



CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

São Caetano do Sul
2020

Fatec
São Caetano
do Sul
Antonio Russo

CPQS
Centro
Paula Souza


**GOVERNO DO ESTADO
DE SÃO PAULO**



Alecsander Torquato Fernandes
Caio Cabesas Aragon
Dennys Magnun Dias Braghirolli
Lucas Domingos Monteiro
Silas Chagas de Moraes

Verde ao Cubo – Verde³ Watcher

São Caetano do Sul/SP
2020
Alecsander Torquato Fernandes
Caio Cabesas Aragon

Dennys Magnun Dias Braghirolli
Lucas Domingos Monteiro
Silas Chagas de Moraes

Verde ao Cubo – Verde³ Watcher

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Caetano do Sul, sob a orientação do professor Me. Adilson Ferreira da Silva. Como requisito parcial para a obtenção do diploma de Graduação no Curso de Análise e desenvolvimento de sistemas

São Caetano do Sul/SP
2020

AGRADECIMENTOS

Alecsander Torquato Fernandes

Aqui venho agradecer ao meu pai por ser o exemplo de pessoa que é, sendo o pilar na estrutura da minha família e alguém que eu me inspiro fortemente. A minha mãe que sempre me ajudou no incentivo do estudo e na busca dos meus objetivos, sempre me apoiando em momentos difíceis e complicados. A minha vó Helena que sempre falou sobre mim em suas orações.

Caio Cabezas Aragon

Eu quero agradecer a todos que de alguma forma me ajudou ao longo do curso. Eu agradeço ao professor Adilson pela orientação no TCC e aos professores da FATEC pelas ensinamentos. Em especial, eu quero agradecer aos meus pais Maria e Miguel, a minha namorada Marina, a minha família e aos meus amigos.

Dennys Magnun Dias Braghirolli

Primeiramente eu quero a agradecer a Deus, que até aqui tem nos sustentado, se não fosse o Senhor nada disso seria possível. Agradeço aos meus pais pelo incentivo e ajuda, meus pais espirituais pelas orientações e oração. Obrigado a minha avó Celisa por sempre me cobrar para nunca parar de estudar. Obrigado a minha esposa por toda força em todos os momentos, pelas alegrias, parceria e pelas risadas. Quero agradecer a minha doce Helena que sempre vem me alegrar e me chamar para brincar. Ao Samuel e Miguel que estão chegando para a turminha kids, papai ama muito vocês. A todos meus colegas do grupo, pela paciência e ajuda. E a todos os professores por terem contribuído nessa minha caminhada estudantil.

Lucas Domingos Monteiro

Primeiramente, tenho um agradecimento enorme à todos estes integrantes de grupo, que me aceitaram posteriormente neste grupo, me abriram uma oportunidade enorme para aprendizado, desenvolvimento pessoal cumprimento do meu dever e oportunidades de extremo valor, impertinentes a este trabalho. Ao meu professor orientador que guiou este trabalho com extrema paciência. Aos meus pais e irmãos pelo incentivo constante.

Silas Chagas de Moraes

Agradecimentos a Deus e a pessoas especiais. À minha Mãe que me acompanha e me apoia sempre, aos caros colegas e amigos que contribuíram direta ou indiretamente para realização desse Trabalho. Agradecimentos também ao professor Orientador Adilson Ferreira da Silva e todos os Professores que nos ajudaram a chegar nessa realização. Agradecimentos também à Fatec Antonio Russo que proporcionou a oportunidade de mais esse aprendizado em nossas vidas.

FERNANDES, Alecsander T.; ARAGON, Caio C.; BRAGHIROLI, Dennys M. D.; MONTEIRO, Lucas D.; MORAES, Silas C. **Verde ao Cubo - Hortas Hidropônicas com IoT**. XXfls. Trabalho de Conclusão de Curso – Faculdade de Tecnologia Antonio Russo de São Caetano do Sul, São Caetano do Sul, 2020.

Com a crescente população nas cidades, a oferta de alimentos com qualidade nutricional tem se tornado um desafio. Entretanto, técnicas de cultivo hidropônico veem se popularizando ao longo das últimas décadas, especialmente nos centros urbanos, mas cultivos hidropônicos precisam de monitoração e manutenção constante. Desta forma, é apresentado neste trabalho o projeto de desenvolvimento de uma solução voltada para o controle e a monitoração remota de sistemas de cultivo hidropônico. Os usuários utilizam um aplicativo mobile para controlar e monitorar seus cultivos. Uma sonda com tomadas inteligentes e sensores instalada no cultivo enviar dados ambientes periodicamente do ambiente e controlar dispositivos externos. A solução possui uma página web para divulgação e cadastro de novos usuários. Devido ao cronograma curto para desenvolver o projeto, foi utilizada uma metodologia de projeto adaptativa, a APF.

Palavras-chave: *horta; hidropônica; raspberry pi; aplicativo*

ABSTRACT

FERNANDES, Alecsander T.; ARAGON, Caio C.; BRAGHIROLI, Dennys M. D.; MONTEIRO, Lucas D.; MORAES, Silas C. **Verde ao Cubo - Hortas Hidropônicas**. XXfls. Trabalho de Conclusão de Curso – Faculdade de Tecnologia Antonio Russo de São Caetano do Sul, São Caetano do Sul, 2020.

With the growing population in cities, the supply of food with traditional quality has become a challenge. However, hydroponic cultivation techniques have become popular over the last few decades, especially in urban centers, but hydroponic cultivations need constant monitoring and maintenance. Thus, this work presents the project to develop a solution aimed at the remote control and monitoring of hydroponic cultivation systems. Users use a mobile application to control and monitor their crops. A probe with smart sockets and sensors installed in the crop periodically send ambient data from the environment and control external devices. The solution has a web page for the dissemination and registration of new users. Due to the short schedule to develop the project, it was used an adaptive design methodology, APF.

Key-words: *garden; hydroponic; raspberry pi; gadgets*

LISTA DE FIGURAS

Figura 1 – Muda em um sistema de cultivo hidropônico

Figura 2 – Um sistema de cultivo hidropônico vertical

Figura 3 – Representação de um sistema hidropônico que utiliza a técnica do fluxo laminar de nutrientes.

Figura 4 – Representação de um sistema de cultivo hidropônico que utiliza a técnica de submersão e drenagem.

Figura 5 – Representação de um sistema de cultivo hidropônico que utiliza um substrato e a técnica de gotejamento.

Figura 6 – Representação de um sistema de cultivo hidropônico que utiliza a técnica de aeroponia.

Figura 7 – Cultivo hidropônico em ambiente aberto

Figura 7.9 - Cultivo hidropônico em ambiente fechado com iluminação artificial.

Figura 8: Diagrama de contexto da solução Verde³ Watcher

Figura 9 – Escopo de definição da solução.

Figura 10 – O ciclo de vida APF

Figura 11– RBS em exemplo

Figura 12 – WBS em exemplo.

Figura 13 – Triângulo de Escopo.

Figura 14 – Exemplo de priorização de variáveis do triângulo de escopo.

Figura 15 – Home (* não está logado)

Figura 16 – Tela de Cadastro.

Figura 17 – Tela de Login.

Figura 18– Home Logado.

Figura 19 – Alteração de Cadastro

Figura 20 – Tela Financeira

Figura 21 – Histórico de Monitoração

Figura 22 – Fale Conosco

Figura 23 - Cancelar Assinatura

Figura 24 – Configuração

Figura 25 – Sensores.

Figura 26 – Sensores Auxiliares

Figura 27 – Sensor de Temperatura

Figura 28 – Tela 1 Home do website.

Figura 29 – Tela 2 Login.

Figura 30 – Tela 3 Erro de login.

Figura 31 - Tela 4 Cadastro.

Figura 32 - Tela 5 Gerenciamento de Conta.

Figura 33 - Home web final.

Figura 34 - Home web final área Serviço

Figura 35 - Home web final Planos.

Figura 36 - Home web final Quem somos

Figura 37 - Home web final Contato

Figura 38 - Home web final Ir para o App.

Figura 39 - Home web final Ir para área de Login.

Figura 40 - Home web final área de Login

Figura 41 - Home web final Tela de Registro.

Figura 42 – Telas iniciais do aplicativo.

Figura 43 – Tela Geral Dispositivos.

Figura 44 – Configuração de Sonda.

Figura 45 – Configuração de Sensores.

Figura 46 – Configuração de Sensor de iluminação.

Figura 47 - Home App final Login.

Figura 48 – Home App final Registro.

Figura 49 – Home App final Registrar a Conta.

Figura 50 – Home App final Confirmação Cadastro.

Figura 51 – Home App final Área para se logar.

Figura 52 – Home App final Home usuário logado.

Figura 53 – Home App final Probe.

Figura 54 – Home App final Logout.

Figura 55 – Home App final Configurar Sonda.

Figura 56 – Home App final Aplicar alteração

Figura 57 – Protótipo final da sonda

Figura 58 - Computador de placa única Raspberry Pi 3 Model B.

Figura 59 - Placa de interface de relés 5V com 4 canais.

Figura 60 - Sensor de temperatura e umidade DHT11 utilizado no módulo de monitoração da sonda

Figura 61 - Resistor LDR utilizado no módulo de monitoração da sonda.

Figura 62 - Fluxograma de Login.

Figura 63 - Fluxograma de Registro.

Figura 64 - Fluxograma exibição da tela do Menu Principal.

Figura 65 - Fluxograma de Logout.

Figura 66 - Fluxograma de Configuração de dispositivos

Figura 67 - Fluxograma de Configuração de sensores

Figura 68 - Diagrama de contexto das sondas da solução Verde³

Figura 69 - Diagrama de classe da sonda

Figura 70 - Fluxograma com o modelo de interação de processos da sonda.

Figura 71 - Diagrama de componentes do programa verde3-probe

Figura 72 - Disposição dos recursos do módulo de controle da sonda e suas conexões com as portas GPIO do módulo de processamento.

Figura 73 - Disposição dos recursos do módulo de monitoração da sonda e suas conexões com as portas GPIO do módulo de processamento.

LISTA DE TABELAS

Tabela 1 – Convenção de nomes adotados no projeto

Tabela 2 - Detalhamento das principais entradas e saídas entre as entidades e os usuários da solução.

Tabela 2.2.e Caso de Testes WebSite

Tabela 2.1 Caso de Testes Aplicativo Mobile

Tabela 2.3 – Relatório de Incidente

Tabela 2.4 – Relatório de Resumo de Teste

Tabela 3 – Requisitos funcionais do sistema

Tabela 4 - Atribuição de papéis APF

Tabela 5. Plano de Teste do sistema Verde Ao Cubo

SUMÁRIO

AGRADECIMENTOS	4
RESUMO	Erro! Indicador não definido.
RESUMO	Erro! Indicador não definido.
LISTA DE FIGURAS	8
LISTA DE TABELAS	11
SUMÁRIO	12
1. Introdução	15
1.1. Hidroponia	17
1.1.1. Tipos de Sistemas Hidropônicos	20
1.1.2. Nutrição Vegetal E Soluções Nutritivas	23
1.1.3. Parâmetros Ambientais Na Hidroponia	25
2. Descrição do Produto	27
2.1. Especificações do Software	27
2.1.1. O Produto	28
2.1.2. Objetivos do Produto	29
2.1.3. Justificativa do Produto	29
2.1.4. Usuários	29
2.1.5. Convenção de Nomes	30
2.1.6. Escopo do Sistema	30
2.2. Requisitos do Sistema	33
3. Desenvolvimento	35
3.1. Metodologia de Desenvolvimento do Projeto	35
3.1.1. APF	35

3.1.2. Objetivo e Solução.....	39
3.1.3. Gestão	39
3.1.4 O Ciclo de Vida APF	40
3.1.4.1. Version Scope (ou “Escopo de Versão”)	41
3.1.4.1.1. POS	41
3.1.4.1.2. RBS e WBS	42
3.1.4.1.3. Triângulo de Escopo.....	44
3.1.4.1.4. Lista Priorizada de Requisitos	45
3.1.4.1.5. Plano de Projeto em Alto Nível.....	45
3.1.4.2. Cycle Plan (ou “Planejamento de Ciclo”).....	45
3.1.4.3. Cycle Build (ou “Construção de Ciclo”)	46
3.1.4.4. Client Check Point (ou “Ponto de Verificação do Cliente”)	46
3.1.4.5. Post-Version Review (ou “Revisão Pós-Versão”).....	46
3.1.4.6. Post-Version Report (ou “Relatório Pós-Versão”)	47
3.1.5. Implementação do APF	47
3.1.5.1. Gestão.....	48
3.1.5.2. Escopo da Versão.....	49
3.1.5.3. Gestão de Ciclos	50
3.1.5.4. Software de apoio à gestão APF	51
3.1.5.4.1 Google Drive.....	51
3.1.5.4.2 Google Docs	51
3.2. Protótipos	51
3.2.1. Protótipos da Página Web	52
3.2.1.1. 1º Protótipo.....	53
3.2.1.2. 2º Protótipo.....	64
3.2.1.3. Protótipos Finais.....	67
3.2.2. Protótipos do Aplicativo Verde ³	73
3.2.2.1. Protótipos Iniciais	73
3.2.2.2. Protótipos Finais.....	78
3.2.3. Protótipos da Sonda.	89
3.3. Tecnologias Utilizadas.....	91
3.3.1. Tecnologias da Página Web	91
3.3.2. Tecnologias do Aplicativo	92
3.3.3. Tecnologias da Sonda	94
3.3.3.1. Módulo de Processamento.....	94

3.3.3.2. Módulo de Controle	95
3.3.3.3. Módulo de Monitoração	96
3.3.3.4. Recursos de Software Utilizados.....	98
3.4. Implementação	98
3.4.1. Implementação em Servidores	98
3.4.2. Implementação da Página Web	104
3.4.3. Implementação do Aplicativo	105
3.4.4. Implementação da Sonda	113
3.5. Montagem da Sonda	117
3.6. Implantação.....	121
3.6.1. Implantação da Página Web.....	Erro! Indicador não definido.
3.6.2. Implantação do Aplicativo	Erro! Indicador não definido.
3.6.3. Implantação da Sonda.....	Erro! Indicador não definido.
4. Testes	122
5. Discussão e Conclusão	138
REFERÊNCIAS BIBLIOGRÁFICAS.....	Erro! Indicador não definido.
APÊNDICE A – DESCRIÇÃO GERAL DO PROJETO.....	149
APÊNDICE B – RBS DO PROJETO.....	151
APÊNDICE C – WBS DO PROJETO.....	152
APÊNDICE D – PRIORIZAÇÃO DAS VARIÁVEIS DO TRIÂNGULO DE ESCOPO	156
APÊNDICE E – PLANO DE PROJETO EM ALTO NÍVEL	157

1. Introdução

Uma dieta saudável fornece ao corpo humano todos os nutrientes essenciais e em quantidades ideais que auxilia a manter ou melhorar a sua saúde. Algumas principais fontes de alimentos são de origem vegetal (verduras, frutas e legumes). O Ministério da Saúde do Brasil (2010) recomenda a ingestão diária de pelo menos três porções de legumes e verduras como parte das refeições e 3 porções ou mais de frutas nas sobremesas e lanches. No entanto, nem todas as pessoas tem condições de consumir esses alimentos diariamente, nas quantidades recomendadas, seja por falta de recursos financeiros ou mesmo pela indisponibilidade desses alimentos (ORSINI et al., 2013).

O crescimento populacional nos centros urbanos e o aumento da renda per capita mundial devem sustentar um crescimento contínuo na demanda mundial por alimentos. Anteriormente, esta demanda foi progressivamente atendida através da expansão da zona rural e do aumento da produtividade nos campos (SAATH; FACHINELLO, 2018). Em contrapartida, devido à acentuada expansão urbana e industrial e aos seus impactos ambientais associados, a utilização do solo na produção de alimentos vem decrescendo nas últimas décadas e pode diminuir ainda mais no futuro. Outro desafio corresponde ao estado de fertilidade dos solos aráveis, cujos níveis de saturação são altíssimos e cuja produtividade não mais pode ser aumentada com o uso de fertilizantes (SARDARE; ADMANE, 2013).

Entretanto, técnicas agrícolas inovadoras como a hidroponia tem se tornado cada vez mais relevantes no cenário desafiador de suprir a demanda mundial por alimentos, uma vez que não utilizam solo para a produção e ainda permitem um gerenciamento eficiente dos recursos naturais utilizados (SARDARE & ADMANE, 2013). Outra abordagem interessante está na utilização de estufas, ambientes controlados, cujas variáveis ambientais podem ser monitoradas e manipuladas e que permitem que condições ideais de cultivo sejam atingidas. Estes ambientes podem ainda ser complementados com sistemas de cultivo hidropônico, passíveis a serem automatizados (FERNANDES et. al., 2018).

Apesar da hidroponia ser considerado um método inovador de produção de alimentos, uma de suas principais desvantagens é a sua necessidade permanente por manutenção e monitoração (NETO; BARRETO, 2012). Neste contexto, foi desenvolvida a solução Verde³ Watcher, um sistema de controle e monitoração para sistemas de cultivo hidropônico.

A solução permite que um produtor defina o nível necessário de controle e monitoração para o seu cultivo hidropônico, de forma remota e prática, pois estas funcionalidades são controladas via aplicativo mobile. Para auxiliar o produtor na busca pelas condições ideais em seu cultivo, a solução contém recursos que permitem a visualização dos dados coletados no cultivo.

As funções de controle e monitoração do cultivo são controladas por uma sonda instalada em um cultivo. Um produtor pode instalar quantas forem necessárias. Estes dispositivos operam quando estão conectados à internet e são compostos por três módulos: processamento, controle e monitoração. O módulo de controle contém tomadas inteligentes. O módulo de monitoração possui sensores de temperatura, umidade e luminosidade. O módulo de processamento é o módulo principal da sonda, responsável por controlar os recursos dos outros módulos, de acordo com configurações definidas pelo produtor, e por enviar os dados coletados ao produtor.

Certamente, um dos diferenciais da solução está no uso de plataformas de computação em nuvem, que proporcionam persistência dos dados e o acesso simultâneos de diferentes dispositivos, desde que conectados à internet.

Para gerir o processo de desenvolvimento da solução, foi implantada uma metodologia baseada no Adaptive Project Framework (Estrutura de Projeto Adaptável), que define um conjunto de princípios e processos utilizados para o gerenciamento de projetos suscetíveis a mudanças (WYSOCKI, 2010).

1.1. Hidroponia

O solo é considerado um dos substratos mais acessíveis e utilizados na agricultura, pois, serve como um meio físico para a sustentação das plantas e ainda contém nutrientes essenciais para o desenvolvimento destes organismos, entretanto, a intensa utilização de pesticidas e fertilizantes tem afetado a produtividade das lavouras (SIDDIQ et al., 2019; SARDARE; ADMANE, 2013). A hidroponia, por sua vez, é descrita na literatura como um conjunto de técnicas de cultivo sem a utilização de solo (Figura 1). Os organismos cultivados adquirem nutrientes e oxigênio através do contato de suas raízes com uma solução aquosa oxigenada, rica em nutrientes minerais inorgânicos dissolvidos (FERNANDES, 2018; GRIFFITHS, 2014; NETO; BARRETO, 2012).

Figura 1 – Muda em um sistema de cultivo hidropônico.



Fonte: General Hydroponics.

Admite-se que o termo hidroponia foi utilizado pela primeira vez em 1937, pelo pesquisador estadunidense Willian Frederick Gericke, da Universidade da Califórnia, responsável por conduzir experimentos em laboratório e posteriormente, responsável por transferir estes conhecimentos empíricos para os campos, empregando de forma pioneira a hidroponia para fins comerciais (NETO; BARRETO, 2012). Desde então, é possível dizer que a hidroponia se popularizou. Na atualidade países como, Holanda, Alemanha, Espanha, Itália, Suécia, Austrália, Japão e

Estados Unidos praticam a hidroponia para a produção de alimentos em escala comercial. A NASA (*National Aeronautics and Space Administration*) tem utilizado a técnica para cultivar espécies de hortaliças em viagens espaciais longas (MARTINEZ, 2002 *apud* NETO; BARRETO, 2012). No Brasil, o cultivo em escala comercial vem crescendo de forma rápida, principalmente nos estados de São Paulo, Minas Gerais e Rio Grande do Sul, devido aos projetos de pesquisa pioneiros que vêm sendo desenvolvidos, voltados para a implantação da hidroponia (SANTOS, 2000 *apud* NETO; BARRETO, 2012).

A hidroponia é considerada por muitos autores como uma técnica inovadora de produção alimentos e que apresenta muitas vantagens quando comparada com agrícolas métodos convencionais (FERNANDES, 2018; GRIFFITHS, 2014; SARDARE; ADMANE, 2013; NETO; BARRETO, 2012). Nesse contexto, países como Estados Unidos, Holanda e China tem intensivamente investido em novas tecnologias aplicadas em cultivos hidropônicos automatizados, que buscam principalmente elevar a produtividade desses sistemas (FERNANDES, 2018). Atualmente, encontram-se diversos tipos de sistemas de cultivo hidropônico na literatura, cada um com suas características específicas, vantagens e desvantagens. Segundo Neto e Barreto (2012), entre as vantagens da hidroponia destacam-se:

- Melhor controle da composição dos nutrientes fornecidos;
- Redução no ciclo de cultura e aumento de produtividade;
- Menor consumo de água e fertilizantes;
- Diminuição drástica do uso de agrotóxicos;
- Redução de risco climáticos como frio ou chuvas excessivas;
- Permitir colheitas fora de estação;
- Melhor qualidade e preço dos produtos;
- Cultivos mais próximos dos consumidores;

Entre as desvantagens, destacam-se segundo os mesmos autores:

- Custo inicial relativamente elevado;
- Necessidade de assistência e conhecimentos técnicos mais efetivos;
- Necessidade de mão de obra especializada;
- Riscos associados a falta de energia elétrica;
- Prejuízos por contaminação da água;

- Necessidade de acompanhamento permanente do sistema;

Sistemas de cultivo hidropônico também podem ser vantajosos ao se considerar a indisponibilidade de áreas cultiváveis nos centros urbanos ou em regiões com condições geográficas ou topográficas desfavoráveis. É importante considerar também a escassez de mão de obra nos campos agrícolas convencionais. Nestas circunstâncias, sistemas de cultivo hidropônico podem ser implantados com sucesso (SARDARE; ADMANE, 2013).

A produção de alimentos nos centros urbanos é conhecida como agricultura urbana. Definida também como a produção de alimentos em casas ou lotes nas cidades ou regiões urbanas periféricas. Nestes locais, empregam-se tanto métodos agrícolas convencionais quanto métodos mais inovadores como a hidroponia, que se destaca novamente devido ao melhor aproveitamento de recursos hídricos e a menor poluição gerada com transportes e embalagens (ORSINI *et al.*, 2013).

Atualmente a produção urbana é considerada como complementar à produção de regiões rurais (ORSINI *et al.*, 2013), entretanto, os desenvolvimentos tecnológicos aplicados na horticultura urbana geraram um modelo promissor de produção de alimentos, denominado: agricultura vertical (Figura 2).

Figura 2 – Um sistema de cultivo hidropônico vertical



. Fonte: GP Solutions.

A agricultura vertical corresponde à prática de produzir alimentos em sistemas modulares que podem ser dispostos verticalmente para produzir muito mais comida por acre do que qualquer cultivo tradicional. Cultivos verticais são um tipo de ambiente agrícola controlado tipicamente com um ambiente climatizado, um sistema de luz artificial e um sistema hidropônico ou aeropônico para nutrir os organismos de cultivo. Estes ambientes são altamente passíveis a automação (CAMBRIDGE CONSULTANTS, 2020).

1.1.1. Tipos de Sistemas Hidropônicos

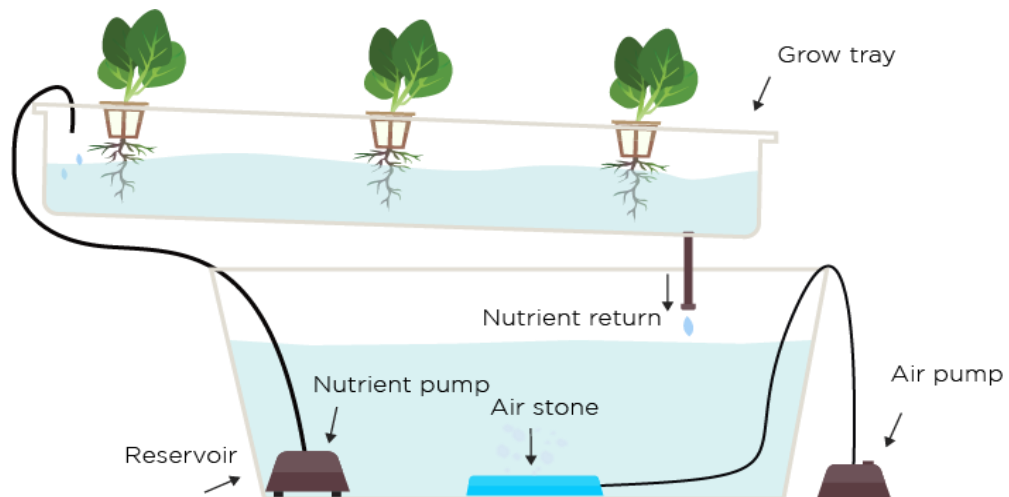
Os sistemas de cultivo hidropônico existentes diferem entre si pela forma de sustentação dos organismos (em meio líquido ou em substrato), pelo fornecimento da solução nutritiva (contínua ou intermitente) e pelo reaproveitamento da solução nutritiva (circulantes e não circulantes) (UFRB, 2007 *apud* NETO; BARRETO, 2012). Sistemas circulantes são muito favoráveis à automação e capazes de prover um ambiente nutritivo estável para os cultivos, mas também trazem riscos caso ocorra uma interrupção no fornecimento de solução nutritiva para a área de cultivo (SARDARE; ADMANE, 2013). Entre os tipos de sistemas hidropônicos, vale apresentar:

- *Nutrient Film Technique* (NFT) ou Técnica de Fluxo Laminar: Sistemas voltados ao cultivo de organismos em canais ou canaletas (Figura 3), por onde circula uma solução nutritiva em intervalos periódicos e controlados. As raízes dos vegetais

são parcialmente submersas na lâmina de solução nutritiva, permitindo que respirem de forma normal (NETO; BARRETO, 2012). O sistema NFT não utiliza nenhum substrato e é classificado como um sistema fechado, pois, a solução nutritiva que circula é reutilizada constantemente (RODRIGUES, 2002 *apud* NETO; BARRETO, 2012).

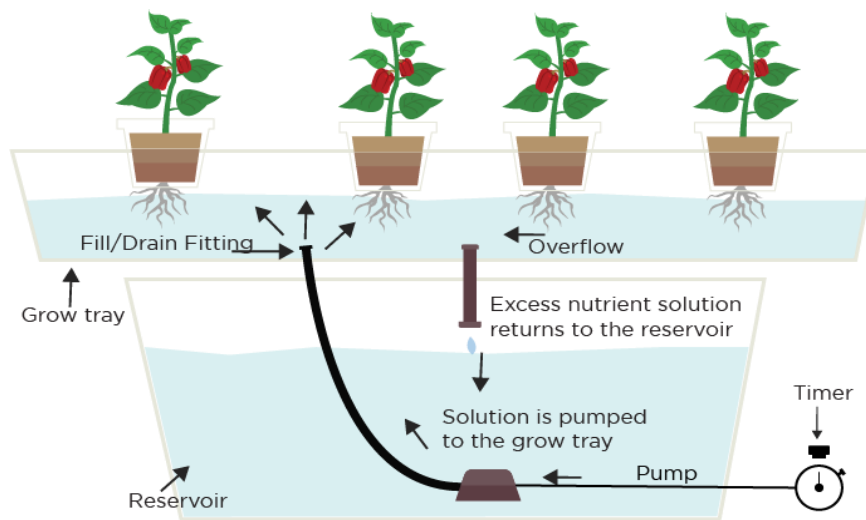
- *Ebb and Flow Technique* ou Técnica de submersão e drenagem: Técnica de cultivo sem substrato. Os organismos são cultivados em vasos que permitem que suas raízes fiquem completamente submersas na solução nutritiva (Figura 4), que de forma intermitente é drenada para um reservatório e bombeada novamente para os vasos (GRIFFITHS, 2014; NETO; BARRETO, 2012). Nos sistemas que utilizam esta técnica de cultivo, a respiração das raízes ocorre nos intervalos entre a submersão e a drenagem.

Figura 3 – Representação de um sistema hidropônico que utiliza a técnica do fluxo laminar de nutrientes.



Fonte: Green and Vibrant (2019).

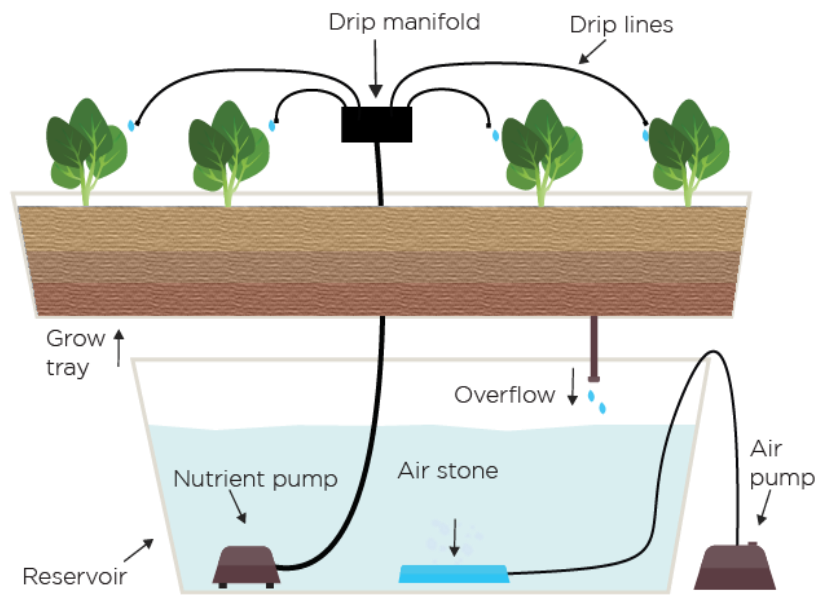
Figura 4 – Representação de um sistema de cultivo hidropônico que utiliza a técnica de submersão e drenagem.



Fonte: Green and Vibrant.

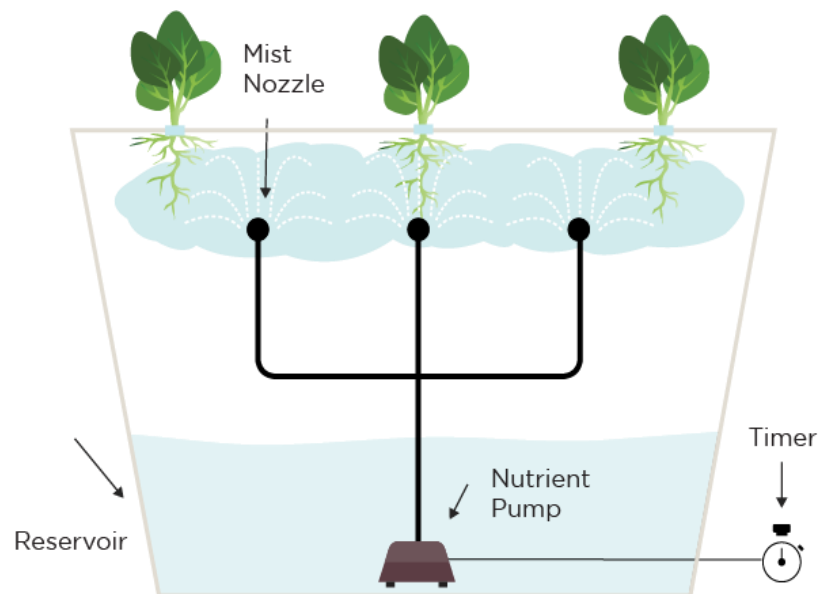
- Cultivo com substratos: Técnica com organismos cultivados em vasos e sustentados por um substrato inerte ou pouco ativo quimicamente, como areia lavada, cascalho e argila expandida (Figura 5). É considerado um sistema aberto, pois, parte da solução nutritiva fica retida no substrato e não retorna ao reservatório. A solução nutritiva pode ser fornecida via gotejamento, submersão ou circulação (NETO; BARRETO, 2012).
- Aeroponia: Similar a hidroponia, esta técnica difere principalmente na forma como é feita de circulação da solução nutritiva, fornecida como um aerossol ou como vapor (SARDARE; ADMANE, 2013). Nesta técnica, os organismos cultivados ficam suspensos pelo caule em um suporte (Figura 6), enquanto suas raízes são mantidas no interior de câmaras opacas protegidas da luz (evitando a proliferação de algas). Sistemas de cultivo aeropônicos ainda são pouco utilizados comercialmente devido aos seus custos de implantação e dificuldades operacionais (NETO; BARRETO, 2012).

Figura 5 – Representação de um sistema de cultivo hidropônico que utiliza um substrato e a técnica de gotejamento.



Fonte: Green and Vibrant.

Figura 6 – Representação de um sistema de cultivo hidropônico que utiliza a técnica de aeroponia.



Fonte: Green and Vibrant.

1.1.2. Nutrição Vegetal E Soluções Nutritivas

Plantas terrestres são organismos autótrofos, isto é, são organismos aptos de produzir seu próprio alimento. Sua nutrição ocorre através da absorção de gás carbônico atmosférico (CO_2), de água (H_2O) e de um conjunto de elementos químicos inorgânicos presentes no solo, classificados em elementos essenciais, benéficos e tóxicos (MALAVOLTA, 2006 *apud* NETO; BARRETO, 2012).

Os elementos essenciais correspondem aos 17 nutrientes necessários para o desenvolvimento dos vegetais, sem os quais, estes organismos não conseguem se desenvolver por completo. No entanto, devido às diferentes demandas nutricionais, estes elementos podem ser classificados como macronutrientes (C, H, O, N, P, K, Ca, Mg e S) ou como micronutrientes (B, Cl, Cu, Fe, Mn, Mo, Ni e Zn). Os elementos benéficos são elementos cuja ausência não traz nenhum prejuízo aos organismos, mas cuja presença em determinadas circunstâncias auxiliam no seu crescimento e conseqüentemente, podem promover o aumento de produtividade de um cultivo. Os elementos tóxicos são desnecessários no cultivo e podem, mesmo em baixas concentrações, causar problemas de toxidez e devastar um cultivo (NETO; BARRETO, 2012).

A solução nutritiva representa a principal forma de fornecimento de nutrientes nos sistemas de cultivo hidropônico e é considerada como um dos componentes mais importantes de todo o sistema (NETO; BARRETO, 2012). Desta forma, se faz necessário monitorar frequentemente sua concentração e também, promover os devidos ajustes para evitar qualquer mudança acentuada na concentração da solução nutritiva (SARDARE; ADMANE, 2013; NETO; BARRETO, 2012).

Toda solução deve conter os macros e micronutrientes essenciais, dissolvidos em água e em quantidades balanceadas para atender às necessidades nutricionais da espécie cultivada, além de apresentar níveis de pH adequados para o cultivo (NETO; BARRETO, 2012). Além do mais, o volume e a frequência do aporte de solução nutritiva no sistema hidropônico depende diretamente do substrato utilizado, da espécie cultivada, do estágio de desenvolvimento da espécie cultivada, do tamanho do reservatório, do tipo de sistema de irrigação e das condições climáticas predominantes no local (SARDARE; ADMANE, 2013). Outro aspecto importante é considerar um suprimento adequado de oxigênio ao sistema radicular do cultivo. Bons resultados são obtidos com o bombeamento vigoroso de ar na solução nutritiva (NETO; BARRETO, 2012).

Na solução nutritiva, é importante monitorar seu pH e a sua condutividade elétrica (CE). O controle do pH é importante não somente na solução nutritiva, mas também no substrato (se utilizado), pois, os organismos perdem a capacidade de absorver diferentes nutrientes devido a variabilidade do pH na solução (GRIFFITHS, 2014). De modo geral, as plantas podem ser cultivadas em uma faixa de pH bastante ampla.

No entanto, o pH ideal para a maioria das culturas se encontra na faixa de 5.5 a 6.5, muito embora, o intervalo ideal de pH seja específico para cada cultura (NETO & BARRETO, 2012).

Após o PH, a condutividade elétrica (CE) é o segundo parâmetro mais importante a ser medido na solução nutritiva (GRIFFITHS, 2014). Como a composição química da solução varia em função da absorção seletiva de água e nutrientes pelas plantas, se faz necessário monitorar diariamente a composição da solução, (COSTA *et al.*, 2001). Recomenda-se que a condutividade elétrica da solução nutritiva seja aferida sempre sob valores constantes de pH, pois, desta forma se obtém um valor de CE diretamente correlacionado com a concentração total de nutrientes dissolvidos na solução (GRIFFITHS, 2014; NETO; BARRETO, 2012).

