



**UNIP – UNIVERSIDADE PAULISTA**

**SISTEMAS DE INFORMAÇÃO/ CIÊNCIA DA COMPUTAÇÃO**

Georgia Heloísa Godoy – N589BD3

Maria Eduarda Liomerio – F2473H0

Thiago Furtado Valcazara – N6546E0

**APS – ATIVIDADES PRÁTICAS SUPERVISIONADAS**

**6º/7º**

## SUMÁRIO

OBJETIVO DO TRABALHO.....	3
CONCEITOS GERAIS.....	
1. METODOLOGIA P/ DESENVOLVIMENTO DE SOFTWARE.....	
DOCUMENTOS DE REQUISITO.....	
MODELAGEM.....	
1. Diagrama de Casos de Uso.....	
2. Descrição dos Casos de Uso.....	
3. Diagrama de Classes.....	
4. Diagrama de Sequências.....	
CONCLUSÃO.....	
BIBLIOGRAFIA.....	
ANEXOS.....	

## **OBJETIVO DO TRABALHO**

O objetivo com esse trabalho é de fazer a modelagem de um software que automatize atividades de manutenção de dados. A manutenção de dados é uma tarefa fundamental em muitas organizações, mas que pode consumir muito tempo e recursos quando realizada manualmente. Com a criação de um software automatizado, poderemos otimizar essas atividades, reduzindo erros exponencialmente e aumentando a eficiência dos processos em si.

Neste documento são apresentados todos os passos tomados, necessários para a realização dessa automação, nossa modelagem de software será projetada para lidar com várias atividades de manutenção de dados, como inserção, atualização, exclusão e consulta de informações em um sistema.

Sendo consideradas seguintes etapas necessárias para o funcionamento, listando aqui algumas delas, listando cada necessidade e sua funcionalidade na prática: Análise de Requisitos, onde são coletados e documentados os requisitos específicos para a automação das atividades de manutenção de dados. Isso incluirá entender as necessidades dos usuários, as funcionalidades desejadas e as restrições técnicas;

Modelagem de Dados, onde são criados um modelo de dados que represente as entidades e relacionamentos relevantes para o sistema. Isso ajudará a organizar as informações e definir as estruturas de dados necessárias para armazenar e manipular os dados;

Diagrama de Casos de Uso, sendo uma representação gráfica utilizada na engenharia de software para descrever as interações entre os atores (usuários) e um sistema. Ele é uma das principais ferramentas da técnica de modelagem de casos de uso, que é amplamente utilizada na fase de análise e design de sistemas. O diagrama de casos de uso mostra os diferentes casos de uso (ou funcionalidades) de um sistema, representados como elipses, e os atores, representados como ícones, que interagem com esses casos de uso. As linhas de comunicação entre os atores e os casos de uso indicam a interação entre eles. O principal objetivo do diagrama de casos de uso é capturar os requisitos funcionais do sistema, ou seja, quais são as

funcionalidades que o sistema deve fornecer aos seus usuários. Ele permite que os desenvolvedores e os stakeholders compreendam as principais funcionalidades do sistema, identifiquem os atores envolvidos e estabeleçam uma base sólida para o desenvolvimento do software.

Um Diagrama de Classes de Entidade, embora não sendo comumente chamada desta maneira, também podendo se referir a apenas “Diagrama de Classes”, é uma representação gráfica utilizada na modelagem de sistemas orientados a objetos. Ele faz parte da Linguagem de Modelagem Unificada (UML) e tem como objetivo visualizar a estrutura estática de um sistema, mostrando as classes do sistema, seus atributos, métodos e os relacionamentos entre elas.

As classes são representadas como retângulos divididos em três compartimentos. O compartimento superior contém o nome da classe, o compartimento do meio lista os atributos da classe (variáveis) e o compartimento inferior mostra os métodos (funções) da classe.

