



Trabalho de Conclusão de Curso

Uma ferramenta educativa para o cultivo de plantas utilizando Internet das Coisas

Clóvis Gabriel Melo do Nascimento

cgm@ic.ufal.br

Orientadora:

Prof^a. Roberta Vilhena Vieira Lopes

Maceió, Agosto de 2019

Clóvis Gabriel Melo do Nascimento

Uma ferramenta educativa para o cultivo de plantas utilizando Internet das Coisas

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientadora:

Prof^a. Roberta Vilhena Vieira Lopes

Maceió, Agosto de 2019

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Taciana Sousa dos Santos – CRB-4 – 2062

N244f Nascimento, Clóvis Gabriel Melo do.

Uma ferramenta educativa para o cultivo de plantas utilizando internet das coisas / Clóvis Gabriel Melo do Nascimento. – 2019.

47 f. : il., figs. color.

Orientadora: Roberta Vilhena Vieira Lopes.

Monografia (Trabalho de Conclusão de Curso em Engenharia da Computação) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2020.

Bibliografia: f. 43-47.

1. Internet das coisas. 2. Sistemas embarcados (Computadores). 3. Computação em nuvem. 4. Aplicativo mobile. 5. Sustentabilidade. I. Título.

CDU: 004.738.5

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação do Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.

Prof^a. Roberta Vilhena Vieira Lopes - Orientador
Instituto de Computação
Universidade Federal de Alagoas

- Examinador

Maceió, Agosto de 2019

Agradecimentos

Agradeço primeiramente a Deus, e em seguida aos meus tios, Maria Aparecida e Genivaldo Sérgio, por terem investido em minha educação ao longo dos anos, procurando sempre fornecer o melhor em termos de estudo. Também agradeço pelo apoio incondicional quanto ao curso que escolhi, por acreditarem em meu potencial, pelos bons valores passados ao longo da minha vida e por me auxiliarem sempre. Agradeço também aos meus pais, Leda e Laécio; meus irmãos, Victor e Victória; e minhas primas, Maria Clara e Marília, por me apoiarem e torcerem por mim. E também agradeço à minha tia Roberta Alessandra pelo apoio e palavras de incentivo ao longo da jornada, como também por ter me acompanhado à UFAL pela primeira vez para realizar a matrícula no curso. Adicionalmente, gostaria de agradecer à minha amiga Abigail Musa pela companhia em momentos de estudos e por estar sempre ao meu lado nos momentos bons e ruins nessa missão.

Agradeço também à minha orientadora, Roberta Vieira, por ser uma professora que admiro muito e que contribuiu bastante com a minha formação, assim como também com esta monografia. Por fim, gostaria de agradecer a todos os meus amigos, familiares e professores que de alguma forma contribuíram para o meu crescimento profissional e pessoal.

“Nossos fracassos, às vezes, são mais frutíferos do que os êxitos.”

– Henry Fords

“Lembre sempre que a sua vontade de triunfar é mais importante do que qualquer outra coisa.”

– Abraham Lincoln

Resumo

Considerada como uma revolução digital, a Internet das Coisas consiste em conectar objetos do cotidiano à internet. Esse advento permite fazer o monitoramento de qualquer situação através da coleta de dados, viabilizada pelo uso de sensores posicionados em locais estratégicos, e também pode permitir a tomada de decisão, através de atuadores que agirão no ambiente. Neste trabalho, será mostrado o desenvolvimento de um sistema em que é possível monitorar a umidade do solo de uma planta em tempo real através de um aplicativo mobile e configurar para regá-la automática ou manualmente. O intuito deste trabalho é fornecer uma ferramenta educativa e tecnológica para desenvolver tanto em crianças quanto em adultos a percepção de quando uma planta precisa ou não de água, auxiliando indivíduos a entender melhor como funciona o mundo ao seu redor e incentivando a prática da sustentabilidade, levando em consideração a importância do cultivo de plantas no planeta e a economia da água, um recurso escasso já em alguns lugares. Além disso, também se deseja verificar neste trabalho se o sistema desenvolvido mantém a planta em umidade aceitável do solo, realizando também a gestão do recurso água.

Palavras-chave: Internet das Coisas, Sistemas *IoT*, Computação em nuvem, Sistemas embarcados, Aplicativos *mobile*, Sustentabilidade

Abstract

Regarded as a digital revolution, the Internet of Things consists of connecting everyday objects to the internet. This advent makes it possible to monitor any situation through data collection, made possible by the use of sensors positioned in strategic locations, and it can also allow decision making, through actuators that will act in the environment. In this work, it will be shown the development of a system in which it is possible to monitor the soil moisture of a plant in real time through a mobile application and configure to water it automatically or manually. The aim of this work is to provide an educational and technological tool to develop in both children and adults the perception of when a plant needs water or not, helping individuals to better understand how the world around them works and encouraging the practice of sustainability, taking into account the importance of growing plants on the planet and saving water, a scarce resource already in some places. In addition, it is also desired to verify in this work if the developed system keeps the plant in acceptable soil moisture, also managing the water resource.

Keywords: Internet of Things, Cloud Computing, IoT systems, Embedded Systems, Mobile apps, Sustainability

Conteúdo

Lista de Figuras	vii
1 Introdução	1
2 Estado da arte	3
2.1 Projetos de conscientização e gestão do uso de água	3
2.1.1 Soluções convencionais	3
2.1.2 Soluções tecnológicas	4
2.2 O processo de conscientização ambiental	6
3 Elementos <i>IoT</i>	7
3.1 Microcontroladores	7
3.1.1 Arduino	7
3.1.2 Esp8266 NodeMCU	8
3.2 Sensores em <i>IoT</i>	10
3.2.1 Redes de sensores sem fio	10
3.2.2 Sensor de umidade do solo	11
3.2.3 Sensor de temperatura e umidade do ar	12
3.2.4 Sensor de luminosidade	13
3.3 Atuadores	15
3.4 Aplicativos mobile	15
3.5 <i>Cloud Computing</i>	18
4 Metodologia	22
4.1 Módulos do Sistema	22
4.2 Ferramentas Utilizadas	24
4.3 Desenvolvimento do sistema <i>IoT</i>	25
4.3.1 <i>Hardware</i>	26
4.3.2 <i>Software</i> embarcado	27
4.4 Desenvolvimento do aplicativo <i>Android</i>	30
4.5 Comunicação com a nuvem	30
4.5.1 Envio de dados	30
4.5.2 Consulta de dados	31
5 Resultados	34
5.1 O Ambiente	35
5.2 Resultados do monitoramento	35
5.2.1 Avenca	36
5.2.2 Avenca variada	38

6 Conclusão e trabalhos futuros**41****Referências bibliográficas****43**

Lista de Figuras

3.1	Microcontrolador Arduino UNO. [Retirada de https://www.techtudo.com.br/noticias/noticia/2013/10/o-que-e-um-arduino-e-o-que-pode-s.html]	8
3.2	Microcontrolador Esp8266 NodeMCU. [Retirada de https://br.banggood.com/Geekcreit-Doit-NodeMcu-Lua-ESP8266-ESP-12F-WIFI-Development-B.html?cur_warehouse=USA]	9
3.3	O chip foi acoplado em um módulo, chamado ESP12, e em seguida o módulo foi acoplado na placa NodeMCU. [Retirada de (Banggood, 2014)]	9
3.4	Funcionamento de uma rede de sensores sem fio. [Retirada de (RSSF UFRJ, 2018)]	10
3.5	Sensor de umidade do solo. [Retirada de https://www.vidadesilicio.com.br/sensor-umidade-solo-higrometro]	11
3.6	Sensor de umidade de temperatura e umidade do ar DHT11. [Retirada de https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura/]	12
3.7	Gráfico que exhibe a relação Pressão vapor x Temperatura. [Retirada de https://commons.wikimedia.org/wiki/File:Water_vapor_pressure_graph.jpg]	13
3.8	Sensor de Luz BH1750FVI Lux. [Retirada de https://www.filipeflop.com/produto/sensor-de-luz-bh1750fvi-lux/]	14
3.9	As possíveis frequências de luz. O espectro visível está expresso em azul claro. [Retirada de (?)]	14
3.10	Iluminância. [Retirada de (?)]	15
3.11	Bomba D'água Submersiva 3-6V 100L/H. [Retirada de https://www.filipeflop.com/produto/bomba-dagua-submersiva-3-6v-100l-h/]	16
3.12	Lucro médio dos 100 principais apps para dispositivos móveis - um comparativo com a App Store e Google Play. [Retirada de https://www.tecmundo.com.br/mercado/142951-app-store-tem-lucro-65-superior-play-store-htm]	17
3.13	Um comparativo da utilização de sistemas operacionais em smartphones de 2012 até 2016. [Retirada de (LTD, 2017)]	18
3.14	Arquitetura de um serviço de computação em nuvem. O centro da imagem exhibe a nuvem, onde tudo fica armazenado. Observe que a comunicação entre os dispositivos e a nuvem é bidirecional. [Retirada de https://carolinadatarecovery.com/cloudy-with-a-chance-of-data-how-the-clo]	19

3.15 Interface gráfica do <i>Firebase</i> , exibida no navegador. [Retirada de https://carolinadatarecovery.com/cloudy-with-a-chance-of-data-how-the-cto-works/].	21
4.1 Fluxograma do sistema.	23
4.2 Modelo de agente e ambiente de Russel et al.	23
4.3 Pinos do Esp8266 NodeMCU. [Retirada de https://blog.eletrogate.com/nodemcu-esp12-introducao-1/].	24
4.4 Tela principal do <i>Android Studio</i>	25
4.5 Esquemático do circuito elétrico da camada de hardware.	26
4.6 O sistema <i>IoT</i> conectado a uma planta.	26
4.7 O circuito elétrico montado na <i>protoboard</i>	27
4.8 A bomba de água utilizada no sistema.	27
4.9 <i>Arduino IDE</i>	28
4.10 Escala utilizada para calibrar o sensor de umidade do solo.	29
4.11 <i>Arduino IDE</i>	29
4.12 Tela principal do aplicativo <i>mobile</i> , que expressa a porcentagem em tempo real da umidade do solo. (a) mostra esta tela com umidade do solo moderada e (b) mostra a mesma tela com umidade do solo baixa.	31
4.13 (a) Gráfico Umidade do solo (%) x Tempo (h). - (b) Tela de histórico que exhibe instante do acionamento de bomba de água.	32
4.14 Tela para configurar rega manual ou automatizada.	32
4.15 A estrutura do armazenamento de dados no <i>RealTime Database</i> da <i>Firebase</i>	33
4.16 A classe <i>Sensors</i> , utilizada para receber os dados da <i>database</i>	33
5.1 Sistema <i>IoT</i> de rega automatizada instalado em uma plantaavenca.	36
5.2 (a) Resultado do gráfico Umidade do solo x Tempo da plantaavenca em um período de 72h. (b) Histórico de acionamento de bomba de água em plantaavenca.	37
5.3 Sistema <i>IoT</i> de rega automatizada instalado em umaavencavariada.	38
5.4 (a) Resultado do gráfico Umidade do solo x Tempo da plantaavencavariada em um período de 72h. (b) Histórico de acionamento de bomba de água em plantaavencavariada.	39

1

Introdução

O termo Internet das Coisas (ou *Internet of Things - IoT*, do inglês) pode ser definido de diversas modos. [Morgan](#) a define de maneira simples como sendo o conceito de uma rede gigante de dispositivos conectados à Internet, ou ainda coisas interconectados, tais como *smartphones*, lâmpadas, geladeiras, cafeteiras, TVs, dentre outros componentes com botão de liga e desliga. O autor vai além, e diz em seu trabalho que Internet das Coisas também pode conectar pessoas, isto é, o relacionamento em Internet das Coisas ocorre entre pessoas-pessoas, pessoas-coisas e coisas-coisas.

A origem deste termo foi atribuído a Kevin Ashton, citado pela primeira vez em 1999 ([Ashton, 2009](#)). Ao passar dos anos, tornou-se possível aplicar a Internet das Coisas em inúmeras áreas, como prédios inteligentes, automação residencial, *smart-agriculture*, cuidados de saúde, entre outros. A Internet das Coisas visa integrar o mundo físico com o mundo virtual usando a Internet como meio de comunicação e troca de informações ([O. Elijah and Hindia, 2018](#)).

Em ([Manyika, 2015](#)), o impacto econômico mundial de Internet das Coisas é estimado entre \$3.9 até \$11 trilhões de dólares em 2025. No Brasil, o governo brasileiro lançou em 2017 seu plano de ação para Internet das Coisas, que foi destacado um possível impacto econômico de \$50 bilhões para \$ 200 bilhões por ano - cerca de 10% do PIB - das aplicações da tecnologia em quatro áreas: saúde, indústria, rural e urbana. A conclusão é do relatório *IoT Barometer 2017/18*, divulgado pela *Vodafone*.

Uma das áreas que a Internet das Coisas pode ser aplicada é para a educação ambiental, visando a preservação do meio ambiente, uso inteligente de água, dentre outros. No contexto de uso de água, é importante conscientizar as pessoas para economizar este recurso indispensável e insubstituível, evitando o desperdício, pois este é um recurso bastante valioso e cada vez mais escasso em alguns lugares ([Presse, 2015](#)). Dessa forma, uma aplicação *IoT* bastante útil é um sistema de monitoramento de umidade do solo de uma planta, que viabiliza o monitoramento da mesma através de um aplicativo e informa ao usuário quando

ela deve ser regada. Assim, evita-se o desperdício de água e o usuário saberá o momento certo em que a planta deve ser regada. Este sistema auxiliará o usuário na conscientização do uso de água e no cultivo de planta.

Para tanto, esta monografia mostra o desenvolvimento de um sistema que faz uso de Internet das Coisas, conectando um sensor de umidade do solo na região da planta a ser regada - sistema *smart water* - e exibindo o valor da porcentagem de umidade em tempo real em um aplicativo para *smartphones*. O próprio *app* exibe a porcentagem e comunica ao usuário se o nível de umidade está baixo, razoável ou alto - isso evita que a região a ser regada não seja inundada ou desperdice o recurso. Caso o nível esteja baixo, o usuário será orientado em como reequilibrar a umidade do solo naquele momento. Após seguir as orientações, ele verá em poucos momentos no aplicativo se o nível foi normalizado. Contudo, é possível configurar no aplicativo a rega automática da planta, sem necessidade de intervenção humana, sendo suficiente apenas o usuário acompanhar a umidade do solo pelo aplicativo.

O sistema preza também por criar uma experiência de usuário eficaz no contexto de cultivo de planta, auxiliando o mesmo a cultivá-la por meio de uma ferramenta tecnológica e educativa.

Este trabalho está dividido nos seguintes capítulos: Capítulo 2 é discutido o estado da arte, que revela diversos trabalhos e projetos de conscientização ambiental no contexto de economia de água realizados em algumas escolas com um grande número de estudantes e docentes, e o impacto que gerou nos ambientes estudados; Capítulo 3 mostra os elementos necessários para rega automatizada com *IoT*, que são os microcontroladores, os sensores, a *Cloud Computing* e a importância de aplicativos *mobile* para disponibilizar os dados captados; Capítulo 4 define a arquitetura do sistema, as ferramentas utilizadas, e revela as etapas de desenvolvimento do sistema; Capítulo 5 apresenta os resultados; e, finalmente, o Capítulo 6 conclui esta monografia .

