



Universidade Estadual do Norte Fluminense Darcy Ribeiro
Centro de Ciência e Tecnologia

Projeto POO Dev

Sistema para o Pescarte

Arthur Cézar Martins Ferreira Mól

Matrícula: 20221100108

Professor: João Luiz de Almeida Filho

28 de outubro de 2025

Sumário

Sumário		1
1	GitHub	2
2	Descrição do sistema	2
2.1	O que é o projeto PEA-Pescarte	2
3	Requisitos funcionais e não funcionais	3
3.1	Requisitos funcionais	3
3.2	Requisitos não-funcionais	3
4	Diagrama de classes inicial	4
4.1	Diagrama de classes (UML)	4
4.2	Diagrama entidade relacionamento (ER)	4
5	Fluxo básico de funcionamento	5
5.1	Diagrama de casos de uso	5
6	Orçamento	5
6.1	Custo do desenvolvimento	5
6.2	Custo de manutenção	5
6.3	Custo total	6
7	Cronograma detalhado por semana	6
8	Realização do projeto	6
8.1	Primeira semana	6
8.2	Segunda semana	8
8.3	Terceira semana	8
8.4	Quarta semana	9
8.5	Quinta semana	9
8.6	Sexta semana	10
8.7	Sétima semana	10

1 GitHub

[Clique aqui](#) para ser redirecionado ao repositório onde o projeto está.

2 Descrição do sistema

Linguagem para o back-end (e orientação a objetos) **Java** utilizando o framework **Springboot**.

Para o front-end será usado **React**.

Para o banco de dados será usado **Postgree**.

O sistema que será criado é uma plataforma interativa para o projeto PEA-Pescarte. O intuito desse sistema é criar uma plataforma onde os pescadores artesanais que participam do projeto possam logar, interagir com a plataforma e utilizar os serviços que serão fornecidos pela plataforma, como o de controle de gastos e lucro, além de informações gerais.

O sistema visa buscar fornecer serviços aos pescadores artesanais de modo a facilitar a vida deles, além de torná-los mais conectados.

As funcionalidades que estarão presentes no sistema são:

- Página **Home**: contendo as informações principais e mais relevantes do momento no site;
- Página **Login**: contendo a tela de login e criação de conta;
- Página **Serviços**: contendo serviços que podem vir a ser relevantes para os pescadores, como indicações de lojas como fábricas de gelo, lojas de artigos de pesca, informações sobre transporte, serviço de mecânico para os barcos;
- Página **Clima**: contendo informações sobre a previsão do tempo para o dia, se existe possibilidade de chuva, velocidade do vento, dentre outros;
- Página **Financeiro**: que faz um "controle" do financeiro dos pescadores, nessa página eles vão poder inserir o que pescaram e a aplicação vai fazer uma estimativa de por quanto eles deveriam vender os peixes baseado em informações estaduais e federais. Além disso, também uma opção de gerar uma planilha financeira com os gastos e lucros dos mesmos;
- Página **Sobre**: essa página vai conter as informações sobre o que é o projeto PEA-Pescarte.

2.1 O que é o projeto PEA-Pescarte

O Projeto PESCARTE tem como sua principal finalidade a criação de uma rede social regional integrada por pescadores artesanais e por seus familiares, buscando, por meio de processos educativos, promover, fortalecer e aperfeiçoar a sua organização comunitária e a sua qualificação profissional, bem como o seu envolvimento na construção participativa e na implementação de projetos de geração de trabalho e renda.

O Projeto de Educação Ambiental Pescarte (PEA Pescarte) trabalha junto às **comunidades de pesca artesanal** de Arraial do Cabo, Búzios, Cabo Frio, Campos, Carapebus, Macaé, Rio das Ostras, Quissamã, São Francisco de Itabapoana e São João da Barra. Desde 2014, atua juntamente aos pescadores artesanais e seus familiares, por meio de processos educativos, promovendo, fortalecendo e aperfeiçoando a organização comunitária e a sua qualificação profissional, bem como o seu envolvimento na construção participativa e na implementação de projetos de geração de trabalho e renda.

O processo educativo é realizado nos 10 municípios e se dá através de oficinas com temas diversos, entre eles: economia solidária, cooperativismo, políticas públicas, licenciamento ambiental, letramento digital e gestão participativa. O projeto promove, também, articulações entre os pescadores e pescadoras, com reuniões do Grupo de Trabalho, Grupo Gestor, Grupo de Acompanhamento de Obras e assembleias municipais.