1.1.3. Parâmetros Ambientais Na Hidroponia

Além da monitoração dos indicadores de qualidade da solução nutritiva, como o pH e a condutividade elétrica (CE), existem outros parâmetros de natureza ambiental que também influenciam o desenvolvimento dos cultivos hidropônicos, e por consequência, acabam interferindo na produtividade e na qualidade dos organismos cultivados (FERNANDES, 2018; ADAMS, 1993).

A luz é essencial para o processo de fotossíntese (processo de conversão de água e gás carbônico em nutrientes primários com o uso da radiação solar) e sua disponibilidade afeta diretamente a produtividade dos cultivos hidropônicos (YEH; CHUNG, 2009; ADAMS, 1993). Sistemas de cultivo hidropônico em ambientes abertos (Figura 7) tendem a experimentar oscilações diárias de temperatura, decorrentes da variabilidade diária e natural de radiação solar. O excesso de radiação solar incidente pode representar um risco aos cultivos, pois, com a elevação da temperatura local ocorre uma diminuição na taxa fotossintética dos organismos dos cultivos, impactando a produtividade do cultivo (LUZ, 2008) e a taxa de crescimento dos mesmos (ADAMS, 1993). Desta forma, torna-se importante a monitoração da temperatura na área de cultivo. Os sistemas de cultivo hidropônico em ambientes fechados (Figura 7.9) tendem a não ser impactados desta maneira, pois, a radiação solar incidente é controlada ou mesmo inexistente, mas ainda assim, é importante monitorar a sua variabilidade local (GRIFFITHS, 2014). Para que os organismos absorvam devidamente os nutrientes disponibilizados na solução

nutritiva é necessário também monitorar a temperatura da solução, cujos valores ideais oscilam entre 18 e 26°C (GRIFFITHS, 2014).

Figura 7 – Cultivo hidropônico em ambiente aberto.



Fonte: Easy Grow.

Figura 7.9 - Cultivo hidropônico em ambiente fechado com iluminação artificial.



Fonte: Osram.

A umidade relativa do ar é outro parâmetro de natureza ambiental que pode trazer impactos tanto positivos quanto negativos aos cultivos hidropônicos, pois, umidade relativa afeta a capacidade de transpiração dos organismos e pode dificultar o movimento interior de nutrientes nos mesmos (ADAMS, 1993). Apesar que um ambiente com alta umidade relativa do ar (maior que 68%) trazem benefícios nos cultivos de frutos. Para hortaliças, um ambiente com baixa umidade relativa (menor que 45%) favorece o desenvolvimento destes organismos (ADAMS, 1993).

Com um sistema de iluminação adequado, sistemas de cultivo hidropônico em ambientes fechados conseguem eliminar a variabilidade natural da radiação solar e

ainda, mitigar potenciais riscos de origem climática ao cultivo (YEH; CHUNG, 2009). Sistemas modernos de iluminação artificial utilizam um conjunto de lâmpadas LED (*light-emitting diodes*) para emitir a radiação necessária para o processo de fotossíntese dos organismos do cultivo hidropônico (Figura 8). O uso desta tecnologia permite que a composição espectral da radiação emitida pelos LEDs seja controlada, desta forma, os organismos do cultivo recebem a radiação que mais estimula seus processos fotossintéticos (luz vermelha e azul), estimulando a produtividade no sistema. Além disso, os LEDs produzem pouco calor latente enquanto ativos, reduzindo possíveis alterações na temperatura do local (YEH; CHUNG, 2009).

2. Descrição do Produto

Um projeto de desenvolvimento de software abrange um determinado conjunto de processos, conhecidos como processos de softwares, que representam uma sequência intercalável de atividades técnicas, colaborativas e gerenciáveis, cujo objetivo principal consiste em especificar, arquitetar, implementar e testar um sistema de software (SOMMERVILLE, 2010). Este conjunto de processos de software permite atribuir mais qualidade, previsibilidade e economia de recursos ao sistema (WASLWICK, 2019).

2.1. Especificações do Software

A especificação do software ou a engenharia de requisitos consiste no processo de compreender e definir quais são os serviços essenciais do sistema, além de identificar suas limitações operacionais e de desenvolvimento, descrevendo o que precisa ser feito, mas não a forma como será feito (SOMMERVILLE, 2010). A engenharia de requisitos busca saber o que é necessário construir antes de iniciar o processo de desenvolvimento do sistema e assim, evita-se o retrabalho (PAETSCH; EBERLEIN; MAURER, 2003)

Desenvolvimento de acordo com o dicionário Houaiss é: ação ou efeito de desenvolver(-se). É algo contínuo, sempre tem coisas a mais para serem acrescentadas. Ele tem partes, que vão se somando para se chegar ao todo. Cada uma dessas partes tem suas especificidades e importância. No caso do Trabalho de Graduação (TCC) o desenvolvimento engloba as partes teórica e prática, onde o

produto finalizado vai demonstrar como foi o processo de construção, a base, o estudo e os testes.

2.1.1. O Produto

Cultivos hidropônicos necessitam de monitoração e manutenção constante. Desta forma, a equipe Verde³ desenvolveu a solução Verde³ Watcher, um sistema de controle e monitoração para cultivos hidropônicos.

O sistema permite que um produtor defina os níveis necessários de controle e monitoração em seu cultivo hidropônico de forma remota. Estas funcionalidades são controladas via aplicativo mobile, proporcionando ao produtor a capacidade de controlar os horários de funcionamento de dispositivos, como bombas de água, e ainda receber informações atualizadas sobre os parâmetros ambientais monitorados no cultivos.

As sondas do Verde³ Watcher operam quando conectadas à internet e são compostas por três módulos: um para processamento, um para controle e outro para monitoração. O módulo de processamento contém todos os recursos necessários para condicionar o funcionamento de sensores e de circuitos de acionamento, além de receber e transmitir informações ao produtor. O módulo de controle contém relés e conectores que controlam dispositivos como bombas de água e sistemas de iluminação. O módulo de monitoração contém sensores de temperatura, de umidade relativa do ar e de luminosidade. Um detalhe interessante, um produtor pode instalar em seu cultivo quantas sondas forem necessárias. É importante informar as sondas se encontram ainda como protótipos funcionais, mas suas funcionalidades estão mantidas na solução.

Dentre todas as características da solução, seu principal diferencial reside no uso de plataformas de computação em nuvem, que proporcionam persistência dos dados de monitoração e o acesso de diferentes dispositivos, desde que possuam acesso à internet. A comunicação *peer-to-peer* é dispensada em favor de um serviço online que não exige maiores conhecimentos do usuário final sobre a configuração de rede ou de seus dispositivos inteligentes.

Por último, para auxiliar o produtor na busca pelas condições ideais em seu cultivo, o Verde³ Watcher permite que séries temporais históricas dos parâmetros ambientais monitoradas no cultivo sejam visualizadas no aplicativo mobile.

2.1.2. Objetivos do Produto

A solução Verde³ Watcher tem como objetivo principal habilitar o controle e a monitoração remota de sistemas de cultivo hidropônico. Os objetivos específicos do software são: desenvolver um aplicativo móvel para facilitar a usabilidade do usuário; desenvolver uma sonda capaz de atuar diretamente no controle e na monitoração de cultivos hidropônicos; desenvolver uma interface de programação de aplicação para permitir a integração entre o aplicativo móvel e a sonda.

2.1.3. Justificativa do Produto

O uso de sistemas de cultivo hidropônico em centros urbanos representa uma forma inovadora de produção de alimentos. Entretanto, uma das desvantagens destes sistemas é a permanente necessidade de monitoração e manutenção (NETO; BARRETO, 2012), além do mais, em sistemas hidropônicos convencionais, os produtores precisam ir regularmente aos locais de cultivo para verificar o sistema e monitorar o ambiente, uma tarefa que requer muito trabalho e atenção. Qualquer negligência pode resultar em baixa qualidade dos produtos ou baixa produtividade do cultivo (SIDDIQ et al., 2019).

Desta forma, se justifica o desenvolvimento da solução Verde³ Watcher, pois o sistema busca justamente habilitar o controle e a monitoração remota de cultivos hidropônicos e pode simplificar, ou pelo menos reduzir a carga de tarefas que os produtores têm com seus cultivos. A longo prazo, a capacidade de controlar remotamente um cultivo hidropônico pode possibilitar que um produtor alcance resultados positivos enquanto diminui o tempo e o esforço necessário para sustentar um ambiente produtivo (GROWLINK, 2018).

2.1.4. Usuários

A solução Verde³ Watcher destina-se a todas as pessoas que possuem um sistema de cultivo hidropônico em operação ou que estão iniciando na área e tem interesse em ter um. Considera-se então, que os usuários da solução serão produtores ou cultivadores com diferentes níveis de experiência. Acredita-se que quanto o maior o

nível de conhecimento e experiência do usuário, melhor será a sua experiência com o sistema. É necessário que os usuários tenham familiaridade com o uso de aplicativo móvel e com o uso da internet.

2.1.5. Convenção de Nomes

O projeto de desenvolvimento da solução Verde³ Watcher utiliza a convenção de nomes apresentados na Tabela 1 com suas respectivas definições.

Tabela 1 – Convenção de nomes adotados no projeto

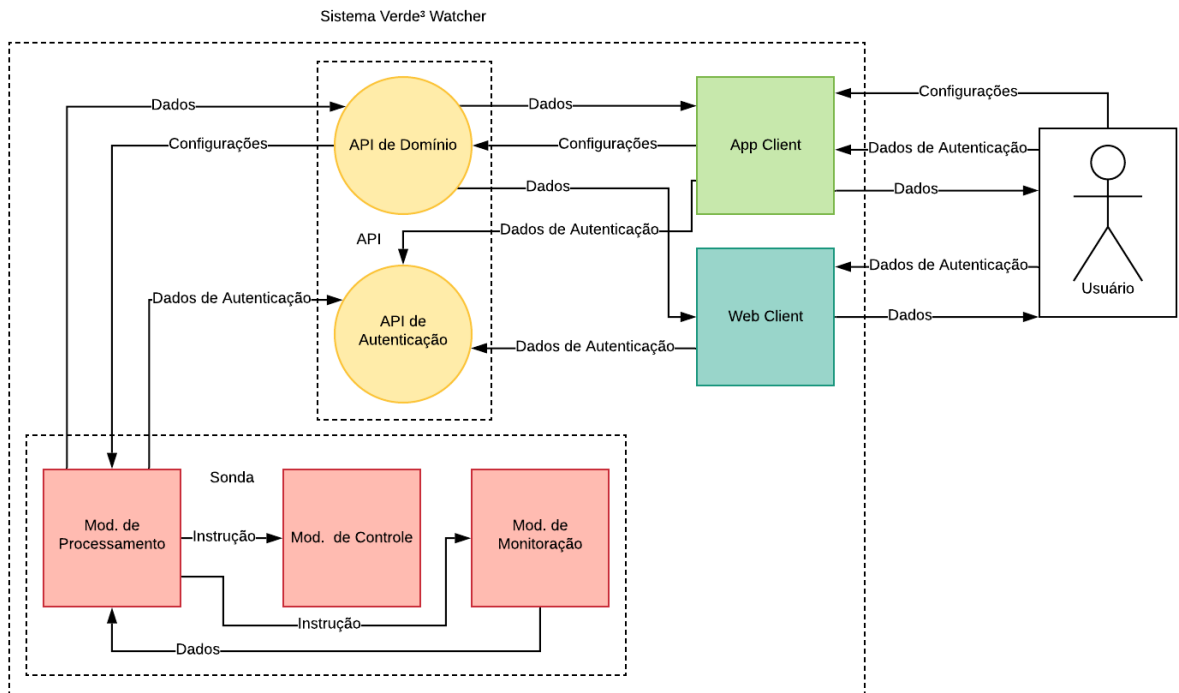
Termo	Definição
App Client	Aplicativo móvel desenvolvido para a solução
Web Client	Página web desenvolvida para a solução
Módulo de Processamento	Componente da sonda com todos os recursos necessários para condicionar o funcionamento de sensores, dos circuitos de acionamento e da transmissão de dados.
Módulo de Controle	Componente da sonda com relés eletrônicos e conectores para o controle de dispositivos externos conectados ao sistema, como bombas de água e sistemas de iluminação
Módulo de Monitoração	Componente da sonda com os sensores de temperatura, umidade relativa do ar e luminosidade

Desenvolvido por Caio Cabezas

2.1.6. Escopo do Sistema

Nas fases iniciais de especificação de um software é necessário definir os limites do sistema, isto é, determinar o que o sistema faz e não faz (SOMMERVILLE, 2010). O diagrama apresentado na Figura 8 representa o modelo de contexto criado para especificar os limites do sistema Verde ao Cubo. De forma simplificada, este diagrama exhibe a interação entre os diferentes atores do sistema suas principais entradas e saídas (Tabela 2).

Figura 8: Diagrama de contexto da solução Verde³ Watcher



Desenvolvedor Caio Cabezas 2020

Tabela 2 - Detalhamento das principais entradas e saídas entre as entidades e os usuários da solução.

ENTIDADE	DESCRIÇÃO
Usuário (Cliente)	Dados de Autenticação: credenciais do usuário e token de autenticação
	Configurações: módulo de controle {estado de operação(e.g.: ativar, ativo e desativado), modo de operação (e.g.: periódica ou contínua),período de atividade (HH:MM:SS), período de inatividade (HH:MM:SS) e data de configuração (AAAA-MM-DD HH:MM:SS)} e módulo de monitoração{tipo de sensor (e.g.: temperatura), estado de operação(e.g.: ativar, ativo e desativado), intervalo de amostragem(HH:MM:SS) e data de configuração (AAAA-MM-DD HH:MM:SS)}
	Dados: tipo de dado (e.g.: temperatura), valor (e.g.: 15°C) e data de aquisição (AAAA-MM-DD HH:MM:SS)
Usuário (Administrador)	Dados de Autenticação: credenciais do usuário e token de autenticação
	Configurações: número serial da sonda (e.g.: 000000000e2e9562)
Web Client	Dados de Autenticação: credenciais do usuário e token de autenticação
	Dados: tipo de dado (e.g.: temperatura), valor (e.g.: 15°C) e data de aquisição (AAAA-MM-DD HH:MM:SS)
App Client	Configurações: módulo de controle {estado de operação(e.g.: ativar, ativo e desativado), modo de operação (e.g.: periódica ou contínua),período de atividade (HH:MM:SS), período de inatividade (HH:MM:SS) e data de configuração (AAAA-MM-DD HH:MM:SS)} e módulo de monitoração{tipo de sensor (e.g.: temperatura),

	estado de operação(e.g.: ativar, ativo e desativado), intervalo de amostragem(HH:MM:SS) e data de configuração (AAAA-MM-DD HH:MM:SS)}
	Dados: tipo de dado (e.g.: temperatura), valor (e.g.: 15°C) e data de aquisição (AAAA-MM-DD HH:MM:SS)
	Dados de Autenticação: credenciais do usuário e token de autenticação
API de Domínio	Configurações: módulo de controle {estado de operação(e.g.: ativar, ativo e desativado), modo de operação (e.g.: periódica ou contínua),período de atividade (HH:MM:SS), período de inatividade (HH:MM:SS) e data de configuração (AAAA-MM-DD HH:MM:SS)} e módulo de monitoração{tipo de sensor (e.g.: temperatura), estado de operação(e.g.: ativar, ativo e desativado), intervalo de amostragem(HH:MM:SS) e data de configuração (AAAA-MM-DD HH:MM:SS)}
	Dados: tipo de dado (e.g.: temperatura), valor (e.g.: 15°C) e data de aquisição (AAAA-MM-DD HH:MM:SS)
API de Autenticação	Dados de Autenticação: credenciais do usuário e token de autenticação
Mod. de Processamento	Dados de Autenticação: número serial da sonda e token de autenticação
	Configurações: módulo de controle {estado de operação(e.g.: ativar, ativo e desativado), modo de operação (e.g.: periódica ou contínua),período de atividade (HH:MM:SS), período de inatividade (HH:MM:SS) e data de configuração (AAAA-MM-DD HH:MM:SS)} e módulo de monitoração{tipo de sensor (e.g.: temperatura), estado de operação(e.g.: ativar, ativo e desativado), intervalo de amostragem(HH:MM:SS) e data de configuração (AAAA-MM-DD HH:MM:SS)}
	Dados: tipo de dado (e.g.: temperatura), valor (e.g.: 15°C) e data de aquisição (AAAA-MM-DD HH:MM:SS)
	Instrução: acionamento (e.g.: ativar ou desativar um dispositivo) e leitura (e.g.: ler sensor de temperatura)
Mod. de Controle	Instrução: acionamento (e.g.: ativar ou desativar um dispositivo)
Mod. de Monitoração	Instruções: leitura (e.g.: ler sensor de temperatura)
	Dados: tipo de dado (e.g.: temperatura), valor (e.g.: 15°C) e data de aquisição (AAAA-MM-DD HH:MM:SS)

Fonte: Caio Cabezas

No sistema, as entidades Web Client e App Cliente respondem pela maior parte da interação com os usuários. No Web Cliente o usuário tem acesso aos recursos para o gerenciamento de sua conta e recursos para a visualização de dados coletados. No App Client o usuário tem acesso aos recursos de visualização de dados coletados e recursos para o usuário definir as configurações de operação de sondas.

A entidade API consiste em duas aplicações do tipo API hospedadas em um servidor na internet. Tem o propósito de servir as outras entidades as Web Client, App Client e Sonda com recursos para a autenticação de usuários e para a comunicação entre as entidades Web Client - App Cliente com a Sonda

A entidade Sonda possui três componentes, o módulo de processamento recebe as configurações de operação via API e as implementa, permitindo o controle sobre os recursos dos módulos de monitoração e controle. Este módulo envia os dados coletados pelo módulo de monitoração às entidades Web Client e App Cliente via API.

2.2. Requisitos do Sistema

A especificação de requisitos do sistema foi feita de acordo com a metodologia de desenvolvimento utilizada no projeto, a APF, que utiliza uma técnica de estrutura de detalhamento de requisitos para especificar os requisitos de um projeto. Esta metodologia será apresentada mais detalhadamente na seção 3.1. Para auxiliar a definição dos requisitos a serem detalhados, foram definidos um conjunto de requisitos funcionais do sistema.

Os requisitos funcionais de um sistema podem ser expressos em diferentes níveis de detalhamento e representam quais os serviços o sistema deve fornecer, a forma como o sistema deve reagir a entradas específicas e como ele deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também exemplificar o que o sistema não deve fazer (SOMMERVILLE, 2010). Idealmente, a especificação dos requisitos funcionais de um sistema deve ser completa (todos serviços e informações solicitados devem estar definidos nos requisitos) e consistente (os requisitos definidos não devem ser contraditórios) (SOMMERVILLE, 2010). Os requisitos funcionais definidos para o sistema Verde³ Watcher é exibido na Tabela 3.

Tabela 3 – Requisitos funcionais do sistema

Identificação o	Descrição
--------------------	-----------

RF01	O sistema deve permitir que o usuário mantenha uma conta
RF02	O sistema deve confirmar o cadastro do usuário via e-mail
RF03	O sistema deve permitir que o usuário possua múltiplas sondas vinculadas à sua conta
RF04	O sistema deve exibir ao usuário todas as sondas vinculadas à sua conta
RF05	O sistema deve permitir que o usuário configure o funcionamento de qualquer dispositivo conectado ao um módulo de controle
RF06	O sistema deve permitir que o usuário ative e desative qualquer dispositivo conectado ao um módulo de controle
RF07	O sistema deve permitir que o usuário configure o funcionamento de qualquer sensor de um módulo de monitoração
RF08	O sistema deve permitir que o usuário ative e desative qualquer sensor de um módulo de monitoração
RF09	O sistema deve permitir que o usuário visualize os últimos dados coletados pelos sensores ativos
RF10	O sistema deve permitir que o usuário visualize as séries históricas de dados coletados
RF11	O sistema deve permitir que as sondas operem somente quando estão conectadas à internet
RF12	O sistema deve sincronizar as configurações das sondas ativas com as configurações definidas pelo usuário

RF13	O sistema deve permitir que as sondas gerenciem a operação dos módulos de controle e de monitoração
RF14	O sistema deve permitir que as sondas enviem os dados coletados pelo módulo de monitoração de acordo com as configurações definidas pelo usuário

Desenvolvido por Caio Cabezas

3. Desenvolvimento

Este trabalho precisou do estudo de diversos conceitos que separados tem sua importância, mas quando juntados houve uma maximização de recursos trazendo abrangência e praticidade no manuseio e cultivo de hortas hidropônicas em diferentes ambientes.

3.1. Metodologia de Desenvolvimento do Projeto

3.1.1. APF

Para a gestão do desenvolvimento do projeto, foi implantada uma metodologia baseada no Adaptive Project Framework (Estrutura de Projetos Adaptável, do Inglês), ou APF, conforme desenvolvido e publicado por Robert K. Wysocki, Ph.D. em seu livro Adaptive Project Framework: Managing Complexity in the Face of Uncertainty (ou Estrutura de Projetos Adaptável: Gerenciando Complexidade Em Face da Incerteza). Trata-se de uma definição de princípios e processos a serem utilizados no gerenciamento de projetos altamente suscetíveis a mudanças. O APF permite maior flexibilidade na gestão de projetos, dentro de restrições de recursos (equipe, tempo e tecnologia), principalmente quando comparado a abordagens tradicionais (por exemplo, o Modelo Cascata).

Esta abordagem foi escolhida no meio do desenvolvimento do projeto, após o surgimento de diversas mudanças fundamentais na estruturação do projeto e de escopo, e a falta de clareza na definição da solução final. A abordagem utilizada até então consistia na especificação de requisitos funcionais, uma modelagem de escopo através de diagramas de contexto, e a divisão das tarefas utilizando a metodologia ágil Kanban.

No entanto, após o surgimento das mudanças supramencionadas, essa metodologia provou ser inadequada. A percepção do atual estado de desenvolvimento do projeto era nebulosa, gerava retrabalho, e não havia uma estrutura para especificação clara de objetivos, requisitos, e finalmente, a validação destes ao final do projeto.

Após novas pesquisas foi encontrado o APF, um modelo de gestão ágil de projetos baseado na adaptação às constantes mudanças de escopo, e elucidação iterativa de uma solução para um objetivo claramente definido. Sua estrutura tornava mais clara a contraposição de objetivos especificados no início do trabalho, com o produto entregue ao seu fim.

Após a escolha desse modelo, deu-se início às tarefas de desenvolvimento de uma metodologia para o gerenciamento do projeto, baseada no APF, adotando parcialmente suas filosofias e definições para o processo de desenvolvimento, conforme identificado necessário para o nosso projeto.

Essa adoção parcial de axiomas é sustentada pela filosofia de Wysocki (2010, p. 16, tradução nossa) sobre o APF, que como uma abordagem ágil, “deve se adaptar ao projeto. Ela não funcionará se forçar o projeto em seu próprio conjunto de regras e convenções”.

Adicionalmente, conforme acordado posteriormente sobre a entrega do software na condição de protótipo – isto é, aceita a ausência de funcionalidades necessárias para a plena distribuição e uso comercial da solução – a metodologia escolhida provou-se novamente adequada, dado o valor de número três (tratado na seção a seguir), e o posicionamento de Wysocki (2010, p. 277, tradução nossa) sobre soluções incompletas:

Por natureza, um projeto APF deixa um monte de buracos nos entregáveis, e vários gerentes sênior terão problemas ao justificar esses tipos de projetos. Se a

conclusão bem-sucedida do projeto é crítica ao negócio, e a solução de fato foi elusiva, então não há alternativa a não ser abordar o projeto como um projeto APF. Isso não elimina a necessidade de justificar o projeto à gerência sênior. Eles irão querer saber que o projeto não está apenas brigando contra moinhos de vento, e que não é um esforço que não irá produzir nada valor para o negócio. Se essa é a realidade, talvez o problema deverá se manter não resolvido!

Entretanto, é válido ressaltar que a presente solução busca ser uma solução funcional, ainda que a nível de protótipo, que agregue valor compatível com os objetivos e a descrição de produtos descritos no início deste trabalho. Não foi aberto espaço para muita experimentação adicional e nem foi contemplada a possibilidade de encerramento prematuro a fim de salvar recursos financeiros, como poderia ser o mais viável em um cenário corporativo de desenvolvimento de projetos.

O APF é um modelo iterativo, onde a cada nova iteração clientes e desenvolvedores revisam o processo e contemplam a possibilidade de alterar o rumo de desenvolvimento do projeto, alterar as definições de prioridade de tarefas e entregáveis, encerrar o projeto, ou deferir entregáveis para novas “versões”, como são chamados os projetos APF.

É ressaltado que o APF não impõe diretrizes estritas de como deve ser aplicado no gerenciamento de projetos. Wysocki (2010, p. 65, tradução nossa) muitas vezes se refere ao APF como uma “receita que não deve ser cegamente seguida”, e deve ser adaptada às necessidades do projeto, e nunca o contrário. Mais adiante, Wysocki defende que o APF sofra modificações para que se adapte ao contexto de desenvolvimento. Wysocki (2010, p. 65, tradução nossa) diz:

Você não vai encontrar listas intermináveis do que se fazer num projeto APF. Isso seria uma perda de tempo e o usuário APF não perde tempo. Como você poderá ler e estudar nas páginas seguintes, espere aprender que vocês, como co-gerentes do APF, precisam começar a pensar sobre o que estão fazendo e como estão trabalhando juntos. Se algo que estão ambos fazendo não faz sentido, provavelmente deve ser mudado ou não deveria sequer ser feito.

Com isso, o APF se define como um guia relativamente solto, um modo de pensamento sobre clientes e como melhor lhes servir valor. Wysocki (2010, p. 68, tradução nossa) diz que “o cliente é o centro das atenções num projeto APF. O cliente

controla a direção do projeto e determina onde valor para o negócio pode ser criado ou aumentado”.

Wysocki (2010, p. 68, tradução nossa) ainda consolida esse modo de pensamento através da definição de seis valores principais, os quais não devem ser exceção para nenhum projeto APF, e devem ser seguidos toda vez:

1. Focado no cliente: As necessidades do cliente sempre devem vir primeiro, contanto que estejam dentro dos limites éticos das práticas de negócios. O valor nunca pode ser comprometido, e times APF devem ir além de simplesmente mantê-lo em mente: Deve ser óbvio através de suas interações com seus colegas do time de desenvolvimento, e todos os membros do time do cliente.

2. Dirigido pelo cliente: Ao extremo, esse valor significaria ter o cliente assumindo o papel e responsabilidades do gerente de projeto.

3. Resultados incrementais cedo e frequentes: No espírito da prototipagem, em um projeto APF você quer entregar uma solução funcional para o cliente o quanto antes e frequente possível. Ela deve entregar valor mesmo que seja através de funcionalidade limitada.

4. Introspecção e questionamento contínuos: Esse valor chave se refere à abertura e honestidade que deve existir entre o Time de Cliente e o Time de Desenvolvimento. Ambos devem estar comprometidos em fazer as melhores decisões de negócio possíveis.

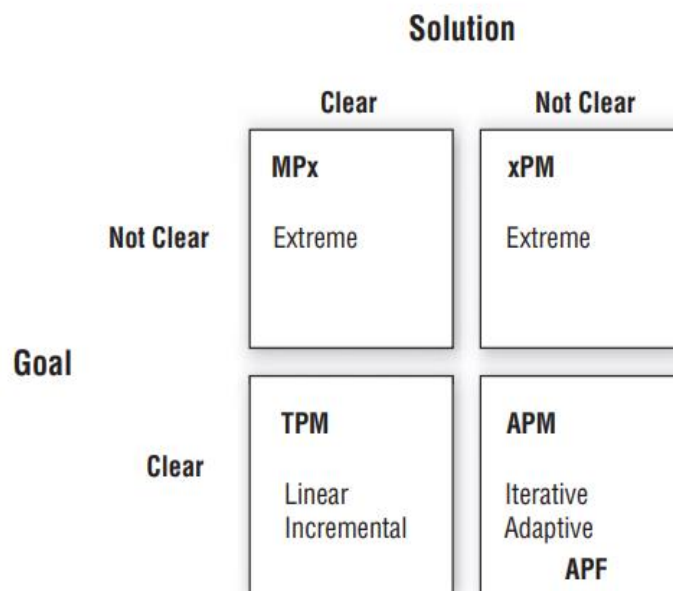
5. Mudança é o progresso para uma solução melhor: Enquanto mudança é necessária para se alcançar a melhor solução, muita mudança passa uma mensagem muito diferente.

6. Não especule demais sobre o futuro: Haverá sempre a tentação de envisionar algum estado ideal e se convencer de que atingi-lo é algo realista. Um time APF deve resistir à essa tentação. O APF retira todo o trabalho que não agrega valor. Adivinhar apenas adiciona este trabalho de volta. Quando em dúvida, deixe de fora.

3.1.2. Objetivo e Solução

O APF é considerado um modelo ágil, mas que se difere de outras abordagens como o Extreme ou Scrum. Segundo Wysocki (2010, p. 34, tradução nossa), o “APF é projetado para aqueles projetos onde o objetivo está claramente declarado e documentado, mas a solução é apenas conhecida parcialmente [...]”. A posição do APF dentro dos quadrantes de definição de objetivo e solução é elucidada pela Figura 9:

Figura 9 – Escopo de definição da solução.



Fonte: WYSOCKI (2010)

3.1.3. Gestão

Projetos APF designam dois co-gerentes para o projeto APF: um Cliente, e um responsável pelo time de Desenvolvimento. Como definido por Wysocki (2010, p. 31, tradução nossa) “o time do Cliente deve ter um único membro responsável pela autoridade de tomada de decisões. Essa pessoa irá servir como co-gerente durante todo o projeto, juntamente com Líder do time de desenvolvimento.”

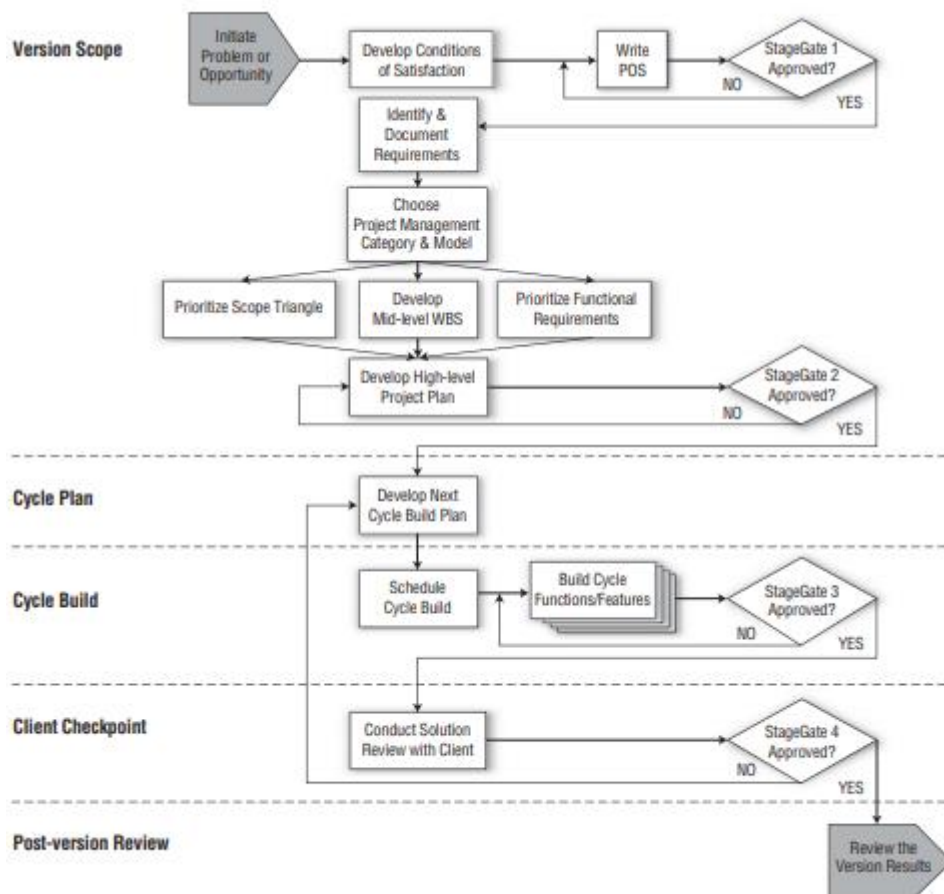
Um projeto APF valoriza o envolvimento ativo do cliente durante o seu desenvolvimento. Segundo Wysocki (2010, p. 63, tradução nossa):

Seria tolice usar uma abordagem APF na ausência de um envolvimento significativo do cliente. Na verdade, sem envolvimento significativo, seria arriscado assumir qualquer projeto, independente do modelo usado.” (WYSOCKI, 2010, p. 63, tradução nossa). Todo PMLC requer algum nível de envolvimento do cliente. Para um projeto APF, no entanto, há muito a se dizer sobre o papel do cliente co-gerente e o papel do desenvolvedor de projeto co-gerente. É melhor pensar sobre ambos como sendo compartilhadores da responsabilidade de gerente de projeto – eles se tornam co-gerentes do projeto, com responsabilidades distintas.

3.1.4 O Ciclo de Vida APF

O ciclo de vida APF é composto de cinco fases, comum aos projetos que o implementam (seja sobre parte do projeto, ou sobre sua totalidade), conforme ilustrado na figura abaixo:

Figura 10 – O ciclo de vida APF.



Fonte: WYSOCKI (2010).

3.1.4.1. Version Scope (ou “Escopo de Versão”)

Nessa fase, os times de Cliente e Desenvolvimento reúnem-se pela primeira vez para discutir ideias, avaliar propostas de escopo, levantar as características fundamentais do projeto e elucidar requisitos. Inicialmente, é identificada um problema ou oportunidade de negócio, em cima da qual a solução será desenvolvida.

Um projeto APF é nomeado “versão”, uma vez que cada projeto que implemente o modelo, pode ser modificada e revisitada, a fim de se agregar ainda mais valor à solução. Wysocki (2010) diz que escolheu usar o termo “versão” para indicar que um projeto subsequente talvez possa levar a solução para o próximo nível de desenvolvimento.

3.1.4.1.1. POS

O primeiro entregável dessa fase, proveniente das primeiras reuniões entre os times, são as Condições de Satisfação do projeto. Isto é, uma lista de critérios que devem ser alcançados pela solução para considerá-la de valor. Essas condições são então utilizadas para dar origem ao primeiro documento da versão, o Project Overview Statement (Declaração de Visão Geral de Projeto, do Inglês), ou POS, cujo papel é servir como manifesto das vontades do cliente para o projeto a ser desenvolvido.

O POS é um documento de poucas ou uma página, a conter cinco seções:

1. Uma declaração sobre o problema ou oportunidade (razão para desenvolver o projeto);
2. Uma declaração de objetivo;
3. Resultados quantificáveis de negócio (que valor de negócio será obtido);
4. Comentários gerais sobre riscos, premissas, e obstáculos para o sucesso;

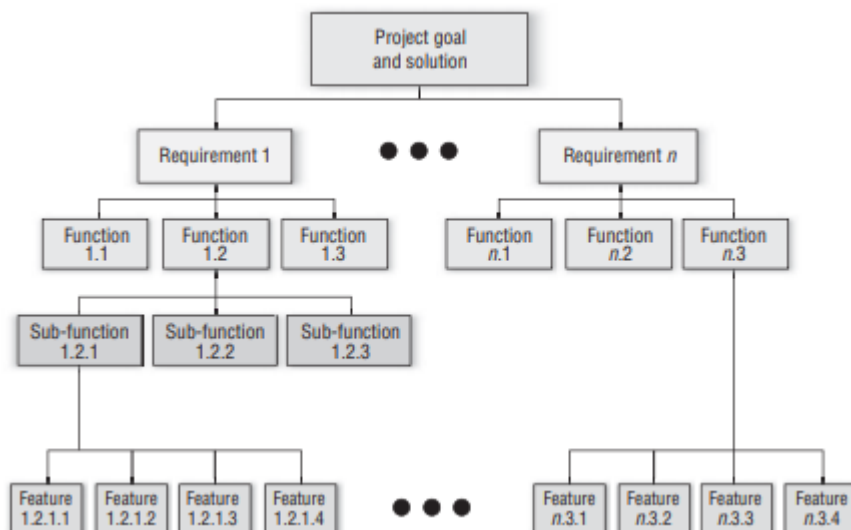
Após a definição do POS, este passa por um processo de aprovação através de uma nova reunião entre o Time de Desenvolvimento e o Time do Cliente, onde este último decide por concordar com o que foi desenvolvido no POS ou não, e se deve ou não dar início ao desenvolvimento do projeto.

