**Documentação de produto de software**

Versão 1.0

Equipe:

| Matrícula | Nome | Título do produto | Turma |
| --- | --- | --- | --- |
| 2022123TADS0039 | Fabrício Andrade Sousa |  | ADS IV |
| 2022123TADS0080 | Beatriz Bezerra do Nascimento |  | ADS IV |
| 2022123TADS0101 | Antonio Marcos Bezerra Uchôa |  | ADS IV |
|  |  |  | ADS IV |

Pedro II, Outubro de 2023

Sumário

1. Introdução
   1. Tema
   2. Objetivo do projeto
   3. Justificativa do tema escolhido
   4. Método de trabalho
   5. Organização do trabalho
2. Descrição geral do Sistemas
   1. Descrição do problema
   2. Principais envolvidos e suas características
   3. Regras de negócio
3. Requisitos do sistema
   1. Requisitos funcionais
   2. Requisitos não-funcionais
   3. Protótipo e telas
   4. Métricas e cronograma
4. Análise e Design
   1. Arquitetura do sistema
   2. Modelo do domínio
   3. Diagramas (desenho do sistema)
   4. Modelo de dados
   5. Ambiente de desenvolvimento
   6. Sistemas e componentes externos utilizados
5. Implementação
6. Testes
7. Implantação
8. Manual do usuário
9. Conclusões e considerações finais
10. Bibliografia
11. **INTRODUÇÃO**

Initial Steps, um site para auxiliar a comunidade acadêmica de Análise e Desenvolvimento de Sistemas do IFPI campus pedro II. Apresentando um acervo de informações relevantes para os alunos do curso.

* 1. **Tema**

Initial Steps: um site com várias informações sobre o curso de Análise de sistemas e conteúdos sobre as disciplinas estudadas durante a graduação.

* 1. **Objetivo do projeto**

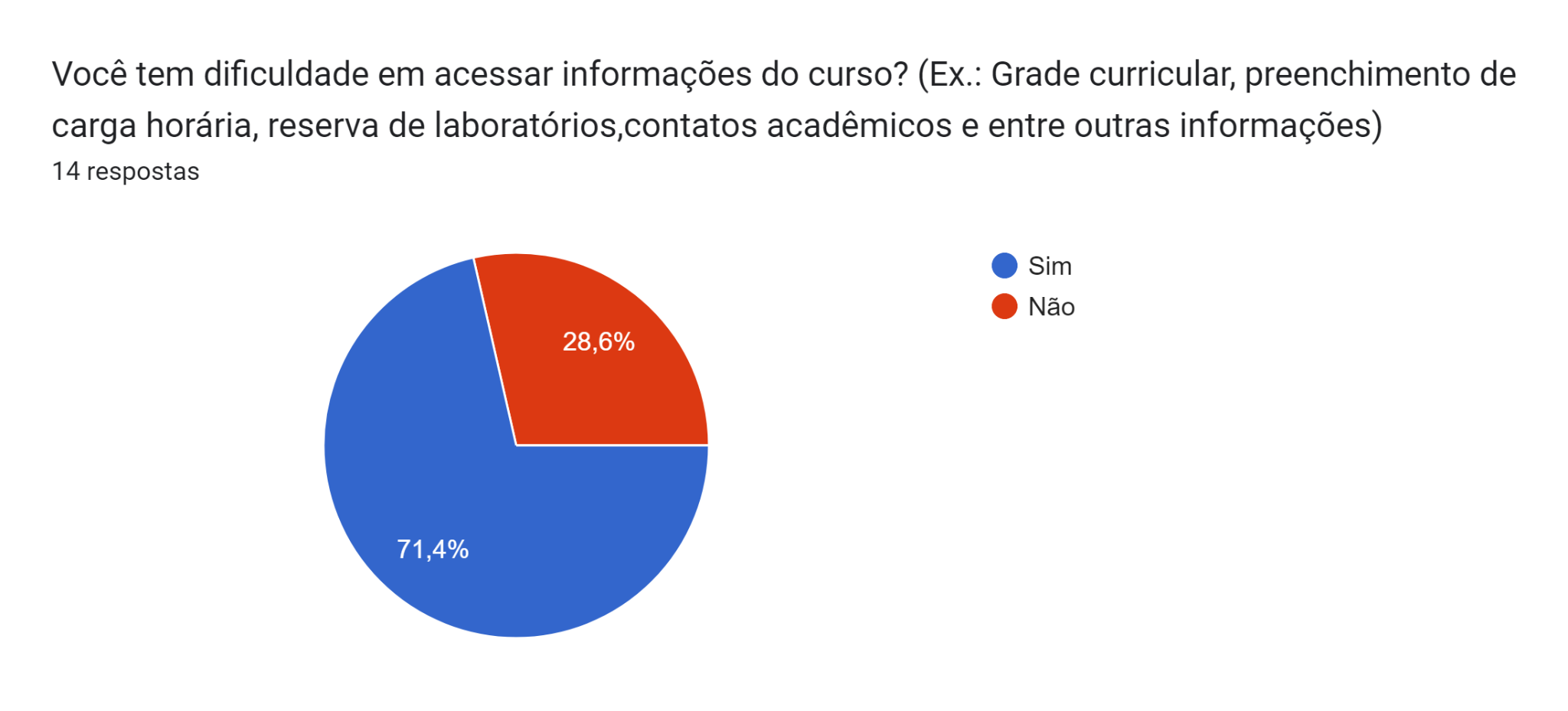
O objetivo geral do projeto é promover uma maior integração entre o campus e os alunos. Promovendo um acesso maior à informações do curso como grade curricular, contatos, projetos realizados e entre outros assuntos.

