

## Arquitetura de um Ambiente para a Associação de Apoio aos Portadores de Câncer de Mossoró e Região (AAPCMR)

### An Architecture of an Environment for the Association of Support for Cancer Carriers of Mossoró and Region (AAPCMR)

Maria Lanuza dos Santos Silva<sup>1\*</sup>, Reudismam Rolim de Sousa<sup>1</sup>, José Wagner Cavalcanti Silva<sup>1</sup>

---

#### RESUMO

A Associação de Apoio aos Portadores de Câncer de Mossoró e Região (AAPCMR) é uma entidade sem fins lucrativos que tem como missão prestar assistência social e amparo ao paciente com câncer, a fim de facultar ao paciente e seu acompanhante, com hospedagem digna e apoiando com ações sociais como cesta básica, fisioterapia e tratamento odontológico. Nesse sentido, a casa de apoio necessita realizar o gerenciamento de suas unidades, para facilitar o atendimento aos pacientes e o controle dos processos. Atualmente, eles gerenciam a associação e suas funções através de ferramentas como o papel e planilha de texto. O principal objetivo deste artigo é propor um sistema que possibilite agilizar a gerência das ações exercidas pelo núcleo de apoio.

**Palavras-chave:** Pacientes; Gerenciamento; Software; Sistema;

---

#### ABSTRACT

The Association of Support for Cancer Carriers of Mossoró and Region (AAPCMR) is a non-profit organization whose mission is to provide social assistance and support to cancer patients, to provide the patient and his companion with decent accommodation and support with social actions such as basic food, physical therapy and dental treatment. In this sense, the support house needs to manage its units, to facilitate patient care and process control. Currently, they manage the association and its functions through tools like paper and text sheets. The main objective of this article is to propose a system that makes it possible to streamline the management of the actions carried out by the support nucleus.

**Keywords:** Patients; Management; Software; System;

---

---

<sup>1</sup> Universidade Federal Rural do Semi-Árido. \*E-mail: maria.silva44240@alunos.ufersa.edu.br

## INTRODUÇÃO

As associações de apoio aos portadores de câncer são institutos importantes para a sociedade, atuando como serviços de apoio psicológico, médico e jurídico. Essas instituições desempenham um papel de assistencialismo e conseguem levantar dados estatísticos relevantes para o aprimoramento de políticas públicas (GLOBO CIÊNCIA, 2012). Um meio de auxiliar os processos e atividades exercidas pela nesses ambientes é a utilização de sistemas de tecnologia da informação. Atualmente, a forma de controlar as ações executadas pela Associação de Apoio aos Portadores de Câncer de Mossoró e Região (AAPCMR) é realizado manualmente com caneta e papel e, algumas ações, como o relatório das movimentações da casa de apoio é feita por planilha eletrônica e editor de texto. Neste sentido, esses processos levam a dificuldades de execução, como o cadastro e a identificação de pacientes, além do acompanhamento do tratamento.

Considerando que a gerência e o controle de todas as ações realizadas pela associação é fundamental, este trabalho propõe uma arquitetura de software para o Sistema Tecnológico Associação de Apoio aos Portadores de Câncer de Mossoró e Região (STAAPCMR), cujo objetivo é reduzir a utilização das ferramentas tradicionais, de modo que a instituição tenha seu processo de atendimento aos pacientes mais ágil, facilitando o controle para os voluntários que realizam os procedimentos diários, como cadastro, ações de apoio, relatórios, identificação de pacientes, entre outros. O STAAPCMR é um software destinado ao ambiente organizacional, que busca obter o controle e a segurança das informações dos pacientes e seus familiares, bem como melhorar o gerenciamento de todas as atividades de apoio. Os usuários do STAAPCMR são os voluntários da instituição, responsáveis pela gestão geral da associação, que devem fazer o uso do sistema, executando as funções de cadastrar pacientes, emitir relatórios, acompanhar tratamentos, editar e/ou excluir dados. Basicamente, estes serão os únicos que terão acesso às informações e funcionalidades disponibilizadas pelo sistema.

Como a casa de apoio faz o uso de ferramentas tradicionais como caneta e papel, em conjunto com a utilização de planilha eletrônica e editor de texto, ela enfrenta alguns problemas pela ausência de um sistema de software que atenda às especificidades do cotidiano da instituição. Os desafios encarados pela unidade de Pau dos Ferros da AAPECMR são: segurança das informações, ausência de agilidade para armazenamento e busca dos dados, lentidão na execução das atividades e processos diários, retrabalho,

outra dificuldade enfrentada é a defasagem dos computadores utilizados, que é uma das causas da lentidão nos processos diários.

Dessa forma, a proposta deste trabalho é orientar uma estrutura arquitetural de um sistema que disponha de funcionalidades essenciais para o funcionamento da instituição, de modo que as atividades possam ser executadas mais rapidamente, possibilitando agilidade e segurança.