Desse documento será desenvolvida uma lista de requisitos priorizada, onde os componentes fundamentais, conforme definidos nas Condições de Satisfação do sistema, irão receber graus de prioridade que serão usados para nortear o desenvolvimento ao longo do projeto.

3.1.4.1.2. RBS e WBS

Os entregáveis que sucedem o POS são a especificação do *Requirement Breakdown Structure* (Estrutura de detalhamento de requisitos, do Inglês), ou RBS, e o *Work Breakdown Structure* (Estrutura de detalhamento de tarefas, do Inglês), sendo este último depende do primeiro, e sendo comparável ao EAP (Estrutura Analítica do Projeto), onde os componentes da solução serão definidos, destrinchados (detalhados), e enfim, transformados em tarefas tangíveis de acordo com a figura 11:

Figura 11– RBS em exemplo.



Fonte: WYSOCKI (2010)

Enquanto o RBS é delegado exclusivamente à definição dos requisitos da solução, organizadas em componentes (“Funções” e “Subfunções”) em hierarquia. Definidos geralmente em três níveis de detalhamento, pode variar em complexidade de acordo com o quanto da solução é conhecida naquele momento.

Wysocki (2010, p.123, tradução nossa) adverte o uso de RBS pelas seguintes razões:

1. O RBS é mais significativo para o consumidor;

2. O RBS é uma abordagem baseada em entregáveis;
3. O RBS pode ser utilizado para mensurar o estado do projeto;
4. O RBS é consistente com o PMBOK;
5. O RBS mantém-se orientado ao consumidor o quanto possível dentro do exercício do planejamento;

O RBS elucida os componentes do projeto, utilizando uma semântica baseada em verbos para designar funções. Seu estado não é reconhecido como completo nesse momento do projeto. Mudanças podem e devem ser introduzidas à medida que a solução seja descoberta nas fases subsequentes à de *Version Scope*.

O RBS é então a principal entrada para a produção do WBS (ou EAP) de nível médio, onde os componentes da solução são transformadas em conjuntos de trabalho, a serem destrinchados em tarefas menores. O WBS segue uma estrutura semântica baseada em Substantivos para a definição de *features* (funcionalidades, do Inglês), e Verbos para a definição de atividade oriundas do detalhamento das funcionalidades. É importante ressaltar que o WBS também não é completo nesse estado do projeto, sofrendo alterações à cada ciclo, conforme mais da solução seja explorada, ou modificada para obedecer às novas concepções do Cliente e Time de Desenvolvimento, de acordo com a figura:

Figura 12 – WBS em exemplo.

Requirement	Function	Sub-function	Feature
Materials	Student workbooks		
	Case studies		
	Exercises		
	Slide presentations		
	Instructor workbook		
	References		
	Evaluations		
		Course materials	
		Instructor	
		Participants (pre & post)	

Fonte: WYSOCKI (2010)

3.1.4.1.3. Triângulo de Escopo

Nesta etapa, é utilizada uma representação geométrica das cinco variáveis de consideração da gerência do projeto definidas pelo APF: Escopo, Qualidade, Tempo, Custo e Disponibilidade de Recursos.

Figura 13 – Triângulo de Escopo.



Fonte: WYSOCKI (2010)

A representação em triângulo representa a relação de conflito entre as variáveis, onde o peso atribuído à umas dessas deve se manter equilibrado em relação à todas as outras. Cada projeto pode ter, de acordo com seu contexto, diferentes prioridades para cada uma dessas variáveis. Assim sendo, uma relação de priorização dessas variáveis é construída, a fim de explicitar a decisão dos gerentes do projeto sobre quais variáveis devem ser priorizadas no desenvolvimento daquela solução, conforme figura:

Figura 14 – Exemplo de priorização de variáveis do triângulo de escopo.

		Priority				
		Critical (1)	(2)	(3)	(4)	Flexible (5)
Variable	Scope				X	
	Quality			X		
	Time	X				
	Cost					X
	Resource Availability		X			

Fonte: WYSOCKI (2010)

3.1.4.1.4. Lista Priorizada de Requisitos

Esta etapa é seguida pela definição da lista de requisitos priorizados, onde os requisitos principais identificados pelos primeiros níveis de detalhamento do RBS, assim como suas subfunções, são caracterizadas por um nível de prioridade, um número inteiro de um a dez, do mais prioritário ao menos prioritário, respectivamente. Essa definição, em conjunto com a lista de variáveis priorizadas da seção anterior, servirão para nortear o rumo de desenvolvimento do projeto e definição de tarefas (reservado para as próximas fases).

3.1.4.1.5. Plano de Projeto em Alto Nível

Como último entregável desta fase, é confeccionado um plano de projeto em alto nível, onde são definidos os ciclos, sua duração (geralmente, são alocadas duas semanas para cada ciclo, a depender das necessidades do projeto). Também são definidas outras variáveis iniciais, como orçamento total do projeto.

3.1.4.2. Cycle Plan (ou “Planejamento de Ciclo”)

A fase de planejamento de ciclo é a primeira fase iterativa do modelo APF, se repetirá para todo ciclo agendado no Plano de projeto de alto nível (vide seção anterior). Nessa fase são definidas as tarefas a serem desenvolvidas no ciclo, os seus responsáveis dentro do time de desenvolvimento e duração estimada. As tarefas são obtidas do WBS de nível médio, que começa a sofrer alterações para se tornar a cada ciclo, mais completo.

Esta fase recebe como entrada: o POS, o WBS, a priorização das variáveis do Triângulo de Escopo, e os pontos de discussão e descobertas levantadas no ciclo anterior. Também são tratadas as entregas planejadas, mas não realizadas no ciclo anterior.

Para facilitar o compartilhamento de tarefas na eventual indisponibilidade ou realocação de componentes do grupo, o APF define a criação de *Work Packages* (ou Pacotes de Tarefas, em tradução livre), onde cada tarefa extraída do WBS é exposta numa matriz de atribuições (com o responsável designado) e uma breve descrição, que facilite a compreensão em detalhes da tarefa pelos outros componentes de grupo, sendo útil em eventuais realocações de tarefas.

3.1.4.3. Cycle Build (ou “Construção de Ciclo”)

Nesta fase são construídas as partes (componentes) planejadas para o ciclo, segundo o cronograma. É importante estipular prazos realistas e não esticar o tempo planejado para este ciclo. Tarefas não concluídas devem ser enviadas ao Banco de Escopo, para serem persistidas no próximo ciclo ou abandonadas, conforme decidido pelo cliente e co-gerente de desenvolvimento.

Problemas são encaminhados ao *Issues Log* (Registros de Problemas, em tradução livre), onde devem ser tratados durante o presente ciclo, idealmente.

Esta fase é, assim como o Planejamento de Ciclo, reiterada para quantos ciclos especificados no Plano de projeto em alto nível ([seção 1.3.1.4.](#)).

3.1.4.4. Client Check Point (ou “Ponto de Verificação do Cliente”)

Ao final do ciclo, todos os componentes se reúnem para discutir seus entregáveis, os pontos (problemas, modificações, ou novas descobertas) levantadas durante o ciclo. O estado do projeto é avaliado pelo cliente, que aprova o que foi desenvolvido, ou é fomentada uma discussão para a resignificação ou até abandono do projeto, para evitar o desperdício de mais recursos.

Trata-se da fase de encerramento do ciclo, e, portanto, decisões importantes de negócio devem ser feitas. É preciso saber identificar o estado atual do projeto nessa fase, e como todos estão contribuindo para o desenvolvimento da solução.

3.1.4.5. Post-Version Review (ou “Revisão Pós-Versão”)

Ao término do desenvolvimento cíclico, o Time de Cliente e Time de Desenvolvimento se reúnem para avaliar o que foi desenvolvido na versão atual. Wysocki (2010, p. 254, tradução nossa), define que:

A conclusão da fase de Revisão Pós-versão marca o encerramento formal ao trabalho na versão atual da solução. É aqui que são taxados o valor de negócio entregue de fato versus o que foi planejado. Se houver discrepância, pode haver razão para o comissionamento de um novo projeto APF para criar uma solução de nova geração para fechar esta lacuna.

Wysocki (2010, p. 256, tradução nossa) também define que nesta fase, procura-se responder às seguintes questões:

1. O time encontrou uma solução aceitável?
2. Os critérios de sucesso foram atingidos?
3. É recomendada uma segunda versão?

Procuram-se também levantar (assim como na fase *Client Checkpoint* de cada ciclo) problemas e sugestões de melhoria ao processo, e o que pode ser feito para tornar a APF mais eficiente, bem como o quão bem o time conseguiu executar a metodologia.

3.1.4.6. Post-Version Report (ou “Relatório Pós-Versão”)

Este documento é a síntese das considerações levantadas ao final do projeto, com o propósito de expor, para posteridade, como o APF é aplicado em um cenário real, bem como servir como manifesto para as descobertas realizadas durante o desenvolvimento da solução.

De formato livre, o Relatório pós-versão deve incluir, segundo Wysocki (2010, p. 264, tradução nossa), em seu conteúdo, comentários sobre as seguintes questões:

1. Sucesso Geral do Projeto;
2. Organização do Projeto;
3. Técnicas Usadas Para obter Resultados;
4. Pontos Fracos e Fortes do Projeto;
5. Recomendações ao Time de Projetos;

3.1.5. Implementação do APF

Para o nosso projeto, foram decididos quais elementos seriam herdados da especificação original do APF por Wysocki, e quais não. Essa decisão, baseada no princípio de flexibilidade do próprio APF, decidiu por abolir entregáveis e cerimônias que foram identificadas como não necessárias para o desenvolvimento, mantendo alguns princípios fundamentais da metodologia.

É válido ressaltar que a condição do presente projeto como trabalho de graduação, sujeito à inconclusões e limitações de escopo, implica no deferimento da implantação da solução final aos últimos ciclos, que diferentemente do esperado para o APF, não estaria pronta para aplicação em produção à cada ciclo, somente com nível de complexidade diferente. Ao invés disso, buscou-se entregar uma solução, mais detalhada a cada ciclo, porém valiosa somente à nível de desenvolvimento, e sob a consideração do projeto como à nível de protótipo.

Dadas as condições de trabalho do time, condicionado ao trabalho e cooperação remotos, foi decidido que as reuniões seriam realizadas por videoconferência e ligações telefônicas, com o auxílio de plataformas digitais para o compartilhamento de informação e documentos, bem como a confecção de entregáveis.

3.1.5.1. Gestão

Para os papéis de co-gerentes, representando os Times do Cliente e Time de Desenvolvimento, respectivamente, foram designados os componentes do grupo Caio C. Aragon e Lucas D. Monteiro.

O Time de Desenvolvimento foi composto por todos os outros integrantes de grupo, porém também pelos co-gerentes supracitados. O papel duplo destes dois componentes é justificado pela escassez de recursos humanos para uma atribuição tradicional, mais separada, desses papéis.

As responsabilidades de Clientes foram atribuídas ao supracitado Caio Aragon devido ao seu conhecimento prévio do domínio do projeto, a Hidroponia, bem como por este ter idealizado e proposto a solução, originalmente. Tido como um ponto à favor dessa decisão foi o maior envolvimento do Cliente com o desenvolvimento do projeto (como defendido nos princípios do APF), conseqüente de sua dupla responsabilidade como Cliente co-gerente e desenvolvedor.

Logo, as atribuições de responsabilidade do projeto podem ser visualizadas na matriz à seguir:

Tabela 4 - Atribuição de papéis APF

Componente	Papel(is)
-------------------	------------------

Alecsander T. Fernandes	Desenvolvedor
Caio C. Aragon	Co-gerente Cliente / Desenvolvedor
Dennys M. D. Braghirolli	Desenvolvedor
Lucas D. Monteiro	Co-gerente Desenvolvimento / Desenvolvedor
Silas C. de Moraes	Desenvolvedor

Desenvolvido por Lucas Monteiro

3.1.5.2. Escopo da Versão

Durante esta fase, conforme especificado pela APF foram levantadas as condições de satisfação e definições gerais sobre o escopo do projeto, como objetivo geral, objetivos específicos e critérios para o sucesso da versão. Estas definições foram sintetizadas no documento “Declaração de Visão Geral do Projeto”, ou POS (apêndice A).

Em seguida, foram confeccionados os entregáveis esperados para essa fase, listados a seguir:

1. As estruturas RBS (apêndice B) e WBS (apêndice C): Foram construídas em lista, para permitir maior flexibilidade, facilitar a visualização e edição simultânea por mais de um componente de grupo.

2. Priorização das variáveis do triângulo de escopo (apêndice D): Construída em matriz, expondo a atribuição de prioridade dada para este projeto à cada uma das cinco variáveis de escopo definidas pela APF.

3. Lista priorizada de requisitos (apêndice D): Neste documento, os componentes principais da solução (Aplicativo, Website e Sonda), são qualificados com um nível de prioridade de um a dez, do mais ao menos prioritário, respectivamente, de acordo com sua importância para a solução.

4. Plano de projeto em alto nível (apêndice E): Nesse documento, é criado o cronograma de ciclos que irão ser iterados durante a versão.

3.1.5.3. Gestão de Ciclos

Para cada ciclo, foram reservados os primeiros dias de sua duração para seu planejamento, feito em reunião por videoconferência, atendida obrigatoriamente pelos co-gerentes designados, onde se sucederam as seguintes tarefas:

1. Revisão e detalhamento adicional do WBS: O WBS é revisado pelos componentes, e as funções e subfunções são divididas em tarefas tangíveis.

2. Cronograma das tarefas: As atividades elucidadas no WBS são atribuídas para os componentes do grupo, com confecção de cronograma.

3. Construção de *Work Packages*: O *Work Packages* (Pacotes de Tarefas), feitos somente até o terceiro ciclo, forneceram maiores detalhes sobre as tarefas elucidadas, mas foram logo dispensadas pois provaram-se desnecessárias para o alinhamento do grupo, feito já durante as videoconferências e comunicação instantânea entre os integrantes.

A fase Client Checkpoint de cada ciclo também foi realizada via videoconferência, com a obrigatoriedade de participação de todos os integrantes, para discutir seus entregáveis, expor dificuldades encontradas e descobertas. Ainda durante as iterações dessa fase, foram feitas considerações para os ciclos subsequentes e acordos sobre a manutenção ou abandono de entregáveis.

3.1.5.4. Relatório Final

Este artefato final foi desenvolvido como parte desta monografia, integrando a seção 5. Discussão e Conclusão, uma vez que seu conteúdo está relacionado com o tópico.

3.1.5.5. Software de apoio à gestão APF

Para apoiar a gestão do projeto, e produção dos entregáveis APF, foram utilizados softwares de terceiros, buscando a viabilização da cooperação integralmente remota dos componentes do grupo.

3.1.5.4.1 Google Drive

Trata-se do software para armazenamento pessoal de informações, vinculadas à uma conta Google. Esta plataforma foi utilizada para a persistência e acesso dos componentes aos documentos produzidos durante o projeto.

3.1.5.4.2 Google Docs

Acessível diretamente de, e nativamente integrado à plataforma supracitada, esta plataforma foi utilizada por permitir a edição simultânea de documentos por dois ou mais membros do grupo (desde que utilizando uma conta Google).

3.2. Protótipos

As definições da palavra protótipo segundo o dicionário Houaiss (online) são “1. primeiro tipo criado; original; 2. algo feito pela primeira vez e, muitas vezes, copiado ou imitado; modelo, padrão, cânone; 5. versão preliminar, ger. reduzida, de um novo sistema de computador ou de um novo programa, para ser testada e aperfeiçoada”. Segundo o livro Design Thinking – Inovação em negócios¹, “o protótipo é a tangibilização de uma ideia, a passagem do abstrato para o físico de forma a representar a realidade – mesmo que simplificada – e propiciar validações.”.

Ainda segundo o Houaiss, a palavra protótipo etimologicamente vem do grego *prōtótupos,os,on* no sentido de 'o primeiro tipo, de criação primitiva, primitivo', pelo lat. *prototypos,a,um* no sentido de 'protótipo, elementar; t. de gram. primitivo'; ver *prot(o)-* e *-típo*; f.hist. 1686 *prototypo*.

De acordo com CARAZATO (2017):

Possuir um processo de desenvolvimento de software é imprescindível para que a qualidade do software seja o objetivo, visto que a indústria desta área está em crescente desenvolvimento, tornando o mercado mais competitivo. Sendo assim, o presente trabalho propõe o desenvolvimento de um modelo de prototipação que possa ser integrado nos diversos modelos de desenvolvimento de software, uma vez que a prototipação é uma técnica que

minimiza muitos problemas e limitações nos processos de desenvolvimento de software, levando em consideração as melhores práticas, os variados tipos de equipes e as diferentes culturas organizacionais

Segundo Pressman (2016)

O paradigma de prototipação costuma ser a melhor escolha de abordagem quando: 1. O cliente define uma série de objetivos gerais para o software, mas não identifica, de forma detalhada, os requisitos para funções e recursos; 2. O desenvolvedor encontra-se inseguro quanto à eficácia de um algoritmo ou quanto à adaptabilidade de um sistema operacional; 3. Quando há dúvida quanto à forma em que deve ocorrer a interação homem/máquina

3.2.1. Protótipos da Página Web

Um dos softwares utilizados para construir o Mockup do projeto foi o Balsamiq, onde é possível criar a interface do sistema.

Este software foi escolhido devido a gama de opções para se criar rascunhos, é de fácil aprendizagem, e muito maleável no sentido de poder desenvolver a interface com detalhes, por ele ser online é possível ainda acessá-lo de qualquer computador desde que haja internet.

Ele possui limitações de cores por exemplo, mas com ele é possível desenvolver com riqueza de detalhes todos os componentes do software (texto, botões, imagens, configuração da estrutura das páginas). A experiência foi muito rica e proveitosa, graças a interface simples e a grande interação.

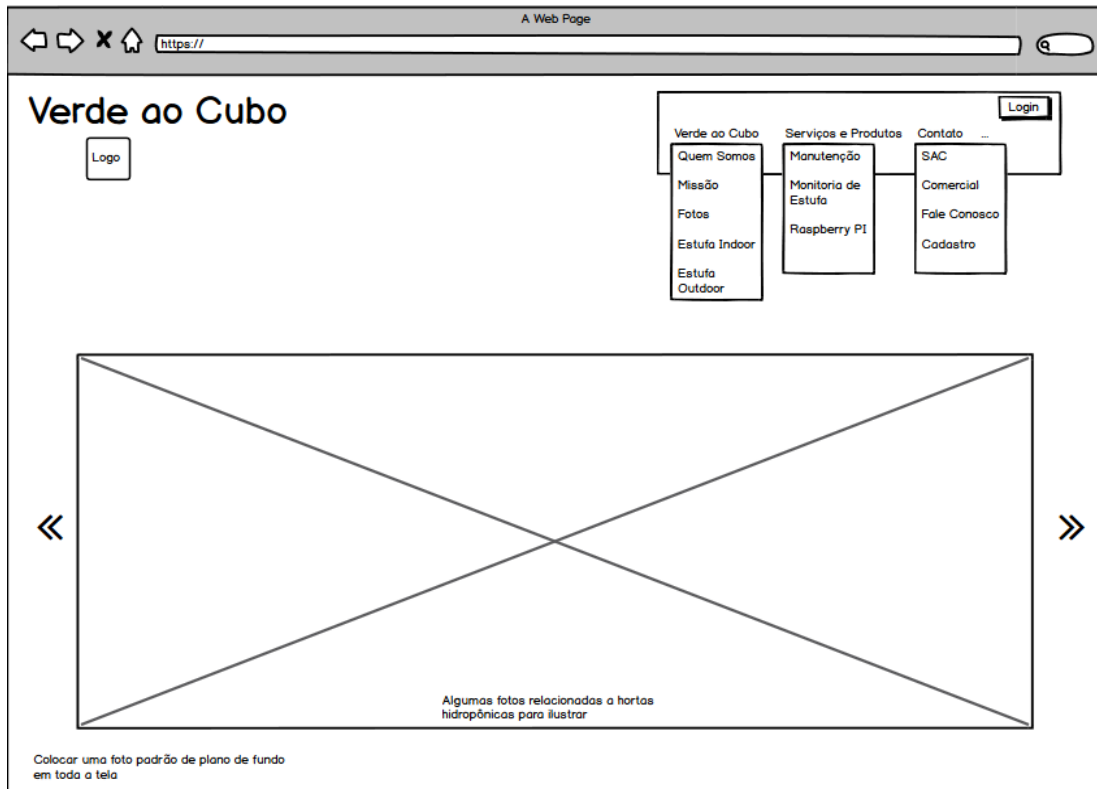
A princípio, a ideia do grupo foi fazer todo o sistema em website, mas depois, foi escolhido desenvolver em mobile todo o sistema e o site ficou como uma extensão do aplicativo de celular, para apresentar a empresa, o sistema e levar o cliente ao modo mobile para uso. A escolha da mudança foi feita devido às grandes possibilidades de se controlar a horta hidropônica onde a pessoa estivesse, com todos os controles a mão, seria mais prático checar o que estava acontecendo e alterar os parâmetros desejados.

Dessa forma foi construído uma primeira versão do website e depois conforme as alterações necessárias, foi desenvolvida a segunda versão como apoio ao aplicativo mobile.

Podemos observar nas figuras os 2 protótipos de telas feitos através do site BalsamiqCloud.com

3.2.1.1. 1º Protótipo

Figura 15 – Home (* não está logado).



Desenvolvido por Dennys Braghirolli 2020.

Figura 16 – Tela de Cadastro.

Cadastro

Nome

CPF

Endereço

CEP

E-mail

Celular

Usuário

Senha

Digite senha novamente

Conforme desce a barra de rolagem e muda a seção, muda-se também a foto do plano de fundo.

Desenvolvido por Dennys Braghirolli 2020

Botão Limpar: o botão limpar vai apagar os dados digitados.

Botão Cadastrar: esse botão vai finalizar e gravar os dados digitados.

Figura 17 – Tela de Login.

A Web Page

https://

Verde ao Cubo

Logo

Verde ao Cubo Serviços e Produtos Contato ...

- Quem Somos
- Missão
- Fotos
- Estufa Indoor
- Estufa Outdoor

Manutenção

- Monitoria de Estufa
- Raspberry PI

SAC

- Comercial
- Fale Conosco
- Cadastro

Login

Usuário

Senha

Desenvolvido por Dennys Braghirolli 2020

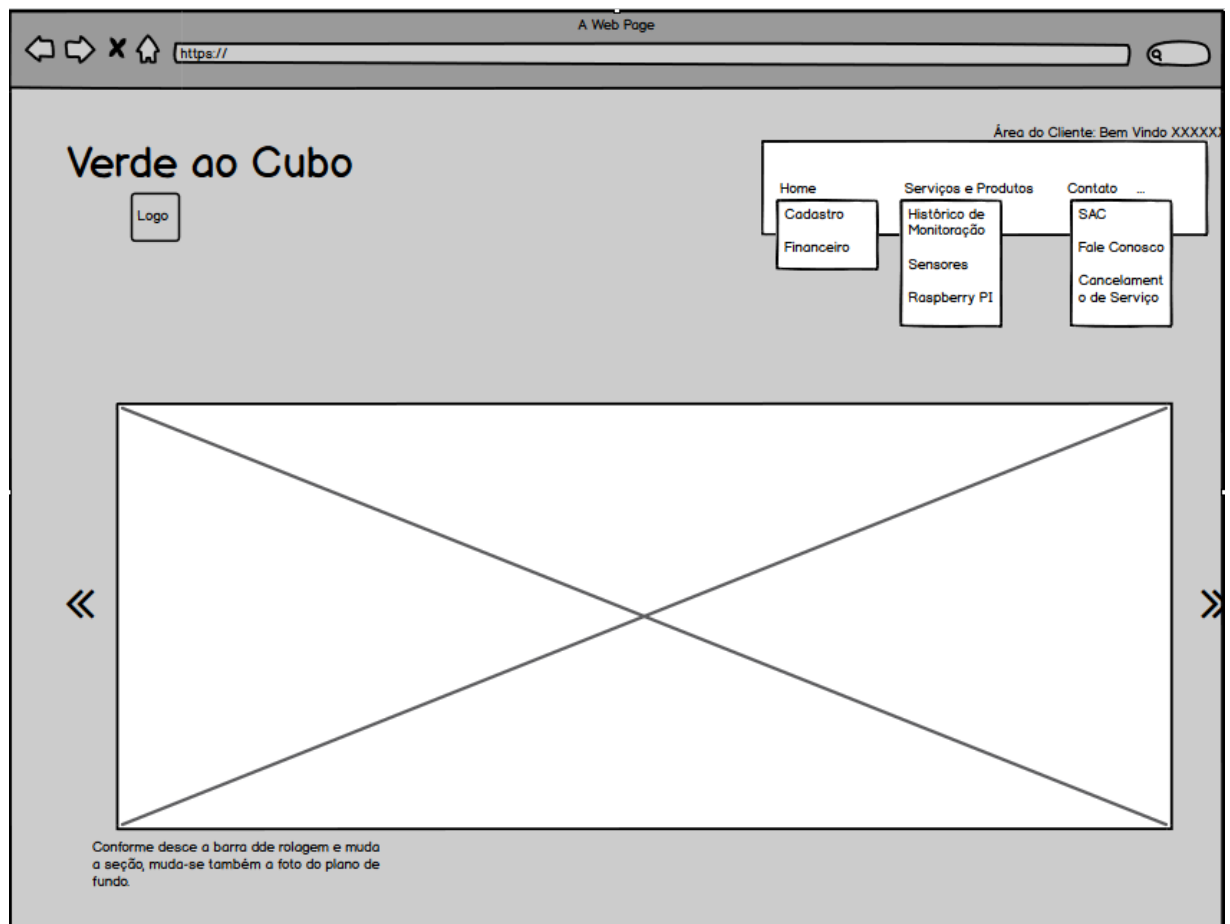
Botão Entrar: após confirmação de usuário e senha esse botão vai liberar o acesso ao sistema.

Botão Esqueceu sua senha: ele envia ao usuário uma nova senha para o e-mail cadastrado.

Botão Limpar.

Botão Cadastrar.

Figura 18– Home Logado.



Desenvolvido por Dennys Braghirolli 2020.

Figura 19 – Alteração de Cadastro.

Verde ao Cubo

Logo

Área do Cliente: Bem Vindo XXXXX

Home Cadastro Financeiro Serviços e Produtos Histórico de Monitoração Sensores Raspberry P1 Contato ... SAC Fale Conosco Cancelament o de Serviço

Conforme desce a barra dde rolagem e muda a seção, muda-se também a foto do plano de fundo.

Alteração de Cadastro

Nome

CPF

Endereço

CEP

E-mail

Celular

Usuário

Senha

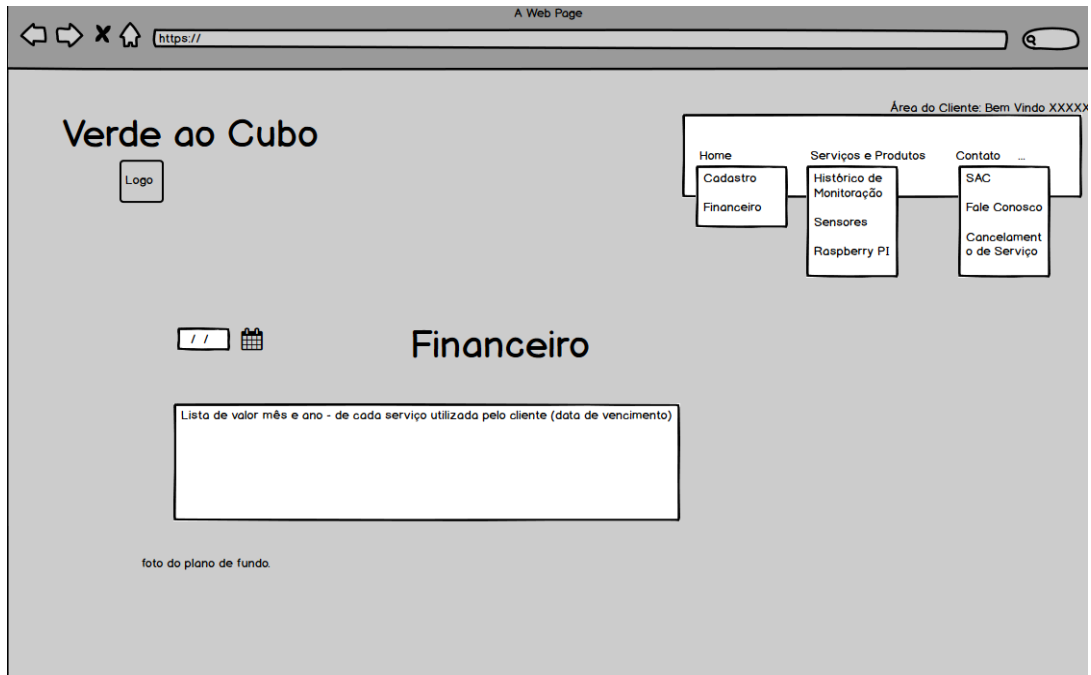
Digite senha novamente

Conforme desce a barra dde rolagem e muda a seção, muda-se também a foto do plano de fundo.

Limpar Editar Gravar

Desenvolvido por Dennys Braghirolli 2020.

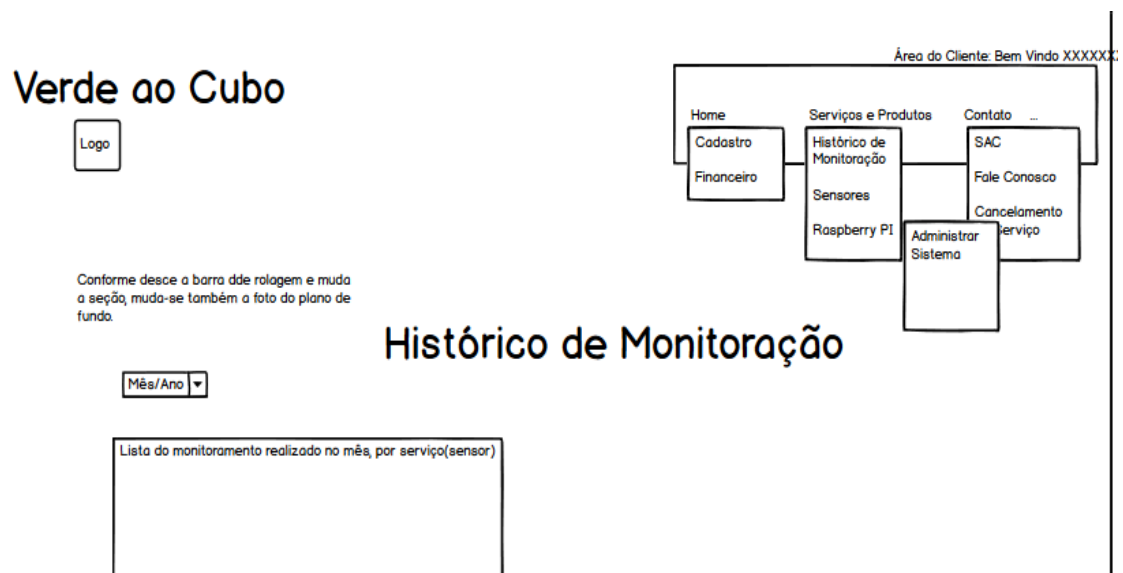
Figura 20 – Tela Financeira.



Desenvolvido por Dennys Braghirolli 2020.

Botão Data: colocar a data e verificar o valor pago de acordo com o serviço utilizado.

Figura 21 – Histórico de Monitoração



. Desenvolvido por Dennys Braghirolli 2020.

Botão Mês/Ano: abre o histórico de monitoração do referido período.

Figura 22 – Fale Conosco.

Fale Conosco

Escreva aqui sua dúvida, crítica ou sugestão.

Desenvolvido por Dennys Braghirolli 2020.

Botão Enviar: após digitar informações, dúvidas, sugestões ou reclamações, esse botão vai gravar a mensagem e enviar para o Verde ao Cubo.

Figura 23 - Cancelar Assinatura.

Cancelamento de Serviço

Lista de Serviços utilizados
Valores a quitar

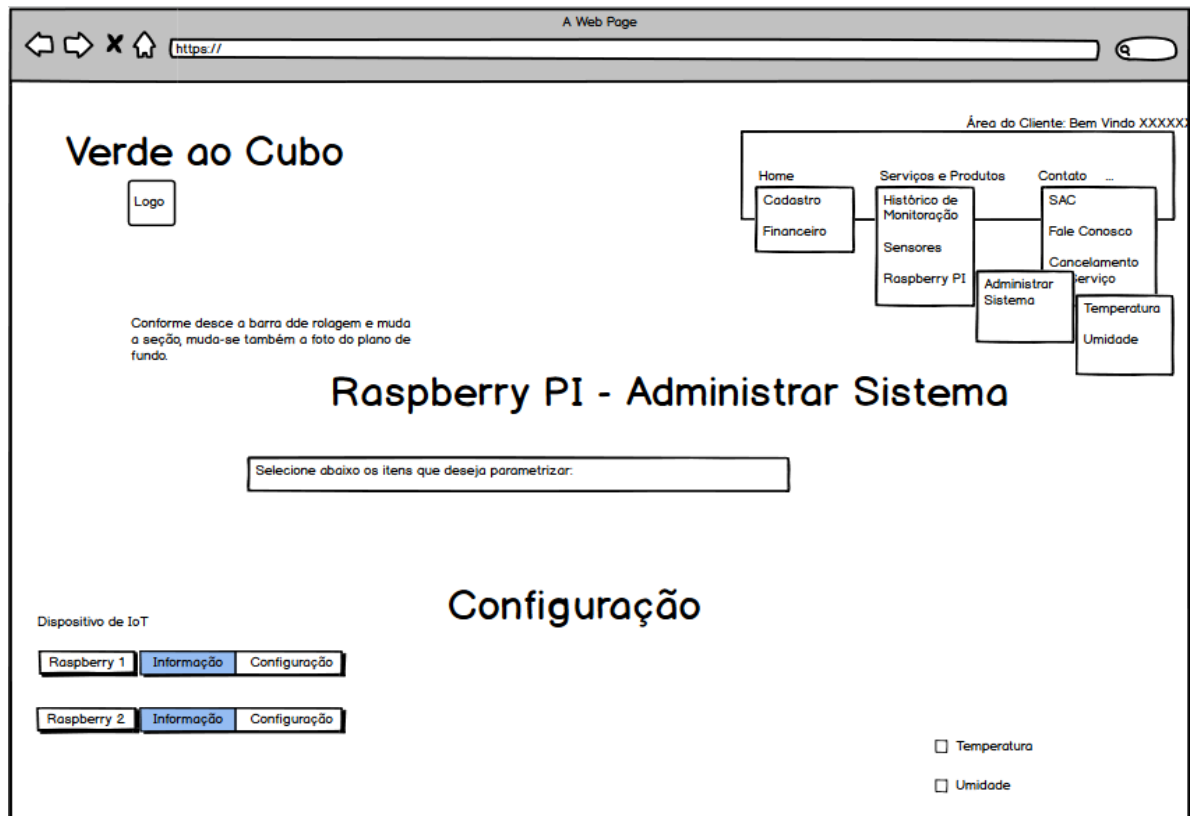
· Cancelamento será efetuado somente após a quitação de todo o serviço.

Conforme desce a barra de rolagem e muda a seção, muda-se também a foto do plano de fundo.

Desenvolvido por Dennys Braghirolli 2020.

Botão Cancelar Assinatura: após clicar nesse botão, aparecerá, caso exista, valores a serem quitados. Caso não haja, o cliente receberá em seu e-mail o cancelamento do serviço com um número de protocolo.

Figura 24 – Configuração.



Desenvolvido por Dennys Braghirolli 2020.

Raspberry 1 Informação: botão para selecionar a guia informação.

Raspberry 1 Configuração: botão para selecionar a guia configuração.