* 1. **Justificativa do tema escolhido**

Com a decorrência do avanço da tecnologia e a alta demanda de mão de obra qualificada na área de construção de softwares, os centros de ensino no brasil tiveram que se adaptar e oferecer cursos que focam na construção de sistemas que ajudem a suprir tal demanda. Neste sentido, as instituições se adequaram às novas mudanças e, cursos de tecnologia, foram implementados, abrindo várias vagas para estudantes que desejam se formar nessa área. Mas, apesar de tal cenário, estes cursos encontram-se com uma grande taxa de evasão que são provocadas por muitos aspectos e entre eles destaca-se alta dificuldade em absorver o conteúdo, principalmente nas disciplinas que envolvem programação ( Rocha & C.França, 2013), como também na forma como a instituição lida com o problema da adaptabilidade do aluno com o curso (Monteiro & Ianuskiewtz, 2018). Diante destas circunstâncias, o uso da Tecnologia da Informação e Comunicação ( TICs ), que podem ser entendidas, de acordo com Oliveira & Moura (2015) como “um conjunto de recursos tecnológicos integrados entre si, que proporcionam por meio das funções de software e telecomunicações, a automação e comunicação dos processos de negócios, da pesquisa científica e de ensino e aprendizagem”(p.78), tem a expectativa de ser um atenuador nos cenário das instituições, já que a solução pode além de promover a disseminação de materiais relevantes a esses cursos também oferece um constante relação entre a instituição e o discente.

Pensando em tal problemática, propomos para amenizar tal situação, a construção de um site que possibilitaria os alunos a ter um maior entendimento e integralização com o curso de análise e desenvolvimento de sistemas, promovendo, com esse projeto, o impulsionamento do curso nos seguintes fatores: aprimorar a interação entre o setor acadêmico e os alunos do curso, aumentar o rendimento escolar referente às áreas de atuação do curso com o disponibilização de materiais didáticos e impulsionar a comunidade de ADS expondo os trabalhos realizados na área de TI do campus IFPI de pedro II.

Para validar nosso ponto de vista perante a realização do projeto foi feita uma pesquisa no módulo 1 de ADS no campus do IFPI de Pedro II. A pesquisa buscou compreender de modo quantitativo a familiaridade e a adaptabilidade dos alunos com o curso. A pesquisa realizada no dia 19 de abril de 2023 obteve 14 respostas de um total de 40 alunos, e coletou as seguintes respostas:



* 1. **Método de trabalho**
  2. **Organização do trabalho**

Neste item deve-se descrever como o documento estar organizado.

**2. DESCRIÇÃO GERAL DO SISTEMAS**

Este capítulo tem como objetivo descrever de forma geral o sistema, o

escopo e as principais funções. A descrição geral do sistema deve abranger os itens a seguir.

1. **Descrição do problema**

Neste contexto, ações para manutenção destes estudantes ou para o apoio em suas atividades iniciais do curso superior são relevantes. Apesar da presença de um site institucional de caráter geral, o IFPI possui lacunas no tocante ao material disponível aos alunos no referido site. Tendo em vista esse paradigma faz exemplo a realidade do curso de Análise e Desenvolvimento de Sistema do IFPI Campus de Pedro II, que apesar de mostrar uma preocupação com os alunos de sua instituição ainda apresenta a ausência de um sistema que ajude os alunos de ADS a ter uma melhor integração com o curso para que eles se tornem mais engajados e se informem sobre todos os assuntos relevantes ao curso, tornado assim essa comunidade menos propensas a desistência ou evasão do campus e reforçando a política de ensino que pretende melhorar na infraestrutura do ensino e na permanência do aluno no ambiente acadêmico (PDI,2020).

1. **Principais envolvidos e suas características**
2. **Usuários do Sistema**

O Initial Steps é voltado para os interessados em entrar no curso de análise e desenvolvimento de sistemas e também para quem já está no curso.

1. **Desenvolvedores do Sistema**

Os desenvolvedores do sistema são alunos do módulo IV do curso de Análise e Desenvolvimento de Sistemas, no Instituto Federal do Piauí - IFPI.

**3. REQUISITOS DO SISTEMA**

Este capítulo tem como objetivo descrever os requisitos do sistema. No caso de sistemas que possuam usuários/solicitantes reais para o levantamento de requisitos, pode-se utilizar o modelo de documento de entrevista com usuários do RUP de Solicitações dos Principais Envolvidos

1. **Requisitos funcionais**

RF1: O sistema deve conter conteúdos relacionados ao curso.

RF2: O sistema deve conter projetos realizados no campus (projeto integrador).

RF3: O sistema deve conter contatos em geral.

RF4: O sistema deve conter calendário acadêmico.

RF5: O sistema deve conter livros disponíveis.

RF6: O sistema deve conter o conteúdo de cada matéria.

RF7: O sistema deve conter links.

RF8: O sistema deve conter explicação sobre os processos.

RF9: O sistema deve conter login e senha para o administrador.

1. **Requisitos não-funcionais**

RNF1: O sistema deve ser responsivo.