2

Estado da arte

Nesta seção serão mostrados alguns estudos realizados no contexto de educação ambiental, conscientização e gerenciamento do uso de água. Dessa forma, foram analisados projetos convencionais e tecnológicos nesse contexto.

2.1 Projetos de conscientização e gestão do uso de água

2.1.1 Soluções convencionais

Analisar a consciência ambiental dos indivíduos quanto ao uso racional de água é importante. No trabalho de (Freitas and Marin, 2015), foi identificado e avaliado como o tema água era abordado em vinte e oito escolas municipais na cidade de São Paulo, isto é, nos projetos especiais destas escolas, nos planos de ensino, nos livros didáticos de ciência e geografia adotados e também como os educadores abordavam o tema. A pesquisa era qualitativa - estudo de caso - e os dados eram obtidos através de questionários e entrevistas estruturadas com os docentes. Observou-se após o estudo que existiam projetos especiais nas escolas que tratavam de educação ambiental de modo geral, não especificamente sobre a água.

Já no trabalho de (N. Melo and Frasson, 2015), realizado em doze escolas públicas no estado de Minas Gerais, na região do Triângulo Mineiro, buscou trabalhar em alunos a conscientização do uso de água na escola, bem como o número grande de vazamentos nas instalações hidráulicas e a utilização de equipamentos sanitários não econômicos. Sendo assim, este estudo teve como objetivo avaliar o consumo de água e a percepção dos usuários quanto ao desperdício. Assim como (Freitas and Marin, 2015), este estudo também considerou realizar questionários e entrevistas aos usuários. Adicionalmente, indagou-se aos entrevistados acerca da quantidade de água que se supunha gastar em um ano anterior, contudo houve inconsistência entre o indicador de consumo e a percepção dos usuários, o que pode ter sido por questão de erro de leitura do hidrômetro (dados incorretos de consumo de água),

vazamentos não detectados, dentre outros motivos. Observou-se também que os usuários das escolas tinham pouco conhecimento sobre como economizar este recurso natural.

É notório, portanto, que a educação ambiental - principalmente no que diz respeito ao uso adequado e econômico da água - deve ser um tema mais abordado, a começar nas escolas e nos primeiros anos do ensino fundamental, para conscientizar as pessoas o mais cedo possível. A consciência do uso da água pode ser abordado em escolas de diversas formas: em livros didáticos de aulas de ciência e geografia; projetos que promovam a economia de água; desenhos animados educativos para as crianças; em feiras de ciência; em seminários; dentre outros.

Um projeto realizado em um colégio estadual do Paraná (J. Tugoz and Brandalise, 2017) viabilizava o aproveitamento das águas da chuva, as quais eram armazenadas em cisternas e utilizadas posteriormente para fins não potáveis, como em equipamentos sanitários. Além de um projeto bastante viável e inteligente, tornando a escola sustentável, também serve como forma de conscientizar seus alunos sobre a importância do uso bem aproveitado do recurso água, assim como incentivar os alunos a práticas de sustentabilidade. Por meio deste projeto, foi possível obter uma redução do gasto de água tratada em até 57,97%.

Outro projeto de educação ambiental em escolas focou em alunos entre oito e treze anos de idade (T. Teixeira and Pereira, 2017), promovendo atividades lúdicas como preservação de água e meio ambiente e separação de recicláveis, com intuito de conscientizar as crianças. O impacto deste projeto foi grande, uma vez que envolveu cerca de seiscentos e vinte alunos de quatro escolas do estado de Minas Gerais. O resultado também foi bastante satisfatório, pois observou-se o aumento da conscientização ambiental em alunos de escolas públicas após a realização das atividades lúdicas e práticas.

Encontraram-se também projetos de educação ambiental recorrendo a tecnologias, como aplicativos móveis e Internet das Coisas. (E. Curry and Derguech, 2018) desenvolveu aplicativos habilitados para Internet das Coisas em diversos ambientes como casa, prédio comercial, escola e universidade, a fim de promover a conscientização, gestão e conservação da água e energia elétrica.

2.1.2 Soluções tecnológicas

Um dos principais desafios na criação de soluções tecnológicas para conscientização ambiental é criar uma experiência eficaz para o usuário com os novos meios digitais, isto é, o design do aplicativo deve atender às preferências do usuário em termos de visualização dos dados e também ser amigável e intuitivo. Adicionalmente, antes da criação das telas do aplicativo se fez necessário estudar a psicologia da persuasão e motivação, para convencer os usuários a mudar seus comportamentos - teorias atitudinais afirmam que influenciar um indivíduo a realizar uma ação positiva que trará um bem para si ou para a humanidade, ele rapidamente a realizará (E. Curry and Derguech, 2018). Para persuadir os usuários a agir,

algumas medidas foram tomadas, tais como a implementação de alertas e notificações geradas pelo aplicativo e solicitações ao usuário para realizar ações físicas no ambiente - no contexto de uso adequado do recurso água, uma ação solicitada pelo aplicativo poderia ser fechar uma válvula.

No trabalho de (Curry et al., 2018), analisaram-se maneiras de persuadir o usuário a utilizar e continuar o uso de aplicativos móveis de sistemas *IoT*, sendo um deles de gerenciamento de água - *smart water*. Observaram-se os aspectos de *design* do aplicativo que mais atraíram os usuários finais, e foi possível concluir que:

- Os usuários têm preferência por aplicativos e visualizações de baixa carga cognitiva;
- Aplicativos complexos com funcionalidade total eram difíceis para que os usuários aprendam e entendam;
- Os usuários queriam aplicativos simples (geralmente de propósito único) em vez de mais elaboradas e multifuncionais.

Este trabalho também visou provar o desempenho de aplicativos para crianças em idade escolar que poderiam aprender mais sobre sustentabilidade e aspectos do meio ambiente.

Na área da educação ambiental, encontram-se diversas soluções. Gunasekera et al. desenvolveu uma infraestrutura de *IoT* com sensores para auxiliar no monitoramento de uma plantação e apoiar o ensino universitário em agricultura e ciência. Os autores prezaram por desenvolver um sistema *IoT* que auxiliasse na educação agrícola, afirmando também que os profissionais desta área devem ter conhecimento das oportunidades apresentadas pela Internet das Coisas.

Ainda na área da educação, o trabalho de (Valpreda and Zonda, 2016) realizado em Amsterdã procurou desenvolver um projeto a fim de ensinar às crianças a cultivar seus próprios alimentos dentro e fora de casa, misturando realidade real e virtual, conectando uma planta à Internet das Coisas. Utilizaram-se sensores e um aplicativo, o que torna esse processo mais divertido e útil para fins educacionais.

No contexto de irrigação na agricultura, o gerenciamento do uso de água é essencial para aumentar o rendimento das lavouras e diminuir os custos, ao mesmo tempo que contribui para a sustentabilidade ambiental (Kamienski et al., 2019). A implantação de um sistema *IoT* para irrigação permite quantificar com exatidão o volume de água necessária às plantas, o que implica em economia de água. Portanto, Kamienski et al. desenvolveu o projeto SWAMP, uma plataforma de gerenciamento inteligente de água baseada em Internet das Coisas para irrigação de precisão na agricultura. Os resultados do trabalho mostraram que o sistema foi capaz de fornecer desempenho adequado. Já no trabalho de (Gultom et al., 2017), também voltado à gestão de uso de água em irrigação, teve como intuito um sistema *IoT* focado no cultivo de plantas de pimentas na Indonésia, uma vez que estas plantas são muito sensíveis

a vários fatores, como umidade do solo e mudanças climáticas, e a falta de monitoramento prejudica a produção, o que resulta em impacto negativo na economia do país.

2.2 O processo de conscientização ambiental

No trabalho de (E. Curry and Derguech, 2018) foram explanadas as cinco etapas da mudança de comportamento alinhado ao processo de conscientização do usuário. Estas etapas serviram de base para o desenvolvimento do presente trabalho. São elas:

- **Pré-contemplação:** o usuário ainda não está ciente do problema. A solução neste caso é conscientizar o usuário acerca do mesmo utilizando argumentos satisfatórios, enfatizando as normas sociais e os benefícios de mudar o comportamento;
- **Contemplação:** o usuário está ciente do problema e possui o desejo de mudar seu comportamento diante disso. Neste caso, deve-se defender o uso do aplicativo, focando nos seus valores e o que ele perderia caso não utilizasse o sistema (no contexto de economia de água, ele perderia uma reeducação ambiental, deixaria de contribuir com o planeta, etc.). O aplicativo pode dispor de um painel que concentre todas as informações que o usuário precise, como dados de sensores;
- **Preparação:** o usuário pretende tomar uma ação positiva, mudar o comportamento. Logo, o aplicativo deve auxiliá-lo a planejar tais mudanças, direcionando-o a estabelecer metas e persuadindo-o a continuar comprometido com o novo objetivo, isto é, criar um novo hábito comportamental;
- **Ação:** o usuário pratica o comportamento desejado. O aplicativo, por sua vez, deve incentivar e motivar o usuário a continuar com o novo comportamento, como sugerindo novas atividades ou recompensando, por exemplo, com uma pontuação. Ver a pontuação subindo à medida que se realiza as tarefas do sistema anima muitos usuários a continuarem, principalmente se a pontuação for utilizada para algum *hanking* envolvendo outras pessoas, pois assim a competição convence diversas pessoas a darem mais de si mesmas;
- **Manutenção:** esta é a última etapa, em que o usuário se esforça para sustentar a mudança de comportamento, para sustentar o novo hábito. O sistema pode gerar alertas e notificações ao usuário lembrando-o sobre suas metas e também exibindo um gráfico sobre seu desempenho nos dias anteriores. Muitos usuários não gostam de ver seu desempenho caindo ao passar dos dias e farão o possível para ver uma melhoria.

3

Elementos *IoT*

Nesta seção serão apresentados os elementos necessários para compor um projeto de rega automatizada utilizando Internet das Coisas. Para tanto, alguns conceitos como microcontroladores e sensores *IoT* serão discutidos.

3.1 Microcontroladores

Um microcontrolador é um circuito integrado compacto projetado para controlar uma operação específica em um sistema embarcado. Um microcontrolador típico inclui periféricos de processador, memória e entrada / saída (E / S) em um único chip (*System on Chip - SoC*) (M. Rouse, 2016). Em outras palavras, o microcontrolador é um dispositivo que mistura hardware e software. Através da programação (geralmente em linguagem C ou Assembly) será possível controlar um hardware para realizar funções específicas de uma maneira relativamente simples e fácil. Algumas das funções que podem ser implementadas: somar dois números e outras operações matemáticas; construir o próprio controle remoto acoplando um teclado e dispositivo de laser para mandar e receber informações de qual botão foi pressionado; acoplar um sensor de temperatura e configurar o microcontrolador para disparar caso a temperatura exceda determinado valor no ambiente; dentre outras.

3.1.1 Arduino

Um dos microcontroladores mais comum é o Arduino, isto é, uma plataforma eletrônica *open source* baseada em hardware e software que permite a implementação de variadas funcionalidades de maneira simples. A placa funciona com tensões entre 3,3 e 5 volts. As funcionalidades são implementadas em linguagem Arduino (extensão .ino) através de uma plataforma própria chamada *Arduino IDE*, também disponível gratuitamente. As placas são capazes de ler entradas, como entradas de sensores de umidade de solo ou de luminosidade

- trazendo para o contexto de rega automatizada com *IoT* - e transformar essas entradas em uma saída, isto é, uma tomada de decisão como ligar o motor de uma bomba ou abrir uma válvula. (Arduino Website, 2019)

O arduino é uma placa muito poderosa, e, sendo assim, pode viabilizar a implementação de diversos projetos, dos mais simples aos mais complexos e criativos - uma funcionalidade bastante popularizada é a automação residencial, como acender luzes, ar condicionado e TV não utilizando os próprios dispositivos. Veja a foto do microcontrolador: figura 3.1.

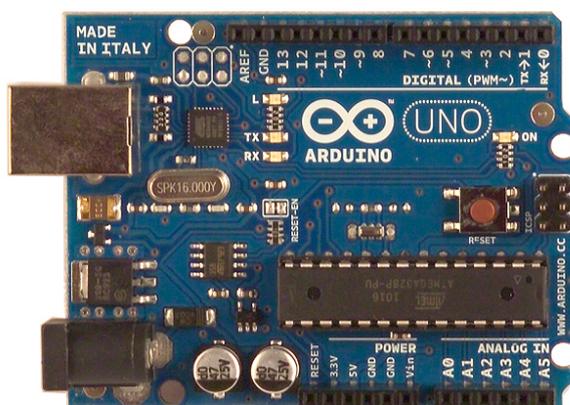


Figura 3.1: Microcontrolador Arduino UNO. [Retirada de <https://www.techtudo.com.br/noticias/noticia/2013/10/o-que-e-um-arduino-e-o-que-pode-ser-feito-com-ele.html>]

O arduino UNO pode ser comprado isoladamente ou junto a um kit, seja ele um kit iniciante ou kit avançado. Caso o desenvolvedor precise de um arduino um pouco mais robusto, ele pode optar pelo arduino MEGA, um arduino com maior número de portas, com mais UARTs, memória flash, RAM e ROM.

3.1.2 Esp8266 NodeMCU

Outro microcontrolador bastante difundido é o Esp8266 NodeMCU. Ele viabiliza a conexão Wi-Fi e realiza conexões TCP/IP, permitindo assim que sensores e atuadores se conectem à internet (Espressif, 2015). É um módulo extremamente econômico, tanto por ser baixo custo quanto por consumir pouca energia - ele opera entre 1,8V e 3V. A figura 3.2 o exibe.

O fabricante dos Esp8266, sistema Espressif, fornece um Kit de Desenvolvimento de Software (SDK) com uma API para programar o módulo (Kolban, 2015). Com a API, pode-se implementar o protocolo MQTT - *Message Queue Telemetry Transport* -, desenvolvido para aplicações de *Internet of Things*.

No mecanismo MQTT, um cliente que precisa de dados (conhecido como *subscriber*) registra seus interesses com um servidor (também conhecido como *broker*). O cliente produzindo dados (conhecido como *publisher*) envia os dados para um servidor e este servidor

encaminha os novos dados para o *subscriber*. Nesse contexto, os nós se inscrevem e publicam mensagens em um canal chamado *topic*. Para que isso seja possível, é necessário escolher um mecanismo de armazenamento de dados, como por exemplo, computação em nuvem (Bhatt and Patoliya, 2016). O armazenamento de dados em sistemas *IoT* serão discutidos mais adiante.