3 Requisitos funcionais e não funcionais

3.1 Requisitos funcionais

Definem a **funcionalidade** que o sistema a ser desenvolvido deverá ter. Ele é um requisito relacionado com um tipo de comportamento produto de uma função do sistema.

Os requisitos funcionais deste projeto são:

- Cadastrar usuários;
- Realizar login;
- Transmitir dados;
- Exibir informações do usuário;
- Mostrar serviços;
- Gerar documentos;
- Consultar documentos.

Os requisitos funcionais se relacionam diretamente a processos¹ que o sistema deve executar. Ex: pesquisar, cadastrar, relatar, verificar, imprimir.

3.2 Requisitos não-funcionais

Indicam propriedades comportamentais que o sistema deve possuir.

Os requisitos não-funcionais deste sistema são:

- Poder acessar a aplicação usando qualquer navegador;
- Poder acessar a aplicação usando qualquer dispositivo: celular, tablet ou computador (responsividade);
- Aplicação com informações expostas de modo que pessoas com dificuldade de usar elementos da tecnologia possa acessá-lo sem problemas (acessibilidade com botões e textos grandes e intuitivos);
- Boa formatação, legibilidade e confiabilidade dos documentos gerados pelo sistema;
- Usuários poderem fazer login somente atrelados a uma Corporativa.