RNF2: O sistema deve ser seguro e protegido contra invasões externas ou acesso não

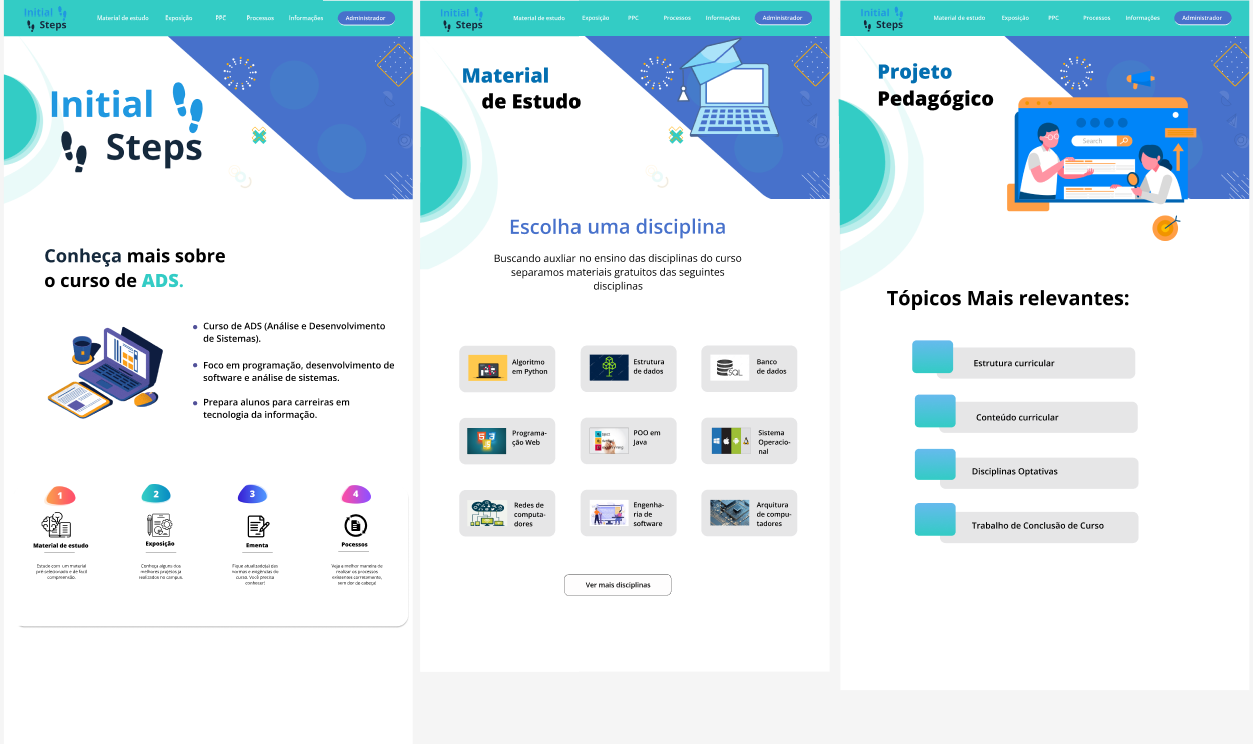
autorizado.

RNF3: O sistema deve ter uma resposta rápida, com tempos de carregamento da página inferior a 2 segundos.

RNF4: O software deve ser fácil de usar, com uma interface amigável e intuitiva.

RNF5: O software deve ser capaz de suportar a carga de trabalho de até 300 usuários simultâneos.

1. **Protótipo**



1. **Métricas e cronograma**

Neste item devem ser estimados os esforços necessários em termos de recursos alocados G e tempo para a obtenção do sistema. Para realizar a estimativa, indica-se o uso de alguma técnica de métrica, como Pontos de Função ou Pontos de Caso de Uso.

Após os cálculos de métricas deve-se elaborar o cronograma detalhado do sistema, que contempla todas as tarefas descritas e os recursos alocados para cada tarefa, com datas para início e término de cada atividade.

A sequência das tarefas e a divisão entre os recursos devem ser realizadas de acordo com o processo de desenvolvimento de software escolhido para o desenvolvimento do sistema, descrito no item 1.5.

**4. ANÁLISE E DESIGN**

Este capítulo tem como objetivo analisar e detalhar a solução do sistema de acordo com os requisitos levantados e avaliados no capítulo 3. Para isso, deve-se ter uma visão geral da arquitetura do sistema e a modelagem da solução do sistema através de diagramas.

1. **Arquitetura do sistema**

Neste item deve ser apresentada a arquitetura de infraestrutura do sistema, demonstrando o tipo de arquitetura que ser utilizada (por exemplo, cliente/servidor de n-camadas), a configuração de hardware, de rede e de software a serem utilizados, bem como o dimensionamento mínimo de