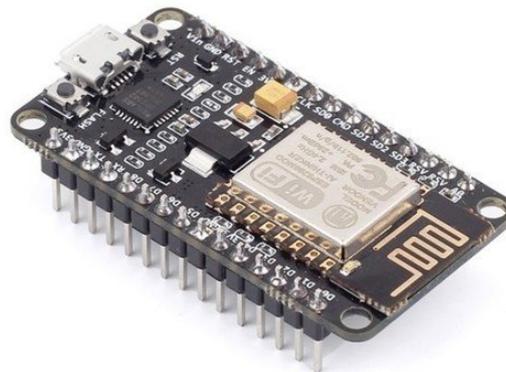


Figura 3.2: Microcontrolador Esp8266 NodeMCU. [Retirada de https://br.banggood.com/Geekcreit-Doit-NodeMcu-Lua-ESP8266-ESP-12F-WIFI-Development-Board-p.html?cur_warehouse=USA]

O chip soldado no Esp8266 NodeMCU foi utilizado também em outras placas e módulos, com o intuito de disseminar e facilitar o uso do mesmo. Existem placas de diversos tamanhos e números de pinos. Veja a figura 3.3 para entender a composição da placa Esp8266 NodeMCU.

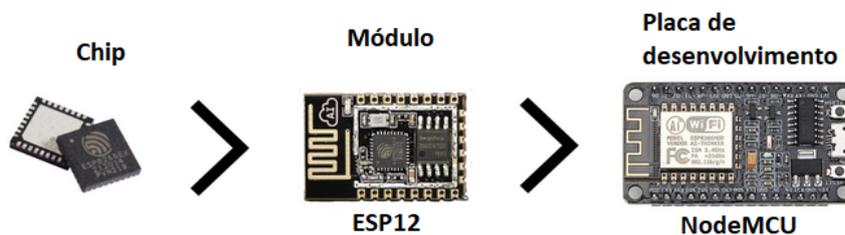


Figura 3.3: O chip foi acoplado em um módulo, chamado ESP12, e em seguida o módulo foi acoplado na placa NodeMCU. [Retirada de (Banggood, 2014)]

3.2 Sensores em IoT

Um sensor é, em suma, um dispositivo capaz de detectar e responder algum estímulo. Existem bastantes tipos de sensores, e estes respondem à diferentes estímulos, tais como: variação de temperatura, de luminosidade, de movimento, de pressão, etc. Após receber um estímulo, o sensor deve, em seguida, emitir um sinal para que outro dispositivo o capte e o interprete. Alguns desses sensores serão discutidos abaixo.

3.2.1 Redes de sensores sem fio

Consiste em um grande número de sensores, possibilitando a coleta, processamento, análise e disseminação de informações valiosas, reunidas em uma variedade de ambientes. (Gubbi et al., 2013) Em outras palavras, uma rede de sensores sem fio é uma rede sem fio de dispositivos como sensores, distribuídos em um ambiente a fim de monitorar fenômenos, como a oscilação de temperatura em uma região. Em uma rede de sensores sem fio, os nós podem se comunicar entre si e cada nó é composto por um sensor, memória, microcontrolador, um transceptor sem fio e uma bateria. A rede é formada por estes nós, o gateway que viabiliza a conexão da rede com o computador, e por fim a estação base, que é o dispositivo que recebe as informações do gateway e direciona para o computador. Uma vez as informações armazenadas no computador, as mesmas podem ser tratadas e usadas como se deseje. A figura 3.4 ilustra o funcionamento de uma rede.

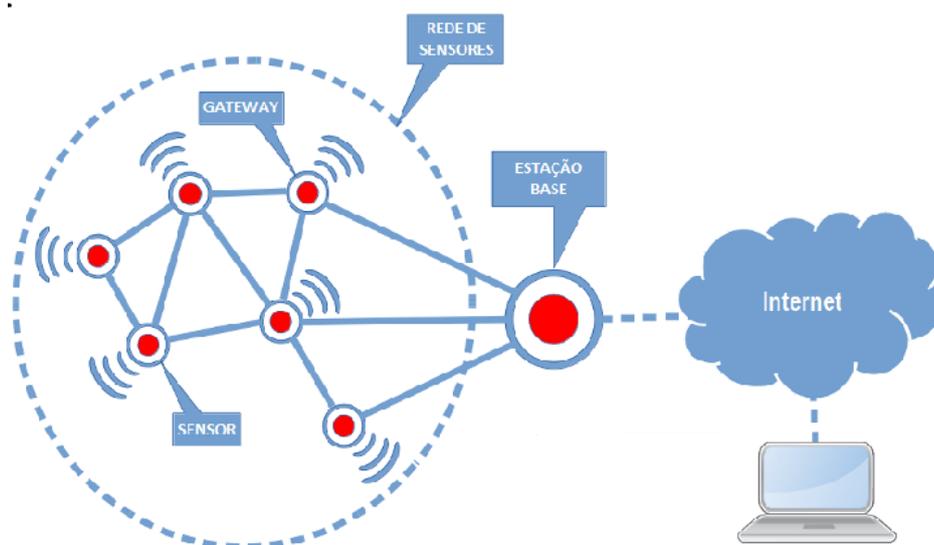


Figura 3.4: Funcionamento de uma rede de sensores sem fio. [Retirada de (RSSF UFRJ, 2018)]

A tecnologia de redes de sensores sem fio tem o potencial de revelar mudanças dinâmicas e refinadas em variáveis monitoradas de uma paisagem externa (Cardell-Oliver et al., 2005). O uso de redes de sensores sem fio foi estudado em muitos trabalhos, como em (TongKe,

2013) e (K. Kansara and Jani, 2015), que utiliza redes de sensores sem fio para desenvolver um sistema de irrigação automatizada *IoT* na Índia, o qual consiste em um sistema com cinco estações distribuídas pelo campo, estação de controle de irrigação e uma estação base. As estações monitoram as condições do solo e do ar, isto é, suas respectivas umidade e temperatura.

3.2.2 Sensor de umidade do solo

Este sensor é responsável por detectar variações de umidade do solo. Com base na resposta de medição deste sensor, é possível calibrá-lo, ajustando seus valores, a fim de determinar se o solo está com baixa ou alta umidade. É um dos sensores considerados mais relevantes em sistemas de rega automatizada com *IoT*, pois sua resposta de medição auxiliará na tomada de decisão de quando a área deve ser irrigada ou não. Sendo assim, trabalhos como (Rajalakshmi and Mahalakshmi, 2016), (Gubbi et al., 2013), (TongKe, 2013), e muitos outros utilizam sensores de umidade de solo. Em (Rajalakshmi and Mahalakshmi, 2016), este sensor possui duas pontas, as quais são cravadas no solo e detectam a umidade do solo através da corrente que passa pelas duas pontas. Se o solo estiver seco, a resistência nele será maior, passando menos corrente. Contudo, se o solo estiver úmido, a resistência será menor e passará mais corrente. Veja a figura 3.5 do sensor de umidade do solo.

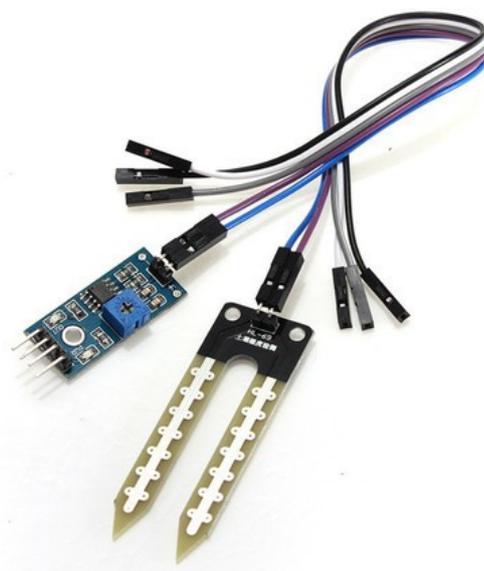


Figura 3.5: Sensor de umidade do solo. [Retirada de <https://www.vidadesilicio.com.br/sensor-umidade-solo-higrometro>]

3.2.3 Sensor de temperatura e umidade do ar

O vapor de água no ar é definido como uma medida da umidade do ar. No contexto de rega *IoT*, quando há uma mudança de temperatura, a umidade do ar também varia, e essa mudança acontece antes e depois de irrigar. (Rajalakshmi and Mahalakshmi, 2016) utiliza o sensor de temperatura DHT11, mostrado na figura 3.6, sensor de baixo custo que utiliza um termistor e um sensor capacitivo para medir a temperatura e a umidade do ar. Junto com o sensor de umidade do solo, os dois são os parâmetros mais relevantes em sistemas *IoT* de rega automatizada.

Este sensor permite realizar leituras de temperatura de 0°C até 50°C com precisão de 2°C e umidade do ar entre 20% e 90% com precisão de 5%. O sensor de temperatura é do tipo termistor NTC e o sensor de umidade do ar é do tipo HR202.

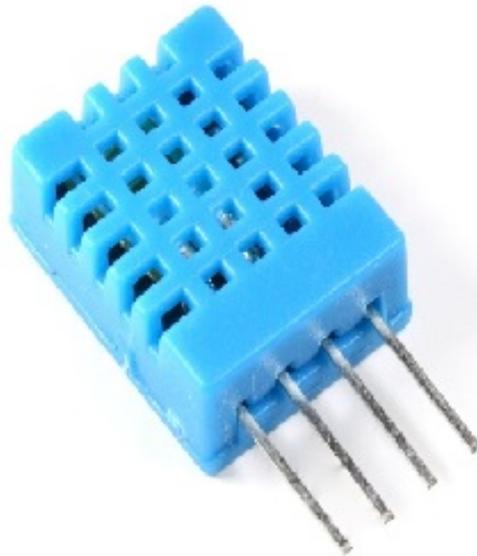


Figura 3.6: Sensor de umidade de temperatura e umidade do ar DHT11. [Retirada de <https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-dht11/>]

A umidade relativa do ar (UR) pode ser definida como a relação da pressão real do vapor de água (ϵ) para a pressão de vapor de equilíbrio sobre um plano de água (ξ). (Lawrence, 2005).

$$UR = 100 \frac{\epsilon}{\xi} \quad (3.1)$$

onde,

- **UR**: Umidade relativa do ar (%)
- ϵ : a pressão real do vapor de água, medida em pascals (Pa)

- ξ : a pressão de vapor de equilíbrio sobre um plano de água, também medida em pascals (Pa)

Na figura 3.7 é possível notar que a pressão do vapor é uma propriedade física que varia à medida que o valor da temperatura aumenta.

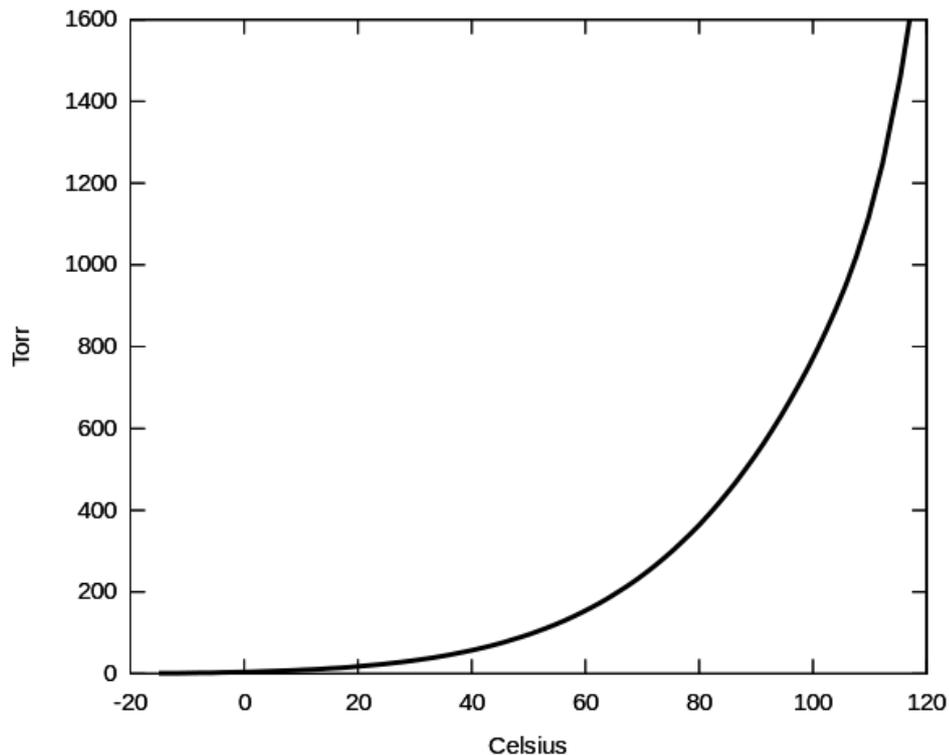


Figura 3.7: Gráfico que exhibe a relação Pressão vapor x Temperatura. [Retirada de https://commons.wikimedia.org/wiki/File:Water_vapor_pressure_graph.jpg]

3.2.4 Sensor de luminosidade

O Sensor de Luminosidade faz uso da variação da resistência para determinar a intensidade da luz. À medida que a luminosidade aumenta, a resistência diminui, e vice-versa. A tensão, por sua vez, aumenta à medida que a luminosidade também aumenta (Rajalakshmi and Mahalakshmi, 2016).

A figura 3.8 mostra um sensor de luz do tipo BH1750. Este sensor também pode ser encontrado com um módulo de sensor de luminosidade GY-302 para facilitar o uso, com faixa de medição entre 1 e 65,535 lux - a iluminação é medida em lux, isto é, lumens por metro quadrado.

A luz é uma forma de radiação eletromagnética que possui uma frequência visível ao olho humano e pode se propagar no vácuo com velocidade aproximada de 300 mil km/s. As frequências de luz visíveis ao olho humano são chamadas de espectro visível, que são ondas de comprimento de 400 até 700 nanômetros. A figura 3.9 o exhibe.



Figura 3.8: Sensor de Luz BH1750FVI Lux. [Retirada de <https://www.filipeflop.com/produto/sensor-de-luz-bh1750fvi-lux/>]

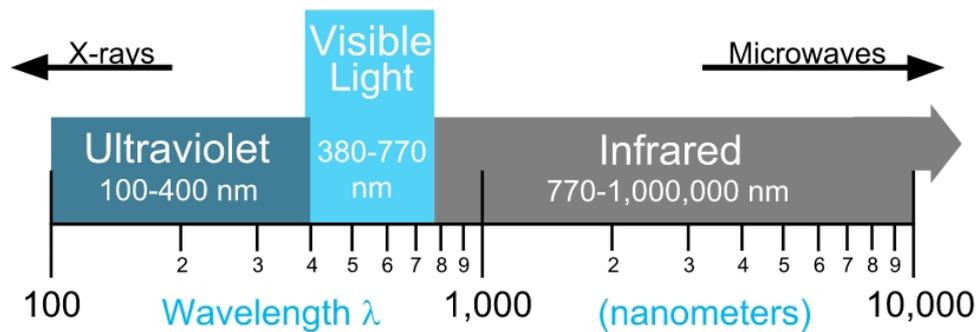


Figura 3.9: As possíveis frequências de luz. O espectro visível está expresso em azul claro. [Retirada de (?)]

