



---

**Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Guilherme Henrique Defavari  
Guilherme de Paula Oliveira  
Matheus Henrique Bignotto Zuculo  
Renato Marques Mendes

**Sistemas de automação residencial para cuidados da terceira idade**

**Americana, SP**  
**2022**

---

**Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”**

**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Guilherme Henrique Defavari  
Guilherme de Paula Oliveira  
Matheus Henrique Bignotto Zuculo  
Renato Marques Mendes

**Sistemas de automação residencial para cuidados da terceira idade**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Carlos Henrique Rodrigues Sarro

Área de concentração: Relatório Técnico.

Guilherme Henrique Defavari  
Guilherme de Paula Oliveira  
Matheus Henrique Bignotto Zuculo  
Renato Marques Mendes

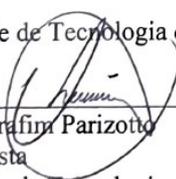
Sistemas de Automação Residencial para Cuidados para a Terceira  
Idade

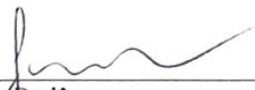
Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana – Ralph Biasi.  
Área de concentração: Desenvolvimento de Sistemas.

Americana, 12 de dezembro de 2022

**Banca Examinadora:**

  
\_\_\_\_\_  
Carlos Henrique Rodrigues Sarro  
Mestre  
Faculdade de Tecnologia de Americana Ministro Ralph Biasi

  
\_\_\_\_\_  
Lucas Serafim Parizotto  
Especialista  
Faculdade de Tecnologia de Americana Ministro Ralph Biasi

  
\_\_\_\_\_  
Jonas Bodê  
Mestre  
Faculdade de Tecnologia de Americana Ministro Ralph Biasi

## Sumário

<b>RESUMO.....</b>	<b>6</b>
<b>INTRODUÇÃO.....</b>	<b>7</b>
<b>OBJETIVO.....</b>	<b>9</b>
<b>METODOLOGIA.....</b>	<b>10</b>
<b>RESULTADOS E DISCUSSÃO .....</b>	<b>12</b>
<b>ETAPAS DE DESENVOLVIMENTO .....</b>	<b>13</b>
Entrega 1 .....	13
Entrega 2 .....	16
Entrega 3 .....	17
Entrega 4 .....	19
<b>INTERFACES DE USUÁRIO .....</b>	<b>21</b>
Código do sistema de detecção de queda .....	22
<i>Dispenser</i> de comprimido .....	23
Código do <i>dispenser</i> .....	24
Aplicativo <i>Mobile</i> .....	25
<b>LEVANTAMENTO DE REQUISITOS.....</b>	<b>26</b>
Requisitos Funcionais.....	26
Requisitos Não Funcionais .....	26
<b>MODELAGEM.....</b>	<b>27</b>
Casos de Uso .....	27
Casos de uso: aplicativo .....	28
Documentação dos Casos de Uso .....	28
Documentação dos Casos de Uso: Aplicativo .....	28
Documentação dos Casos de Uso: sistema/site .....	30
Documentação dos Casos de Uso: Análise de imagens .....	32
Documentação dos Casos de Uso: Alertas de emergência .....	33
Documentação dos Casos de Uso: <i>Dispenser</i> automático de remédios .....	34
Documentação dos Casos de Uso: Integração aplicativo. ....	35
Documentação dos Casos de Uso: Integrações futuras. ....	36
Plano de Testes .....	36
Banco de Dados.....	38
Diagrama de Entidade e Relacionamento .....	38
<b>Resultados obtidos e considerações finais .....</b>	<b>39</b>

<b>Anexo A – Código do programa em Python de detecção de quedas .....</b>	<b>40</b>
<b>Anexo B – Código do programa em C++ do <i>dispenser</i> de comprimidos.....</b>	<b>42</b>
<b>Bibliografia.....</b>	<b>50</b>

## RESUMO

O projeto visa desenvolver um software para gerir múltiplos dispositivos inteligentes na casa de pessoas de terceira idade ou para pessoa que possuem alguém tipo de cuidado especial, proporcionando uma melhor qualidade de vida para os mesmos. A tecnologia evoluiu muito e com isso, muitas coisas em nossa volta mudaram junto, desde as formas de trabalho como as atividades do dia a dia. Visando isso, decidimos buscar melhorias para a vida das pessoas em atividades ou situações que ainda possuíam carência de recursos, foi então que após perceber que as ferramentas para ajudar no cuidado de pessoas especiais são precárias, decidimos pensar em soluções que poderiam melhorar de forma significativa a vida dos mesmos.

O principal objetivo desse projeto é entregar soluções que ainda não existem no mercado, ou se existem, que possamos fazer melhorias significativas para tais, buscando integrar os recursos disponíveis com as novas ideias propostas. Visando esses objetivos, a ideia é solucionar a preocupação que as pessoas possuem com quem toma remédios diariamente com horários regrados e quem tem a necessidade de monitorar pessoas que correm risco de sofrer quedas enquanto estão sozinhas.

As ideias principais são desafiadoras e necessitam de uma atenção especial para o bom funcionamento. Os testes realizados mostraram que a chance de sucesso para atender o mercado é promissora, então com investimento e tempo para dedicação, com certeza haverá retorno satisfatório.

Palavras-Chave: Monitoramento, IoT, Gerontologia.

## INTRODUÇÃO

Existem hoje no Brasil uma grande gama de empresas e produtos que prometem deixar a sua casa “smart”, além de muitas outras empresas que prometem cuidar das pessoas mais idosas de sua família, porém com os Conceitos de Internet das coisas (ou Internet of Things, IoT, em inglês) podemos ter uma nova visão do mundo e tentar integrar esses serviços oferecidos por essas diferentes empresas em um único negócio, pois essas tecnologias vem ganhando muito espaço nos últimos anos e se desenvolvendo de maneira acelerada, logo será parte comum da rotina das pessoas. A Internet das Coisas vai facilitar, e muito, a vida das pessoas, principalmente nos aspectos de saúde e segurança.

Sabemos que boa parte dos idosos moram sozinhos, e com isso surgem algumas preocupações e cuidados por parte dos familiares. No Japão existem casas com sensores espalhados pelos cômodos que registram cada movimento que o idoso faz dentro de casa e envia essas informações diretamente para o celular de um familiar.

Estamos diante de um envelhecimento crescente da população, e uma importante parte dessa população possui uma ou mais condições crônicas. Com a utilização do monitoramento remoto de peso, pressão arterial, pulso e ECG (eletrocardiograma), os problemas podem ser detectados de maneira mais rápida e simples, sem necessitar de uma visita médica.

Para solucionar esses problemas decidimos desenvolver um software, onde estará disponível para usuários registrados ao sistema terem acesso à relatórios diários, semanais ou mensais da rotina da pessoa que será monitorada, assim como, o uso de *smart* câmaras espalhadas pela casa para detectar uma possível queda e mandar uma mensagem de alerta para algum responsável. Além de produzir um *dispenser* automático de remédios para liberar o medicamento na hora correta, poderíamos acrescentar uma Alexa ou Google Home, para alertar sobre os horários de medicamentos e outros compromissos para o idoso.

É esperado que todos os dados fiquem sincronizados entre a um aplicativo e ou site, e que possuam uma interface *user friendly*.

Atualmente, as empresas similares que encontramos que oferecem opções de “Smartificação” de domicílios ou cuidados de pessoas da terceira idade usando tecnologia na nossa região são:

- **Audiolab:** comparação com a nossa proposta, eles oferecem uma grande variedade de possibilidades para o conforto dos moradores do imóvel, abrangendo áreas como controle da iluminação de maneira automática ou por dispositivos móveis, alto-falantes por todo o imóvel além de outras coisas, porém com foco em conforto e praticidade (AUDIOLAB, 2022).
- **Koten-haus:** é uma empresa com foco em gerar uma maior economia e eficiência para casas e empresas, por meio de uso de dispositivos smart, automação residencial e painéis solares (KOSTEN-HAUS, 2022).
- **Tecnosenior:** é uma empresa que busca trazer uma melhor qualidade de vida para idosos, através de tecnologias assistivas. Tendo a facilidade e a proximidade como pilares, *dispenser* de remédios (TECNOSENIOR, 2022).
- **Helpcare:** é um serviço especializado de atendimento e ajuda imediata em casos de urgência ou emergência por meio de uso de pulseiras ou relógios com um botão de emergência e rastreadores de GPS (HELPCARE, 2022).

Levando estes aspectos em consideração, foi elaborada a Tabela 1 mostrando as principais diferenças entre as empresas citadas acima e o sistema que estamos desenvolvendo nesse trabalho desenvolvido neste trabalho.

**Tabela 1 – Comparativo de funcionalidades entre empresas.**

Funcionalidade	Audiolab	Kotenhaus	Tecnosenior	Helpcare	Nosso projeto
Login do usuário					
Monitoramento de Sinais vitais					
Perfil de saúde usuário					
Relatório em PDF					
Conectividade de múltiplos dispositivos					
Acompanhamentos 24h por dia					
Botão de socorro					
Disponível para iOS					
Disponível para Android					
F.A.Q.					

Fonte: Elaborado pelo autor (2022).

## OBJETIVO

Temos como objetivo desenvolver um sistema que detecte quedas com base de visão computacional usando python, um *dispenser* de comprimidos usando Arduino e impressão 3D e por último um aplicativo que conecte as tecnologias anteriormente citadas, caso consigamos desenvolver isso em tempo hábil tentaremos trazer mais integrações ao sistema tais como conexão a outros dispositivos *smart* que possam ter na casa, como outros tipos de sensores, *smartwatch's*, Alexa ou Google Home, além de gatilhos de emergência para trazer mais segurança a família e a pessoa que está sendo monitorada.