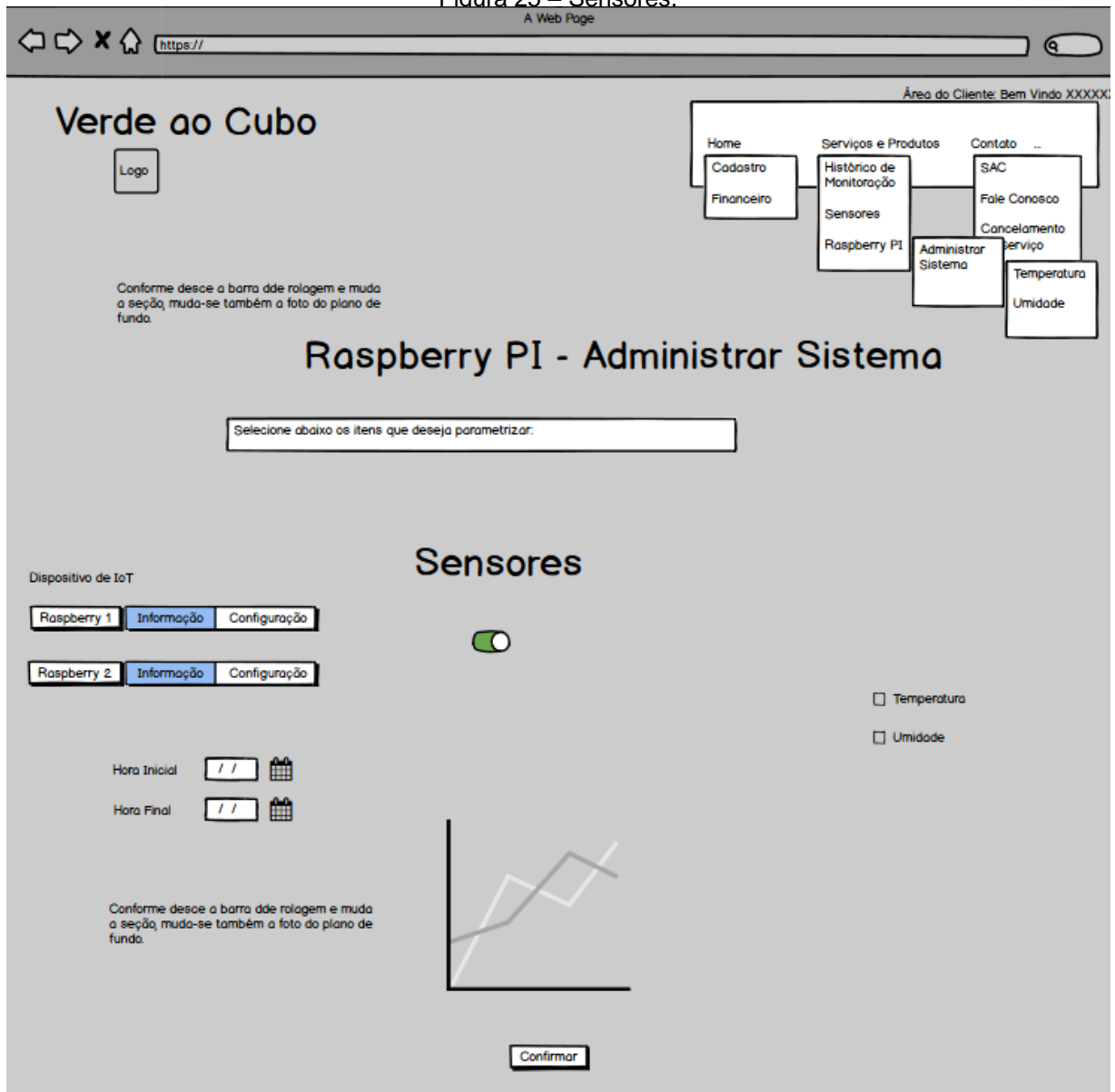
Raspberry 2 Informação: botão para selecionar a guia informação.

Raspberry 2 Configuração: botão para selecionar a guia configuração.

Botão Temperatura: botão de seleção do sensor de temperatura

Botão Umidade: botão de seleção do sensor de umidade

Figura 25 – Sensores.



Desenvolvido por Dennys Braghirolli 2020.

Raspberry 1 Informação: botão para selecionar a guia informação.

Raspberry 1 Configuração: botão para selecionar a guia configuração.

Raspberry 2 Informação: botão para selecionar a guia informação.

Raspberry 2 Configuração: botão para selecionar a guia configuração.

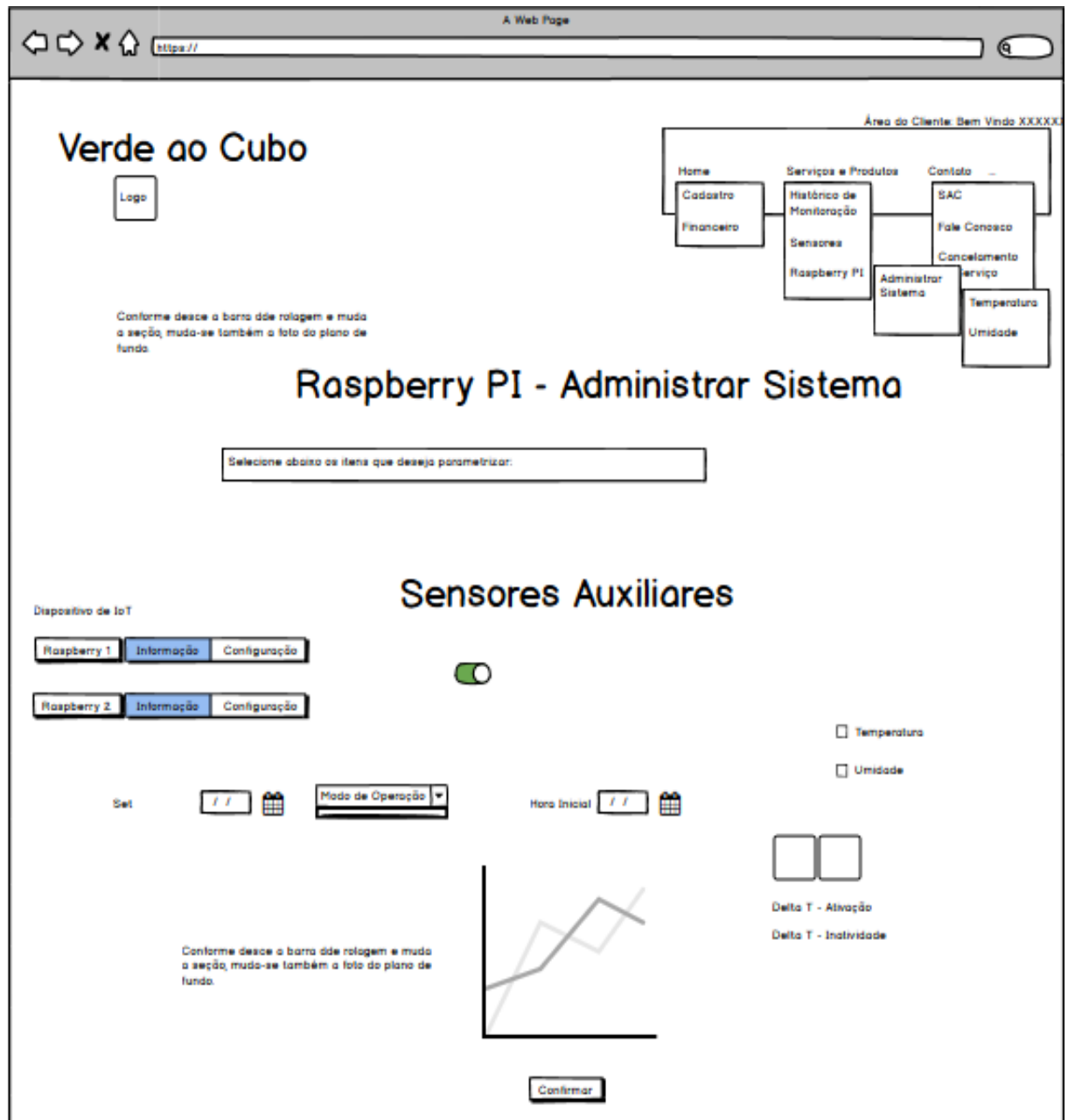
Botão Temperatura: botão de seleção do sensor de temperatura

Botão Umidade: botão de seleção do sensor de umidade

Botão Hora Inicial: digitar a hora inicial a ser contemplada

Botão Hora Final: digitar a hora final a ser contemplada

Figura 26 – Sensores Auxiliares.



Desenvolvido por Dennys Braghirolli 2020.

Raspberry 1 Configuração: botão para selecionar a guia configuração.

Raspberry 2 Informação: botão para selecionar a guia informação.

Raspberry 2 Configuração: botão para selecionar a guia configuração.

Botão Temperatura: botão de seleção do sensor de temperatura

Botão Umidade: botão de seleção do sensor de umidade

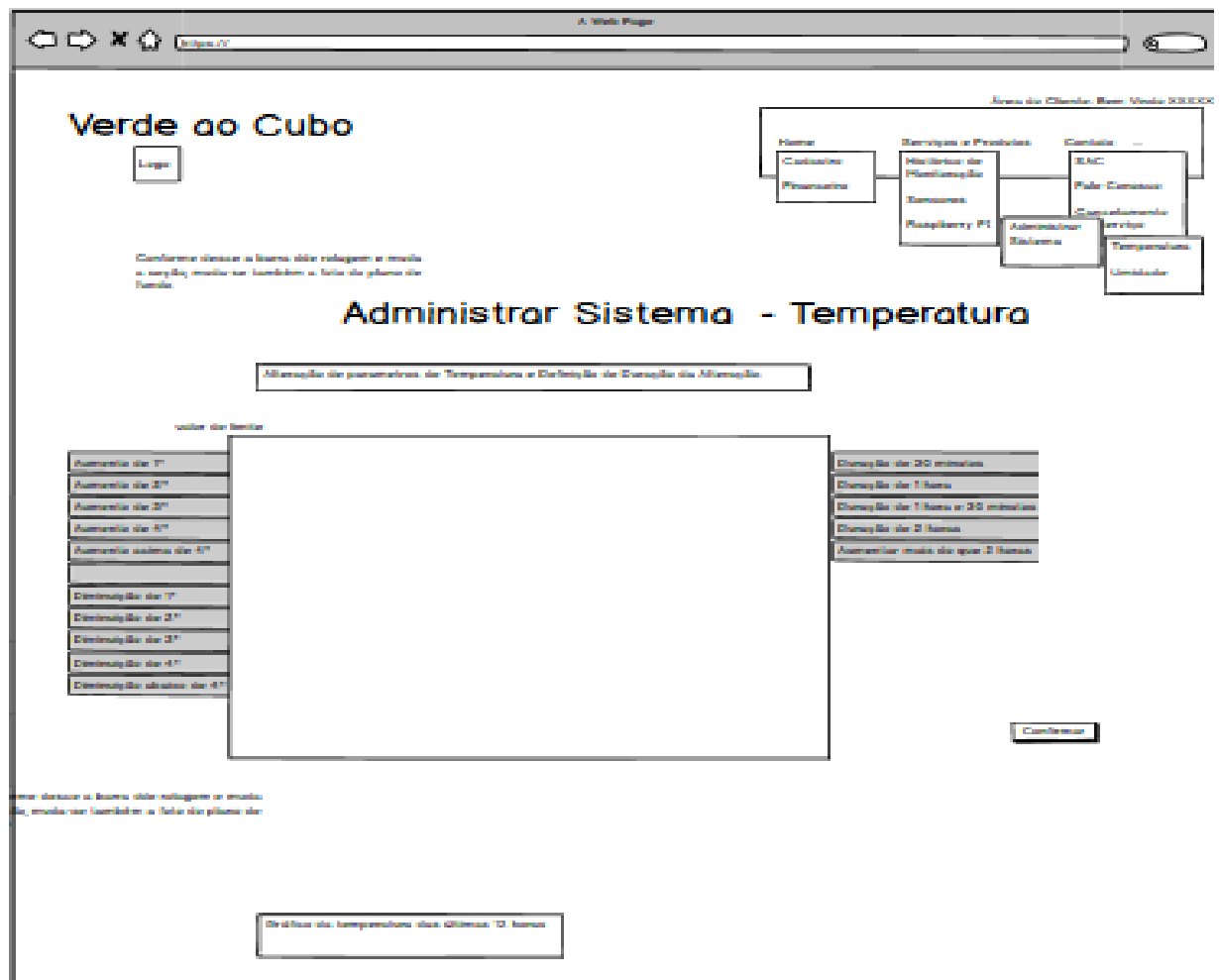
Botão Selecionar Dia: digitar o dia a ser pesquisado

Botão Hora Inicial: digitar a hora a ser pesquisada

Botão Modo de operação: escolher o tipo de operação.

Botão Confirmar: confirmar os dados solicitados.

Figura 27 – Sensor de Temperatura.



Desenvolvido por Dennys Braghirolli 2020.

Botão aumentar 1º: aumentar a temperatura em 1º.

Botão aumentar 2º: aumentar a temperatura em 2º.

Botão aumentar 3º: aumentar a temperatura em 3º.

Botão aumentar 4º: aumentar a temperatura em 4º.

Botão aumentar mais do que 4º: aumentar a temperatura em mais do que 4º.

Botão diminuir 1º: diminuir a temperatura em 1º.

Botão diminuir 2º: diminuir a temperatura em 2º.

Botão diminuir 3º: diminuir a temperatura em 3º.

Botão diminuir 4º: diminuir a temperatura em 4º.

Botão diminuir mais do que 4º: diminuir a temperatura mais do que 4º.

Botão duração de 30 minutos: duração de 30 minutos do aumento de temperatura.

Botão duração de 1 hora: duração de 1 hora do aumento de temperatura.

Botão duração de 1 hora e 30 minutos: duração de 1 hora e 30 minutos do aumento de temperatura.

Botão duração de 2 horas: duração de 2 horas do aumento de temperatura.

Botão de aumentar mais do que 2 horas: duração de mais de 2 horas do aumento de temperatura.

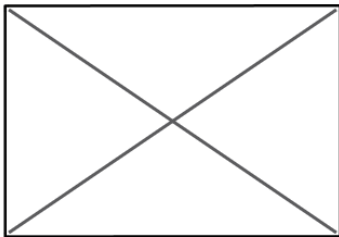
Botão Confirmar: confirmar a alteração da duração da temperatura.

3.2.1.2. 2º Protótipo

Figura 28 – Tela 1 Home do website.

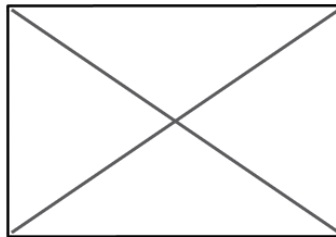


Microgreens



Descrição o que são Microgreens

Microgreens frutas



Descrição o que são Microgreens frutas

Informações adicionais

Para saber mais sobre os produtos e serviços da Verde ao Cubo, entre em contato através do nosso canal de atendimento.

Contato

Rua Bell Alliance 225, São Caetano do Sul, São Paulo - Brasil.
contato@verdeocubo.com
+ 11 XXXX-XXXX
+ 11 9XXXX-XXXX

© 2020 Copyright: verdeocubo.com

Desenvolvido por Dennys Braghirolli 2020

A tela inicial, a homepage, tem um banner giratório, e abaixo há um link para explicações de microgreens, ainda tem os canais de contato.

Figura 29 – Tela 2 Login.

Login

Usuário:

Senha:

Desenvolvido por Dennys Braghirolli 2020.

Figura 30 – Tela 3 Erro de login.

Confirmação de E-mail

E-mail confirmado com sucesso!

E-mail não encontrado, por favor tente de novo.

Usuário:

Senha:

Desenvolvido por Dennys Braghirolli 2020.

Caso o usuário coloque o e-mail ou senha errado, aparecerá uma mensagem de erro, assim como aparecerá a mensagem de confirmação caso o cliente coloque corretamente seu usuário.

Figura 31 - Tela 4 Cadastro.

Cadastro

NOME:

E-MAIL:

SENHA:

Gravar

Desenvolvido por Dennys Braghirolli 2020.

Para se cadastrar o cliente irá criar um usuário, colocar seu e-mail e criar uma senha.

Figura 32 - Tela 5 Gerenciamento de Conta.

Gerenciamento de Conta

E-mail:

Alterar E-mail:

Senha:

Nova Senha:

Repetir Senha:

Confirmar

Desenvolvido por Dennys Braghirolli 2020.

Nesta tela será possível alterar a senha do usuário.

3.2.1.3. Protótipos Finais

Após o desenvolvimento dos mockups e todo o processo de criação e adaptação, chegamos ao produto final dos aplicativos web e mobile.

A tela inicial do sistema web contempla todos os recursos os recursos no alto da tela, bastando clicar no item que se deseja ver as informações ou utilizar a barra de rolagem da tela.

Figura 33 - Home web final.



Desenvolvido por Lucas Monteiro 2020

No campo superior esquerdo fica a guia Serviço, com o descritivo do que a empresa realiza.

Figura 34 - Home web final área Serviço

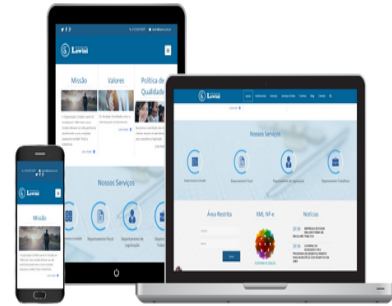
O Serviço

Cultivos hidropônicos necessitam de monitoração e manutenção constante. Desta forma, a equipe Verde³ desenvolveu a solução **Verde³ Watcher**, um sistema de controle e monitoração para cultivos hidropônicos.

A solução permite que um produtor defina os níveis necessários de controle e monitoração em seu cultivo hidropônico de forma remota. Estas funcionalidades são controladas via aplicativo mobile, proporcionando ao produtor a capacidade de controlar os horários de funcionamento de dispositivos, como bombas de água, e ainda receber informações atualizadas sobre os parâmetros ambientais monitorados no cultivos.

Dentre todas as características da solução, seu principal diferencial está no uso de plataformas de computação em nuvem, que proporcionam persistência dos dados de monitoração e o acesso entre diferentes dispositivos, desde que possuam acesso à internet.

A comunicação *peer-to-peer* é dispensada em favor de um serviço online que não exige maiores conhecimentos sobre configuração de rede ou de seus dispositivos inteligentes.



Desenvolvido por Lucas Monteiro 2020

Depois vem os tipos de plano a ser contratado, onde a quantidade de sonda vai definir o valor da mensalidade

Figura 35 - Home web final Planos.

Planos		
Starter	Plus	Pro
<ul style="list-style-type: none">✓ Acesso ao Verde³ Watcher (Apenas web)✓ 1x Sonda <i>Raspberry Pi</i>✓ 1x Sensor de temperatura✓ 1x Sensor de umidade✓ 1x Sensor de luminosidade	<ul style="list-style-type: none">✓ Acesso ao Verde³ Watcher (Web & Mobile)✓ 3x Sonda <i>Raspberry Pi</i>✓ 3x Sensor de temperatura✓ 3x Sensor de umidade✓ 3x Sensor de luminosidade	<ul style="list-style-type: none">✓ Acesso ao Verde³ Watcher (Web & Mobile)✓ Sondas <i>Raspberry Pi</i> ilimitadas (mediante aquisição)✓ Sensores ilimitados (mediante aquisição)✓ Ferramentas web para análise de dados
R\$ 9,99/mês	R\$ 12,99/mês	R\$ 19,99/mês
Adquirir	Adquirir	Adquirir

Desenvolvido por Lucas Monteiro 2020.

Quem somos, tem o descritivo dos membros da equipe de desenvolvimento e criação de toda a empresa.

Figura 36 - Home web final Quem somos

Quem somos

A Verde³ é uma AGROTECH, um tipo de startup voltada ao desenvolvimento de tecnologias inovadoras para a agricultura e para o agronegócio. Fundada em 2019 pelos alunos do Curso de Análise e Desenvolvimento de Sistemas como Trabalho de Conclusão de Curso pela Fatec São Caetano, na cidade São Caetano do Sul (SP), a Verde³ busca desenvolver sistemas inteligentes para o controle e o monitoramento de cultivos hidropônicos.

O uso de sistemas de cultivo hidropônico em centros urbanos representa uma forma inovadora de produção de alimentos e auxilia a redução do desperdício de água, o uso de embalagens, o uso de agrotóxicos e os custos de transporte na produção de alimentos. No entanto, os produtores precisam realizar diariamente um conjunto de tarefas e atividades para manter um ambiente de cultivo produtivo e consistente. A Verde³ busca desenvolver sistemas que reduzam a carga de atividades e tarefas de seus usuários, permitindo que um cultivo hidropônico opere de forma mais autônoma e que seus usuários possam focar nas demais tarefas e atividades necessárias para atingir a produtividade desejada.

Desenvolvedores:



Caio Aragon



Lucas Monteiro



Alecsander Torquato



Silas Moraes



Dennys Braghioroli

Professor Orientador:



Adilson Ferreira

Desenvolvido por Lucas Monteiro 2020.

A guia contato tem as informações sobre a empresa para o cliente os contatar de acordo com os canais site, e-mail e telefone.

Figura 37 - Home web final Contato

Nome E-mail

Mensagem

INFORMAÇÕES ADICIONAIS
Para mais informações, por favor entre em contato com nosso canal de atendimento

CONTATO
Rua Bell Alliance 225, São Caetano do Sul, São Paulo - Brasil.
projetoverdeacubo@gmail.com
+ 11 XXXX-XXXX
+ 11 9XXXX-XXXX

Copyright © Verde Ao Cubo 2020

Desenvolvido por Lucas Monteiro 2020.

Ir para o App vai levar o cliente para o aplicativo.

Figura 38 - Home web final Ir para o App.



Desenvolvido por Lucas Monteiro 2020.


Figura 39 - Home web final Ir para área de Login.



Desenvolvido por Lucas Monteiro 2020.

Entrar leva o usuário até a área de login.

Figura 40 - Home web final área de Login.



Entrar

Endereço de e-mail

Senha

Entrar

[Não possui conta? Clique aqui para se registrar!](#)

Desenvolvido por Lucas Monteiro 2020.

E também há a área Registrar-se, onde o cliente poderá criar a sua conta.

Figura 41 - Home web final Tela de Registro.



Criar conta

Nome completo

Endereço de e-mail

Senha

Confirmar Senha

Registrar-se

[Já possui conta? Clique aqui para fazer login!](#)

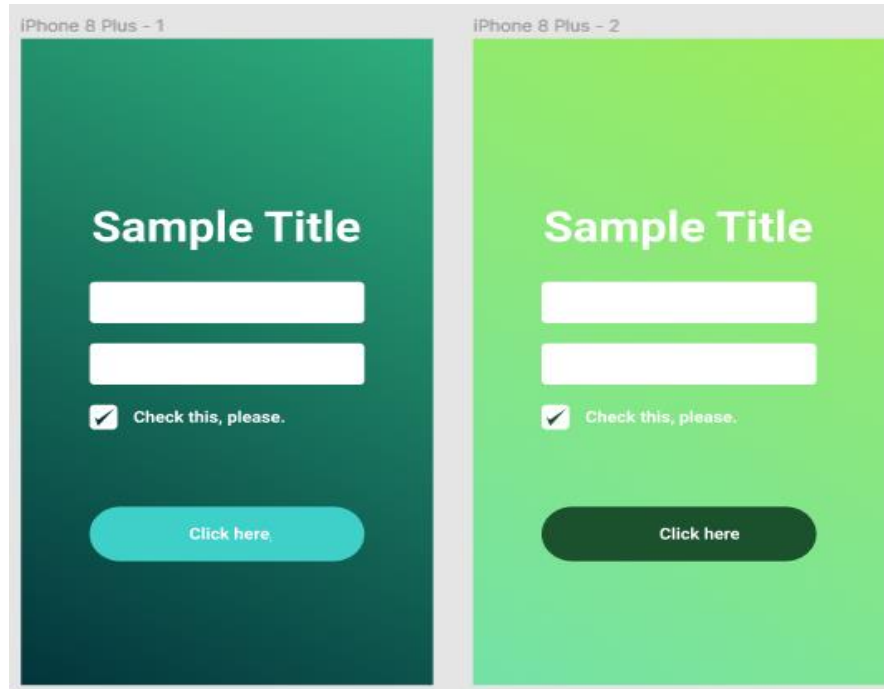
Desenvolvido por Lucas Monteiro 2020.

3.2.2. Protótipos do Aplicativo Verde³

3.2.2.1. Protótipos Iniciais

Os protótipos do aplicativo móvel são exibidos nas figuras abaixo. Na tela inicial o usuário coloca o seu login e a senha e marca o checkbox e depois clica no local indicado.

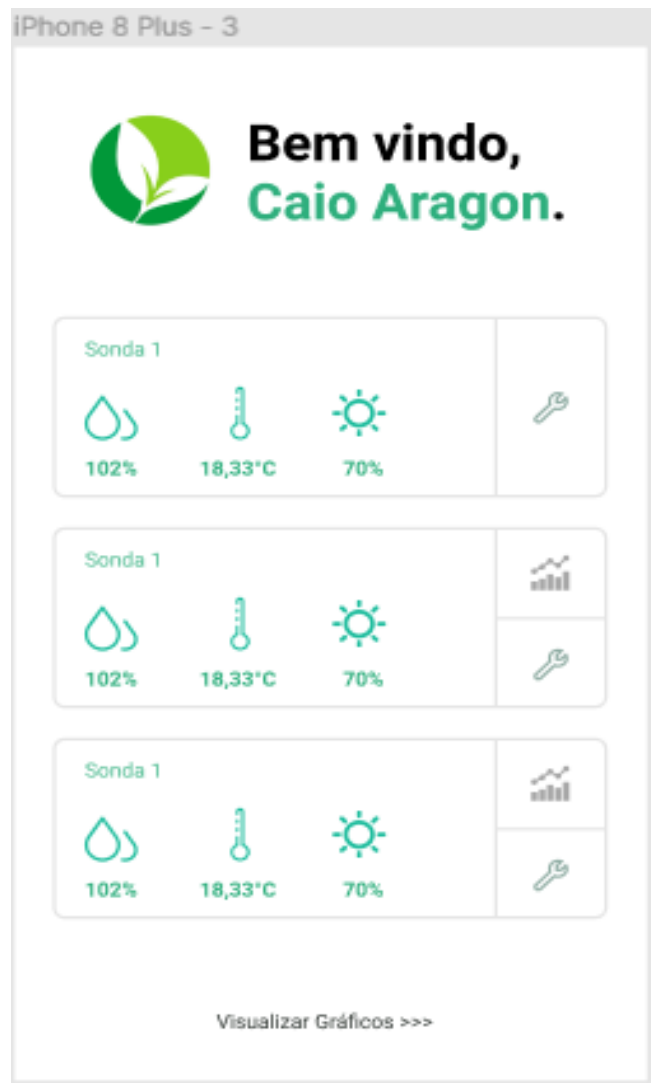
Figura 42 – Telas iniciais do aplicativo.



Desenvolvido por Lucas Monteiro 2020.

Depois vem a tela de abertura do aplicativo com o detalhe das monitorações das Sondas.

Figura 43 – Tela Geral Dispositivos.



Desenvolvido por Lucas Monteiro 2020.

Na tela seguinte há a configuração da tabela de dispositivos.

Figura 44 – Configuração de Sonda.



Desenvolvido por Lucas Monteiro 2020.

Figura 45 – Configuração de Sensores.



Desenvolvido por Lucas Monteiro 2020.

E por último, o controle de configuração da iluminação.

Figura 46 – Configuração de Sensor de iluminação.



Desenvolvido por Lucas Monteiro 2020.

3.2.2.2. Protótipos Finais

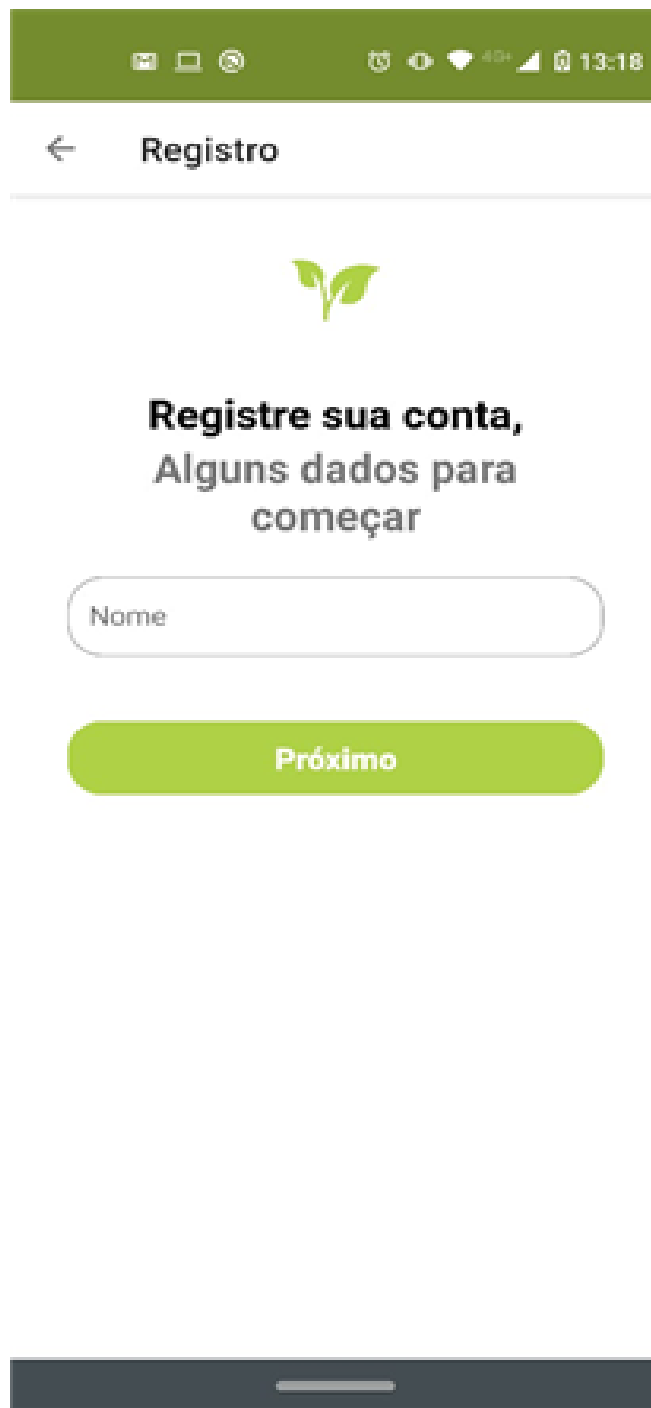
Figura 47 - Home App final Login.



Desenvolvido por Alecsander Torquato 2020

A tela inicial possui o Botão “Entrar” que direciona o usuário a tela de login do aplicativo, que segue a jornada de login. O Botão “Registrar” que direciona o usuário a primeira tela de Registro no sistema, que segue a jornada de registro.

Figura 48 – Home App final Registro.



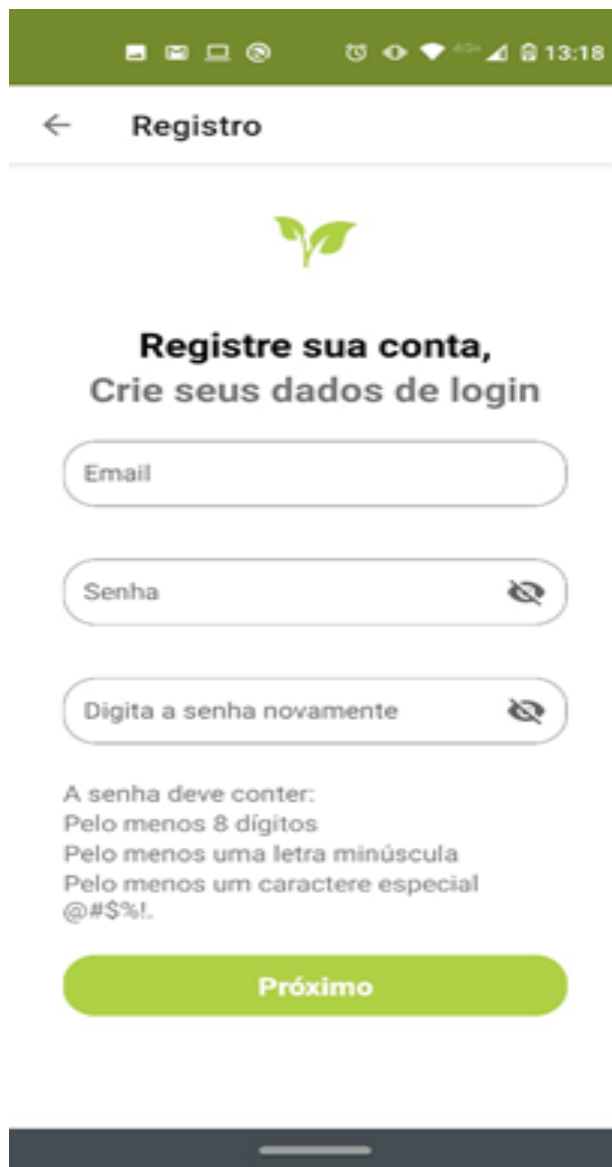
Desenvolvido por Alecsander Torquato 2020

A primeira tela de registro possui os seguintes componentes:

Campo “Nome” onde o usuário coloca o seu nome para registro no sistema.

Botão “Próximo” que valida o nome digitado e direciona o usuário à segunda tela de registro.

Figura 49 – Home App final Registrar a Conta.



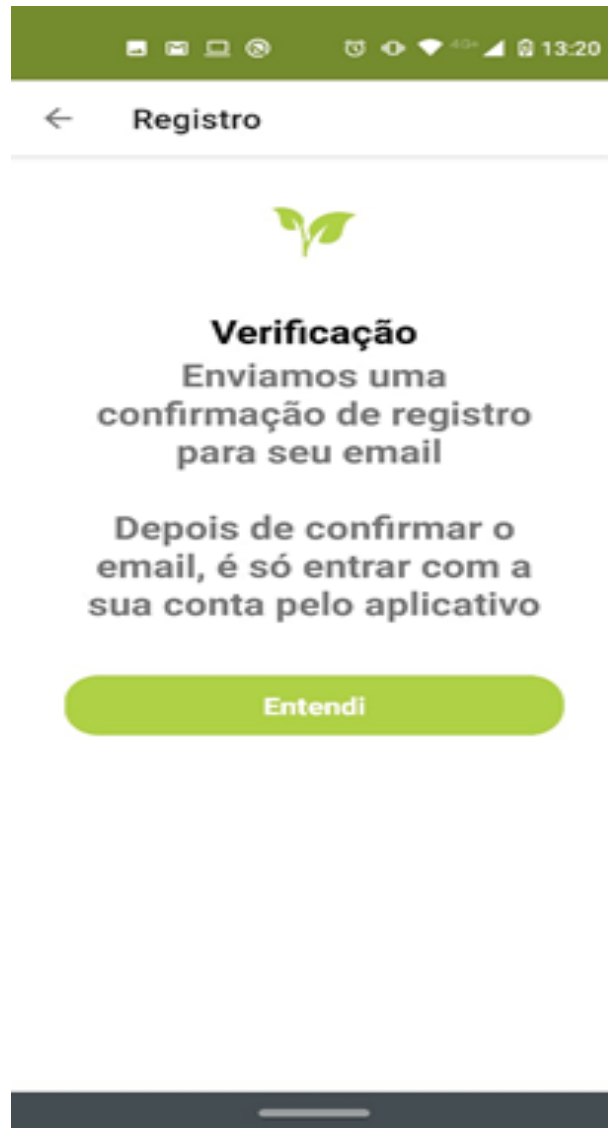
Desenvolvido por Alecsander Torquato 2020.

A segunda tela de registro possui os seguintes componentes:

Campo "Email" onde o usuário coloca o seu e-mail para registro. Campo "Senha" onde o usuário coloca a senha do seu novo registro. Campo "Digite a senha novamente" onde o usuário coloca a mesma senha para validação.

Botão "Próximo" que valida o e-mail e senha digitados, e direciona o usuário à terceira tela de registro.

Figura 50 – Home App final Confirmação Cadastro.

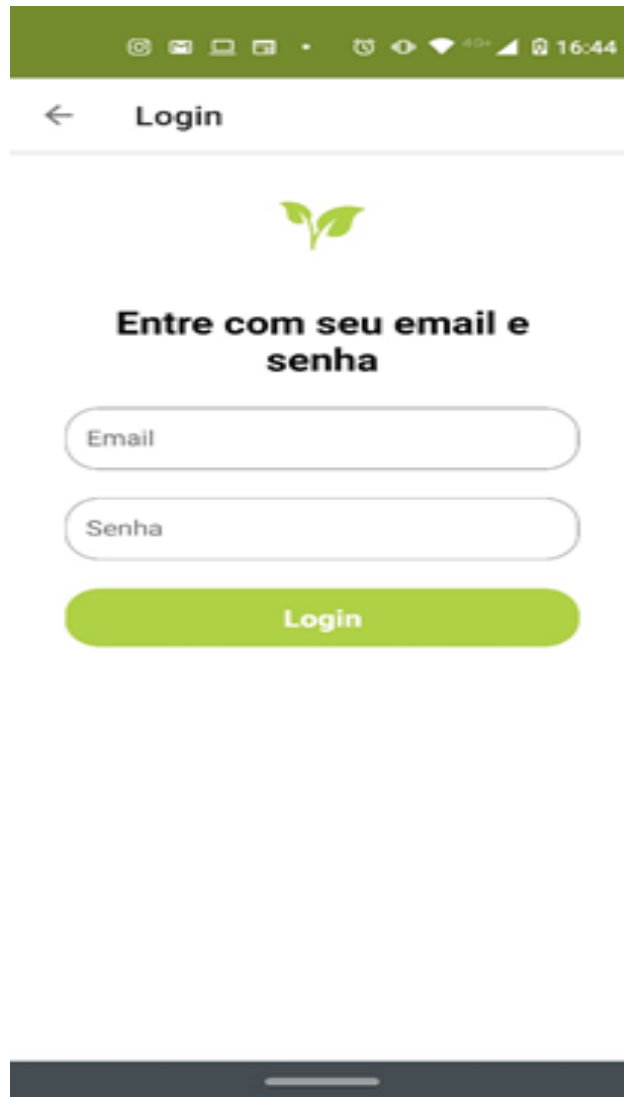


Desenvolvido por Alecsander Torquato 2020.