A intensidade de iluminação ou iluminância (E) é uma grandeza de luminosidade medida em lux que relaciona o fluxo luminoso refletido em uma superfície e a área. Abaixo é apresentada uma ilustração da iluminância na figura 3.10.

$$E = \frac{\phi}{A} \quad (3.2)$$

onde,

- E : a Iluminância, medida em lux
- ϕ : o fluxo luminoso, medido em lumens
- A : a área, medida em metros quadrados (Universidade USP, 2018)

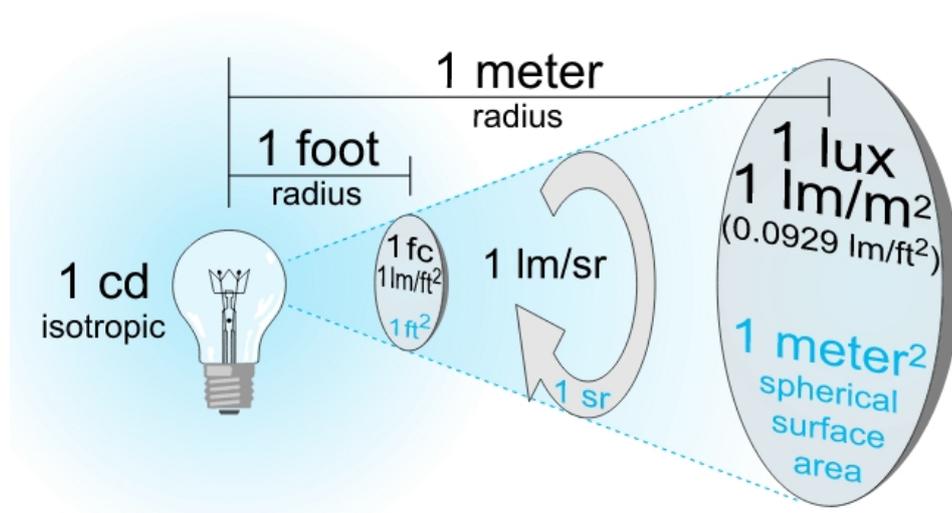


Figura 3.10: Iluminância. [Retirada de (?)]

3.3 Atuadores

Atuadores são componentes que realizam alguma ação no ambiente. Eles recebem comandos para serem realizados em um determinado contexto, como por exemplo, um atuador pode ser uma lâmpada que será acesa para alertar que existe algum problema no sistema.

Em um ambiente de rega, podem-se utilizar alguns atuadores, tais como uma lâmpada e uma bomba de água. A lâmpada poderia ser acesa próxima à planta quando o sistema detectasse que a temperatura ambiente não está propícia para a mesma. A bomba de água, por sua vez, poderia ser acionada sempre que a umidade do solo estiver baixa, a fim de regular o nível. Mostra-se na figura 3.11 uma mini bomba de água utilizada em projetos *IoT*.

3.4 Aplicativos mobile

Os dispositivos móveis, que são dispositivos eletrônicos de fácil mobilidade como *smartphones* e *tablets*, popularizam-se bastante nos últimos anos a ponto de se tornarem um item essencial para as pessoas, visto que passou a auxiliar de diversas maneiras no dia a dia, como melhorando a comunicação com sua rede de contatos, utilização de redes sociais que já existiam em versão *web*, informar-se sobre acontecimentos no mundo inteiro, ver e compartilhar vídeos e imagens, etc. Sendo assim, surgiram mais oportunidades para criação de aplicativos mobile de inúmeros ramos - aplicativos comerciais, hospitalares, de gestão de empresas e muitos outros.

(M. Lima, 2018) revelou em uma pesquisa que no Brasil existem 220 milhões de *smartphones* em uso, isto é, mais do que um *smartphone* por habitante. Isso mostra o quanto se pode lucrar com a criação de aplicativos, e a figura 3.12 apresenta um comparativo de lucro entre aplicativos com sistemas operacionais *Android* e *iOS* nos últimos anos.



Figura 3.11: Bomba D'água Submersiva 3-6V 100L/H. [Retirada de <https://www.filipeflop.com/produto/bomba-dagua-submersiva-3-6v-100l-h/>]

Segundo (LTD, 2017), o sistema operacional mais utilizado em 2016 é o *Android*, presente em mais de 86% dos *smartphones*, seguido pelo *iOS*, com mais de 12%. Veja a figura 3.13 para acompanhar a utilização dos sistemas operacionais nos últimos anos.

Para desenvolver aplicativos mobile, é necessário escolher a melhor abordagem, por exemplo, optar por uma tecnologia que não seja ultrapassada rapidamente, que limite suas opções de inovação ou que crie dependência aos desenvolvedores que criaram o *app*. Cada abordagem possui pontos fortes e fracos, e nos momentos iniciais do desenvolvimento do aplicativo é importante conhecê-los para optar pelo melhor caminho para a implementação, evitando refazer parte ou todo o código. Segue abaixo as principais abordagens tecnológicas.

- **Apps nativos (*Android* e *iOS*):** A melhor abordagem em caso de buscar ótima performance, processamento de informações e integrações com o sistema operacional de forma rápida, isto é, utilizam os recursos do *hardware*. As linguagens utilizadas são Java para *Android* e *Swift* para *iOS*, com disponibilidade de IDEs (Ambiente de Desenvolvimento Integrado) para auxiliar no desenvolvimento, pois possuem ferramentas que facilitam a implementação e o *debugging* do código. No entanto, será necessário desenvolver duas aplicações distintas, uma para *Android* e outra para *iOS*, o que reflete no orçamento do sistema, principalmente a longo prazo, uma vez que serão necessários realizar manutenção e suporte contínuos. Deve-se também considerar a publicação e atualização dos aplicativos nativos nas duas lojas, *App Store* e *Google Play* (?). **Cenário ideal para uso:** aplicativos mobile complexos, focado na interface gráfica do usuário ou que possui processamento de dados pesados.
- **Página *mobile* responsiva:**

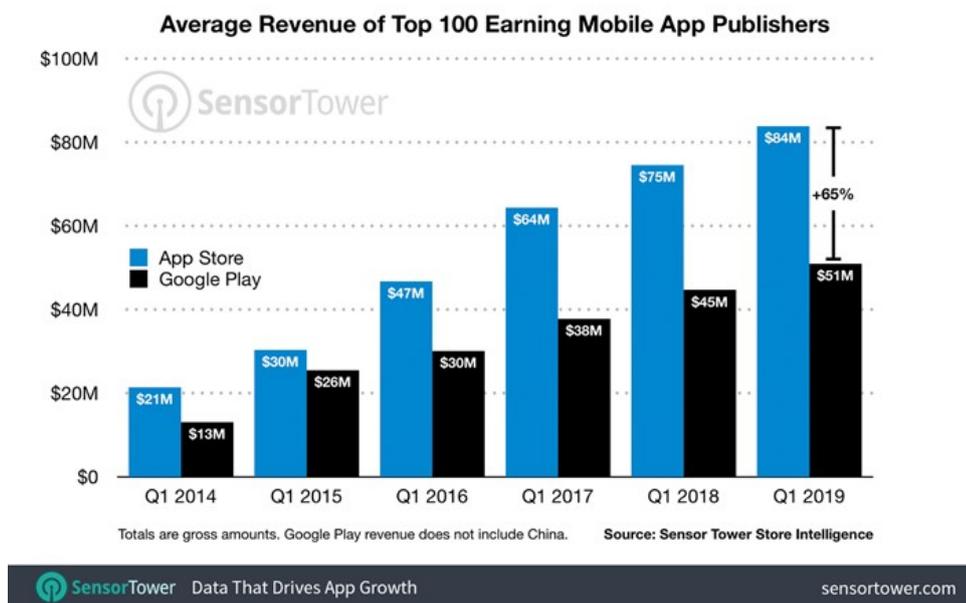


Figura 3.12: Lucro médio dos 100 principais apps para dispositivos móveis - um comparativo com a *App Store* e *Google Play*. [Retirada de <https://www.tecmundo.com.br/mercado/142951-app-store-tem-lucro-65-superior-play-store-1-trimestre.htm>]

Esta abordagem adequa-se ao tamanho da tela do usuário, viabilizando a criação de um site que funcione no *browser* (versão *web*) e também nos *smartphones* ou *tablets*. Comparando a um aplicativo nativo, esta solução é desenvolvida mais rápida e possui um menor custo de manutenção, pois existe apenas uma versão para gerenciar e realizar manutenção, independente do sistema operacional do dispositivo. Além disso, é mais fácil e rápido de atualizar o sistema, já que se trata de uma página e não é preciso atualizar em uma loja virtual. Entretanto, aplicativos com esta abordagem possui desempenho mais lento, uma vez que o processamento ocorre no servidor *web*, e também é mais limitado, pois como se trata de uma página *web* não é possível usar recursos como a câmera. (Corral et al., 2012) **Cenário ideal:** desenvolvimento de um aplicativo de baixo investimento, com interface gráfica mais simples e com baixo processamento de dados, uma vez que o processamento é feito no servidor *web*.

- **Plataforma híbrida:**

Permite o desenvolvimento de um aplicativo em uma única linguagem de forma que seja aproveitado para dispositivos com sistemas operacionais *Android* e *iOS*. Diz-se plataforma híbrida pois esta abordagem combina o desempenho de um aplicativo nativo com a facilidade de uso de uma página *mobile* responsiva. O *React Native* é a plataforma referência atualmente para este tipo de abordagem, baseado em *JavaScript*. Assim como a página *mobile* responsiva, a plataforma híbrida também só necessita de

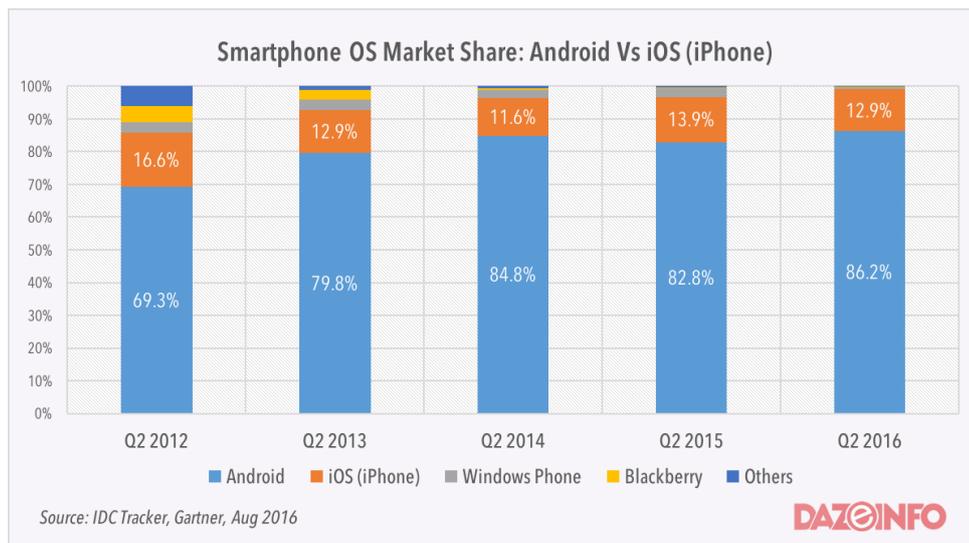


Figura 3.13: Um comparativo da utilização de sistemas operacionais em *smartphones* de 2012 até 2016. [Retirada de (LTD, 2017)]

um código para atender os dois sistemas operacionais, e assim os custos de desenvolvimento e manutenção são bem menores. Além disso, o desempenho é parecido com o de um app nativo quando o aplicativo é razoavelmente simples, e é possível usar recursos como câmera, GPS e outros. Contudo, aplicativos mais complexos exigem linhas de código mais específicas para cada tipo de sistema operacional, o que pode acabar impactando o fato de gerenciar apenas um código (Dalmasso et al., 2012). **Cenário ideal:** aplicativos que necessitem gerenciar dados e exibir uma interface gráfica rápida para o usuário.

Logo, para se escolher a melhor abordagem é importante conhecer o sistema a ser criado. Quanto mais detalhes do sistema os desenvolvedores possuírem, mais fácil será escolher a melhor forma de desenvolver o aplicativo.

3.5 Cloud Computing

Cloud Computing ou Computação em Nuvem é o fornecimento de serviços de computação, tais como servidores, aplicativos, serviços, armazenamento, software, inteligência, através da Internet (a "nuvem") com o intuito de oferecer inovações mais rápidas, recursos flexíveis e economias de escala com gerenciamento simples (Zhang et al., 2010). A figura 3.15 ilustra a arquitetura de um serviço de nuvem.

Normalmente se paga pelo uso de serviço de nuvem, o que auxilia na redução dos custos da empresa. A computação em nuvem é um serviço; logo, ela elimina gastos como a compra de hardware e software, configuração e execução de datacenters, especialistas para realizar suporte e manutenção, dentre outros.

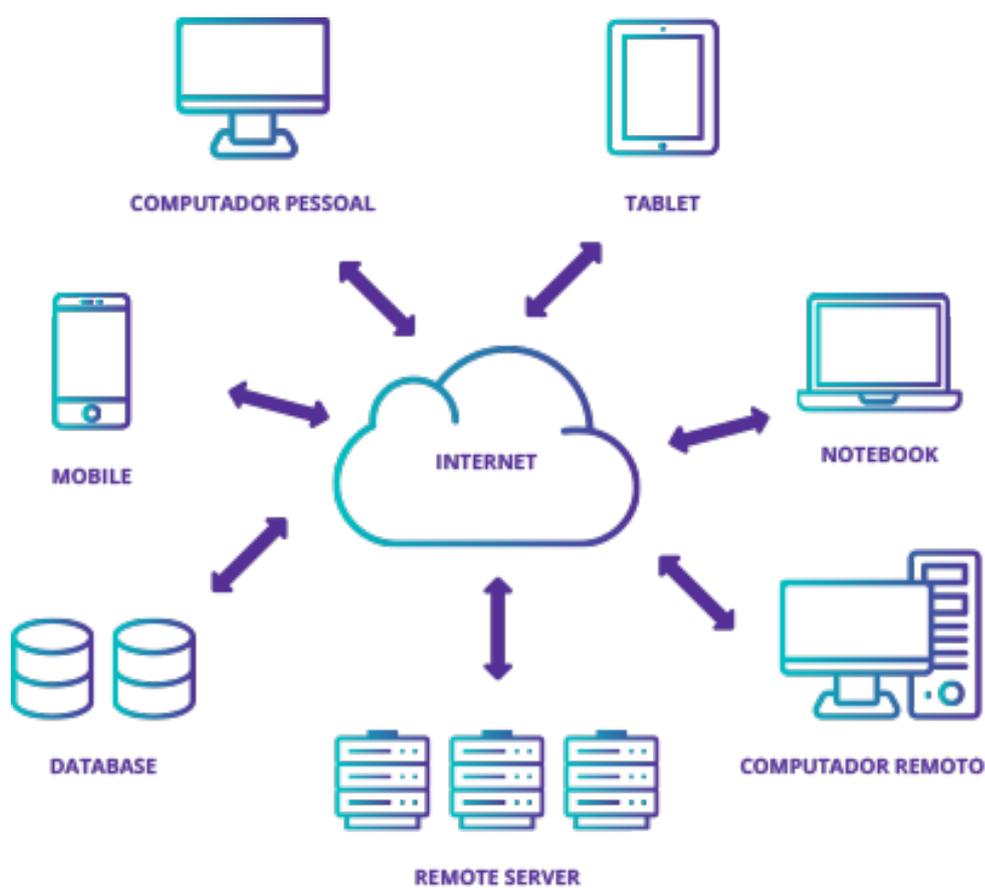


Figura 3.14: Arquitetura de um serviço de computação em nuvem. O centro da imagem exhibe a nuvem, onde tudo fica armazenado. Observe que a comunicação entre os dispositivos e a nuvem é bidirecional. [Retirada de <https://carolinadatarecovery.com/cloudy-with-a-chance-of-data-how-the-cloud-works/>]

Outro benefício da computação em nuvem é a escala, isto é, disponibilizar a quantidade adequada de recursos, assim como potência de computação, armazenamento e largura de banda sempre que preciso. A produtividade também aumenta, uma vez que a empresa não deve se preocupar com a manutenção e execução em datacenters, abstraindo essa complexidade. *Cloud computing* ainda impacta positivamente a questão de desempenho, confiabilidade dos dados e segurança.