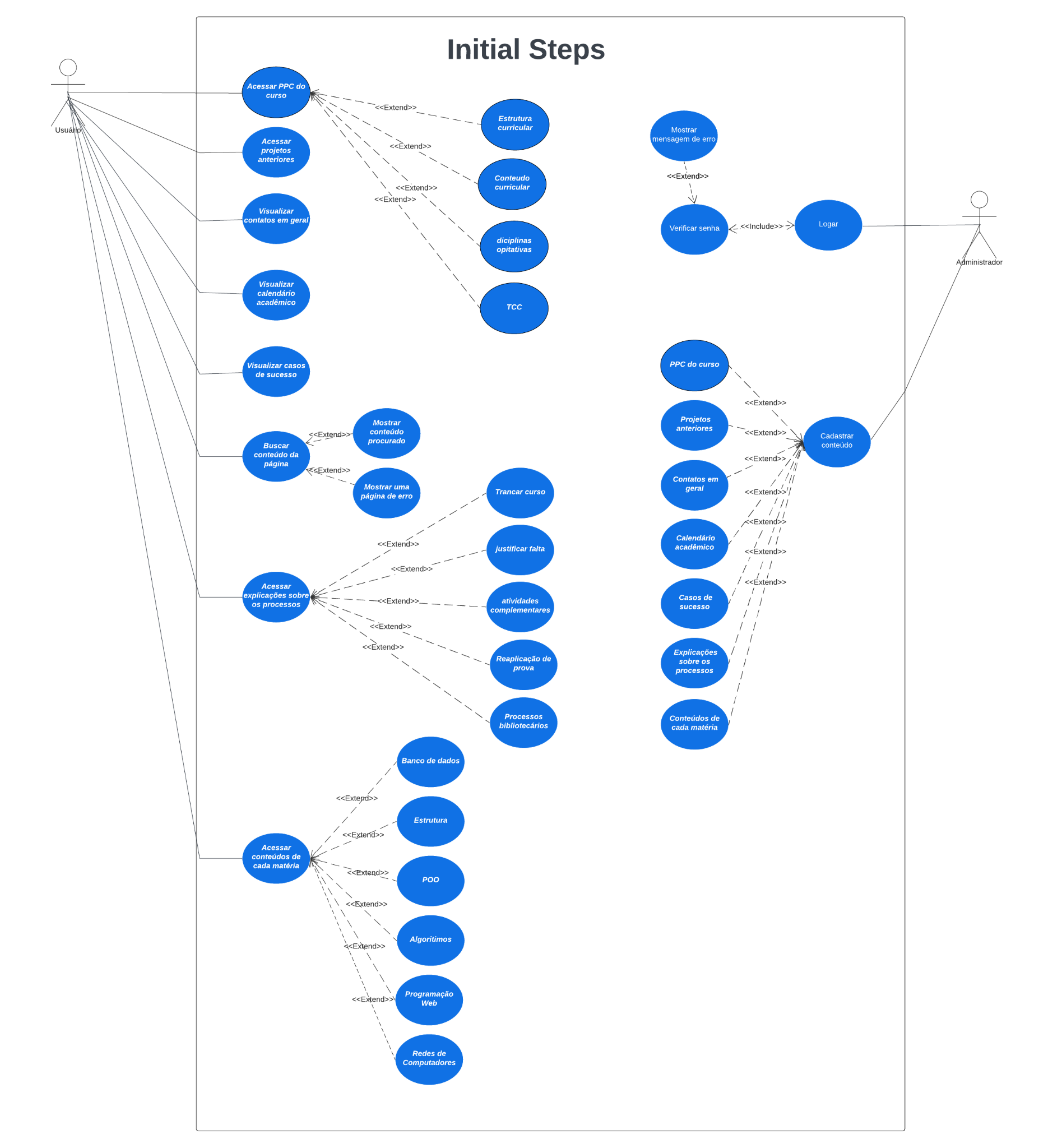
conexões.

1. **Modelo do domínio**

Neste item deve ser apresentado o modelo do domínio, que representa um primeiro modelo conceitual do diagrama de classes. Posteriormente, esse diagrama deve ser validado e complementado para compor o diagrama de classes final.

