

**UNIFBV | Wyden**  
**CENTRO UNIVERSITÁRIO FACULDADE BOA VIAGEM**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**SISTEMA DE CADASTRO DE USUÁRIOS SIMPLIFICADO**

DOCENTE.: FAUSTO GOMINHO  
DISCENTE.: NEY AUGUSTO DA SILVA VICENTE

2023  
RECIFE – PE

# **Sumário**

## **1. DIAGNÓSTICO E TEORIZAÇÃO**

- 1.1. Identificação das partes interessadas e parceiros**
- 1.2. Problemática e/ou problemas identificados**
- 1.3. Justificativa**
- 1.4. Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos)**
- 1.5. Referencial teórico (subsídio teórico para propositura de ações da extensão)**

## **2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO**

- 2.1. Plano de trabalho (usando ferramenta acordada com o docente)**
- 2.2. Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los.**
- 2.3. Grupo de trabalho (descrição da responsabilidade de cada membro)**
- 2.4. Metas, critérios ou indicadores de avaliação do projeto**
- 2.5. Recursos previstos**
- 2.6. Detalhamento técnico do projeto**

## **3. ENCERRAMENTO DO PROJETO**

- 3.1. Relatório Coletivo (podendo ser oral e escrita ou apenas escrita)**
- 3.2. Avaliação de reação da parte interessada**
- 3.3. Relato de Experiência Individual**

### **3.1. CONTEXTUALIZAÇÃO**

### **3.2. METODOLOGIA**

### **3.3. RESULTADOS E DISCUSSÃO:**

### **3.4. REFLEXÃO APROFUNDADA**

### **3.5. CONSIDERAÇÕES FINAIS**

## 1. ELABORAÇÃO DO SISTEMA DE CADASTRO ÚNICO ONCOLÓGICO DO ESTADO DE PERNAMBUCO (SICOPE)

### 1.1 IDENTIFICAÇÃO DAS PARTES INTERESSADAS E PARCEIROS

O câncer está entre as primeiras causas de morte em todo o mundo e sua frequência é crescente, o enfrentamento ao câncer, portanto, requer que se pense em respostas mais eficazes e eficientes para a prevenção, diagnóstico precoce e tratamento adequados.

A integração de diversos programas e serviços de saúde faz-se necessária para promover o cuidado integral com articulação e organização das ações de prevenção, detecção precoce, tratamento e reabilitação.

O Estado de Pernambuco tem fortalecido a regionalização e interiorização da assistência oncológica com vistas a assegurar ao usuário o conjunto de ações e serviços de que necessita com efetividade e eficiência. **O Plano Estadual de Atenção Oncológica** descreve a atenção oncológica no estado de Pernambuco e o planejamento para a expansão da rede de oncologia do estado, em consonância com as portarias ministeriais e resoluções da comissão Inter gestores Bipartite.

O Estado de Pernambuco através da Secretaria estadual de Saúde, suas Secretarias executivas e Secretarias Municipais de Saúde têm empenhado esforços no sentido de estruturar a linha de cuidado na atenção oncológica ampliando a oferta de acesso aos diversos serviços de rastreio, diagnóstico, tratamento, reabilitação e cuidados paliativos em oncologia, promovendo tratamento oportuno e consequente melhoria na qualidade de vida da população Pernambucana. **A Rede Estadual de Atenção Oncológica** ainda é insuficiente para atender plenamente as demandas em oncologia no estado, dada à complexidade dos procedimentos envolvidos no tratamento do câncer em todos os níveis da atenção à saúde.

Através de pesquisas recentes realizadas pela Secretaria Estadual de Saúde de Pernambuco - SES/PE com o intuito de melhorar o atendimento à população Pernambucana, elaborou o **Plano de Atenção para o Diagnóstico e o Tratamento do Câncer no Estado de Pernambuco**.

A elaboração deste plano levou em consideração aspectos como educação, pesquisa, promoção à saúde, prevenção, diagnóstico, tratamento, reabilitação, cuidados paliativos, sistemas de informação e regulação assistencial previstos na legislação vigente, tendo como objetivo principal ofertar serviços qualificados num tempo oportuno.

