

FECAP

**PROJETO.....**

**Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas**

São Paulo

2024

## **INTEGRANTES DO PROJETO e RA'S**

Jean da Costa Silva	– 22023948
Caroline Gomes	– 23024619

- **INTRODUÇÃO**

O Íntima Learn visa promover a conscientização e o conhecimento sobre educação sexual entre seu público-alvo. Sua característica principal é uma trilha de conhecimento que apresenta perguntas de múltipla escolha abrangendo diversos subtemas da educação sexual. Conforme o usuário avança na trilha, sua pontuação aumenta a cada resposta correta. Além disso, o Íntima Learn oferece recomendações de livros, podcasts e filmes relacionados ao tema como uma funcionalidade secundária.

- **Teste de Software**

- **Apresentar 2 testes unitários.**

Realizamos 3 testes unitários, dentre eles:

- Funcionamento do servidor;
- Obtenção do nome do usuário;
- Exclusão de usuário.

```

const assert = require("assert");
const request = require("supertest");
const app = require("../index"); // Substitua 'seu_arquivo_servidor' pelo nome do seu arquivo de servidor

describe("Testes da API", function () {
  // Teste para verificar se o servidor está funcionando
  it('Deve retornar "Servidor funcionando!"', function (done) {
    request(app).get("/").expect(200).expect("Servidor funcionando!", done);
  });

  // Teste para verificar se a rota de obtenção do nome do usuário funciona corretamente
  it("Deve obter o nome do usuário pelo email", function (done) {
    request(app)
      .post("/obterNomeUsuario")
      .send({ email: "leocatadral1000@gmail.com" })
      .expect(200)
      .expect("Leonardo", done);
  });

  // Teste para verificar se a rota de exclusão de usuário funciona corretamente
  it("Deve excluir um usuário", function (done) {
    request(app)
      .delete("/deletarUsuario")
      .send({ email: "teste12345@gmail.com", senha: "12345" })
      .expect(200, done);
  });
});

```

O retorno da API mostra que os testes funcionaram corretamente.

The screenshot shows a code editor with three tabs: 'index.js M', 'README.md', and 'tests.js U X'. The 'tests.js' tab is active, displaying the same code as the first block. Below the code editor, there is a terminal window with the following output:

```

> start
> Terminal X
> Terminal

```

The terminal output shows the results of the API tests:

```

Testes da API
✓ Deve retornar "Servidor funcionando!"
✓ Deve obter o nome do usuário pelo email
✓ Deve excluir um usuário

```

As próximas imagens demonstram os mesmos testes, porém, realizamos a alteração de alguns dados para verificar se o servidor apontaria o erro, e obtivemos sucesso.

```
const assert = require("assert");
const request = require("supertest");
const app = require("../index"); // Substitua 'seu_arquivo_servidor' pelo nome do seu arquivo de servidor

describe("Testes da API", function () {
  // Teste para verificar se o servidor está funcionando
  it('Deve retornar "Servidor funcionando!"', function (done) {
    request(app).get("/").expect(200).expect("Servidor funcionando!", done);
  });

  // Teste para verificar se a rota de obtenção do nome do usuário funciona corretamente
  it("Deve obter o nome do usuário pelo email", function (done) {
    request(app)
      .post("/obterNomeUsuario")
      .send({ email: "leocatedral1000@gmail.com" })
      .expect(200)
      .expect("Leonardo", done);
  });

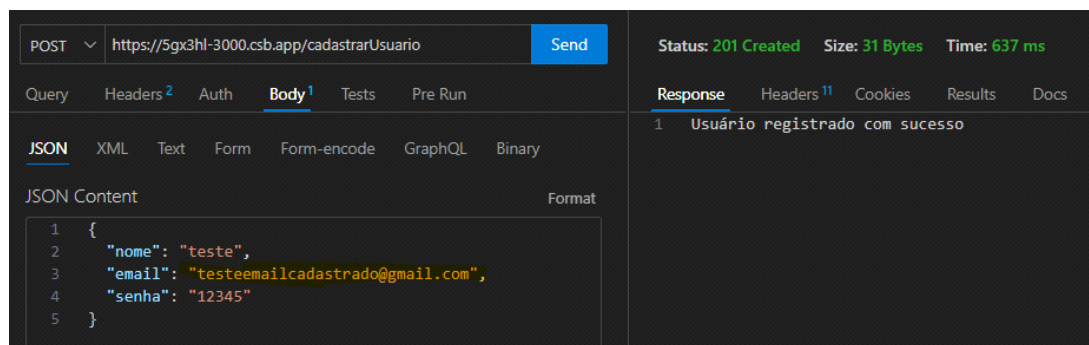
  // Teste para verificar se a rota de exclusão de usuário funciona corretamente
  it("Deve excluir um usuário", function (done) {
    request(app)
      .delete("/deletarUsuario")
      .send({ email: "teste12345@gmail.com", senha: "12345" })
      .expect(200, done);
  });
});
```

#### Testes da API

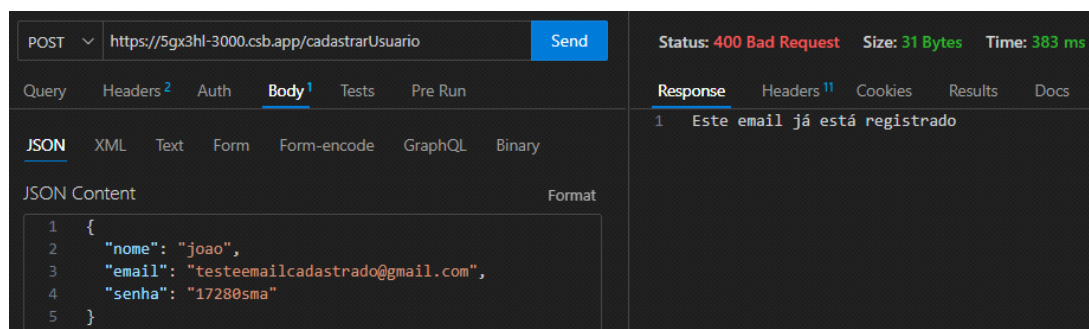
- ✓ Deve retornar "Servidor funcionando!"
- ✓ Deve obter o nome do usuário pelo email
- 2) POST /cadastrarUsuario cadastra um novo usuário
- 3) Deve excluir um usuário

- Apresentar 2 testes de componentes

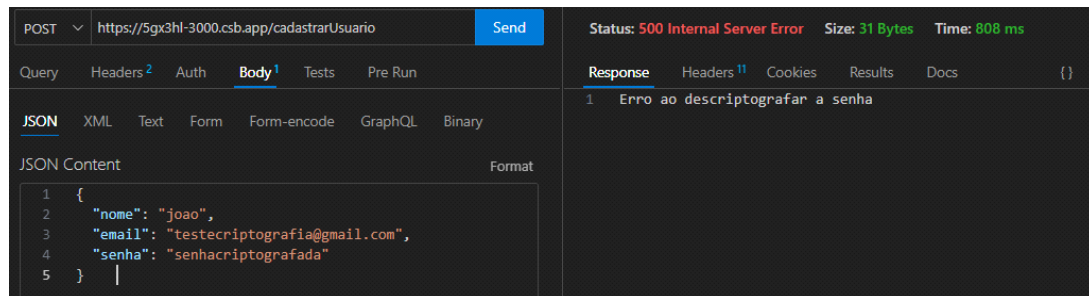
**Teste 01** – No primeiro teste o usuário utiliza determinado e-mail para realizar seu cadastro no app e o registro é completado com sucesso.