## METODOLOGIA

Esta seção contempla as ferramentas de programação e os conceitos necessários para o desenvolvimento do sistema:

- **Python:** É uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente, possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. (MELO, 2020)
- **OpenCV:** Originalmente, desenvolvida pela Intel, em 2000, é uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de Visão computacional, bastando seguir o modelo de licença BSD Intel. O *OpenCV* possui módulos de Processamento de Imagens e Vídeo I/O, Estrutura de dados, Álgebra Linear, GUI (Interface Gráfica do Usuário) básica com sistema de janelas independentes, Controle de mouse e teclado, além de mais de 350 algoritmos de Visão computacional como: Filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural e outros. (CEDRO TECHNOLOGIES, 2018)
- **FireBird:** É um sistema gerenciador de banco de dados. Roda em Linux, Windows, Mac OS, em grandes sistemas operacionais e uma variedade de plataformas Unix. A Fundação FirebirdSQL coordena a manutenção e desenvolvimento do Firebird, sendo que os códigos fonte são disponibilizados sob o CVS da SourceForge. (CANTU, 2010)
- **Vscode:** É um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código. Ele é customizável, permitindo que os usuários possam mudar o tema do editor, teclas de atalho e preferências. Ele é um software livre e de código aberto, apesar do download oficial estar sob uma licença proprietária. (HANASHIRO, 2021)
- **GitHub:** GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados

e/ou *Open Source* de qualquer lugar do mundo. GitHub é amplamente utilizado por programadores para divulgação de seus trabalhos ou para que outros programadores contribuam com o projeto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam repositórios remotos. (REDAÇÃO IUGU, 2021)

- **Asana:** Asana é uma plataforma de gerenciamento de trabalho móvel e web desenvolvida para ajudar as equipes a organizar, acompanhar e gerenciar seu trabalho. (KOVACS, 2020)
- **HTML:** HTML abreviação para a expressão inglesa *HyperText Markup Language*, que significa: "Linguagem de Marcação de Hipertexto" é uma linguagem de marcação utilizada na construção de páginas na Web. Documentos HTML podem ser interpretados por navegadores. A tecnologia é fruto da junção entre os padrões HyTime e SGML. (MARQUES, 2019)
- **CSS:** CSS é a linguagem para descrever a apresentação das páginas web, incluindo cores, *layout* e *fonts*. Permite que alguém adapte a apresentação das páginas para diferentes tipos de dispositivos (OKUBO, 2021)
- **Javascript:** JavaScript, frequentemente abreviado como JS, é uma linguagem de programação interpretada de alto nível, caracterizada também, como dinâmica, fracamente tipada, *prototype-based* e multi-paradigma. (ROVEDA, 2021)
- **React Native:** React Native é uma biblioteca Javascript criada pelo Facebook. É usada para desenvolver aplicativos para os sistemas Android e iOS de forma nativa. (BECKER, 2021)

## RESULTADOS E DISCUSSÃO

Decidimos adotar o scrum por ser uma metodologia ágil para gestão e planejamento de projetos de *software*, apresentada como um *framework* em que os envolvidos podem abordar e resolver problemas complexos e adaptativos, enquanto de forma produtiva e criativa, entregam produtos com o maior valor possível. (SCHWABER e SUTHERLAND, 2017). Se adaptar às necessidades do cliente é uma habilidade essencial para o sucesso final de um projeto de *software*, devido à grande variabilidade nos requisitos ao decorrer do curso do projeto.

O *Time Scrum* é constituído pelo *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*. O modelo de time no Scrum é projetado para aperfeiçoar a flexibilidade, criatividade e produtividade. (SCHWABER e SUTHERLAND, 2017).

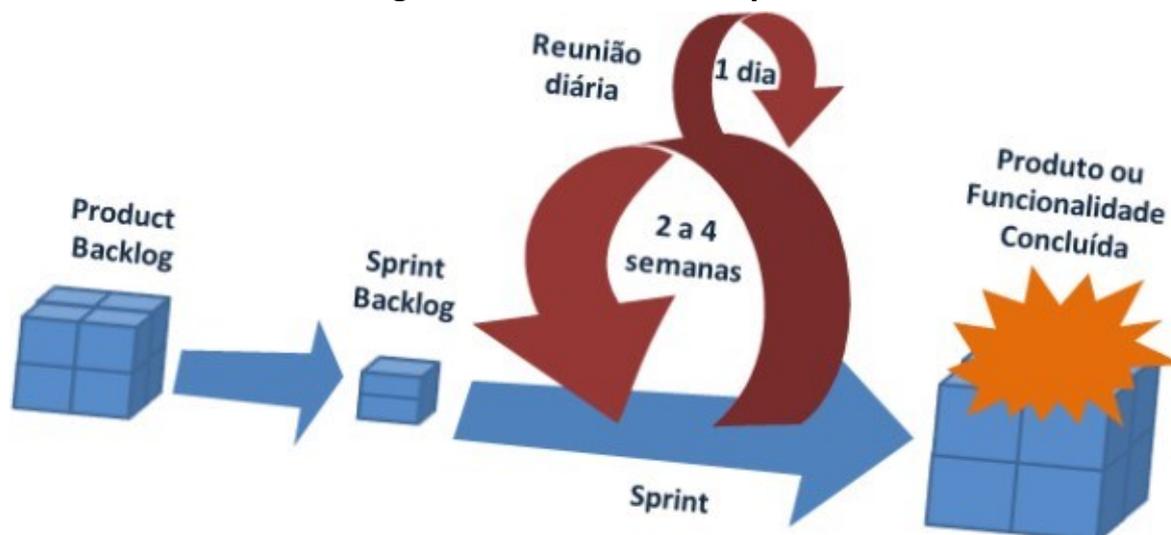
O *Product Owner*, é o responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento. O Time de Desenvolvimento consiste em profissionais que realizam o trabalho de cumprir com a entrega das funcionalidades. O *Scrum Master* é responsável por garantir que o *Scrum* seja entendido e aplicado.

No *Scrum*, os projetos são desmontados em ciclos chamados de *Sprints* que representa um espaço de tempo dentro do qual um conjunto de atividades deve ser executado. (SCHWABER e SUTHERLAND, 2017).

As funcionalidades que serão implementadas no decorrer de um projeto são mantidas em uma lista chamada *Product Backlog*. No início de cada *Sprint*, realiza-se, uma reunião de planejamento na qual o *Product Owner* especifica e prioriza os itens do *Product Backlog* enquanto a equipe seleciona as atividades que ela será capaz de implementar durante a *Sprint*. As tarefas alocadas em um *Sprint* são transferidas do *Product Backlog* para o *Sprint Backlog*.

Durante a *Sprint* a equipe faz uma breve reunião com o intuito de falar sobre o que foi feito, identificar impedimentos e priorizar o trabalho do dia que se inicia. Ao final da *Sprint*, são apresentadas as funcionalidades implementadas, e faz-se uma *Sprint Retrospective* e a equipe parte para o planejamento do próximo *Sprint*. Uma nova *Sprint* inicia imediatamente após a conclusão da *Sprint* anterior. A Figura abaixo demonstra o ciclo de uma *Sprint*.

**Figura 1 – Ciclo de uma *Sprint*.**



**Fonte: Scrum Master (2021).**

## ETAPAS DE DESENVOLVIMENTO

Para cada entrega foi convencionado realizar a divisão de tarefas de acordo com a similaridade entre elas. Foi decidido que para cada entrega teria um prazo máximo de 21 dias. Resumidamente, as entregas ocorrem da seguinte forma.

- **Entrega 1:** Início do desenvolvimento (pesquisa e desenvolvimento do sistema de queda); primeira versão do sistema, com recursos limitados.
- **Entrega 2:** Refinamento do sistema, procura de possíveis métodos para melhoria do sistema, pesquisa e início do desenvolvimento do *dispenser* de comprimidos.
- **Entrega 3:** Refinamento e *bug hunting* dos sistemas existentes e início de desenvolvimento do aplicativo e criação da identidade visual da empresa.
- **Entrega 4:** Finalização dos recursos do aplicativo, desenvolvimento do site da empresa, terminar a documentação, correção de erros em geral e aprimorar as funcionalidades desenvolvidas nas últimas entregas.

### Entrega 1

Após a formação do time de desenvolvimento e escolha do tema, o grupo através de conversas realizadas por aplicativos de mensagens definiu-se o planejamento do primeiro entregável. Através dessas mensagens os membros

definiram as atividades e seus níveis de dificuldades – representados por um esquema de pontuação. De modo geral, essas atividades estão relacionadas ao desenvolvimento inicial de um projeto de *software*, criação da documentação inicial referente ao banco de dados e das primeiras tabelas além das primeiras funcionalidades do *software*. A Tabela 1 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

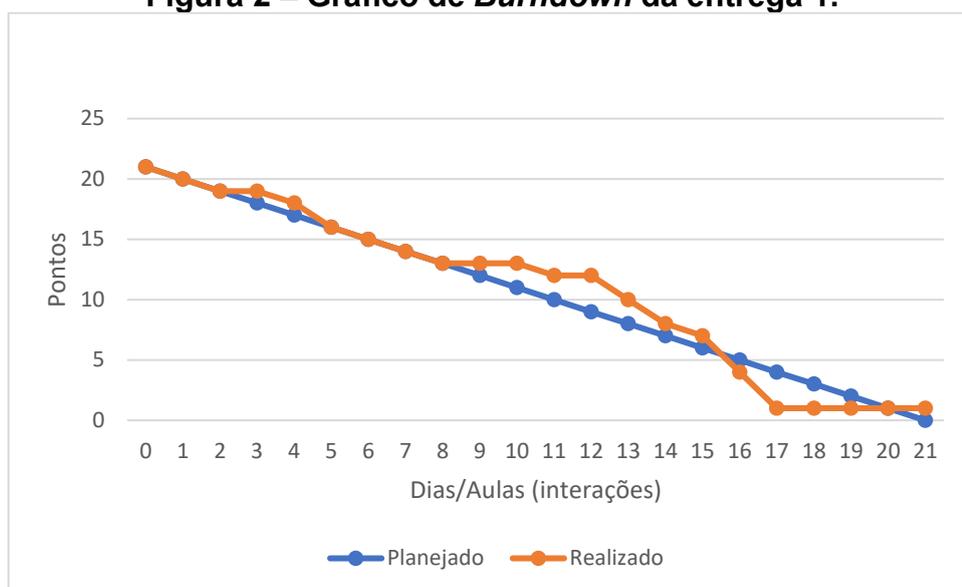
**Tabela 1 - Planejamento Sprint 1**

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Discussão do projeto	3	5
Pesquisa sobre processos de implementação	2	2
Desenvolvimento Inicial do sistema	2	3
Modelagem conceitual do banco de dados	2	3
Modelo Lógico (relacional)	2	4
Criar banco de dados	3	5
<b>Total</b>	21	45

**Fonte: Elaborado pelo autor (2022).**

Durante os 21 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 2 apresenta o gráfico de Burndown da entrega 1.

**Figura 2 – Gráfico de *Burndown* da entrega 1.**



**Fonte: Elaborado pelo autor (2022).**

Percebe-se que o desenvolvimento ficou estagnado durante a fase inicial do desenvolvimento do *software*. Essa demora para a conclusão das funcionalidades propostas se deu em virtude do processo de levantamento de requisitos e coincidência com múltiplos feriados no mês de fevereiro.

