UNIVERSIDADE ESTADUAL DE GOIÁS

CÂMPUS TRINDADE

SISTEMAS DE INFORMAÇÃO

MARCO ANTONIO BORGES NUNES

PROA – PROJETO DE APOIO A REGULARIZAÇÃO EM OPERAÇÕES AQUAVIÁRIAS

MONOGRAFIA DE GRADUAÇÃO

TRINDADE - GO

2024

MARCO ANTONIO BORGES NUNES

**DESENVOLVIMENTO DE SOFTWARE PARA GERAÇÃO DE DOCUMENTOS UTILIZADOS EM SERVIÇOS DE ASSESORIA NAVAL**

Monografia de Graduação a ser apresentada para a Coordenação do curso de Sistemas de Informação da Universidade Estadual de Goiás, como requisito para obtenção do título de “Bacharel em Sistemas de informação”, sob a orientação do Prof. Olegário Correia da Silva Neto.

TRINDADE - GO

2024

**AGRADECIMENTOS**

Agradeço primeiramente a Deus pela oportunidade de concluir esta etapa tão importante em minha vida acadêmica.

À minha mãe, Edivania, pelo apoio incondicional e incentivo constante durante toda minha trajetória, sendo fundamental para que eu chegasse até aqui.

Ao meu orientador, Prof. Olegário, pelas valiosas contribuições e direcionamentos que tornaram possível a realização deste trabalho.

Aos professores do curso de Sistemas de Informação, que compartilharam seus conhecimentos e experiências ao longo destes anos, em especial à Profa. Nágela, pela dedicação e paciência em todo o processo de aprendizagem.

Aos colegas de turma, pelo companheirismo e troca de experiências durante nossa jornada acadêmica, e pela paciência em me ouvir discorrer sobre o tema ao longo do último ano.

Ao meu amigo Edward Junior, assessor náutico, que se disponibilizou e foi peça fundamental no desenvolvimento deste projeto, contribuindo com seu conhecimento técnico e experiência prática.

**RESUMO**

Este trabalho apresenta o Sistema PROA, uma solução de software desenvolvida para automatizar a geração de documentos em serviços de assessoria náutica, em conformidade com as exigências da Marinha do Brasil. O sistema visa simplificar os processos de regulamentação de embarcações e obtenção de habilitação náutica, otimizando o trabalho de profissionais e empresas do setor. A arquitetura tecnológica integra Spring Boot e Angular, permitindo o gerenciamento eficiente de clientes, embarcações e prestadores de serviço. O sistema gera dinamicamente os anexos normativos baseados nas regulamentações NORMAM 211 e 212, garantindo precisão e conformidade legal. A segurança dos dados é assegurada através de autenticação via Firebase, enquanto a aplicação de diagramas UML proporcionam uma experiência de usuário intuitiva e funcional. Os resultados demonstram que o PROA contribui significativamente para a modernização do setor aquaviário brasileiro, oferecendo ganhos expressivos em eficiência, precisão e agilidade na gestão documental. Esta solução representa um importante avanço na desburocratização dos processos administrativos do setor náutico nacional.

**Palavras-chave:** Anexos, NORMAM, Serviços náuticos, Marinha do Brasil, Spring Boot, Angular, Gestão de documentos.

**ABSTRACT**

This work presents the PROA System, a software solution designed to automate document generation for nautical advisory services in compliance with Brazilian Navy requirements. The system aims to streamline vessel regulation processes and nautical licensing procedures, optimizing operations for industry professionals and companies. The technological architecture integrates Spring Boot and Angular, enabling efficient management of clients, vessels, and service providers. The system dynamically generates normative attachments based on NORMAM 211 and 212 regulations, ensuring accuracy and legal compliance. Data security is maintained through Firebase authentication, while the application of UML diagrams and interface requirements provides an intuitive user experience. Results demonstrate that PROA significantly contributes to the modernization of the Brazilian waterway sector, offering substantial gains in efficiency, accuracy, and agility in document management. This solution represents an important advance in streamlining administrative processes in the national nautical sector.

**Keywords:** Attachments**,** Nautical services, Brazilian Navy, Spring Boot, Angular, Document management.

**Lista de Ilustrações**

[Figura 1 - Diagrama de Atividade PROA Fonte: Autoria Própria 31](#_Toc182209563)

[Figura 2 – Caso de Uso “Manter Cliente” Fonte: Autoria própria 34](#_Toc182209564)

[Figura 3 - Detalhamento Gerar Documentação PF Fonte: Autoria Própria 34](#_Toc182209565)

[Figura 4 – Caso de uso “Manter Embarcação” Fonte: Autoria Própria 35](#_Toc182209566)

[Figura 5 – Detalhamento Gerar Documentação Embarcação Fonte: Autoria Própria 35](#_Toc182209567)

[Figura 6 – MER – Modelo de Entidade Relacionamento Fonte: Autoria Própria 37](#_Toc182209568)

[Figura 7 – Diagrama de Classe PROA Fonte: Autoria Própria 38](#_Toc182209569)

[Figura 9 – Protótipo de tela – Login 48](#_Toc182209570)

[Figura 10 – Protótipo de tela – Menu principal 48](#_Toc182209571)

[Figura 11 – Protótipo de tela – Cadastro de Cliente 49](#_Toc182209572)

[Figura 12 – Protótipo de tela – Listar Embarcações 49](#_Toc182209573)

[Figura 13 – Protótipo de tela – Motores 50](#_Toc182209574)

[Figura 14 – Protótipo de tela – Nota fiscal 50](#_Toc182209575)

[Figura 15 – Protótipo de tela – Cadastro Embarcação 51](#_Toc182209576)

[Figura 16 – Protótipo de tela – Emitir anexos 51](#_Toc182209577)

[Figura 17 – Protótipo de tela – Modal 52](#_Toc182209578)

[Figura 18 – Protótipo de tela – Anexo Gerado 52](#_Toc182209579)

[Figura 19 – Protótipo de tela – Emitir anexos por serviço 53](#_Toc182209580)

[Figura 20 – Protótipo de tela – Cadastro Empresa 53](#_Toc182209581)

[Figura 21 – Protótipo de tela – Cadastro Representantes / Organizações Militares 54](#_Toc182209582)

**Lista de Tabelas**

[Tabela 1 - Requisitos Funcionais do CSU1 – Manter Cliente 21](#_Toc182209677)

[Tabela 2 – Requisitos Funcionais do CSU2 – Manter Embarcação 22](#_Toc182209678)

[Tabela 3 – Requisitos Funcionais do CSU3 – Manter Usuário 23](#_Toc182209679)

[Tabela 4 – Requisitos Funcionais CSU4 – Emitir Anexos Fonte: Autoria própria 24](#_Toc182209680)

[Tabela 5 – Requisitos Funcionais CSU5 – Emitir Anexos por Serviço Fonte: Autoria própria 26](#_Toc182209681)

[Tabela 6 – Requisitos de interface Homem – Computador Fonte: Autoria própria 27](#_Toc182209682)

[Tabela 7 – Requisitos de interface externa Fonte: Autoria própria 27](#_Toc182209683)

[Tabela 8 – Requisitos de plataforma de software Fonte: Autoria própria 28](#_Toc182209684)

[Tabela 9 – Requisitos de desempenho Fonte: Autoria própria 28](#_Toc182209685)

[Tabela 10 – Requisitos de segurança Fonte: Autoria própria 29](#_Toc182209686)

[Tabela 11 – Dicionário de dados tabela ‘cliente’ Fonte: Autoria própria 45](#_Toc182209687)

[Tabela 12 – Dicionário de dados tabela ‘embarcacao’ Fonte: Autoria própria 46](#_Toc182209688)

[Tabela 14 – Dicionário de dados tabela ‘motor’ Fonte: Autoria própria 46](#_Toc182209689)

**SUMÁRIO**

[1. **INTRODUÇÃO** 10](#_Toc181774587)

[1.1. OBJETIVOS GERAIS 11](#_Toc181774588)

[1.2. OBJETIVOS ESPECÍFICOS 11](#_Toc181774589)

[1.3. JUSTIFICATIVA 11](#_Toc181774590)

[1.4. METODOLOGIA 12](#_Toc181774591)

[1.5. ESTRUTURA DO TRABALHO 13](#_Toc181774592)

[2. **REFERENCIAL TEÓRICO** 14](#_Toc181774593)

[2.1. GERENCIAMENTO DE REQUISITOS 14](#_Toc181774594)

[2.2. TECNOLOGIAS UTILIZADAS 15](#_Toc181774595)

[2.3. PADRÕES DE PROJETO 17](#_Toc181774596)

[3. **REQUISITOS DO SISTEMA** 19](#_Toc181774597)

[3.1. LEVANTAMENTO DE REQUISITOS 19](#_Toc181774598)

[**3.1.1.** **Requisitos Normativos – Normas De Autoridade Marítima (Normam)** 19](#_Toc181774599)

[**3.1.2.** **Requisitos Funcionais** 20](#_Toc181774600)

[**3.1.3.** **Requisitos Não Funcionais** 26](#_Toc181774601)

[4. **MODELAGEM DO SOFTWARE** 29](#_Toc181774602)

[4.1. DIAGRAMAS DE ATIVIDADE 29](#_Toc181774603)

[4.2. DIAGRAMAS DE CASOS DE USO 32](#_Toc181774604)

[4.3. MODELO DE ENTIDADE-RELACIONAMENTO 36](#_Toc181774605)

[4.4. DIAGRAMA DE CLASSE 38](#_Toc181774606)

[5. **CONCLUSÃO** 40](#_Toc181774607)

[5.1. RESULTADOS ALCANÇADOS 40](#_Toc181774608)

[5.2. BENEFÍCIOS OBTIDOS 40](#_Toc181774609)

[5.3. DIFICULDADES ENCONTRADAS 41](#_Toc181774610)

[5.4. MELHORIAS FUTURAS 41](#_Toc181774611)

[5.5. CONSIDERAÇÕES FINAIS 42](#_Toc181774612)

[6. **REFERÊNCIAS** 44](#_Toc181774613)

[Anexo A – Dicionário de dados. 46](#_Toc181774614)

[Anexo B – Protótipos de tela 49](#_Toc181774615)

# INTRODUÇÃO

Com o avanço tecnológico temos visto uma rápida e crescente tendência de automação e informatização dos mais diversos meios. Como observa Castells (2020), “vivemos em uma sociedade em rede, onde a tecnologia da informação é o centro das transformações sociais”. Amplos setores da sociedade avançam então com a tecnologia, fornecendo serviços inovadores para a população em geral, proporcionando assim uma significativa reorganização no modo de vida e nas expectativas das pessoas, um fenômeno que Schwab (2016) caracteriza como "a Quarta Revolução Industrial, que está criando um mundo onde os sistemas físicos e digitais se fundem de maneiras sem precedentes".

Contudo, há setores da sociedade que enfrentam uma miríade de problemas com essa rápida mudança de paradigmas. No Brasil, setores governamentais e correlatos a muito buscam se adaptar a esse novo cenário, mas uma conjuntura de fatores como o alto nível de burocracia, a dificuldade de alteração de sistemas e processos legados, a intervenção de forças diversas e interesses políticos distintos geram uma prestação de serviço onerosa, burocrática, lenta e simplesmente incompatível com as demandas e expectativas da sociedade atual. Essa realidade é evidenciada em estudos como o relatório do Banco Mundial "Doing Business 2020", que posicionou o Brasil na 124ª posição entre 190 países no quesito facilidade de fazer negócios, destacando especialmente a complexidade burocrática do país (World Bank, 2020).

Um exemplo prático desse tema, que abordaremos no decorrer desse projeto, pode ser observado nos diversos serviços prestados pela Marinha do Brasil, onde ainda hoje são requeridos um conjunto de documentos impressos, assinados e com firma registrada para abertura de processos como habilitação ou regularização, até mesmo em processos de renovação onde todos os dados já existem junto a Marinha tal documentação ainda é exigida de forma física e presencial. Até pouco tempo atrás sequer o preenchimento digital dos dados era aceito, forçando os usuários do serviço a preencher de próprio punho toda a documentação.

Dentre tantos desafios, como podemos utilizar dos meios de tecnologia para facilitar a vida de pessoas que muitas vezes buscam apenas uma forma rápida e eficiente de estar na legalidade?

OBJETIVOS  
  
A seguir serão apresentados os objetivos gerais e específicos do trabalho.

## OBJETIVOS GERAIS

Desenvolver um sistema de cadastro de usuários, embarcações e prestadores de serviço que possibilite a geração automatizada de documentos solicitados pela Marinha do Brasil para cada respectivo serviço prestado pelo órgão.

## OBJETIVOS ESPECÍFICOS

* Analisar os requisitos legais e normativos estabelecidos pela Marinha do Brasil para a regularização de embarcações, identificando os documentos necessários e os procedimentos exigidos.
* Identificar as principais demandas e necessidades dos assessores náuticos no que diz respeito aos processos de cadastro de usuários e geração dos documentos para emissão de habilitação tipo Arrais amador e Motonauta.
* Projetar e desenvolver uma interface intuitiva e amigável para o sistema, com funcionalidade que permitam o cadastro eficiente de usuários e a geração automatizada de documentos conforme as exigências legais.
* Projetar e desenvolver uma base de dados robusta e segura que suporte o armazenamento e gerenciamento eficiente das informações dos usuários e demais dados necessários.
* Implementar recursos de segurança que permitam a validação automática dos dados inseridos, bem como a geração automática e precisa dos documentos a serem emitidos.

## JUSTIFICATIVA

A introdução apresentada contextualiza de maneira clara a problemática a ser abordada, com destaque para a lacuna entre a rapidez da evolução tecnológica e a lenta e morosa adaptação dos setores governamentais a essa evolução. O exemplo prático fornecido ilustra bem a incompatibilidade entre a demanda da sociedade por praticidade e eficiência em contraste com a burocracia adotada pelos órgãos competentes, em nosso caso específico a Marinha do Brasil. Tal cenário evidencia a necessidade de soluções que integrem a tecnologia aos processos administrativos, proporcionando uma experiência ágil e simplificada aos usuários.

Portanto, este projeto de T.C. propõe a investigação e desenvolvimento de um sistema capaz de permitir o cadastro e manutenção das informações dos usuários, de suas embarcações e demais dados correlatos e a partir desses dados preparar de forma ágil a documentação exigida pela Marinha do Brasil, de acordo com o serviço a ser solicitado. A relevância deste projeto reside na sua capacidade de promover uma significativa melhoria na eficiência dos serviços prestados, contribuindo para a modernização e desburocratização de um setor essencial para a navegação comercial e turística no país.

## METODOLOGIA

As soluções propostas para resolução dos problemas apresentados foram obtidas através do processo de levantamento de requisitos, que consistiu no primeiro momento em ações junto ao potencial usuário a fim de se obter um melhor entendimento das regras de negócio envolvidas. Para isso foram realizadas entrevistas, onde um questionário foi elaborado a partir das informações colhidas.