Pode-se criar um serviço de nuvem de três maneiras, são elas: nuvem pública, nuvem privada e nuvem híbrida. A nuvem pública pertencem a um serviço de nuvem terceirizado e administrado por ele, como a *Amazon Web Services*. O provedor de serviço de nuvem possui toda a infraestrutura para execução e não é necessário se preocupar com a manutenção e suporte. Já a nuvem privada é um serviço de nuvem usados exclusivamente por uma única empresa, um serviço personalizado, mantido em uma rede privada. A nuvem privada pode ser física ou a empresa paga a hospedagem em um provedor de serviço de nuvem. Por fim, a nuvem híbrida, que combina nuvens públicas e privadas de modo que seus dados podem ser compartilhados entre elas.

A grande maioria dos modelos de serviços em nuvem dividem-se em cinco categorias, são elas: IaaS (infraestrutura como serviço), PaaS (plataforma como serviço), SaaS (software como serviço), BaaS (*backend* como serviço) e Computação sem servidor.

A mais básica categoria é a IaaS, pois ela permite que se alugue a infraestrutura (servidores, armazenamento, etc.) e o pagamento é definido conforme o uso (Bhardwaj et al., 2010).

A categoria PaaS fornece serviço de computação em nuvem com um ambiente para desenvolvimento, teste e gerenciamento. Assim, o usuário se encarrega de criar seus próprios aplicativos, aumentando a produtividade do programador, produtos criados pela própria empresa e redução de custos de desenvolvimento (Lawton, 2008).

A SaaS refere-se a um método para disponibilizar *softwares* sob demanda pela Internet. Os provedores de nuvem hospedam, gerenciam e dão suporte a esses *softwares*. A conexão é muito simples, basta o acesso à Internet, normalmente por um *browser* (Website, 2018).

O BaaS, por sua vez, é um serviço que funciona semelhante a um *middleware*, isto é, um programa que viabiliza a comunicação entre *softwares*, como por exemplo comunicar aplicações *mobile* e/ou *web* a serviços a partir de APIs e SDKs. Em outras palavras, o desenvolvedor não precisa se preocupar com a implementação do backend, apenas com o *frontend*, aumentando assim a produtividade (?). Um exemplo de serviço BaaS bastante utilizado é o *Firebase*, lançado pelo *Google* em 2008. Com essa ferramenta o desenvolvedor consegue abstrair funcionamento em vários tipos de dispositivo, infraestrutura, segurança, implementação de técnicas de autenticação e escalabilidade. O *Firebase* é acessado na *web*, e a figura 3.15 exibe a interface principal da ferramenta.

E por fim, a Computação sem servidor, que foca na criação de funcionalidades dos aplicativos, sem se preocupar com o gerenciamento de servidor e a infraestrutura necessária.

Diversos elementos apresentados neste capítulo foram usados no desenvolvimento do *IO* para rega de plantas a ser descrito no próximo capítulo.

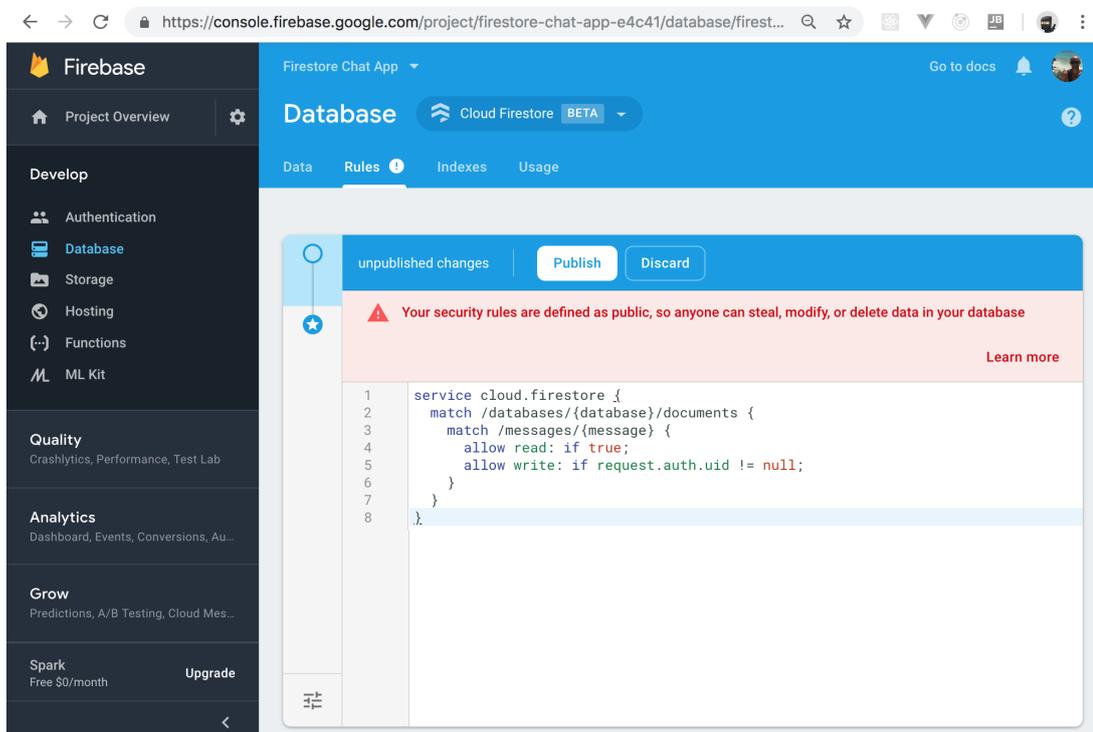


Figura 3.15: Interface gráfica do *Firebase*, exibida no navegador. [Retirada de <https://carolindatarecovery.com/cloudy-with-a-chance-of-data-how-the-cloud-works/>]



Metodologia

Neste capítulo serão apresentados os módulos do sistema *IoT* de rega automatizada desenvolvido. Adicionalmente, serão mostradas as tecnologias utilizadas, a criação do dispositivo, as telas da aplicação *mobile*, desde a configuração até a conexão com a *Firebase*, etc.

Este sistema *IoT* compreende o uso de apenas um sensor: o sensor de umidade do solo. Serão desconsideradas no momento outras variáveis como temperatura e luminosidade.

4.1 Módulos do Sistema

O sistema pode ser compreendido em três módulos: sistema embarcado, camada de nuvem e aplicativo *mobile*.

O **sistema embarcado** consiste em dois dispositivos, um sensor de umidade do solo e um módulo Wi-Fi Esp8266 NodeMCU. O sensor envia dados para o NodeMCU através da porta analógica A0 do módulo. Em seguida, é realizada uma calibração no valor recebido pelo sensor. Por fim, o módulo, uma vez conectado a uma rede Wi-Fi previamente configurada com os dados de nome e senha, se comunica com a *Firebase* e envia os dados recebidos pelo sensor junto com a hora e a data. Cada um dos dados são enviados com uma *label* de identificação - para facilitar o reconhecimento na nuvem -, encapsulados em um JSON (*JavaScript Object Notation*) e finalmente enviados para o *Real-Time Database* da *Firebase*.

A **camada de nuvem** é um serviço de nuvem que utiliza a plataforma BaaS, a qual permite abstrair a implementação do *backend*, chamada *Firebase*. Graças a esse tipo de modelo de nuvem, seria possível implementar soluções para *Android*, *iOS* e *Web*, apenas é necessário saber usar a API. A *Firebase* separa os dados em *labels*, facilitando a identificação das variáveis que foram recebidas por diversos sensores, se houver.

O **aplicativo *mobile*** é a interface gráfica do usuário, pois ele exibe os dados do sensor de umidade do solo em tempo real. O aplicativo é conectado com a *Firebase*, e solicita o recebimento de dados novos, fica "escutando" caso algum dado novo seja armazenado na

nuvem. O aplicativo deve solicitar os dados para cada *label* que exista na nuvem. No caso deste sistema, a única *label* que está configurada é a do sensor de umidade do solo. Por fim, o aplicativo *mobile* exibe os dados do sensor de umidade do solo em tempo real. Em caso de valor do sensor com porcentagem baixa de umidade do solo, uma notificação é lançada através do app para que o usuário tome uma providência, como por exemplo, regar a planta.

Para visualizar o fluxo em um diagrama, veja a figura 4.1.

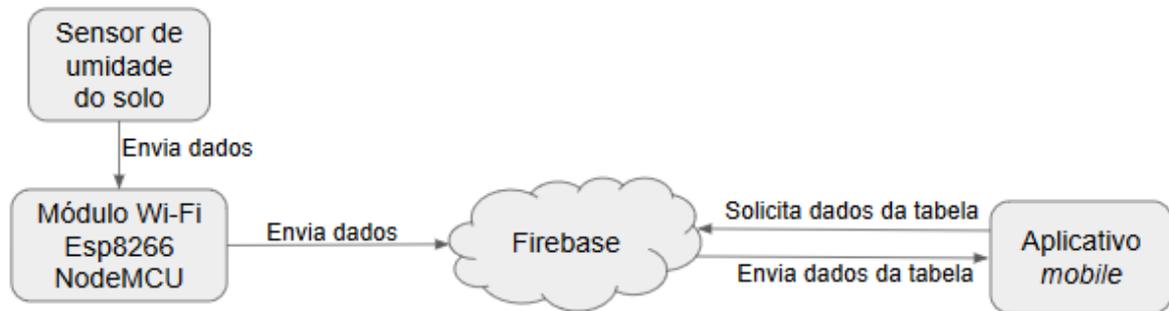


Figura 4.1: Fluxograma do sistema.

É interessante também observar o sistema sob a perspectiva de agentes e ambientes. Agente é qualquer entidade que perceba seu ambiente através de sensores e atue sobre o ambiente através de atuadores (Russel et al., 2013). Nesta aplicação, o agente no ambiente é o sensor de umidade do solo e o atuador é a bomba de água. A figura 4.2 ilustra esse modelo.

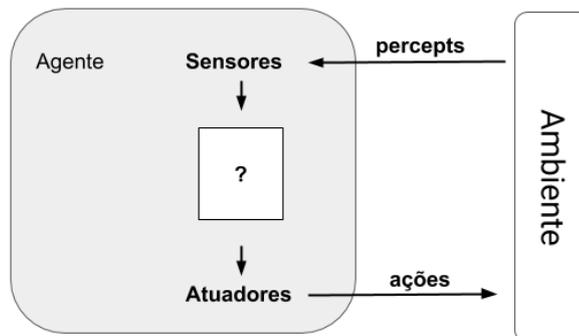


Figura 4.2: Modelo de agente e ambiente de Russel et al.

Para complementar o entendimento do modelo, deve-se entender alguns conceitos:

- **percept**: uma observação, uma entrada em um instante. É a coleta do dado do sensor de umidade do solo em um determinado momento;
- **história perceptiva**: sequência completa de todas as observações, todas as possíveis entradas;
- **ações**: saídas especificadas por um mapeamento de uma sequência perceptiva em uma ação.

A história perceptiva do sistema *IoT* desenvolvido consiste em sempre que o agente do ambiente coletar uma umidade de solo abaixo de 65%, a bomba de água ser acionada por determinado tempo a fim de restabelecer a umidade.

4.2 Ferramentas Utilizadas

No contexto de **sistema embarcado**, utilizaram-se o módulo Wi-Fi Esp8266 NodeMCU e o sensor de umidade do solo.

O módulo consiste em uma placa de desenvolvimento que possui o chip Esp8266, uma porta USB e um regulador de tensão para 3,3V. A programação do módulo foi realizada e compilada na *Arduino IDE*, com a transferência de código por um cabo micro USB (usado para carregar *smartphone Android*). Note na figura 4.3 que o NodeMCU possui os pinos *Vin* e *Ground*, os quais fornecem energia para o sensor (o *Vin* é o pino de alimentação externa, recomendado 5V / 1A positivo e o *Ground*, o aterramento da placa). Este sistema é ligado à energia elétrica por meio da entrada micro USB e uma fonte de alimentação. Sobre a conexão com o sensor de umidade, ela é realizada no pino ADC0, usado para receber dados analógicos.

O sensor de umidade do solo utilizado neste sistema foi o mesmo apresentado na figura 3.5. Na sessão 3, mostraram-se mais detalhes sobre ele. Note que o sensor possui duas pontas, e as mesmas devem estar cravadas no solo.

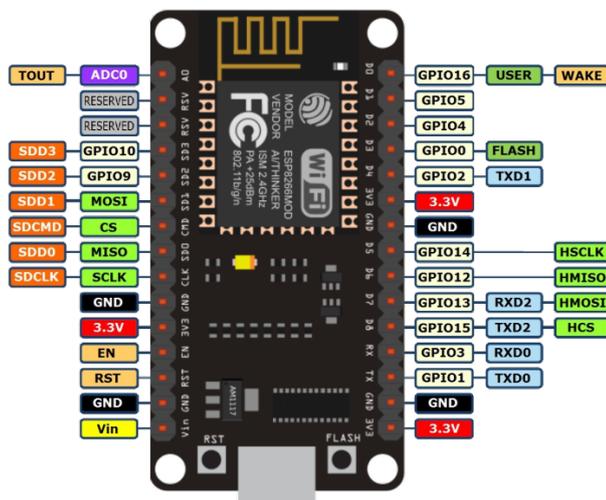


Figura 4.3: Pinos do Esp8266 NodeMCU. [Retirada de <https://blog.eletrogate.com/nodemcu-esp12-introducao-1/>]

Na **camada de nuvem** foi utilizada a *Firebase*, a qual possui *Realtime Database* e *Cloud Firestore*. Neste sistema foi utilizado o *Realtime Database*, o qual consiste em um banco de dados NoSQL - isto é, um banco de dados não relacional - na nuvem para armazenar e sincronizar dados do aplicativo. Esses dados, armazenados em JSON, são mantidos localmente

e os eventos em tempo real são acionados mesmo em modo *offline*. Como a *Firebase* é propriedade da *Google*, também é possível enviar os dados para a *Google Cloud*.

No contexto de segurança das informações, isto é uma responsabilidade da *Google* e o desenvolvedor a abstrai. A *Firebase* também garante alta escalabilidade e performance e o baixo custo, uma vez que o banco de dados está na nuvem e a empresa não precisa se preocupar em instalar *datacenters*.