A terceira tela de registro possui os seguintes componentes:

Botão “Entendi” que direciona o usuário à tela inicial do aplicativo.

Figura 51 – Home App final Área para se logar.



Desenvolvido por Alecsander Torquato 2020.

A tela de login possui os seguintes componentes:

Campos “Email” onde o usuário coloca o seu e-mail. Campo “Senha” onde o usuário coloca sua senha.

Botão “Login” que valida o login e em caso de sucesso na validação, direciona o usuário ao menu principal do aplicativo.

Figura 52 – Home App final Home usuário logado.



Desenvolvido por Alecsander Torquato 2020.

A tela do menu principal carrega informações relevantes ao usuário como o nome que ele inseriu ao fazer o registro no sistema e uma lista contendo CARDS que representam as suas sondas.

A tela do menu principal possui os seguintes componentes:

Botão “Menu curto” representado pelo ícone de três pontos no topo direito da tela, abre um novo componente na tela e direciona o usuário a jornada de logout do aplicativo. Botão “Séries temporais” dentro de cada CARD na lista de sondas, representado pelo ícone “Histograma com curva”, que abre uma nova tela e direciona o usuário a jornada de visualização de séries temporais. Botão “Configuração”, dentro de cada CARD na lista de sondas, representado pelo ícone “Chave de boca”, que abre uma nova tela e direciona o usuário a jornada de configuração de sonda.

Figura 53 – Home App final Probe.



Desenvolvido por Alecsander Torquato 2020.

O ícone três pontos quando selecionado, apresenta o seguinte menu. Opção “Sair” que ao ser selecionado, apresenta na tela um novo componente esperando a confirmação de logout do usuário.

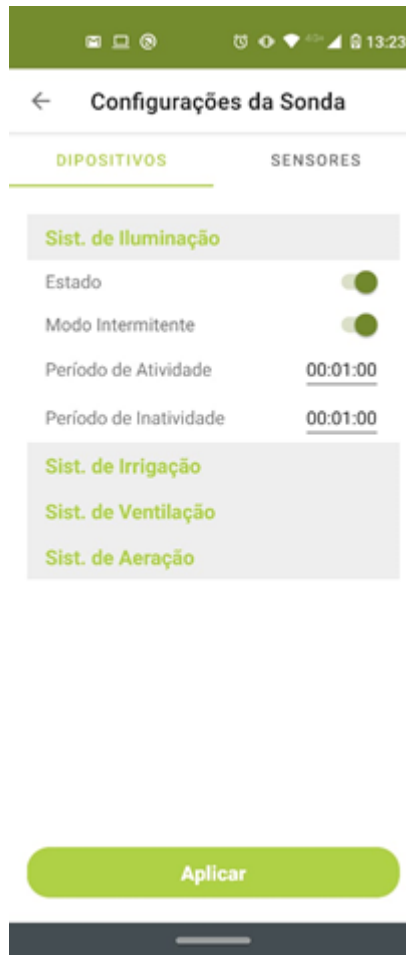
Figura 54 – Home App final Logout.



Desenvolvido por Alecsander Torquato 2020.

A confirmação de logout exibe um componente na tela que aguarda a confirmação do usuário. O botão “Sim” realiza o logout do usuário do aplicativo, abrindo a tela inicial. O botão “Não” fecha o novo componente exibido, permanecendo na tela de menu principal.

Figura 55 – Home App final Configurar Sonda.



Desenvolvido por Alecsander Torquato 2020.

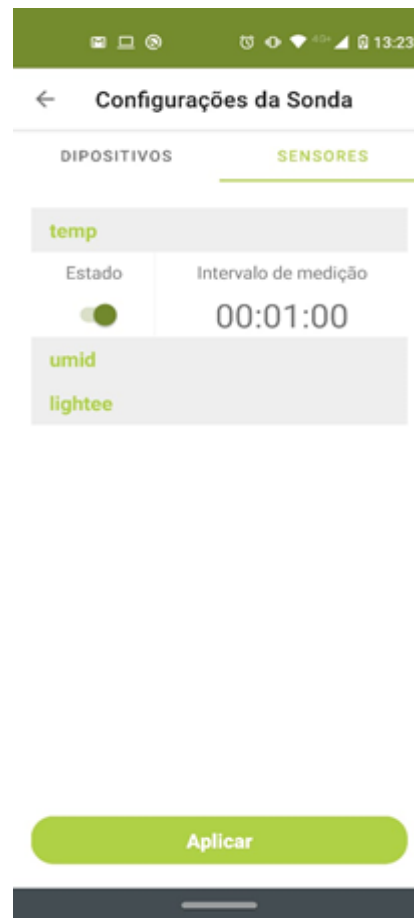
A tela de configuração de sonda, na aba dispositivos, apresenta uma lista contendo todos os dispositivos associados a sonda selecionada com os seguintes componentes:

Lista de dispositivos. Cada dispositivo é apresentado inicialmente somente com o seu nome, quando o usuário seleciona um dispositivo da lista, o componente expande. Dentro desse componente expandido são apresentados novos componentes. O Switch “Estado” que define se aquele dispositivo está ligado. O Switch “Modo intermitente” que define se o modo de ativação do dispositivo será periódico,

Caso o “Modo intermitente” esteja ativado o campo “Período de Atividade” que ao ser selecionado apresenta um componente para edição do tempo de atividade em horas, minutos e segundos do dispositivo e o campo “Período de Inatividade” que ao ser selecionado apresenta um componente para edição do tempo de atividade em horas, minutos e segundos do dispositivo, que ficam ativos para a seleção.

Botão “Aplicar” que verifica a lista de dispositivos e aplica as mudanças feitas pelo usuário.

Figura 56 – Home App final Aplicar alteração.



Desenvolvido por Alecsander Torquato 2020.

A tela de configuração de sonda, aba sensores, apresenta uma lista contendo todos os sensores associados a sonda selecionada. A aba sensores possui os seguintes componentes:

Lista de sensores. Cada sensor é apresentado inicialmente somente com o seu nome, quando o usuário seleciona um sensor da lista, o componente expande. Dentro desse componente expandido são apresentados novos componentes. O Switch “Estado” que define se aquele sensor está ligado. O componente “Intervalo de medição” que ao ser selecionado apresenta um componente para edição do tempo de atividade em horas, minutos e segundos do sensor.

Botão “Aplicar” que verifica a lista de sensores e aplica as mudanças feitas pelo usuário.

3.2.3. Protótipos da Sonda.

Com relação aos componentes de hardware da sonda, seu protótipo foi modelado tendo como referências os guias “*Photo-resistor light sensor on Raspberry Pi*” (FREVA, 2020), “*How To Set Up The Dht11 Humidity Sensor On The Raspberry Pi*” (CIRCUIT BASICS, 2020) e o vídeo tutorial “*RASPBERRY PI – All About controlling Relay Boards for Home Automation*” (SARAVANAN AL, 2020). Os resultados deste processo estão apresentados na Figura 57.

Figura 57 – Protótipo final da sonda

3.3. Tecnologias Utilizadas

3.3.1. Tecnologias da Página Web

Linguagens de programação utilizadas para desenvolvimento da página web:

Linguagem de Marcação de Hipertexto - HTML (HyperText Markup Language), ele é utilizado para marcação nas construções web, das páginas. É através dela que o navegador interpreta o algoritmo para gerar a página com seus recursos. Ele possui interligação de imagens, textos, vídeos e links para responder aos comandos do usuário. O HTML possui as chamadas "Tags" que são marcadores, palavras entre parênteses que são os comandos de formatação da linguagem. Elemento é composto por uma tag, atributos, valores e por outros elementos ou texto que são chamados filhos.

Cascading Style Sheets (CSS) é a folha de estilos, ela possui camadas que são responsáveis pela aparência da página web. É através dela que se define a exibição dos elementos dos códigos onde se separa o formato do conteúdo da página da internet.

Javascript é uma linguagem de programação criada no fim do século passado com o intuito de ajudar os desenvolvedores a dinamizar as formas de construção da criação das páginas seja para mobile ou para computadores. Hoje o desenvolvimento é feito a partir do Javascript casado com outras estruturas.

Sassy Cascading Style Sheets significa Folhas de Estilo em Cascata (SCSS), ele vai funcionar como um pré-compilador com estilos de folhas e com uma sintaxe avançada, que são processadas pelo programa para depois transformar essas folhas no estilo CSS normal.

Spectre é uma estrutura do CSS ligado ao SCSS, sendo responsivo, leve e moderno para um desenvolvimento mais rápido.

NodeJS é uma plataforma de execução construída sobre o JavaScript executando como server-side, onde é possível desenvolver aplicações em larga escala, sem depender de um navegador para a execução.

Express é um framework para aplicativos do Node.js, oferecendo diversos recursos tanto para aplicativo móvel quanto web. Ele dispõe de várias soluções de configuração, integração de view entre outros recursos.

Express Generator é uma ferramenta geradora de aplicativos, pois através de uma base inicial ele monta em poucos segundos um projeto. É um recurso muito prático que traz grande auxílio no desenvolvimento.

Nodemon é um módulo ligado ao Express Generator em que roda um script que tem como função monitorar uma aplicação que teve alterações em arquivos, podendo até se necessário, reiniciar o servidor.

Embedded JavaScript (EJS) é o motor, é uma linguagem de visualização, usando códigos em javascript para páginas web via marcação em HTML, levando dados do back para o front-end.

3.3.2. Tecnologias do Aplicativo

Segundo Mark Zuckerberg (2012) Quando sou introspectivo sobre os últimos anos, acho que o maior erro que cometemos, como empresa, foi apostar demais no HTML5 em oposição ao nativo.

Para o desenvolvimento do aplicativo, foi escolhida uma solução nativa que é uma entre diversas categorias existentes para o desenvolvimento de aplicativos.

Aplicações nativas podem usufruir diretamente do Sistema Operacional onde ele está instalado, podendo ter acesso às diferentes tecnologias implementadas em seu SO como GPS, leitor biométrico, acelerômetro, e a demais funcionalidades pré-existentes, de uma forma mais rápida e compatível.

Soluções de desenvolvimento nativas são mais rápidas que as outras soluções devido a facilidade que ela tem em utilizar os recursos específicos providos de seu sistema operacional.

Ao desenvolver qualquer solução de software, uma das maiores preocupações está relacionada a interface do usuário e como será sua experiência de navegação. A forma utilizada para realizar o desenho da interface é se aproveitar das pré-experiências de navegação de outros aplicativos.

Visuais de aplicativos tendem a estarem padronizados como por exemplo, possuírem um menu na lateral esquerda da tela representado por um ícone de três barras ou possuírem um ícone de seta indicando ao usuário que ele pode voltar para a tela anterior. Esses padrões estão relacionados justamente as tecnologias implementadas no sistema operacional, existe um padrão e ele tende a ser seguido pois o usuário já está acostumado com aquilo.

O desenvolvimento hoje sugere a implementação de diferentes bibliotecas geradas por terceiros, porém, essas bibliotecas surgem de uma hora para outra podendo causar problemas de segurança devido à falta de garantia de segurança que aquela biblioteca será continuada e atualizada. Isso não acontece em aplicações nativas já que o seu desenvolvimento não depende de nenhuma biblioteca de terceiros.

A plataforma Android oferece duas linguagens para seu desenvolvimento, são elas Java e Kotlin. Para efeito de programação, foi utilizado a linguagem Java pois ela foi a linguagem mais abordada no decorrer do curso, gerando assim uma maior afinidade ao desenvolvedor.

Durante o desenvolvimento do aplicativo, foram utilizadas diversas ferramentas que auxiliaram a edificação do projeto mobile. Para o desenvolvimento de mockups de interface, foi utilizado a ferramenta de edição de interfaces Figma que permite uma construção intuitiva, interações e é otimizada para interfaces mobile. Ele oferece um conteúdo muito amplo, mesmo utilizando o seu plano gratuito, possibilitando a edição e visualização do mesmo projeto por mais de uma pessoa.

De acordo com Kezz Bracey (2018). O Figma fornece todas as ferramentas necessárias para a fase de design do projeto, incluindo ferramentas vetoriais capazes de ilustração completa, além de recursos de prototipagem e geração de código para entrega.

Para efeito de conexão com API e chamadas de requisições WEB, foi utilizado o Retrofit que possibilita transformar uma API HTTP em uma interface facilmente implementada pela linguagem Java. No Retrofit, existem anotações que podem gerar requisições com corpos em multipartes, conversão de corpos de requisições em objetos Java e adição de parâmetros na URL ou juntamente ao corpo da requisição.

Segundo Lars Vogel, Simon Scholz, David Weiser (2018), o Retrofit é um cliente REST para Java e Android. Isso torna relativamente fácil recuperar e fazer upload de JSON (ou outros dados estruturados) por meio de um serviço da Web baseado em REST. No Retrofit, você configura qual conversor é usado para a serialização de dados. O retrofit usa a biblioteca OkHttp para solicitações HTTP.

Para estruturação do código da aplicação mobile, foi utilizado uma arquitetura que trata de projetos mobile, a arquitetura model-view-presenter ou MVP.

O MVP é uma variação do popular modelo arquitetural model-view-controller (MVC). Quando se usa MVP, toda a responsabilidade lógica do código é feita na camada presenter, assim separando toda a persistência lógica fora da classe de interface como Activity e Fragment.

No projeto, o MVP foi utilizado por conseguir separar bem as camadas de apresentação e de lógica em relação ao projeto. Essa escolha de modelo de arquitetura permite o baixo custo na manutenção, caso haja necessidade, mudanças serão fáceis de serem implementadas ou removidas na parte do código.

3.3.3. Tecnologias da Sonda

Esta seção apresenta todas os recursos de hardware e software utilizados no desenvolvimento da sonda da solução Verde³ Watcher. Inicialmente serão apresentados os recursos de hardware utilizados nos módulos de processamento, controle e monitoração e em seguida, os recursos de software utilizados na parte desenvolver a parte lógica do módulo de processamento.

3.3.3.1. Módulo de Processamento.

O módulo de processamento exibido, contém todos os recursos necessários para condicionar o funcionamento de sensores no módulo de monitoração e dos circuitos de acionamento no módulo de controle, além dos recursos necessários para a transmissão de dados entre a API e a sonda. Foram usados nestes componente da sonda um Raspberry Pi 3 Model B e uma fonte 5V.

O Raspberry Pi 3 Model B é um computador de placa única (Figura 58) e foi escolhido para exercer a função de módulo de processamento da sonda devido ao seu tamanho reduzido, recursos computacionais e custo.

Figura 58 - Computador de placa única Raspberry Pi 3 Model B.



Fonte: RASPBERRY PI FOUNDATION (2020).

Desenvolvido pela Fundação Raspberry Pi, o Raspberry Pi 3 Model B contém um processador Quad Core 1.2GHz Broadcom BCM2837 64bit, 1GB de RAM, Bluetooth 4.1, Wireless LAN, uma porta Ethernet, 40 pinos estendidos GPIO, quatro portas USB 2.0, um conector de interface HDMI, uma entrada para micro SD, uma porta Micro USB para alimentação de até 2.5A e uma saída de vídeo composto e áudio com conector P4 (RASPBERRY PI FOUNDATION, 2020). Apesar de ser classificado como computador de placa única, o Raspberry Pi tem a função de microcontrolador.

3.3.3.2. Módulo de Controle

O módulo de controle contém um circuito de acionamento com relés e conectores. Este componente da sonda é responsável por controlar o acionamento de dispositivos externos ao sistema e presentes em muitos sistemas de cultivo hidropônico, como bombas de água ou um conjunto de lâmpadas, que formam um sistema de iluminação artificial. Foram utilizados neste componente uma placa de interface de relés 5V com 4 canais (Figura 59), quatro tomadas padrão 2P+T 10A/250V, um conector múltiplo com 2.5mm de diâmetro, cabos jumper M/F e fios condutores de 2.5mm.

Figura 59 - Placa de interface de relés 5V com 4 canais.



Fonte: SUNFOUNDER (2020).

A placa de interface possui quatro relés de 5V para correntes 10A - 250V AC ou 10A 30V DC, quatro canais que operam sob 15-20mA, uma interface padrão que permite seu controle por um microcontrolador e um optoacoplador, que inibe interferências elétricas nos relés e no microcontrolador, otimizando as conexões entre a placa e o microcontrolador. Os relés funcionam como interruptores inteligentes, controlando a passagem de energia no circuito através de sinais da placa controladora, como o Raspberry Pi 3 Model B (SUNFOUNDER, 2020).

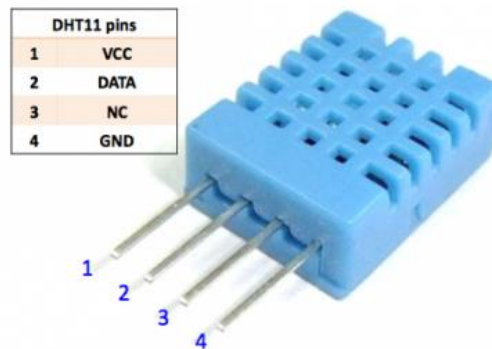
3.3.3.3. Módulo de Monitoração

O módulo de monitoração da sonda é responsável pela coleta de dados ambientais e contém sensores de temperatura, umidade e luminosidade. Foram utilizados dois sensores DHT11, um fotoresistor LDR, uma placa ensaio eletrônica de 400 furos, um capacitor de 1 μ F, dois resistores de 10kOhm, um resistor de 2kOhm e cabos jumper M/F.

O DHT11 (Figura 60) é um sensor temperatura e umidade de baixo custo, mas com estabilidade e com saída calibrada para sinais digitais, que assegura a confiabilidade dos dados coletados e auxilia sua comunicação com microcontroladores. (MOUSER ELETRONICS, 2020). O sensor inclui um tipo de componente resistivo para medições de umidade, um componente NTC para medições de temperatura e pode ser conectado com microcontroladores de 8-bit de alta performance. O sensor pode medir valores de temperatura entre 0°C e 50°C e valores de umidade entre 20% e 90% com acurácia de $\pm 2^\circ\text{C}$ e $\pm 5\%$ respectivamente. O sensor possui quatro pinos,

mas é necessário que se conecte apenas três. O pino 1 corresponde a entrada de alimentação do sensor (3.3 – 5.0V DC), o pino 2 a comunicação de dados e o pino 4 ao condutor neutro ou terra (MOUSER ELETRONICS, 2020).

Figura 60 - Sensor de temperatura e umidade DHT11 utilizado no módulo de monitoração da sonda.



Fonte: VIDA DE SILÍCIO (2020).

O LDR (Light Dependant Resistor) ou resistor dependente de luz é um componente eletrônico passivo, do tipo resistor variável (Figura 61), mais especificamente, um resistor cuja resistividade varia conforme a intensidade de luz que incide sobre ele. Quanto maior a intensidade de luz, menor a resistência do LDR (ELECTRONIC NOTES, 2020). Na ausência de luz a resistência de um LDR atinge valores da ordem de 10^6 Ohms. Já na presença de luz intensa, encontra-se valores da ordem de 10^2 Ohms. Esta diferença de resistividade indica quão sensíveis são os resistores LDR (ELECTRONIC NOTES, 2020).

Figura 61 - Resistor LDR utilizado no módulo de monitoração da sonda.



Fonte: ELETROPEÇAS (2020).

3.3.3.4. Recursos de Software Utilizados

Foi utilizado uma distribuição 64bit do Raspbian, o sistema operacional utilizado em Raspberry Pi, agora chamado de Raspberry Pi OS. O software do módulo de processamento foi desenvolvido em Python, que contém recursos importantes para a interação do software com os pinos programáveis de entrada e saída do Raspberry Pi, os pinos GPIO (*General Purpose Input Output*). Estes recursos são encontrados na biblioteca RPi.GPIO. A biblioteca Adafruit_DHT também foi utilizada, pois é específica ao sensor DHT11.

3.4. Implementação

3.4.1. Implementação em Servidores

A implementação dos componentes de aplicação (*Website*, *Probe* e Aplicativo) formam somente três das unidades do ecossistema desenvolvido para o projeto. A solução consiste, também, em duas aplicações para servidores, acessadas remotamente pelas partes supracitadas, na finalidade de garantir o acesso autenticado aos recursos da aplicação, e informações para a implementação destes últimos.

Foram desenvolvidas duas aplicações, servindo ao propósito de API (Application Programming Interface, ou Interface de Programação de Aplicações, do Inglês), fornecendo dados para a operação dos softwares clientes (*Website* e Aplicativo).

3.4.1.1. REST

Acrônimo para Representational State Transfer (Transferência Representacional de Estado, do inglês), é um estilo de arquitetura de software, introduzido por Roy Thomas Fielding, em sua dissertação para doutorado em filosofia, no ano 2000. Este estilo de arquitetura tem sido uma das bases sólidas para a construção de APIs modernas e aplicações web escaláveis.

Field (2000, p.86, tradução nossa) descreve:

O estilo Transferência Representacional de Estado (REST) é uma abstração dos elementos de arquitetura contidos em um sistema distribuído de hipermídia. O REST ignora os detalhes de implementação de componentes e sintaxe de protocolo a fim de focar nos papéis dos componentes, as restrições sobre sua interação com outrem, e sua interpretação de elementos de dados

significativos. Ele engloba as restrições fundamentais sobre componentes, conectores, e dados que definem a base da arquitetura Web, e portanto a essência de seu comportamento como uma aplicação baseada em rede.

3.4.1.1.1. Recursos

Assim são nomeadas os conjuntos de informações que podem ser devolvidos pela aplicação. Recursos devem ser organizados por semântica, já que um recurso pode ter seu valor, isto é, o que é representado como entidade, identificado pelo seu nome.

Segundo Fielding (2010, p. 88, tradução nossa), “qualquer conceito que talvez possa ser alvo da referência em hipertexto do autor, deve se enquadrar dentro da definição de recurso”.

3.4.1.1.2. Segurança e Escalabilidade

Uma das principais características do estilo de arquitetura REST, é a definição “stateless” (sem estado, do inglês). Isto é, nenhuma informação sobre seu estado de sessão é conhecido pela API. O acesso é controlado por artefatos de segurança (um *token*, por exemplo), que não armazenam dados sobre requisições anteriores. Deste modo, cada requisição à API é feita de forma independente. Segundo Wysocki (2010, p. 78, tradução nossa):

[...] cada requisição feita do cliente para o servidor deve conter toda a informação necessária para entender a requisição, e não deve tirar vantagem de nada armazenado no contexto no servidor. O estado da sessão é portanto mantida inteiramente no cliente.

Ainda segundo o Wyocki, tal característica age em benefício de visibilidade, confiabilidade e escalabilidade. Dentre destas, sem dúvidas a característica mais importante para o desenvolvimento do projeto é a escalabilidade, visto que o software é pretendido para lidar com uma quantidade enorme de informação, à medida em que é expandido para se tornar uma solução comercial cada vez mais viável e completa.

Portanto, o estilo REST combina perfeitamente com as intenções de se desenvolver um protótipo majoritariamente aproveitável para o desenvolvimento de uma futura solução comercial completa.

3.4.1.2. JSON

Acrônimo e referência popular para *JavaScript Object Notation* (Notação de Objetos JavaScript), é um formato representação de dados de uma entidade em um conjunto de parâmetros separados por caracteres especiais. Os documentos JSON são nativamente interpretados pela linguagem de programação JavaScript (adotada no desenvolvimento). Se assemelha ao XML (Extensible Markup Language, ou Linguagem de Marcação Extensível, do inglês), também utilizado amplamente na confecção de aplicações web em padrão REST para a representação de recursos e transferência de dados, mas em comparação à esta última, o JSON obtém notável vantagem em velocidade de transferência e aproveitamento de volume de dados, uma vez que pode representar uma mesma entidade utilizando muito menos *bytes* que uma representação XML.

Figura – Representação JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<person>
  <name>Thomas John</name>
  <age>27</age>
  <email>thomas.john@address.domain.com</email>
</person>
```

Fonte: Lucas Monteiro (2020)

Figura B.2 – Representação JSON.