O objetivo principal é o de ofertar serviços qualificados em tempo oportuno, tornando o estado uma referência na assistência em oncologia. Para tanto o plano envolveu todo o desenho da rede estadual de saúde e seus desdobramentos para a assistência oncológica integral, indo desde o diagnóstico precoce até os cuidados paliativos, passando por todas as fases do tratamento, além do registro de câncer e a pesquisa em saúde.

**O Plano de Atenção para o Diagnóstico e o Tratamento do Câncer no Estado de Pernambuco** configura-se como instrumento de fortalecimento dos serviços municipais ofertando referências regionalizadas para o seguimento de pacientes, fortalecendo os arranjos das redes de saúde regionais, encurtando distâncias para o diagnóstico e tratamento, levando atendimento especializado a todas as regiões do estado, sobretudo as mais necessitadas.

## 1.2 PROBLEMÁTICA E/OU PROBLEMAS IDENTIFICADOS

Após a implantação do **Plano de Atenção para o Diagnóstico e o Tratamento do Câncer no Estado de Pernambuco** pela Secretaria Estadual de Saúde de Pernambuco - SES/PE, foi realizada uma pesquisa com o intuito de averiguar o nível de satisfação com a **Rede Estadual de Atenção Oncológica**.

Com os resultados apurados, ficou observado um grande descontentamento por parte dos entrevistados (pacientes, médicos e funcionários), alguns dos motivos foram:

- Demora para Realização do Cadastramento (Sistema Obsoleto e de difícil manuseio)
- Demora para a Realização das Consultas
- Demora para a Realização dos Exames
- Demora nas Entregas dos Resultados dos Exames
- Médicos com dificuldade para acessar o prontuário dos Pacientes
- Demora na Troca de Informações dentre as GERES (Gerência Regional de Saúde)

Visto os problemas encontrados, a Secretaria Estadual de Saúde de Pernambuco – SES/PE realizou uma licitação para elaboração de um Sistema de Cadastro ao qual agilizasse/melhorasse tanto o acesso da População quanto a troca de informações dentre as GERES (Gerência Regional de Saúde), tentando assim, resolver os problemas elencados acima através da pesquisa.

## 1.3 JUSTIFICATIVA

Com o término da licitação, fomos escolhidos como responsáveis para realizar a melhoria no serviço de acesso a **Rede Estadual de Atenção Oncológica**, onde **deveria ser criado um serviço que fosse acessível e de fácil manuseio para os usuários, que os médicos pudessem acessar o prontuário dos pacientes em tempo Real, assim como os funcionários pudessem inserir ou alterar dados dos usuários.**

Com as informações repassadas, foi iniciado o processo de criação do **SICOPE** (Sistema de Cadastro Único Oncológico do Estado de Pernambuco).

#### **1.4 OBJETIVOS/RESULTADOS/EFEITOS A SEREM ALCANÇADOS (EM RELAÇÃO AO PROBLEMA IDENTIFICADO E SOB A PERSPECTIVA DOS PÚBLICOS ENVOLVIDOS)**

Através da Demanda recebida, irá ser criado um Sistema de Cadastro que seja simples, operacional, resiliente e tolerante a falhas na Linguagem de **Programação JAVA**, por ser uma linguagem segura, de alto desempenho, **orientada a objetos** (é um paradigma de programação onde tudo é representado como um objeto, que passam mensagens uns para os outros decidindo o que fazer com uma mensagem recebida, focando nos estados e comportamentos de cada objeto).

O SICOPE será elaborado para que todxs (mesmo aqueles sem muito contato com tecnologia) consigam utilizá-lo, com uma simples interface, mas com todos os recursos que a Linguagem **JAVA** oferece.

- Criar um Cadastro de Simples Acesso;
- Que tenha como características Robustez e Segurança
- Utilizar todo o Conhecimento Acadêmico em Orientação á Objetos em **Java** (Classe, Atributo, Objeto, Métodos, Herança, Polimorfismo, Tratamento de Exceções etc.)