Os relacionamentos entre as classes são representados por linhas, que podem ser de diferentes tipos, tais como: Associação, que mostra uma relação entre duas classes, indicando que uma está relacionada a outra de alguma forma; Agregação, indicando uma relação de “todo-parte” entre classes, onde uma classe é composta por outras classes, mas pode existir independentemente; Composição, semelhante a agregação, porém com uma relação mais forte de “todo-parte”, nesse caso uma classe é composta por outras classes mas que não pode existir sem elas; Herança: representando uma relação de especialização/generalização entre classes, onde uma classe em si herda parte mais específicas das características de uma classe mais genérica do que a outra primária em si.

Esperamos que a criação desse sistema possa servir de facilidade na prática de gestão de projetos para futuras avaliações de atividades de dados novas e mais práticas. Um sistema de automatização numa empresa é um progresso que faz ela alavancar na frente das outras e por isso fundamental para se ter em posse de um o quanto antes.

## **CONCEITOS GERAIS**

### **METODOLOGIA PARA DESENVOLVIMENTO DE SOFTWARE**

Dentre muitas metodologias diferentes para se testar a modelagem do software, é esperado que o usuário possa compartilhar com suas opiniões e feedbacks a respeito do software realizado, portanto uma abordagem mais iterativa seria o ideal para realização e abordagem escolhida.

Com mais pesquisas sendo coletadas pelos usuários, as melhorias seriam até mais concretas de se aprimorar no software, portanto uma metodologia de protótipo seria o mais ideal de se utilizar durante o escopo do projeto. Em resumo, a Metodologia do Protótipo é uma abordagem de desenvolvimento de software que enfatiza a criação rápida de versões iniciais do sistema, chamadas de protótipos, para obter feedback dos usuários e aprimorar o projeto. Com mais feedbacks, mais melhorias ao sistema virão futuramente.

Nessa metodologia, o desenvolvimento ocorre em ciclos iterativos e incrementais, em contraste com abordagens mais tradicionais que seguem um processo linear. A ideia é que a criação de protótipos permita uma melhor compreensão dos requisitos do sistema e das necessidades dos usuários, bem como a identificação antecipada de problemas e oportunidades de melhoria. O processo de desenvolvimento do protótipo geralmente passa por etapas detalhadas: Levantamento de requisitos; Projeto do protótipo; Desenvolvimento do protótipo; Testes e avaliação e Refinamentos e iteração do usuário final. Cada etapa será mais detalhada a seguir no documento.

Um levantamento de requisitos busca a compreensão das necessidades e requisitos do sistema, juntamente com a definição dos objetivos do protótipo sendo trabalhado. Ao chegar na criação de uma versão inicial do sistema, chegamos no primeiro protótipo, com foco nos recursos e funcionalidade essenciais aos usuários.

Durante seu desenvolvimento, finalmente é implementado o protótipo do sistema, usando uma abordagem ágil e interativa com os usuários, para uma obtenção de resultados o quanto tangíveis e rápidos. Logo os usuários

começam a realização de testes do protótipo do sistema e começam a equipe do projeto a coletar os feedbacks dos usuários, buscando identificar problemas, para solucioná-los o quanto antes na oportunidade de melhorias.

Com base no feedback recebido, o protótipo do sistema passa a ser refinado e aprimorado com o passar do tempo por novas atualizações em ciclos subsequentes de testes e desenvolvimento. Assim o projeto continua vivo e atualizado sempre.

A Metodologia do Protótipo é especialmente útil quando os requisitos do sistema não estão bem definidos totalmente ou quando há incerteza sobre as preferências e necessidades dos usuários, que por isso passam pelos feedbacks afins de melhorar o sistema com clareza. Ao permitir uma interação rápida e contínua com os usuários, a metodologia ajuda a reduzir riscos, melhorar a qualidade do sistema e aumentar a satisfação do cliente.

Claro, essa é uma escolha de Metodologia que na prática leva um certo tempo para adquirir os resultados desejados, visto que precisam fazer pesquisas de campo com os usuários afins de adquirir os problemas detectados e novas adições ao sistema, que pode começar simples ou não muito bem definidos com as preferências dos usuários.

A Metodologia empregada ajuda a reduzir os riscos, melhorar a qualidade do sistema e aumentar suas satisfações ao coletar seus feedbacks para transformar em melhorias aos próprios. Manter a transparência com seus usuários sem que deixe tudo muito complexo de entender, tornando algo automatizado, porém prático e mais padronizado para entendimento melhor dos usuários que, no público-alvo, são pessoas com Alzheimer.

## **DOCUMENTO DE REQUISITOS**

Assim como a Metodologia de Protótipo empregada durante o projeto, a documentação de requisitos precisa ser contínua e iterativa. Manter a comunicação constante com stakeholders e estar aberto a revisões. O projeto durante o processo da metodologia passará por mudanças e atualizações que eventualmente ocorrerão durante o ciclo de vida do projeto.

Uma melhor identificação durante o projeto garantirá que ele suceda com mais certezas de forma clara aos stakeholders e usuários. A começar, uma breve introdução do que será esclarecido no projeto ajudará ao usuário entender os motivos da realização dele, após todas as esclarecimentos começa o levantamento de requisitos, estes sendo uma análise completa e detalhada das necessidades e expectativas dos usuários.

Identificando os requisitos funcionais e os separando dos não-funcionais, teremos uma visão detalhada de cada requisito presente dentro do software. Prosseguindo com revisões e validações de stakeholders relevantes garantimos o progresso afirmativo, completo e que atendam às expectativas dos usuários. A validação dos usuários e feedbacks deles são o mais fundamental para que o sistema cresça e melhore com passar do tempo.

Introdução:

O sistema está sendo desenvolvido ao público-alvo sendo pessoas com Doença de Alzheimer, um transtorno neurodegenerativo progressivo e fatal que se manifesta pela deterioração cognitiva e da memória, comprometendo as atividades de vida diária e funções mentais importantes do dia a dia.

O sistema entregará a função de estabelecer várias tarefas como distribuir melhor os horários do dia a dia, permitir criar listas de tarefas, criação de rotinas e lembretes como ir ao mercado ou precisar tomar uma medicação, sempre notificando o usuário de cada passo tomado dentro do sistema.

Requisitos Funcionais do Sistema:

1. O sistema deve entregar aos usuários notificações de todas suas atividades dentro do aplicativo, para que não esqueçam de cada ação

que fizeram dentro dele, nem mesmo esqueçam de verificar o aplicativo periodicamente para resolução de novas ações.

2. O sistema deve permitir o usuário de utilizar as configurações do aplicativo que forem preparadas justamente para o usuário apenas.
3. O sistema deve permitir que usuários sejam capazes de ver suas rotinas criadas, para que visualizem as rotinas criadas e próximas a seguir dentro do aplicativo.
4. O sistema deve fazer a verificação de rotinas criadas anteriormente.
5. O sistema deve permitir que os usuários criem rotinas, para que se lembrem do que forem ordenados ao aplicativo de avisá-los futuramente pelas notificações.
6. O sistema deve permitir que usuários excluam suas rotinas criadas anteriormente, caso tenham errado ou apenas não seja mais necessária.
7. O sistema deve permitir que os usuários sejam capazes de ver sua lista de tarefas, para que visualizem as tarefas criadas e as já concluídas dentro do aplicativo.
8. O sistema deve fazer a verificação de listas de tarefas criadas anteriormente.
9. O sistema deve permitir que os usuários criem listas de tarefas, para que se lembrem do que for ordenado ao aplicativo de avisá-los futuramente pelas notificações.
10. O sistema deve permitir que usuários excluam suas listas de tarefas criadas anteriormente, caso tenham errado ou apenas não seja mais necessária.
11. O sistema deve permitir que os usuários criem tarefas em listas já existentes, para adicionar novas ações numa lista já criada sem que precisem deletar a antiga para a adição de tarefas novas.
12. O sistema deve permitir que usuários excluam tarefas criadas anteriormente, caso tenham errado ou apenas não seja mais necessária.



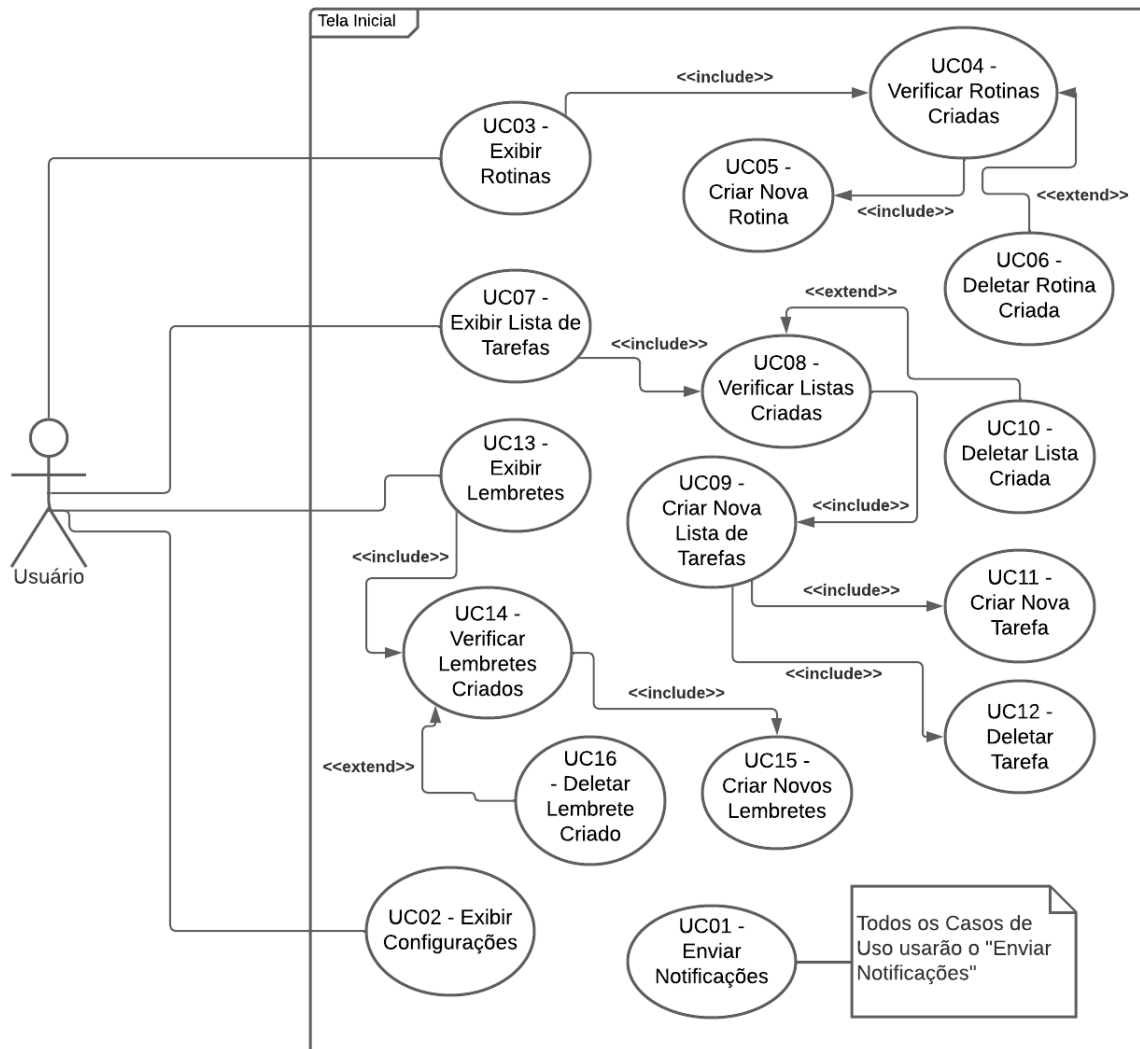
13. O sistema deve permitir que os usuários sejam capazes de ver seus lembretes, para que visualizem os lembretes já criados e os já concluídos dentro do aplicativo.
14. O sistema deve fazer a verificação de lembretes criados anteriormente.
15. O sistema deve permitir que os usuários criem lembretes, para que se lembrem do que for ordenado ao aplicativo de avisá-los futuramente pelas notificações.
16. O sistema deve permitir que usuários excluam lembretes criados anteriormente, caso tenham errado ou apenas não seja mais necessária.

#### Requisitos Não-Funcionais do Sistema:

1. O sistema deve ser capaz de processar indeterminadas rotinas criadas pelo usuário.
2. O sistema deve ser capaz de processar indeterminadas listas de tarefas criadas pelo usuário.
3. O sistema deve ser capaz de processar indeterminados lembretes criados pelo usuário.
4. O tempo de resposta do sistema não deve exceder 10 segundos em qualquer operação.
5. O sistema deve ser compatível com dispositivos Android.
6. O sistema deve manter os dados do aplicativo dentro do armazenamento do dispositivo.

## MODELAGEM DO SISTEMA

### DIAGRAMA DE CASOS DE USO



## DESCRIÇÃO DOS CASOS DE USO

### UC01 – Enviar Notificações

#### 1. Nome do Caso de Uso:

Enviar Notificações

#### 2. Breve Descrição:

Todas as ações envolvendo os outros casos de uso do usuário serão notificados para que se lembre das ações que fez e não esqueça o que acabou de executar.

3. Atores:

Usuário

4. Fluxo de Eventos:

4.1 Fluxo Principal:

1. O caso de uso começa quando o ator abre o aplicativo
2. O sistema exibe a tela inicial do aplicativo
3. O sistema requisita acesso a notificações do aplicativo
4. O ator permite o acesso.
5. O sistema notifica o ator a cada ação que entre em conjunto com outros casos de uso (Criar Nova Rotina, Deletar Rotina Criada etc.)

4.2 Fluxo Alternativo:

1. Logo após o sistema requisitar acesso a notificações do aplicativo, o ator recusa o acesso.
2. O sistema envia uma mensagem sugerindo que ative as notificações
3. O sistema não enviará notificações ao ator.

UC02 – Exibir Configurações

1. Nome do Caso de Uso:

Exibir Configurações

2. Breve Descrição:

O usuário poderá acessar as configurações do aplicativo para configurar horas, reconfigurar as notificações caso precise etc.

3. Atores:

Usuário

4. Fluxo de Eventos

4.1 Fluxo Principal:

1. O caso de uso começa quando o ator abre o menu
2. O sistema exibe a tela contendo as opções, dentro delas estão as Configurações
3. O ator clica na opção de “Configurações”

### UC03 – Exibir Rotinas

1. Nome do Caso de Uso:

Exibir Rotinas

2. Breve Descrição:

Ao usuário abrir o aplicativo o menu iniciar aparecerá, exibindo todas as rotinas (caso tenham sido criadas algumas).

3. Atores:

Usuário

4. Fluxo de Eventos

4.1 Fluxo Principal

1. O caso de uso começa quando o ator entra no aplicativo.
2. O sistema exibirá todas as rotinas criadas pelo usuário anteriormente.

4.2 Fluxo Alternativo:

FA01 – O sistema não possui rotinas criadas anteriormente

1. O sistema não exibirá rotinas criadas pelo usuário
2. O sistema sugerirá para que sejam criadas rotinas
3. O fluxo principal continua no passo 2

### UC04 – Verificar Rotinas Criadas

1. Nome do Caso de Uso:

Verificar Rotinas Criadas

2. Breve Descrição: O sistema irá verificar rotinas criadas anteriormente pelo usuário (caso sejam criadas anteriormente)

3. Atores: Sistema

4. Fluxo de Eventos

4.1 Fluxo Principal

1. O caso de uso começa quando o ator clica no Exibir Rotinas
2. O sistema irá verificar rotinas criadas anteriormente pelo usuário.

#### 4.2 Fluxo Alternativo

FA01 – Sistema não detecta rotinas criadas

1. O sistema sugere para que sejam criadas rotinas para prosseguir
2. O caso de uso é encerrado.

#### UC05 – Criar Nova Rotina

1. Nome do Caso de Uso

Criar Nova Rotina

2. Breve Descrição

O usuário poderá criar rotinas ao acessar a aba de Exibir Rotinas

3. Atores:

Usuário

4. Fluxo de Eventos

##### 4.1 Fluxo Principal

1. O caso de uso começa quando o usuário clicar em Criar Nova Rotina
2. O sistema permitirá que o usuário escolha as configurações acessíveis a personalização da rotina
3. O sistema encerrará o caso de uso quando a rotina for criada.