```
{
  "name": "Thomas John",
  "age": 27,
  "email": "thomas.john@address.domain.com"
}
```

Fonte: Lucas Monteiro (2020)

3.4.1.3. API de domínio

Trata-se do repositório central de informações do sistema. Sendo acessível através de qualquer um dos clientes por uma conexão à internet, esta API segue o estilo REST (ver seção a seguir). Através dos recursos expostos em rotas, é possível

se obter dados das entidades cadastradas. Isto é, usuários, *probes* (sondas) além de dados de monitoração e controle.

O acesso à estes recursos é restrito, permitido somente a entidades previamente cadastradas no sistema, mediante autenticação utilizando um *token* de acesso, fornecido pela API de autenticação (tratada em seção posterior). Todos os recursos são lidos de, e retornados ao cliente em formato JSON (ver seção anterior).

3.4.1.4. API de Autenticação

De forma a se conformar com os padrões “stateless” impostos pela aplicação do padrão REST na API de domínio, foi escolhida uma tecnologia de autenticação baseada na entrega de *tokens* JWT (acrônimo para JSON Web Token). Essa entrega de *tokens* toma como parâmetro de entrada, um documento JSON embutido na requisição HTTP feita ao servidor, contendo as credenciais de acesso do cliente.

A entrega de *tokens* que autenticam as requisições feitas à API de domínio é dividida em três tipos, aos quais os direitos de acesso são reservados à entidade entendida para o *token*. São explicados na lista a seguir:

1. Cliente: Este tipo de *token* é entendido para usuários comuns, clientes do negócio. Pode ser requisitado e obtido por administradores, para operações de usuário comum.

2. Administrador: *Token* que permite o acesso à funções exclusivas aos administradores (usuários desenvolvedores do projeto e mantenedores de conteúdo/sondas), como cadastro de *probes*, deleção de usuários, edição de dados de negócio da página inicial do *website*.

3. *Probe*: Este último tipo de *token* é estritamente reservado ao uso por parte do software de *Sondas*, para obtenção e envio dos dados de configuração e monitoração, respectivamente.

3.4.1.5. Implementação das APIs

Por utilizar a linguagem de programação JavaScript para o desenvolvimento de ambas as APIs, aliada ao framework Express, o projeto necessitou de definição de arquitetura flexível e que evitasse a repetição de código, obedecendo aos padrões

multiparadigma da linguagem. Isto é, o fato da linguagem JavaScript permitir a mistura entre paradigma de Programação Orientação a Objetos e Programação Funcional, clamou por uma estrutura que aproveitasse ambas.

A arquitetura planejada para as APIs foi baseada no MVC (Model-View-Controller, ou Modelo-Visão-Controllador, do Inglês), porém com a abolição de Views, sendo estas substituídas por *Routers* (Roteadores). As arquiteturas planejadas para as APIs são elucidadas nos diagramas a seguir:

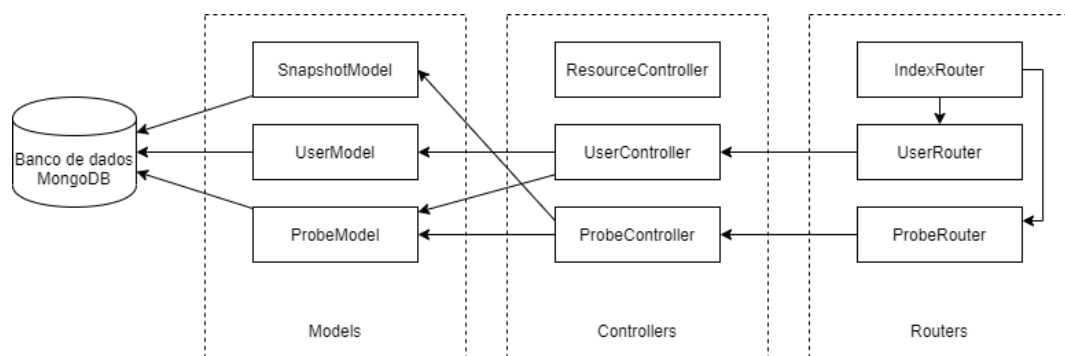


Figura C.2 – Diagrama de Contexto da API de domínio.

3.4.1.5.1. Fluxo Operacional das APIs

Sendo ambas construídas sobre o protocolo HTTP, o funcionamento de ambas as APIs é baseada na resposta às requisições, baseada nos códigos HTTP. O correto fluxograma de operação das APIs é elucidada pelo fluxograma à seguir:

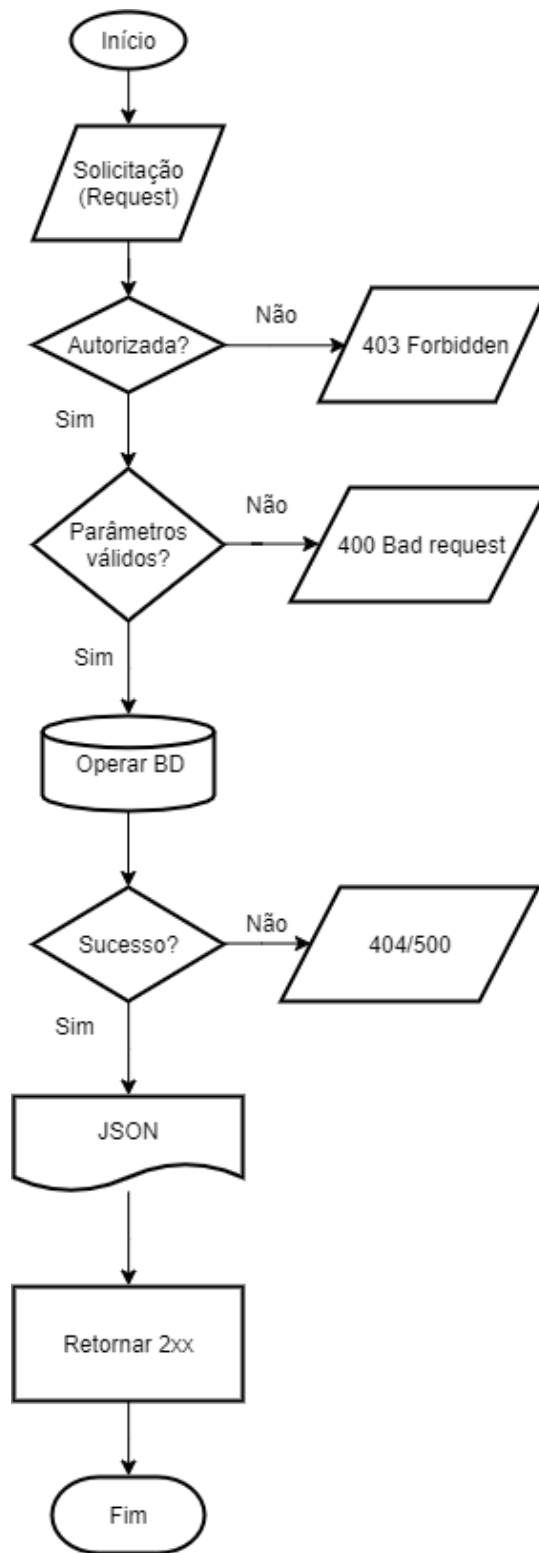


Figura C.1 – Flograma de funcionamento das APIs

C.1.1 Rotas

A relação entre rotas (recursos) e tipo de *token* necessário para o seu acesso é detalhada na tabela a seguir:

Tabela B.1 – Rotas da API de domínio

Método HTTP	Rota / Recurso	Token de acesso
GET	/users/	Administrador
GET	/users/me	Usuário
PATCH	/users/me	Usuário
DELETE	/users/me	Usuário
GET	/users/me/probes	Usuário
GET	/probes/	Administrador
PST	/probes/	Administrador
GET	/probes/<id>	Usuário
PATCH	/probes/<id>	Usuário
DELETE	/probes/<id>	Administrador
GET	/probes/<id>/snapshots	Usuário
DELETE	/probes/<id>/snapshots	Usuário
GET	/probes/<id>/snapshots/latest	Usuário

Fonte: Lucas Monteiro (2020)

3.4.2. Implementação da Página Web

Foi desenvolvido o website com o intuito de apresentar a empresa aos clientes, lá é possível conhecer mais sobre o assunto, há telas de contato, é possível se cadastrar para posterior acesso ao aplicativo onde se administra a horta hidropônica.

É importante ressaltar que o website tem o propósito de fazer o primeiro contato com o cliente, apresentar a proposta de trabalho contemplando cada um dos itens mencionados, além de demonstrar visualmente de forma simplificada como funciona todo o processo.

Há três diferentes tipos de planos, onde cada um contempla uma gama de serviços, isso auxilia de acordo com o tamanho da horta necessária para o cliente. A variação de mensalidade especificado quantos sensores ou tipo de acesso o cliente terá.

Na guia “O Serviço”, está explicado mais profundamente sobre a análise dos cultivos hidropônicos, e que é possível definir os parâmetros de monitoração tudo de forma remota. As funcionalidades são controladas via aplicativo mobile, colocando o produtor da horta no controle dos horários de funcionamento dos diferentes dispositivos, podendo receber informações atualizadas do ambiente em tempo real.

Entre as características da solução, há o uso de plataformas de computação em nuvem, via internet, para armazenar, checar, salvar e analisar os dados coletados para posterior entendimento do andamento dos cultivos.

Outro item que consta no site é quem são os membros responsáveis pela criação e desenvolvimento do serviço. Foi importante deixar registrado que a empresa surgiu de um trabalho de conclusão de curso, sendo que o foco do serviço é totalmente explicitado naquilo que ele contribui não só para o cliente, mas também para o meio ambiente.

Aliando o produto com o serviço se pode definir a escala do cultivo de acordo com a especificidade do cliente, de maneira que, por ser um ambiente controlado é possível se ter um maior controle em relação a vários aspectos como pragas, periodicidade, desperdício e a relação cultivo versus colheita.

3.4.3. Implementação do Aplicativo

Um fluxograma é uma representação gráfica simples de um processo, seguido por etapas. Cada figura do fluxograma tem a sua representação e significado único. O fluxograma é utilizado quando existe a necessidade de explicar um processo, algoritmo ou algum fluxo.

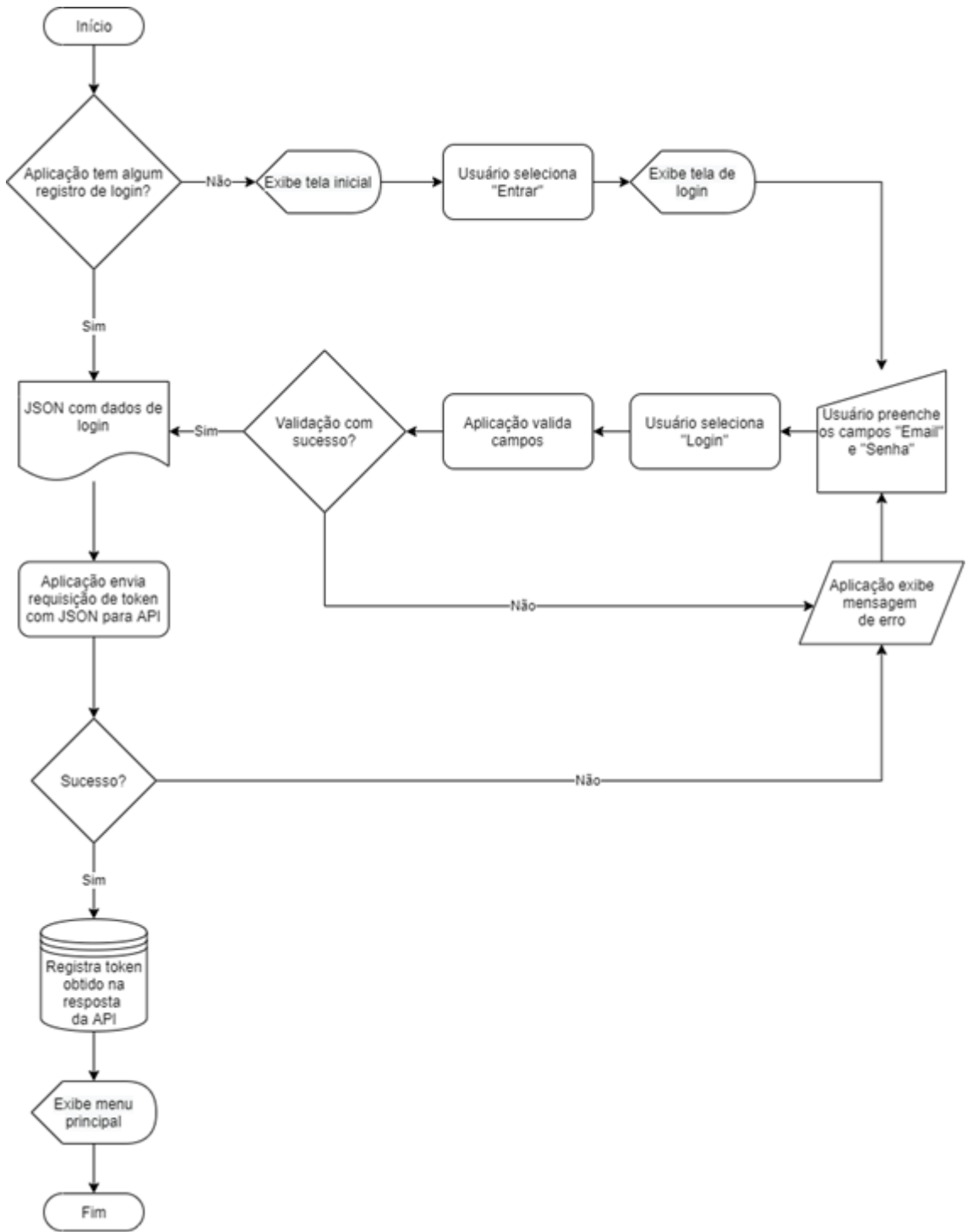
Sua utilização serve para descrever um fluxo de trabalho de forma visual (com símbolos) de como será a rotina de maneira sequencial, através de linhas e setas. Também pode ser utilizado juntamente a outros processos, representando conexões entre os fluxogramas.

William Adkins (2017, p.1) define o assunto

An important benefit of flowcharts is that they provide documentation of a process. This is useful when problems arise because flow charts let you trace the process from one step to the next. This lets you easily identify when the problem took place. The flowchart also provides guidance for managers overseeing operations and helps ensure compliance with policy and regulations.

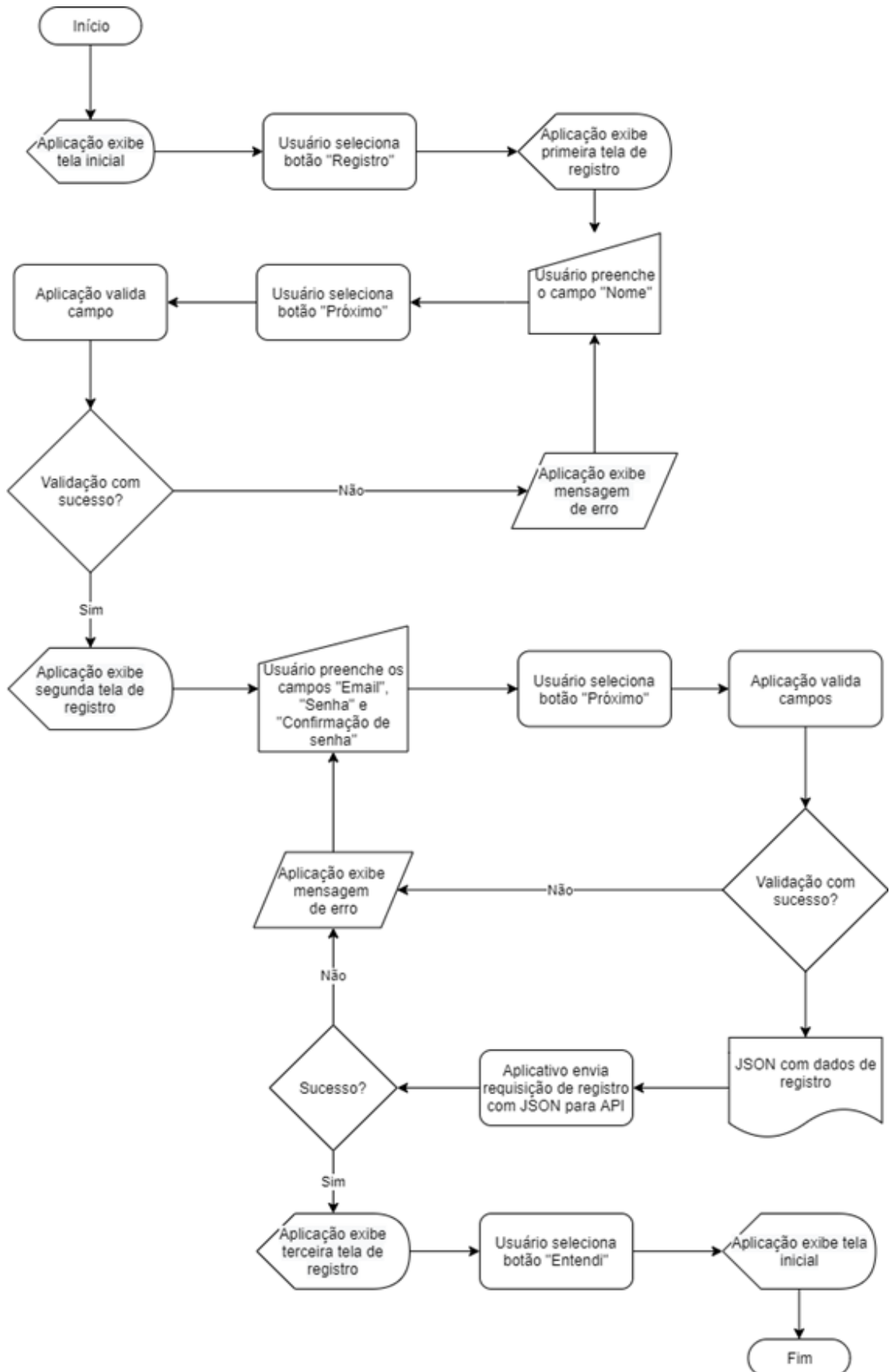
Devido a sua clareza e eficiência no visual, o fluxograma foi escolhido para representar o fluxo de trabalho do aplicativo. Os seguintes fluxogramas descrevem o comportamento das interações do aplicativo com o usuário e aplicativo com a API.

Figura 62 - Fluxograma de Login.



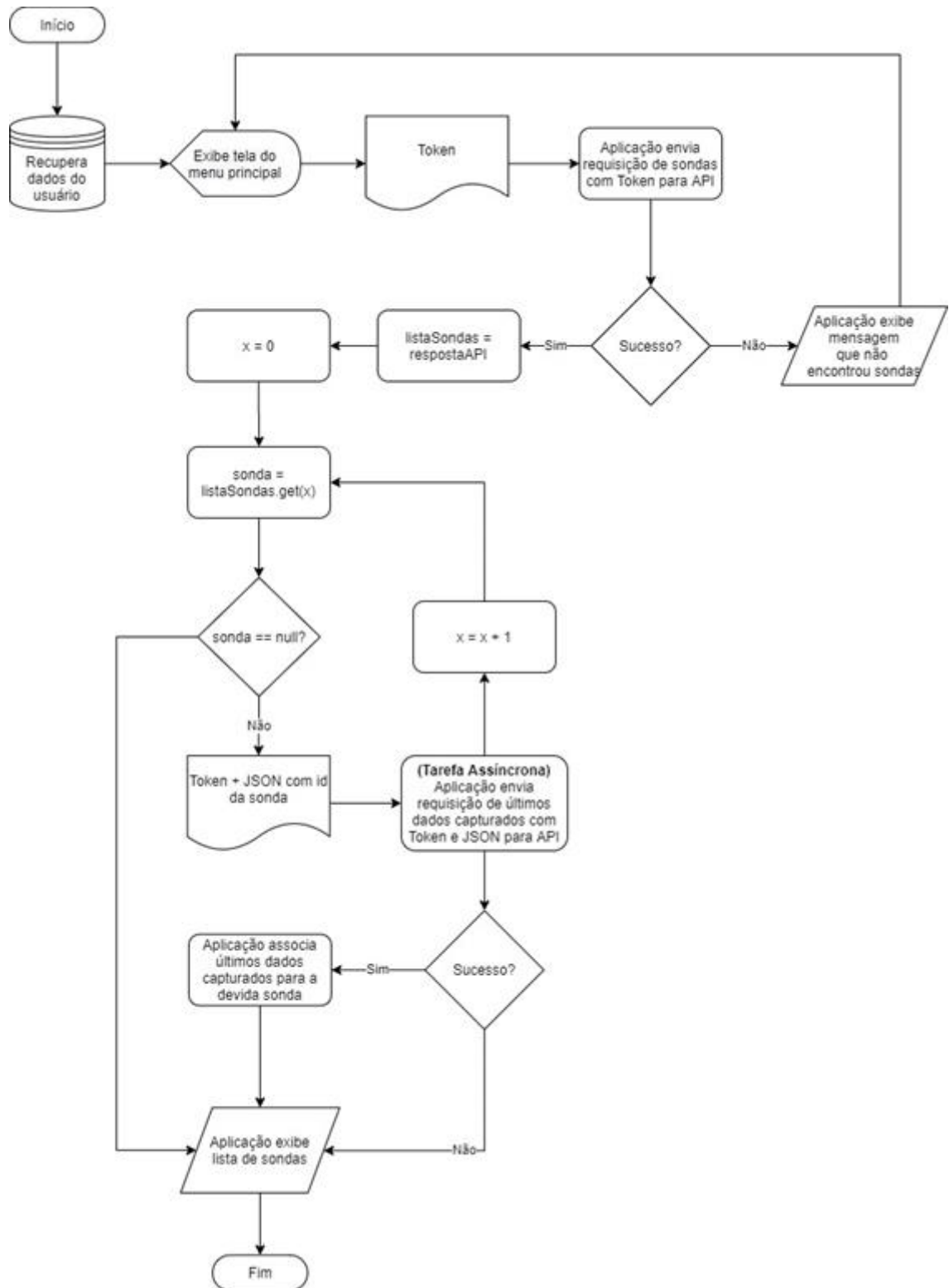
Desenvolvido por Aleksander Torquato 2020

Figura 63 - Fluxograma de Registro.



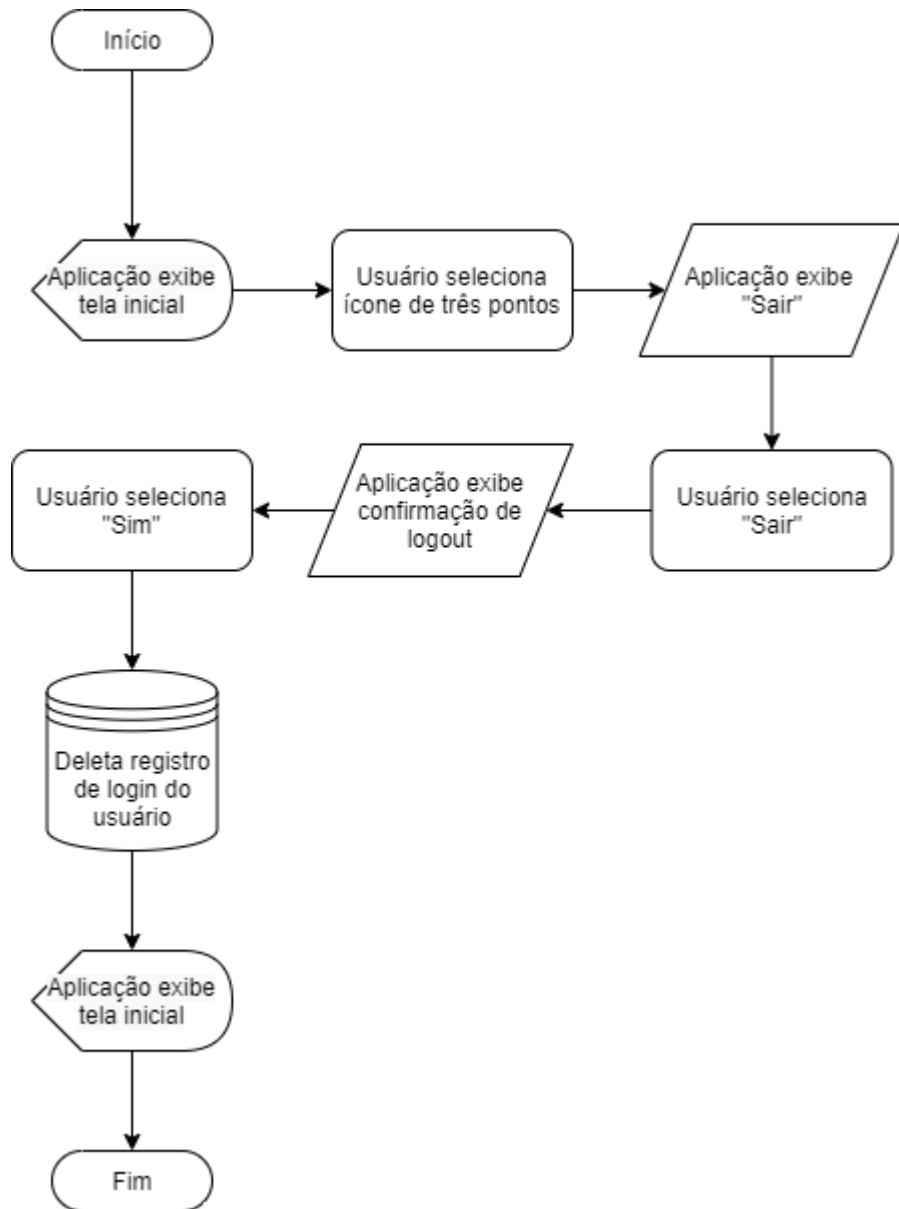
Os seguintes fluxogramas descrevem o comportamento do aplicativo a partir da tela do Menu Principal do aplicativo.

Figura 64 - Fluxograma exibição da tela do Menu Principal.



Desenvolvido por Alecsander Torquato 2020.

Figura 65 - Fluxograma de Logout.



Desenvolvido por Alecsander Torquato 2020

Figura 66 - Fluxograma de Configuração de dispositivos

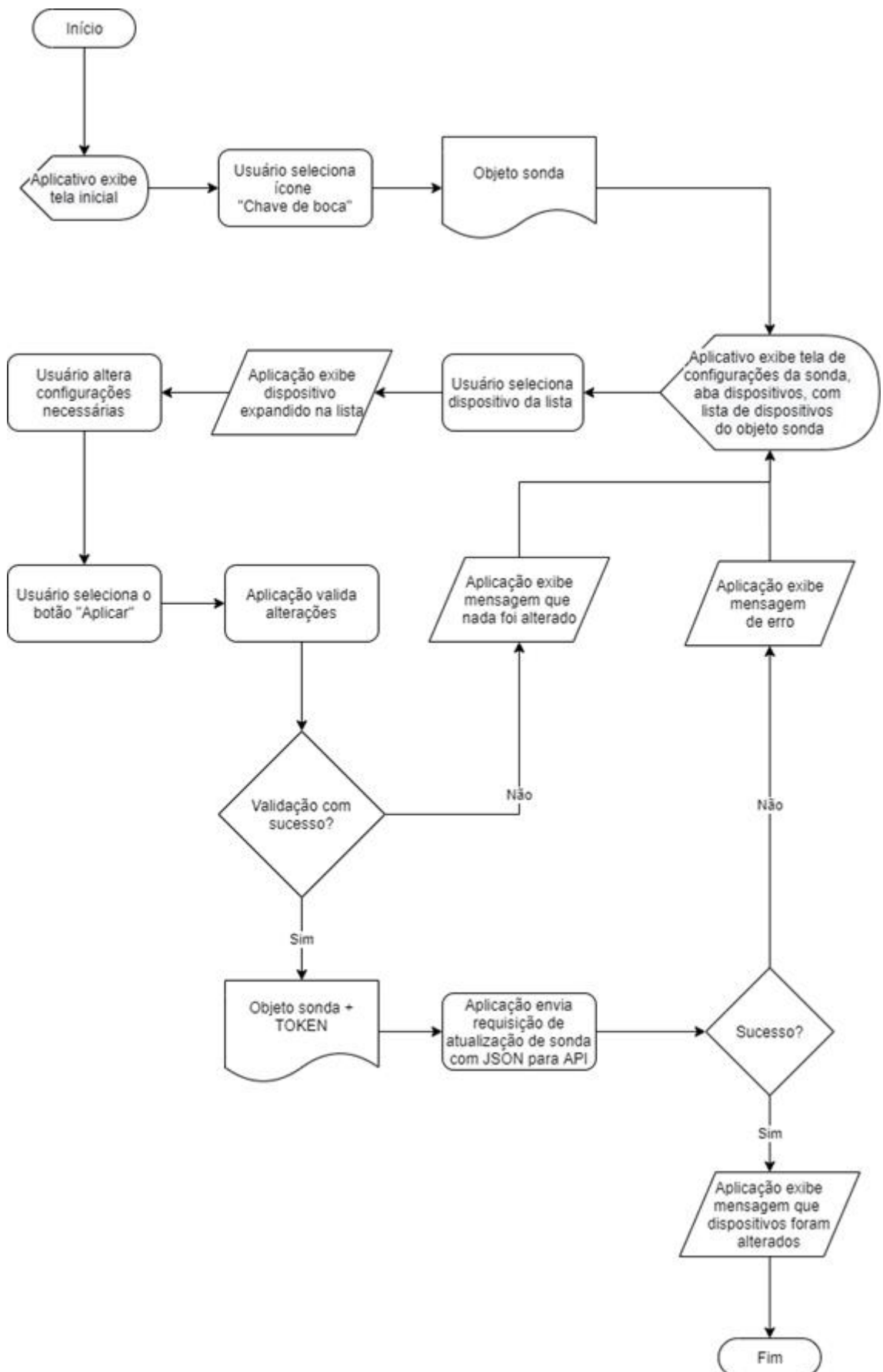
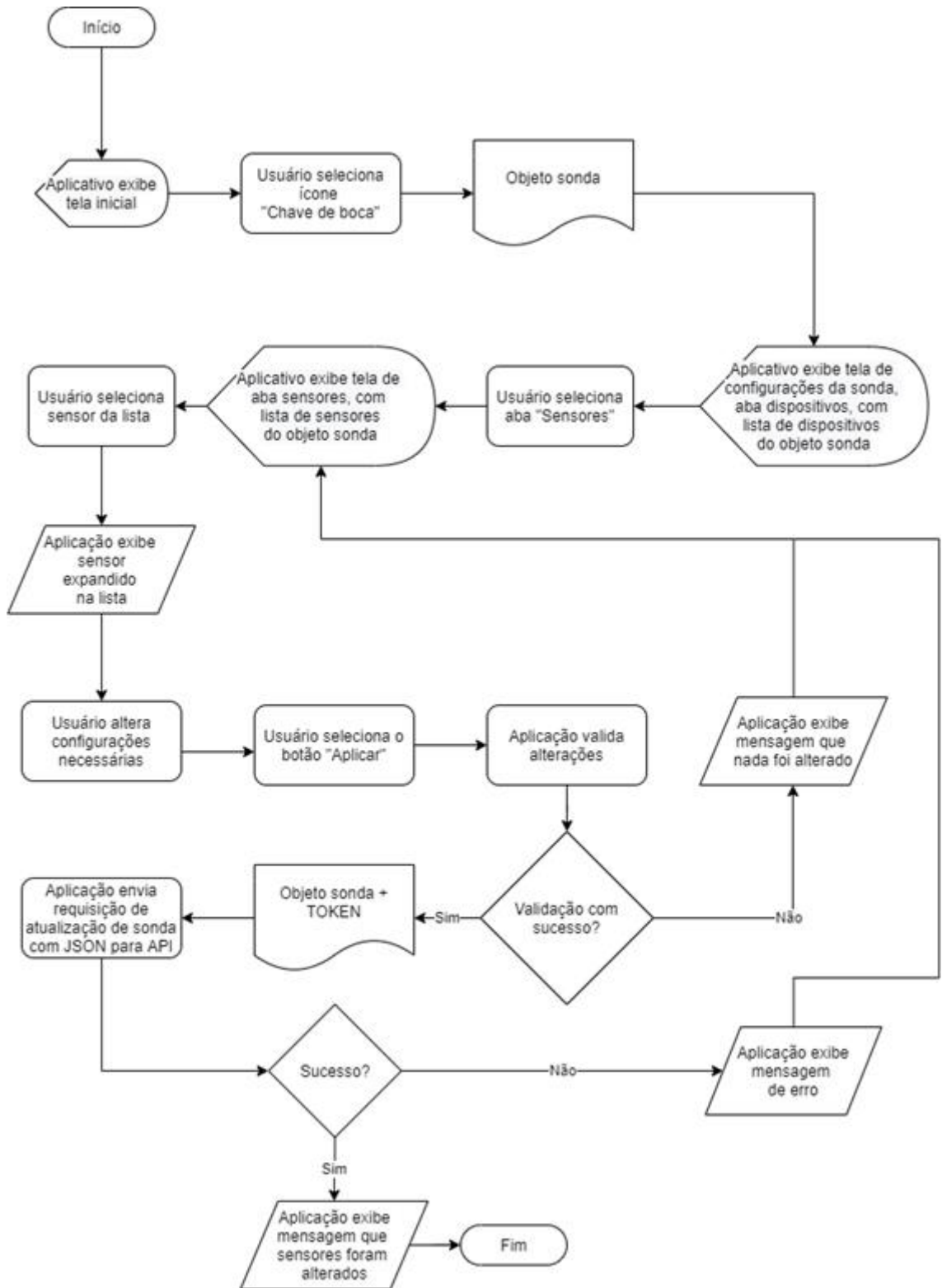


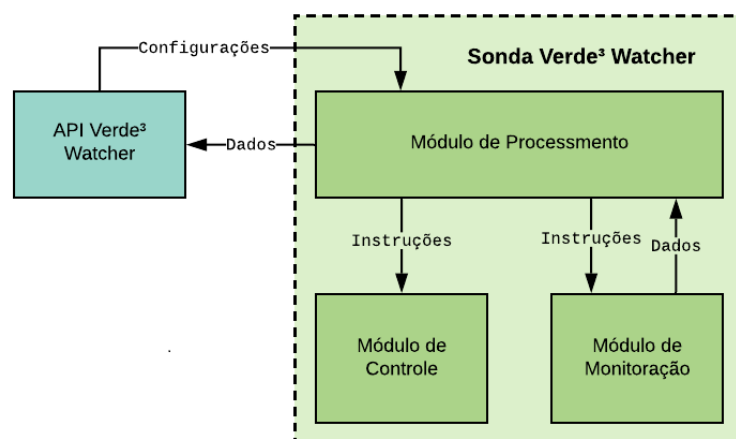
Figura 67 - Fluxograma de Configuração de sensores



3.4.4. Implementação da Sonda

O protótipo realizado da sonda e dos módulos exibem a configuração dos componentes físicos da sonda. Quando ao software que irá ser executado no dispositivo, antes de implementá-lo, foi criado um diagrama de contexto (Figura 68) para definir os limites da sonda e de seus componentes.

Figura 68 - Diagrama de contexto das sondas da solução Verde³ Watcher.



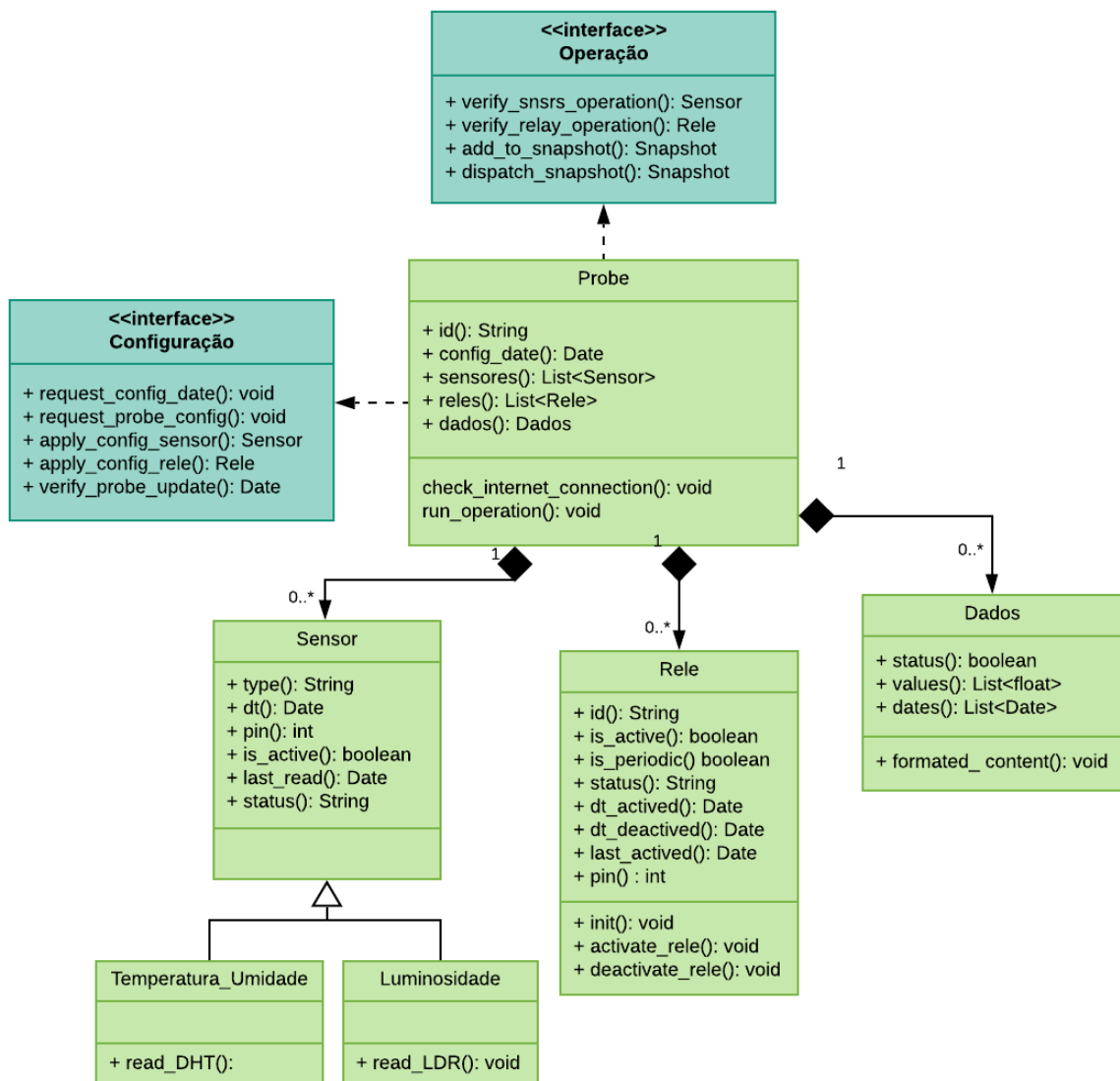
Desenvolvido por Caio Cabezas 2020

No contexto da solução, a sonda é responsável pelo controle de dispositivos e sensores, que ocorre conforme as configurações de operação recebidas da API Verde³ Watcher. A sonda também é responsável por enviar à API os dados coletados pelos sensores. As sondas possuem um ID único e que permite acesso aos recursos da API. O sistema possui três componentes, cujo o principal é o módulo de processamento, responsável por receber configurações da API, encaminhar instruções aos módulos de controle e de monitoração e, receber e enviar dados a API. O módulo de controle responde pela execução de instruções recebidas. O módulo de monitoração responde pela execução de instruções e pelo de dados ao módulo de processamento.

O design de arquitetura de software preocupa-se em entender como um sistema de deve ser organizado e como ele deve ser o design geral de suas estruturas (SOMMERVILLE, 2010). A arquitetura deste sistema foi modelada sob o ponto de vista de processo e de desenvolvimento.

A arquitetura sob o ponto de vista de desenvolvimento exibe como o software será decomposto para o desenvolvimento, isto é, ela mostra a separação do software em componentes implementáveis (SOMMERVILLE, 2010). Considerando as principais atribuições da sonda, foi gerado um modelo estrutural do sistema, que descreve a estrutura estática do sistema utilizando classes de objetos e suas relações (SOMMERVILLE, 2010). Este modelo é apresentado no diagrama de classe da Figura 69.

Figura 69 - Diagrama de classe da sonda



Desenvolvido por Caio Cabezas 2020

A classe Probe representa uma sonda do sistema e possui uma identificação, sensores, reles, dados e uma data de configuração. A sonda pode verificar a sua conectividade e iniciar a operação dos sensores e reles.

A interface Configurações representa as operações de configuração dos sensores e reles da sonda. Ela possui operações para obter a data da última atualização feita no sistema, obter a configuração atual da sonda, aplicar configurações em sensores e reles e verificar se existem atualizações.

A interface Operações representa o conjunto de operações de verificação de acionamento de sensores e reles, e de verificação de envio de dados. Ela possui operações para controlar sensores, reles e o envio de dados.

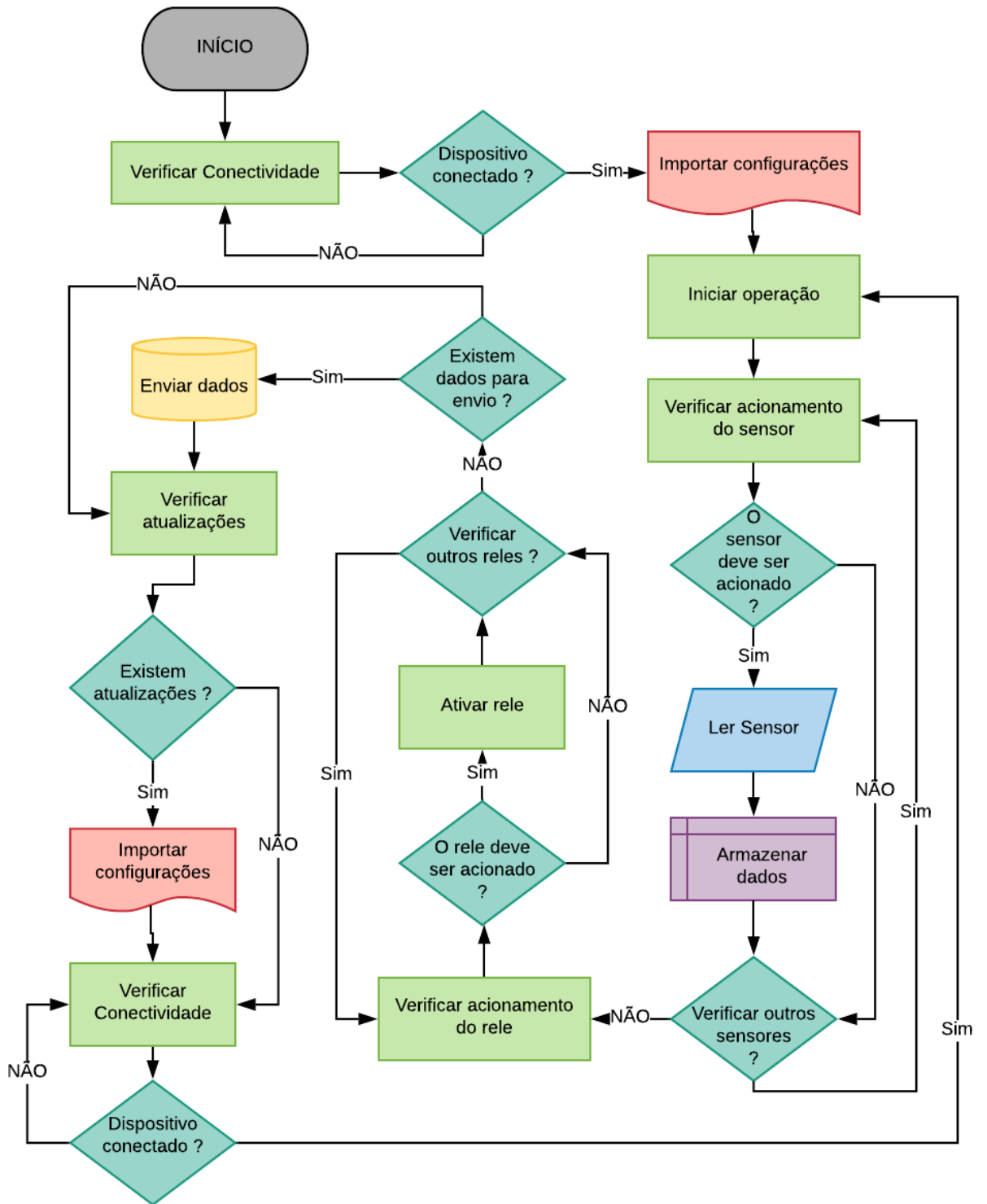
A classe Rele representa um rele presente no módulo de controle da sonda. Possui uma identificação, um estado de acionamento, um estado de periodicidade, um estado de operação, um tempo de atividade, um tempo de inatividade, uma data contendo o horário da última ativação e um pino para a comunicação com o rele físico. O rele pode ser, inicializado, ativado e desativado.

A classe Sensor representa um sensor presente no módulo de monitoração e possui um tipo (LDR ou DHT), um intervalo de acionamento, um pino para a comunicação com o sensor físico, um estado de ativação, uma data contendo o horário da última leitura e um estado de operação. O sensor pode fazer leituras.

A classe Dados representa um agrupamento de dados enviado pelos sensores. Possui um estado de envio, valores e datas dos últimos dados coletados. Os dados podem ter seu conteúdo formatado antes de serem enviado à API.

A arquitetura sob o ponto de vista de processo exibe um sistema como uma sequência de interação de processos (SOMMERVILLE, 2010). Considerando as atribuições da sonda, seu sistema foi modelado de acordo com a sequência de processos exibida no fluxograma da Figura 70. É importante informar que os processos que controlam o acionamento de sensores e reles da sonda são dependentes do tempo, pois estes recursos devem operar em intervalos temporais pré definidos pelo usuário ao configurar a sonda. Estes processos são identificados no modelo como “verificar acionamento do sensor” e “verificar acionamento do rele” e representam o processo de verificar a diferença entre dois instantes de tempo com relação a um limite, determinado pelas configurações da sonda, indicando se um sensor ou rele deve ou não ser ativado.

Figura 70 - Fluxograma com o modelo de interação de processos da sonda.

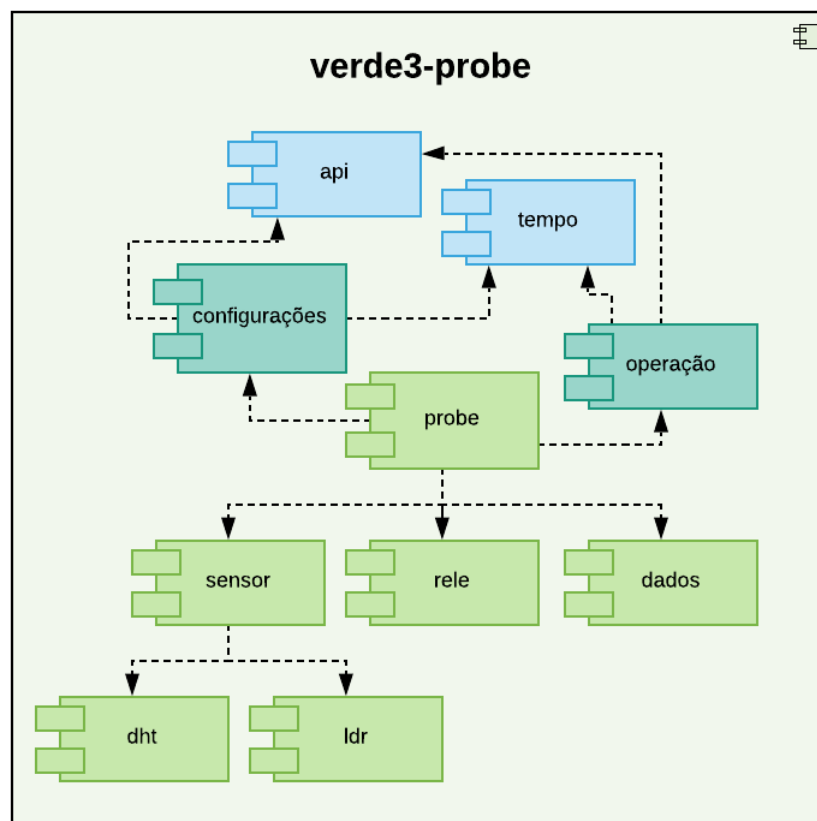


Desenvolvido por Caio Cabezas 2020

O sistema foi implementado de acordo com os processos modelados no fluxograma da Figura 70 e com o modelo estrutural do sistema, apresentado no diagrama de classes da Figura 69.

O sistema foi programado em python 3. Todos as classes e interfaces do diagrama de classe (Figura X.Y) foram implementados como componentes do software verde3-probe.py, como mostra o diagrama de componentes da Figura X.Y. Todos os componentes do programa verde3-probe estão contidos em um único módulo (diretório /home/pi/verde3-probe) que contém as classes de probe, sensores, reles, dados, operação, configuração e as bibliotecas api e tempo, utilizadas para operações via API e operações com unidades de tempo. Os componentes operação e configuração dependem destas bibliotecas. A probe integra todos componentes do programa, por que é dependente de todos eles. O sensor depende dos componentes ldr e dht.

Figura 71 - Diagrama de componentes do programa verde3-probe



Desenvolvido por Caio Cabezas 2020

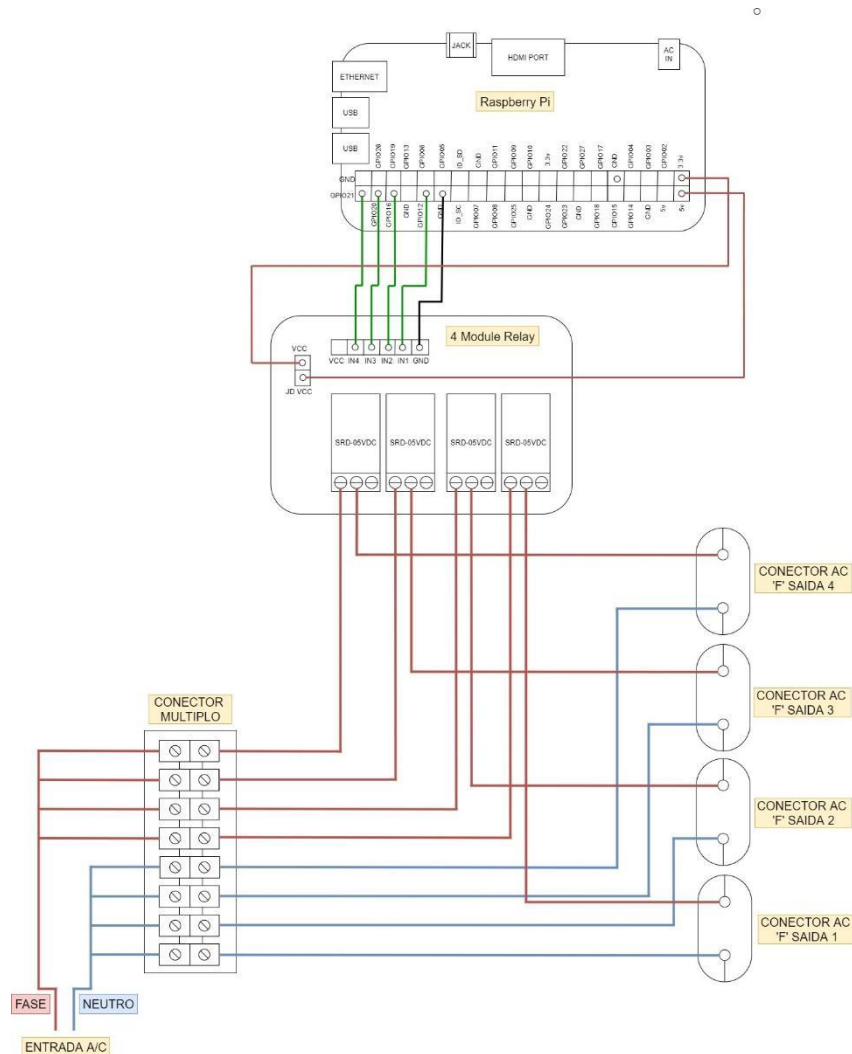
3.5. Montagem da Sonda

A sonda solução Verde³ Watcher é composta por três módulos. Esta seção irá apresentar os procedimentos para a montagem dos módulos de controle, monitoração e processamento da sonda, incluindo como estão conectados,

informando detalhes sobre os pinos GPIO utilizados do módulo de processamento, um Raspberry Pi 3 Model B.

O módulo de controle é composto pela interface de reles 5V com 4 canais, quatro tomadas padrão 2P+T 10A/250V, um conector múltiplo com 2.5mm de diâmetro, cabos jumper M/F e fios condutores de 2.5mm (Figura 72).

Figura 72 - Disposição dos recursos do módulo de controle da sonda e suas conexões com as portas GPIO do módulo de processamento.



Desenvolvido por Silas Chagas 2020

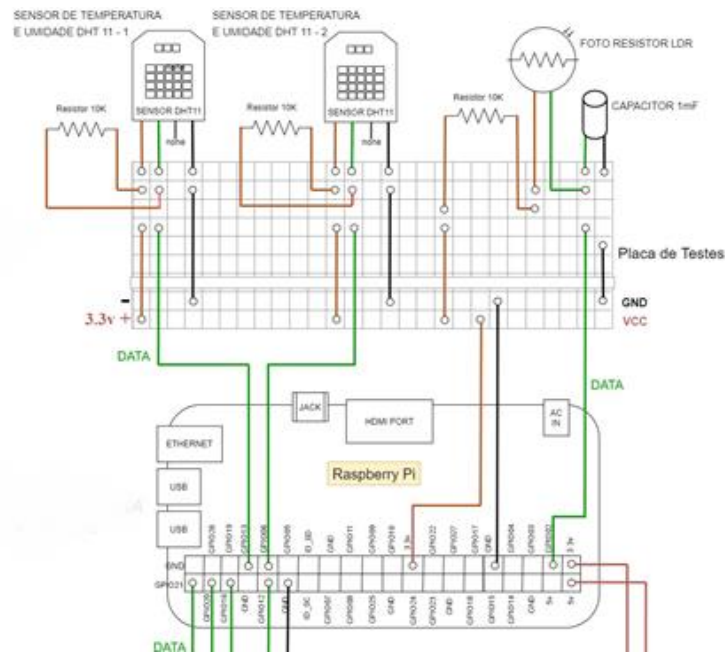
O processo de montagem deste módulo iniciou com a conexão de seus pinos de alimentação. Como os reles da placa de interface de possuem um optoacoplador, um recurso que inibe interferências elétricas no sistema, este componente do módulo foi alimentado com fontes de diferentes voltagens, uma de 5V para o controle do optoacoplador, que aciona o rele e outra de 3.3V que comanda o sinal para acionar

o rele (comunicação com a GPIO do Raspberry não pode ultrapassar 3,3V). O pino JD-VCC da placa de interface foi conectado na saída de 5V do Raspberry Pi e o pino VCC da placa na saída de 3,3V do Raspberry Pi. O GND (M. Rele) na entrada GND da Raspberry Pi

Para controlar os canais da placa de reles, conectamos as seguintes portas GPIO do Raspberry Pi nas entradas do da placa: GPIO 12 = IN1, GPIO 16 = IN2, GPIO 20 = IN3 e GPIO 21 = IN4. Na sequência, foram conectados as saídas do rele nas tomadas padrão 2P+T 10A 250V. Passando os cabos da fase através do modulo rele. Se o sinal for low na saída do rele, a saída será acionada, se for high o rele será interrompido. Cada canal da placa possui três conectores para o cabeamento. O contato interno será feito com o sinal digital do Raspberry Pi, liberando a conexão do cabo fase, o cabo neutro será ligado diretamente ao outro polo da tomada

O módulo de monitoração é composto por dois sensores DHT11, um fotoresistor LDR, uma placa ensaio eletrônica de 400 furos, um capacitor de 1 μ F, dois resistores de 10kOhm, um resistor de 2kOhm e cabos jumper M/F (Figura X.Y). O processo de montagem do módulo consistiu em montar dois circuitos para os sensores de temperatura e umidade DHT11 e montar um circuito para que o fotoresistor opere como um sensor.

Figura 73 - Disposição dos recursos do módulo de monitoração da sonda e suas conexões com as portas GPIO do módulo de processamento.



Desenvolvido por Silas Chagas 2020

A montagem dos sensores DHT11 no módulo incluiu afixar os sensores na placa de ensaio, e entre os pinos 1 e 2 fixar um resistor de 10kOhm (Figura 73). Na sequência foram conectados o pino 1 do sensor a uma saída da 3.3V da placa, que é alimentada pelo Raspberry Pi, o pino 2 foi conectado ao GPIO5 do Raspberry Pi para a transmissão de dados e o pino 4 foi conectado a uma das entradas de retorno da placa, conectada ao pino físico 6 do GPIO (GND).

A montagem sensor LDR incluiu afixar o fotoresistor LDR na placa de ensaio, fixar um resistor de 2kOhm entre seu pino de alimentação e a saída de 3.3v da placa (Figura 73), em seguida foi conectado o pino de saída do LDR ao pino de entrada de do capacitor de 1µF e ao pino GPIO2 do Raspberry Pi. Por último, foi conectado o pino de saída do capacitor a uma das entradas de retorno (GRD) da placa.

Para montar o módulo de processamento, foi necessário conectar o Raspberry Pi 3 Model B a fonte de 5V. Antes de iniciar o dispositivo, foi instalado uma distribuição 64bit do Raspberry Pi OS no dispositivo. Após a inicialização do dispositivo, foram adicionadas duas bibliotecas ao ambientes de variáveis do python 3, a saber: Adafruit_DHT e python-dateutil.

3.6. Implantação

De acordo com Venturebeat (2017). A Google Play atinge 8 bilhões de novas instalações de aplicativos por mês, obtém novos recursos para desenvolvedores e programa de recompensas por bugs.

A Google Play Store é uma loja de aplicativos desenvolvida pela Google. Seu objetivo principal é o de disponibilizar para download aos usuários diversos aplicativos de diversos assuntos.

Segundo Kristijan Lucic (2020). A Google Play Store abriga não apenas aplicativos e jogos, mas também alguns conteúdos digitais adicionais. Você pode encontrar filmes, livros e músicas na Play Store. A Google basicamente lançou a Play Store como um hub para seus usuários obterem todo tipo de conteúdo digital. O conteúdo principal provavelmente seria aplicativos e jogos, pois a Play Store foi lançada principalmente para smartphones Android.

Todos os aplicativos nesta loja podem ser baixados de acordo com a limitação de hardware e versão do Sistema Operacional do usuário, os custos variam de cada aplicativo, podendo ser pago ou gratuito. A Google Play Store implementa em sua loja um tipo de assinatura para os aplicativos, assim quando o desenvolvedor deseja disponibilizar o seu aplicativo nesse mercado, é necessário a criação de uma chave de identificação, dando uma assinatura para aquele aplicativo, chamado de assinatura Google.

A Google Play Store pode ser o principal mercado quando se trata de disponibilização de aplicativos Android, porém não é o único. Existem outros aplicativos de loja que servem e funcionam como a Google Play Store, disponibilizando seus aplicativos de forma parecida, porém essas lojas podem não ser tão confiáveis pois algumas não tratam questões importantes como compatibilidade de dispositivo por exemplo.

Além das lojas, podemos disponibilizar o aplicativo de uma forma desenvolvedor-usuário, onde o desenvolvedor gera um APK para instalação no dispositivo do usuário. Um APK é comparável a um arquivo executável do Windows, ele contém tudo que é necessário para instalação no dispositivo celular Android.

A aplicação mobile Verde ao Cubo Watcher não está sendo disponibilizada no mercado da Google Play Store, pois se trata de uma aplicação protótipo. Em questões de Sistema Operacional, a aplicação pode ser instalada em qualquer dispositivo Android que esteja com a versão Lollipop/Android 5.0 (API de nível 21) ou acima. Sua disponibilização inicial na fase de protótipo está disponível apenas pela geração de um arquivo APK.

Após a implementação do programa verde3-probe, todos seus componentes foram transferidos no diretório /home/pi/Verde3-Probe. Muito embora o sistema tenha sido implementado, a sonda ainda se encontra em nível de protótipo funcional e por este motivo este modelo poderá ser disponibilizado apenas para testes.

4. Testes

A definição de um caso de teste segundo Myers (2012):

“Software testing is a process, or a series of processes, designed to make sure computer code does what it was designed to do and, conversely, that it does not do anything unintended(p.2)”

Os testes são compostos de ações que vão validar cada um dos passos percorridos no diagrama, para validar o processo. Cada ação que o usuário tomar vai ter uma resposta. Podendo ser feitos manualmente ou automatizados.

Casos de testes é uma parte importante do projeto, pois demonstra a qualidade do produto que está sendo desenvolvido, segundo Chrissis (2003):

“O teste de software é reconhecido como um processo fundamental para obtenção de software de boa qualidade (ISO/IEC 2003)”.

Mas tudo isso vai ter um custo, mas é um custo que vai trazer retorno quando todo o sistema estiver pronto e totalmente funcional. A isso tal como afirma Habib et. Al 2008:

“qualquer melhoria de processo de software significante requer um investimento significativo, tempo e dinheiro”.

Para que os testes sejam bem sucedidos, é preciso que haja informações consistentes e completas de cada parte desse desenvolvimento, pois a falha vai

existir exatamente nas partes em que haja buracos ou falta de informação de acordo com Pressman (2002).

“Informações desatualizadas ou inconsistentes levam a tomada de decisões incorretas, implicando, entre outras coisas, em testes incompletos ou inapropriados no software.”

Os casos de testes servem para checar uma série de ações de diferentes partes do sistema, com o intuito de encontrar possíveis bugs, observando o tempo de resposta e como o sistema vai se comportar na interação com o usuário. Técnicas de testes também podem auxiliar nesse processo já que possuem focos específicos no seu desenrolar.

Há um padrão IEEE-Std-829 (que define em geral as atividades em testes, os procedimentos e casos de uso), esse material auxilia o período de testes e pode se adequar a cada especificidade exigida.

Os casos de testes estão totalmente ligados a várias outras partes do escopo do projeto, onde é possível perceber, na grande maioria dos casos, se algum item dos passos anteriores foi feito de maneira correta ou não. Conforme indica Sommerville (2011):

Geração de casos de teste. Os requisitos devem ser testáveis. Se os testes forem concebidos como parte do processo de validação, isso frequentemente revela problemas de requisitos. Se é difícil ou impossível projetar um teste, isso normalmente significa que os requisitos serão difíceis de serem implementados e devem ser reconsiderados. O desenvolvimento de testes a partir dos requisitos do usuário antes de qualquer código ser escrito é parte integrante do Extreme Programming.(p.77)

O próprio teste é dividido em etapas, como teste de desenvolvimento, teste de sistema e teste de aceitação. No primeiro item, conforme Sommerville (2011) quem desenvolve o sistema vai realizando os testes:

Os componentes do sistema são testados pelas pessoas que o desenvolveram. Cada componente é testado de forma independente, separado dos outros. Os componentes podem ser entidades simples, como funções ou classes de objetos, ou podem ser agrupamentos coerentes dessas entidades

Já no segundo teste, para se formar um sistema completo, é necessário que haja uma integração dos componentes. O terceiro teste de aceitação, segundo Sommerville (2011), são usados dados fornecidos pelo próprio cliente:

Esse é o estágio final do processo de testes, antes que o sistema seja aceito para uso operacional. O sistema é testado com dados fornecidos pelo cliente, e não com dados advindos de testes simulados. O teste de aceitação pode revelar erros e omissões na definição dos requisitos do sistema, pois dados reais exercitam o sistema de formas diferentes dos dados de teste. Os testes de aceitação também podem revelar problemas de requisitos em que os recursos do sistema não atendam às necessidades do usuário ou o desempenho do sistema seja inaceitável.

Todo esse o processo de teste traz um peso de documentar todo o projeto, detalhando cada parte, item, teste, de maneira completa.

Segundo Phillips (1998), ele sugere que os testes de um Software em sistemas que não sejam muito grandes ou complexos, utilize:

Norma IEEE-std-829- 1998: plano de teste, especificação de procedimento de teste e relatório-resumo de teste é um aspecto crítico para seu sucesso, sendo necessário um projeto detalhado e o mais completo possível. Uma documentação criteriosa, bem estruturada, baseada em informações atualizadas e consistentes, melhora a visibilidade das várias fases do teste, contribui efetivamente para o controle gerencial e técnico da sua execução e pode ser utilizada como mecanismo de comunicação entre as partes interessadas e envolvidas.

Assim, conforme afirma Pressman (2006) sobre projetos de casos de uso:

“The design of tests for software and other engineering products can be as challenging as the initial design of the product itself. Yet ... software engineers often treat testing as an afterthought, developing test cases that ‘feel right’ but have little assurance of being complete. Recalling the objectives of testing, we must design tests that have the highest likelihood of finding the most errors with a minimum amount of time and effort.”

Portanto, o resultado das entradas e saídas de dados, é ter a real noção sobre o desenvolvimento do sistema.