O trabalho está disposto da seguinte forma: na Seção 2 é apresentada a arquitetura do sistema, composta pela definição dos requisitos do sistema e dos artefatos que possibilitam à equipe de desenvolvimento entender a organização, implementação e evolução do software. Por sua vez, na Seção 3 é apresentada a avaliação da arquitetura do sistema, que permite realizar validações e análise dos documentos arquiteturais através de procedimentos e métodos sistêmicos. Por fim, na Seção 4 são apresentadas as considerações finais do trabalho.

## **ARQUITETURA**

Uma arquitetura é a estrutura do sistema, composta de elementos de softwares, das propriedades externamente visíveis desses elementos e dos relacionamentos entre eles (BASS et al., 2012). O termo também tem relação com a documentação, facilitando a comunicação entre os stakeholders, permitindo a tomada de decisões iniciais e permite o reuso do projeto dos componentes e padrões. O projeto de arquitetura é uma das primeiras fases do desenvolvimento do software. Ele identifica os principais componentes estruturais do sistema e os relacionamentos existentes entre eles. O resultado de todo o processo é um modelo de arquitetura que descreve como o sistema está organizado em um conjunto de estruturas, componentes, estrutura de implantação, dentre outros elementos.

## **REQUISITOS DE SISTEMAS**

Os requisitos de um sistema são as descrições das funcionalidades que o sistema deve desempenhar e as restrições em relação ao seu funcionamento (VALENTE, 2022).

Esses requisitos são essenciais para o desenvolvimento do software, de modo a atender as necessidades dos clientes e dos usuários. Existem dois tipos de requisitos, os funcionais, que são definidas as funções do sistema e os não funcionais, que estão relacionados com as características de qualidade (BASS et al., 2012). Nesta seção, aborda-se o levantamento dos requisitos funcionais e não funcionais.

O levantamento de requisitos é importante para continuação das atividades e etapas seguintes do desenvolvimento, pois essa etapa permitirá que os colaboradores envolvidos no projeto tenham o entendimento do sistema e alinhamento dos pontos definidos em relação ao produto final. Para executar esta etapa, algumas técnicas de levantamento precisam ser aplicadas, tais como entrevistas com os clientes, questionários, pesquisas com os usuários, prototipagem, pesquisas utilizando etnografia, entre outros (VALENTE, 2022).

Os sistemas têm como objetivo principal permitir realizar operações essenciais para o funcionamento da AAPCMR, de modo que as atividades possam ser executadas rapidamente, com agilidade e segurança. Os usuários do sistema são definidos no Quadro 1, sendo eles o servidor, que podem determinar os acessos; e o administrador, que é responsável por gerenciar as funcionalidades existentes no sistema, como cadastro de paciente e ações de apoio, emissão de relatórios, identificação, listagens, visualização, exclusão, adição e edição das informações dos pacientes.

**Quadro 1 – Usuários do STAAPCMR**

<b>Ator</b>	<b>Descrição</b>
Servidor	O sistema deve permitir cadastrar um novo administrador com login e senha.
Administrador	O sistema deve permitir a edição dos dados do administrador. O sistema deve permitir que o usuário administrador execute todas as funcionalidades.

Fonte: Autoria própria (2022)

## REQUISITOS FUNCIONAIS

Os requisitos funcionais têm o propósito de definir as funções do sistema, descrevendo os conjuntos de entradas e saídas e seus respectivos comportamentos (VALENTE, 2022). A elicitação dos requisitos propostos neste documento foi realizada através de questionários e entrevistas realizadas com os voluntários responsáveis pela associação. Na ocasião, as principais atividades e ações executadas pela instituição foram detalhadas e documentadas, tendo também suas prioridades atribuídas. Com isso, no Quadro 2 estão descritos os requisitos.

**Quadro 2** – Lista de requisitos do Sistema

<b>Requisito</b>	<b>Descrição</b>	<b>Prioridade</b>
<b>RF001-Login Administrador</b>	O sistema deve permitir que o usuário administrador realize login no sistema. Para isso, ele deve informar e-mail e senha.	Essencial
<b>RF002-Cadastro Administrador</b>	O sistema deve permitir que o usuário administrador realize um cadastro no sistema. Ele deve informar os dados: nome, e-mail e senha.	Essencial
<b>RF003-Editar Cadastro do Administrador</b>	O sistema deve permitir que o usuário administrador possa editar as informações contidas nos campos do cadastro do administrador.	Essencial