Aos requisitos estão associados os principais problemas do desenvolvimento de software. Requisitos que não refletem as reais necessidades dos usuários, incompletos e/ou inconsistentes, mudanças em requisitos já acordados e a dificuldade para conseguir um entendimento comum entre usuários e desenvolvedores são as principais dificuldades relatadas, provocando retrabalho, atrasos no cronograma, custos ultrapassados e a insatisfação dos clientes e usuários de software (Blaschek, 2002)

Uma vez em posse dessas informações, foram iniciadas as demais etapas do processo de desenvolvimento, sendo construída a modelagem inicial a partir dos requisitos levantados anteriormente e na sequência iniciou-se as etapas de desenvolvimento e validação.

O presente trabalho pode ser classificado como um estudo de caso, se observado do ponto de vista técnico, ou como uma pesquisa exploratória aplicada, se baseado no ponto de vista da natureza de sua concepção. A metodologia utilizada consistiu na coleta e análise de informações sobre uma organização, com objetivo de identificar possíveis melhorias a serem realizadas por meio de um software a ser desenvolvido.

Com isso, buscou-se a aplicação prática dos conhecimentos obtidos em sala para obtenção de uma solução, que resultou no desenvolvimento de protótipos para um sistema para gerenciamento dos dados de clientes e geração automatizada de documentos a partir desses dados.

## ESTRUTURA DO TRABALHO

Este documento está organizado em seis capítulos, conforme descrito a seguir:

1. **Introdução**

Apresenta o contexto e a problemática que deram embasamento e motivação ao desenvolvimento do PROA. Também expõe os objetivos gerais e específicos, a justificativa do projeto e a metodologia utilizada para a coleta e análise das informações necessárias.

1. **Referencial Teórico**

Fornece a base teórica necessária para o entendimento do trabalho, abordando os principais conceitos de gerenciamento de requisitos e as tecnologias utilizadas, como Spring Boot e Angular, além de descrever padrões de projeto que fundamentam o desenvolvimento do sistema.

1. **Requisitos do Sistema**

Apresenta o levantamento dos requisitos realizado, com enfoque nos requisitos normativos e funcionais, conforme as normas da Marinha do Brasil e nos requisitos não funcionais, abordando aspectos de segurança e desempenho.

1. **Modelagem do Software**

Descreve a modelagem do sistema, incluindo diagramas de atividade, casos de uso, modelo entidade-relacionamento e diagrama de classe, essenciais para o entendimento da estrutura e do funcionamento do sistema PROA.

1. **Conclusão**

Apresenta os resultados obtidos com o desenvolvimento do sistema, os benefícios identificados, as dificuldades enfrentadas e as possíveis melhorias futuras para a evolução do projeto.

1. **Referências**

Lista as fontes bibliográficas e documentos que fundamentaram o desenvolvimento do trabalho e a elaboração dos conteúdos apresentados nos capítulos anteriores.

# REFERENCIAL TEÓRICO

Neste capítulo será demonstrado o referencial teórico utilizado neste projeto, demonstrando as características e conceitos dos fundamentos de especificação do software bem como as tecnologias utilizadas no projeto.

## GERENCIAMENTO DE REQUISITOS

Segundo Sommerville (2011): "Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento.

Adicionalmente, podemos definir o Gerenciamento de Requisitos como um conjunto de atividades que ajuda a equipe do projeto a identificar, controlar e rastrear requisitos e mudanças nos requisitos em qualquer momento à medida que o projeto prossegue (Wiegers e Beatty, 2013). Quando a organização não dispõe deste processo formalmente definido e amplamente divulgado, os desenvolvedores elaboram as especificações de requisitos de uma forma empírica, executando atividades não padronizadas e definidas individualmente. Se isto ocorre, a qualidade da especificação dependerá exclusivamente da experiência e formação das pessoas, havendo assim uma elevada probabilidade de ocorrerem conflitos e retrabalho (BLASCHEK, 2014)

O uso de ferramentas que ajudam na gestão de requisitos de software traz diversos benefícios, criando um ambiente que favorece a colaboração e a responsabilidade. Organizações que adotam essas ferramentas e mantêm uma documentação precisa e detalhada dos requisitos conseguem dividir tarefas de forma mais eficaz, estimar prazos com maior exatidão e, assim, ganham tempo para gerenciar riscos e realizar manutenções necessárias devido às novas alterações. No gerenciamento de requisitos de software, existem várias técnicas estabelecidas para a coleta desses requisitos. Um levantamento eficaz é aquele que resulta em uma definição clara do projeto, na conclusão bem-sucedida do mesmo, na apresentação das informações essenciais para um diagnóstico preciso e na proposição de soluções inteligentes. Como afirma Moraes (2014): “As técnicas de levantamento de requisitos têm por objetivo superar as dificuldades relativas a esta fase. Todas as técnicas possuem um conceito próprio e suas respectivas vantagens e desvantagens, que podem ser utilizadas em conjunto pelo analista.”

Estabelecidos os conceitos utilizados nos requisitos e em seu gerenciamento, segue agora algumas das técnicas que podem ser empregadas no levantamento de requisitos.

Entrevistas: Uma entrevista pode ser definida como uma reunião entre um Stakeholder e o analista de requisitos que tem como objetivo auxiliar ambas as partes na melhor compreensão dos problemas envolvido, nas necessidades do usuário e no esclarecimento dos objetivos envolvidos no processo. (SOMMERVILLE, 2011)

Questionário: Utilizado como complemento a outras formas de levantamento de requisitos, os questionários são especialmente importantes em análises que abrangem uma grande quantidade de usuários, pois permite coletar informações de forma eficiente e objetiva. São especialmente úteis quando não há viabilidade de entrevistar todos os potenciais usuários envolvidos no projeto. (WIEGERS & BEATTY, 2013)

Cenários: Composto por um conjunto de descrições do possível funcionamento do sistema em determinadas situações, os Cenários incluem uma sequência de passos descrevendo a interação entre o usuário e o sistema, bem como o contexto que envolve essa interação. Seu objetivo é que os stakeholders consigam visualizar como o sistema se comportará em diferentes situações. (SOMMERVILLE, 2011)

Prototipagem: Considerada uma técnica avançada, a construção de protótipos consiste em uma técnica para obtenção de informações específicas sobre os requisitos através da demonstração de um modelo relativamente funcional do sistema antes de realmente construí-lo por completo ((PRESSMAN & MAXIM, 2016).

## TECNOLOGIAS UTILIZADAS

Java SpringBoot

O Spring Boot tem se tornado uma plataforma fundamental para a criação de aplicações Java modernas, especialmente em cenários de microsserviços e sistemas RESTful. De acordo com Turnquist (2017), ele torna o processo de configuração e desenvolvimento de aplicações Java muito mais simples, oferecendo uma abordagem baseada em convenções, que elimina a necessidade de especificar manualmente um grande número de parâmetros e dependências que eram comuns anteriormente.

Essa ferramenta permite o desenvolvimento de aplicações autônomas que podem ser executadas sem a exigência de configurar um servidor de aplicação separado, como o Tomcat, pois já vem integrado. Sua popularidade no desenvolvimento de APIs RESTful se deve à facilidade na integração com bibliotecas e à sólida estrutura para injeção de dependências, além do suporte nativo ao JSON.

Angular

Angular é um framework voltado para o desenvolvimento front-end, bastante popular na criação de aplicações web modernas e eficientes. Criado e mantido pelo Google, ele utiliza uma arquitetura orientada a componentes, o que torna a modularização, o reuso de código e a manutenção das aplicações mais simples. Um dos grandes diferenciais do Angular é oferecer uma solução abrangente para o desenvolvimento de aplicações, incluindo recursos para gerenciamento de rotas, formulários reativos, injeção de dependências e interação com APIs externas (Freeman, 2019).

PDF-LIB

A biblioteca PDF-LIB é uma ferramenta TypeScript leve e eficaz para a criação e edição de arquivos PDF diretamente no navegador, o que a torna ideal para uso com Angular, dispensando a necessidade de manipulação no backend para esse contexto. Permite operações como preenchimento de formulários, adição de texto e imagens, e outras manipulações, todas realizadas de forma offline e segura, sem dependências externas (PDF-LIB, 2023). Sua aplicação em Angular possibilita, por exemplo, a personalização e geração de documentos PDF a partir de dados fornecidos pelos usuários, proporcionando uma solução prática e acessível para necessidades de manipulação de documentos no frontend.

PostGre

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional (SGBD) de código aberto, conhecido por sua solidez, adaptabilidade e conformidade com os padrões SQL. Conforme indicado pelo PostgreSQL Global Development Group (2023), ele foi desenvolvido para processar grandes volumes de dados e é utilizado por empresas de variados setores devido à sua habilidade em realizar operações complexas, como transações ACID e consultas sofisticadas. Ademais, o PostgreSQL permite o uso de extensões, possibilitando a incorporação de funcionalidades personalizadas para atender a diversas demandas das aplicações (Documentação do PostgreSQL, 2023).

Insomnia

O Insomnia é uma ferramenta de código aberto bastante popular para realizar testes e desenvolver APIs REST e GraphQL. Com uma interface clara e intuitiva, essa aplicação permite que os desenvolvedores façam requisições HTTP, testem endpoints e monitorem as respostas, simplificando a integração entre front-end e back-end nos projetos de software. Ademais, o Insomnia disponibiliza suporte para autenticação, gerenciamento de cookies e variáveis de ambiente, o que contribui para a automação e controle em diversos ambientes de desenvolvimento e produção (Kanjilal, 2022).

GitHub

O GitHub é uma plataforma destinada ao armazenamento de código-fonte e ao gerenciamento de versões de forma distribuída, sendo amplamente utilizada para a colaboração em projetos de desenvolvimento de software. Baseado no sistema Git, ele possibilita que equipes trabalhem simultaneamente em diferentes seções de um projeto, facilitando o controle das alterações e a integração contínua do código. Além de funcionar como um repositório, o GitHub oferece várias funcionalidades, como rastreamento de problemas (issues), revisão de código e integração com serviços externos, tornando-se uma ferramenta indispensável para o desenvolvimento de software contemporâneo. Segundo Loeliger e McCullough (2012), o GitHub melhora o fluxo de trabalho permitindo que os desenvolvedores façam pull requests, o que promove revisões antes da inclusão de novas funcionalidades no código principal.

Adobe Acrobat

Adobe Acrobat é uma ferramenta amplamente utilizada para criar, visualizar, editar e assinar arquivos Portable Document Format (PDF). Segundo a própria Adobe, o Acrobat “permite converter documentos em arquivos PDF, editar, revisar e compartilhar, além de proteger documentos com criptografia” (Adobe, 2023). Essa versatilidade faz do Acrobat a escolha de empresas e indivíduos para gerenciar documentos digitais com segurança e eficiência.

## PADRÕES DE PROJETO

MVC (Model-View-Controller)

O padrão Model-View-Controller (MVC) é uma arquitetura de software bastante utilizada no desenvolvimento de aplicações web. De acordo com Gamma et al. (1994, p. 4), "o MVC divide o modelo do domínio, a apresentação e as ações baseadas na entrada do usuário em três classes distintas".

Reenskaug e Coplien (2009) comentam que o MVC foi inicialmente criado para aplicações desktop, mas com o passar do tempo passou por adaptações, tornando-se um dos pilares da arquitetura web. Os autores mencionam que "a finalidade do MVC é criar uma distinção clara entre a lógica de negócios e a lógica de apresentação em uma aplicação".

Em relação a uma aplicação web contemporânea, o MVC pode ser estruturado da seguinte maneira:

🡪 Model: Este componente representa os dados e a lógica de negócios. No caso de um sistema que utiliza Spring Boot e PostgreSQL, Walls (2022, p. 35) ressalta que "o Spring Data JPA facilita a camada de persistência, permitindo que os desenvolvedores se concentrem na definição das interfaces de repositório, enquanto o framework gera implementações em tempo real".

🡪 View: A camada de Visualização é responsável por apresentar os dados ao usuário. Freeman (2022, p. 12) esclarece que "o Angular, como um framework de front-end, opera principalmente na camada de View do padrão MVC, oferecendo uma estrutura sólida para desenvolver interfaces de usuário dinâmicas e responsivas".

🡪 Controller: A função do Controlador é gerenciar a transferência de dados entre o Modelo e a Visualização. De acordo com Johnson et al. (2022, p. 89), "No Spring Boot, os controladores costumam ser classes marcadas com @RestController, que tratam das solicitações HTTP e coordenam a interação entre o cliente (Angular) e o servidor".

Seshadri e Green (2021, p. 15) descrevem como a integração desses componentes em um sistema web contemporâneo acontece: "A junção do Angular no front-end com o Spring Boot no back-end proporciona uma clara divisão de responsabilidades, onde o Angular cuida da interface do usuário enquanto o Spring Boot se ocupa da lógica de negócios e da persistência dos dados".

# REQUISITOS DO SISTEMA

## LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma fase crítica no desenvolvimento de software, onde são identificadas e documentadas as expectativas e necessidades dos usuários e stakeholders (SOMMERVILLE, 2011). Segundo Pressman e Maxim (2016, p. 131), "nesta etapa, são definidos requisitos funcionais, que especificam as funcionalidades e ações que o sistema deve realizar, e requisitos não-funcionais, que detalham características de desempenho, usabilidade, segurança e compatibilidade".

## **Requisitos Normativos – Normas De Autoridade Marítima (Normam)**

Emitidas pela Diretoria de Portos e Costas (DPC) com vigência em todo território nacional, as Normas de Autoridade Marítima (NORMAM) são portarias que regulamentam e estabelecem as diretrizes de segurança e padronização para a navegação e operações aquaviárias no Brasil. Elas asseguram a conformidade de atividades e embarcações aos requisitos legais e operacionais necessários à segurança dos envolvidos e proteção do meio ambiente(DPC, 2024)**.** Neste projeto, as NORMAM 211 e 212 foram fundamentais para definir os requisitos técnicos do sistema e desenvolver funcionalidades específicas, como emissão e gestão de documentos exigidos para a regularização de embarcações.

* **NORMAM 211 –** Esta norma define os requisitos de segurança e regularização de embarcações, bem como os procedimentos e documentos obrigatórios para o cadastro e operação de embarcações de uso recreativo e comercial.
* **NORMAM 212 –** Focada em regulamentações específicas para motos aquáticas e motonautas, esta norma abrange os critérios para habilitação, segurança, e operações de embarcações da categoria, visando a segurança dos condutores e terceiros.

As NORMAM 211 e 212 orientaram diretamente o levantamento de requisitos do sistema PROA, pois estabeleceram critérios de conformidade e segurança específicos para o tipo de embarcação e operação. A NORMAM 211 forneceu diretrizes para o registro e controle de embarcações de esporte e recreio, essencial para atender aos requisitos legais. Já a NORMAM 212, com foco em motos aquáticas, exigiu a implementação de regras específicas para emissão de documentos e processos de habilitação de motonautas, atendendo às normas de segurança e proteção da Marinha do Brasil (DPC, 2024).

### **Requisitos Funcionais**

CSU1- Manter Cliente

Atores: Prestador de serviço  
Categoria: Primário

Descrição:

Este caso de uso ocorre quando um cliente solicita um determinado serviço para o prestador. Para cada tipo de serviço há um conjunto de dados que devem ser preenchidos, de forma que se deve então conduzir um cadastro completo do cliente em sistema, para que os dados estejam disponíveis quando forem necessários em etapas futuras.

Requisitos Funcionais:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.** | **Descrição** | **Categoria** | **Prioridade** |
| RFUN 1.1 | Permitir aos Prestadores a manutenção completa do cadastro de seus clientes (Inclusão, alteração e exclusão) | Evidente | Alta |
| RFUN 1.2 | Validar os dados inseridos no cadastro de acordo com os padrões legais. | Evidente | Alta |
| RFUN 1.3 | Registrar Log com os dados do cadastro (Data, Hora, operador, etc) | Evidente | Alta |

Tabela 1 - Requisitos Funcionais do CSU1 – Manter Cliente  
Fonte: Autoria própria

CSU2- Manter Embarcação

Atores: Prestador de serviço  
Categoria: Primário

Descrição:

Este caso de uso ocorre quando um cliente solicita um serviço relacionado a embarcações para o prestador. Para cada tipo de serviço há um conjunto de dados que devem ser preenchidos, de forma que se deve então conduzir um cadastro completo do Cliente e da Embarcação em sistema, para que os dados estejam disponíveis quando forem necessários em etapas futuras.

Requisitos Funcionais:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.** | **Descrição** | **Categoria** | **Prioridade** |
| RFUN 2.1 | Permitir aos Prestadores a manutenção completa do cadastro de Embarcações (Inclusão, alteração, exclusão e listagem) | Evidente | Alta |
| RFUN 2.2 | Caso a Embarcação já esteja motorizada, deve ser realizado o cadastro do motor. | Evidente | Alta |
| RFUN 2.3 | As informações referentes a motor podem ser alteradas por todos com acesso a embarcação | Evidente | Média |
| RFUN 2.4 | Validar os dados inseridos no cadastro de acordo com os padrões legais. | Evidente | Alta |
| RFUN 2.5 | Registrar Log com os dados do cadastro (Data, Hora, operador, etc) | Evidente | Média |

Tabela 2 – Requisitos Funcionais do CSU2 – Manter Embarcação  
Fonte: Autoria própria

CSU3- Manter Empresa/Prestador

Atores: Prestador de serviço  
Categoria: Primário

Descrição:

Este caso de uso ocorre quando um prestador deseja iniciar a utilização do sistema. Para isso, ele deverá realizar o cadastro da sua empresa com os dados pertinentes, realizar o próprio cadastro como cliente e o cadastro como cliente dos demais funcionários e sinalizar que estes atuam pela empresa.

Requisitos Funcionais:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.** | **Descrição** | **Categoria** | **Prioridade** |
| RFUN 3.1 | Permitir o cadastro da empresa com seus respectivos dados. | Evidente | Alta |
| RFUN 3.2 | Permitir vincular colaboradores cadastrados como clientes para que componham os dados dos anexos onde forem pertinentes. | Evidente | Alta |
| RFUN 3.3 | Garantir que os clientes cadastrados por prestadores da empresa estejam vinculados a ela. | Evidente | Alta |
| RFUN 3.4 | Não permitir que usuários de uma empresa a visualização de dados de clientes de outras empresas. |  |  |

Tabela 3 – Requisitos Funcionais do CSU3 – Manter Empresa/Prestador  
Fonte: Autoria Própria

CSU4- Emitir Anexos

Atores: Prestador de serviço  
Categoria: Primário

Descrição:

Este caso de uso ocorre quando Cliente e Embarcação estão devidamente cadastradas, podendo então emitir os anexos necessários a partir de uma seleção individual. O cliente ou prestador deve acessar a respectiva página e selecionar qual anexo é necessário e as informações do cliente ou embarcação para consulta. Feito isso o sistema irá disponibilizar uma nova página contendo o anexo preenchido com os dados para impressão.

Requisitos Funcionais:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.** | **Descrição** | **Categoria** | **Prioridade** |
| RFUN4.1 | Cada prestador somente pode visualizar dados de clientes vinculados a sua própria empresa | Evidente | Alta |
| RFUN 4.2 | O prestador somente pode emitir anexos de seus respectivos clientes | Evidente | Alta |
| RFUN 4.3 | Anexos emitidos não devem ser salvos e nem permanecer acessíveis por outros que não o emissor. | Evidente | Média |
| RFUN 4.4 | Registrar Emissor, Data e hora de cada anexo emissão. | Evidente | Baixa |
| RFUN 4.5 | Os anexos devem ser formatados seguindo estritamente os modelos disponíveis na regulamentação. | Evidente | Alta |
| RFUN 4.6 | Os anexos que requerem reconhecimento de firma na assinatura devem ser sinalizados. | Evidente | Média |

Tabela 4 – Requisitos Funcionais CSU4 – Emitir Anexos  
Fonte: Autoria própria

CSU5- Emitir anexos por Serviço

Atores: Prestador de serviço  
Categoria: Primário

Descrição:

Este caso de uso ocorre quando Cliente e Embarcação estão devidamente cadastradas, podendo então emitir os anexos necessários para atendimento a um determinado serviço. O prestador deve acessar a respectiva página e selecionar o cliente, a embarcação e qual serviço será prestado. A partir disso o sistema deverá disponibilizar a opção de envio dos respectivos anexos daquele serviço para o e-mail do cliente ou realizar uma impressão.

Requisitos Funcionais:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.** | **Descrição** | **Categoria** | **Prioridade** |
| RFUN5.1 | Gerar os anexos necessários para um determinado serviço | Evidente | Alta |
| RFUN 5.2 | Permitir que os anexos gerados possam ser imediatamente impressos. | Evidente | Alta |
| RFUN 5.3 | Permitir que os anexos gerados possam ser encaminhados para o email cadastrado do cliente. | Evidente | Média |
| RFUN 5.4 | Anexos emitidos não devem ser salvos e nem permanecer acessíveis por outros que não o emissor. | Evidente | Média |
| RFUN 5.5 | Registrar Emissor, Data e hora de cada anexo emissão. | Evidente | Baixa |
| RFUN 5.6 | Os anexos que requerem reconhecimento de firma na assinatura devem ser sinalizados. | Evidente | Média |
| RFUN 5.7 | Os anexos devem ser formatados seguindo os modelos disponíveis na Regulamentação. | Evidente | Alta |

Tabela 5 – Requisitos Funcionais CSU5 – Emitir Anexos por Serviço  
Fonte: Autoria própria

### **Requisitos Não Funcionais**

Este item descreve os requisitos e restrições não funcionais que o sistema deve atender para garantir desempenho, segurança, usabilidade, e outros aspectos importantes que não estão diretamente relacionados às funcionalidades específicas, mas que impactam a experiência geral e a eficácia do sistema.

Requisitos/Restrições de Interface Homem-Computador (RHIC)

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RIHC1 | A interface do sistema deve conter somente as informações pertinentes a função em uso, garantindo que novos usuários possam operar o sistema com eficácia após um breve treinamento. | Todos |
| RIHC2 | O sistema deve ser compatível com os principais navegadores (Google Chrome, Mozilla Firefox, Microsoft Edge e Safari), garantindo uma experiência de uso consistente em todos eles. | Todos |

Tabela 6 – Requisitos de interface Homem – Computador  
Fonte: Autoria própria

Requisitos e Restrições de Interface Externa (RIEX)

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RIEX1 | O sistema deverá utilizar a plataforma de autenticação FireBase para permitir o acesso. | CSU1 |
| RIEX2 | Nas telas de cadastro deverá ser disponibilizado a pesquisa de CEP através da API pública “Busca CEP” para facilitar o preenchimento dos campos de endereço. | CSU1, CSU2 |

Tabela 7 – Requisitos de interface externa  
Fonte: Autoria própria

Requisitos/Restrições de Plataforma de Software (RPSW)

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RPSW1 | O sistema deve ser desenvolvido com uma arquitetura modular, utilizando os recursos do Spring Boot para o backend e Angular para o frontend. Isso permite que componentes individuais sejam atualizados ou substituídos sem afetar outras partes do sistema, facilitando a manutenção e futuras expansões. | Todos |
| RPSW2 | O sistema deve ser projetado para suportar testes automatizados. | Todos |
| RPSW3 | A implementação deve garantir que a aplicação seja compátivel com o banco de dados PostgreSQL. | Todos |

Tabela 8 – Requisitos de plataforma de software  
Fonte: Autoria própria

Requisitos/Restrições de Desempenho (RDES)

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RDES1 | A geração de anexos PDF, incluindo a coleta de dados do banco de dados e a renderização dos documentos, deve ser concluída em no máximo 10 segundos, garantindo uma experiência fluida para o usuário. | Todos |
| RDES2 | O sistema deve suportar, sem degradação perceptível de desempenho, ao menos 30 conexões simultâneas ao servidor Spring Boot, garantindo que múltiplos usuários possam operar o sistema ao mesmo tempo. | Todos |
| RDES3 | O PostgreSQL deve ser otimizado para suportar um grande volume de dados, incluindo o uso de índices, e otimização de consultas para garantir o desempenho à medida que o volume de dados cresce. | Todos |

Tabela 9 – Requisitos de desempenho  
Fonte: Autoria própria

Requisitos/Restrições de Segurança de Acesso (RSEG)

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RSEG1 | O sistema utilizará o Firebase Authentication para gerenciar a autenticação dos usuários no front-end Angular. Todos os usuários devem se autenticar via Firebase antes de acessar qualquer funcionalidade protegida do sistema. | Todos |
| RSEG2 | Após a autenticação via Firebase, um token JWT (JSON Web Token) será gerado e utilizado para manter a sessão do usuário. Esse token será enviado em todas as requisições ao backend Spring Boot para validação da identidade do usuário. | Todos |
| RSEG3 | O sistema deve implementar mecanismos automáticos de renovação de token, garantindo que as sessões permaneçam ativas enquanto o usuário estiver utilizando o sistema. Tokens expirados devem exigir que o usuário faça login novamente. | Todos |
| RSEG4 | O sistema deve garantir que todos os usuários forneçam consentimento explícito para a coleta e processamento de seus dados pessoais, conforme exigido pela Lei Geral de Proteção de Dados (LGPD). | Todos |

Tabela 10 – Requisitos de segurança  
Fonte: Autoria própria

# MODELAGEM DO SOFTWARE

A modelagem de software desempenha um papel fundamental no desenvolvimento de sistemas, pois proporciona uma visão clara e organizada da solução proposta antes de sua execução. Neste capítulo, iremos abordar os diferentes aspectos da modelagem do PROA.

Essa modelagem inclui desde a representação dos requisitos funcionais por meio de diagramas de casos de uso até a detalhada estruturação das classes e entidades do sistema. Empregamos a Linguagem de Modelagem Unificada (UML) para elaborar representações visuais que capturam a essência do sistema, suas interações e os fluxos de dados envolvidos.

## DIAGRAMA DE ATIVIDADE

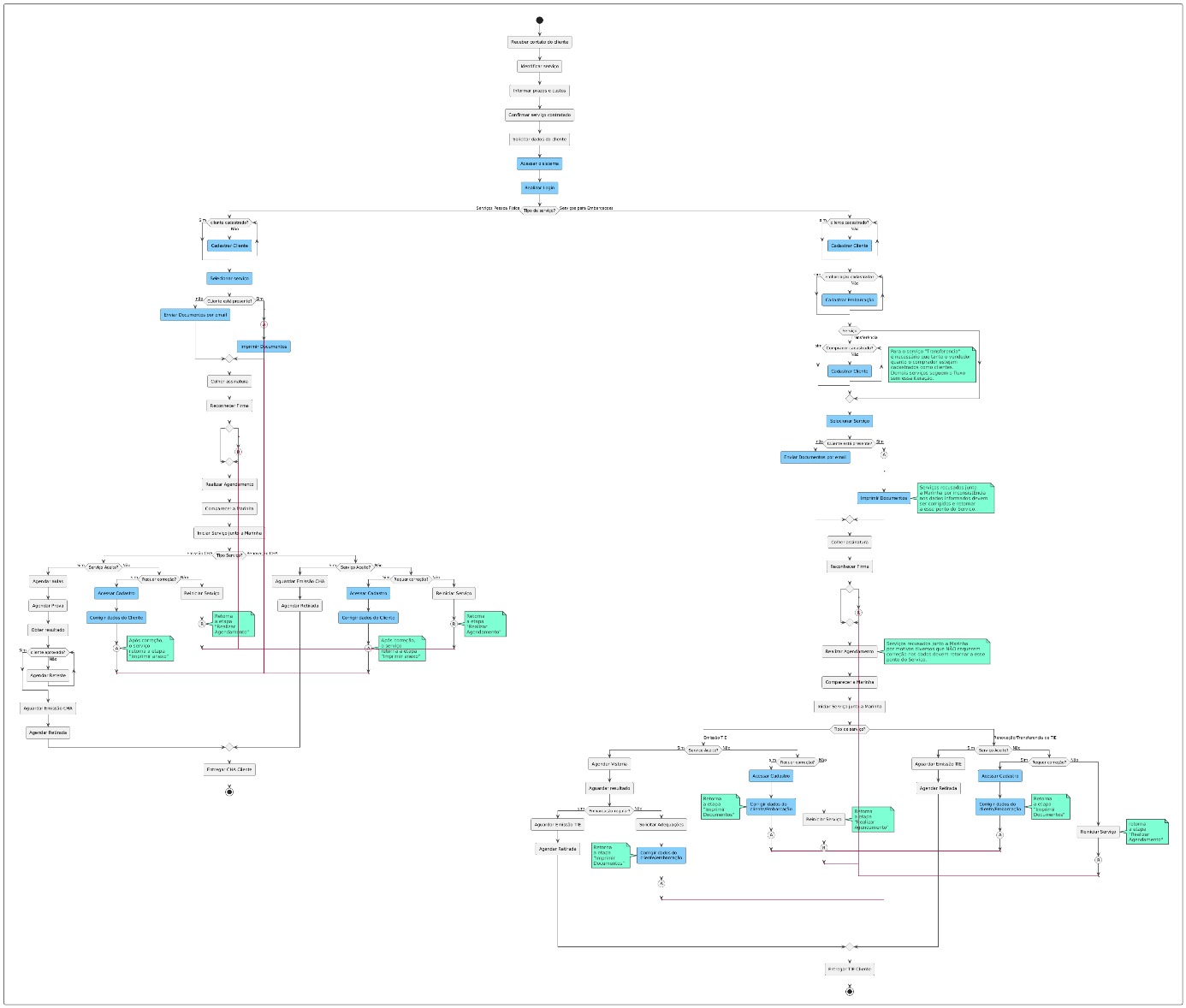
O Diagrama de Atividade é uma ferramenta essencial na modelagem de software, oferecendo uma representação visual dos fluxos de trabalho e processos dentro de um sistema (BOOCH et al., 2005). Este diagrama "permite ilustrar a sequência de ações, decisões e fluxos que ocorrem durante a execução de uma funcionalidade específica" (PRESSMAN; MAXIM, 2016, p. 847). Como destaca Sommerville (2011, p. 128), "os diagramas de atividade são particularmente úteis para modelar o fluxo de trabalho e visualizar como as atividades se relacionam umas com as outras".

Figura 1 - Diagrama de Atividade PROA  
Fonte: Autoria Própria

O diagrama de Atividade apresentado na figura 1 descreve o fluxo de trabalho detalhado para a prestação de serviços náuticos, dividindo as atividades de atendimento ao cliente em diferentes etapas, com base no tipo de serviço solicitado. As principais ações envolvem etapas de verificação, cadastramento, envio ou impressão de documentação, agendamentos e, quando necessário, correções de dados para reinício do processo. Esse fluxo é representado por duas ramificações principais: uma para serviços destinados a pessoas físicas e outra para embarcações.

**Etapas de Atendimento Inicial e Identificação de Serviço**

1. **Recebimento de Contato e Identificação do Serviço**: Inicialmente, o prestador de serviço recebe o contato do cliente, identifica o tipo de serviço solicitado, e informa prazos e custos. Uma vez que o cliente confirme a contratação, solicita-se os dados necessários para início do serviço.
2. **Acesso e Login no Sistema**: O acesso ao sistema é feito por meio de login e senha, permitindo o acesso às funcionalidades necessárias para prosseguir.

**Fluxo para Serviços de Pessoa Física**

1. **Cadastro e Seleção de Serviço**: Caso o cliente ainda não esteja cadastrado, o sistema realiza o cadastro e seleciona o serviço solicitado.
2. **Envio de Documentação e Agendamento**: Se o cliente não estiver presente, os documentos necessários podem enviados por e-mail, onde o cliente deverá imprimir e assinar e então encaminhar de volta ao prestador. Caso contrário, os documentos são impressos imediatamente. O reconhecimento de firma é obrigatório para alguns anexos e será feito pelo cliente ou pelo prestador de serviço em um cartório indicado pelo cliente. Com a documentação pronta e firma reconhecida é possível então realizar o agendamento junto a Marinha do Brasil para início do processo.
3. **Início do Serviço e Processamento**:
   * Para o serviço de emissão de Certificado de Habilitação Amadora (CHA), é necessário passar pelas aulas obrigatórias, teóricas e práticas, e por fim demonstrar capacidade em uma prova teórica de conhecimento. Caso o cliente seja reprovado é possível realizar um reteste. Já o serviço de renovação não requer nenhuma ação adicional, sendo emitido diretamente após aceitação do processo pela Marinha.
   * Caso o serviço seja recusado e precise de correção, os dados do cliente são alterados no cadastro, retornando o processo à etapa de impressão de anexos. Se recusado por qualquer motivo que não implique em correção dos dados, o processo retorna a etapa de agendamento.

**Fluxo para Serviços de Embarcações**

1. **Cadastro de Cliente e Embarcação**: Similar ao processo para pessoas físicas, o cadastro de cliente é realizado caso não exista. Além disso, um cadastro específico para a embarcação também é necessário.
2. **Tipo de Serviço**: Especificamente para o serviço de transferência de titularidade será exigido o cadastro tanto do vendedor quanto do comprador, uma vez que os documentos pertinentes ao serviço exigem ambas informações.
3. **Envio de Documentação e Agendamento**: Como nos serviços de pessoa física, caso o cliente não esteja presente, a documentação é enviada por e-mail, ou então impressa para assinatura e após reconhecimento de firma e retorno da documentação ao prestador é feito o agendamento para o comparecimento na Marinha.
4. **Execução do Serviço e Correções**:
   * No caso de emissão do TIE (Título de Inscrição de Embarcação) é necessário realização de vistoria e aguardo da aprovação e emissão do TIE. Feito isso o serviço é finalizado com a entrega do documento ao cliente.
   * Em caso de inconsistências nos dados com recusa do serviço pela Marinha, o fluxo permite a correção e retorna à etapa de impressão de anexos. Se recusado por qualquer motivo que não requira correção dos dados, o processo retorna a etapa de agendamento para uma nova tentativa.

**Conclusão e Entrega**

O processo é finalizado com a entrega do CHA ou TIE ao cliente, dependendo do tipo de serviço. Esse fluxo modularizado e ramificado demonstra uma organização lógica das etapas, assegurando que todos os requisitos de cada tipo de serviço sejam devidamente atendidos.

## DIAGRAMAS DE CASOS DE USO

Os diagramas de casos de uso são fundamentais na modelagem de sistemas, pois "representam graficamente as interações entre os usuários (atores) e o software, evidenciando suas principais funcionalidades" (FOWLER, 2014, p. 103). Como destacam Booch et al. (2005, p. 227), "eles permitem visualizar como cada ator interage com o sistema para alcançar determinados objetivos, o que facilita a compreensão dos requisitos funcionais".

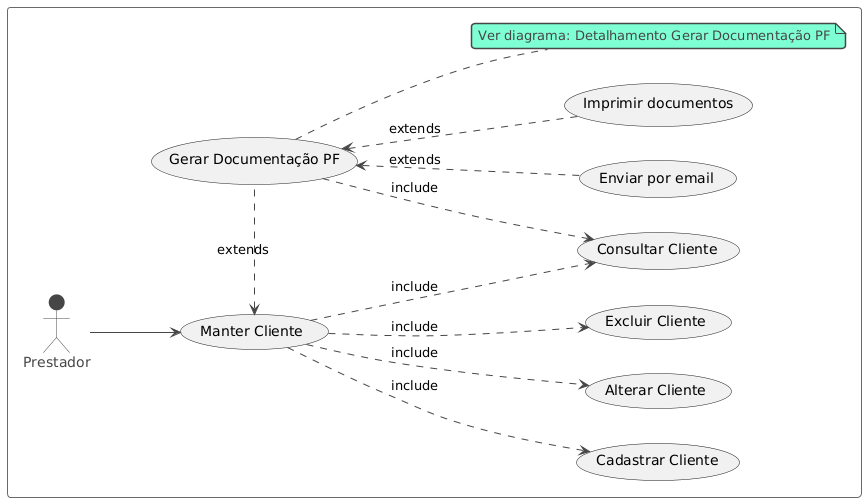


Figura 2 – Caso de Uso “Manter Cliente”  
Fonte: Autoria própria

Na Figura 2, o ator "Prestador" tem acesso ao caso de uso "Manter Cliente", que engloba as operações de cadastro, alteração, consulta e exclusão de clientes, essencial para organizar os dados dos clientes que irão solicitar o CHA. O caso de uso "Gerar Documentação PF" é uma extensão de "Manter Cliente", representando a geração de documentação personalizada para pessoas físicas (PF), onde também se incluem as funcionalidades de "Enviar por email" e "Imprimir documentos".

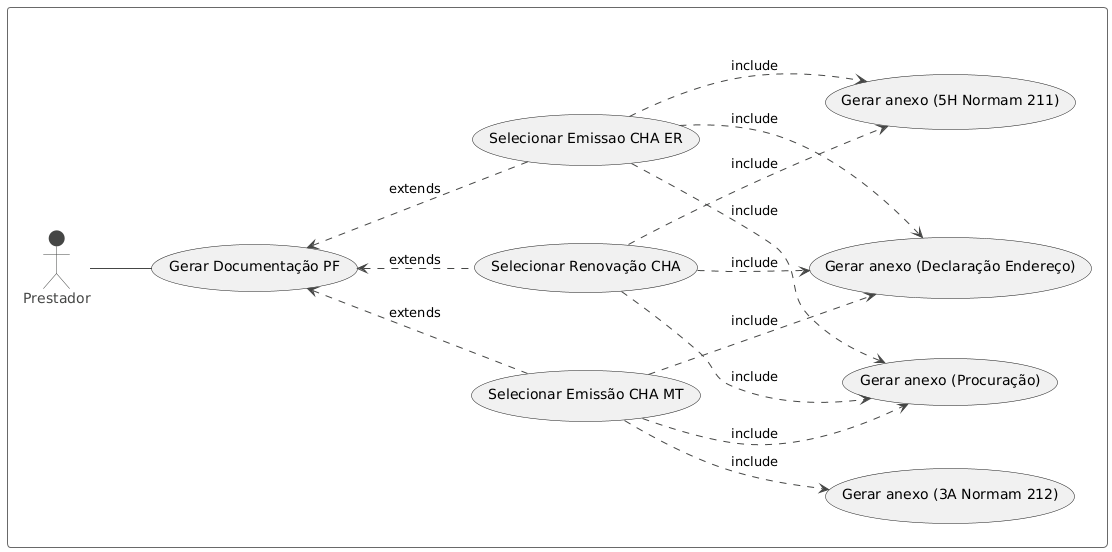


Figura 3 - Detalhamento Gerar Documentação PF  
Fonte: Autoria Própria

A figura 3 expande o caso de uso "Gerar Documentação PF", detalhando os diferentes tipos de emissão e renovação de CHA, cada uma com suas particularidades de anexos. Para a emissão CHA-MT e CHA-ER, bem como para a renovação de CHA, o sistema gera documentos como declarações e procurações além dos anexos 3A e 5H da NORMAM 212 e 211, respectivamente.

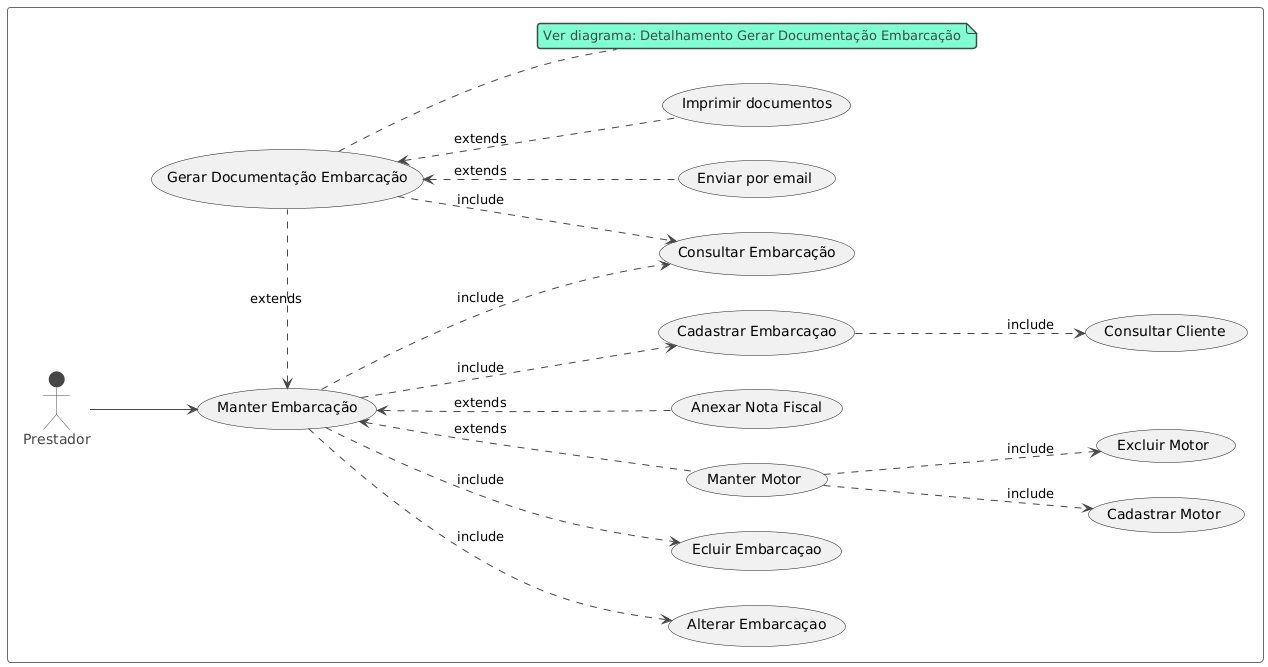


Figura 4 – Caso de uso “Manter Embarcação”  
Fonte: Autoria Própria

No Caso de uso “Manter Embarcação”, o ator "Prestador" gerencia dados e registros de embarcações e motores. Este caso de uso inclui as operações de cadastro, alteração, consulta e exclusão de embarcações, integrando uma funcionalidade de "Anexar Nota Fiscal" e acessando dados do cliente para garantir que as embarcações estejam devidamente vinculadas aos seus proprietários. O "Manter Motor" permite ao usuário cadastrar ou excluir motores, anexando esses dados à documentação da embarcação.

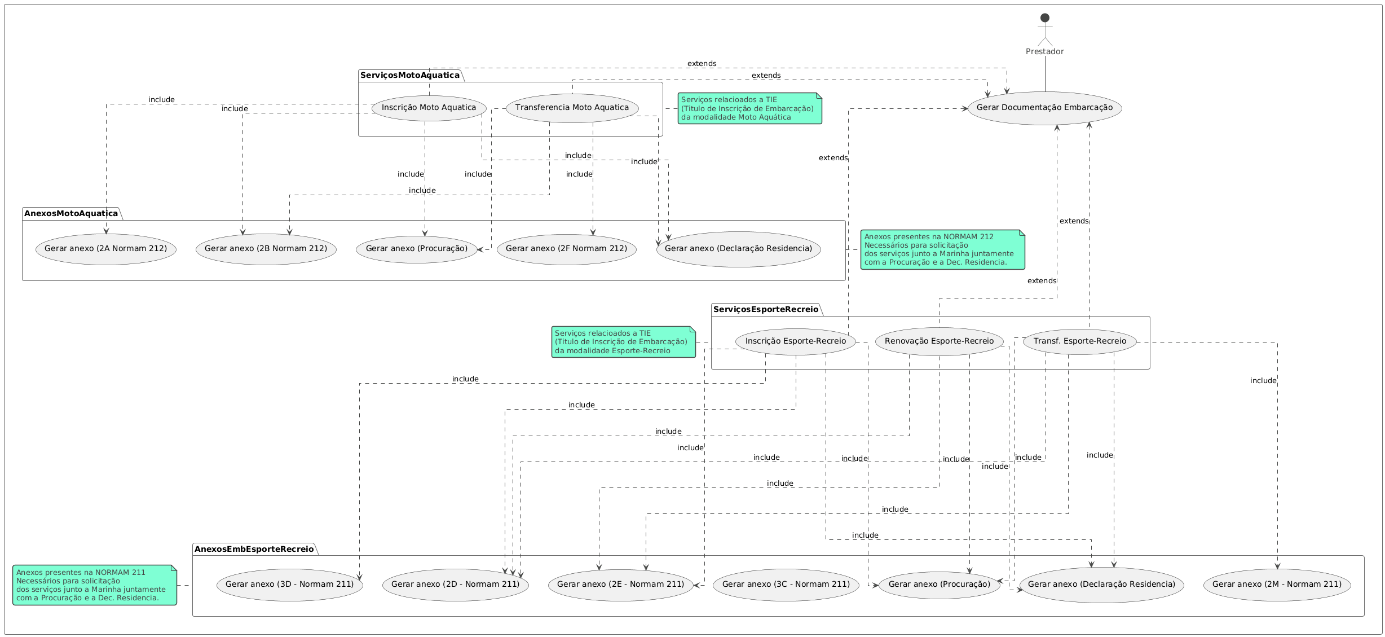


Figura 5 – Detalhamento Gerar Documentação Embarcação  
Fonte: Autoria Própria

Na Figura 5, o caso de uso "Gerar Documentação Embarcação" é detalhado para as operações de emissão e transferência de TIE nas modalidades de Moto Aquática e Esporte-Recreio. Cada serviço relacionado ao TIE exige anexos específicos, conforme normativas NORMAM 211 e 212. Para Moto Aquática, o sistema gera anexos como o "2A", "2B", e "2F" da NORMAM 212, além de uma procuração e uma declaração de residência. Similarmente, para Esporte-Recreio, os anexos incluem "2D", "2E", "3C" e "3D" da NORMAM 211, junto com a procuração e a declaração de residência.

## MODELO DE ENTIDADE-RELACIONAMENTO

O Modelo de Entidade-Relacionamento (MER) é uma representação gráfica que descreve a estrutura de dados de um sistema, identificando as entidades principais, seus atributos e as relações entre elas. Silberschatz, Korth e Sudarshan (2006) destacam que "o modelo entidade-relacionamento é uma ferramenta poderosa para o projeto conceitual de banco de dados, pois permite que os analistas e desenvolvedores visualizem o sistema em um nível abstrato, antes de se preocuparem com detalhes de implementação". Esse modelo é essencial para projetar um banco de dados organizado e eficiente, pois permite visualizar como os dados se conectam e interagem. No desenvolvimento de sistemas, o MER auxilia a equipe a entender melhor o fluxo e o armazenamento de informações, assegurando que todos os dados necessários sejam devidamente estruturados e acessíveis conforme os requisitos definidos​.

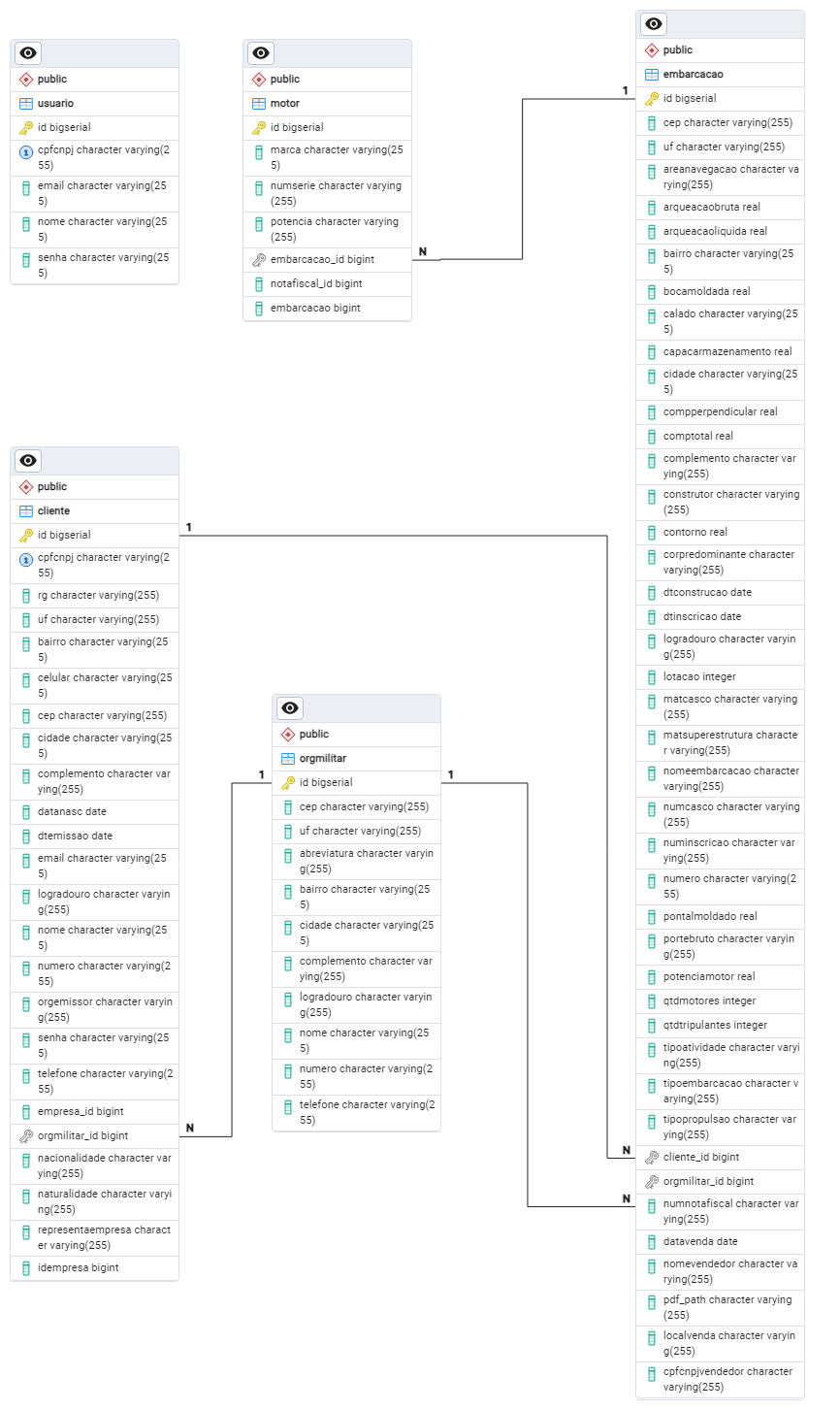


Figura 6 – MER – Modelo de Entidade Relacionamento  
Fonte: Autoria Própria

## DIAGRAMA DE CLASSE

O Diagrama de Classe é uma ferramenta central na modelagem de software orientado a objetos, pois representa a estrutura estática do sistema, detalhando as classes, atributos, métodos e relacionamentos entre elas. Este diagrama facilita a visualização da organização e hierarquia dos componentes, permitindo que a equipe de desenvolvimento compreenda como as diferentes partes do sistema se conectam e interagem. Conforme afirmam Sommerville e Sawyer (1997), "o diagrama de classes é essencial para a modelagem de software orientado a objetos, pois permite que os desenvolvedores compreendam como as diferentes partes do sistema se conectam e interagem, facilitando a manutenção e evolução futura do software". Com o Diagrama de Classe, é possível planejar e estruturar o código de forma organizada, promovendo a reutilização de componentes e facilitando futuras expansões e manutenções no sistema

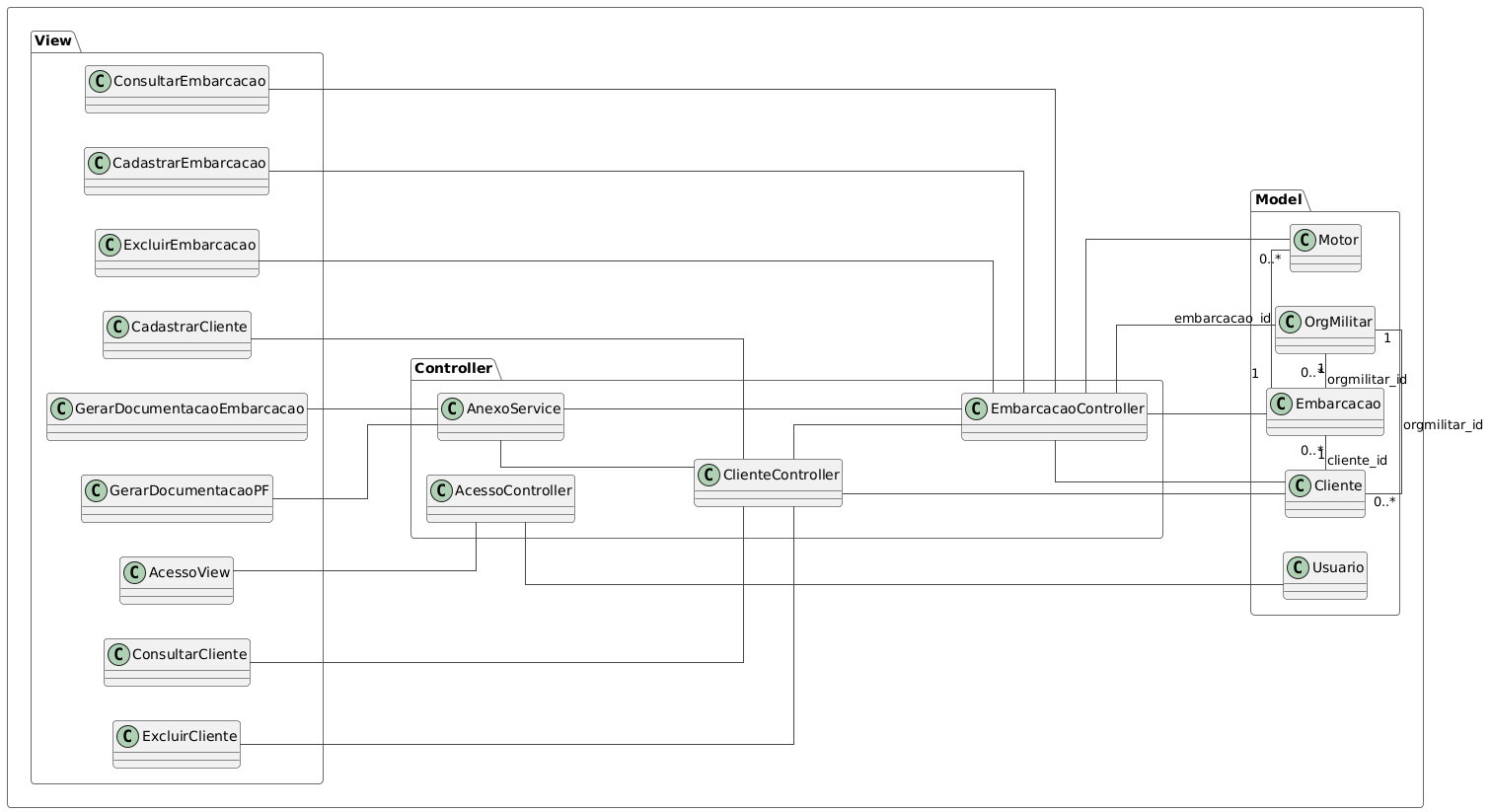


Figura 7 – Diagrama de Classe PROA  
Fonte: Autoria Própria

Esse diagrama reflete a estrutura do sistema, onde as camadas de Model, Controller, e View interagem para permitir o cadastro, consulta, e gerenciamento de dados de clientes, embarcações e documentos, além de gerenciar o acesso dos usuários. Esse design facilita o controle sobre o fluxo de informações, promovendo uma separação clara entre os dados (modelos), a lógica de negócios (controladores) e as interfaces de usuário (visões), essenciais para a conformidade com os requisitos previamente definidos.

# CONCLUSÃO

## RESULTADOS ALCANÇADOS

Como resultado do levantamento de requisitos e desenvolvimento do sistema através dos recursos previamente citados, foi possível projetar uma plataforma robusta onde será possível realizar com segurança a manutenção do cadastro de usuários e suas respectivas embarcações e a partir disso emitir os anexos necessários para solicitação de serviços junto a Marinha do Brasil. O sistema desenvolvido foi apresentado aos stakeholders que o recepcionaram com empolgação e interesse, participando não somente das etapas de levantamento de requisitos, mas acompanhando e validando cada nova etapa do desenvolvimento do projeto.

## BENEFÍCIOS OBTIDOS

Dentre os principais benefícios obtidos com o PROA, podemos citar:

* Automatização de processos: O sistema permite a geração automatizada e recorrente dos diversos anexos exigidos pela Marinha do Brasil e garante a manutenção dos dados de cada cliente para ações futuras.
* Aumento de produtividade: A disponibilidade dos diversos anexos a partir de um cadastro único do cliente e embarcação permite um aumento significativo da produtividade de todos usuários do sistema. O tempo anteriormente gasto no preenchimento manual dos anexos pode agora ser direcionado a outras ações necessárias.
* Validação de dados e redução de erros: As validações dos dados inseridos no cadastro do usuários e durante a emissão dos anexos garantem que a documentação gerada está correta e de acordo com o padrão exigido pelo órgão.
* Melhoria na comunicação: A disponibilidade de envio direto dos anexos gerados em cada serviço permite uma comunicação direta e eficaz com clientes atendidos em regiões distantes além de promover melhores práticas no atendimento aos clientes.
* Segurança: A autenticação aplicada ao projeto garante que apenas usuários com permissão terão visibilidade dos dados de clientes, protegendo os envolvidos no projeto contra o acesso e uso indevido de dados sensíveis.

## DIFICULDADES ENCONTRADAS

O desenvolvimento do PROA como era esperado enfrentou uma série de dificuldades inerentes a projetos de TC. Dentre os quais destaco a limitação imposta pela Marinha do Brasil nos anexos, não permitindo em absoluto nenhum tipo de alteração dos mesmos. Foi um desafio significativo que exigiu uma extensa análise em busca de soluções factíveis que atendessem a necessidade. Dentre as soluções consideradas, optou-se por utilizar os próprios anexos originais com adição de campos tipo formulário para os dados pertinentes. Essa solução permitiu a inserção dos dados necessários em tempo de execução sem alteração na forma e conteúdo dos anexos originais, garantindo assim a pronta aceitação do mesmo pelo órgão.

Para além disso diversas outras dificuldades foram encontradas e superadas no decorrer do desenvolvimento, como a definição de padrões de segurança, onde foi preciso considerar entre a autenticação interna proprietária ou utilização de algum autenticador externo como o Firebase e o entendimento e definição dos requisitos junto aos stakeholders foram etapas sensíveis que exigiram uma atenção especial para garantir a conclusão eficiente do projeto.

## MELHORIAS FUTURAS

O Projeto como apresentado até o momento atende a sua proposta inicial e resolve uma série de necessidades dos seus usuários. Mas há ainda uma série de melhorias futuras a serem implementadas, sendo elas:

* Simulador de avaliação e Banco de questões

Uma das etapas do processo de habilitação junto a Marinha consiste na aplicação de prova que avalia se o avaliado está apto ou não. O simulador permite que o cliente possa simular a avaliação e testar sua capacidade previamente.

* Gestão de serviços prestados e recuperação de clientes

Como os serviços da Marinha possuem validade fixa e são renováveis em intervalos de tempo é interessante que o sistema disponha desses dados e gerencie o contato junto ao cliente para renovação prévia, quando necessário.

* Expansão dos serviços oferecidos

Avaliar outros setores públicos e privados que possam tomar benefício dos recursos implementados a fim de expandir a atuação.

* Aplicação de técnicas de gestão e BI

Desenvolvimento de relatórios visuais com dados que facilitem a gestão e tomada de decisão pelos usuários do sistema.

* Adicionar como melhoria futura: Desenvolvimento e testes

## CONSIDERAÇÕES FINAIS

O PROA representa um avanço importante no processo de regularização de pilotos e embarcações junto a Marinha do Brasil, respondendo à demanda por desburocratização e rapidez em um setor ainda bastante dependente de processos manuais e presenciais. A plataforma disponibiliza uma solução para a criação dos documentos necessários e para a gestão dos dados envolvidos, reduzindo a necessidade de interações presenciais e acelerando o atendimento dos clientes.

Por meio das fases de levantamento de requisitos, modelagem e desenvolvimento, o sistema PROA conseguiu atingir seus principais objetivos, incluindo a disponibilização de uma interface intuitiva e a automação dos processos de geração de documentos. O sistema se mostrou eficiente ao permitir o registro e a manutenção centralizada e segura dos dados dos clientes e das embarcações. Essa abordagem não apenas melhora a produtividade dos usuários do sistema, mas também ajuda a diminuir erros e retrabalho, aumentando assim a confiabilidade das informações.

Adicionalmente, a adoção de mecanismos de segurança, como autenticação via Firebase e proteção de dados conforme as diretrizes da LGPD, reforça o compromisso do projeto com a privacidade e segurança das informações sensíveis dos usuários. Isso é um fator crucial para que o sistema seja bem aceito pelos seus usuários.

# REFERÊNCIAS

BANCO MUNDIAL. Doing Business 2020: Comparing Business Regulation in 190 Economies. Washington, DC: World Bank, 2020. Disponível em: <https://openknowledge.worldbank.org/handle/10986/32436>. Acesso em: 05 nov. 2024.

BLASCHEK, José Roberto. O papel dos requisitos no insucesso dos projetos de software. 2002. Dissertação (Mestrado em Informática) - Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2002.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. The Unified Modeling Language User Guide. 2. ed. Boston: Addison-Wesley Professional, 2005.

CASTELLS, Manuel. A Sociedade em Rede. 22. ed. São Paulo: Paz e Terra, 2020.

DIRETORIA DE PORTOS E COSTAS (DPC). NORMAM 211/DPC - Normas da Autoridade Marítima para Embarcações de Esporte e Recreio. Disponível em: <https://www.marinha.mil.br/sites/default/files/atos-normativos/dpc/normam/normam-211.pdf>. Acesso em: 05 nov. 2024.

DIRETORIA DE PORTOS E COSTAS (DPC). NORMAM 212/DPC - Normas da Autoridade Marítima para Motos Aquáticas e Motonautas. Disponível em: <https://www.marinha.mil.br/sites/default/files/atos-normativos/dpc/normam/normam-212.pdf>. Acesso em: 05 nov. 2024.

FOWLER, M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3. ed. Boston: Addison-Wesley Professional, 2014.

FREEMAN, A. Pro Angular 9: Build Powerful and Dynamic Web Apps. Apress, 2022.

GAMMA, E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.

INSOMNIA. Get Started – Insomnia Documentation. Disponível em: <https://docs.insomnia.rest/insomnia/get-started>. Acesso em: 5 nov. 2024.

JOHNSON, R. et al. Spring Boot in Action. 2. ed. Manning Publications, 2022.

LOELIGER, J.; MCCULLOUGH, M. Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development. O'Reilly Media, 2012.

MORAES, Janaina. Técnicas para levantamento de requisitos. DevMedia, 2014. Disponível em: <https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>. Acesso em: 24 set. 2024.

PDF-LIB. "PDF-LIB Documentation." Acesso em 10 de novembro de 2024. Disponível em: <https://pdf-lib.js.org>

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: Uma Abordagem Profissional. 8ª ed. Porto Alegre: AMGH, 2016.

REENSKAUG, T.; COPLIEN, J. O. The DCI Architecture: A New Vision of Object-Oriented Programming. 2009. Disponível em: <http://www.artima.com/articles/dci_vision.html>. Acesso em: 27 set. 2024.

SCHWAB, Klaus. A Quarta Revolução Industrial. São Paulo: Edipro, 2016.

SESHADRI, S.; GREEN, B. Angular Development with TypeScript. Manning Publications, 2021.

SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SOMMERVILLE, I.; SAWYER, P. Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 1997.

SPRING. Spring Framework: Reference Documentation (versão 6.0.0). Disponível em: <https://docs.spring.io/spring-framework/docs/6.0.0/reference/pdf/spring-framework.pdf>. Acesso em: 5 nov. 2024.

TURNQUIST, G. Learning Spring Boot 2.0: Simplify the development of lightning-fast applications based on microservices and reactive programming. 2. ed. Packt Publishing, 2017.

WALLS, C. Spring in Action. 6. ed. Manning Publications, 2022.

WIEGERS, Karl; BEATTY, Joy. Software Requirements. 3. ed. Redmond: Microsoft Press, 2013.

POSTGRESQL. Página Principal da Wiki do PostgreSQL. Disponível em: <https://wiki.postgresql.org/wiki/Main_Page/pt#Funcionalidades>. Acesso em: 5 nov. 2024.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Database System Concepts. McGraw-Hill, 2006.

Anexo A – Dicionário de dados.

|  |  |  |  |
| --- | --- | --- | --- |
| **Coluna** | **Tipo** | **Restrições** | **Descrição** |
| id | bigserial | PK/NOT NULL | Identificador único do cliente |
| cpfcnpj | varchar(255) | UNIQUE | CPF ou CNPJ do cliente |
| rg | varchar(255) | - | Número do RG |
| uf | varchar(255) | - | Unidade Federativa |
| bairro | varchar(255) | - | Bairro do endereço |
| celular | varchar(255) | - | Número de celular |
| cep | varchar(255) | - | CEP do endereço |
| cidade | varchar(255) | - | Cidade do endereço |
| complemento | varchar(255) | - | Complemento do endereço |
| datanasc | date | - | Data de nascimento |
| dtemissao | date | - | Data de emissão do documento |
| email | varchar(255) | - | Endereço de email |
| logradouro | varchar(255) | - | Logradouro do endereço |
| nome | varchar(255) | - | Nome do cliente |
| numero | varchar(255) | - | Número do endereço |
| orgemissor | varchar(255) | - | Órgão emissor do documento |
| senha | varchar(255) | - | Senha do usuário |
| telefone | varchar(255) | - | Número de telefone fixo |
| empresa\_id | bigint | FK | Referência à tabela empresa |
| orgmilitar\_id | bigint | FK | Referência à tabela orgmilitar |
| nacionalidade | varchar(255) | - | Nacionalidade do cliente |
| naturalidade | varchar(255) | - | Naturalidade do cliente |
| representaempresa | varchar(255) | - | Indica se representa uma empresa |
| idempresa | bigint | DEFAULT 0 | ID da empresa representada |

Tabela 11 – Dicionário de dados tabela ‘cliente’  
Fonte: Autoria própria

|  |  |  |  |
| --- | --- | --- | --- |
| **Coluna** | **Tipo** | **Restrições** | **Descrição** |
| id | bigserial | PK/NOT NULL | Identificador único da embarcação |
| cep | varchar(255) | - | CEP da localização da embarcação |
| uf | varchar(255) | - | UF da localização da embarcação |
| areanavegacao | varchar(255) | - | Área de navegação permitida |
| arqueacaobruta | real | - | Arqueação bruta da embarcação |
| arqueacaoliquida | real | - | Arqueação líquida da embarcação |
| bairro | varchar(255) | - | Bairro da localização |
| bocamoldada | real | - | Boca moldada da embarcação |
| calado | varchar(255) | - | Calado da embarcação |
| capacarmazenamento | real | - | Capacidade de armazenamento |
| cidade | varchar(255) | - | Cidade da localização |
| compperpendicular | real | - | Comprimento perpendicular |
| comptotal | real | - | Comprimento total |
| complemento | varchar(255) | - | Complemento do endereço |
| construtor | varchar(255) | - | Nome do construtor |
| contorno | real | - | Medida do contorno |
| corpredominante | varchar(255) | - | Cor predominante da embarcação |
| dtconstrucao | date | - | Data de construção |
| dtinscricao | date | - | Data de inscrição |
| logradouro | varchar(255) | - | Logradouro da localização |
| lotacao | integer | - | Lotação máxima |
| matcasco | varchar(255) | - | Material do casco |
| matsuperestrutura | varchar(255) | - | Material da superestrutura |
| nomeembarcacao | varchar(255) | - | Nome da embarcação |
| numcasco | varchar(255) | - | Número do casco |
| numinscricao | varchar(255) | - | Número de inscrição |
| numero | varchar(255) | - | Número do endereço |
| pontalmoldado | real | - | Pontal moldado |
| portebruto | varchar(255) | - | Porte bruto |
| potenciamotor | real | - | Potência do motor |
| qtdmotores | integer | - | Quantidade de motores |
| qtdtripulantes | integer | - | Quantidade de tripulantes |
| tipoatividade | varchar(255) | - | Tipo de atividade |
| tipoembarcacao | varchar(255) | - | Tipo de embarcação |
| tipopropulsao | varchar(255) | - | Tipo de propulsão |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| cliente\_id | bigint | FK | Referência à tabela cliente |
| orgmilitar\_id | bigint | FK | Referência à tabela orgmilitar |

Tabela 12 – Dicionário de dados tabela ‘embarcacao’  
Fonte: Autoria própria

|  |  |  |  |
| --- | --- | --- | --- |
| **Coluna** | **Tipo** | **Restrições** | **Descrição** |
| id | bigserial | PK/NOT NULL | Identificador único do motor |
| marca | varchar(255) | - | Marca do motor |
| numserie | varchar(255) | - | Número de série |
| potencia | varchar(255) | - | Potência do motor |
| embarcacao\_id | bigint | FK | Referência à tabela embarcacao |
| notafiscal\_id | bigint | FK | Referência à tabela notafiscal |
| embarcacao | bigint | - | ID da embarcação (redundante) |

Tabela 14 – Dicionário de dados tabela ‘motor’  
Fonte: Autoria própria

|  |  |  |  |
| --- | --- | --- | --- |
| **Coluna** | **Tipo** | **Restrições** | **Descrição** |
| id | bigserial | PK/NOT NULL | Identificador único da Org. Militar |
| Sigla | varchar(255) | - | Sigla da Organização Militar |
| uf | varchar(255) | - | Unidade Federativa |
| bairro | varchar(255) | - | Bairro do endereço |
| cep | varchar(255) | - | CEP do endereço |
| cidade | varchar(255) | - | Cidade do endereço |
| complemento | varchar(255) | - | Complemento do endereço |
| email | varchar(255) | - | Endereço de email |
| logradouro | varchar(255) | - | Logradouro do endereço |
| nome | varchar(255) | - | Nome do Organização Militar |
| numero | varchar(255) | - | Número do endereço |
| telefone | varchar(255) | - | Número de telefone fixo |

Tabela 15 – Dicionário de dados tabela ‘OrgMilitar’  
Fonte: Autoria própria

|  |  |  |  |
| --- | --- | --- | --- |
| **Coluna** | **Tipo** | **Restrições** | **Descrição** |
| id | bigserial | PK/NOT NULL | Identificador único do Usuário |
| email | varchar(255) | - | Endereço de email do usuário |
| Senha | varchar(255) | - | Senha do usuário |
| nome | varchar(255) | - | Nome do Usuário cadastrado |

Tabela 16 – Dicionário de dados tabela ‘Usuario’  
Fonte: Autoria própria

# Anexo B – Protótipos de tela

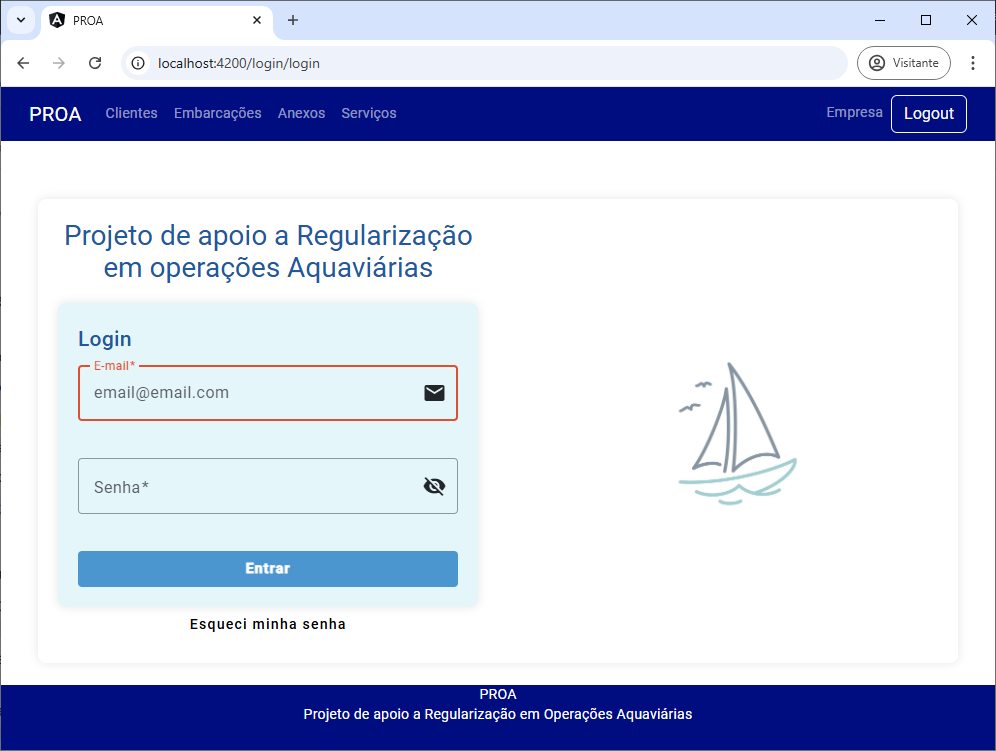


Figura 9 – Protótipo de tela – Login

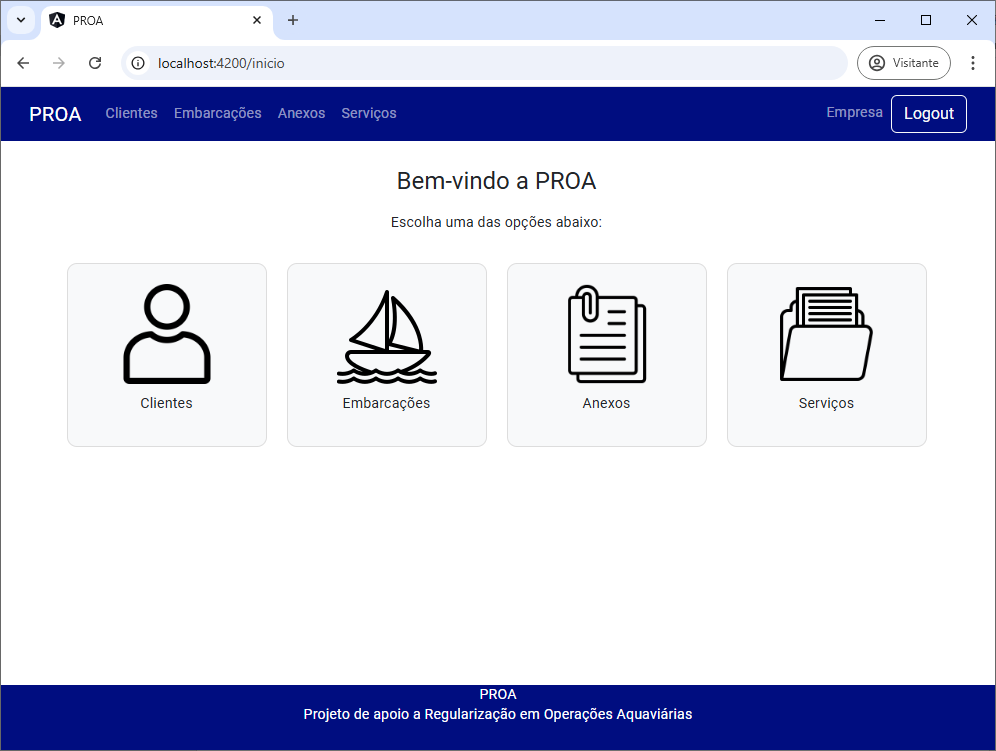


Figura 10 – Protótipo de tela – Menu principal

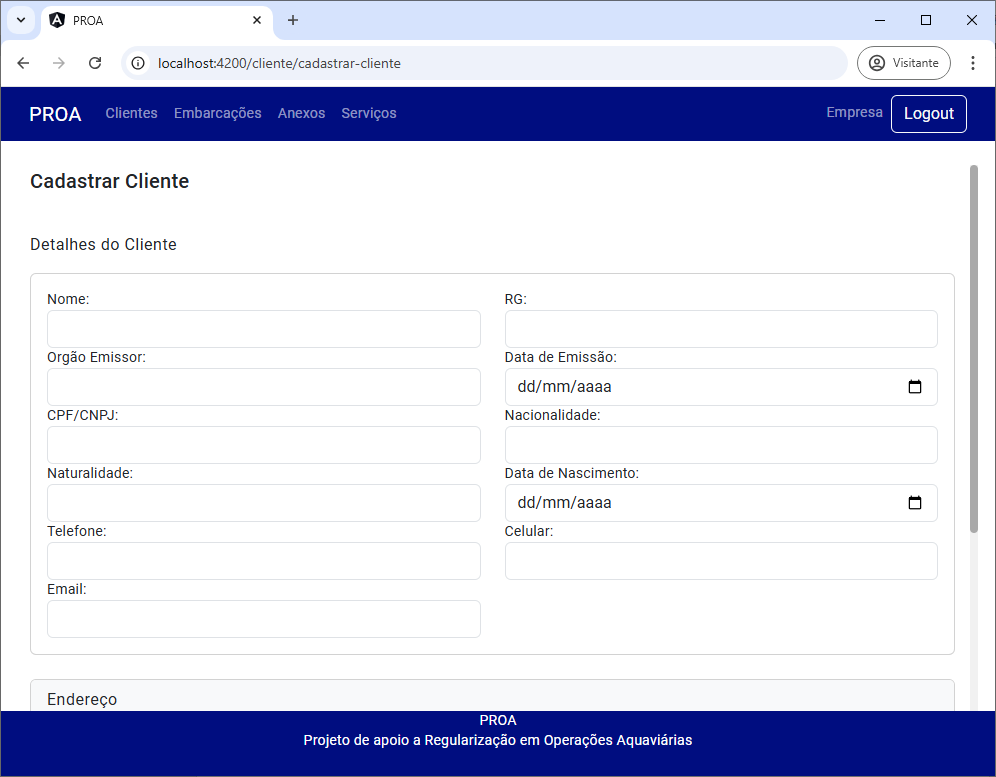


Figura 11 – Protótipo de tela – Cadastro de Cliente

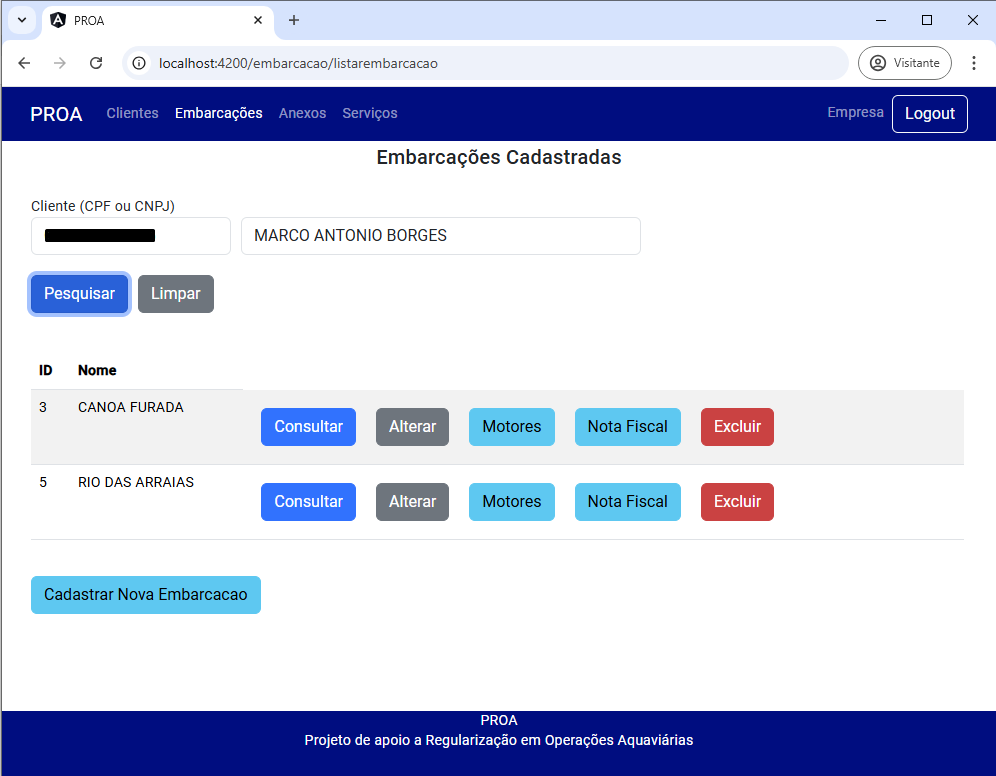


Figura 12 – Protótipo de tela – Listar Embarcações

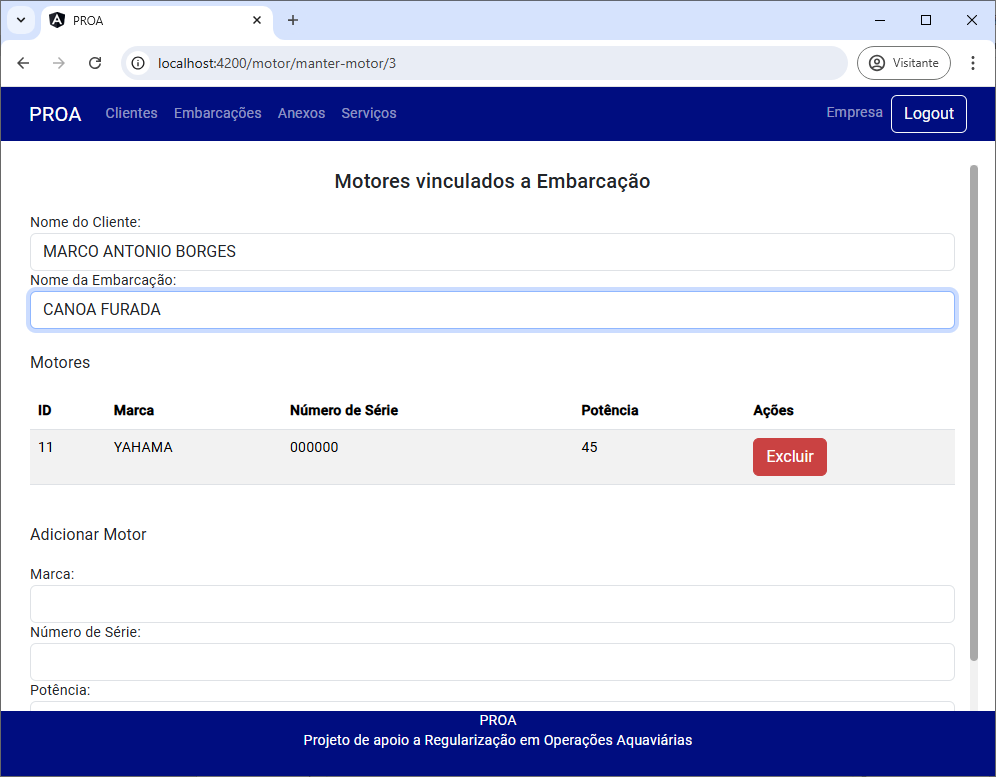


Figura 13 – Protótipo de tela – Motores

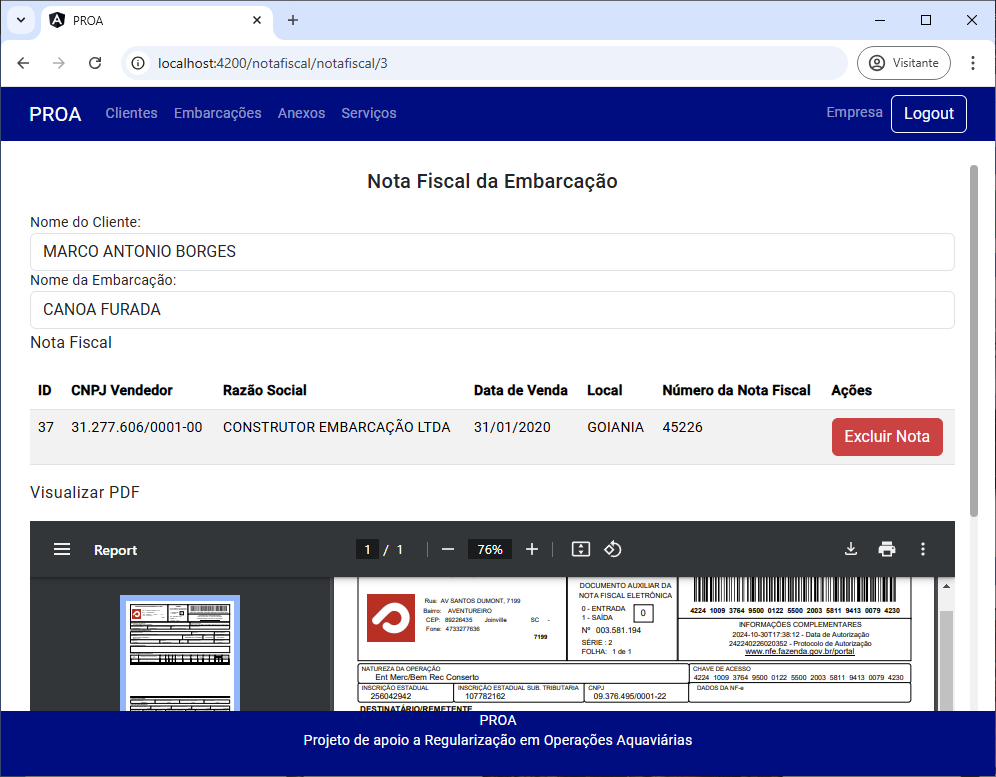


Figura 14 – Protótipo de tela – Nota fiscal

