 400 Bad Request
 - 404 Not Found
-

Method: DELETE

- **URL:** /dispositivo/:id
- **Descrição:** Remove dispositivo do usuário
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
}
```
- **Resposta (Response):**

```
{ "mensagem": "Dispositivo removido com sucesso" }
```
- **Status Codes:**
 - 200 Sucess
 - 400 Bad Request
 - 404 Not Found

Cenas

Method: GET

- **URL:** /cena
- **Descrição:** Lista todas as cenas do usuário com ações vinculadas.
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
}
```
- **Resposta (Response):**

```
[
  {
    "id": 1,
    "nome": "Cena Noite",
    "descricao": "Apaga luzes",
    "estado": false,
    "acoes": [
      { "nome": "Luz Sala", "estadoDispositivo": false }
    ]
  }
]
```
- **Status Codes:**
 - 200 Sucess
 - 400 Bad Request

Method: GET

- **URL:** /cena/:id
 - **Descrição:** Retorna uma cena específica pelo ID
 - **Parâmetros: (Query ou URL):** Nenhum
 - **Corpo da requisição (Body):**

```
{
}
```
 - **Resposta (Response):**

```
{
}
```
 - **Status Codes:**
 - 200 Sucess
 - 400 Bad Request
 - 404 Not Found
-

Method: POST

- **URL:** /cena
 - **Descrição:** Cria uma nova cena
 - **Parâmetros: (Query ou URL):** Nenhum
 - **Corpo da requisição (Body):**

```
{
  "nome": "string",
  "descricao": "string",
  "estado": true
}
```
 - **Resposta (Response):**

```
{
}
```
 - **Status Codes:**
 - 201 Sucess
 - 400 Bad Request (Nome obrigatório)
 - 400 Bad Request (Erro na requisição)
-

Method: PUT

- **URL:** /cena/:id
- **Descrição:** Atualiza dados da cena
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
  "nome": "string",
  "descricao": "string",
  "estado": false
}
```
- **Resposta (Response):**

```
{
}
```
- **Status Codes:**
 - 200 Sucesso
 - 400 Bad Request (Nenhum campo foi fornecido)
 - 400 Bad Request
 - 404 Not Found

Method: DELETE

- **URL:** /cena/:id
- **Descrição:** Remove cena do usuário
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
}
```
- **Resposta (Response):**

```
{
}
```
- **Status Codes:**
 - 200 Sucesso
 - 400 Bad Request
 - 404 Not Found

Method: PUT

- **URL:** /cena/:id/toggle
- **Descrição:** Alterna o estado da cena (ativo/inativo)
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
}
```
- **Resposta (Response):**

```
{ "id": 1, "nome": "Cena Noite", "estado": true }
```
- **Status Codes:**
 - 200 Sucesso
 - 400 Bad Request
 - 404 Not Found

Cena_Ação

Method: POST

- **URL:** /cena_acao
- **Descrição:** Adiciona uma ação a uma cena
- **Parâmetros: (Query ou URL):** Nenhum
- **Corpo da requisição (Body):**

```
{
  "cena_id": 1,
  "dispositivo_id": 3,
  "intervalo": 5000,
  "estadoDispositivo": true,
  "ordem_execucao": 1
}
```
- **Resposta (Response):**

```
{
  "id": 10,
  "cena_id": 1,
  "dispositivo_id": 3,
  "intervalo": 5000,
  "estadoDispositivo": true,
  "ordem_execucao": 1
}
```
- **Status Codes:**

201 Sucesso
400 Bad Request

MAPA DE HABILIDADES

NOME	NÍVEL	OBSERVAÇÃO
Lina	Frontend: Confortável Backend: Confortável -> banco de dados e Mais ou menos -> API	Acredito que aprender um framework/biblioteca para o frontend vai ser muito legal, e os conceitos do backend/API já estão fazendo muito sentido.
Meir	Frontend: Confortável Backend: Confortável	Fazendo o trabalho percebi que ainda tenho muito a aprender de backend, mas não tive tanta dificuldade, já com o frontend sinto a necessidade de aprender mais a fundo sobre bibliotecas e frameworks.
Iago	Frontend: Mais ou menos Backend e Banco de Dados : Confortável	Durante o trabalho o meu maior problema foi instalar o postgresql no ubuntu. Eu entendo muitos conceitos no frontend mas tenho certa dificuldade de implementar, acredito que por falta de prática. No backend eu só não sou fluente em alguma linguagem, mas entendo a arquitetura.