Documentação dos Testes do Software de Verde ao Cubo

Esse teste foi desenvolvido com base no padrão IEEE-std-829 que tem como um dos critérios de avaliação o uso de 4 tabelas:

1º. Plano de Testes;

2º. Caso de Testes;

3º. Relatório de Incidente;

4º. Relatório Resumo de Teste

Tabela 5. Plano de Teste do sistema Verde Ao Cubo

Plano de Teste
Nome do Projeto: Sistema de Monitoração Verde ao Cubo
Pessoas Envolvidas/Responsabilidade
Dennys M. Braghirolli – Criação de casos de testes.
Alecsander T. -Criação de casos de testes.
Silas Chagas. - Execução de testes
Funcionalidades ou Módulos
Cadastro de Administrador, Cadastro de Cliente, Leitura de dados e Relatórios
Equipamentos/Softwares
O sistema deve funcionar em um servidor Web com acesso via browser em desktop ou aplicativos Mobile
Cronograma
Data de Início e Fim do Projeto: 05/02/2020 – 15/07/2020
Data de Início e Fim do Teste: 15/06/2020 - 30/06/2020
Local de Testes
O sistema não possui um local específico para testes. Os mesmos serão executados nas máquinas das pessoas envolvidas em locais aleatórios
Critérios para considerar o teste finalizado:
O teste será considerado finalizado ao preencher a coluna resultado do teste do documento de casos de testes como “Executado com Sucesso”.

Observação:

O relatório de incidente será enviado para todos os desenvolvedores por email assim que alguma alteração tenha sido feita. Serão criados os casos de teste, os relatórios de incidentes e o relatório de resumo de teste

Caso de Testes			
Nome do Projeto: Sistema de Monitoração Verde ao Cubo			
ID	Módulo	Descrição	Resultado do teste
1	Accionar dispositivo	Dentro da tela de configuração de sondas, o usuário acciona um dispositivo	A tela Configurações da Sonda, é exibida conforme descrito em Resultado Esperado. Pode ser selecionada a opção de configurações da Probe.
			Resultado Esperado
			<p>1) O aplicativo exibe uma tela "Configurações da Sonda" com duas abas: "Dispositivos" e "Sensores". A aba "Dispositivos" está previamente selecionada, dentro dessa aba o aplicativo carrega uma lista de dispositivos relacionados a PROBE escolhida</p> <p>2) O aplicativo expande o dispositivo escolhido, carregando opções de configuração</p> <p>3) O aplicativo efetua a ação relacionada ao switch. Se ele estava ativado deve desativar e vice-versa</p> <p>4) O aplicativo esconde a lista de dispositivos e o botão "Aplicar", e mostra um novo componente na tela, indicando que está havendo um loading. Quando o loading acaba, a lista e o botão são exibidos novamente, e a mensagem "Dispositivos atualizados com sucesso" é exibida na tela</p>
			Resultado do teste
2	Configurar dispositivo	Dentro da tela de configuração de sondas, o usuário configura um dispositivo	A tela Configurações da Sonda, é exibida conforme descrito em Resultado Esperado. Pode ser selecionada a opção de configurações da Probe.
			Resultado Esperado
			<p>1) Seleciona a opção de "Configurações" identificada pelo ícone "Chave de boca" dentro do CARD da probe exibida</p> <p>2) Seleciona um dispositivo da lista</p> <p>3) Ativar/desativa o switch mostrado ao lado da opção "Modo Intermitente"</p> <p>4) Seleciona o campo "Período de atividade"</p> <p>5) Escolhe a hora, minutos e segundos que deseja e seleciona a opção "OK"</p> <p>6) Necessário estar com o "Modo Intermitente" ativado. Seleciona o campo "Período de inatividade"</p> <p>7) Escolhe a hora, minutos e segundos que deseja e seleciona a opção "OK"</p> <p>8) Seleciona o botão de "Aplicar"</p>
			Resultado do teste
			<p>1) O aplicativo exibe uma tela "Configurações da Sonda" com duas abas: "Dispositivos" e "Sensores". A aba "Dispositivos" está previamente selecionada, dentro dessa aba o aplicativo carrega uma lista de dispositivos relacionados a PROBE escolhida</p> <p>2) O aplicativo expande o dispositivo escolhido, carregando opções de configuração</p> <p>3) O aplicativo efetua a ação relacionada ao switch. Se ele estava ativado deve desativar e vice-versa. Caso o switch tenha sido ativado, o aplicativo habilita para seleção os campos: "Período de Atividade" e "Período de Inatividade". Caso tenha sido desativo, os campos "Período de Atividade" e "Período de Inatividade" são desabilitados de seleção</p> <p>4) O aplicativo mostra um novo componente para escolha de hora, minutos e segundos</p> <p>5) O aplicativo fecha o componente exibido e atualiza o campo de "Período de atividade" com o tempo escolhido</p> <p>6) O aplicativo mostra um novo componente para escolha de hora, minutos e segundos</p> <p>7) O aplicativo fecha o componente exibido e atualiza o campo de "Período de Inatividade" com o tempo escolhido</p> <p>4) O aplicativo esconde a lista de dispositivos e o botão "Aplicar", e mostra um novo componente na tela, indicando que está havendo um loading. Quando o loading acaba, a lista e o botão são exibidos novamente, e a mensagem "Dispositivos atualizados com sucesso" é exibida na tela</p>
			Os ajustes selecionados são fixados no mostrador da tela em "DISPOSITIVOS".
			Os ajustes selecionados são fixados no mostrador da tela em "DISPOSITIVOS".
			O aplicativo exibe um relógio para a escolha do tempo de atividade do dispositivo, conforme descrito.
			O aplicativo exibe um relógio para a escolha do tempo de inatividade do dispositivo.
			O botão para ligar ou desligar o dispositivo responde ao toque, o botão de modo intermitente, funciona como esperado, quando acionado permite que ajustes sejam realizados no período de atividade ou inatividade do dispositivo.
			As opções de dispositivo são exibidas, e todos os dispositivos podem ser expandidos e todos os botões dos dispositivos respondem ao toque de ligar/desligar
			Após os ajustes selecionados, e pressionado o botão "Aplicar", os ajustes foram gravados. O dispositivo é ligado ou desligado. O teste foi realizado com sucesso
			Todos os dispositivos respondem ao acionamento e as opções de intervalo de atividade/inatividade podem ser alterados
			As opções de dispositivo são exibidas, e todos os dispositivos podem ser expandidos e todos os botões dos dispositivos respondem ao toque de ligar/desligar

Desenvolvido por Silas Chagas de Moraes 2020

ID	Módulo	Descrição	Roteiro	Resultado Esperado	Resultado do teste
3	Accionar sensor	Dentro da tela de configuração de sondas, o usuário aciona um sensor	<p>1) Necessário estar logado, dentro do menu principal. Seleciona a opção de "Configurações" identificada pelo ícone "Chave de boca" dentro do CARD da probe exibida</p> <p>2) Seleciona a aba "Sensores"</p> <p>2) Seleciona um sensor da lista</p> <p>3) Seleciona o switch mostrado a baixo da opção "Estado"</p> <p>4) Seleciona o botão de "Aplicar"</p>	<p>1) O aplicativo exibe uma tela "Configurações da Sonda" com duas abas, "Dispositivos" e "Sensores". A aba "Dispositivos" está previamente selecionada</p> <p>2) O aplicativo carrega uma lista de sensores relacionados a PROBE escolhida</p> <p>2) O aplicativo expande o sensor escolhido, carregando opções de configuração</p> <p>3) O aplicativo efetua a ação relacionada ao switch. Se ele estava ativado deve desativar e vice-versa</p> <p>4) O aplicativo esconde a lista de dispositivos e o botão "Aplicar", e mostra um novo componente na tela, indicando que está havendo um loading. Quando o loading acaba, a lista e o botão são exibidos novamente, e a mensagem "Sensores atualizados com sucesso" é exibida na tela</p>	<p>A tela Configurações da Sonda, é exibida conforme descrito em Resultado Esperado. Pode ser selecionada a opção de configurações da Probe.</p> <p>A aba sensores responde conforme descrito em Resultado Esperado.</p> <p>O sensor escolhido é expandido conforme descrito em Resultado Esperado.</p> <p>O sensor pode ser ligado ou desligado conforme descrito em Resultado Esperado</p> <p>No momento em que é pressionado "Aplicar" o sensor liga (envia dados) ou desligado envia mais dados de captura). A mensagem "Sensores atualizados com sucesso" é exibida. Pode se observar na tela inicial a atualização conforme a seleção ajustada. O teste foi realizado com sucesso.</p>
4	Configurar sensor	Dentro da tela de configuração de sondas, o usuário configura um sensor	<p>1) Necessário estar logado, dentro do menu principal. Seleciona a opção de "Configurações" identificada pelo ícone "Chave de boca" dentro do CARD da probe exibida</p> <p>2) Seleciona a aba "Sensores"</p> <p>2) Seleciona um sensor da lista</p> <p>3) Seleciona a configuração "Intervalo de medição"</p> <p>7) Escolhe a hora, minutos e segundos que deseja e seleciona a opção "Ok"</p> <p>8) Seleciona o botão de "Aplicar"</p>	<p>1) O aplicativo exibe uma tela "Configurações da Sonda" com duas abas, "Dispositivos" e "Sensores". A aba "Dispositivos" está previamente selecionada</p> <p>2) O aplicativo carrega uma lista de sensores relacionados a PROBE escolhida</p> <p>2) O aplicativo expande o sensor escolhido, carregando opções de configuração</p> <p>3) O aplicativo mostra um novo componente para escolha de hora, minutos e segundos</p> <p>7) O aplicativo fecha o componente exibido e atualiza o tempo mostrado a baixo do texto "Intervalo de medição" com o tempo escolhido</p> <p>4) O aplicativo esconde a lista de dispositivos e o botão "Aplicar", e mostra um novo componente na tela, indicando que está havendo um loading. Quando o loading acaba, a lista e o botão são exibidos novamente, e a mensagem "Sensores atualizados com sucesso" é exibida na tela</p>	<p>A tela Configurações, é exibida conforme descrito em Resultado Esperado. Pode ser selecionada a opção de configuração da Probe</p> <p>A aba sensores responde conforme descrito em Resultado Esperado.</p> <p>O sensor escolhido é expandido conforme descrito em Resultado Esperado.</p> <p>O teste corresponde ao descrito em "Resultado Esperado"</p> <p> Ao pressionar o botão "Aplicar", verificou-se que os ajustes sofreram alterações nos intervalos selecionados, através do menu principal. O teste foi realizado com sucesso.</p>
5	Registrar usuário	Registrar um novo usuário	<p>1) Necessário estar na tela de escolha para login ou registro. Seleciona o botão "Registrar"</p> <p>2) Informa o nome para registro e seleciona o botão "Próximo"</p> <p>2) Informa os dados nos campos de "Email", "Senha" e "Confirmação de senha", e seleciona o botão "Próximo"</p> <p>3) Seleciona o botão "Entendi"</p>	<p>1) O aplicativo exibe uma nova tela de "Registro"</p> <p>2) O aplicativo valida o nome informado e exibe uma nova tela para inserção de dados nos campos de "Email", "Senha" e "Confirmação de senha"</p> <p>2) O aplicativo valida o email inserido e as senhas inseridas, e abre uma nova tela informando ao usuário "Enviamos uma confirmação de registro para seu email. Depois de confirmar o email, é só entrar com a sua conta pelo aplicativo"</p> <p>3) O aplicativo retorna para a tela de escolha de login ou registro</p>	<p>O aplicativo apresentou a tela de Registro.</p> <p>O teste corresponde ao Resultado Esperado.</p> <p>O aplicativo valida a entrada de dados, (precisa estar com no mínimo 8 caracteres). Foi informado um erro "Não foi possível realizar o registro. Tente novamente mais tarde"</p>

Desenvolvido por Silas Chagas de Moraes 2020

6	Login	Usuário realiza o login	<p>Roteiro</p> <p>1) Necessário estar na tela de escolha para login ou registro. Seleciona o botão "Login"</p> <p>2) Informa o email no campo "Email" e a senha no campo "Senha"</p>	<p>1) O aplicativo exibe uma nova tela "Login"</p> <p>2) O aplicativo valida o acesso e exibe o menu principal do aplicativo, informando o nome do usuário e um componente loading que ao terminar exibe uma lista de probes do usuário</p>	<p>O aplicativo apresentou a tela Login realizado com sucesso.</p>
7	Modulo Login automático	Aplicativo realiza o login automático	<p>Roteiro</p> <p>1) Necessário estar logado. Usuário fecha o aplicativo e abre de novo</p>	<p>Resultado Esperado</p> <p>1) O aplicativo exibe uma nova tela ao usuário, informando "Entrando..." com um componente loading. Quando o loading acaba, o menu principal do aplicativo é exibido, informando o nome do usuário e um componente loading que ao terminar exibe a lista de probes do usuário</p>	<p>No momento em que o aplicativo se principal do Android for acessado, o aplicativo é mostrado rapidamente. O teste foi executado com sucesso.</p>
8	Modulo Logout	Usuário realiza o logout do aplicativo	<p>Roteiro</p> <p>1) Necessário estar logado, dentro do menu principal. Seleciona o menu no topo direito da tela, indicado por um ícone de três pontos</p> <p>2) Seleciona a opção "Sair"</p> <p>3) Seleciona a opção "Sim"</p>	<p>Resultado Esperado</p> <p>1) O aplicativo exibe a opção "Sair"</p> <p>2) O aplicativo mostra um novo componente na tela, informando "Você realmente deseja sair?"</p> <p>3) O aplicativo realiza o logout e exibe a tela de escolha para login ou registro</p>	<p>O aplicativo faz o logout, e nao retorn login nao for efetuado novamente. O teste foi realizado com sucesso.</p>
9	Modulo Gráfico de séries temporais	Usuário observa a curva de valores de sensores, através de gráfico.	<p>Roteiro</p> <p>1) Necessário estar logado, dentro do menu principal. Seleciona a opção de "Séries temporais", identificada pelo ícone "Histograma com curva" dentro do CARD da probe exibida</p> <p>2) Seleciona o menu drop down e escolhe uma opção entre as opções de período de medição apresentadas.</p>	<p>Resultado Esperado</p> <p>1) O aplicativo exibe uma nova tela "Séries Temporais", onde podem ser observados em um gráfico informações referentes aos períodos de medição de temperatura, umidade e luminosidade da probe escolhida. Pode-se deslizar o gráfico para selecionar outros períodos de medição.</p> <p>2) O aplicativo exibe somente os pontos apresentados de acordo com o período de medição escolhido pelo usuário.</p>	<p>O usuário pode deslizar o gráfico pelo período selecionado, para ver mais gráfico. O teste foi realizado com su</p>

Desenvolvido por Silas Chagas de Moraes 2020

6	Modulo	Descrição	Roteiro	Resultado Esperado	Resultado do teste
	Login	Usuário realiza o login	<p>1) Necessário estar na tela de escolha para login ou registro. Seleciona o botão "Login"</p> <p>2) Informa o email no campo "Email" e a senha no campo "Senha"</p>	<p>1) O aplicativo exibe uma nova tela "Login"</p> <p>2) O aplicativo valida o acesso e exibe o menu principal do aplicativo, informando o nome do usuário e um componente loading que ao terminar exibe uma lista de probes do usuário</p>	O aplicativo apresentou a tela Login
7	Modulo	Descrição	Roteiro	Resultado Esperado	Resultado do teste
	Login automático	Aplicativo realiza o login automático	<p>1) Necessário estar logado. Usuário fecha o aplicativo e abre de novo</p>	<p>1) O aplicativo exibe uma nova tela ao usuário, informando "Entrando..." com um componente loading. Quando o loading acaba, o menu principal do aplicativo é exibido, informando o nome do usuário e um componente loading que ao terminar exibe a lista de probes do usuário</p>	No momento em que o aplicativo está no menu principal do Android foi acessado, e o aplicativo é mostrado rapidamente a tela de login. O teste foi executado com sucesso.
8	Modulo	Descrição	Roteiro	Resultado Esperado	Resultado do teste
	Logout	Usuário realiza o logout do aplicativo	<p>1) Necessário estar logado, dentro do menu principal. Seleciona o menu no topo direito da tela, indicado por um ícone de três pontos</p> <p>2) Seleciona a opção "Sair"</p> <p>3) Seleciona a opção "Sim"</p>	<p>1) O aplicativo exibe a opção "Sair"</p> <p>2) O aplicativo mostra um novo componente na tela, informando "Você realmente deseja sair?"</p> <p>3) O aplicativo realiza o logout e exibe a tela de escolha para login ou registro</p>	O teste foi realizado com sucesso.
9	Modulo	Descrição	Roteiro	Resultado Esperado	Resultado do teste
	Gráfico de séries temporais	Usuário observa a curva de valores de sensores, através de gráfico.	<p>1) Necessário estar logado, dentro do menu principal. Seleciona a opção de "Séries temporais" identificada pelo ícone "Histograma com curva" dentro do CARD da probe exibida</p> <p>2) Seleciona o menu drop down e escolhe uma opção entre as opções de período de medição apresentadas.</p>	<p>1) O aplicativo exibe uma nova tela "Séries Temporais", onde podem ser observados em um gráfico informações referentes aos períodos de medição de temperatura, umidade e luminosidade da probe escolhida. Pode-se deslizar o gráfico para selecionar outros períodos de medição.</p> <p>2) O aplicativo exibe somente os pontos apresentados de acordo com o período de medição escolhido pelo usuário.</p>	E exibida uma tela com um botão de "opções de intervalos temporais como diárias, todas as medições. Ao selecionar o intervalo de tempo, o gráfico é exibido.
					O usuário pode deslizar o gráfico pela tela para selecionar o período de medição desejado. O teste foi realizado com sucesso.

Tabela 2.2.e Caso de Testes WebSite

Caso de Testes

Nome do Projeto: Sistema de Monitoração Verde ao Cubo

ID	Módulo	Descrição	Roteiro	Resultado Esperado	Resultado do teste
1	Website - Manter usuário	Cadastro de Usuário	<ol style="list-style-type: none"> 1) Clicar em "Registrar-se" 2) Preencher todos os campos com dados válidos 3) Clicar em "Registrar-se" 	A exibida uma mensagem de confirmação é exibida e é enviado um e-mail de confirmação ao usuário.	Os campos estão disponíveis, e podem ser preenchidos. Ao clicar em Registrar-se o e-mail é enviado para instruções de confirmação. O teste foi realizado com sucesso.
2	Website - validar interface visual e funcionalidade de links	Validar intuitividade das informações na interface	<ol style="list-style-type: none"> 1) Verificar se todas as guias levam aos seus respectivos campos; 2) Usuário não cadastrado e não autenticado no sistema; 3) Verificar funcionalidade dos links 	O usuário deve acessar o ítem em que clicou o botão e voltar a home	Guia correspondem a opção selecionada. Uma conta que não está cadastrada não pode ser logada. Os links redirecionam para outras paginas. O teste foi realizado com sucesso.
3	Website - validar interface visual	Validar as imagens da interface	<ol style="list-style-type: none"> 1) Verificar se todas se todos os campos estão com as respectivas imagens; 2) Pré-condição - Carregamento total da página; 3) Verificar se não há mensagens de erro no lugar das imagens 	A exibição de todas as imagens em todos os ítem da página.	Todas as guias correspondem ao resultado esperado. Somente a guia dos 'planos' possuem elementos.
4	Website - validar funcionalidades de botões e campos de preenchimento	Validar os botões.	<ol style="list-style-type: none"> 1) Verificar se o botão leva até o app; 2) Usuário não cadastrado e não autenticado no sistema; 3) Verificar funcionalidade do botão 	O botão leve ao ítem em que foi clicado, no caso, o app	O teste foi executado com sucesso, o botão 'App' realmente leva à interface do aplicativo.
5	Website - validar acesso de conta	Validar área de login	<ol style="list-style-type: none"> 1) Verificar se o usuário consegue se logar no site; 2) Usuário precisa estar logado; 3) Usuário deve clicar em Entrar na Home 	O usuário se logar no site.	O usuário preenche os campos porém o login não persiste.
6	Website - validar acesso de conta	Validar entrada do usuário	<ol style="list-style-type: none"> 1) Verificar se o cliente consegue se logar no site com seu usuário e senha; 2) Usuário precisa ter um cadastro; 3) Usuário deve clicar em Entrar na Home, colocar usuário e senha. 	Verificar se usuário consegue se logar e se cadastrar novamente com os mesmos dados.	O cadastro pode ser realizado, porém o acesso a conta não persiste.
7	Website - manter usuário	Validar criação de conta	<ol style="list-style-type: none"> 1) Verificar se o usuário consegue se cadastrar; 2) Usuário não pode estar cadastrado; 3) Usuário deve clicar em Registrar; 	Criar um novo usuário.	O cadastro pode ser realizado, porém o acesso a conta não persiste.
8	Website - manter usuário	Validar cadastro de um novo usuário	<ol style="list-style-type: none"> 1) Verificar se há duplicação de usuário; 2) Usuário não pode ter cadastro; 3) Usuário deve criar conta com nome, e-mail e senha 	Deve aparecer mensagem em caso de duplicidade de cadastro.	Ao tentar registrar uma conta com um e-mail já existente, o sistema bloqueia o cadastro e informa que o e-mail já tem cadastro. O teste foi realizado com sucesso.

Tabela 2.3 – Relatório de Incidente

Relatório de Incidente					
Nome do Projeto: Sistema de Monitoração Verde ao Cubo					
ID	Status	Responsável pela Correção	Prioridade da Correção	Descrição do Erro	Data e nome de quem corrigiu
1	Pronto para testar novamente	Alecsander T.	Alta	O aplicativo valida a entrada de dados, (precisa estar com no minimo 8 caracteres). Foi informado um erro "Não foi possivel realizar o registro. Tente novamente mais tarde"	Alecsander Torquato. 17/06/2020
2	Pronto para testar novamente	Lucas M.	Alta	Todas as guia correspondem ao resultado esperado. Somente a guia dos 'planos' nao possuem elementos.	Lucas Monteiro. 20/06/2020
3	Pronto para testar novamente	Lucas M.	Alta	O cadastro pode ser realizado, porem o acesso a conta não persiste.	Lucas Monteiro. 23/06/2020

Tabela 2.4 – Relatório de Resumo de Teste

Relatório Resumo de Teste	
Nome do Projeto:	Sistema de Monitoração Verde ao Cubo
Data início teste:	15/06/2020
Data fim teste:	30/06/2020
Descrição Teste	
Com a criação dos casos de uso e caso de teste, facilitaram a detecção e a correção dos erros. O relatório de incidência de teste demonstrou de forma clara a correção do programador e o momento certo para o retorno para equipe de teste, na qual foi realizado o teste novamente.	
Pessoas Envolvidas	
Dennys M. Braghirolli, Silas Chagas., Alecsander Torquato, Lucas Monteiro.	

Número do Teste	
Casos de teste criados antes do teste	10
Casos de teste criados durante o teste	0
Casos de testes executados	17
Casos de teste com sucesso	12
Casos de teste com erros	5
Casos de testes enviados para correção	3
Percentual	
Casos de teste executados	100%
Casos de testes executados com sucesso	70,5%
Casos de testes com incidência de erro	29,5%
Casos de testes corrigidos pelo desenvolvedor	60%

Conclusão

No momento da confecção de um software para atendimento da necessidade de algum cliente ou para alguma solução que está sendo desenvolvida e utilizada existem muitas metodologias que podem ser adotadas, para uma construção bem feita, e que atenda da melhor maneira a solicitação.

Durante o desenvolvimento de um sistema podemos executar testes para avaliar o seu funcionamento, e se os seus requisitos estão sendo respeitados, para tal são

adotadas metodologias que visam avaliar diversos pontos de um sistema em desenvolvimento. Neste documento foi utilizada a metodologia IEEE 829.

Essa metodologia propôs a geração de quatro documentos fáceis de implementar. O documento está organizado da seguinte maneira: um plano de Testes no começo com informações inerentes ao projeto testado e um no final para ser um relatório de resumo. Esses são alterados uma única vez mas os outros Casos de Testes e Relatório de Incidentes precisam ser alterados a cada realização e correção de teste.

Essa metodologia ajuda bastante na objetividade da execução dos testes, e podem ser usadas para referências pelas pessoas responsáveis pelo teste, e o relatório de incidentes podem ser muito úteis para programadores.

Neste projeto foram testados as partes 'Web' e 'Mobile' do sistema. Na parte Mobile foram verificados, em um primeiro instante que as partes de funcionamento de probes responderam com êxito às solicitações dos testes aplicados. A parte de Registro de Usuário apresentou falhas. Na seção Web todos os Links e botões correspondem às solicitações dos testes aplicados, a parte de registro de usuário apresentou falha.

O teste de software permite que sejam observadas falhas e erros durante a fabricação destes, e não somente no final do projeto, pois são realizados em consonância com o desenvolvimento. Desse modo o tempo gasto para refazer pequenas partes durante o projeto é menor do que fazer uma grande parte no final.

Os resultados que os testes proporcionam à implementação são satisfatórios, existe uma tendência com o aumento da tecnologia de que os testes sejam cada vez mais utilizados.

5. Discussão e Conclusão

Existem muitos formatos de relatórios finais, mas o seu conteúdo pode contemplar o sucesso geral do projeto, a organização do projeto, as técnicas utilizadas para obtenção de resultados, pontos fortes e fracos do projeto e as recomendações da equipe de projeto.

SUCESSO GERAL DO PROJETO: Segundo a APF, uma das formas de medir o sucesso geral do projeto pode é avaliar se o produto e o seu processo de produção

estão de acordo com as métricas de sucesso consideradas na descrição geral do projeto, encontrada no Apêndice A deste trabalho.

De acordo com as métricas, o projeto foi concretizado no tempo máximo permitido pelo cronograma e pode ser considerado com bem sucedido, porém, alguns ciclos do projeto se mostraram curtos demais para a realização de algumas atividades de implementação. Desta forma, houve casos de entregas atrasadas, que acabaram afetando a carga de atividades dos ciclos seguintes. O que de certa forma é esperado pela metodologia da APF, uma vez que o planejamento detalhado do projeto é feito de forma iterativa.

Estes eventuais atrasos impossibilitaram a validação de uma das métricas, a que define que as sondas devessem ser capazes de funcionar por sete dias completos não pode ser medida, porque nunca foi testada esta condição no protótipo da sonda.