A ferramenta utilizada no módulo de **aplicativo mobile** foi o *Android Studio*, uma IDE gratuita, mostrada na figura 4.4, utilizada para desenvolver aplicativos *Android* nas linguagens Java, C++, Kotlin e também o framework Flutter. A aplicação *mobile* deste sistema foi codificada em Flutter, o qual faz uso da linguagem Dart e é um SDK de código aberto criado pela *Google* para desenvolvimento de aplicativos *Android*, *iOS*, *Desktop* e *Web*.

A Flutter foi escolhida pois é uma solução para desenvolvimento híbrido de aplicativos, isto é, possui apenas uma base de código que gera aplicativos *Android* e *iOS*. Além disso, o Flutter fica na camada de interface gráfica e não chama componentes nativos do sistema operacional, e é desenhado em um canvas que aumenta a performance e a fluidez a nível de um aplicativo nativo. Se necessário, ainda é possível aproveitar esta aplicação *mobile* para criar uma aplicação *web*.

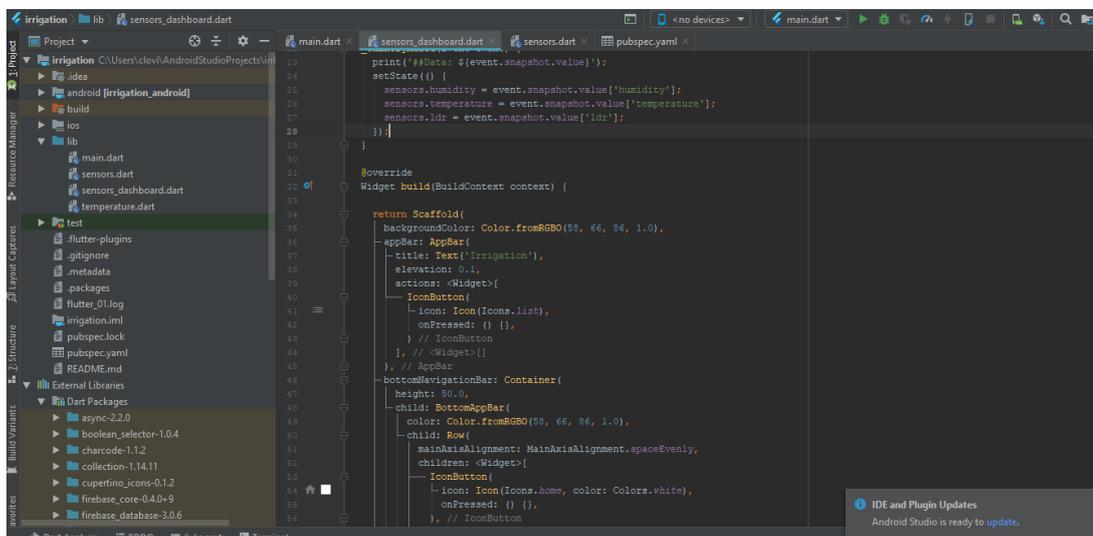


Figura 4.4: Tela principal do *Android Studio*.

4.3 Desenvolvimento do sistema IoT

O desenvolvimento foi dividido em dois módulos: o *hardware*, que engloba o NodeMCU, o sensor de umidade do solo, o relê e a bomba de água; e o *software* embarcado, que se trata da comunicação entre o módulo NodeMCU e a *Firebase*. O sistema construído é exibido na figura 4.6 e a bomba de água utilizada, na figura 4.8.

4.3.1 Hardware

Em termos de *hardware*, o sistema é composto pelo módulo Wifi ESP8266 NodeMCU, o sensor de umidade do solo, o relê e a bomba de água. Utilizou-se uma *protoboard* para montar o circuito elétrico entre esses componentes. Na prática, o circuito elétrico pode ser visto no esquemático da figura 4.5, e na prática, na figura 4.7, a qual mostra as conexões realizadas para permitir o funcionamento dos componentes em uma planta. O sistema é ligado à energia elétrica por meio da entrada micro USB do módulo NodeMCU, a qual suporta uma voltagem de até 3.6V e corrente de 1A. Já a bomba de água, conectada ao relê, é ligada à energia elétrica por meio de uma fonte de alimentação de 5V. Este mesmo dispositivo montado na *protoboard* foi utilizado para todas as plantas em análise neste sistema.

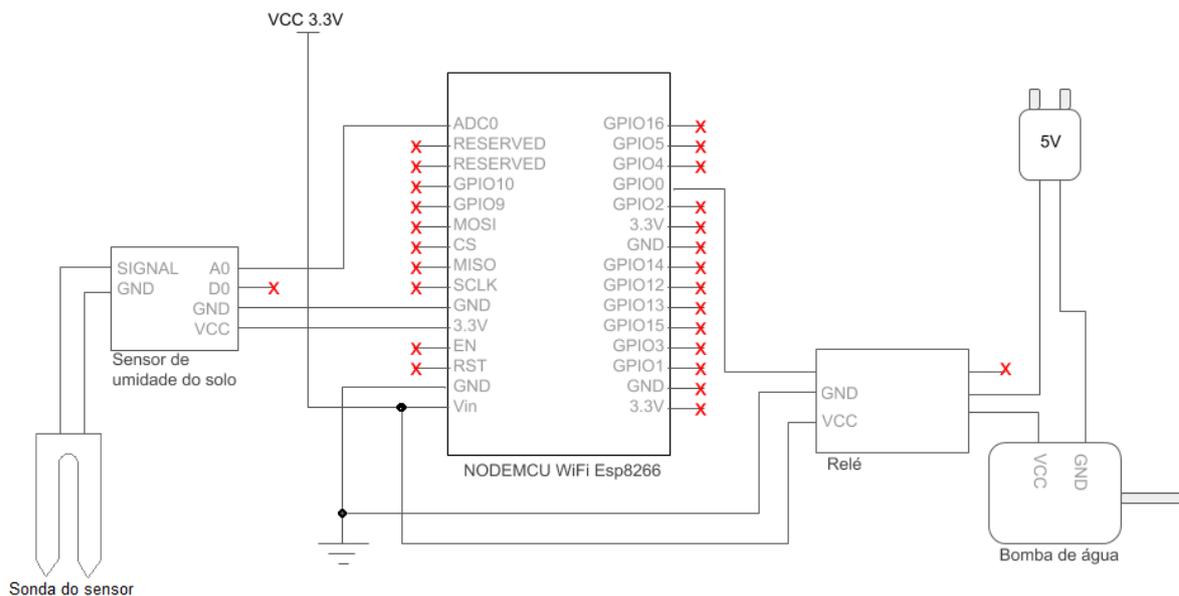


Figura 4.5: Esquemático do circuito elétrico da camada de hardware.

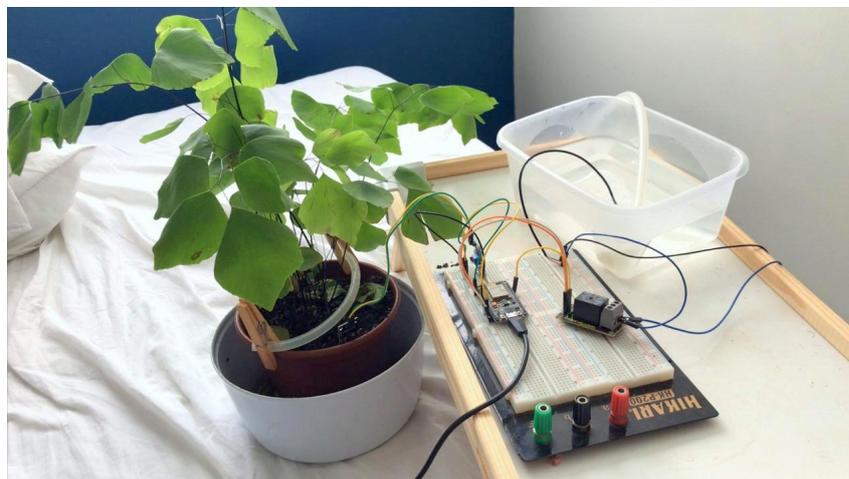


Figura 4.6: O sistema IoT conectado a uma planta.



Figura 4.7: O circuito elétrico montado na *protoboard*.

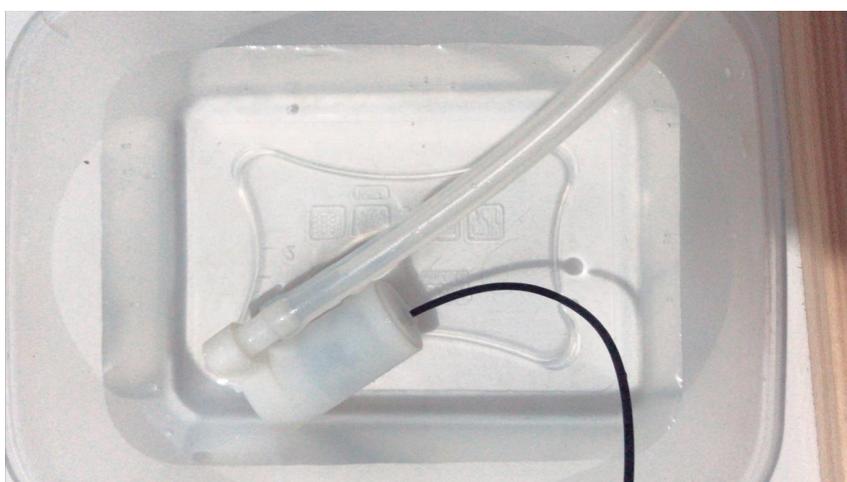


Figura 4.8: A bomba de água utilizada no sistema.

4.3.2 *Software embarcado*

O *software* embarcado no módulo NodeMCU foi desenvolvido na linguagem de programação C++ através da IDE do *Arduino*, mostrada na figura 4.11. Fez-se necessário incluir algumas bibliotecas para o funcionamento da aplicação, são elas: **ESP8266WiFi** e **FirebaseArduino**.

A biblioteca ESP8266WiFi, baseada no ESP8266 SDK e na biblioteca *Arduino Wifi Library*, possibilita a conexão WiFi a partir do nome e senha da rede. A biblioteca foi implementada em C++ e possui uma variedade de métodos e propriedades para configurar um ESP8266. Já a biblioteca FirebaseArduino, também desenvolvida em C++, foi utilizada para propiciar a conexão com o banco de dados *Firebase*. Atente-se na figura 4.11 para as duas constantes definidas no *header* do código, são elas: **FIREBASE_HOST** e **FIREBASE_AUTH**. A constante FIREBASE_HOST guarda a URL a ser acessada para o banco de dados, que é o endereço de *host*, e a constante FIREBASE_AUTH guarda uma *key* para solicitar autorização do cliente a

acessar o banco.



```

sketch | Arduino 1.8.11 (Windows Store 1.8.28.0)
Datei Bearbeiten Sketch Werkzeuge Hilfe

sketch

#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
// Set these to run example.
#define FIREBASE_HOST "irrigation-fe0dd.firebaseio.com"
#define FIREBASE_AUTH "huJMxcZEQF3hKasyoo9vcE1DkNV2jbjxQyhBLOWz"
#define WIFI_SSID "Nao realmente"
#define WIFI_PASSWORD "tacertol23"

#define PUBLISH_INTERVAL 1000

void setup() {
  Serial.begin(9600);

  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("conectando");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("conectado: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
void loop() {

```

Figura 4.9: *Arduino IDE*.

Após a conexão WiFi e a conexão com a *Firebase* estarem estabelecidas, as portas GPIO (*General Purpose Input/Output*) do NodeMCU se tornam ativas e permitem a comunicação com o sensor de umidade do solo.

Antes de enviar os dados do sensor para a *Firebase*, realiza-se uma calibração no valor recebido. Para calibrar o sensor, primeiramente realizou-se um teste: observou-se que o sensor em um ambiente totalmente seco, sem umidade, retornou valor 1024 (que representa 0%), enquanto que o sensor mergulhado em um ambiente totalmente úmido, ou seja, na própria água, retornou valor 361 (que representa 100%). Sendo assim, foi criada uma escala que varia de 361 a 1024 (isto é, de 100% a 0%). A escala é ilustrada na figura 4.10.

Da escala, foi possível extrair a seguinte equação:

$$C = \frac{100}{663}(1024 - S) \quad (4.1)$$

onde,

- **S**: o valor recebido pelo sensor de umidade do solo
- **C**: o resultado do valor do sensor após a calibração

Em seguida, o resultado da calibração é salva em um objeto do tipo JSON que é constituído pelo valor calibrado da umidade do solo e a hora e data da coleta do dado. Estes

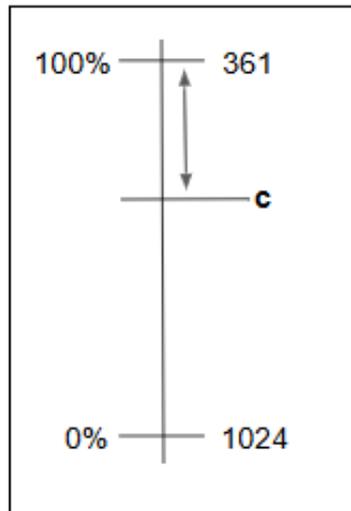


Figura 4.10: Escala utilizada para calibrar o sensor de umidade do solo.

```
sketch | Arduino 1.8.11 (Windows Store 1.8.28.0)
Datei Bearbeiten Sketch Werkzeuge Hilfe

sketch

#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
// Set these to run example.
#define FIREBASE_HOST "irrigation-fe0dd.firebaseio.com"
#define FIREBASE_AUTH "huJMxcZEQF3hKasyoo9vcE1DkNV2jbjxQyhBLOWZ"
#define WIFI_SSID "Nao realmente"
#define WIFI_PASSWORD "tacertol23"

#define PUBLISH_INTERVAL 1000

void setup() {
  Serial.begin(9600);

  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("conectando");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("conectado: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
```

Figura 4.11: *Arduino IDE*.

objetos JSON são enviados a cada 30 minutos para a *Firebase* por meio de uma função *push*. Para tanto, basta realizar uma assinatura com a *label* que se deseja acessar na *Firebase* (no caso deste sistema, a *label* de identificação é chamada **root**) e o objeto JSON.

4.4 Desenvolvimento do aplicativo *Android*

A criação desta aplicação tem o intuito de exibir aos usuários os dados captados pelos sensores, devidamente calibrados, por meio de uma interface gráfica simples e intuitiva que auxilie os usuários na tomada de decisão - se ele deve ou não regar a planta, por exemplo - ou então configurar o aplicativo para irrigar a planta automaticamente. Em caso de rega manual, o usuário receberá uma notificação alertando sobre a necessidade de regar a planta. Já na configuração de rega automatizada, a bomba (atuador) irá iniciar a rega assim que o sensor coletar um nível de umidade baixa.

O aplicativo é atualizado em tempo real com as informações recebidas dos sensores no ambiente monitorado e foi desenvolvido na plataforma *Android*. O desenvolvimento do *app* realizou-se em *Flutter*, o qual faz uso da linguagem *Dart*, e a plataforma de desenvolvimento utilizada foi o *Android Studio IDE*. Na sessão 4.2 deste capítulo foram esclarecidos os motivos de uso do framework *Flutter*.