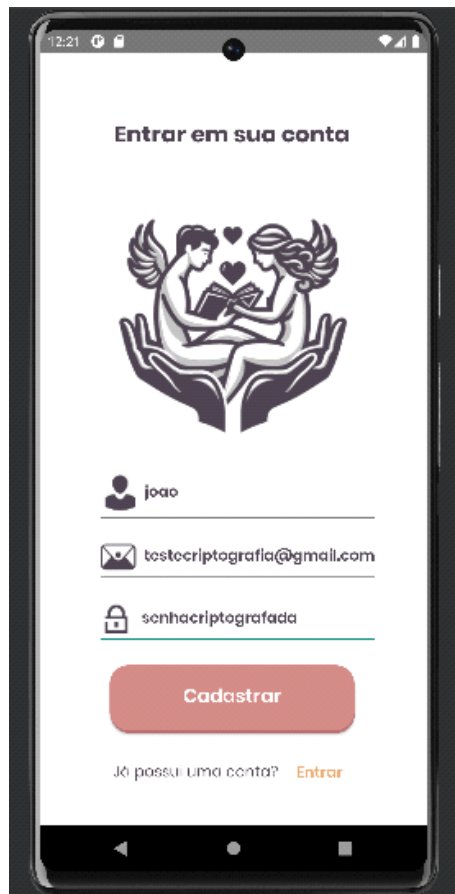
Em seguida, um outro usuário tenta realizar o cadastro utilizando o mesmo e-mail do usuário anterior que já consta em nossa base de dados, retornando para ele **[Este email já está registrado]**.



**Teste 02** - Neste teste de componente testamos a funcionalidade da criptografia. Na primeira imagem tentamos realizar um cadastro passando uma saída que não está pronta para ser recebida do lado do servidor, que espera uma senha criptografada.



Nesta imagem tentamos realizar o cadastro diretamente do lado do android, passando as mesmas informações usadas anteriormente, que possui a lógica de criptografia que o servidor espera.



Na imagem 3, podemos ver que a mesma senha utilizada no android foi criptografada, passando assim uma saída correspondente a esperada do lado do servidor.

```
i-20 21:23:30.692 3246-3246 CLIENTE com.example.intimalearn D Nome: joao
i-20 21:23:30.692 3246-3246 CLIENTE com.example.intimalearn D Email: testecriptografia@gmail.com
i-20 21:23:30.692 3246-3246 CLIENTE com.example.intimalearn D Senha: qwZlpuM8Co3oGPhE3YWMR86GUu7a17N/x83/0FMjESM=
```

Retorno no servidor - Nesta imagem podemos ver que o usuário foi devidamente cadastrado e a senha registrada foi automaticamente descriptografada.

```
,',
{
  "id": 10,
  "nome": "joao",
  "email": "testecriptografia@gmail.com",
  "senha": "senhacriptografada",
  "pontuacao": 0,
  "premiacoes": "",
  "resetToken": null,
  "resetTokenExpires": null
}
```

**OBS:** Isso previne que pessoas mau intencionadas tentem realizar o cadastro sem ser pelo app, além de prevenir o vazamento de dados.

- **Apresentar um teste de sistema.**

O teste de sistema foi feito em vídeo e está anexado junto a documentação com o nome - [ **Testequaldevops\_Atividade\_Teste\_Sistema** ].

- **Qualidade de Software**

- **Indicar 4 atributos de qualidade de software e informar como foi aplicado no projeto integrador (PI)**

**Confidencialidade:** As senhas são criptografadas para garantir a confidencialidade dos dados.

**Experiência do usuário:** Nosso layout garante que a interação do usuário com o aplicativo seja fluida e intuitiva.

**Interatividade:** A funcionalidade principal do aplicativo (trilha de conhecimento) promove a participação ativa do usuário e incentiva a interação contínua.