Quanto a métrica que permite um usuário visualizar uma série de dados em qualquer intervalo temporal, esta não pôde ter sido verificada por ser uma métrica pouco específica. Como o processo de desenvolvimento foi interativo e gradual, ao implementar os recursos de visualização de dados, a equipe de desenvolvimento entrou em acordo com o cliente para que os usuários possam visualizar os dados do último mês, semana e dia. Sob esta nova métrica, o projeto foi considerado como bem sucedido.

ORGANIZAÇÃO DO PROJETO: De modo geral, o projeto foi bem organizado, entretanto, assim como mencionado anteriormente, alguns ciclos foram curtos para determinadas atividades. Sendo assim, em projetos futuros os ciclos poderão ser mais longos. As reuniões de final de ciclo auxiliaram a interação entre o cliente e equipe de desenvolvimento. Com entregas semanais, foi possível adequar o processo de implementação de forma dinâmica, sempre com o acompanhamento do cliente. Outros detalhes importantes foi a cooperação exclusivamente remotas da equipe de desenvolvimento e do cliente.

TÉCNICAS PARA A OBTENÇÃO DE RESULTADOS: Como o tempo máximo do projeto foi de dois meses, foi priorizado logo no início do primeiro ciclo do projeto, definir claramente o objetivo do projeto e que este estivesse em comum acordo entre o cliente e a equipe de desenvolvimento. Uma medida adota pela equipe de

desenvolvimento foi como repassar tarefas para seus integrantes com base em sua experiência em uma determinada tecnologia.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMS, P. **Crop nutrition in hydroponics**. In: *Symposium on Soil and Soilless Media under Protected Cultivation in Mild Winter Climates* 323. p. 289-306, 1993.

BINANCE ACADEMY. **Encriptação Simétrica e Assimétrica**. Disponível em: <https://www.binance.vision/pt/security/symmetric-vs-asymmetric-encryption>. Acesso em 19 de out. 2019.

BINANCE ACADEMY. **O que é Criptografia de Chave Simétrica?**. Disponível em: <https://www.binance.vision/pt/security/symmetric-vs-asymmetric-encryption>. Acesso em 19 de out. 2019.

BIZ FLUENT. **What Are the Benefits of a Flow Chart?**. Disponível em: <https://bizfluent.com/list-6753493-benefits-flow-chart-.html>. Acesso em 29 de junho de 2020.

BLOG DA CONSULTORIA TECNICA. **Utilizando a engine EJS para aplicações em NodeJS**. Disponível em: <https://tsdn.tecnospeed.com.br/blog-da-consultoria-tecnica-tecnospeed/post/utilizando-a-engine-ejs-para-aplicacoes-em-nodejs>. Acesso em 28 de junho de 2020.

BLOGREVELO. **Qual é a diferença entre CSS e SASS?** Disponível em: <https://blog.revelo.com.br/diferenca-entra-css-e-sass/>. Acesso em 27 de junho de 2020.

BOYKIEMACKAY. *Adaptive Project Framework Managing Complexity in the Face of Uncertainty*. Disponível em: <http://www.boykiemackay.com/wp-content/uploads/2013/09/Robert-K.-Wysocki-Adaptive-Project-Framework-eBook-PDF.pdf>. Acesso em 26 de abril 2020.

CAMBRIDGE CONSULTANTS. **Vertical farming: from Babylon to New York**. Disponível em: <https://www.cambridgeconsultants.com/insights/opinion/vertical-farming-babylon-new-york>. Acesso em junho de 2020.

CANALTECH. **O que é e como funciona a linguagem JavaScript?** Disponível em: <https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>. Acesso em 28 de junho de 2020.

CARAZATO, N. A. **Gaia Protótipo: Um Modelo de Prototipação para Processos de Desenvolvimento de Software**. 48 p.

CIRCUIT BASICS. **How to set up the DHT11 humidity sensor on the Raspberry Pi**. Disponível em: <https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/>. Acesso em maio de 2020.

CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. CMMI – **Guidelines for Process Integration and Product Improvement**, Addison-Wesley, 2003.

COSTA, P. C.; DIDONE, E. B.; SESSO, T. M.; CAÑIZARES, K. A. L.; GOTO, R. **Condutividade elétrica da solução nutritiva e produção de alface em hidroponia**. Scientia Agricola, v. 58, n. 3, p. 595-597, 2001.

RAGGETT, Dave et al. **Raggett on HTML 4**. Addison-Wesley Longman Publishing Co., Inc., 1998.

DEVELOPER ANDROID. **Conheça o Android Studio**. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em 27 de junho de 2020.

DEVPLENO. Nodemon: **Monitore alterações e reinicie automaticamente seus scripts**. Disponível em: <https://devpleno.com/hands-on-nodemon/>. Acesso em 28 de junho de 2020.

EASY GROW. **Outdoor Hydroponics Farm**. Disponível em <https://www.easy-grow.co.uk/home/outdoor-hydroponics-farm-2/>. Acesso novembro de 2019.

ELETROPEÇAS **Sensor Fotoresistor LDR 10mm**. Disponível em <https://www.eletronics.com/Produto/sensor-fotoresistor-ldr-10mm>. Acesso em maio de 2020.

ELECTRONIC NOTES. **Light Dependent Resistor LDR: Photoresistor**. Disponível em: https://www.electronic-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php. Acesso em maio de 2020.

EXPRESSJS. **Gerador de aplicativos do Express**. Disponível em: <https://expressjs.com/pt-br/starter/generator.html>. Acesso em 28 de junho de 2020.

FERNANDES, M. B.; COSTA, B. A.; LEMOS, J. M. **Hydroponic Greenhouse Crop Optimization**. In: *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*. IEEE, 2018. p. 270-275, 2018.

FREVA. **Photo-resistor light sensor on Raspberry Pi**. Disponível em: <https://www.freva.com/2019/06/12/light-sensor-on-raspberry-pi/>. Acesso em maio de 2020.

GREEN AND VIBRANT. **Systems**. Disponível em <https://www.greenandvibrant.com/hydroponic-systems>. Acesso em novembro de 2019.

GENERAL HYDROPONICS. **BASIC HYDROPONICS**. Disponível em: <https://generallydroponics.com/basic-hydroponics>. Acesso em nov. 2019.

GITHUB. **An implementation of JSON Web Tokens**. Disponível em <https://github.com/auth0/node-jsonwebtoken>. Acesso em 27 de outubro de 2019.

GP SOLUTIONS. **VERTICAL FARMING SYSTEMS FOR INDOOR ORGANIC FARMING**. Disponível em <https://www.growpodsolutions.com/>. Acesso em maio 2020.

GREEN AND VIBRANT. **Hydroponic Systems - Different Types and How They Work**. Disponível em: <https://www.greenandvibrant.com/hydroponic-systems>. Acesso em nov. 2019.

GRIFFITHS, Mark. **The Design and Implementation of a Hydroponics Control System**. Dissertação de Mestrado - Oulu University of Applied Sciences, Oulu 2014.

GROWLINK. **YOUR FARM SIMPLIFIED**. Disponível em: <https://growlink.com/>. Acesso em junho de 2020.

ICS.UCI. **Architectural Styles and the Design of Network-based Software Architectures**. Disponível em: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf. Acesso em 27 de outubro de 2019.

ICS.UCI. ***Life beyond Distributed Transactions: an Apostate's Opinion.*** Disponível em: <https://www.ics.uci.edu/~cs223/papers/cidr07p15.pdf>. Acesso em 27 de outubro de 2019.

INFORMIT. ***Introduction to the Adaptive Project Framework.*** Disponível em: <https://www.informit.com/articles/article.aspx?p=1554968>. Acesso em 26 de abril 2020.

IMASTER. ***Autenticação JSON Web Token (JWT) em Node.js.*** Disponível em <https://imasters.com.br/back-end/autenticacao-json-web-token-jwt-em-node-js>. Acesso em 27 de outubro de 2019.

IT-SWARM.DEV. ***Qual a diferença entre o SCSS e o Sass?.*** Disponível em: <https://www.it-swarm.dev/pt/css/qual-diferenca-entre-o-scss-e-o-sass/971287871/>. Acesso em 27 de junho de 2020.

JWT. Introduction to JSON Web Tokens. Disponível em: <https://jwt.io>. Acesso em: 19 de out. 2019

LUCIDCHART. ***What Is Adaptive Project Framework.*** Disponível em: <https://www.lucidchart.com/blog/what-is-adaptive-project-framework>. Acesso em 24 de abril 2020.

LUCIDCHART. ***What is a Flowchart.*** Disponível em: <https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial>. Acesso em 29 de junho de 2020.

LUZ, G. L. ***Frequência de irrigação no cultivo hidropônico da alface.*** Dissertação de Mestrado - Universidade Federal de Santa Maria, Santa Maria, 2008.

MDNWEBDOCS. ***Introdução Express/Node.*** Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introdu%C3%A7%C3%A3o. Acesso em 28 de junho de 2020

MEDIUM. Express ***Generator — salvando seu início.*** Disponível em: <https://medium.com/@tofanelli/express-generator-salvando-seu-in%C3%ADcio-8fe8b11dfdb2>. Acesso em 28 de junho de 2020

MOUSER ELETRONICS. ***DHT11 Technical Data Sheet***. Disponível em: https://www.mouser.com/datasheet/2/7_58/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf. Acesso em maio de 2020.

MYERS, G. J.; SANDLER, C.; BADGETT, T. ***The art of software testing***. John Wiley & Sons, 2011.

NETO, E. B.; BARRETO, L. P. **As técnicas de hidroponia**. Anais da Academia Pernambucana de Ciência Agronômica, v. 8, p. 107-137, 2012.

NODEBR. **O que é Node.js?**. Disponível em: <http://nodebr.com/o-que-e-node-js/>. Acesso em 28 de junho de 2020

NODEMON. nodemon reload, automatically. Disponível em: <https://nodemon.io/>. Acesso em 28 de junho de 2020

NYTIMES. Como a Grande Indústria Viciou o Brasil em Junk Food. Disponível em: <https://www.nytimes.com/2017/09/16/health/brasil-junk-food.html>. Acesso em 19 de out. 2019

OPUSSOFTWARE. **Node.js – O que é, como funciona e quais as vantagens**. Disponível em: <https://www.opus-software.com.br/node-js/>. Acesso em 28 de junho de 2020

OSRAM. **Horticulture Lighting**. Disponível em: <https://www.osram.pt/os/applications/horticulture-lighting/index.jsp>. Acesso nov. 2019.

ORSINI, F.; Kahane, R.; Nono-Womdim, R.; Gianquinto, G. **Urban agriculture in the developing world: a review**. *Agronomy for sustainable development*, v. 33, n. 4, p. 695-720, 2013.

PAETSCH, F.; EBERLEIN, A.; MAURER, F. **Requirements engineering and agile software development**. In: **WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises**, IEEE, p. 308-313, 2003.

PPLWARE. **Tutorial Node.js: Como usar o engine EJS no seu projeto.** Disponível em: <https://pplware.sapo.pt/tutoriais/tutorial-node-js-como-usar-o-engine-ejs/>. Acesso em 28 de junho de 2020

PRESMAN. Introdução ao UML. 2011.

PRESSMAN, R.; MAXIM, B. Engenharia de Software-8a Edição. [S.I.]: McGraw Hill Brasil, 2016.

RASPBERRY PI FOUNDATION. **Raspberry Pi 3 Model B.** Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em maio de 2020.

SAATH, K. C. de O.; FACHINELLO, A. L. **Crescimento da demanda mundial de alimentos e restrições do fator terra no Brasil.** Revista de Economia e Sociologia Rural, v. 56, n. 2, p. 195-212, 2018.

SARAVANAN AL. **RASPBERRY PI – All About controlling Relay Boards for Home Automation.** Disponível em: <https://www.youtube.com/watch?v=My1BDB1ei0E>. Acesso em maio de 2020.

SARDARE, M. D.; ADMANE, Shraddha V. **A review on plant without soil-hydroponics.** *International Journal of Research in Engineering and Technology*, v. 2, n. 03, p. 299-304, 2013.

SIDDIQ, A.; TARIQ, M. O.; ZEHRA, A.; MALIK, S. **ACHPA: A sensor based system for automatic environmental control in hydroponics.** Food Science and Technology, n. AHEAD, 2019.

SIMONMCCADE. **Native Apps Vs Web Apps: A Product Designer's Perspective.** Disponível em: <https://www.simonmccade.com/blog/native-apps-vs-web-apps-a-designers-perspective>. Acesso em 27 de junho de 2020.

SCRUMMASTER. **The Beginner's Guide to Project Management Methodologies.** Disponível em: http://www.scrummaster.dk/lib/AgileLeanLibrary/Topics/_ProjectManagement_General/the_beginners_guide_to_project_management_methodologies.pdf. Acesso em 24 de abril 2020.

SPECTRE.CSS. A Lightweight, Responsive and Modern CSS Framework. Disponível em: <https://picturepan2.github.io/spectre/>. Acesso em 28 de junho de 2020.

SOMMERVILLE, I. **Software engineering**. Addison-Wesley/Pearson, 2010.

SUNFOUNDER. **Light Dependent Resistor LDR: Photoresistor**. Disponível em: https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php. Acesso em maio de 2020.

TECMUNDO. **O que é CSS?**. Disponível em: <https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm>. Acesso em 28 de junho de 2020.

TINHKTHYME. Adaptive Project Framework Methodology. Disponível em: <https://thinkthyme.com/project-management/adaptive-project-framework-methodology>. Acesso em 26 de abril 2020.

THIENGO. **MVP Android**. Disponível em: <https://www.thiengo.com.br/mvp-android>. Acesso em 27 de junho 2020.

VENTUREBEAT. **Google Play hits 8 billion new app installs per month, gets new developer features and bug bounty program**. Disponível em: <https://venturebeat.com/2017/10/19/google-play-hits-8-billion-new-app-installs-per-month-and-gets-new-developer-features/>. Acesso em 27 de junho de 2020.

VIDA DE SILÍCIO. **DHT11 - Sensor de Umidade e Temperatura**. Disponível em: <https://www.vidadesilicio.com.br/dht11-sensor-umidade-e-temperatura>. Acesso em maio de 2020.

VOGELLA. **Using Retrofit 2.x as REST client – Tutorial**. Disponível em: <https://www.vogella.com/tutorials/Retrofit/article.html>. Acesso em 27 de junho de 2020.

YEH, N.; CHUNG, J. P. **High-brightness LEDs—Energy efficient lighting sources and their potential in indoor plant cultivation**. *Renewable and Sustainable Energy Reviews*, v. 13, n. 8, p. 2175-2180, 2009.

WAZLAWICK, Raul. **Engenharia de software: conceitos e práticas**. Elsevier Editora Ltda., 2019.

WEBDESIGN.TUTSPLUS. **What is Figma?**. Disponível em: <https://webdesign.tutsplus.com/articles/what-is-figma--cms-32272>. Acesso em 27 de junho de 2020.

WYSOCKI, Robert K. ***Adaptive project framework: managing complexity in the face of uncertainty***. Addison-Wesley Professional, 2010.

APÊNDICE A – DESCRIÇÃO GERAL DO PROJETO

Tabela 4 - Tabela de descrição geral do projeto

DESCRIÇÃO GERAL DO PROJETO	Nome do projeto Verde Ao Cubo	Nº do projeto 01	Gerente(s) do projeto Caio Aragon Lucas Monteiro
Problema/Oportunidade O aumento da coletadas de informações em sistemas de cultivos hidropônicos beneficiam a produtividade destes sistemas, pois, permite que os cultivadores tenham a capacidade de melhor diagnosticar problemas e de melhor compreender seus cultivos. Geralmente, os produtores se limitam às informações coletadas por um único sensor, realizadas em diferentes pontos do dia ou quando necessário.			
Objetivo Geral Projetar e desenvolver um software para controlar dispositivos e monitorar variáveis ambientais em sistemas de cultivo hidropônico.			
Objetivos Específicos <ol style="list-style-type: none">1. Projetar e desenvolver um aplicativo mobile para o sistema.2. Desenvolver sistema de infraestrutura em servidores para suporte da aplicação.3. Adquirir materiais e desenvolver protótipos de <i>probes</i>*.4. Implementar uma interface de sistema para visualizar de dados de monitoração.5. Implementar uma interface de sistema para ajuste de <i>probes</i>*.6. Projetar e desenvolver um site para divulgação do sistema.7. Implementar interface de administração do sistema.8. Desenvolver e executar9. testes para o sistema (aplicativo, servidores, e <i>probes</i>*). * dispositivos de monitoração e controle			
Critérios de sucesso			

1. Concretização do desenvolvimento do projeto no tempo máximo definido pelo cronograma.
2. Projeto aprovado por orientadores do corpo docente.
3. Quando energizadas, as probes devem ser capazes de funcionar sem interrupções por pelo menos sete dias completos.
4. O aplicativo deve ser capaz de exibir as informações coletadas nos cultivos em qualquer intervalo temporal definido pelo cliente.
5. No mínimo 60% de disponibilidade dos sistemas de suporte (servidores) após a implantação do sistema.

Suposições, Riscos, Obstáculos

1. Sensores para monitorar parâmetros de qualidade da água (oxigênio dissolvido, pH e condutividade elétrica) e de gás carbônico podem ser caros demais para serem incluídos inicialmente nas probes.
2. Falta de esclarecimento quanto ao escopo do projeto podem prolongar atividades mais do que o esperado.
3. Indisponibilidade mútua ou conflito de agendas dos integrantes podem impossibilitar o cumprimento das tarefas no tempo definido em cronograma.
4. Eventuais atrasos no desenvolvimento do sistema que podem inviabilizar a entrega do projeto no tempo definido.
5. Artefatos de desenvolvimento não aprovados pela orientação docente.
6. A equipe tem o arcabouço necessário para desenvolver o projeto.

Preparado por
Caio Aragon
Lucas Monteiro

Data
20/04/2020

Aprovado por
Caio Aragon

Data
20/04/2020

Desenvolvido por Caio Cabezas e Lucas Monteiro

APÊNDICE B – RBS DO PROJETO

Requisitos RBS - Verde Ao Cubo

1. Mobile

1.1. Conta

1.1.1. Confirmar e-mail

1.1.2. Manter Conta

1.1.3. Permitir Acesso

1.1.2. Interface de Monitoração

1.1.2.1. Visualizar Dados

1.1.2.2. Selecionar Probes

1.1.2.3. Configurar Probe

1.1.2.3.1. Probe de Monitoração

.1. Acionamento

.2. Frequência de Amostragem

1.1.2.3.2. Probe de Controle

.1. Acionamento

.2. Modo de Operação

.1. Contínuo

.2. Intermitente

.1. Período de Atividade

.2. Período de Inatividade

.3. Adicionar Probe

2. Probe

2.1. Operação

2.1.1. Conectado à Rede

2.1.2. Configurações

2.1.2.1. Verificar Atualizações

2.1.2.2. Importar Configurações

2.1.2.3. Aplicar Configurações

2.1.3. Manter em Operação

2.2. Transmitir Dados

3. Web

3.1. Conta (idem req. 1.1.)

3.2. Interface Administradores

3.2.1. Visualizar relatórios operacionais

3.2.2. Listar usuários

3.2.3. Bloquear usuários

3.2.4. Listar probes

3.2.5. Cadastrar probes.

3.3. Interface de Monitoração e Controle

3.3.1. Visualizar dados de monitoração

3.3.2. Configurar operação da Probe

APÊNDICE C – WBS DO PROJETO

1. Criar app mobile
 - 1.1. Desenvolver mock-ups
 - 1.1.1. Confeccionar mockup da Tela de Login
 - 1.1.2. Confeccionar mockup da Tela de Cadastro
 - 1.1.3. Confeccionar mockup da Tela de Confirmação de e-mail
 - 1.1.4. Confeccionar mockup da Tela de Gerenciamento de conta
 - 1.1.5. Confeccionar mockup da Interface Gráficos
 - 1.1.6. Confeccionar mockup da Interface Lista de Probes
 - 1.1.7. Confeccionar mockup do Painel de Configuração
 - 1.2. Implementar registro de clientes
 - 1.2.1. Confirmação por e-mail
 - ~~1.2.1.1. Desenvolver Back-end de Confirmação por e-mail~~
 - 1.2.1.2. Desenvolver Interface de Confirmação por e-mail
 - 1.3. Implementar interface de acesso
 - 1.3.1. Autenticação
 - ~~1.3.1.1. Desenvolver Back-end de Autenticação~~
 - 1.3.1.2. Desenvolver Interface de Login
 - 1.3.1.2. Desenvolver Interface de Cadastro
 - 1.4. Implementar interface de probes
 - 1.4.1. Painel de Gráficos
 - 1.4.1.1. Desenvolver front-end da interface Gráficos
 - 1.4.1.2. Desenvolver back-end da interface Gráficos
 - 1.4.2. Lista de probes
 - 1.4.2.1. Desenvolver front-end da interface Lista de Probes
 - 1.4.2.2. Desenvolver back-end da interface Lista de Probes
 - 1.4.3. Painel de Configuração
 - 1.4.3.1. Probe de Monitoração
 - 1.4.3.1.1. Desenvolver front-end de Monitoração
 - 1.4.3.1.2. Desenvolver back-end de Monitoração
 - 1.4.3.2. Probe de Controle
 - 1.4.3.2.1. Desenvolver front-end de Controle
 - 1.4.3.2.2. Desenvolver back-end de Controle
 - 1.5. Implementar software para servidores
 - 1.5.1. API (REST) de domínio
 - 1.5.1.1. Modelar entidades para recursos
 - 1.5.1.2. Definir rotas de acesso
 - 1.5.1.3. Implementar rotas
 - 1.5.1.4. Realizar deploy
 - 1.5.1.5. Otimizar consumo de recursos
 - 1.5.2. API de autenticação
 - 1.5.2.1. Definir rotas de acesso
 - 1.5.2.2. Implementar rotas
 - 1.5.2.3. Realizar deploy
 - 1.6. Desenvolver e executar testes
 - 1.6.1. Metodologias de teste
 - 1.6.1.1. Pesquisar metodologias para teste de software
 - 1.6.2. Planos de teste
 - 1.6.2.1. Definir e documentar planos de teste de software
 - 1.6.2.2. Confeccionar Casos de teste

- 1.6.3. Execução de Testes
 - 1.6.3.1. Realizar e documentar os testes planejados
- 1.7. Documentar o desenvolvimento do aplicativo
 - 1.7.1. Prototipação
 - 1.7.1.1. Descrever os protótipos do aplicativo
 - 1.7.1.2. Descrever os protótipos da api
 - 1.7.2. Tecnologias aplicadas
 - 1.7.2.1. Descrever as tecnologias utilizadas no aplicativo
 - 1.7.2.2. Descrever as tecnologias utilizadas na api
 - 1.7.3. Implementação
 - 1.7.3.1. Descrever a implementação do aplicativo
 - 1.7.3.2. Descrever a implementação da api
 - 1.7.4. Implantação
 - 1.7.4.1. Descrever a implantação do aplicativo
 - 1.7.4.2. Descrever a implantação da api

2. Criar site web

- 2.1. Desenvolver mock-ups
 - 2.1.1. Confeccionar mockup da Tela de Login
 - 2.1.2. Confeccionar mockup da Tela de Cadastro
 - 2.1.3. Confeccionar mockup da Tela de Confirmação de e-mail
 - 2.1.4. Confeccionar mockup da Tela Inicial
 - 2.1.5. Confeccionar mockup da Tela de Gerenciamento de conta
 - 2.1.6. Confeccionar mockup da Tela de Relatórios Administrativo (CRUD de Contas, Probes, Snapshots por Probe)
- ~~2.2. Implementar registro de administradores~~
 - ~~2.2.1. Confirmação por e-mail~~
- 2.3. Implementar interface de acesso
 - 2.3.1. Autenticação
 - 2.3.1.1. Implementar Tela de Login
 - 2.3.1.2. Implementar Tela de Cadastro
 - 2.3.1.3. Implementar Tela de Confirmação de e-mail
 - 2.3.2. Gerenciamento
 - 2.3.2.1. Implementar Tela de Gerenciamento de conta
 - 2.3.3. Navegação
 - 2.3.3.1. Implementar Tela Inicial
 - 2.3.4. Back-end para client web
 - 2.3.4.1. Implementar back-end
- 2.4. Implementar interface administrativa
 - 2.4.1. Relatórios
 - 2.4.1.1. Implementar Tela de Relatórios Administrativos
 - 2.4.2. Gerenciamento de Clientes
 - 2.4.2.1. Implementar Tela de Gerenciamento de Clientes
 - 2.4.3. Gerenciamento de Probes
 - 2.4.3.1. Implementar Tela de Gerenciamento de Probes
 - 2.4.4. Confirmação por e-mail
 - 2.4.4.1. Implementar Tela de Confirmação de E-mail
- 2.5. Desenvolver e executar testes
 - 2.5.1. Metodologias de teste
 - 2.5.1.1. Pesquisar metodologias para teste de software
 - 2.5.2. Planos de teste
 - 2.5.2.1. Definir e documentar planos de teste de software

- 2.5.2.2. Confeccionar Casos de teste
 - 2.5.3. Execução de Testes
 - 2.5.3.1. Realizar e documentar os testes planejados
 - 2.6. Desenvolver a identidade de marca
 - 2.6.1. Logotipo
 - 2.6.1.1. Confeccionar logotipo da marca
 - 2.6.2. Paleta de cores
 - 2.6.2.1. Definir paleta de cores
 - 2.6.3. Tipografia
 - 2.6.3.1. Definir tipografia
 - 2.7. Implementar monitoração e controle
 - 2.7.1. Interface de Monitoração
 - 2.7.1.1. Desenvolver mockups de interface
 - 2.7.1.2. Implementar front-end
 - 2.7.1.3. Implementar back-end
 - 2.7.2. Interface de Configuração
 - 2.7.1.1. Desenvolver mockups de interface
 - 2.7.1.2. Implementar front-end
 - 2.7.1.3. Implementar back-end
 - 2.8. Documentar o desenvolvimento do website
 - 2.8.1. Prototipação
 - 2.8.1.1. Descrever os protótipos do software
 - 2.8.2. Tecnologias aplicadas
 - 2.8.1.2. Descrever as tecnologias utilizadas
 - 2.8.3. Implementação
 - 2.8.1.3. Descrever o processo de implementação do software
 - 2.8.4. Implantação
 - 2.8.1.4. Descrever o processo de implantação do software
- 3. Desenvolver Protótipo de Probe
 - 3.1. Confeccionar protótipo
 - 3.1.1. Raspberry
 - 3.1.1.1. Pesquisar materiais
 - 3.1.1.2. Adquirir materiais
 - 3.1.1.3. Confeccionar um mockup do módulo de sensores
 - 3.1.1.4. Confeccionar um mockup do módulo de controle
 - 3.1.2. Sensores
 - 3.1.2.1. Confeccionar protótipo de *hardware* do módulo de sensores
 - 3.1.2.2. Confeccionar protótipo de *software* do módulo de sensores
 - 3.1.3. Controle
 - 3.1.3.1. Confeccionar protótipo de *hardware* do módulo de controle
 - 3.1.3.2. Confeccionar protótipo de *software* do módulo de controle
 - 3.2. Implementar transmissão de dados
 - 3.2.1. Atualizações
 - 3.2.1.1. Desenvolver rotina de verificação de conectividade
 - 3.2.1.2. Desenvolver rotina de verificação de atualizações de sistema 21/05 - 24/05
 - 3.2.1.3. Desenvolver rotina de verificação de atualizações de usuário
 - 3.2.2. Configurações
 - 3.2.2.1. Desenvolver rotina para importar configurações

- 3.2.2.2. Desenvolver rotina para aplicar configurações
- 3.3. Implantar operação automática
 - 3.3.1. Dispositivos
 - 3.3.1.1. Desenvolver rotina de operação do módulo de controle
 - 3.3.1.2. Desenvolver rotina final de operação do módulo de controle 21/05 - 24/05
 - 3.3.2. Sensores
 - 3.3.2.1. Desenvolver rotina de operação do módulo de sensores
 - 3.3.1.2. Desenvolver rotina final de operação do módulo de sensores 21/05 - 24/05
- 3.4. Planejar e executar teste
 - 3.4.1. Metodologias de teste
 - 3.4.1.1. Pesquisar metodologias para teste de software
 - 3.4.2. Planos de teste
 - 3.4.2.1. Definir e documentar planos de teste de software
 - 3.4.2.2. Confeccionar Casos de teste
 - 3.4.3. Execução de Testes
 - 3.4.3.1. Realizar e documentar os testes planejados
- 3.5. Refatoração e otimização do software da Probe.
 - 3.5.1. Referências
 - 3.5.1.1. Buscar referências de modularização
 - 3.5.2. Refatoração
 - 3.5.2.1. Refatorar software
- 3.6. Documentar o desenvolvimento da probe
 - 3.6.1. Prototipação
 - 3.6.1.1. Descrever os protótipos do software
 - 3.6.2. Tecnologias aplicadas
 - 3.6.1.2. Descrever as tecnologias utilizadas
 - 3.6.3. Implementação
 - 3.6.1.3. Descrever o processo de implementação do software
 - 3.6.4. Implantação
 - 3.6.1.4. Descrever o processo de implantação do software

APÊNDICE D – PRIORIZAÇÃO DAS VARIÁVEIS DO TRIÂNGULO DE ESCOPO

Priorização das variáveis do Triângulo de Escopo:

Tabela 5 - Tabela de priorização das variáveis do Triângulo de Escopo

		Prioridade				
		Crítica (1)	(2)	(3)	(4)	Flexível (5)
Variável	Escopo				X	
	Qualidade			X		
	Tempo	X				
	Custo					X
	Disponibilidade de Recursos		X			

Desenvolvido por Caio Cabesas e Lucas Monteiro

APÊNDICE E – PLANO DE PROJETO EM ALTO NÍVEL

Plano de projeto - em alto nível

São definidos nesse documento, conforme especificações do *Adaptive Project Framework* (APF), variáveis necessárias para o plano geral de projeto.

Tempo: Dois meses

Orçamento: R\$ 0,00

Duração de ciclo: Sete dias (semanal, de segunda à sábado), durante três ciclos,

Tabela 6 - Tabela de plano de projeto

Ciclo	Início	Fim
Ciclo 1	27/04	03/05
Ciclo 2	04/05	10/05
Ciclo 3	11/05	17/05
Ciclo 4	18/05	31/05
Ciclo 5	01/06	14/06
Ciclo 6	15/06	21/06
Ciclo 7	22/06	29/06

Desenvolvido por Caio Cabezas e Lucas Monteiro

OBJETIVOS: