

# **PROJETO EducaFácil**

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

**São Paulo**

**2024**

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

Giovanne Braga	RA: 23025648
Icaro Silva	RA: 23025413
Giovane Estevan	RA: 23025240

### **Sumário**

<b>1. INTRODUÇÃO.....</b>	<b>2</b>
<b>2. DOCUMENTO DE ABERTURA DO PROJETO.....</b>	<b>2</b>
<b>3. REQUISITOS DE SISTEMA.....</b>	<b>8</b>
3.1.    REQUISITOS FUNCIONAIS DE SOFTWARE.....	8
3.2.    REQUISITOS NÃO-FUNCIONAIS DE SOFTWARE.....	10
<b>4. CASOS DE USO.....</b>	<b>12</b>
<b>5. AQUITETURA DE SISTEMA.....</b>	<b>13</b>
<b>6. REFERÊNCIAS BIBLIOGRAFICAS.....</b>	<b>14</b>

## **1. INTRODUÇÃO**

Tendo em vista a crescente necessidade de recursos educacionais acessíveis e de qualidade, o projeto EducaFácil visa criar uma plataforma que facilite o acesso de alunos a materiais escolares e videoaulas, além de permitir que ONGs contribuam com conteúdos educacionais. Este documento detalha os requisitos, a arquitetura e os casos de uso do sistema, alinhados aos objetivos estratégicos da organização.

## **2. DOCUMENTO DE ABERTURA DO PROJETO**

### **Prefácio**

Este documento é destinado aos desenvolvedores, gerentes de projeto e partes interessadas no desenvolvimento do sistema EducaFácil. A versão atual inclui a definição inicial dos requisitos e arquitetura do sistema, justificando a necessidade de um recurso centralizado para compartilhamento de materiais educativos. Revisões subsequentes detalharão os requisitos e melhorias conforme o projeto avança.

### **Introdução**

O EducaFácil é necessário para suprir a lacuna de recursos educacionais disponíveis para alunos de diferentes níveis de ensino. O sistema permitirá que ONGs façam upload de materiais educativos e videoaulas, categorizando-os por tema e nível de ensino, enquanto alunos poderão acessar, buscar e baixar esses materiais conforme suas necessidades. Este sistema contribuirá para os objetivos estratégicos de ampliar o acesso à educação de qualidade e apoiar ONGs em suas missões educativas.

## Glossário

- **Usuário:** Aluno ou indivíduo que utiliza a plataforma para acessar materiais educativos.
- **ONG:** Organização não-governamental que contribui com materiais educativos para a plataforma.
- **Material Escolar:** Documentos, PDFs, videoaulas e outros recursos educativos disponibilizados na plataforma.
- **Upload:** Processo de enviar arquivos para o sistema.
- **Download:** Processo de baixar arquivos do sistema.

## Definição de Requisitos de Usuário

O sistema EducaFácil fornece serviços como cadastro de usuários e ONGs, login e autenticação, upload e categorização de materiais, busca e navegação de conteúdos, além de download e acesso aos materiais. Os requisitos não funcionais incluem segurança, desempenho, escalabilidade, usabilidade, confiabilidade e compatibilidade.

## Arquitetura do Sistema

A arquitetura do sistema EducaFácil será composta por três camadas principais:

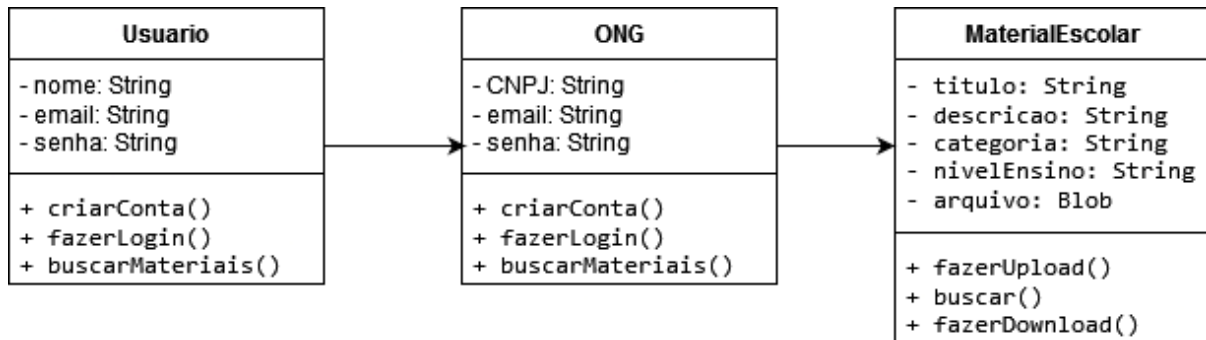
1. Camada de Apresentação (Frontend) utilizando CSS, JavaScript e React para interface de usuário.
2. Camada de Aplicação (Backend) com Node.js e Express.js para lógica de negócios e autenticação.
3. Camada de Dados (Banco de Dados) utilizando SQL para armazenamento de dados.

## Especificação de Requisitos do Sistema

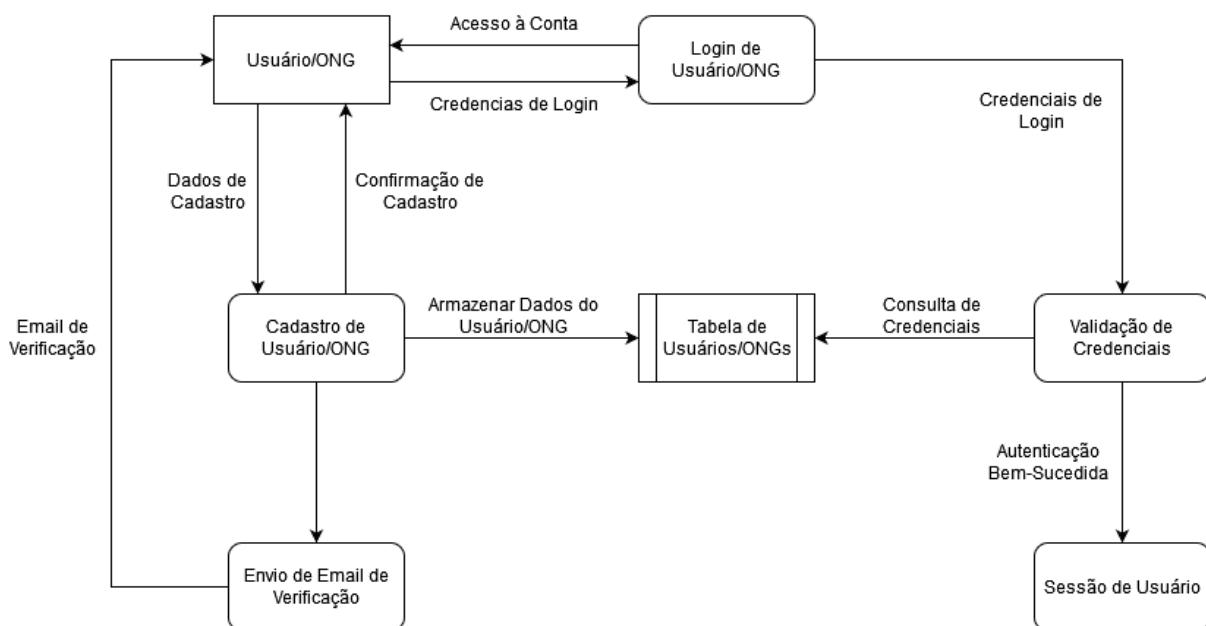
Os requisitos funcionais e não funcionais são detalhados nas seções subsequentes.

## Modelos do Sistema

### Diagrama de Classe



### Diagrama de Fluxo de Dados (DFD)



### Evolução do Sistema

O sistema EducaFácil é fundamentado em vários pressupostos que garantem sua funcionalidade inicial e sua capacidade de evolução conforme mudanças tecnológicas e necessidades dos usuários. A seguir, são detalhados esses pressupostos e as possíveis mudanças previstas para o sistema.

## Pressupostos Fundamentais

### 1. Infraestrutura Tecnológica:

- **Backend:** Utiliza Node.js e Express.js para garantir uma lógica de negócios robusta e escalável.
- **Frontend:** Implementado com CSS, JavaScript e React para uma interface de usuário interativa e responsiva.
- **Banco de Dados:** Baseado em SQL, permitindo consultas eficientes e integrações futuras com outros sistemas de armazenamento.

### 2. Segurança:

- Implementação de criptografia de senhas e dados sensíveis.
- Utilização de autenticação baseada em tokens para sessões de usuário.

### 3. Escalabilidade e Desempenho:

- Arquitetura projetada para suportar um grande número de acessos simultâneos.
- Uso de técnicas de otimização de desempenho, como caching e balanceamento de carga.

### 4. Usabilidade e Compatibilidade:

- Interface intuitiva e design responsivo para garantir uma experiência de usuário satisfatória.
- Compatibilidade com os principais navegadores e dispositivos.

## Mudanças Previstas

### 1. Evolução de Hardware:

- **Atualizações de Servidores:** A infraestrutura de servidores será atualizada periodicamente para garantir que o sistema possa lidar com um número crescente de usuários e materiais educativos.

### 2. Mudanças nas Necessidades dos Usuários:

- **Funcionalidades Adicionais:** Com base no feedback dos usuários, novas funcionalidades podem ser adicionadas, como fóruns de discussão, quizzes interativos e funcionalidades de aprendizado adaptativo.
- **Interface Personalizável:** Desenvolver funcionalidades que permitam aos usuários personalizar a interface de acordo com suas preferências.

## Apêndices

### Requisitos de Hardware

Para garantir o desempenho e a funcionalidade adequados do sistema EducaFácil, são recomendadas as seguintes especificações mínimas de hardware:

- **Servidor:**
  - Processador multicore de pelo menos 2,5 GHz.
  - Memória RAM de 8 GB ou superior.
  - Armazenamento SSD de pelo menos 100 GB para o sistema operacional e aplicação.
  - Conexão de rede estável com largura de banda adequada para suportar o tráfego de dados esperado.

- **Clientes (computadores dos usuários):**

- Navegador web moderno e atualizado (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, etc.).
- Conexão de internet estável para acessar os recursos online do EducaFácil.

**Requisitos de Banco de Dados:**

O banco de dados do sistema EducaFácil será baseado em tecnologia SQL para garantir a integridade e eficiência dos dados. As especificações incluem:

- **Sistema de Gerenciamento de Banco de Dados (SGBD):**

- Recomenda-se o uso de sistemas como MySQL, PostgreSQL ou SQL Server.

- **Estrutura de Banco de Dados:**

- Tabelas para armazenamento de dados de usuários, ONGs, metadados, etc.
- Relacionamentos definidos entre as tabelas para garantir integridade referencial e eficiência nas consultas.

- **Capacidade e Escalabilidade:**

- O banco de dados deve ser dimensionado para lidar com um grande volume de dados, considerando o crescimento esperado da plataforma ao longo do tempo.
- Estratégias de otimização de consulta e indexação devem ser implementadas para garantir o desempenho mesmo com um grande número de acessos simultâneos.

### **3. REQUISITOS DE SISTEMA**

### 3.1 REQUISITOS FUNCIONAIS DE SOFTWARE

<b>RFS01</b>	
<b>Função</b>	Cadastro de Usuários
<b>Descrição</b>	Permitir que usuários (alunos) criem contas no sistema.
<b>Entradas</b>	Nome, e-mail, senha, etc.
<b>Fonte</b>	Formulário de cadastro.
<b>Saídas</b>	Confirmação de cadastro.
<b>Ação</b>	Armazenar dados do usuário e enviar e-mail de verificação.

<b>RFS02</b>	
<b>Função</b>	Cadastro de ONGs
<b>Descrição</b>	Permitir que ONGs criem contas no sistema.
<b>Entradas</b>	E-mail, CNPJ, senha, etc.
<b>Fonte</b>	Formulário de cadastro.
<b>Saídas</b>	Confirmação de cadastro.
<b>Ação</b>	Armazenar dados da ONG e enviar e-mail de verificação.

<b>RFS03</b>	
<b>Função</b>	Login e Autenticação
<b>Descrição</b>	Permitir que usuários e ONGs façam login no sistema.
<b>Entradas</b>	E-mail e senha.
<b>Fonte</b>	Formulário de login.



<b>Saídas</b>	Acesso à conta.
<b>Ação</b>	Validar credenciais e permitir acesso ao sistema.

<b>RFS04</b>	
<b>Função</b>	Upload de Materiais
<b>Descrição</b>	Permitir que ONGs façam upload de materiais escolares e videoaulas.
<b>Entradas</b>	E-mail e senha.
<b>Fonte</b>	Formulário de upload.
<b>Saídas</b>	Material disponível no sistema.
<b>Ação</b>	Armazenar arquivos e metadados.

<b>RFS05</b>	
<b>Função</b>	Busca de Materiais
<b>Descrição</b>	Permitir que usuários busquem e naveguem pelos materiais disponíveis.
<b>Entradas</b>	Termos de busca, filtros.
<b>Fonte</b>	Barra de busca, filtros.
<b>Saídas</b>	Lista de materiais.
<b>Ação</b>	Recuperar e exibir materiais relevantes.

<b>RFS06</b>	
<b>Função</b>	Acesso e Download de Materiais
<b>Descrição</b>	Permitir que usuários façam download de materiais e assistam videoaulas.
<b>Entradas</b>	Seleção de material.
<b>Fonte</b>	Lista de materiais.
<b>Saídas</b>	Download do material ou vídeo.

<b>Ação</b>	Fornecer link para download ou streaming.
-------------	---

### 3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

<b>RFS01</b>	
<b>Função</b>	Segurança
<b>Descrição</b>	Proteger dados pessoais dos usuários e ONGs.
<b>Entradas</b>	Dados sensíveis.
<b>Fonte</b>	Cadastro e login.
<b>Saídas</b>	Dados criptografados.
<b>Ação</b>	Implementar criptografia de senhas e dados sensíveis.