A equipe realizou a revisão das funcionalidades e apresentação do entregável, procurando identificar os principais acertos e erros, para propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao término, a equipe fez a revisão da *Sprint*, focando em alguns pontos:

- **O que deu certo:** A academia ajudou bastante no desenvolvimento da *Sprint*, muito pelo fator NECESSIDADE, o segmento do projeto estar extremamente ligado a tecnologia (aparelhos, *wearables*, *apps*) e com muitas possibilidades para criar;
- **O que deu errado:** Devido ao feriado prolongado a equipe não se comunicou bem impossibilitando uma organização eficiente, apenas um membro domina as ferramentas para o desenvolvimento *Web* acarretando sobrecarga de atividades, não foram concluídas todas as atividades propostas;
- **Ações de melhorias:** Realizar cursos em plataformas como Udemy, Alura etc.; e fazer uma melhor distribuição das atividades e tempo baseado nos cronogramas diários de cada um;
- **Observações:** Semana de Carnaval, fazer checagens semanais ou diárias e não

deixar tudo para a última semana.

## Entrega 2

Após a apresentação do entregável deu-se início a Sprint 2, o grupo através de conversas realizadas por aplicativos de mensagens e reuniões feitas durante as aulas de Laboratório de Engenharia de Software, definiu o planejamento do segundo entregável.

A prioridade era concluir as atividades que foram herdadas da Sprint 1 e realizar melhorias baseadas no feedback que foi recebido na primeira apresentação. O *redesigning* do site foi planejado e um slide introdutório pautado nos valores da escola. A Tabela 2 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

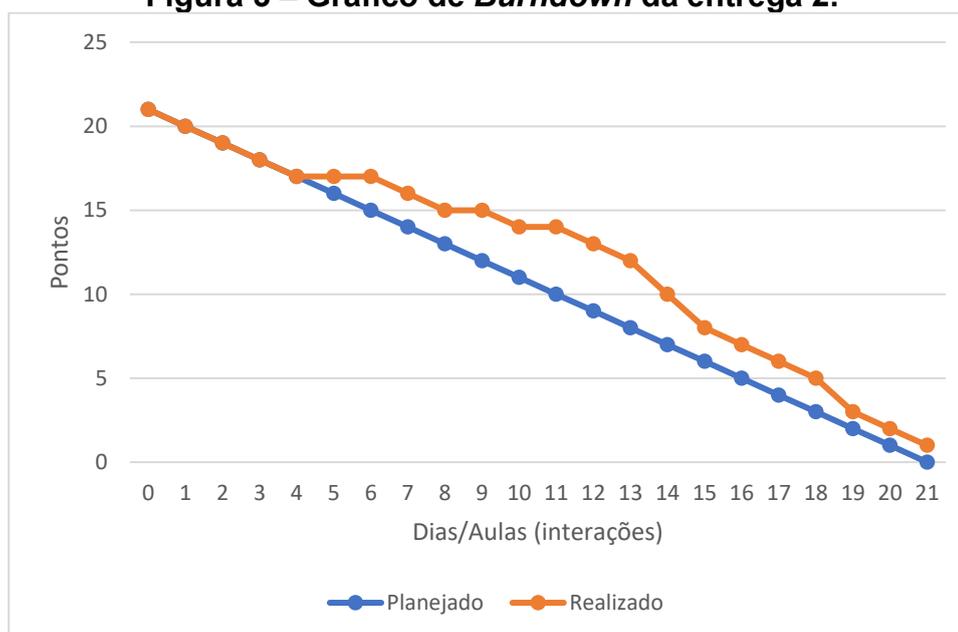
**Tabela 2 – Planejamento realizado para primeira entrega.**

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Criar Sistema de reconhecimento de queda	5	5
Desenvolver o timer para envio da notificação	5	5
Pesquisa sobre como produzir um <i>dispenser</i> automatizado	1	3
Definir o design do <i>dispenser</i>	3	4
Codificação do <i>dispenser</i>	4	5
Prototipação	1	2
Testes	2	5
<b>Total</b>	<b>20</b>	<b>29</b>

**Fonte: Elaborado pelo autor (2022).**

Durante os dias de desenvolvimento, a equipe realizava baixas na pontuação conforme as atividades eram concluídas. A Figura 3 apresenta o gráfico de *Burndown* da entrega 2.

**Figura 3 – Gráfico de *Burndown* da entrega 2.**



**Fonte: Elaborado pelo autor (2022).**

Percebe-se a melhora da equipe no desenvolvimento da Sprint 2, chegando em alguns momentos até estar adiantado, as funções especificadas pelo plano de negócio foram simples e de fácil execução, o processo de desenvolvimento do *dispenser* e decisão do design foi mais difícil do que era imaginado pelo grupo.

A equipe realizou a revisão das funcionalidades e apresentação do entregável, um membro não estava presente e pouco contribuiu com o desenvolvimento da Sprint, o processo de análise foi repetido procurando identificar o que deu certo, e o que deu errado, para que seja possível propor ações de melhorias. A equipe fez uma avaliação da Sprint, evidenciando alguns aspectos:

- **O que deu certo:** Todo e redesign foi aceito pelos membros do grupo;
- **O que deu errado:** Tivemos dificuldade em entrar em um acordo sobre qual seria o melhor design para ser implementado no *dispenser* de comprimidos, na implementação do algoritmo de *machine learning* e no sistema de queda.
- **Ações de melhorias:** Fazer mais cursos sobre *machine learning* e achar meios para melhorar a agilidade na comunicação entre todos os membros do grupo.

### Entrega 3

Se deu início a terceira *Sprint* do desenvolvimento do projeto de *software*, após a apresentação, foi realizada uma reunião entre os membros presentes, onde foi

discutido o não cumprimento das atividades, as faltas e quais ações seriam tomadas para correção futura. Posteriormente foram realizadas reuniões no teams e whatsapp para propor as atividades que seriam desenvolvidas nesse espaço de tempo. Em princípio, essas atividades estão ligadas a conseguir finalizar a programação das atividades que estão em atraso, dos passos anteriores, assim como, o início do desenvolvimento do aplicativo móvel que receberá as informações coletadas pelo sistema. A Tabela 3 exibe as atividades, seu tempo de realização em dias e a pontuação.

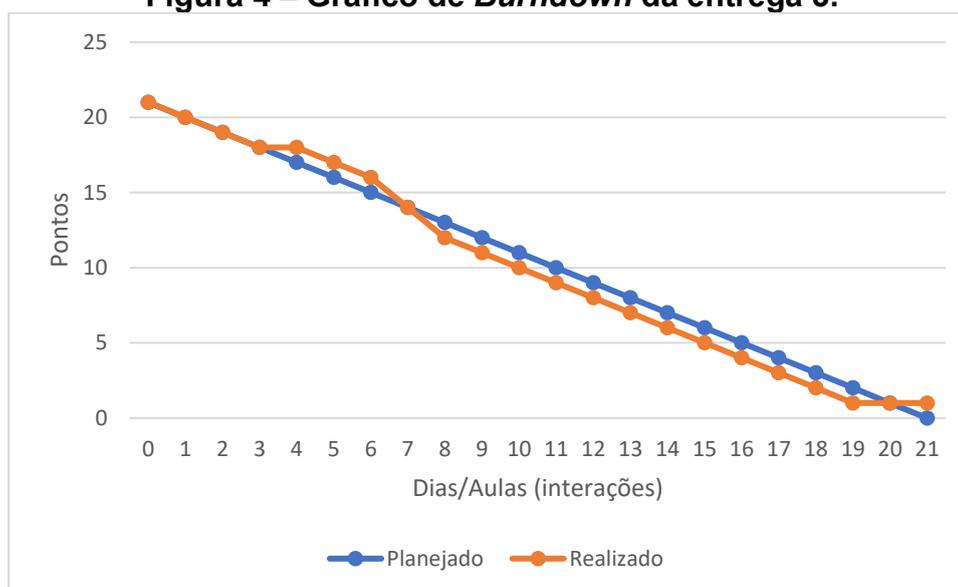
**Tabela 3 – Planejamento realizado para terceira entrega.**

<b>Atividade</b>	<b>Tempo (em dias)</b>	<b>Pontos</b>
Programação do sistema em atraso	10	10
Decisão do design do aplicativo móvel	2	5
Desenvolvimento do aplicativo móvel	5	5
Testes e <i>bugbounty</i> de todos os sistemas	4	4
<b>Total</b>	<b>21</b>	<b>24</b>

**Fonte: Elaborado pelo autor (2022).**

A Figura 4 apresenta o gráfico de *Burndown* da entrega 3. Durante os dias de desenvolvimento, a equipe realizou abatimentos na pontuação conforme as atividades eram concluídas.

**Figura 4 – Gráfico de *Burndown* da entrega 3.**



**Fonte: Elaborado pelo autor (2022).**

Nota-se que pela primeira vez todas as atividades propostas foram cumpridas, nessa entrega, todos do time de desenvolvimento trabalharam melhor e se comunicaram efetivamente, o que resultou em avanços significativos na construção do site e do aplicativo, as medidas tomadas na reunião se mostraram satisfatórias.

- **O que deu certo:** A parte do *back-end* do projeto avançou, o grupo trabalhou melhor e cumpriu tudo o que havia planejado com resultados satisfatórios;
- **O que deu errado:** O problema do número de faltas de alguns integrantes perdurou por mais uma *Sprint*;
- **Observações:** Arrumar fontes, integrantes devem estar atentos com o número máximo de faltas permitidas.

#### Entrega 4

Começou a quinta e última entrega, essa *Sprint* é a maior de todas na questão de tempo disponível, devido ao tempo de término da matéria de Laboratório de Engenharia de *Software*. Para essa entrega foi definida a importância de checar o que já havia sido feito, terminar a extensa documentação proposta (Documentação do *software* somado ao plano de negócios), continuar o desenvolvimento dos sistemas em adição de começar a fazer a integração final entre todas as partes. Além de se organizar para a apresentação do sistema e seu modelo de negócio. A Tabela 4

apresenta e exibe as atividades, seu tempo de realização em dias e a pontuação.

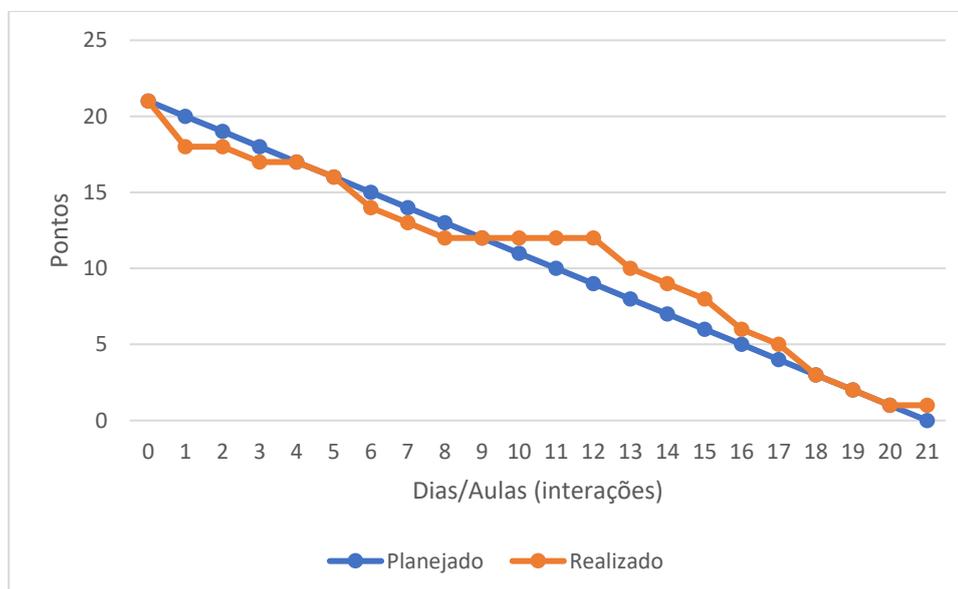
**Tabela 4 – Planejamento realizado para terceira entrega.**

Atividade	Tempo (em dias)	Pontos
Terminar plano de negócio	5	5
Concluir todos os diagramas pedidos	3	4
Terminar a documentação do Software	11	10
Fazer as checagens das funcionalidades implantadas	1	5
Terminar a apresentação final	1	10
<b>Total</b>	<b>21</b>	<b>34</b>

Fonte: Elaborado pelo autor (2022).

A Figura 5 apresenta o gráfico de *Burndown* da entrega final.

**Figura 5 – Gráfico de *Burndown* da entrega 4.**



Fonte: Elaborado pelo autor (2022).

Percebe-se que o grupo não trabalhou bem nos primeiros dias devido ao número de exigências de outras matérias do curso de Análise e Desenvolvimento de Sistemas, as atividades propostas nessa *Sprint* tinham alto nível de complexidade e demandavam muito tempo para serem realizadas, entretanto, todas as obrigações do

grupo foram cumpridas no tempo estipulado. Como retrospectiva desse último entregável vale ressaltar:

- **O que deu certo:** Toda a documentação e as funções propostas do sistema e aplicativo foram feitas;
- **O que deu errado:** A organização nessa *Sprint* maior não seguiu as boas práticas das duas últimas, os dias adicionais passaram uma falsa ideia de tranquilidade;
- **Ações de melhorias:** Procurar manter a comunicação e organização das *Sprints* anteriores e realizar mais reuniões com a presença do proprietário da academia;
- **Observações:** Documentação extensa e problemas de compatibilidade com os horários de Trabalho de Graduação I.

## INTERFACES DE USUÁRIO

As figuras a seguir representam algumas telas e funcionalidades do sistema:

Detecção de queda

**Figura 6 – Detecção de queda.**



**Fonte: Elaborado pelo autor (2022).**

A detecção de queda é uma das principais inovações que o sistema traz ao mercado, pois ele implementa uma nova layer de funcionalidade para as câmeras de segurança que tem na casa de pessoas de terceira idade, pois permitirá que

implementemos uma funcionalidade de alertar os familiares caso o monitorado sofra uma queda.

Código do sistema de detecção de queda

**Figura 7 – Código do sistema de detecção de queda**

```

Projeto_Care > Alerta.queda > Queda3.py > ...
1  import cv2
2  import time
3  from email import send
4
5  webcam = cv2.VideoCapture(1)
6  analisador = cv2.CascadeClassifier(
7      'C:/Users/T-Gamer/Desktop/Projeto_Care/Alerta.queda/pedestrian.xml')
8  file = "C:/Users/T-Gamer/Desktop/Projeto_Care/Alerta.queda/Imagens/Queda1.png"
9
10
11 while True:
12     camera, frame = webcam.read()
13     cinza = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
14     detecta = analisador.detectMultiScale(cinza, 1.08, 3)
15     for (x, y, l, a) in detecta:
16         cv2.rectangle(frame, (x, y), (x+l, y+a), (0, 255, 0), 2, 2)
17
18     if l > a:
19         q += 1
20
21     while q > 10:
22         cv2.rectangle(frame, (x, y), (x+l, y+a), (0, 0, 255), 2, 2)
23         start = time.time()

```

**Fonte: Elaborado pelo autor (2022).**

Essa imagem é uma representação de uma pequena parte do código, onde podemos ver a configuração inicial para o código do *python* para detectar a câmera. Três das principais bibliotecas que utilizamos nessa versão do *software* são, *Opencv*, que é uma biblioteca pública de análise de imagens para *Python*, *C*, *C++* e *Java*, a biblioteca *time*, que utilizamos até o momento como uma maneira de diferenciar a pessoa deitada em relação a queda, partindo da lógica de que se o período de queda for menor que um tempo X (25 segundos até o momento no código) a pessoa só está agachada, se o tempo for maior que o mesmo X é considerado como queda, será gerado um *log* para o relatório. Caso o tempo seja maior que Y (40 segundos até o momento no código) será gerado um alerta no aplicativo do responsável da pessoa

monitorada, e a última biblioteca, que é a do *E-mail*, para que seja enviado um *e-mail* para o responsável notificando o alerta. Até o presente momento não temos uma função de alertar a queda, pelo aplicativo, implementada.

*Dispenser* de comprimido

**Figura 8 – *Dispenser* de comprimido**



**Fonte: Elaborado pelo autor (2022).**

O dispenser automático de comprimidos foi desenvolvido com base no projeto *open source* existente na internet (QUIRINO, 2022), com o objetivo de ajudar as pessoas a tomar o remédio nos horários corretos e evitar que a pessoa tome muitos comprimidos de uma única vez. O dispositivo foi desenvolvido usando impressora 3D e a parte eletrônica foi desenvolvida com base na utilização de um Arduino para controlar todo o funcionamento do dispositivo.

Código do *dispenser*Figura 9 – Código do *dispenser* de comprimido

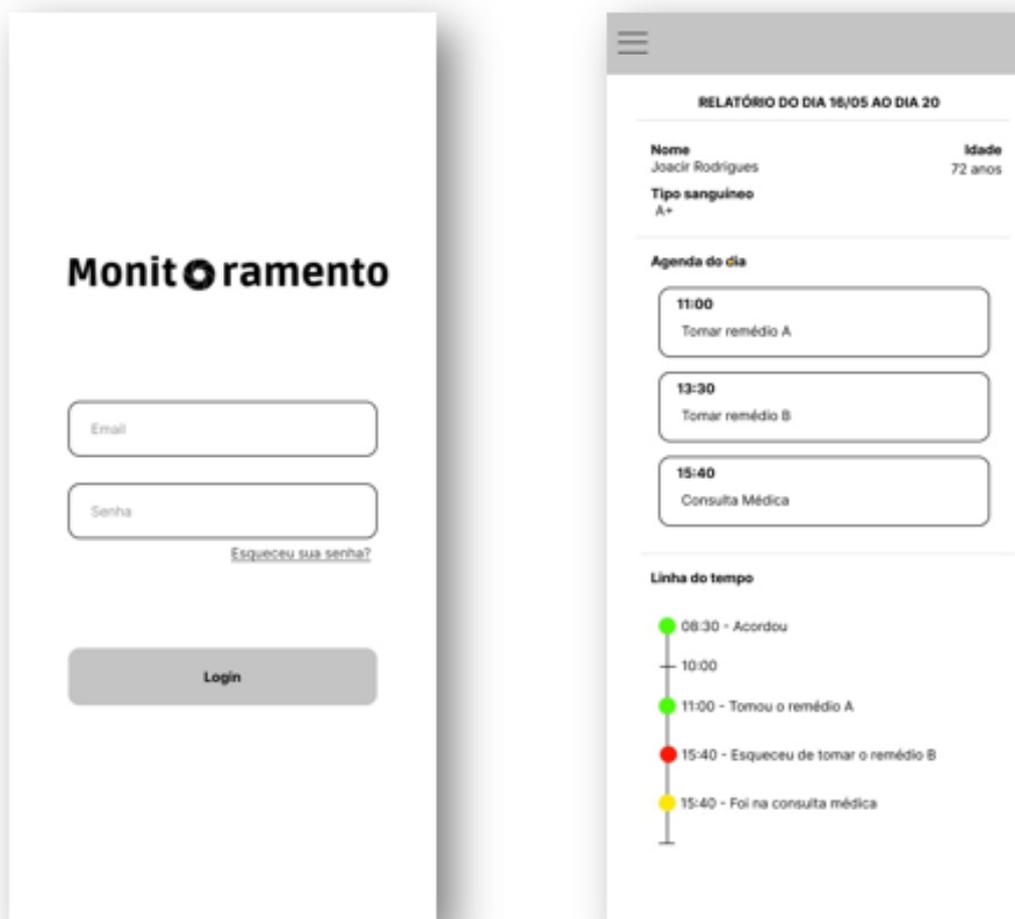
```

Projeto_Care > Dispenser_Comprimido > Arduino_Programming > med_pill_version_1.3.ino > Task1code(void *)
138 void Task1code( void * pvParameters ){
139     Serial.print("Task1 running on core ");
140     Serial.println(xPortGetCoreID());
141
142     for(;;){
143         // hora = timeClient.getFormattedTime();
144         delay(1000);
145         timeClient.update(); // Update ntp a cada 60 segundos
146         if (millis() - last >=2000 && x < 5)
147         {
148             last = millis();
149             i++;
150             if (x == 5){
151                 Serial.printf("ok 3s");
152             }
153         }
154         unsigned long currentMillis = millis(); // Verifica conexão WiFi e Reconecta cada ha
155         // if WiFi is down, try; // reconnecting every CHECK_WIFI_TIME seconds
156         if ((WiFi.status() != WL_CONNECTED) && (currentMillis - previousMillis >=interval))
157             // Lgbq
158             Serial.print(millis());
159             Serial.println("Reconnecting to WiFi...");
160             WiFi.disconnect();

```

Fonte: Elaborado pelo autor (2022).

Essa imagem mostra um pequeno trecho do código do *dispenser*, no qual está sendo executado a função de conectar o *dispenser* a internet para que haja a sincronização do relógio.