##### 4.2 Fluxo Alternativo

FA01 – O usuário cancela a criação de uma nova rotina

1. O sistema cancela a personalização da Nova Rotina e será encerrado.
2. O usuário é devolvido ao caso de uso do Exibir Rotinas
3. O caso de uso é encerrado.

#### UC06 – Deletar Rotina Criada

1. Nome do Caso de Uso

Deletar Rotina Criada

2. Breve Descrição:

O usuário poderá deletar rotinas ao acessar a aba de Exibir Rotinas

3. Atores:

Usuário

4. Fluxo de Eventos

4.1 Fluxo Principal

1. O caso de uso começa quando uma rotina criada é selecionada.
2. O sistema dá a opção ao usuário de Deletar Rotina Criada.
3. O usuário seleciona a rotina.
4. O sistema avisa que a rotina será permanentemente apagada
5. O usuário confirma a seleção
6. O sistema apaga a rotina.

4.2 Fluxo Alternativo

FA01 - O usuário cancela a eliminação de uma rotina

1. O sistema cancela o processo de eliminação da rotina
2. O usuário é devolvido ao caso de uso do Exibir Rotinas
3. O caso de uso é encerrado.

UC07 – Exibir Lista de Tarefas

1. Nome do Caso de Uso

Exibir Lista de Tarefas

2. Breve Descrição:

Ao usuário abrir o aplicativo o menu iniciar aparecerá, exibindo todas as listas de tarefas (caso tenham sido criadas algumas).

3. Atores: Usuário

4. Fluxo de Eventos

4.1 Fluxo Principal

1. O caso de uso começa quando o ator entra no aplicativo.
2. O sistema exibirá todas as listas de tarefas criadas pelo usuário anteriormente

#### 4.2 Fluxo Alternativo

FA01 – O sistema não possui listas de tarefas criadas recentemente

1. O sistema não exibirá listas de tarefas criadas pelo usuário
2. O sistema sugerirá para que sejam criadas listas de tarefas
3. O fluxo principal continua no passo 2

#### UC08 – Verificar Listas Criadas

1. Nome do Caso de Uso  
Verificar Listas Criadas

2. Breve Descrição:

O sistema irá verificar listas de tarefas criadas anteriormente pelo usuário (caso sejam criadas anteriormente)

3. Atores:

Sistema

4. Fluxo de Eventos

##### 4.1 Fluxo Principal

1. O caso de uso começa quando o ator clica no Exibir Lista de Tarefas
2. O sistema irá verificar listas de tarefas criadas anteriormente pelo usuário.

##### 4.2. Fluxo Alternativo

FA01 – Sistema não detecta lista de tarefas criadas

1. O sistema sugere para que sejam criadas lista de tarefas para prosseguir
2. O caso de uso é encerrado.

#### UC09 – Criar Nova Lista de Tarefas

1. Nome do Caso de Uso  
Criar Nova Lista de Tarefas

## 2. Breve Descrição

O usuário poderá criar listas de tarefas ao acessar a aba de Exibir  
Lista de Tarefas

## 3. Atores:

Usuário

## 4. Fluxo de Eventos

### 4.1 Fluxo Principal

1. O caso de uso começa quando o usuário clicar em Criar Nova Lista de Tarefas
2. O sistema permitirá que o usuário escolha as configurações acessíveis a personalização da lista de tarefas
3. O sistema encerrará o caso de uso quando a lista de tarefas for criada.

### 4.2 Fluxo Alternativo

FA01 – O usuário cancela a criação de uma nova lista de tarefas

1. O sistema cancela a personalização da Nova Lista de Tarefas e será encerrado.
2. O usuário é devolvido ao caso de uso do Exibir Lista de Tarefas
3. O caso de uso é encerrado.