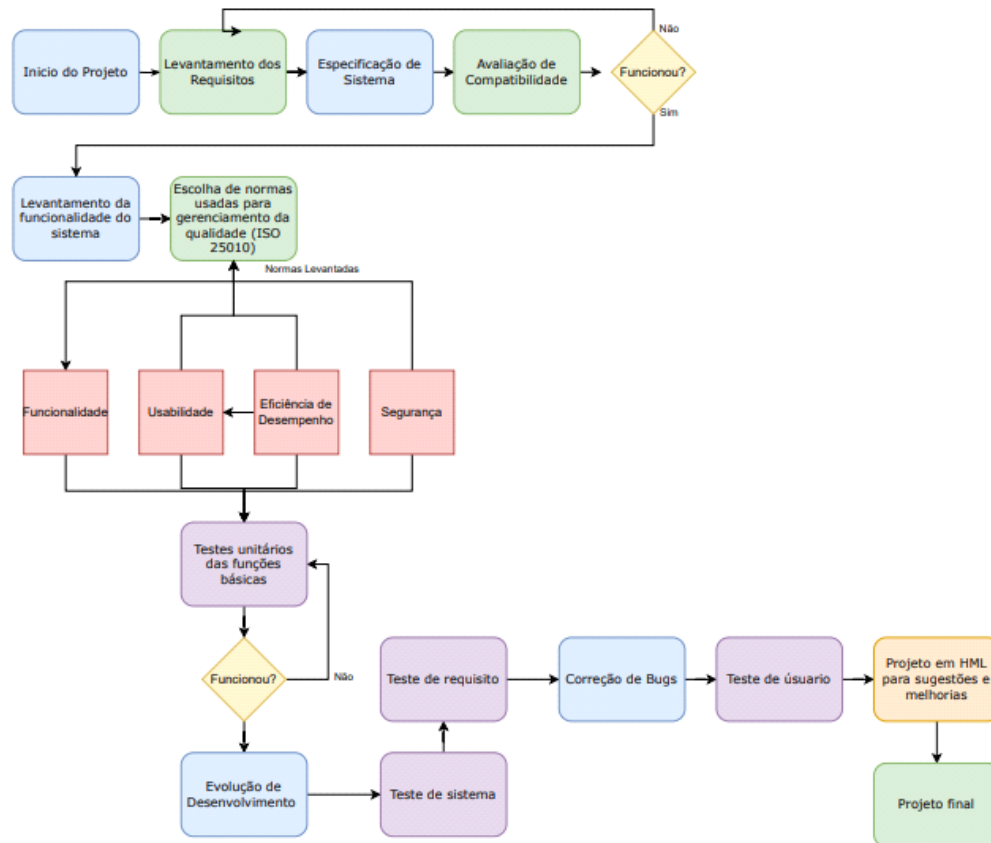
**Funcionalidade Apropriada:** O aplicativo é bem direto em relação ao tema proposta sem nenhuma funcionalidade que fuja da proposta inicial.

- Apresentar um Modelo de qualidade de software

	QiUWeP	
Segurança	Confidencialidade	Garante a proteção de informações sensíveis, evitando acesso não autorizado.
	Integridade	Assegura que os dados permaneçam completos e precisos, sem manipulação não autorizada.
	Autenticação	Verifica a identidade dos usuários, controlando o acesso aos sistemas e dados.
	Responsabilidade	Define claramente quem é responsável por ações e decisões dentro do projeto, promovendo a transparência.
Funcionalidade	Funcionalidade de complitude	Oferece um conjunto diversificado de recursos para atender às necessidades variadas dos usuários.
	Funcionalidade apropriada	Prioriza recursos que são relevantes e úteis para os usuários, evitando excesso de funcionalidades desnecessárias.
	Funcionalidade de correção	Garante que o sistema opere corretamente, minimizando erros e falhas.
Usabilidade	Interatividade	Promove a participação ativa dos usuários, melhorando a experiência e engajamento.
	Experiência do Usuário	Foca em proporcionar uma interação agradável e eficiente para os usuários, aumentando a satisfação e fidelidad
	Intuitividade	Facilita o uso do sistema, tornando-o fácil de entender e operar mesmo para usuários inexperientes.
Performance	Capacidade	Garante que o sistema seja capaz de lidar com uma carga de trabalho esperada, mantendo desempenho adequado.
	Tempo de resposta	Minimiza o tempo necessário para o sistema responder às solicitações dos usuários, melhorando a eficiência e a usabilidade.



- Apresentar um Processo (plano) de gerenciamento de qualidade de software



#### 4. REFERÊNCIAS BIBLIOGRÁFICAS

SOMMERVILLE, I. **Engenharia de Software**. 11ª Edição. São Paulo: Pearson Addison-Wesley, 2017.