Figura 10 – Aplicativo *Mobile*

Fonte: Elaborado pelo autor (2022).

O aplicativo *mobile* é uma das principais partes do sistema, pois nele será inserido algumas informações de controle do *dispenser* de comprimidos e poderá consultar relatórios das informações coletadas pelos sensores da casa, além de receber os alertas de emergência que poderão ser gerados pelo sistema.

## LEVANTAMENTO DE REQUISITOS

A engenharia de requisitos (RE – Requirements Engineering) é o processo de descobrir, analisar, documentar e verificar requisitos de um sistema. O procedimento de levantamento de requisitos, quando bem elaborado, garante qualidade, eficiência, confiabilidade e integridade ao software desenvolvido.

Um requisito pode ser definido como a descrição das características do sistema, tal como suas restrições operacionais ou serviços fornecidos pelo mesmo. É uma condição ou capacidade que deve ser alcançada ou estar presente em um sistema para satisfazer um contrato, padrão ou especificação. Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

### Requisitos Funcionais

Os requisitos funcionais descrevem as funções que o sistema deve executar, isto é, determinam a funcionalidade esperada do software (SOMMERVILLE, 2011). A Tabela 5 apresenta os requisitos funcionais deste projeto.

**Tabela 5 – Requisitos funcionais do projeto.**

Identificação	Requisito Funcional	Prioridade
RF001	Acesso a dispositivos <i>smart</i> da casa	Importante
RF002	Conectividade com a internet (para envios de relatórios)	Essencial
RF003	Monitoramento de sinais vitais por meio de <i>smartwatch</i>	Importante
RF004	Monitoramento das horas de sono e possíveis quedas por câmeras inteligentes	Essencial
RF005	Uso de visão computacional para reconhecimento de padrões e rotinas	Essencial
RF006	Notificações dos horários de remédios e outros compromissos	Desejável
RF007	Notificações de emergências em aplicativo para IOS e Android	Importante
RF008	<i>Download</i> de relatórios periódicos da pessoa monitorada via aplicativo ou <i>website</i>	Essencial
RF009	Mensagens de erro caso falha do sistema	Desejável

**Fonte: Elaborado pelo autor (2022).**

### Requisitos Não Funcionais

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2011). São requisitos que pertencem ao processo de desenvolvimento como custo, performance, segurança, confiabilidade, manutenibilidade, portabilidade, custos operacionais entre outros. A Tabela 6 apresenta os requisitos não funcionais deste projeto.

**Tabela 6 – Requisitos não funcionais do projeto.**

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	Login e Senha	Software	Importante
RNF002	Fluidez	Desempenho	Essencial
RNF003	Responsividade	Usabilidade	Essencial
RNF004	FAQ	Padrões	Desejável
RNF005	Design	Usabilidade	Desejável
RNF006	Multiplataforma	Hardware e Software	Essencial

**Fonte: Elaborado pelo autor (2022).**

## MODELAGEM

Na fase da modelagem é feita a documentação do aplicativo, trata-se de diagramas que facilitam na compreensão do projeto de forma padronizada.

A documentação deste trabalho utilizará a linguagem de modelagem *Unified Modeling Language*<sup>1</sup> (UML) para modelar os casos de uso e o diagrama de classe

### Casos de Uso

Os diagramas de caso de uso descrevem um cenário de funcionalidades do ponto de vista do usuário, catalogando os requisitos funcionais do sistema. Dentro do diagrama são retratados os atores (representado pelos bonecos), as funcionalidades (representadas pelos balões com a ação escrita por dentro) e as relações (representadas pelas linhas).

Serão divididos entre os diagramas do sistema/site e aqueles do aplicativo.

---

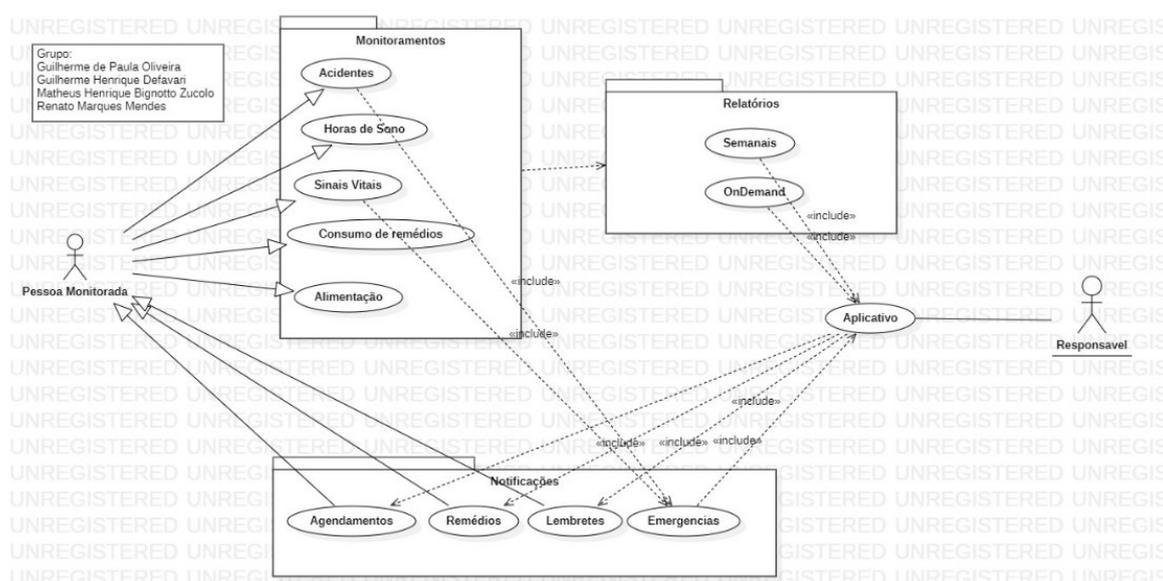
<sup>1</sup> *Unified Modeling Language* ou Linguagem Unificada de Modelagem (UML) é uma linguagem padrão para modelagem e documentar os sistemas orientados a objetos

Casos de uso: aplicativo

Os atores que interagem com o aplicativo são:

- **Usuário** é o ator que representa os utilizadores do aplicativo mobile. Ele pode, por exemplo, fazer login, inserir medidas corporais, acessar um programa de treino, acessar dados sobre a matrícula, entre outros.
- **Responsável** é o profissional responsável pelo gerenciamento do conteúdo disponível no aplicativo, gerenciamento que é feito através do sistema. Ele, por exemplo. A Figura 11 apresenta o caso de uso do aplicativo.

**Figura 11 – Diagrama de caso de uso do aplicativo.**



**Fonte: Elaborado pelo autor (2022).**

### Documentação dos Casos de Uso

A seguir serão divididos entre documentação dos Casos de Uso do aplicativo e documentação dos Casos de Uso do sistema. O sistema será feito para uso em dispositivos moveis para recebimento das informações da pessoa monitorada.

### Documentação dos Casos de Uso: Aplicativo

Cada funcionalidade dos diagramas de casos de uso do aplicativo será descrita na Tabela 6 à Tabela 7.

Tabela 6 – Caso de uso do aplicativo: fazer login.

<b>Nome do caso de uso</b>	Fazer login.
<b>Atores envolvidos</b>	Usuário.
<b>Objetivo</b>	Este caso de uso descreve o procedimento para o usuário do aplicativo efetuar o <i>login</i> .
<b>Prioridade de desenvolvimento</b>	Importante
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. O usuário clica em <i>login</i> .	1.A. Será solicitado preenchimento das credenciais.
2. O usuário insere o <i>user</i> e senha.	2.A. O aplicativo verificará se as credenciais foram preenchidas corretamente.
3. O usuário clica em Logar.	3.A. O sistema valida o <i>login</i> e caso for valido o usuário é direcionado para dentro do sistema.
3.2 Digita novamente.	3.B Mensagem de erro.
3.1 Clica em recuperar senha.	3.C O sistema envia um <i>e-mail</i> para recuperação da senha.
3.3 cadastrar nova senha.	3.D O sistema altera a senha do usuário.
3.4 O usuário insere o <i>user</i> e senha novamente.	3.E O sistema valida o <i>login</i> e caso for valido o usuário é direcionado para dentro do sistema.
4. Inicia sessão.	
<b>Validações</b>	Para que o <i>login</i> seja efetuado, o usuário deve entrar com seu usuário e senha providos pelo provedor do sistema.

Fonte: Elaborado pelo autor (2022).

Tabela 7 – Caso de uso do aplicativo: acessar dados sobre o relatório.

<b>Nome do caso de uso</b>	Acessar dados sobre o relatório.
<b>Atores envolvidos</b>	Usuário.
<b>Objetivo</b>	Este caso de uso descreve o procedimento para o usuário do aplicativo acessar dados da pessoa monitorada.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. O usuário clica em relatórios.	1. Sistema exibe a lista de relatórios.
2. Cliente escolhe o filtro dos relatórios.	2. O sistema carrega as informações referentes aos filtros.

3. O usuário clica em um dos relatórios.	3. O sistema exibe o relatório selecionado.
4. O usuário tem acesso as informações do relatório selecionado.	
<b>Validações</b>	Para que o usuário acesse os dados ele deve estar logado.

**Fonte: Elaborado pelo autor (2022).**

Documentação dos Casos de Uso: sistema/site

Cada funcionalidade dos diagramas de casos de uso do sistema/site será descrita da Tabela 8 à Tabela 10.

**Tabela 8 – Caso de uso do sistema/site: fazer *login*.**

<b>Nome do caso de uso</b>	Fazer <i>login</i> .
<b>Atores envolvidos</b>	Usuário.
<b>Objetivo</b>	Este caso de uso descreve o procedimento para algum usuário estar fazendo o <i>login</i> no site/sistema.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. O Usuário clica em <i>login</i> administrativo.	
2. O usuário insere o <i>user</i> e senha.	
	3.A. O sistema valida o <i>login</i> .
4. Inicia sessão.	
	3.B. Dados incorretos (mostra mensagem de erro).
<b>Validações</b>	Para que o <i>login</i> seja efetuado, a escola deve entrar com seu usuário e senha providos pelo <i>staff</i> que administra o sistema.