## 1.4 REFERENCIAL TEÓRICO

Durante a elaboração do Projeto, foram acessadas fontes teóricas que serão descritas a seguir.:

Segundo Chagas, Clayton Escouper das. **Java Básico e Orientação a Objeto: volume único**. Rio de Janeiro: Fundação CECIERJ, 2010. “ **Java oferece várias camadas de controle de segurança para a proteção contra código malicioso, além de possuir um compilador altamente rigoroso que evita erros básicos de programação, ajudando a manter a integridade do software, um dos conceitos essenciais em Governança. Além dessas características, há um diferencial que deve ser considerado: Java é uma solução gratuita.**”

[Chagas, Clayton Escouper das. Java Básico e Orientação a Objeto: volume único / Clayton Escouper das Chagas, Cássia Blondet Baruque, Lúcia Blondet Baruque. – Rio de Janeiro: Fundação CECIERJ, 2010. 238p.; 19 x 26,5 cm. ISBN: 978-85-7648-648-0 1. Linguagem de programação. 2. Java. 3. Programação gráfica. I. Baruque, Cássia Blondet. II. Baruque, Lúcia Blondet. III. Título.](#)

Segundo Claro D. B. e Sobral J. B. M. **PROGRAMAÇÃO EM JAVA**. Florianópolis, SC: Copley Pearson Education. “**A orientação a objetos tem como objetivo principal modelar o mundo real, e garantir que as taxas de manutenibilidade (manutenção) serão maiores diante deste contexto. Isso é possível, pois utilizando uma linguagem de programação orientada a objetos consegue-se obter um desenvolvimento mais rápido, visto que este desenvolvimento ocorre em módulos, em blocos de códigos correspondentes aos objetos e seus acoplamentos. Através da orientação a objetos pode-se obter uma maior qualidade e agilidade no desenvolvimento, pois o fator reutilização permite que se reutilize outros objetos que foram anteriormente desenvolvidos e podem ser facilmente incorporados na aplicação. A reutilização também garante uma maneabilidade melhor do programa, pois os testes referentes aos componentes, já foram previamente executados, garantindo assim a utilização coesa dos objetos.**”

[Claro D. B. e Sobral J. B. M. PROGRAMAÇÃO EM JAVA. Copley Pearson Education. Florianópolis, SC.](#)

Segundo Goldman, Alfredo; Silva, Paulo J. S.; Camargo, Raphael Y. de (eds. e revs.) no livro **Linguagem de Programação 2. Programação orientada a objeto 3. Java**. São Paulo: Universidade de São Paulo, Instituto de Matemática e Estatística. Departamento de Ciência da Computação. “**Uma das principais abordagens utilizadas é a reutilização de código. Se conseguirmos escrever código que é reutilizado em diversos programas diferentes, teremos economizado tempo pois não será necessário reescrever este código para cada novo programa. A biblioteca de classes de Java é um exemplo de classes que foram escritas pelos programadores da Sun e que são reutilizadas diariamente por milhares de programadores em todo o mundo. Porém, algumas vezes, o programador necessita fazer alguma modificação em uma classe existente pois ela não atende exatamente às necessidades de um dado programa. Em alguns casos deseja-se modificar a implementação de um de seus métodos, em outros casos deseja-se acrescentar alguma funcionalidade extra à classe, com a adição de novos métodos e atributos. Esta reutilização de código com extensões e modificações, pode ser obtida em linguagens orientadas a objetos através do conceito de Herança**”

[Linguagem de Programação 2. Programação orientada a objeto 3. Java I. Goldman, Alfredo II. Silva, Paulo J. S. III. Camargo, Raphael Y. de ed. e rev. IV. Universidade de São Paulo, Instituto de Matemática e Estatística. Departamento de Ciência da Computação.](#)

## 2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO

Utilizando o conhecimento acadêmico obtido na cadeira de **Programação Orientada a Objeto em JAVA**, foi construído um Sistema de Cadastro de pacientes, capaz de ser utilizado por qualquer usuário mesmo que não tenha muita afinidade com tecnologia.