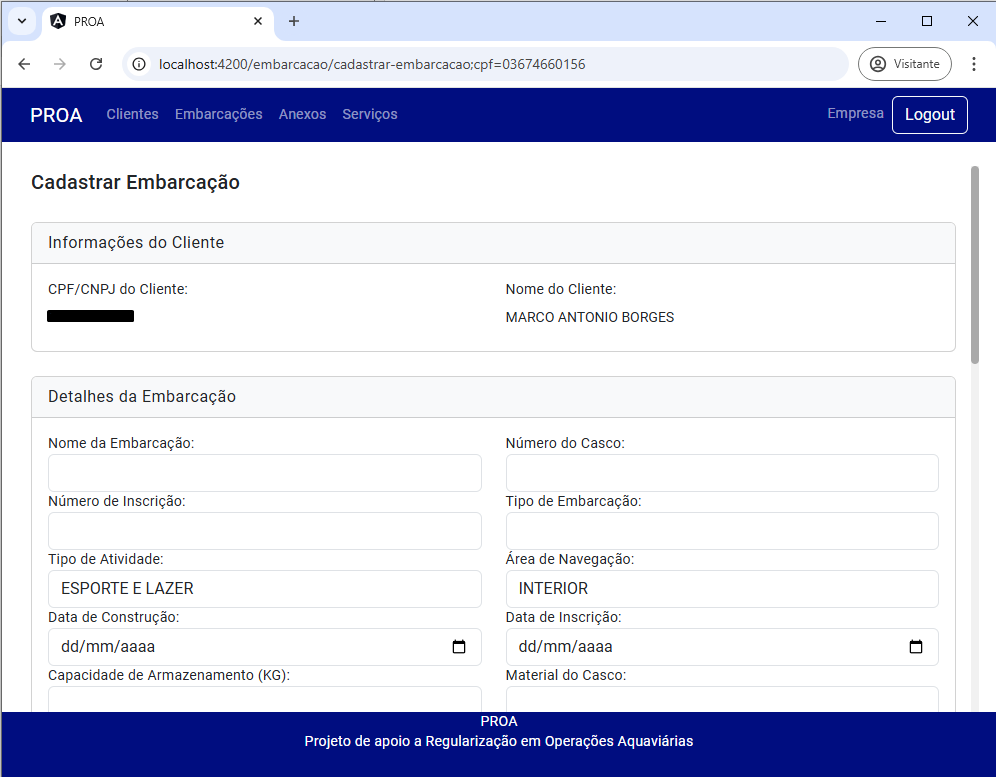


Figura 15 – Protótipo de tela – Cadastro Embarcação

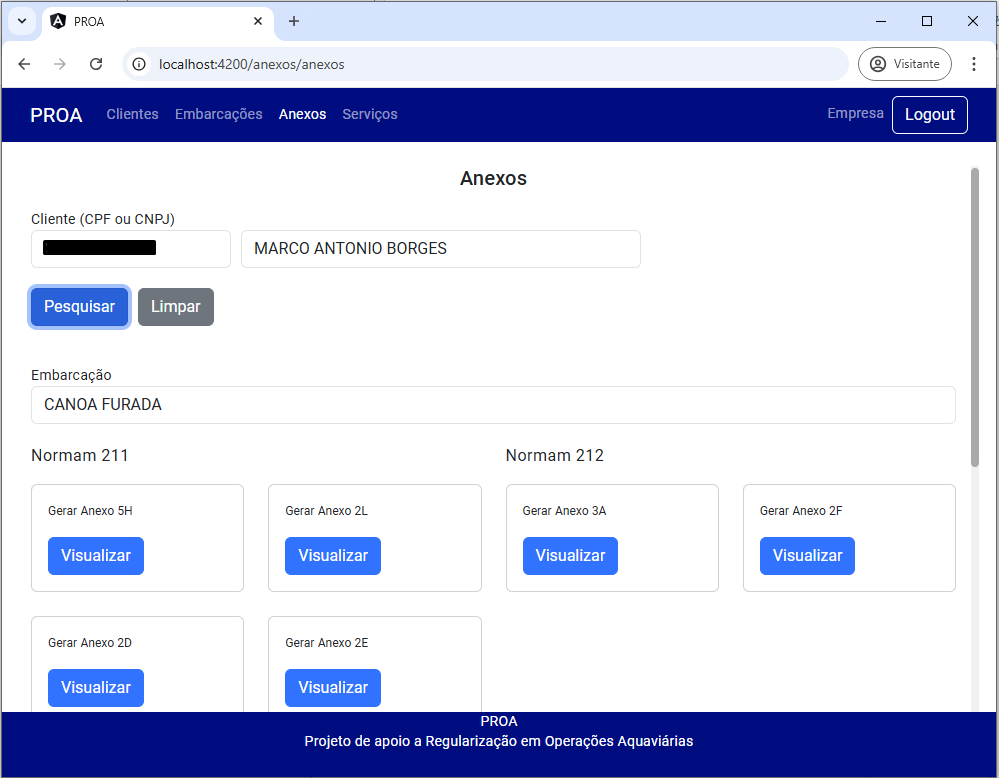


Figura 16 – Protótipo de tela – Emitir anexos

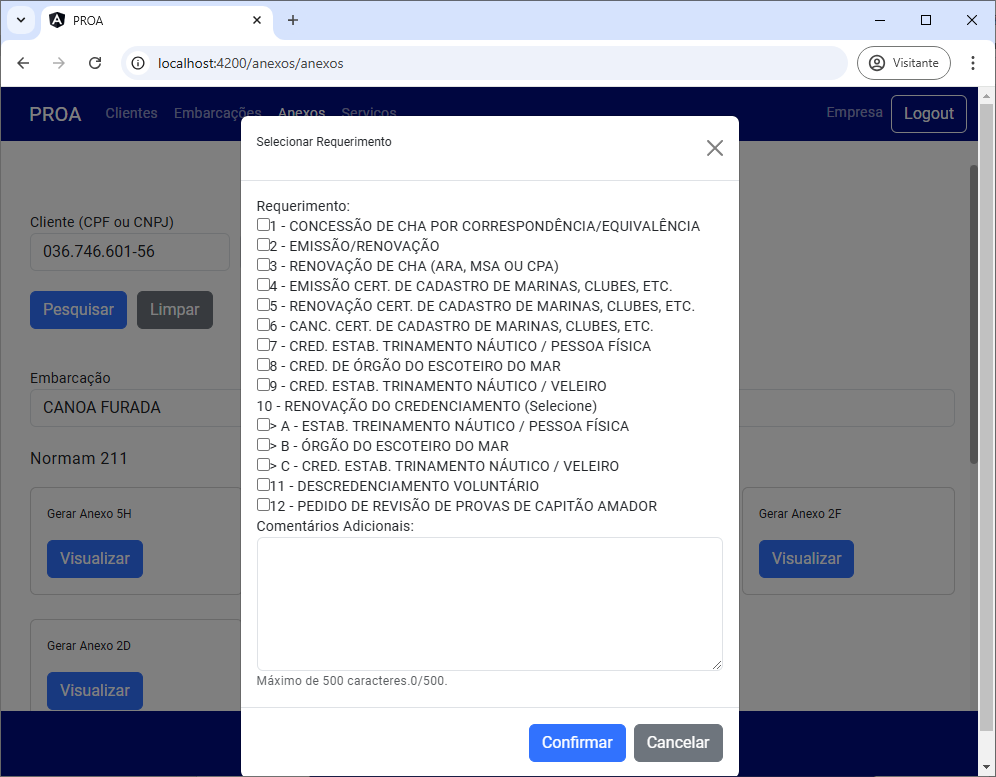


Figura 17 – Protótipo de tela – Modal

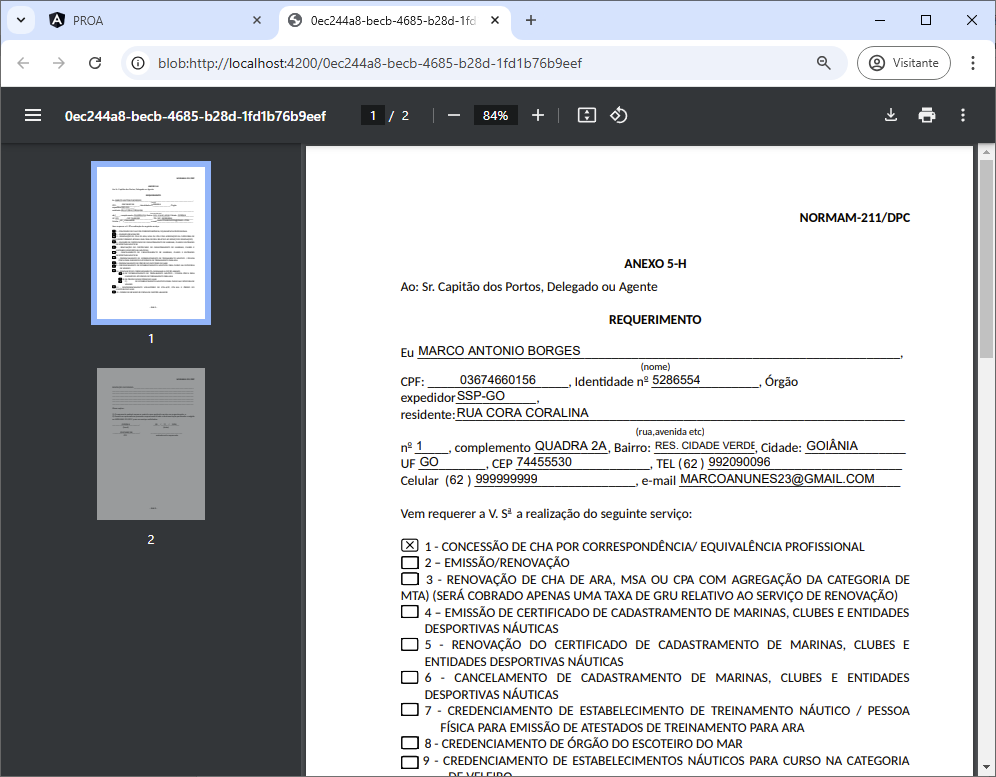


Figura 18 – Protótipo de tela – Anexo Gerado

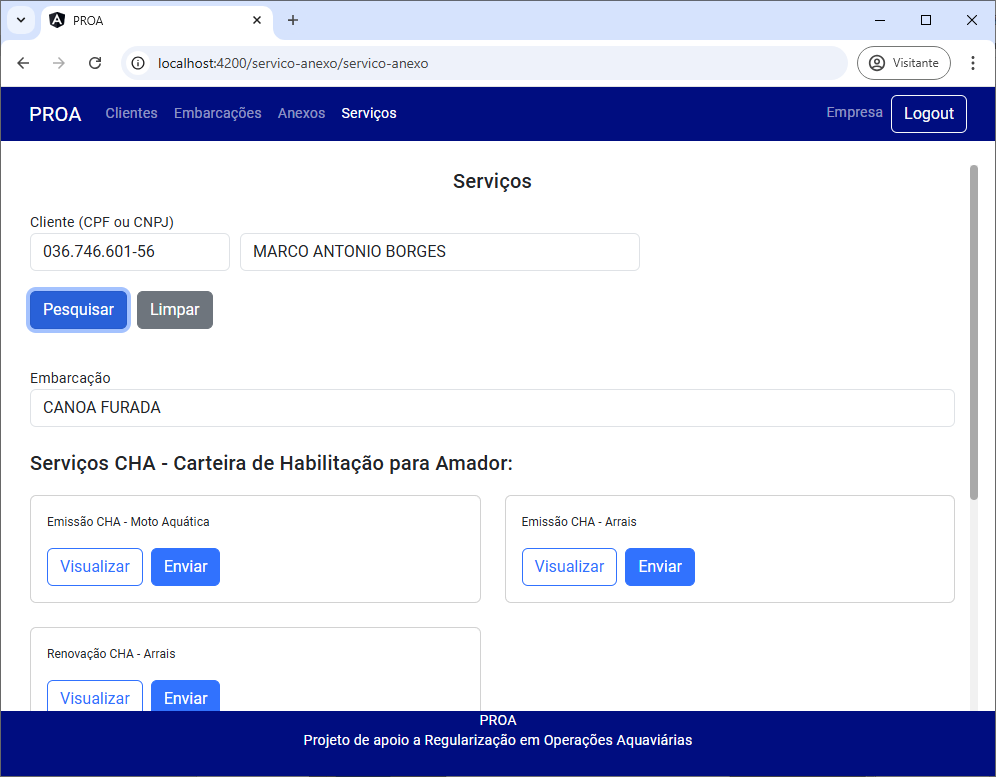


Figura 19 – Protótipo de tela – Emitir anexos por serviço

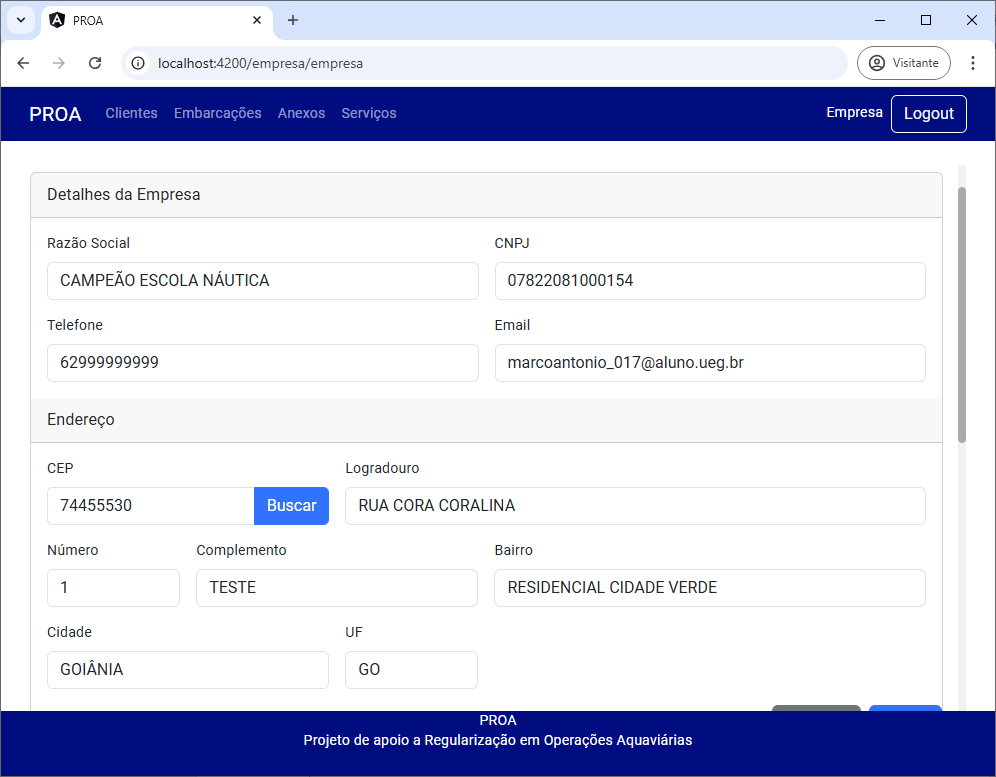


Figura 20 – Protótipo de tela – Cadastro Empresa

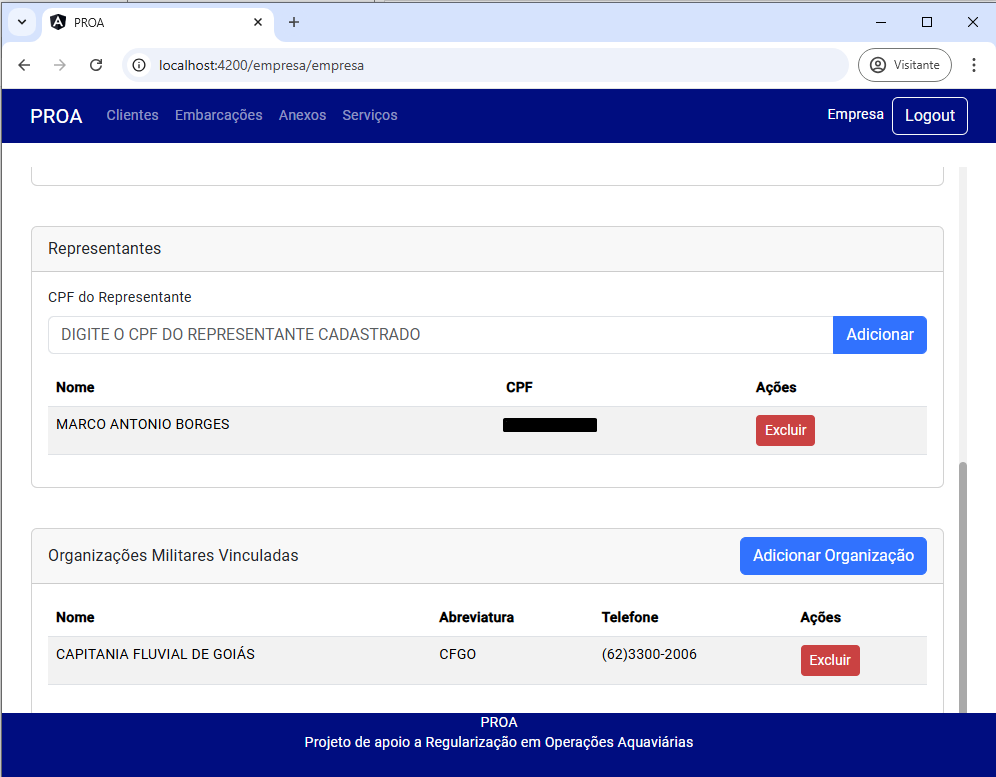


Figura 21 – Protótipo de tela – Cadastro Representantes / Organizações Militares