**Fonte: Elaborado pelo autor (2022).**

**Tabela 9 – Caso de uso do sistema/site: acessar formas de contato com a empresa.**

<b>Nome do caso de uso</b>	Acessar formas de contato com a empresa.
<b>Atores envolvidos</b>	Usuário, empresa.
<b>Objetivo</b>	Este caso de uso descreve o procedimento para um visitante do site ter acesso às formas de contato com a empresa, que as fornece através do sistema.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. O usuário vai até a parte do site onde ficam expostos os dados para contato.	
	2. O sistema mostra os dados que a empresa forneceu e deixou gravado no sistema.
3. O usuário tem acesso às informações.	
<b>Validações</b>	Para que o acesso seja feito, basta que o usuário tenha acesso à internet.

**Fonte: Elaborado pelo autor (2022).**

**Tabela 10 – Caso de uso do sistema/site: acessar relatórios financeiros**

<b>Nome do caso de uso</b>	Acessar relatórios financeiros.
<b>Atores envolvidos</b>	Usuário.
<b>Objetivo</b>	Este caso de uso descreve o procedimento para o usuário acessar relatórios financeiros gerados pelo sistema.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. O usuário entra na opção relatórios financeiros.	
	2. O sistema mostra os débitos em aberto e futuros boletos a serem pagos pelo próprio usuário.
3. O Usuário tem acesso às informações.	
<b>Validações</b>	Para que o acesso seja feito, o usuário deve fazer o <i>login</i> primeiro.

**Fonte: Elaborado pelo autor (2022).**

## Documentação dos Casos de Uso: Análise de imagens

Cada funcionalidade dos diagramas de casos de uso da análise de imagens será descrita da Tabela 11 à Tabela 13.

**Tabela 11 – Caso de uso do sistema: Análise de imagens.**

<b>Nome do caso de uso</b>	Acessar as câmeras da casa para fazer avaliações de situações específicas pré-programadas descritas abaixo
<b>Atores envolvidos</b>	Empresa
<b>Objetivo</b>	Este caso de uso descreve o procedimento para o sistema acessar as câmeras para executar o monitoramento
<b>Prioridade de desenvolvimento</b>	Essencial
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. A empresa faz a configuração do sistema para ela localizar as câmeras	
	2. O sistema acessa as câmeras para começar a executar o monitoramento
<b>Validações</b>	O técnico fazendo a configuração para exibir a imagem da câmera momentaneamente

**Fonte: Elaborado pelo autor (2022).**

**Tabela 12 – Caso de uso do sistema: Análise de imagens/queda.**

<b>Nome do caso de uso</b>	Acessar as câmeras da casa para fazer avaliações de situações específicas de queda.
<b>Atores envolvidos</b>	Empresa.
<b>Objetivo</b>	Este caso de uso descreve o procedimento que é executado para avaliar a queda de uma pessoa.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. A empresa faz a configuração do sistema para ela localizar as câmeras.	
	2. O sistema acessa as câmeras para começar a executar o monitoramento.
	3. É gerado um registro no sistema.

<b>Validações</b>	É executado uma sequência de vídeos contendo múltiplas quedas para verificar se o sistema está realmente reconhecendo quedas para emitir alertas.
-------------------	---

**Fonte: Elaborado pelo autor (2022).**

**Tabela 13 – Caso de uso do sistema: Análise de imagens/Horas de sono.**

<b>Nome do caso de uso</b>	Acessar as câmeras da casa para fazer avaliações de situações específicas como horas de sono.
<b>Atores envolvidos</b>	Empresa.
<b>Objetivo</b>	Este caso de uso descreve o procedimento que é executado para avaliar a queda de uma pessoa.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. A empresa faz a configuração do sistema no quarto para quando a pessoa monitorada estiver deitada em uma área predeterminada será considerado como ela estar dormindo ao invés de queda.	
	2. O sistema acessa as câmeras para começar a executar o monitoramento.
<b>Validações</b>	É executado uma sequência de vídeos contendo múltiplas quedas para verificar se o sistema está realmente reconhecendo quedas para emitir alerta.

**Fonte: Elaborado pelo autor (2022).**

Documentação dos Casos de Uso: Alertas de emergência

A funcionalidade do diagrama de casos de uso sobre alertas de emergência será descrita na Tabela 14.

**Tabela 14 – Caso de uso do sistema: Alertas de emergência.**

<b>Nome do caso de uso</b>	Caso o sistema de monitoramento detecte uma situação de queda.
<b>Atores envolvidos</b>	Sistema.
<b>Objetivo</b>	Este caso de uso descreve o procedimento que é executado pelo sistema caso ele detecte uma queda.
<b>Prioridade de desenvolvimento</b>	Essencial.

<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. O sistema visualiza a queda da pessoa monitorada, e a mesma permanece caída por um período de tempo maior que 25 segundos.	
	2. O sistema emite uma mensagem de emergência para o aplicativo vinculado a conta informando da queda e que o responsável precisa verificar o acontecido.
	3. é gerado um registo no sistema do acidente do acidente.
<b>Validações</b>	É executado um teste no dia da instalação para verificar o funcionamento do mesmo.

**Fonte: Elaborado pelo autor (2022).**

Documentação dos Casos de Uso: *Dispenser* automático de remédios.

A funcionalidade do diagrama de casos de uso sobre o dispenser automático de remédios será descrita na Tabela 15.

**Tabela 15 - Caso de uso do sistema: *Dispenser* automático de remédios.**

<b>Nome do caso de uso</b>	Funcionamento do <i>dispenser</i> de remédio.
<b>Atores envolvidos</b>	Empresa, usuário.
<b>Objetivo</b>	Este caso de uso descreve o funcionamento do <i>dispenser</i> automático de remédios fornecido pela empresa.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. A empresa fornece um <i>dispenser</i> automático de comprimidos.	
2. o usuário faz a instalação na tomada do dispositivo e coloca os comprimidos que devem ser tomados em horários específicos do dia em compartimentos individualizados.	
3. o usuário usa o aplicativo para configurar o horário que o remédio será liberado.	
	4. No horário pré-programado o equipamento fara a liberação do comprimido e tocara um alarme (futuramente o alarme poderá ser integrado a uma Alexa ou Google Home).

	5. é gerado um registo no sistema da liberação e retida dos remédios.
<b>Validações</b>	É executado um teste de liberação de comprimido no local para verificar.

**Fonte: Elaborado pelo autor (2022).**

Documentação dos Casos de Uso: Integração aplicativo.

A funcionalidade do diagrama de casos de uso sobre a integração aplicativo será descrita na Tabela 16.

**Tabela 16 – Caso de uso do sistema: Integração aplicativo.**

<b>Nome do caso de uso</b>	Discrição das funções que serão executadas pelo aplicativo.
<b>Atores envolvidos</b>	Empresa, usuário.
<b>Objetivo</b>	Este caso de as funcionalidades do aplicativo.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. A empresa fornece um aplicativo para controlar os alertas, relatórios e agendamentos.	
	2. O usuário efetua o login com os dados fornecidos pela empresa.
3. O Usuário Solicita os dados de relatório coletados pelo sistema do dia, semana ou mês.	3. O sistema retorna o relatório do período selecionado.
	4. Caso haja algum alerta de queda será encaminhado apresentado uma notificação no celular do usuário.
5. O Usuário Cadastra os horários em que o remédio será liberado.	5. O <i>Dispenser</i> atualiza para liberar os comprimidos no horário programado.
<b>Validações</b>	Será um simples teste de conectividade entre o dispositivo e a internet e <i>dispenser</i> .

**Fonte: Elaborado pelo autor (2022).**

Documentação dos Casos de Uso: Integrações futuras.

A funcionalidade do diagrama de casos de uso sobre as integrações futuras será descrita na Tabela 17.

**Tabela 17 – Caso de uso do sistema: Integrações futuras.**

<b>Nome do caso de uso</b>	Descrição de integrações futuras a ainda serem desenvolvidas e implementadas ao sistema.
<b>Atores envolvidos</b>	Empresa.
<b>Objetivo</b>	Este caso de uso descreve as partes em desenvolvimento para integração do sistema.
<b>Prioridade de desenvolvimento</b>	Essencial.
<b>Ações em desenvolvimento</b>	<b>Ações do Sistema executara.</b>
1. Integração com <i>smartwatch</i> 's.	1. O sistema coletará dados de um <i>smartwatch</i> vinculado para coletar dados básicos de saúde do usuário.
2. Integrar o sistema com Alexa e Google Home.	2. O sistema poderá apresentar notificações de consultas médicas e outros agendamentos, além de poder tocar alarmes personalizados para o usuário final e todas as outras comodidades trazidas por um <i>home assistance</i> .
3. Integração com outros dispositivos <i>smart</i> .	3. Trazer uma integração com outros dispositivos de monitoramento de <i>smarthome</i> , como detectores de presença, medidores de consumo inteligente entre outras coisas.

**Fonte: Elaborado pelo autor (2022).**

### Plano de Testes

Para obter um sistema com garantia de integridade e a prova de erros, vem a necessidade de implementar testes, de modo a validar todo e qualquer erro que possa ser caracterizado como uma falta de validação por parte dos desenvolvedores dele. Como o projeto tem um diagrama de caso de uso definido, a implementação segue de acordo com cada caso de uso, validando todos as ações que causam possíveis falhas em cada caso. A seguir estão as Tabelas 18 e 19 com o plano de testes de seus respectivos casos de uso:

Tabela 18 - Caso de teste Login

Caso de Teste – Login	
Não informar o e-mail e senha	É exibido o alerta "Entre com o seu e-mail e senha."
Informar somente e-mail	É exibido o alerta "O campo Senha é obrigatório."
Informar somente senha	É exibido o alerta "O campo E-mail é obrigatório."
Informar e-mail inválido	É exibido o alerta "Digite um endereço de e-mail válido."
<b>Login</b> Informar senha inválida	É exibido o alerta "A senha deve ter pelo menos 6 caracteres."
Verificar atribuições do usuário (teste de segurança)	No momento em que ocorrer o <i>login</i> , apenas as funcionalidades com permissões atribuídas devem ser exibidas.
<i>Login</i> com mesmo usuário em diferentes dispositivos	É exibido um alerta dizendo que a sessão do outro usuário no devido dispositivo, terá a sessão encerrada automaticamente.
<i>Login</i> sendo executado	Todos os componentes devem estar desabilitados para impossibilitar a navegação afim de evitar erros fatais.

Fonte: Elaborado pelo autor (2022).

Tabela 19 - Caso de teste Recuperar Senha

Caso de Teste – Recuperar Senha		
<b>Recuperação de Senha</b>	Não informar e-mail.	É exibido o alerta "O campo E-mail é obrigatório."
	Informar e-mail inválido	É exibido o alerta "Digite um endereço de e-mail válido."
	Falha ao recuperarsenha	É exibido o alerta "Falha ao recuperar a senha."
	Sucesso ao recuperarsenha	É exibido o alerta "Verifique seu e-mail para recuperar sua senha."
	Recuperação de senha sendo executada	Todos os componentes devem estar desabilitados para impossibilitar a navegação afim de evitar erros fatais.