<b>RF004-Excluir Cadastro do Administrador</b>	O sistema deve permitir que o usuário exclua o cadastro de um administrador.	Essencial
<b>RF005-Cadastro Paciente</b>	O sistema deve permitir que o usuário administrador cadastre o paciente no sistema. Para isso, ele deve informar os dados: nome, estado civil, naturalidade, data de nascimento, RG, CPF, endereço, telefone residencial, celular, diagnóstico, ano de descoberta do diagnóstico, médico que acompanha o tratamento, hospital de tratamento, status do Tratamento, observação e data.	Essencial
<b>RF006-Editar informações do cadastro Paciente</b>	O sistema deve permitir que o administrador edite as informações contidas nos campos de cadastro do paciente.	Essencial
<b>RF007-Excluir cadastro do Paciente</b>	O sistema deve permitir que o administrador exclua o cadastro do paciente.	Essencial
<b>RF008-Identificação do Paciente</b>	O sistema deve permitir que o administrador identifique um paciente. Para isso, ele deve informar os dados: nome do pai, nome da mãe, religião, profissão e participação de programa do governo.	Essencial

<b>RF009-Editar informações de identificação do Paciente</b>	O sistema deve permitir que o administrador edite as informações contidas nos campos de identificação do paciente.	Essencial
<b>RF010-Informações sobre o Tratamento</b>	O sistema deve armazenar informações sobre o tratamento do paciente. Para isso, deve conter as seguintes informações: tipo de cirurgia, realização de tratamento anterior, histórico familiar de câncer, uso de álcool/droga.	Essencial
<b>RF011-Editar informações sobre o Tratamento</b>	O sistema deve permitir que o administrador edite as informações contidas nos campos do tratamento do paciente.	Essencial
<b>RF012-Composição Familiar</b>	O sistema deve armazenar as informações sobre a composição familiar do paciente. Para isso, ele deve informar os dados: nome, idade, vínculo, ocupação e renda.	Importante
<b>RF013-Editar informações da Composição Familiar</b>	O sistema deve permitir que o administrador edite as informações contidas nos campos de composição familiar.	Importante
<b>RF014-Situação Habitacional</b>	O sistema deve armazenar as informações sobre a situação habitacional do paciente. Para isso, ele deve informar os dados: casa, moradia, número de cômodos, instalação sanitária, luz, água e transporte.	Importante

<b>RF015-Editar informações de Situação Habitacional</b>	O sistema deve permitir que o administrador edite as informações contidas nos campos de situação habitacional do paciente.	Importante
<b>RF016-Listar Pacientes</b>	O sistema deve permitir que o administrador possa listar os pacientes cadastrados no sistema.	Essencial
<b>RF017-Visualizar as informações dos Pacientes</b>	O sistema deve permitir que o usuário administrador ao clicar em um paciente na listagem, consiga visualizar suas informações.	Essencial
<b>RF018-Listar pacientes que recebem Cesta Básica</b>	O sistema deve permitir que o usuário administrador consiga listar por ordem alfabética os pacientes que recebem cesta básica da instituição.	Essencial
<b>RF019-Relatório mensal</b>	O sistema deve permitir que o usuário administrador baixe um relatório. Contendo as seguintes informações: Movimentações, Data, Entrada, Saída, Saldo, Total de entradas, Total de saídas e Saldo do mês.	Essencial
<b>RF020-Ações de Apoio</b>	O sistema deve permitir que o usuário administrador consiga controlar as ações realizadas pela instituição. As ações de apoio da associação são: Trabalhos odontológicos,	Essencial

	fisioterapia, Trabalhos em grupo com auxílio de voluntários, Cesta básica aos familiares do paciente.	
<b>RF021-Cadastrar Ações de Apoio Semanal</b>	O sistema deve permitir que o usuário administrador consiga fazer o cadastro das ações de apoio que serão realizadas durante a semana. Contendo as seguintes informações: Tipo de Ação, Dia e Horário.	Essencial
<b>RF022-Exibir as Ações de Apoio da Associação</b>	O sistema deve permitir que o usuário possa visualizar as ações realizadas pela instituição, exibindo estas em forma de calendário.	Essencial
<b>RF023-Editar as Ações de Apoio</b>	O sistema deve permitir que o usuário consiga editar as informações contidas nos campos de ações de apoio efetuadas pela associação.	Essencial
<b>RF024-Excluir informações das Ações de Apoio</b>	O sistema deve permitir que o usuário administrador possa excluir informações contidas nos campos de ações de apoio.	Essencial

Fonte: Autoria própria (2022)

## REQUISITOS NÃO FUNCIONAIS

Além dos requisitos funcionais, na definição da arquitetura de um sistema também são considerados requisitos relacionados à qualidade do sistema e restrições impostas ao software (BASS et. al., 2012; SOMMERVILLE, 2019).

Os requisitos de qualidade usualmente são expressados por cinco elementos, conforme elencado por Bass et. al. (2012): 1) estímulo, que representa o evento que disparou a ação, afetando alguma qualidade do sistema (e.g., usabilidade, segurança, performance) ou ao desenvolvimento do sistema (e.g., um pedido por adição de uma funcionalidade no software); 2) fonte do estímulo, que denota a fonte que originou o evento; 3) resposta, que caracteriza a resposta que é dada ao receber o estímulo; 4) medida de resposta, que denota a métrica que determina se a resposta foi satisfatória. 5) ambiente, que caracteriza o contexto em que o estímulo ocorreu; e 6) artefato, que representa a porção do sistema ao qual o estímulo se aplica.