## UC10 – Deletar Lista Criada

### 1. Nome do Caso de Uso

Deletar Lista Criada

### 2. Breve Descrição:

O usuário poderá deletar listas de tarefas criadas ao acessar a aba de  
Exibir Lista de Tarefas

### 3. Atores:

Usuário

### 4. Fluxo de Eventos

#### 4.1 Fluxo Principal

1. O caso de uso começa quando uma lista de tarefas criada é selecionada.



2. O sistema dá a opção ao usuário de Deletar Lista de Tarefas.
3. O usuário seleciona a lista.
4. O sistema avisa que a lista será permanentemente apagada
5. O usuário confirma a seleção
6. O sistema apaga a lista.

#### 4.2 Fluxo Alternativo

FA01 - O usuário cancela a eliminação de uma lista

1. O sistema cancela o processo de eliminação da lista
2. O usuário é devolvido ao caso de uso do Exibir Lista de Tarefas
3. O caso de uso é encerrado.

### UC11 – Criar Nova Tarefa

1. Nome do Caso de Uso

Criar Nova Tarefa

2. Breve Descrição:

O usuário poderá adicionar uma nova tarefa a uma lista de tarefas já existente.

3. Atores:

Usuário

4. Fluxo de Eventos

#### 4.1 Fluxo Principal

1. O caso de uso começa quando o usuário decide criar uma nova tarefa numa lista já existente
2. O sistema abre a lista existente
3. O sistema entra no modo de personalização para adicionar a nova tarefa a lista.

#### 4.2 Fluxo Alternativo

FA01 – Tarefa já existente

1. O usuário tenta colocar uma tarefa de mesmo nome de outra já existente
2. O sistema aponta erro dizendo que "A tarefa já está na lista"

FA02 – O usuário cancela a criação de uma nova tarefa

1. O sistema cancela a criação de uma nova tarefa
2. O usuário é devolvido ao caso de uso do Exibir Lista de Tarefas
3. O caso de uso é encerrado.

UC12 – Deletar Tarefa

1. Nome do Caso de Uso

Deletar Tarefa

2. Breve Descrição:

O usuário poderá deletar tarefas criadas nas listas e tarefas ao acessar a aba de Exibir Lista de Tarefas

3. Atores:

Usuário

4. Fluxo de Eventos

#### 4.1 Fluxo Principal

1. O caso de uso começa quando uma tarefa é selecionada.
2. O sistema dá a opção ao usuário de Deletar Tarefa.
3. O usuário seleciona a Tarefa.
4. O sistema avisa que a tarefa será permanentemente apagada
5. O usuário confirma a seleção
6. O sistema apaga a tarefa.

#### 4.2 Fluxo Alternativo

FA01 - O usuário cancela a eliminação de uma tarefa

1. O sistema cancela o processo de eliminação da tarefa
2. O usuário é devolvido ao caso de uso do Exibir Lista de Tarefas.
3. O caso de uso é encerrado.

UC13 - Exibir Lembretes

1. Nome do Caso de Uso

Exibir Lembretes

2. Breve Descrição:

Ao usuário abrir o aplicativo o menu iniciar aparecerá, exibindo todos os lembretes (caso tenham sido criadas alguns).

3. Atores:

Usuário

4. Fluxo de Eventos

4.1. Fluxo Principal

1. O caso de uso começa quando o ator entra no aplicativo.
2. O sistema exibirá todos os lembretes criados pelo usuário anteriormente

4.2. Fluxo Alternativo

FA01 – O sistema não possui lembretes criados recentemente

1. O sistema não exibirá lembretes criados pelo usuário
2. O sistema sugerirá para que sejam criados lembretes.
3. O fluxo principal continua no passo 2.

UC14 - Verificar Lembretes Criados

1. Nome do Caso de Uso

Verificar Lembretes Criados

2. Breve Descrição:

O sistema irá verificar lembretes criados anteriormente pelo usuário (caso sejam criados anteriormente)

3. Atores:

Sistema

4. Fluxo de Eventos

4.1. Fluxo Principal

1. O caso de uso começa quando o ator clica no Exibir Lembretes
2. O sistema irá verificar lembretes criados anteriormente pelo usuário.