O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e métodos de cada classe, e os relacionamento entre elas.

1. **Diagramas (desenho do sistema)**

Diagrama de caso de uso:

Use

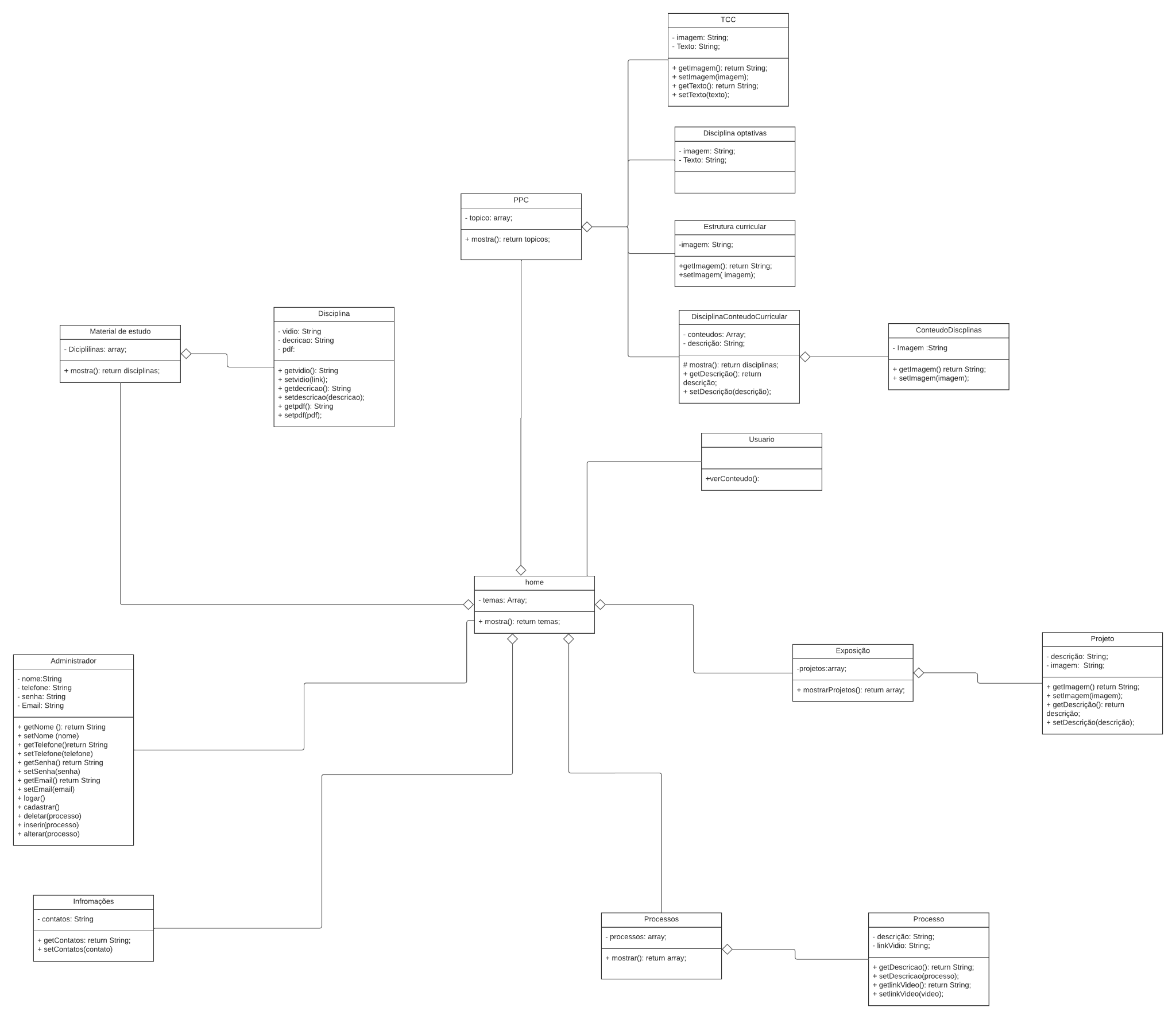
Diagrama de classe:

Diagrama de objetos:

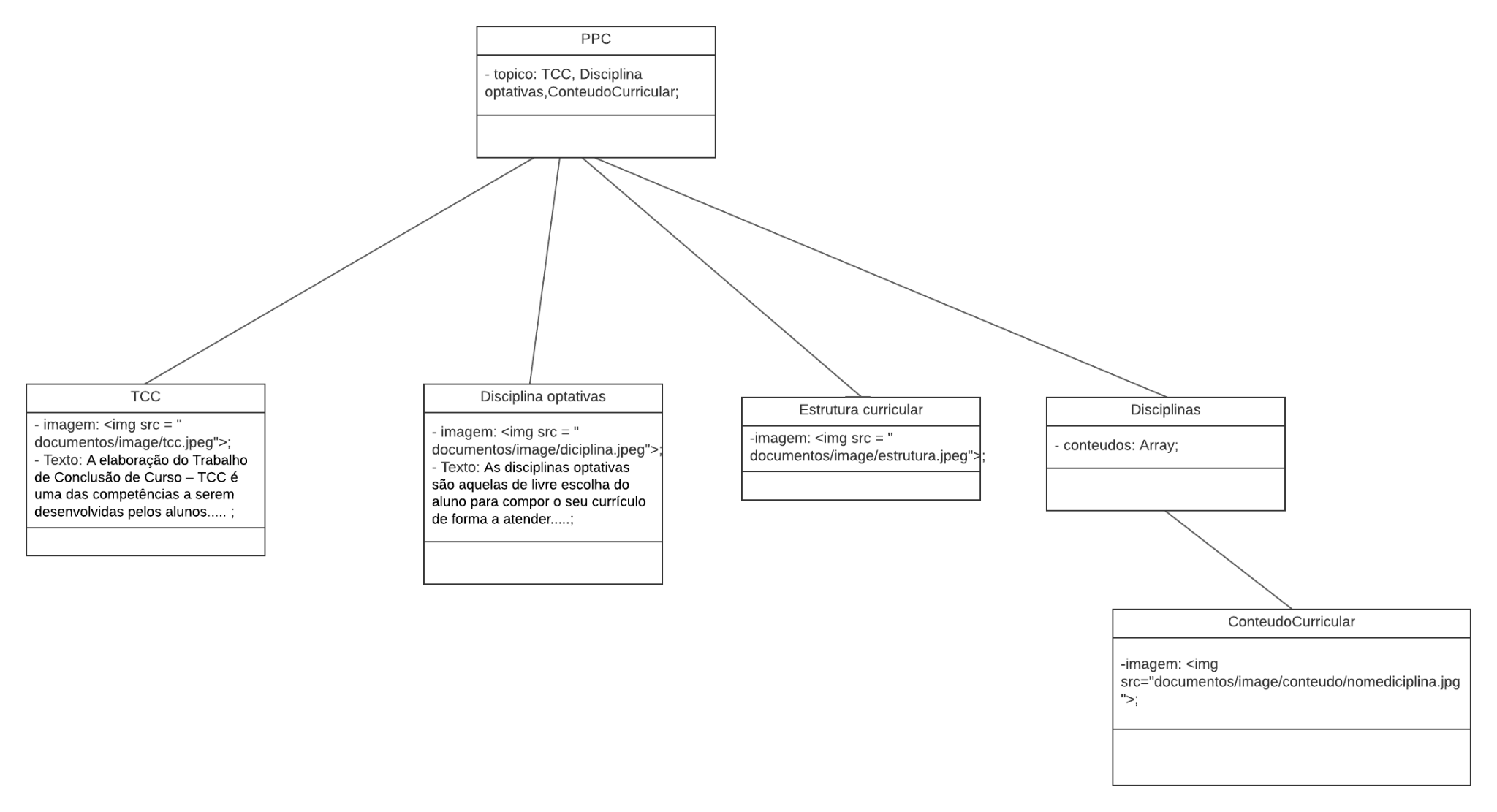


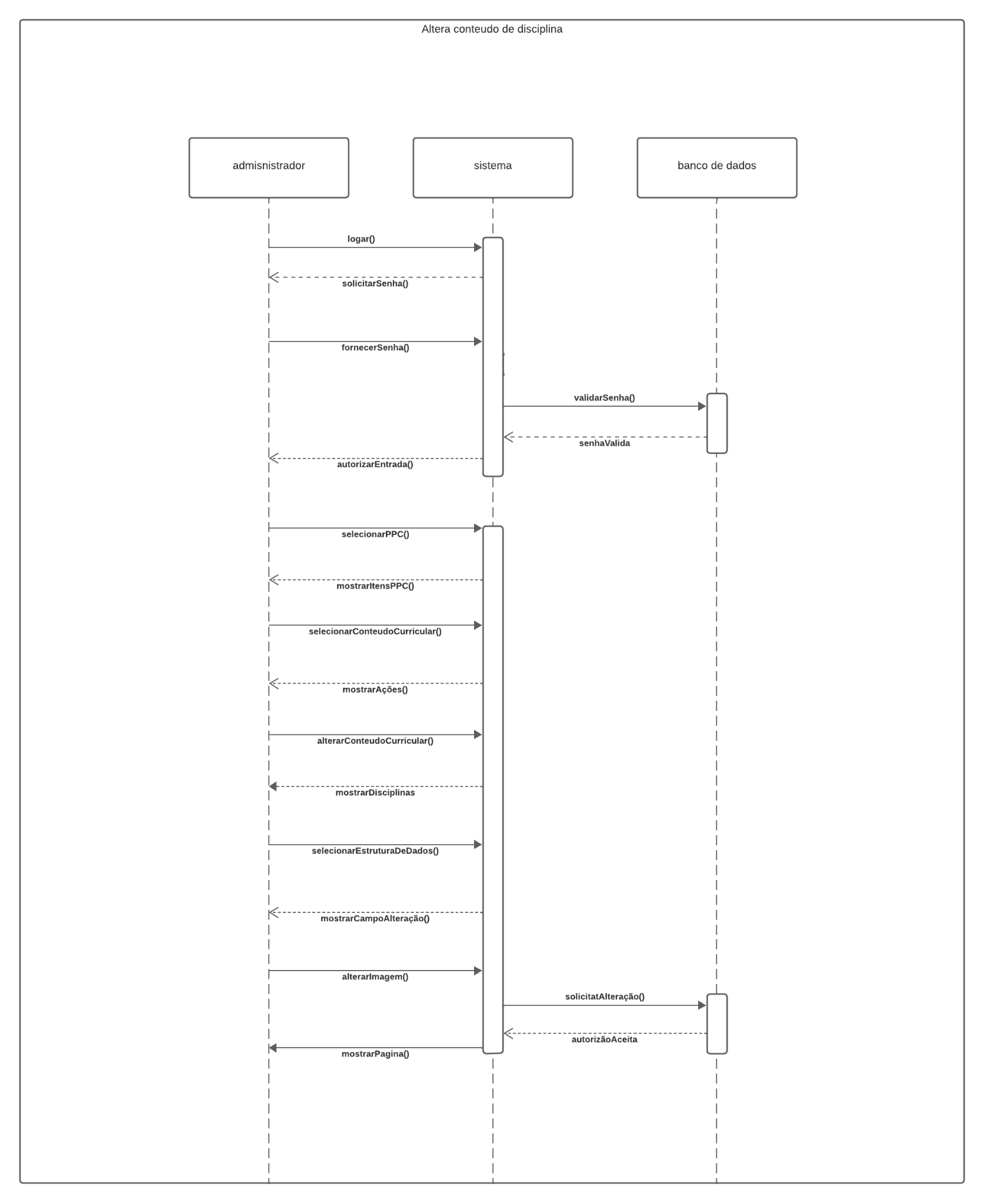
Diagrama de sequência:

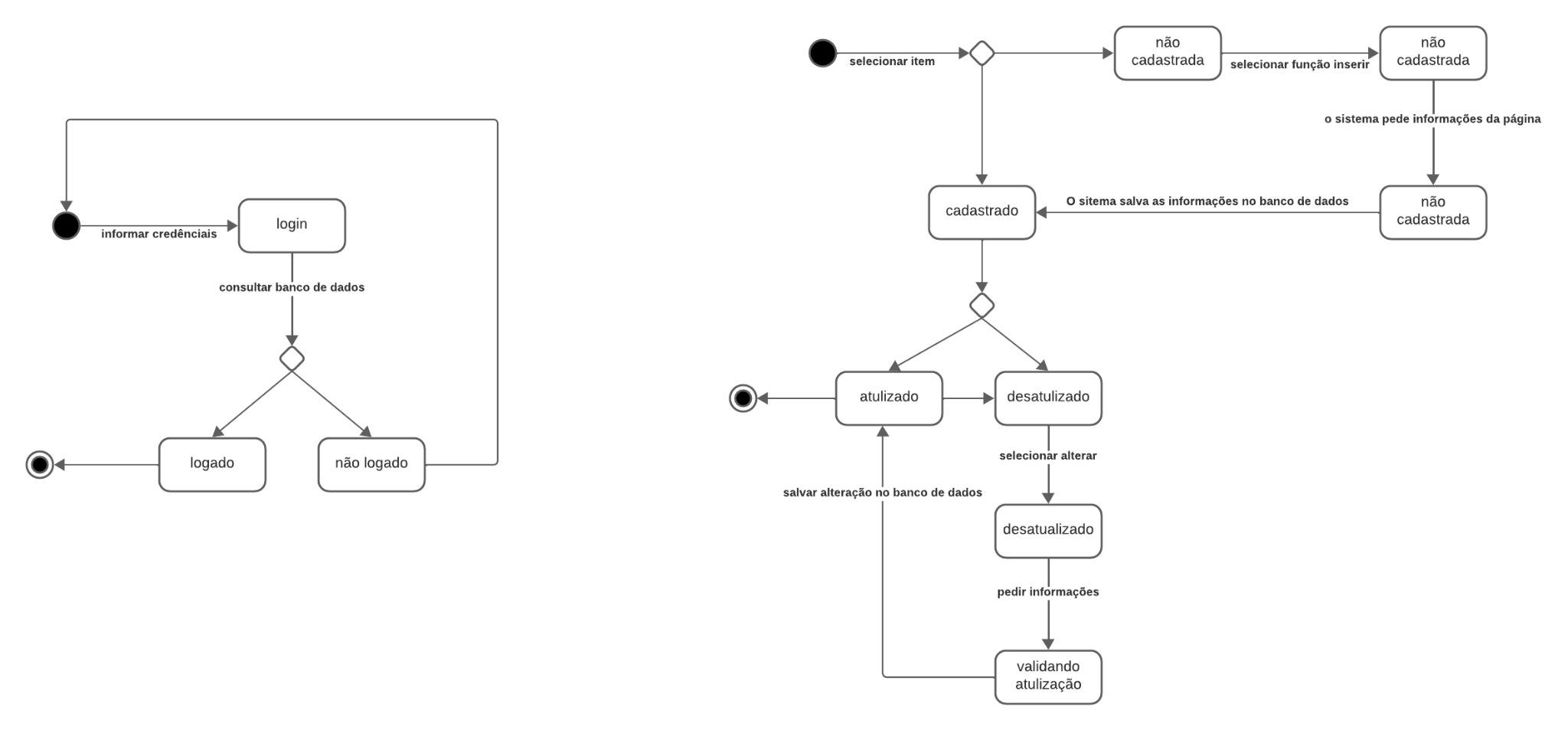
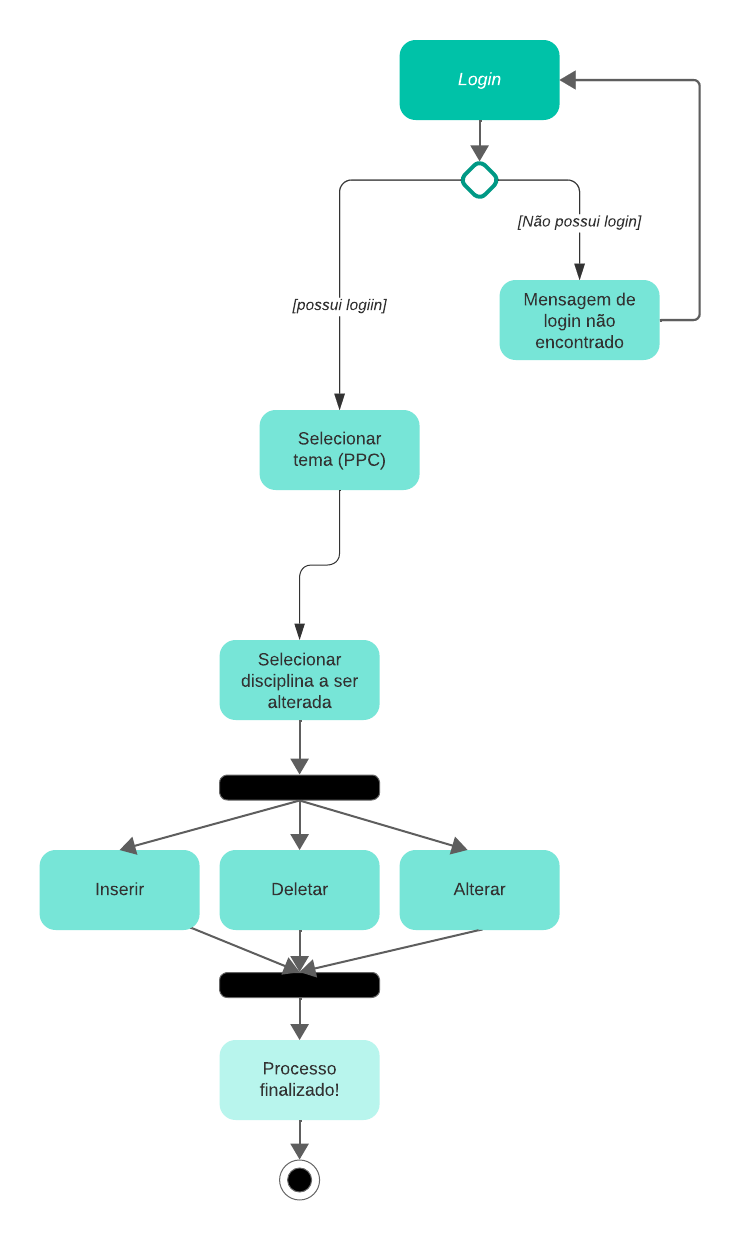
Diagrama de estado:

Diagrama de atividade:



1. **Modelo de dados**

Modelo Lógico da Base de Dados

Neste item deve ser apresentado o modelo lÛgico da base de dados, que pode ser o modelo entidade-relacionamento ou objeto da base de dados. No caso do modelo entidade-relacionamento, o modelo lógico deve passar por todas as regras de normalização.

Como base para geração do modelo lógico pode-se utilizar o diagrama de classes.

Geralmente ferramentas CASE geram automaticamente o modelo lÛgico da base de dados a partir do diagrama de classes.

1. Criação Física do Modelo de Dados

Neste item deve ser realizada a criação física do banco de dados, ou seja, a criação de scripts.

1. Dicionário de Dados

Neste item deve ser criado o dicionário de dados do banco de dados, com o objetivo de documentar todas as tabelas, atributos, stored procedures.

1. **Ambiente de desenvolvimento**

Neste item devem ser apresentados os softwares de desenvolvimento (linguagem de programação, banco de dados, ferramentas, etc.), equipamentos de hardware e redes que sejam essenciais para o

desenvolvimento do sistema.

1. **Sistemas e componentes externos utilizados**

Neste item devem ser descritos os sistemas e componentes externos que serão utilizados no sistema. Por exemplo, sistemas que serão integrados ao sistema desenvolvido, componentes comprados ou livre que estão sendo utilizados para facilitar ou complementar o desenvolvimento do sistema.

**5. IMPLEMENTAÇÃO**

Este capítulo tem como objetivo a implementação das classes em termos de componentes, ou seja, toda a implementação deve ser realizada de acordo com as definiÁıes das fases anteriores e todos os recursos da programação orientada a objetos que a linguagem escolhida oferece.

Geralmente ferramentas CASE geram automaticamente pseudocÛdigos fontes (dependendo da linguagem utilizada) baseados no diagrama de classes.

Algumas boas práticas de programação devem ser seguidas para um maior entendimento do código.

Algumas delas são:

• Cabeçalho de funções contendo campos como descrição, data de criação, autor, etc;

• Comentários no código;

• Padronização de nomes de variáveis, parâmetros, funções, tabelas, stored procedures, etc;

• Verificação de declaração das variáveis;

• Tratamento de erros;

• Criação de stored procedures para acesso aos dados da base;

• Utilização de padrões de projeto;

• Otimização do código, utilizando os melhores algoritmos e funções de recursividade.

**6. TESTES**

Este capítulo tem como objetivo identificar defeitos no sistema, validar as funções do sistema, verificar se os requisitos foram implementados de forma adequada e avaliar a qualidade do software.

**6.1. Plano de Testes**

Neste item deve ser criado o plano de testes do sistema, permitindo a validação do sistema por parte do desenvolvedor, através da verificação dos requisitos do sistema desenvolvido. Inicialmente, identificam-se os requisitos técnicos e funcionais do sistema, e listam-se todas as situações que podem ocorrer com o sistema (essas situaÁıes podem ser elaboradas atravÈs do diagrama de caso de uso e dos diagramas de seq ̧Íncia). Deve-se realizar testes de consistência de campos, funcionalidades, desempenho, etc. O Plano de Testes do Sistema deverá conter, no mínimo.

1. Número do Teste;
2. Descrição do Teste;
3. Resultado esperado.

Por conter todos os testes do sistema, este plano poderá ser um anexo na documentação do sistema.

Alguns tipos de testes a serem realizados são: teste de funcionalidades, teste de usabilidade, teste de desempenho, teste de carga, teste de stress, teste de volume, teste de segurança e controle de acesso, teste de tolerância a falhas e recuperação, teste de configuração, teste de instalação, etc..

**6.2. Execução do Plano de Testes**

Neste item devem ser registrados os testes realizados no sistema tendo como base o Plano de Testes do Sistema.

O registro dos testes deve conter a identificação do sistema, o nome do realizador dos testes e a configuração do ambiente onde foi realizado o teste. AlÈm disso, para cada teste, deve-se ter os seguintes dados:

1. Número do teste;
2. Resultado Obtido; e
3. Comentários (se necessário).

**7. IMPLANTAÇÃO**

Este capítulo tem como objetivo apresentar informações relevantes para a implantação e funcionamento do sistema.

**7.1. Diagrama de Implantação**

Neste item deve ser apresentado o diagrama de implantação que representa a parte física do sistema, exibindo os dispositivos, as máquinas de processamento em tempo de execução e os componentes que nelas serão instalados.

**7.2. Manual de Implantação**

Neste item deve ser elaborado o manual de instalação. Este manual deve conter a descrição passo a passo de como deve ser realizada a instalação do sistema.

**8. MANUAL DO USUÁRIO**

Este capítulo tem como objetivo a elaboração de um manual do usuário. Este manual deve conter a descrição passo a passo de como utilizar o sistema.

**9. CONCLUSÕES E CONSIDERAÇÕES FINAIS**

Este capítulo tem como objetivo apresentar e demonstrar a aplicabilidade dos resultados obtidos, suas limitações, inovações, possíveis integrações com outros projetos e continuação do sistema em trabalhos futuros.

**10. BIBLIOGRAFIA**

BATTLE, Robert; BENSON, Edward. Bridging the semantic Web and Web 2.0 with representational state transfer (REST). Journal of Web Semantics, v. 6, n. 1, p. 61-69, 2008.