Fonte: Elaborado pelo autor (2022).

## Banco de Dados

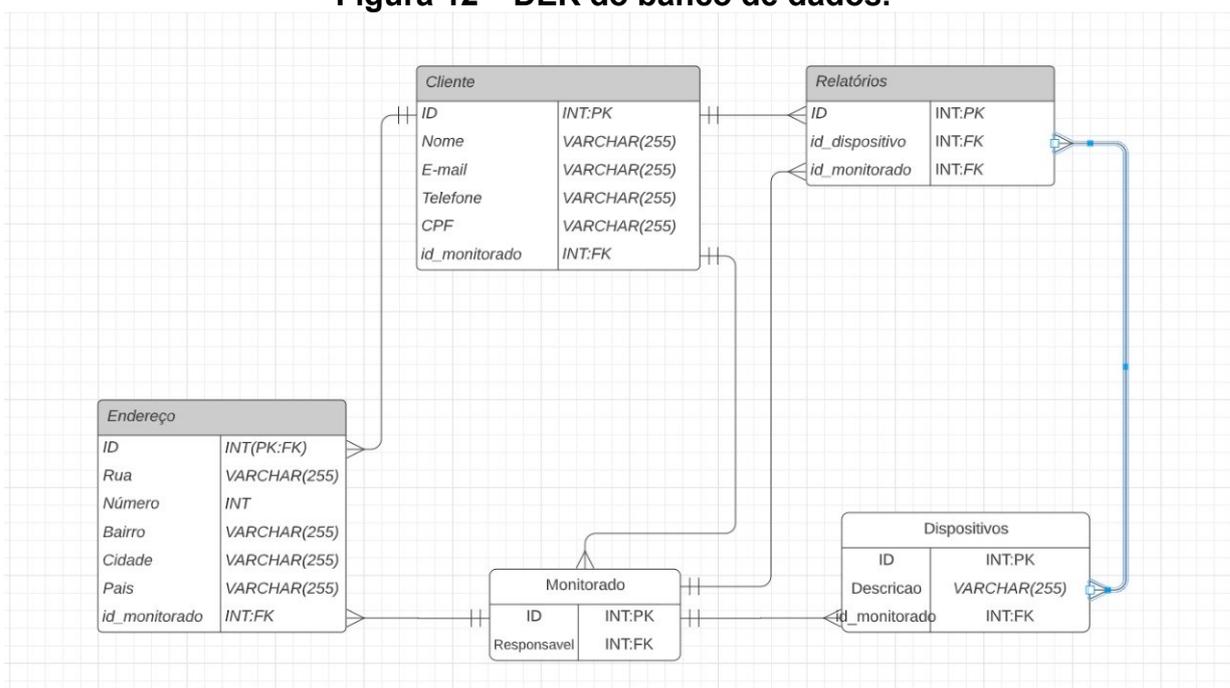
Um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”, ou seja, informações agrupadas que se relacionam (SILBERSCHATZ, KORTH e SUDARSHAN, 2006).

## Diagrama de Entidade e Relacionamento

Diagrama Entidade Relacionamento (DER) é um modelo que descreve o modelo de dados de um sistema. É a principal representação do Modelo de Entidades e Relacionamentos. Sua maior aplicação é para a visualização do relacionamento entre tabelas de um banco de dados.

A Figura 12 mostra o Diagrama de Entidade e Relacionamento do banco de dados do projeto.

**Figura 12 – DER do banco de dados.**



**Fonte: Elaborado pelo autor (2022).**

## Resultados obtidos e considerações finais

Este trabalho teve como objetivo final detalhar os processos de desenvolvimento e documentação de um sistema para o cuidado de pessoas da terceira idade. Composto por um aplicativo, um dispenser de remédios e um sistema *back-end* para o monitoramento de câmeras. O sistema encontra-se dentro de um software que será executado no computador do usuário final.

O trabalho inicialmente apresentou o conceito do sistema, ou seja, os problemas que existem, que motivaram o desenvolvimento do mesmo. O funcionamento do sistema de maneira geral, apresentamos os concorrentes comparando as funções que eles oferecem entre si e qual metodologia foi usada no desenvolvimento.

Na sequência, apresentou-se todo o processo de desenvolvimento do aplicativo e do programa, começando pela listagem das ferramentas e recursos que usamos para desenvolver até o momento. Em seguida foi detalhada a parte de modelagem, ou seja, os diagramas UML feitos, incluindo Diagramas de Casos de Uso e sua documentação, Diagramas de Classe, Atividades, Sequência, Diagrama de Entidade e Relacionamento e por fim o dicionário de dados. Detalhamos o desenvolvimento, entrando em detalhes sobre a metodologia ágil Scrum. Depois foram detalhadas as etapas de desenvolvimento e o que foi entregue nas cinco entregas periódicas.

No desenvolvimento do aplicativo, houve algumas dificuldades em relação ao planejamento das entregas, comunicação entre membros e a divisão de tarefas não foi feita de maneira apropriada, sobrecarregando alguns membros.

A interface do usuário foi desenvolvida com o intuito de deixar a experiência de uso o mais fácil possível, deixando ao alcance, todas as ferramentas necessárias e relevantes, além de permitir ao usuário a personalização da sua experiência.

Ao fim das entregas do programa cumprem o que promete, faltando apenas o aprimoramento e adição de algumas funcionalidades. Como possíveis trabalhos futuros, poderia ser feito uma integração completa entre o aplicativo e o sistema, introdução de outros meios de monitoramento, tais como, *smartwatches*, sensores de *smarthome* para complementar o sistema, integração de Alexa e Google Home ao sistema.

## Anexo A – Código do programa em Python de detecção de quedas

```

import cv2
import time

webcam = 1 # usar o número 0 caso for utilizar uma webcam de notebook ou 1 para
uma.
record = # Caminho para um arquivo de vídeo para executar os testes.
fitToEllipse = False
cap = cv2.VideoCapture(webcam)
time.sleep(2)
queda = False

camera = cv2.createBackgroundSubtractorMOG2()
j = 0

while(1):
    ret, frame = cap.read()

    # Converte as imagens do video para cinza, assim facilitando a detecção
    try:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        cammask = camera.apply(gray)

    # Acha os contornos
    contours, _ = cv2.findContours(
        cammask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    if contours:

        # Lista e mantém as áreas de interesse
        areas = []

        for contour in contours:
            ar = cv2.contourArea(contour)
            areas.append(ar)

        max_area = max(areas, default=0)

        max_area_index = areas.index(max_area)

        cnt = contours[max_area_index]

        M = cv2.moments(cnt)

        x, y, w, h = cv2.boundingRect(cnt)

        cv2.drawContours(cammask, [cnt], 0, (255, 255, 255), maxLevel=0)

        if h < w:

```

```
    j += 1

    if j > 10:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
        queda = True

    if h > w:
        j = 0
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    cv2.imshow('video', frame)

    if cv2.waitKey(1) == ord('f'):
        break
except Exception as e:
    break

camera.release()
cv2.destroyAllWindows()
```

## Anexo B – Código do programa em C++ do *dispenser* de comprimidos

```

#include <WiFi.h>
#include <esp_task_wdt.h>
#include <Stepper.h>
#include <SPI.h>
#include <NTPClient.h> //Biblioteca do NTP.
#include <WiFiUdp.h> //Biblioteca do UDP.
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <WiFiClientSecure.h>
#include <HTTPClient.h>

TaskHandle_t Task1;
TaskHandle_t Task2;

const int stepsPerRevolution = 768; // Numero de passos no motor para 21 posições

const char* ssid = "<nome>"; // Nome SSID - WiFi
const char* pass = "<Password>"; // Nome Password - WiFi

//watchdog
#define WDT_TIMEOUT 1200000
void(* resetFunc) (void) = 0;

// token Telegram

#define BOTtoken "<Hash Telegram>" // Hash Bot Telegram
const char* CHAT_ID_01 = "<ID usuário 1>"; // ID usuário 01
const char* CHAT_ID_02 = "<ID usuário 2>"; // ID usuário 01
const char* CHAT_ID_03 = "<ID usuário 3>"; // ID usuário 01
const char* server_02 = "api.telegram.org"; // Endereço API Telegram

// token IFTTT

#define webhookskey "<Hash IFTTT>" //Webhook IFTTT
#define eventname1 "<Evento>" // Evento IFTTT
const char* trigger1 = "/trigger/" eventname1 "/with/key/" webhookskey;

// IFTT
char const* event = trigger1;
const char* server = "maker.ifttt.com"; // Endereço API IFTTT
const int port = 80;
const int httpsPort = 443;
const int API_TIMEOUT = 15000;
bool result = true;

WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);
int Bot_mtbs = 1000;

```

```

long Bot_lasttime;

String hora; // String para informar horario - Debug

WiFiUDP ntpUDP;
WiFiClient net_pill;

// By default 'pool.ntp.org' is used with 60 seconds update interval and
// no offset
NTPClient timeClient(ntpUDP);

// ULN2003 Motor Driver Pins
#define IN1 19
#define IN2 18
#define IN3 5
#define IN4 17
// REED e BUZZER Pins

#define BUZZ 25
#define REED 26

// booting
int i = 0;
int x = 0;
int last = millis();

//variavel de tempo
unsigned long previousMillis = 0;
unsigned long interval = 30000;

// initialize the stepper library
Stepper myStepper(stepsPerRevolution, IN1, IN3, IN2, IN4);

// Conexão com rede Wi-Fi
void initWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, pass);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
  }
  IPAddress localIP = WiFi.localIP();
}

void setup() {
  Serial.begin(115200);
  delay(50);
  initWiFi();
  Serial.print("RSSI: ");

```

```

Serial.println(WiFi.RSSI());
timeClient.begin();
timeClient.setTimeOffset(-10800);
client.setCACert(TELEGRAM_CERTIFICATE_ROOT);
myStepper.setSpeed(10);
pinMode (REED, INPUT);
pinMode (BUZZ, OUTPUT);
pinMode(LED_BUILTIN, OUTPUT);
esp_task_wdt_init(WDT_TIMEOUT, true);
esp_task_wdt_add(NULL);
//create a task that will be executed in the Task1code() function, with priority 1 and
executed on core 0
xTaskCreatePinnedToCore(
    Task1code, /* Task function. */
    "Task1", /* name of task. */
    10000, /* Stack size of task */
    NULL, /* parameter of the task */
    1, /* priority of the task */
    &Task1, /* Task handle to keep track of created task */
    0); /* pin task to core 0 */
delay(500);

//create a task that will be executed in the Task2code() function, with priority 1 and
executed on core 1
xTaskCreatePinnedToCore(
    Task2code, /* Task function. */
    "Task2", /* name of task. */
    10000, /* Stack size of task */
    NULL, /* parameter of the task */
    1, /* priority of the task */
    &Task2, /* Task handle to keep track of created task */
    1); /* pin task to core 1 */
delay(500);
}

```

*// Core 0 no ESP32 com finalidade de execução do NTP e WiFi Reconnect.*

```

void Task1code( void * pvParameters ){
    Serial.print("Task1 running on core ");
    Serial.println(xPortGetCoreID());

    for(;;){
// hora = timeClient.getFormattedTime();
        delay(1000);
        timeClient.update(); // Update ntp a cada 60 segundos
        if (millis() - last >=2000 && x < 5)
        {
            last = millis();
            i++;
            if (x == 5){

```

```

    Serial.printf("ok 3s");
  }
}
unsigned long currentMillis = millis(); // Verifica conexão WiFi e Reconecta cada
haja perda de conexão
// if WiFi is down, try; // reconnecting every CHECK_WIFI_TIME seconds
if ((WiFi.status() != WL_CONNECTED) && (currentMillis - previousMillis
>=interval)) {
  // lgbq
  Serial.print(millis());
  Serial.println("Reconnecting to WiFi...");
  WiFi.disconnect();
  WiFi.reconnect();
  previousMillis = currentMillis;
}
}
}

// Core 1 no ESP32 com finalidade de acesso ao step motor e alertas
void Task2code( void * pvParameters ){
  Serial.print("Task2 running on core ");
  Serial.println(xPortGetCoreID());

  for(;;){
    inicial();

    if (timeClient.getFormattedTime() == "06:59:00">//Se a hora atual for igual à que
definimos, girar roda
    {
      music();
      gaveta();
      telegram();
      delay(600000);
      pote();
    }

    if (timeClient.getFormattedTime() == "11:59:00">//Se a hora atual for igual à que
definimos, girar roda
    {
      music();
      gaveta();
      telegram();
      delay(600000);
      pote();
    }

    if (timeClient.getFormattedTime() == "18:59:00">//Se a hora atual for igual à que
definimos, girar roda
    {
      music(); // Executa alerta via Alexa
      gaveta(); // Entrega medicamento e toca alarme até a retirada.
    }
  }
}

```

```

    telegram(); // Envia comunicado ao BOT Telegram informando que o
    medicamento foi retirado.
    delay(600000); // Aguarda 15 min
    pote(); // Caso o pote de medicamento não seja colocado no equipamento gera
    um alarme
  }

  digitalWrite(LED_BUILTIN, HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
}
}

void loop() {
}

void gaveta() {
  myStepper.step(stepsPerRevolution); // Aciona o step motor para 768 passos
  if (digitalRead(REED)== HIGH); { // Alarme para retirar medicamento
    do
    {
      digitalWrite(BUZZ, HIGH);
      delay(250);
      digitalWrite(BUZZ, LOW);
      delay(100);
      digitalWrite(BUZZ, HIGH);
      delay(250);
      digitalWrite(BUZZ, LOW);
      delay(100);
      digitalWrite(BUZZ, HIGH);
      delay(500);
      digitalWrite(BUZZ, LOW);
      delay(100);
    } while (digitalRead(REED) == LOW);
  }
}

void pote () { // Alarme para colocar o pote do medicamento no local
  if (digitalRead(REED)== LOW); {
    do
    {
      digitalWrite(BUZZ, HIGH);
      delay(100);
      digitalWrite(BUZZ, LOW);
      delay(100);
      digitalWrite(BUZZ, HIGH);
      delay(100);
      digitalWrite(BUZZ, LOW);
      delay(100);
    }
  }
}

```

```

    digitalWrite(BUZZ, HIGH);
    delay(100);
    digitalWrite(BUZZ, LOW);
    delay(100);
    led();
  } while (digitalRead(REED) == HIGH);
}
}

void telegram() {
  Serial.print("Connecting to ");
  Serial.print(server_02);

  WiFiClient client;

  for (int tries = 0; tries < 5; tries++) {      //Faça 5 tentativas de acesso ao site antes
  de enviar o comunicado
    client.setTimeout(API_TIMEOUT);
    if(client.connect(server_02, port)) break; // Sair do loop caso não haja resposta
    Serial.print(".");
    delay(1000);
  }

  Serial.println();
  if(!client.connected()) {
    Serial.println("Failed to connect to server");
    client.stop();
    return;
  }
  Serial.print("Request event: ");              //Executar o comando.
  Serial.println(event);
  bot.sendMessage(CHAT_ID_01, "Pegou remedio", ""); // Comando para enviar
  mensagem via telegram
  delay(1000);
  bot.sendMessage(CHAT_ID_02, "Pegou remedio", "");
  delay(1000);
  bot.sendMessage(CHAT_ID_03, "Pegou remedio", "");
  int timeout = 50;                             //Aguardar 5 segundos para fechar conexão
  while(!client.available() && (timeout-- > 0)){
    delay(100);
  }

  if(!client.available()) {
    Serial.println("No response to GET");
    client.stop();
    return;
  }
  while(client.available()){
    Serial.write(client.read());
  }
}

```

```

Serial.println("\nclosing connection");
delay(1000);
client.stop();
}

void music(){
  Serial.print("Connecting to ");
  Serial.print(server);

  WiFiClientSecure client;

  for (int tries = 0; tries < 5; tries++) {      //Faça 5 tentativas de acesso ao site antes
de enviar o comunicado
    client.setTimeout(API_TIMEOUT);
    client.setInsecure();
    if(client.connect(server, httpsPort)) break; // Sair do loop caso não haja resposta
    Serial.print(".");
    delay(2000);
  }

  Serial.println();
  if(!client.connected()) {
    Serial.println("Failed to connect to server");
    client.stop();
    return;
  }
  Serial.print("Request event: ");              // Comando para enviar mensagem via
IFTTT
  Serial.println(event);
  client.print(String("GET ") + event +
    " HTTP/1.1\r\n" +
    "Host: " + server + "\r\n" +
    "Connection: close\r\n\r\n");

  int timeout = 50;
  while(!client.available() && (timeout-- > 0)){
    delay(100);
  }

  if(!client.available()) {
    Serial.println("No response to GET");
    client.stop();
    return;
  }
  while(client.available()){
    Serial.write(client.read());
  }
  Serial.println("\nclosing connection");
  delay(1000);
  client.stop();
}

```

```
}  
  
void led() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(50);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(50);  
}  
  
void inicial() { // Gera alerta sonoro quando houver conexão a rede WiFi.  
  if (millis() - last >= 2000 && i < 2) {  
    Serial.println("inicial booting...");  
    digitalWrite(BUZZ, HIGH);  
    delay(500);  
    digitalWrite(BUZZ, LOW);  
    delay(100);  
    last = millis();  
    i++;  
    if (i == 2) {  
      Serial.println("in 3s");  
    }  
  }  
}
```

## Bibliografia

AUDIOLAB. AudioLab. **AudioLab**, 2022. Disponível em: <<https://www.audiolab.com.br/>>. Acesso em: 06 Mar 2022.

BECKER, L. O que é React Native? **Orgânica Digital**, 2021. Disponível em: <<https://www.organicadigital.com/blog/o-que-e-react-native/>>. Acesso em: 10 Fev 2022.

CEDRO TECHNOLOGIES. OpenCV: Uma breve introdução à visão computacional com python. **Blog Cedrotech**, 2018. Disponível em: <<https://blog.cedrotech.com/opencv-uma-breve-introducao-visao-computacional-com-python>>. Acesso em: 8 Fev 2022.

CERON, G. F. Desenvolvedor Android Iniciante. **Udemy**, 2019. Disponível em: <<https://www.udemy.com/course/draft/1071470/>>. Acesso em: Junho 2022.

HANASHIRO, A. VS Code - O que é e por que você deve usar? **Treina Web**, 2021. Disponível em: <<https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>>. Acesso em: 8 Fev 2022.

HELPCARE. A HelpCare. **HelpCare Teleassistência Emergencial**, 2022. Disponível em: <<https://www.helpcarebrasil.com.br/a-helpcare>>. Acesso em: 06 Mar 2022.

KOSTEN-HAUS. Sobre a Kosten-Haus. **Kosten-Haus Tecnologia Residencial**, 2022. Disponível em: <<https://www.kostenhaus.com.br/sobre>>. Acesso em: 06 Mar 2022.

KOVACS, L. O que é o Asana? **TecnoBlog**, 2020. Disponível em: <<https://tecnoblog.net/responde/o-que-e-o-asana/>>. Acesso em: 08 Fev 2022.

MARQUES, R. O que é HTML? Entenda de forma descomplicada. **Home Host**, 2019. Disponível em: <<https://www.homehost.com.br/blog/tutoriais/o-que-e-html/>>. Acesso em: 10 Fev 2022.

MELO, D. O que é Python. **TecnoBlog**, 2020. Disponível em: <<https://tecnoblog.net/responde/o-que-e-python-guia-para-iniciantes/>>. Acesso em: 6 Fev 2022.

OKUBO, B. Você Sabe o que é CSS? Entenda Como Funciona e Para que Serve. **GoDaddy**, 2021. Disponível em: <<https://br.godaddy.com/blog/voce-sabe-o-que-e-css-entenda-como-funciona-e-para-que-serve/>>. Acesso em: 10 Fev 2022.

QUIRINO, L. G. B. Med\_pill. **GitHub**, 2022. Disponível em: <[https://github.com/quibueno/med\\_pill](https://github.com/quibueno/med_pill)>. Acesso em: Abril 2022.

REDAÇÃO IUGU. O que é GitHub e qual sua importância para a programação? **iugu**, 2021. Disponível em: <<https://www.iugu.com/iugu4devs/blog/github>>. Acesso em: 8 Fev 2022.

SCHWABER, K.; SUTHERLAND, J. The Scrum Guide: The Defenitive Guide to Scrum: The Rules of the Game. **Scrum Guides**, 2017. Disponivel em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>>. Acesso em: 12 Mai 2022.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de Banco de Dados**. São Paulo: Campus, 2006.

TECNOSENIOR. Sobre a Tecnosenior. **Tecnosenior**, 2022. Disponivel em: <<https://tecnosenior.com/quem-somos/>>. Acesso em: 06 Mar 2022.