A escolha da linguagem **JAVA** deve-se ao fato de possuir características importantes como exemplo.:

- Oferecer várias camadas de controle de segurança para a proteção contra código malicioso;
- Possuir um compilador altamente rigoroso que evita erros básicos de programação, ajudando a manter a integridade do software, um dos conceitos essenciais em Governança;
- Simplicidade.: por se parece com a Linguagem C++, é simples de aprender, permitindo a você produtividade desde o início e possui um pequeno número de construtores.
- Processamento Distribuído: através de classes que suportam a distribuição, dando suporte a aplicações em rede, ademais, Java suporta vários níveis de conectividade através das classes contidas nos pacotes Java.net
- Portabilidade.: Java pode rodar em qualquer plataforma o que permite sua alta portabilidade. Isto se deve ao fato de que seu compilador não gera instruções específicas a uma plataforma, mas sim um programa em um código intermediário. Tal programa, denominado bytecode, pode ser caracterizado como uma linguagem de máquina destinada a um processador virtual que não existe fisicamente.
- Multithreading.: Java fornece suporte a múltiplos threads de execução, que podem manipular diferentes tarefas.
- Coletor de Lixo.: Sua função é a de varrer a memória de tempos em tempos, liberando automaticamente os blocos que não estão sendo utilizados, evitando, assim, erros na alocação de memória.
- Ser uma solução gratuita;

## **2.1 PLANO DE TRABALHO (USANDO FERRAMENTA ACORDADA COM O DOCENTE)**

Como se tratou de um Projeto elaborado por um só desenvolver, ele foi responsável pelos pontos elencados abaixo.:

### **1. Entendimento do Problema:**

- Compreender completamente o problema que o software propõe um resolvedor.
- Realizar uma análise de requisitos detalhados para identificar as necessidades dos usuários.

### **2. Definição de Objetivos:**

- Estabelecer claramente os objetivos do projeto, incluindo funcionalidades principais, metas de desempenho e critérios de sucesso.

### **3. Levantamento de Requisitos:**

- Identificar e documentar todos os requisitos do sistema, incluindo requisitos funcionais e não funcionais.
- Envolver as partes interessadas (interessadas) para garantir uma compreensão precisa das expectativas.

### **4. Análise e Design:**

- Realizar uma análise aprofundada dos requisitos para criar uma arquitetura eficiente do sistema.
- Desenvolver um design detalhado, arquitetura de software e os componentes do sistema.

### **5. Desenvolvimento:**

- Implementar o código de acordo com as especificações do design.
- Utilize boas práticas de conformidade, padrões de projeto e garanta a modularidade e a manutenibilidade do código.

### **6. Testes:**

- Realizar testes unitários, de integração e de sistema para identificar e corrigir bugs.
- Implementar testes de acessibilidade do usuário para garantir que o software atenda aos requisitos.

### **7. Implantação:**

- Preparar o ambiente de produção para implementação do software.
- Realizar a implantação em produção, garantindo um processo suave e impacto mínimo aos usuários.

### **8. Documentação:**

- Criar documentação abrangente, incluindo manuais de usuário, manuais técnicos e qualquer documentação relevante para os desenvolvedores e usuários.

### **9. Treinamento:**

- Fornecer treinamento para os usuários finais e para a equipe de suporte, se necessário.

### **10. Manutenção e Atualizações:**

- Monitorar o desempenho do software em produção.
- Corrigir bugs e implementar atualizações conforme necessário.

### **11. Feedback e Melhoria Contínua:**

- Coletar feedback dos usuários para melhorar continuamente o software.
- Avaliar o desempenho do projeto e identificar áreas de melhoria para futuros projetos.



## 2.2 DESCRIÇÃO DA FORMA DE ENVOLVIMENTO DO PÚBLICO PARTICIPANTE NA FORMULAÇÃO DO PROJETO, SEU DESENVOLVIMENTO E AVALIAÇÃO, BEM COMO AS ESTRATÉGIAS PELO GRUPO PARA MOBILIZÁ-LOS.

Como passo importante para criação do SICOPE, foram realizadas pesquisas sobre como a população espera um sistema rápido e eficiente, para que assim pudéssemos atender a Demanda esperada.

Para a elaboração do Projeto, foram utilizados todos os conceitos aprendidos na academia e através de consultas em livros (já descritos no referencial teórico), que possuem extrema importância para que seja executado com Qualidade.