<b>RFS02</b>	
<b>Função</b>	Desempenho
<b>Descrição</b>	Suportar grande número de acessos simultâneos.
<b>Entradas</b>	Acessos de usuários.
<b>Fonte</b>	Sistema.
<b>Saídas</b>	Resposta eficiente.
<b>Ação</b>	Otimizar performance e capacidade do servidor.

<b>RFS03</b>	
<b>Função</b>	Escalabilidade
<b>Descrição</b>	Suportar crescimento no número de usuários e materiais.
<b>Entradas</b>	Novos usuários e conteúdos.
<b>Fonte</b>	Sistema.

<b>Saídas</b>	Manutenção de desempenho.
<b>Ação</b>	Implementar arquitetura escalável.

<b>RFS04</b>	
<b>Função</b>	Usabilidade
<b>Descrição</b>	Interface intuitiva e fácil de usar.
<b>Entradas</b>	Interações de usuário.
<b>Fonte</b>	Frontend.
<b>Saídas</b>	Experiência de usuário satisfatória.
<b>Ação</b>	Desenvolver design responsivo e intuitivo.

<b>RFS05</b>	
<b>Função</b>	Confiabilidade
<b>Descrição</b>	Garantir alta disponibilidade do sistema.
<b>Entradas</b>	Operações do sistema.
<b>Fonte</b>	Infraestrutura.
<b>Saídas</b>	Sistema disponível.
<b>Ação</b>	Minimizar downtime e implementar recuperação rápida.

<b>RFS06</b>	
<b>Função</b>	Compatibilidade
<b>Descrição</b>	Compatível com principais navegadores e dispositivos.
<b>Entradas</b>	Acessos de diferentes plataformas.
<b>Fonte</b>	Sistema.
<b>Saídas</b>	Acesso universal.

<b>Ação</b>	Testar compatibilidade e adaptar conforme necessário.
-------------	---

#### 4. CASOS DE USO

- **Caso de Uso 1: Cadastro de Usuário**

- **Ator Principal:** Aluno
- **Descrição:** Um aluno deseja se cadastrar no sistema para acessar materiais escolares e videoaulas.
- **Fluxo Principal:**
  1. O aluno acessa a página de cadastro.
  2. O aluno preenche os dados necessários (nome, e-mail, senha, etc.).
  3. O aluno confirma o cadastro.
  4. O sistema envia um e-mail de verificação.
  5. O aluno verifica o e-mail e confirma a conta.
- **Pós-condição:** O aluno está cadastrado e pode fazer login no sistema.

- **Caso de Uso 2: Upload de Materiais pela ONG**

- **Ator Principal:** ONG
- **Descrição:** Uma ONG deseja fazer upload de materiais escolares e videoaulas no sistema.

- **Fluxo Principal:**

1. A ONG faz login no sistema.
2. A ONG acessa a seção de upload de materiais.
3. A ONG preenche os detalhes do material (título, descrição, categoria, nível de ensino).
4. A ONG seleciona os arquivos para upload.
5. A ONG confirma o upload.

- **Pós-condição:** O material está disponível no sistema para os alunos.

- **Caso de Uso 3: Busca e Acesso a Materiais**

- **Ator Principal:** Aluno

- **Descrição:** Um aluno deseja buscar e acessar materiais escolares e videoaulas.

- **Fluxo Principal:**

1. O aluno faz login no sistema.
2. O aluno acessa a seção de busca de materiais.
3. O aluno utiliza filtros para refinar a busca (categoria, tema, nível de ensino).
4. O aluno seleciona um material da lista de resultados.
5. O aluno faz o download do material ou assiste à videoaula online.

- **Pós-condição:** O aluno acessa o material desejado.

## 5. ARQUITETURA DO SISTEMA

O sistema EducaFácil é projetado com uma arquitetura em camadas, visando modularidade, escalabilidade e manutenibilidade. As principais camadas incluem:

1. **Camada de Apresentação (Frontend):**

- Responsável pela interface do usuário.
- Desenvolvida utilizando tecnologias como CSS, JavaScript e React para garantir uma experiência interativa e responsiva.
- Oferece recursos para os usuários navegarem pelos materiais educativos, realizar buscas e interagir com o sistema de forma intuitiva.

## **2. Camada de Aplicação (Backend):**

- Lida com a lógica de negócios e processamento de dados.
- Implementada com Node.js e Express.js para fornecer uma infraestrutura robusta e escalável.
- Gerencia funcionalidades como autenticação de usuários, upload e categorização de materiais, e busca de conteúdo.

## **3. Camada de Dados (Banco de Dados):**

- Armazena e gerencia os dados do sistema.
- Utiliza tecnologia SQL para garantir a integridade e eficiência dos dados.
- Contém tabelas para armazenamento de informações de usuários, ONGs, materiais educativos e metadados relacionados.

## **6. REFERÊNCIAS BIBLIOGRÁFICAS**

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