#### 4.2. Fluxo Alternativo

FA01 – Sistema não detecta lembretes criados

1. O sistema sugere para que sejam criados lembretes para prosseguir
2. O caso de uso é encerrado.

#### UC15 - Criar Novos Lembretes

1. Nome do Caso de Uso

Criar Novos Lembretes

2. Breve Descrição

O usuário poderá criar lembretes ao acessar a aba de Exibir Lembretes

3. Atores:

Usuário

4. Fluxo de Eventos

- 4.1. Fluxo Principal

1. O caso de uso começa quando o usuário clicar em Criar Novo Lembrete
2. O sistema permitirá que o usuário escolha as configurações acessíveis a personalização do lembrete
3. O sistema encerrará o caso de uso quando o lembrete for criado.

- 4.2. Fluxo Alternativo

FA01 – O usuário cancela a criação de um novo lembrete

1. O sistema cancela a personalização do Criar Novo Lembrete e será encerrado.
2. O usuário é devolvido ao caso de uso do Exibir Lembretes.
3. O caso de uso é encerrado.

#### UC16 – Deletar Lembrete Criado

1. Nome do Caso de Uso

Deletar Lembrete Criado

2. Breve Descrição:

O usuário poderá deletar lembretes criados ao acessar a aba de Exibir Lembretes

3. Atores:

Usuário

4. Fluxo de Eventos

4.1. Fluxo Principal

1. O caso de uso começa quando um lembrete é selecionado.
2. O sistema dá a opção ao usuário de Deletar Lembrete.
3. O usuário seleciona o Lembrete.
4. O sistema avisa que o Lembrete será permanentemente apagado.
5. O usuário confirma a seleção.
6. O sistema apaga o lembrete.

4.2. Fluxo Alternativo

FA01 - O usuário cancela a eliminação de um lembrete.

1. O sistema cancela o processo de eliminação do lembrete.
2. O usuário é devolvido ao caso de uso do Exibir Lembretes.
3. O caso de uso é encerrado.

## CONCLUSÃO

A conclusão deste trabalho é estabelecer a modelagem do software e sua automatização e suas atividades de manutenção de dados. Com seus fundamentos necessários em modelagem de software que serão necessários à manutenção do projeto tendo sua finalidade em sua prática oportunidade de aperfeiçoamento sendo estabelecidos através de feedback dos usuários.

Esperamos ter a oportunidade de encontrar melhorias durante o projeto, sempre atendendo conforme as necessidades estabelecidas pelo usuário a procura de identificações, melhorias ao projeto que tragam o necessário com clareza tendo a identificação mais rápida pelo usuário atendendo todas as finalidades e a todos os propósitos estabelecido usando as metodologias.

Sendo um aplicativo ao público específico como a doença de Alzheimer, suas atividades de manutenção permitem ajudar o portador e cuidador das pessoas.

Este trabalho tem como objetivo realizar um aplicativo para o Alzheimer com o princípio de ajudar as pessoas portadoras do Alzheimer que precisam de muito apoio tanto para a pessoa portadora do Alzheimer tanto para quem cuida da pessoa portadora. É uma doença que judia muito delas e este aplicativo tem o princípio de ajudar a estabelecer uma ajuda a mais para estas pessoas.

Esse é o princípio da ajuda que estamos estabelecendo com esse aplicativo, portanto é esperado que ele possa ajudar a melhorar a vida dos demais afetados, bem como auxiliar para vidas futuras e educar com atenção.

## **BIBLIOGRAFIA**

<https://encurtador.com.br/aqrx6> - Revista Engenharia de Software 10

<https://encurtador.com.br/duxN3> - 9 Principais Metodologias de Desenvolvimento de Software - CuboUP Serviços Online Ltda © 2019 – 2023  
All Rights Reserved.

<https://encurtador.com.br/cdft0> - Modelo de Caso de Uso - dtic.tjpr.jus.br

<https://encurtador.com.br/jqQR9> - Repositório de Requisitos de Casos de Uso