¹ processos = verbos

4 Diagrama de classes inicial

4.1 Diagrama de classes (UML)

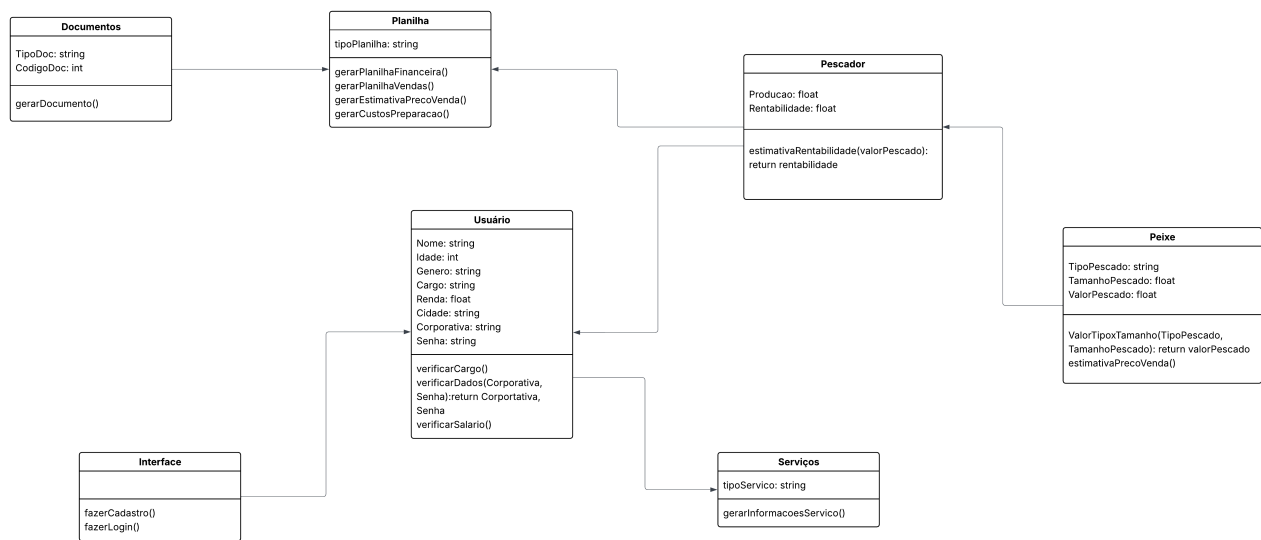


Figura 1 – Diagrama UML do projeto

4.2 Diagrama entidade relacionamento (ER)

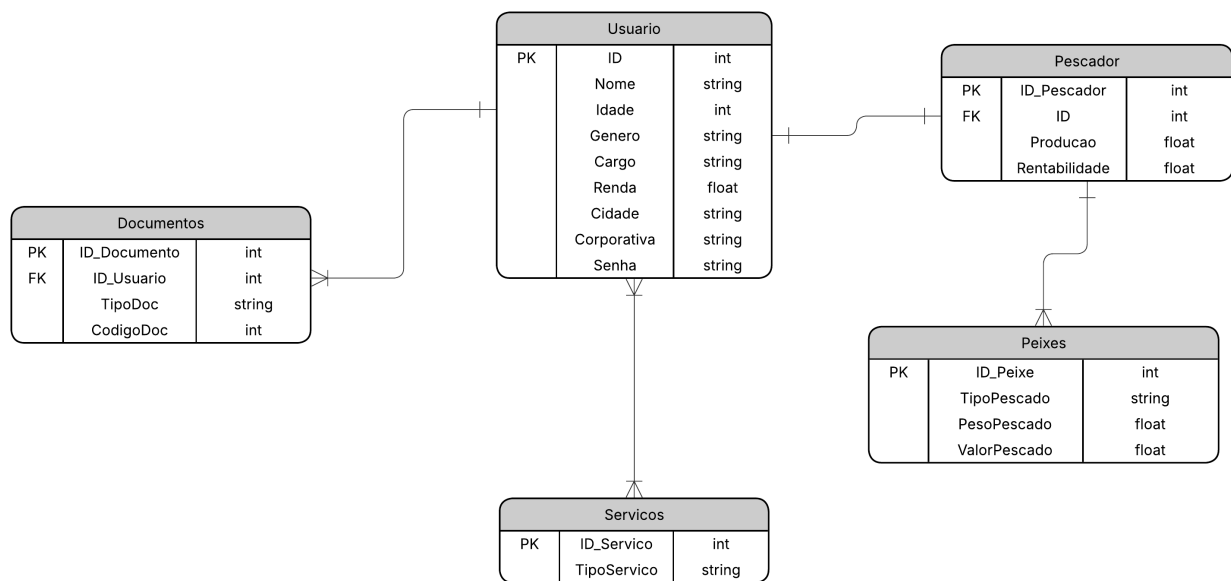


Figura 2 – Diagrama ER do projeto

5 Fluxo básico de funcionamento

5.1 Diagrama de casos de uso

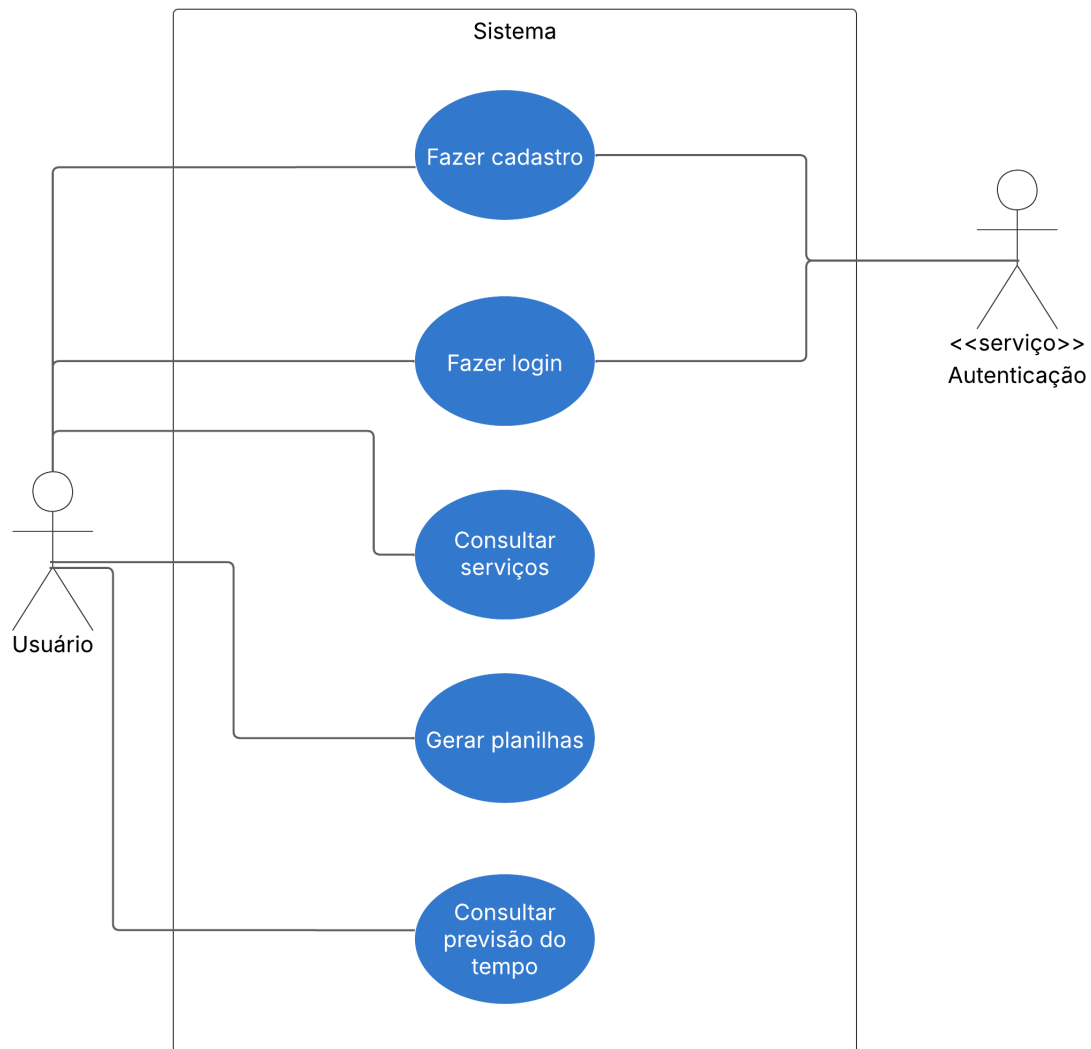


Figura 3 – Diagrama de Casos de Uso do projeto

6 Orçamento

6.1 Custo do desenvolvimento

Cobrando R\$100,00 a hora. Por dia seriam utilizadas pelo menos 2 horas de trabalho, considerando que seriam trabalhados 5 dias da semana no mínimo 1 hora durante 3 meses, o custo final do desenvolvimento seria por volta de R\$3000,00.

6.2 Custo de manutenção

Mensalmente seria cobrado o custo mensal da hospedagem do site, que seria cerca de R\$300,00. Além do custo mensal de manutenção, que seria cerca de 2 horas de trabalho por mês exclusivamente para manutenção,

o que sairia cerca de R\$200,00. Resultando um total de R\$500,00 mensais.

6.3 Custo total

O preço total do projeto sairia pelo preço total R\$3000,00 mais R\$500,00 por mês que o cliente continuar contratando os serviços de manutenção.

7 Cronograma detalhado por semana

Semana	Datas	Atividade
1	15/09 – 19/09	Planejamento do modelo do front-end
2	22/09 – 26/09	Criação do esqueleto do front-end