Para o ambiente proposto, pode-se citar como requisito de qualidade o controle de acesso e o suporte à modificabilidade (Quadro 3 e Quadro 4).

**Quadro 3** – Requisitos de qualidade para Controle de acesso

<b>Controle de acesso</b>	
<b>Estímulo</b>	O administrador pretende entrar no sistema.
<b>Fonte do estímulo</b>	Necessidade do administrador em operar o sistema.
<b>Resposta</b>	O sistema verifica as credenciais do administrador. Se o login e a senha do usuário forem de um administrador cadastrado no ambiente, este usuário poderá acessar o ambiente. Caso o login ou a senha seja inválida, uma mensagem de erro deve ser fornecida para o usuário, o informando para adicionar um login ou uma senha válida.
<b>Medida de resposta</b>	A resposta deve ser fornecida em até 5 segundos.
<b>Ambiente</b>	O sistema está operando em condições normais de uso (e.g., 4 GB+ de memória RAM, processador Celeron+, poucos outros programas em uso).
<b>Artefato</b>	Módulo responsável por verificar as credenciais do usuário.

Fonte: Aatoria própria (2022)

**Quadro 4** – Requisitos de qualidade para Modificabilidade

<b>Modificabilidade</b>	
<b>Estímulo</b>	O time de desenvolvimento é acionado para desenvolver uma nova funcionalidade para o ambiente.
<b>Fonte do estímulo</b>	Solicitação do cliente.
<b>Resposta</b>	A equipe de desenvolvimento implementa a funcionalidade e testa o ambiente.
<b>Medida de resposta</b>	A nota funcionalidade de ser adicionada ao sistema e testada em no máximo 5 dias.
<b>Ambiente</b>	A equipe de desenvolvimento está em período normal de desenvolvimento (i.e., não está próximo da próxima entrega do sistema.)
<b>Artefato</b>	A implementação da funcionalidade deve afetar apenas as classes voltadas aos atores que a funcionalidade de refere (e.g., paciente, administrador).

Fonte: Aatoria própria (2022)

## RESTRICÇÕES

As restrições definem decisões que foram tomadas e que a arquitetura do software deve suportar, a exemplo do uso de uma determinada linguagem de programação ou framework específico. Para o ambiente proposto foi especificado como restrição que o ambiente deve ser voltado para o ambiente Web.

## VISÕES ARQUITETURAIS

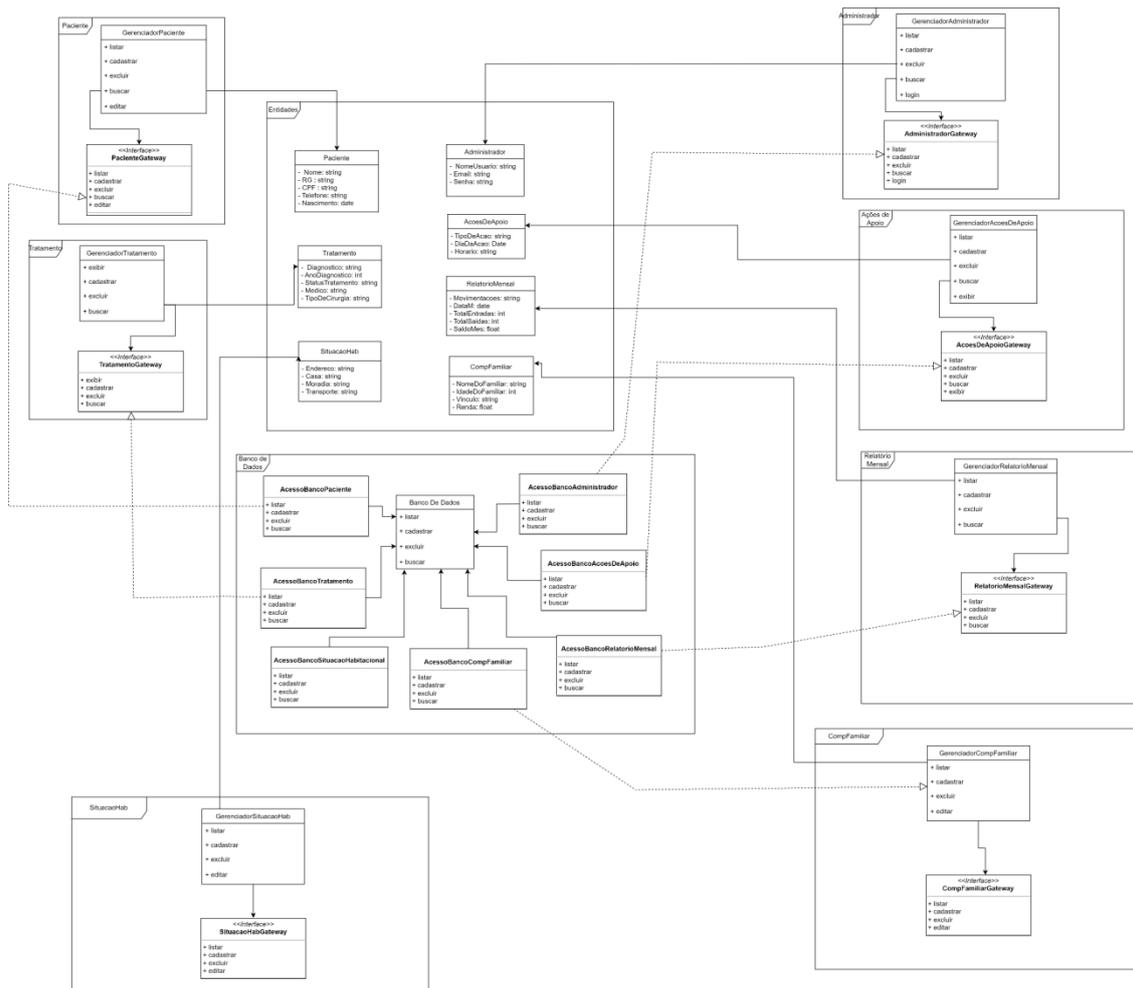
Os sistemas de software quando estão em construção, podem ser observados de várias formas distintas, de modo que as visões permitem projetar um modelo sob determinada perspectiva. Neste sentido, existem alguns tipos de representações gráficas para as visões. Porém, os principais tipos utilizados são as visões de módulo, componente & conector e alocação (BASS et al., 2012).

## VISÃO DE MÓDULO

A visão de módulo permite visualizar o sistema em termos de estruturas de código ou unidade de dados (BASS et al., 2012). Os módulos possibilitam entender as responsabilidades dos elementos do código (classes, camadas, etc.), com ênfase em como o código está estruturado de forma estática, sem atenção a como os elementos se relacionam em tempo de execução, permitindo responder diversas perguntas, tais como as funcionalidades de cada módulo; as unidades de código que os módulos dependem e interagem ou se ocorre relação de herança entre as unidades de código (BASS et al., 2012).

Uma das formas de visualizar a visão de módulos é por meio de um diagrama de classes. Na Figura 1 pode ser visto o diagrama de classes para o ambiente proposto neste trabalho.

**Figura 1** – Diagrama de classes do ambiente



Fonte: Autoria própria (2022)

Para definir as classes que representam o ambiente foi criado um pacote com as classes que caracterizam o sistema (e.g., Paciente, Administrador, Ações de Apoio, Situação Habitacional, Composição Familiar, Relatório Mensal, Tratamento). Cada uma dessas classes possui atributos específicos que as caracterizam e possuem um gerenciador para lidar com suas principais operações.

Ao longo da estruturação dos elementos no diagrama de classes, buscou-se seguir os princípios elencados pelo projeto de software definido pelo SOLID (MARTIN, 2019). Por exemplo, as classes são separadas de acordo com as suas responsabilidades, Princípio da Responsabilidade Única (i.e., relação com um ator específico) e empregou-se o Princípio da Inversão de Dependência para que as classes fiquem desacopladas do banco de dados.

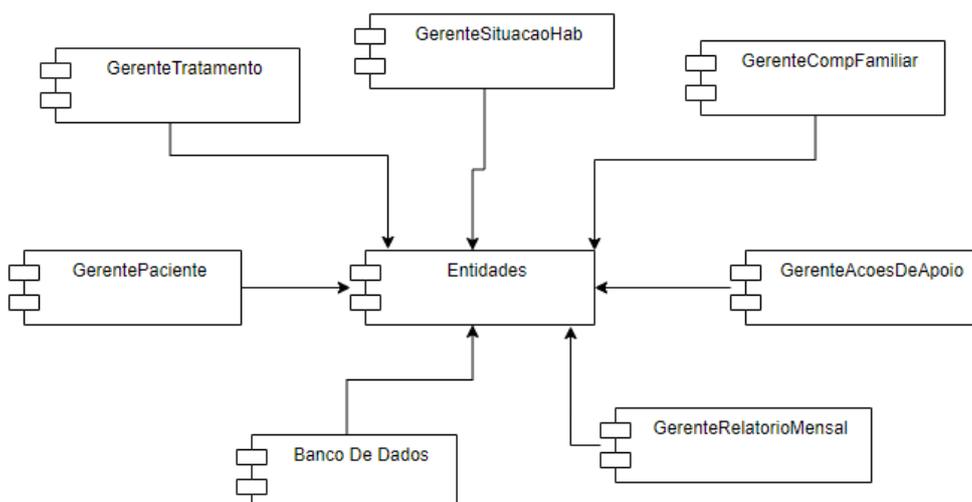
## VISÃO COMPONENTE & CONECTOR

A visão componente & conector permite visualizar o ambiente em termos de componentes, que representam elementos com comportamentos em tempo de execução (serviços, clientes, servidores, etc.) e de conectores, que elencam como os elementos do ambiente interagem entre si (mensagens, chamada de método, etc.) (BASS et al. 2012). Essa visão permite responder a perguntas como se há dados compartilhados entre os elementos; que componentes podem rodar em paralelo ou como os dados progridem ao longo do sistema (BASS et al. 2012).

Os componentes foram definidos de modo que apresentem gerentes para os atores do sistema (e.g., Paciente, Tratamento, Composição Familiar). Essa organização buscou seguir os princípios de Coesão de Componentes propostos por Martin (2019), a exemplo do *Princípio do Fechamento Comum*, que informa para reunir em componentes as classes que mudam pela mesma razão e momentos e para separar em componentes distintos as classes que não obedecem a essa característica.

Uma das maneiras de representar a visão componente & conector é por meio de diagramas de componentes. A visão de módulo para o ambiente pode ser vista na Figura 2.

**Figura 2** – Diagrama de componentes do ambiente



Fonte: Autoria própria (2022)

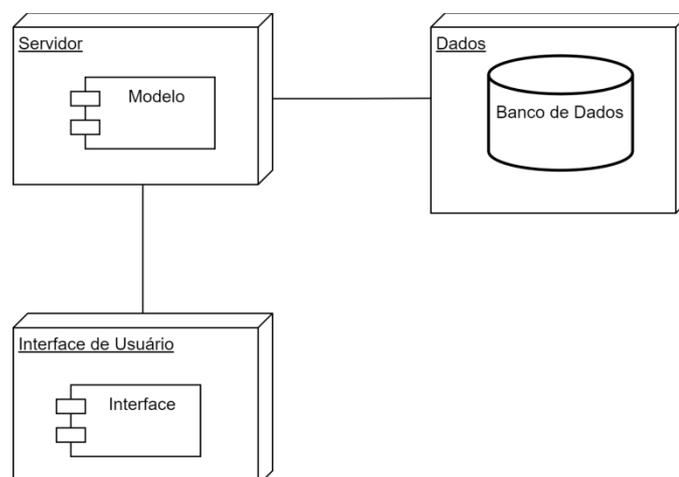
Além disso, os componentes foram organizados de forma a seguir os princípios relacionados ao acoplamento entre componentes. Por exemplo, usou-se o *Princípio da Dependência Acíclica*, que não permite ciclos no grafo de componentes (note na Figura 2, que não é possível seguir um caminho pelo grafo de componentes e voltar ao mesmo componente). Ademais, empregou-se o *Princípio da Dependência Estável*, que determina para depender na direção da estabilidade. Dessa forma, as dependências (setas no grafo) são direcionadas da região mais externa para o centro, visto que classes relacionadas a banco de dados são estáveis (mudam com frequência), do mesmo modo os gerentes. Como nenhum outro componente depende das classes mais externas, essas possuem liberdade para evoluir, favorecendo a modificabilidade, um dos requisitos de qualidade do ambiente.

## VISÃO DE ALOCAÇÃO

A visão de alocação possibilita visualizar como o sistema está integrado a outras estruturas, que não são de software, tais como equipes de desenvolvimento, sistema de arquivo, etc, permitindo visualizar como o sistema está distribuído e quem são os responsáveis pelo desenvolvimento/manutenção de cada elemento (BASS et al. 2012).

Uma das formas de representar uma visão de alocação é por meio de diagramas de implantação. Na Figura 3 pode ser visto o diagrama de implantação para o sistema proposto.

**Figura 3** – Diagrama de implantação do ambiente



Fonte: autoria própria (2022)

O sistema foi organizado permitindo que o usuário possa acessar o sistema por meio de um navegador Web, representado no sistema pelo bloco Interface de Usuário. A interface de usuário permite ao sistema acesso os elementos do modelo, representado pelo bloco Servidor. Por sua vez, as unidades do modelo acessam a base de dados, para executar as suas operações, representado no diagrama pelo bloco Banco de Dados.

No tocante a capacidade de ser modificado, caso ocorra necessidades de modificações, por exemplo se a equipe desejar que o software suporte a doação. O usuário precisa criar classes que representam as entidades referente às doações, cria também o gerenciador para a nova entidade e as correspondentes classes associadas a base de dados. O uso de princípios relacionados ao projeto de arquitetura de software permite que o sistema possua a capacidade de ser modificado sem altos custos.

## **AVALIAÇÃO DA ARQUITETURA**

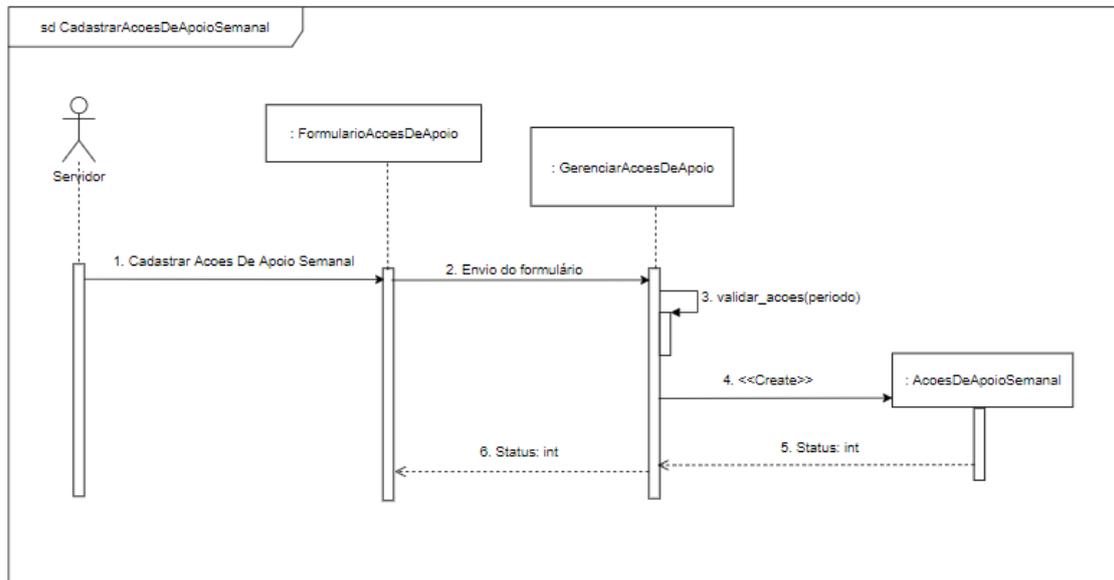
Uma avaliação arquitetural tem como objetivo determinar se a arquitetura atende ao objetivo para a qual foi projetada (BASS et al., 2012). Um dos métodos utilizados para avaliação da arquitetura é o Método de Análise Arquitetural baseado em *Tradeoffs* (*Architecture Tradeoff Analysis Method*) - ATAM, que consiste de uma série de fases, pessoas e etapas para avaliar a arquitetura proposta (BASS et al. 2012).

Como o ATAM requer um conjunto determinado de processos e pessoas, torna-se custoso para equipes pequenas. Neste sentido, para avaliar a arquitetura do ambiente proposta, utilizou-se uma abordagem baseada em cenários, que consiste em um dos principais elementos utilizados nas avaliações arquiteturais.

Na Figura 4 pode ser visto o diagrama de sequência, representando o cenário de uso para cadastrar ações de apoio semanal.

Para cadastrar uma ação de apoio semanal, o usuário (servidor) acessa o formulário de ações de apoio. Ao enviar o formulário, a solicitação é recebida pelo gerenciador de ações de apoio, que efetivamente cria uma ação de apoio.

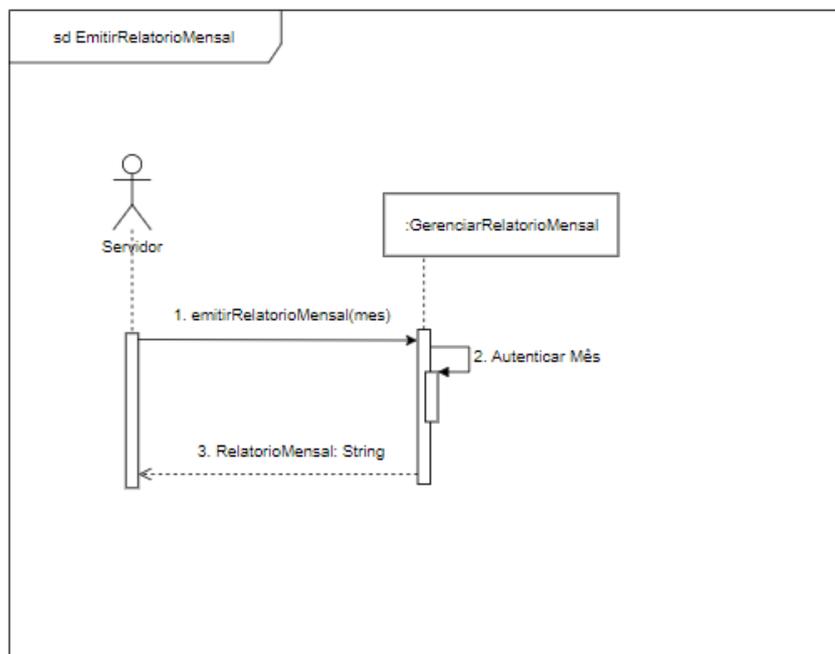
**Figura 4** – Diagrama de sequência cadastrar ações de apoio semanal



Fonte: Autoria própria (2022)

No tocante à emissão de relatório mensal, os principais elementos envolvidos podem ser vistos na Figura 5.

**Figura 5** – Diagrama de sequência emitir relatório mensal



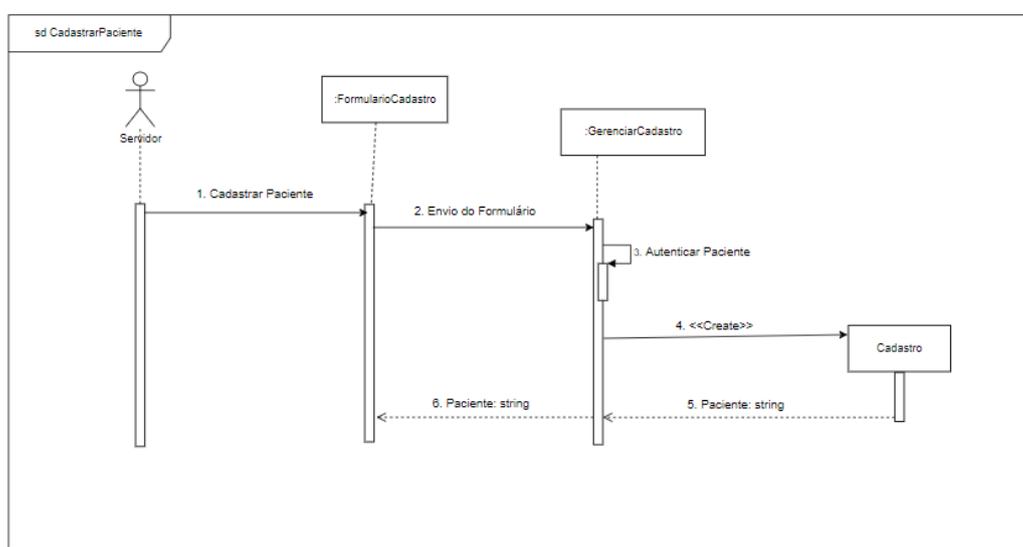
Fonte: Autoria própria (2022)

Para emitir um relatório mensal, o usuário aciona a operação de emissão de relatório mensal e o gerenciador de relatório mensal, envia as informações solicitadas.

O último cenário referente às funcionalidades do sistema é cadastrar o paciente, que pode ser visto na Figura 6.

Para cadastrar um paciente, o usuário acessa o formulário de cadastro, que ao solicitar o cadastro o gerenciador de cadastro cria o cadastro em si do paciente.

**Figura 6** – Diagrama de sequência cadastrar paciente



Fonte: Autoria própria (2022)

## CONCLUSÃO

Neste trabalho foi proposto um Sistema Tecnológico para a Associação de Apoio aos Portadores de Câncer de Mossoró e Região. O sistema tem como objetivo melhorar o processo de gerenciamento e controle das atividades da instituição, que até o momento eram realizadas por meio de papel e planilhas de texto. Com a utilização do sistema, operações como cadastros, identificações, relatórios e listagens serão realizadas de forma automatizada.

Para organizar o escopo e definir as características do sistema, foram realizadas técnicas de levantamento de requisitos como questionários e entrevistas com os colaboradores. Após o refinamento das informações gerais do sistema, desenvolveram-se

diagramas de classes, componentes, implantação e sequência, para prover uma melhor comunicação e alinhamento entre as partes interessadas no projeto. Além disso, foram apresentadas ferramentas necessárias para o desenvolvimento e implementação do software, os requisitos funcionais e não funcionais, e uma avaliação arquitetural do sistema. Neste sentido, espera-se que o sistema seja utilizado, mantido e evoluído, de maneira que atenda de forma exponencial as necessidades do ambiente.

Futuramente, pretende-se potencializar a evolução e manutenção do software, com práticas preventivas e corretivas, em conformidade com o feedback do cliente e dos usuários da aplicação. Assim, faz-se necessário também realizar uma avaliação formal com os usuários do sistema, visando identificar possíveis correções e melhorias.

## **AGRADECIMENTOS**

Agradecemos ao grupo de pesquisa LIS - Laboratório de Inovações em Software e à Universidade Federal Rural do Semi-Árido - UFRSA, por meio da Pró-Reitoria de Extensão e Cultura (PROEC) através do Edital PROPPG Nº 16/2021, pelo apoio na produção deste trabalho.

## **REFERÊNCIAS**

GLOBO. Associações para pacientes com câncer vão além do assistencialismo. **Globo Ciência**, 18 jun. 2016.

BASS, Len; CLEMENTS, Paul e KAZMAN, Rick. **Software architecture in practice** (3rd. Ed.). Addison - Wesley professional, 2012.

MARTIN, Robert C. **Arquitetura Limpa: o guia do artesão para estrutura e design de software**. 1. ed. Rio de Janeiro: alta books, 2019.

VALENTE, Marco Tulio. **Engenharia de Software Moderna**. 1. ed. Minas Gerais: independente, 2022.

SOMMERVILLE, Ian. **Engenharia de Software**. 10, ed. São Paulo: pearson universidades, 2019.