• **ORIENTAÇÃO A OBJETO:** Em Java, os elementos de um programa são objetos, exceto os tipos primitivos. O código é organizado em classes, que podem estabelecer relacionamentos de herança simples entre si.

- **Classe.:** é a definição de um conjunto de objetos
- **Modificadores de Acesso.:** implementam restrições quanto ao acesso aos elementos da classe (public, private, protected e default)
- **Classe Abstrata.:** tipo de classe especial que não pode ser instanciada, apenas herdada.
- **Objetos.:** são as instâncias das minhas classes, os seres que existem no meu mundo real quando são inseridos no mundo computacional.
- **Atributos.:** são as características que os objetos do mundo real possuem, e estes são sua representação no mundo computacional.
- **Métodos.:** são os comportamentos ou as ações executadas por esses objetos
- **Sobrecarga de Métodos.:** são métodos com o mesmo nome, mas com comportamentos diferentes de acordo com os argumentos passados.
- **Encapsulamento.:** serve para controlar o acesso aos atributos e métodos de uma classe.
- **Pacotes.:** responsável por organizar as classes e recursos de um projeto.
- **Herança.:** técnica da Orientação a Objeto cujo objetivo é criar um modelo com objetos do mundo real, sendo definida pela palavra reservada **extends** (utilizada pela classe que quer herdar as características onde recebe o nome de **subclasse**, classe que está sendo utilizada na herança recebe o nome de **superclasse**).
- **Polimorfismo.:** se refere à capacidade de um objeto assumir diferentes formas, ou seja, uma mesma operação pode ser realizada por diferentes objetos de classes diferentes.
- **Exceção.:** é um evento não esperado que ocorre no sistema quando está em tempo de execução (Runtime), geralmente quando o sistema captura alguma exceção o fluxo do código fica interrompido.

## 2.3 ATUAÇÃO NA CRIAÇÃO DO SICOPE

Como se tratou de um Projeto elaborado por um só desenvolver, ele foi responsável pelos pontos elencados abaixo.:

### 12. Entendimento do Problema:

- Compreender completamente o problema que o software propõe um resolvedor.
- Realizar uma análise de requisitos detalhados para identificar as necessidades dos usuários.

### 13. Definição de Objetivos:

- Estabelecer claramente os objetivos do projeto, incluindo funcionalidades principais, metas de desempenho e critérios de sucesso.

### 14. Levantamento de Requisitos:

- Identificar e documentar todos os requisitos do sistema, incluindo requisitos funcionais e não funcionais.
- Envolver as partes interessadas (interessadas) para garantir uma compreensão precisa das expectativas.

### 15. Análise e Design:

- Realizar uma análise aprofundada dos requisitos para criar uma arquitetura eficiente do sistema.
- Desenvolver um design detalhado, arquitetura de software e os componentes do sistema.

### 16. Desenvolvimento:

- Implementar o código de acordo com as especificações do design.
- Utilize boas práticas de conformidade, padrões de projeto e garanta a modularidade e a manutenibilidade do código.

### 17. Testes:

- Realizar testes unitários, de integração e de sistema para identificar e corrigir bugs.
- Implementar testes de acessibilidade do usuário para garantir que o software atenda aos requisitos.

### 18. Implantação:

- Preparar o ambiente de produção para implementação do software.
- Realizar a implantação em produção, garantindo um processo suave e impacto mínimo aos usuários.

### 19. Documentação:

- Criar documentação abrangente, incluindo manuais de usuário, manuais técnicos e qualquer documentação relevante para os desenvolvedores e usuários.

### 20. Treinamento:

- Fornecer treinamento para os usuários finais e para a equipe de suporte, se necessário.

### 21. Manutenção e Atualizações:

- Monitorar o desempenho do software em produção.
- Corrigir bugs e implementar atualizações conforme necessário.

### 22. Feedback e Melhoria Contínua:

- Coletar feedback dos usuários para melhorar continuamente o software.
- Avaliar o desempenho do projeto e identificar áreas de melhoria para futuros projetos.

## **2.4 METAS, CRITÉRIOS OU INDICADORES DE AVALIAÇÃO DO PROJETO**

Para a realizar este item, foram traçados planos que foram descritos anteriormente na documentação, os quais foram atendidos dentro do prazo.:

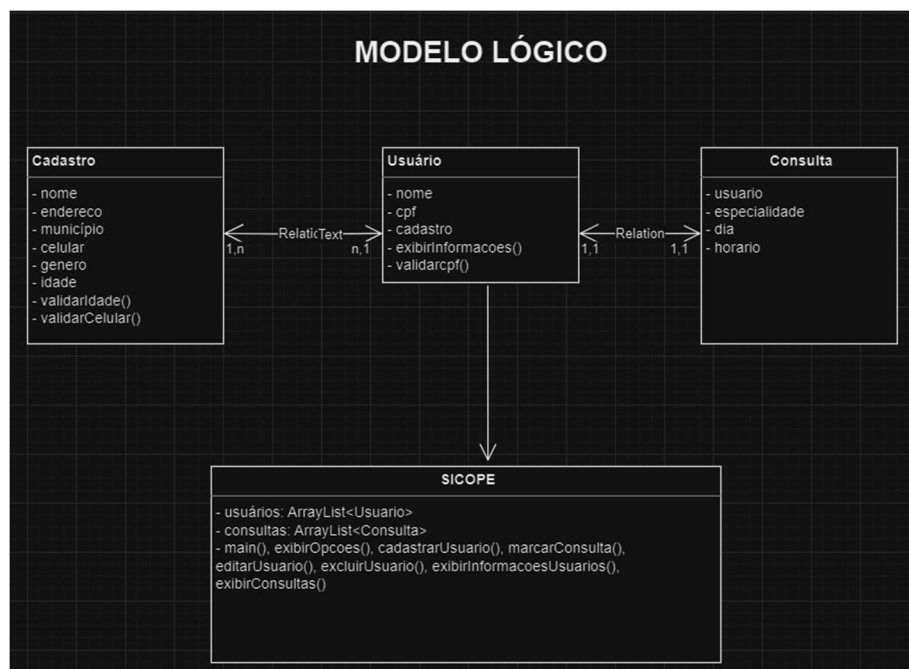
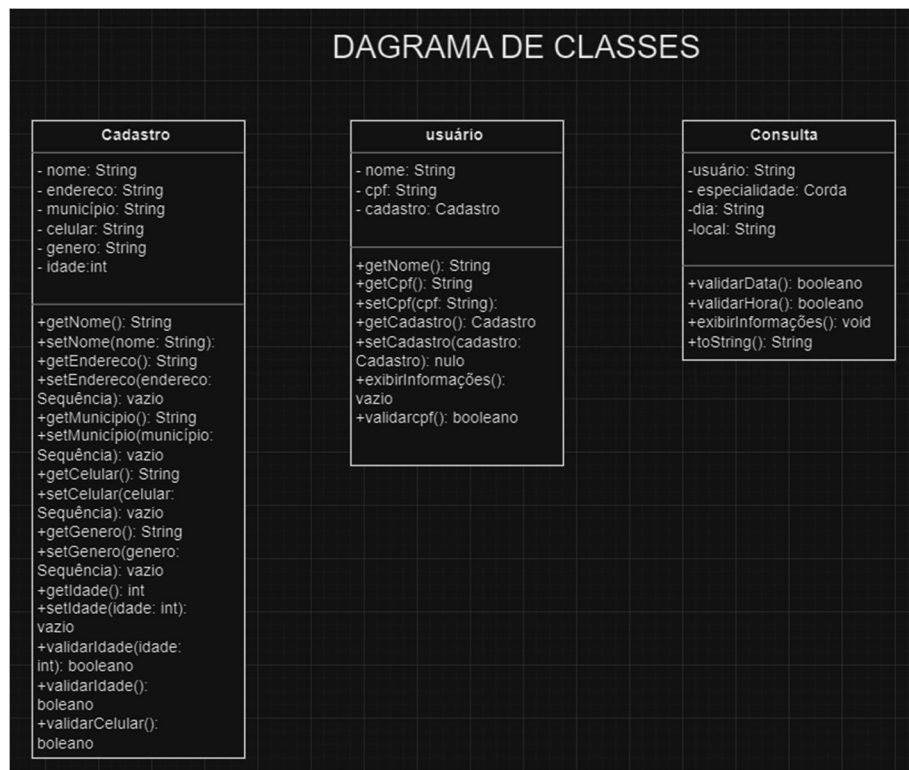
- Foi criado um sistema de cadastro simples, que foi testado e aprovado pelos usuários;
- O sistema foi construído atingindo os critérios de segurança e robustez;
- O sistema foi construído seguindo todos os Conhecimento Acadêmico aprendidos em **Java**, utilizando a Programação Orientada a Objeto;

## **2.5 RECURSOS PREVISTOS**

Para a execução do Projeto, não foram utilizadas ferramentas que onerassem sua elaboração por ser gratuita (VsCode), logo, a única despesa seria o tempo gasto pelo programador na elaboração do Projeto.

## 2.6 DETALHAMENTO TÉCNICO DO PROJETO

Como já salientado na documentação, foi utilizada a Linguagem de Programação **Java**, irá ser demonstrado a seguir como ficou a modelagem ao término do projeto.:



### **3. ENCERRAMENTO DO PROJETO**

#### **3.1 RELATÓRIO COLETIVO (PODENDO SER ORAL E ESCRITA OU APENAS ESCRITA)**

Por se tratar de um Projeto individual, será descrito no item 3.3 do documento.

##### **3.1.1 AVALIAÇÃO DE REAÇÃO DA PARTE INTERESSADA**

Por se tratar de um Projeto onde o cliente é fictício, não será relatado algo no tópico

#### **3.2 RELATO DE EXPERIÊNCIA INDIVIDUAL**

Discente.: Ney Augusto da Silva Vicente

Matrícula.: 2021040828-7

##### **3.2.1 CONTEXTUALIZAÇÃO**

Sobre a experiência adquirida na elaboração do Projeto, posso salientar o aumento significativo de minha Hard Skill, através das horas dedicadas ao estudo da Linguagem, o conhecimento adquirido foi e será importante para meu desenvolvimento Técnico no âmbito profissional, pois ao conhecer a Linguagem **Java** (temida por muitos) surgiu uma tremenda busca em melhorar o conhecimento sobre ela, para assim agalgar cargas que tem uma grande demanda no mercado, mas não profissionais qualificados para entendê-la.

##### **3.2.2 METODOLOGIA**

Através das consultas realizadas com vizinhos e familiares (de diversas faixas etárias e grau de escolaridade), o projeto foi focado em atender uma simples da maioria dos entrevistados, algo que fosse simples de utilizar.

Mas também foi analisado o lado do cliente, onde o sistema além de ser simples, deveria ser seguro e atender os critérios exigidos pela academia, que foi seguido a risca e executado com precisão.

##### **3.2.3 RESULTADOS E DISCUSSÃO:**

Particularmente a Linguagem **Java**, sempre me trouxe fascínio porque para ser executada corretamente o desenvolvedor tem de demonstrar conhecimento dela, por “falta de tempo” nunca consegui me dedicar. Após o término do Projeto, me desperto o sentimento de conhecer profundamente **Java** ao ponto de já ter adquirido um curso para me especializar.

Tive algumas dificuldades com o projeto, inicialmente teria conexão com Banco de Dados MySQL (o que não foi obrigatório) resultando na alteração do projeto inicial, preferi realizar a entrega de um Projeto mais simplório, mas que realmente atuei ao invés de auxílio onde poderia não ter adquirido conhecimento técnico.

#### **3.2.4 REFLEXÃO APROFUNDADA**

Como não houve relato coletivo, não irei discorrer no tópico.

#### **3.2.5 CONSIDERAÇÕES FINAIS**

Como consideração final, poderia citar alguns pontos.:

- O Projeto pode ser utilizado por qualquer empresa do ramo Médico-Hospitalar;
- O Projeto poderá ser implementado com uma conexão a banco de dados;
- Implementar campos na área de cadastro;