3	29/09 – 03/10	Planejamento e criação do banco de dados
4	06/10 – 10/10	Planejamento do modelo do back-end
5	13/10 – 17/10	Criação do back-end
6	20/10 – 24/10	Aprimoramento do back-end
7	27/10 – 31/10	Aprimoramento do back-end
8	03/11 – 07/11	Aprimoramento do front-end
9	10/11 – 14/11	Aprimoramento do front-end
10	17/11 – 21/11	Conexão entre front-end, back-end e banco de dados
11	24/11 – 28/11	Revisão geral e melhorias pontuais
12	01/12 – 03/12	Entrega do projeto pronto

Tabela 1 – Cronograma de desenvolvimento do projeto

8 Realização do projeto

Nesta seção irei detalhar o cumprimento ou não cumprimento das tarefas propostas na seção 7.

8.1 Primeira semana

Objetivo: Planejamento do modelo front-end

Realizado: Foi feito um wireframe (figura 4) para representar uma base de como o site vai ser. Um wireframe é um esboço visual básico e de baixa fidelidade de um site ou aplicativo, que ilustra a estrutura e o layout de uma página ou tela.



Figura 4 – Wireframe do projeto

Além disso, também foi feito o que foi proposto para a Semana 2: foram criadas as pastas iniciais do projeto (Figura 5) e também foi criada uma estrutura inicial para o front-end, contendo uma NavBar funcional, que altera as páginas das seções do site corretamente e os conteúdos principais de cada página (Figura 6).

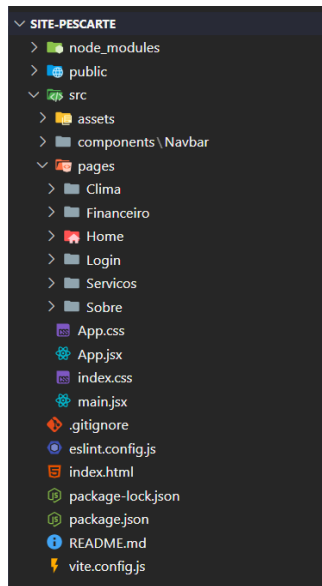


Figura 5 – Estrutura inicial das pastas do projeto



Figura 6 – Estrutura inicial do site

8.2 Segunda semana

Objetivo: Criação do esqueleto do front-end

Realizado: O que foi proposto na segunda semana já foi realizado na primeira. Na segunda semana nada foi feito em detrimento à semana de provas.

8.3 Terceira semana

Objetivo: Planejamento e criação do banco de dados

Realizado: Foi atualizado o diagrama entidade relacionamento (Figura 2). Além disso foi criada a estrutura base do banco de dados, com todas as tabelas e atributos (Figura 7).

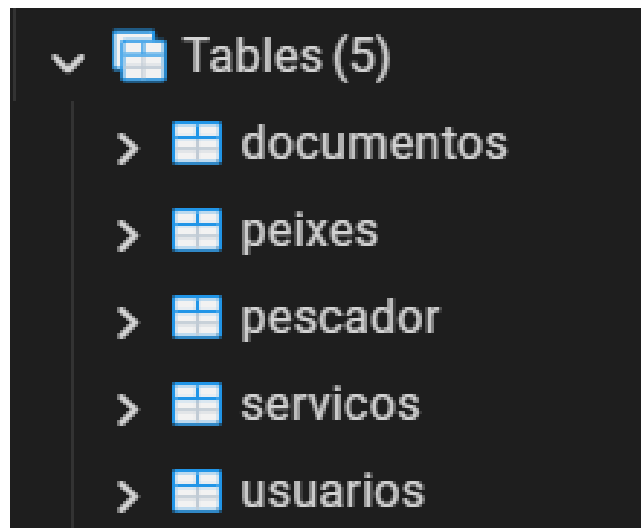


Figura 7 – Estrutura do Banco de Dados

8.4 Quarta semana

Objetivo: Planejamento do modelo do back-end

Realizado: Foi criada a estrutura do back-end, como é possível ver na figura 8.

```
src/main/java/com/pescarte/  
├─ config/  
│   ├── SecurityConfig.java // Configuração do Spring Security (JWT, CORS)  
│   └── CorsConfig.java    // Configuração global do CORS (Conexão com React)  
├─ controller/  
│   ├── AuthController.java // Endpoint para /api/auth/login  
│   ├── ClimaController.java // Endpoint para /api/clima  
│   └── DocumentoController.java // Endpoint para /api/documentos/gerar  
├─ model/  
│   └── User.java           // Entidade JPA para o usuário  
├─ repository/  
│   └── UserRepository.java // Interface Spring Data JPA  
├─ service/  
│   ├── ClimaService.java // Lógica para buscar na API de clima externa  
│   └── DocumentoService.java // Lógica para gerar documentos  
└─ security/  
    └── JwtTokenProvider.java // (No caso de usar JWT para autenticação)
```

Figura 8 – Estrutura do Back-end

Além disso também foram criadas tuplas para começar a popular as tabelas do banco de dados.

8.5 Quinta semana

ATRASSO

8.6 Sexta semana

ATRASO

8.7 Sétima semana

Objetivo: Aprimoramento do back-end

Realizado: Configuração da conexão do banco de dados com o back-end, como podemos no código [1](#)

```
1 // Configuracao do PostgreSQL
2 spring.datasource.driver-class-name=org.postgresql.Driver
3 spring.datasource.username=postgres
4 spring.datasource.password=190603
5 spring.datasource.url=jdbc:postgresql://localhost:5432/pescarte
6 spring.jpa.generate-ddl=true
7
8 // Configuracao do Hibernate (JPA)
9 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
10 spring.jpa.hibernate.ddl-auto=update
11 spring.jpa.show-sql=true
```

Listing 1 – Conexão banco de dados com back-end