A tela inicial do *app* exibe o dado mais recente do sensor de umidade do solo juntamente com a hora da coleta, e pode ser vista na figura 4.12. Note que a mensagem e o círculo de porcentagem de umidade do solo estão na cor verde, com o intuito de passar para o usuário que o sistema se encontra em um nível aceitável de umidade. Já a figura ?? mostra a mesma tela inicial quando a umidade do solo está baixa - observe que a cor do círculo e a mensagem são mostradas na cor vermelha, com o objetivo de alertar o usuário. É possível ainda ver o histórico do acionamento da bomba nas últimas horas, na figura 4.13, o qual mostra a variação da porcentagem da umidade do solo ao longo da variável tempo. Já a figura ??, informa ao usuário um histórico dos horários em que a bomba de água foi acionada e a porcentagem de umidade do solo no instante do acionamento. Em termos de configuração, o usuário pode dar ao sistema a autonomia de tomada de decisão, permitindo a rega automatizada quando identificado um nível baixo de umidade do solo. Para tanto, basta setar esta configuração na aba de Configurações, exibida na figura 4.14.

Os dados são atualizados no sistema a cada trinta minutos. Não é necessário a autenticação do usuário para acessar o sistema por não se tratar de dados sigilosos. Contudo, em caso de trabalhos futuros, é possível facilmente criar uma tela de *Login* que solicite a autenticação.

4.5 Comunicação com a nuvem

4.5.1 Envio de dados

O envio de dados para a *Firebase* é permitido após a configuração da *access key* citada nesta sessão. A figura 4.15 mostra como os dados são armazenados na nuvem, isto é, por meio de uma árvore na qual cada nó é um dado enviado pelo sensor. Os dados são salvos como uma

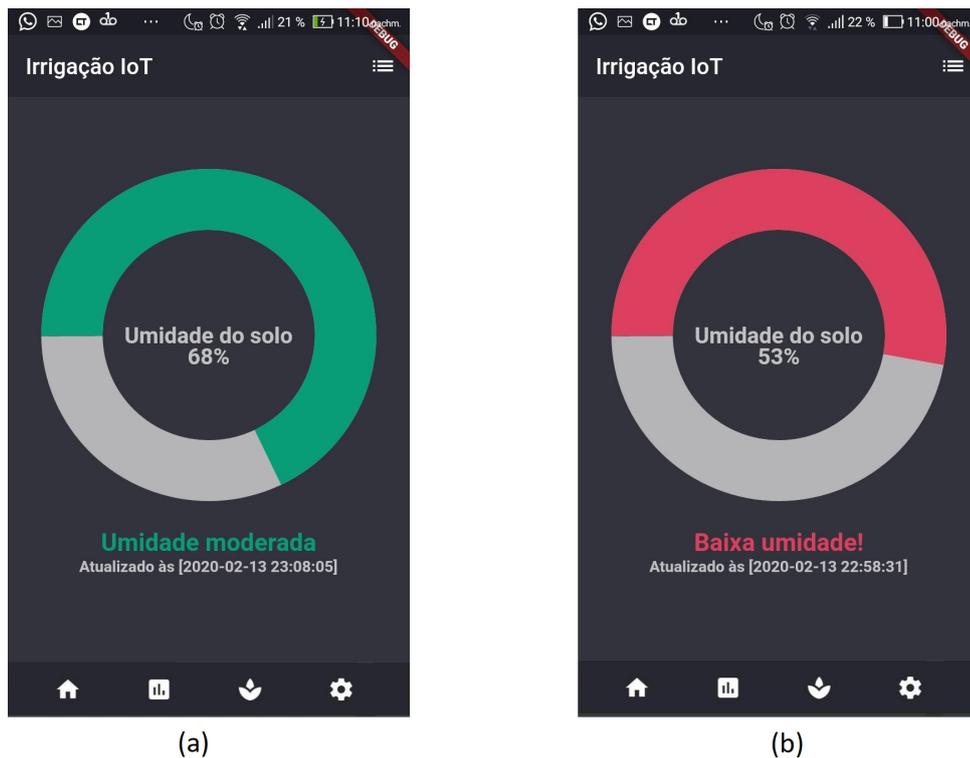


Figura 4.12: Tela principal do aplicativo *mobile*, que expressa a porcentagem em tempo real da umidade do solo. (a) mostra esta tela com umidade do solo moderada e (b) mostra a mesma tela com umidade do solo baixa.

árvore, em que os nós são objetos com atributos *humidity*, que mostra o dado calibrado da umidade do solo, e *time*, a hora da coleta do dado.

4.5.2 Consulta de dados

Para consultar os dados da nuvem é necessário configurar o aplicativo com um arquivo *google-services.json* gerado pelo *Firebase*. Após a configuração, o aplicativo já consegue se comunicar com a nuvem e já consegue ler os dados da *database*. Os dados também são recebidos em objeto JSON, como mostra a figura 4.16. Note que existem uma variável *humidity*, que é do tipo inteiro e armazena o valor da umidade do solo, e a variável *time*, que é do tipo *string* e armazena um formato de hora e data.

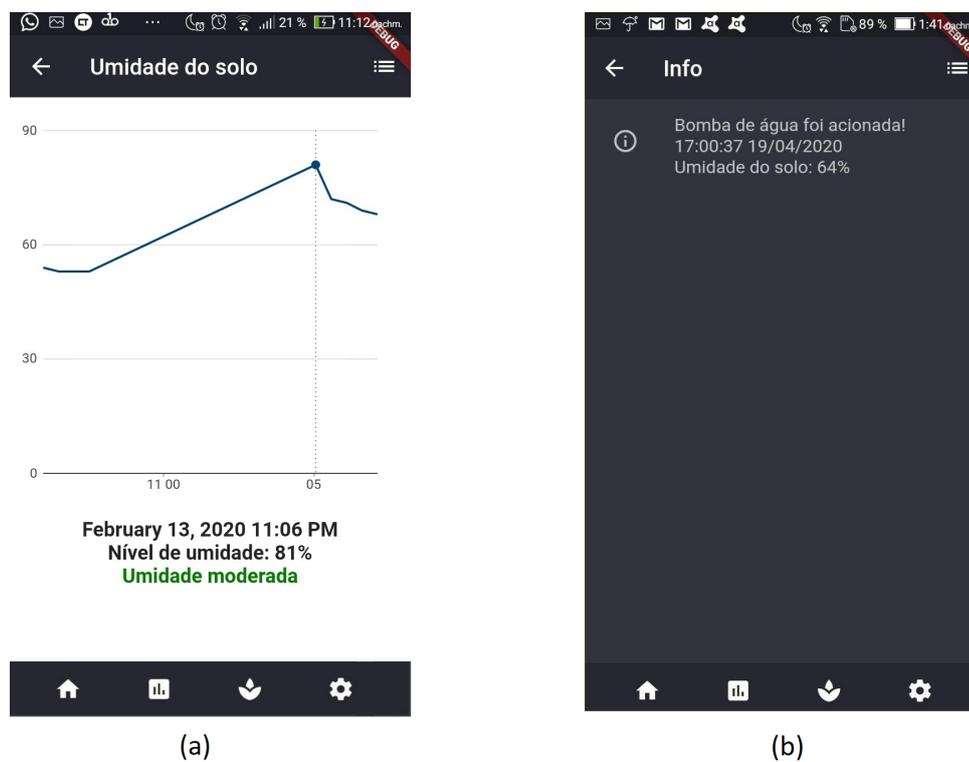


Figura 4.13: (a) Gráfico Umidade do solo (%) x Tempo (h). - (b) Tela de histórico que exhibe instante do acionamento de bomba de água.

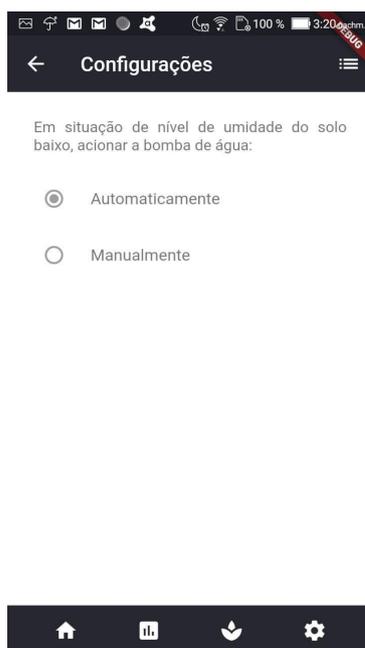


Figura 4.14: Tela para configurar rega manual ou automatizada.

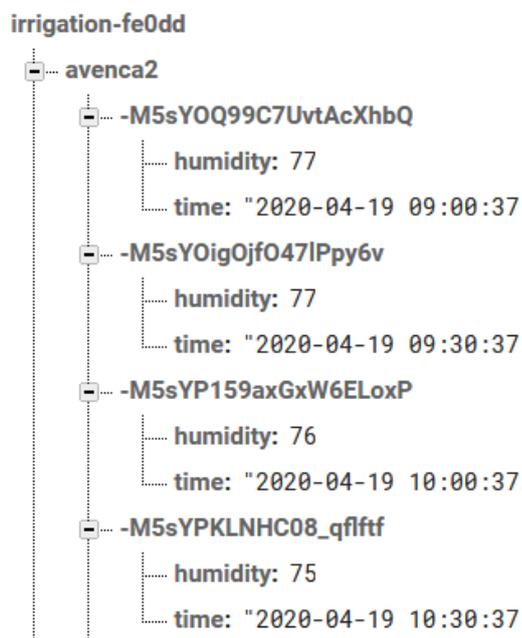


Figura 4.15: A estrutura do armazenamento de dados no *RealTime Database* da *Firebase*.

```
class Sensors {
    String key;
    int humidity;
    String time;
    int ldr;

    Sensors(this.key, this.humidity, this.time);

    Sensors.map(dynamic obj) {
        this.key = obj['key'];
        this.humidity = obj['humidity'];
        this.time = obj['time'];
    }

    String get v => humidity.toString();

    Sensors.fromSnapshot(DataSnapshot snapshot) {
        humidity = snapshot.value['humidity'].toInt();
        time = snapshot.value['time'];
    }
}
```

Figura 4.16: A classe *Sensors*, utilizada para receber os dados da *database*.

5

Resultados

Nesta seção serão exibidos os resultados em duas plantas, uma aveia e uma aveia variada, que têm como objetivo analisar se o sistema *IoT* obtém êxito em manter o nível de umidade do solo das referidas plantas em status "umidade moderada".

Os resultados foram obtidos através do aplicativo, o qual fornece um gráfico e, adicionalmente, um histórico de acionamento da bomba de água. O gráfico mostra a curva Umidade do Solo (%) x Tempo em um período de três dias - 72 horas. O histórico de acionamento da bomba exibe a data e hora em que a bomba foi ligada e em que porcentagem estava a umidade do solo naquele momento. Além de gráficos e históricos de monitoramento, serão dadas informações do volume de água utilizado nos três dias de rega para cada planta e o número de dados armazenados na nuvem neste período.

Para cada rega automática realizada, a bomba de água era ligada por 5 segundos, o que resultava em um uso de água de aproximadamente 120 mL. A necessidade de 120 mL de água foi descoberta por tentativa e erro através de um experimento realizado manualmente: em um momento de umidade do solo baixa, regou-se o solo da planta com 80, 100 e 120 mL de água e chegou-se à conclusão de que o volume de 120 mL era o mais apropriado para restaurar o nível da umidade para valores na faixa de 80%. O experimento foi realizado 3 vezes para cada planta, e o resultado foi o mesmo, obtendo valores de umidade entre 78% e 82% após a rega de 120 mL. Em seguida, testou-se com a bomba de água quantos segundos eram necessários deixá-la ligada para que o volume de água liberado se aproximasse dos 120 mL, o que mostrou ser 5 segundos.

Os gráficos foram gerados automaticamente no aplicativo através da biblioteca *Charts Flutter*, uma biblioteca desenvolvida em *Dart* para visualização de dados através de gráficos de variados tipos, tais como, gráfico de barras horizontal e vertical, gráfico pizza, gráfico linha, e muitos outros. Na tela que exibe o gráfico também é exibida a data e hora da última coleta e o nível corrente da umidade do solo.

Também é importante ressaltar que o aplicativo de monitoramento estava configurado para rega automática, isto é, sem intervenção humana. O controle da umidade do solo ficou por conta do sistema *IoT* durante todo o tempo para as duas plantas, com acionamento automático quando o nível de umidade do solo fosse abaixo de 65%.

Segundo (Samambaias Lindas, 2019), o solo de uma planta avenca deve estar sempre úmido, regado com moderação, e jamais encharcado, pois pode apodrecer as raízes. Sendo assim, observaram-se as condições do solo juntamente com as leituras do sensor de umidade do solo e concluiu-se que:

- o sensor de umidade do solo coletou medidas acima de 88% na iminência de solo encharcado;
- o sensor de umidade do solo coletou medidas abaixo de 65% na iminência de solo seco.

Sendo assim, por questão de segurança, a classificação de umidade foi convencionalizada da seguinte forma:

- percentuais entre 0% e 65% são considerados como baixa umidade do solo;
- percentuais entre 66% e 90% são considerados como umidade do solo moderada;
- percentuais entre 91% e 100% são considerados como umidade do solo elevada.

Procura-se, neste sistema, manter a planta em níveis de umidade do solo moderada, pois se tratando de avencas é importante irrigá-las de maneira eficaz para que suas raízes não apodreçam com a rega excessiva e tampouco fiquem secas e morram com a ausência de água.

5.1 O Ambiente

As plantas escolhidas para testar este sistema possuem características em comum, dentre elas, a necessidade de um ambiente que possua luz indireta (pede-se que evite a luz solar diretamente na planta) e que seja moderadamente ventilado. Dessa forma, escolheu-se um local com boa iluminação e ventilação.

5.2 Resultados do monitoramento

Os dados do sistema foram coletados para cada planta durante três dias, iniciando a coleta exatamente às 9h da manhã. O sistema foi executado para a planta avenca da data 15/04/2020 até 18/04/2020 e para a planta avenca variada da data 19/04/2020 até 22/04/2020. Não houve problema de conexão de internet e o sistema se comportou de maneira esperada durante o período de monitoramento das plantas.

5.2.1 Avenca

A avenca é uma planta nativa brasileira, um tipo de samambaia, e pode chegar a uma altura de até 70 cm. O cultivo desta planta necessita de muita luz, contudo a luz solar deve incidir indiretamente sobre ela (Cultura Mix, 2011). A temperatura indicada deve estar entre 10°C e 30°C, sendo que em dias quentes ela deve ser regada com uma maior frequência já que ela necessita de umidade. No caso de uso da avenca para este sistema, ela esteve em um ambiente com temperatura aceitável. Mesmo em dias quentes com temperatura elevada, a ventilação diminuía o seu impacto.

Esta planta foi escolhida para monitoramento pois necessita estar frequentemente úmida, o que justifica o uso de um sistema *IoT* para controlar o seu nível de umidade do solo. Além disso, esta planta também foi escolhida devido a adaptação do seu cultivo no ambiente descrito acima.

A figura 5.1 exibe a avenca no ambiente já descrito. Observe o circuito elétrico com o sensor de umidade do solo conectado ao NodeMCU e também a sua outra parte cravada no vegetal. Adicionalmente, note a bomba de água em um pequeno recipiente e uma mangueira que liga a bomba à planta. A mangueira, por sua vez, possui furos na região onde está a avenca, para que possa irrigá-la quando a bomba for acionada.



Figura 5.1: Sistema *IoT* de rega automatizada instalado em uma planta avenca.

O monitoramento se iniciou às 9h da manhã do dia 15/04/2020, com a primeira coleta de umidade do solo com valor de 85% como mostra o gráfico 5.2 gerado dinamicamente no aplicativo. A coleta era feita a cada trinta minutos, sendo possível acompanhar no *app* o

gráfico tomando forma à medida que o tempo se passava. Após os três dias de funcionamento do sistema, observou-se que o mesmo se comportara de maneira esperada, sempre elevando o nível de umidade do solo quando identificado um nível inaceitável para a planta. A figura 5.2 exibe o histórico do acionamento da bomba, isto é, o nível de umidade do solo, a hora e a data em que o percentual baixo foi identificado e acionou-se a bomba. O sistema foi desligado às 9h do dia 18/04/2020, isto é, após completar 72h.

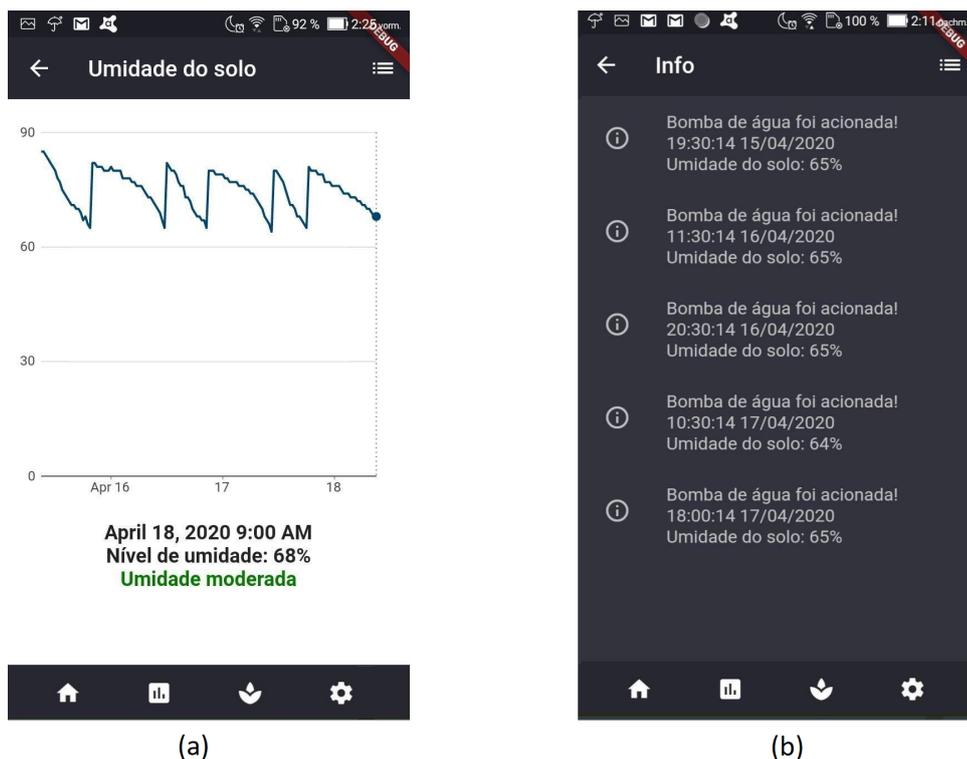


Figura 5.2: (a) Resultado do gráfico Umidade do solo x Tempo da planta aveia em um período de 72h. (b) Histórico de acionamento de bomba de água em planta aveia.

Notou-se que o sistema *IoT* respondia de maneira eficaz quando o nível de umidade do solo estava baixo, com o menor valor de umidade coletado em 64%. Na próxima coleta já se via que a umidade do solo havia sido restaurada para um valor considerado moderado - entre 79% e 83%. Também foi possível observar graficamente em 5.2 que a umidade do solo diminuía mais rápido durante o dia do que durante a noite, devido à variação de temperatura do ambiente.

Pode-se notar o que foi citado no parágrafo acima ao tomar como exemplo a data 16/04/2020, considerando-se o período da tarde entre o pico de umidade moderada coletada em 82% às 12:00 e o pico de umidade baixa de 65% às 20:30, tem-se, portanto, um total de oito horas e meia até a bomba ter sido novamente acionada. Agora tomando o período entre o pico de umidade alta coletada em 80% às 21:00 e o pico de umidade baixa de 64% às 10:30, tem-se um período de treze horas e meia até o novo acionamento. Além de ser possível notar isso graficamente, as informações do histórico na figura 5.2 permitem confirmar

os fatos.

No aplicativo é possível acompanhar toda a curva, deslizando o dedo por ela, e abaixo as informações de data, hora e porcentagem de umidade do solo são exibidas. O pico mais alto do monitoramento da aveca após o primeiro acionamento da bomba foi 82%, enquanto que o pico mais baixo foi 64%, já mostrado em 5.2.

O volume de água utilizado neste período de três dias para irrigar a planta foi de aproximada 600 mL, visto que a bomba foi acionada cinco vezes.

5.2.2 Aveca variada

A aveca variada, como o nome já diz, é uma variação da aveca e também é chamada como "aveca gigante". Ainda assim, a aveca variada possui as mesmas características de cultivo da aveca tradicional - tais como luz indireta, temperatura entre 10°C e 30°C, etc. Portanto, as mesmas recomendações da aveca tradicional foram tomadas para a aveca variada.

A figura 5.3 exibe a aveca variada no ambiente já descrito. Observe o circuito elétrico com o sensor de umidade do solo conectado ao NodeMCU e também a bomba de água com uma mangueira que liga a bomba à planta.

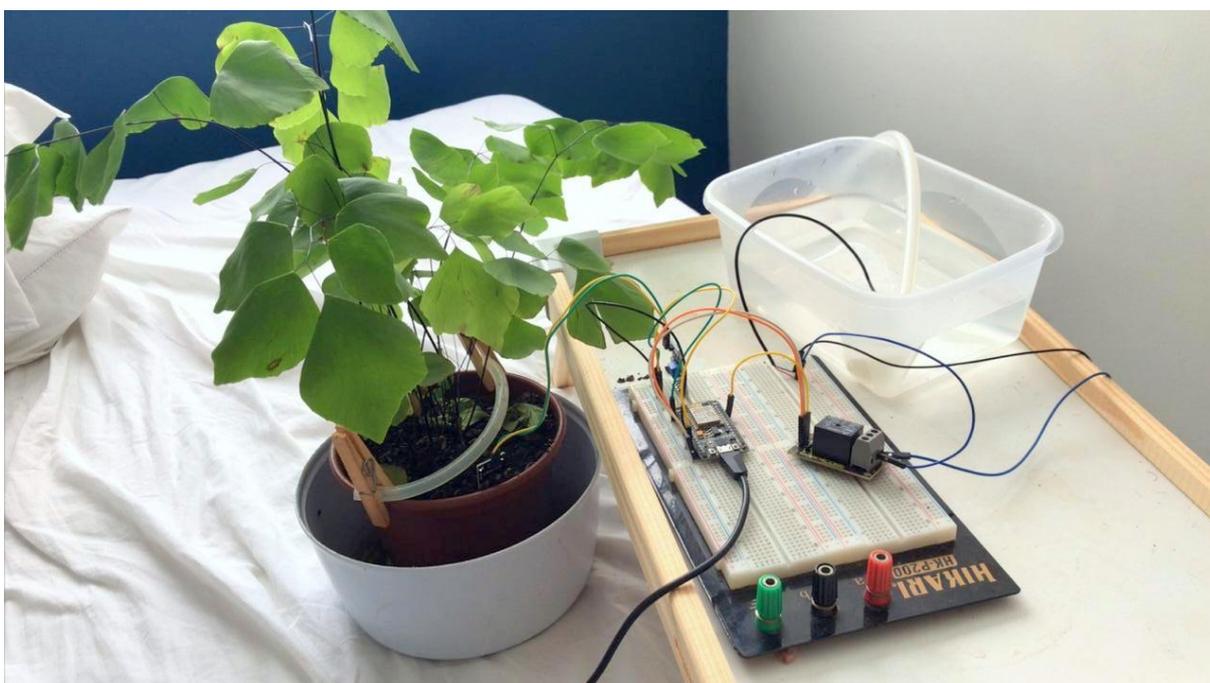


Figura 5.3: Sistema *IoT* de rega automatizada instalado em uma aveca variada.

O monitoramento se iniciou às 9h da manhã do dia 19/04/2020, com a primeira coleta de umidade do solo com valor de 79% como mostra o gráfico 5.4. O sistema se comportou de maneira esperada durante os três dias de funcionamento. A tabela exibe o nível de umidade do solo, a hora e a data em que o percentual baixo foi identificado. O sistema foi desligado às 9h do dia 22/04/2020, isto é, após completar 72h.

Em momentos de baixa umidade do solo, com menor valor de umidade no valor de 64%, o sistema sempre se comportou de maneira eficaz, acionando a bomba. Na próxima coleta já se via que a umidade do solo havia sido restaurada para um valor considerado moderado - entre 79% e 83%. Também foi possível observar graficamente em 5.4 que, assim como a planta avenca, a umidade do solo da avenca variada também diminuía mais rápido durante o dia do que durante a noite, devido à variação de temperatura no ambiente. Em 20/04/2020, tomando-se o período entre o pico de umidade moderada de 81% coletado às 11:00 e o pico de umidade baixa de 65% às 20:30, tem-se, portanto, um total de nove horas e meia até que a bomba se acionou. Agora considerando o período entre o pico de umidade alta coletada em 80% às 21:00 e o pico de umidade baixa de 64% às 12:00, tem-se um período de quinze horas até que a bomba de água foi acionada. Além de ser possível notar graficamente, as informações da figura 5.4 permitem confirmar os fatos, basta verificar o tempo entre dois acionamentos da bomba de água.

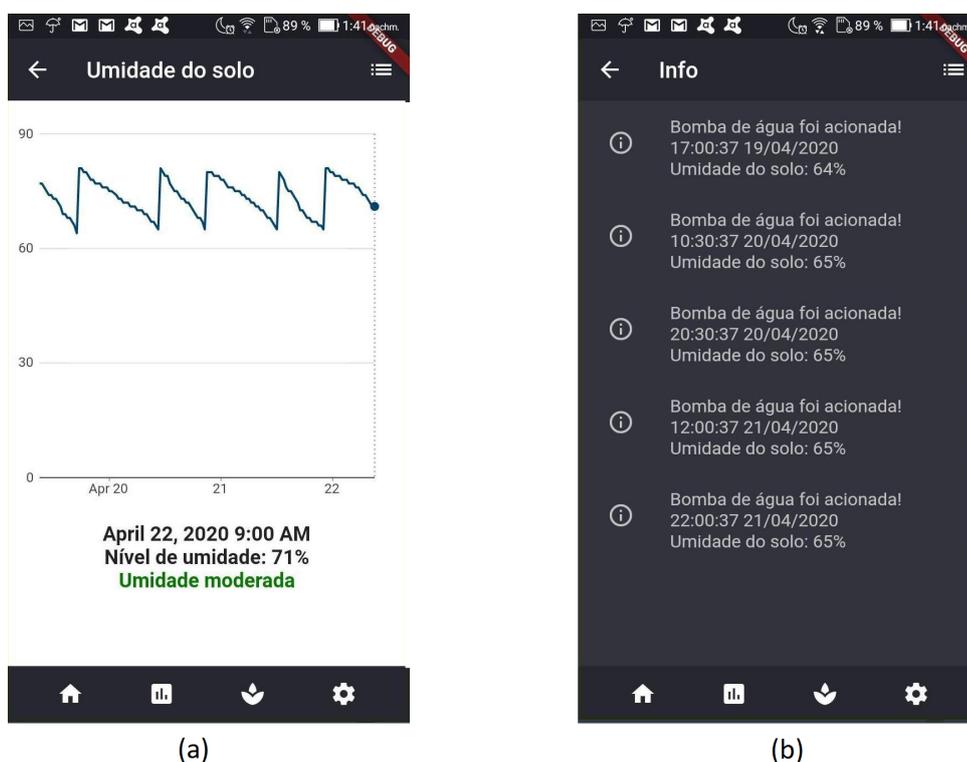


Figura 5.4: (a) Resultado do gráfico Umidade do solo x Tempo da planta avenca variada em um período de 72h. (b) Histórico de acionamento de bomba de água em planta avenca variada.

Como já explicado, no aplicativo é possível acompanhar os valores na curva da figura 5.4, deslizando o dedo por ela, bem como as informações de data, hora e porcentagem de umidade do solo são exibidas. O pico mais alto do monitoramento da avenca após o primeiro acionamento da bomba foi 81%, enquanto que o pico mais baixo foi 64%.

O volume de água utilizado neste período de três dias para irrigar a planta foi de 600 mL, pois a bomba também foi acionada cinco vezes.

6

Conclusão e trabalhos futuros

Este trabalho abordou o desenvolvimento de um sistema *IoT* de rega assistida e automatizada que pode ser usado em diversos contextos, tais como nas escolas, para desenvolver em crianças a percepção de quando uma planta (de variados tipos, tais como, hidrófita, higrófito, mesófito e xerófito) precisa ou não de água, quanto também em adultos atarefados que precisam ser lembrados de irrigar (por meio de notificações no *smartphone*) ou que preferem configurar o sistema para irrigar automaticamente por estarem viajando e desejarem cuidar e monitorar suas plantas à distância.

O sistema se vê muito útil para crianças, pois não basta dizê-las que precisam cuidar do meio ambiente, ou estimulá-las a fazer um jardim, sem darmos ferramentas intelectuais a elas para auxiliá-las a cultivar plantas. O sistema possui avisos baseado em cores, o que auxilia muito crianças e até mesmo adultos - quando a umidade está baixa, a mensagem de umidade baixa e o círculo da porcentagem aparecem na cor vermelha; quando a umidade está moderada, o sistema exibe uma mensagem e círculo de porcentagem na cor verde; e quando a umidade está alta, a mensagem e o círculo são mostrados na cor amarela. Sem este monitoramento, as crianças que não obtiverem êxito no cultivo podem ficar frustradas por notar que seu trabalho foi em vão.

O sistema proposto poderia ser adaptado para receber as especificação de cada tipo de planta em função da sua classificação e do tipo de solo onde se encontra. Como mostrado, o sistema foi testado em duas plantas reais, para a qual os parâmetros de umidade do solo foram colocados diretamente no programa.

A principal dificuldade encontrada durante esse trabalho foi adaptar a biblioteca *Flutter Charts* para gerar os gráficos dinamicamente a partir dos dados inseridos também dinamicamente na *Firebase Realtime Database*.

Espera-se que, em trabalhos futuros, o módulo de aplicativo *mobile* permita a configuração para diferentes tipos de planta e solos, além da adição de mais sensores, tais como um LDR para medir o nível de luminosidade incidente sobre a planta, um sensor de umidade do

ar e um sensor de temperatura. Outro caminho a ser investigado neste sistema é acrescentar novas capacidades para o mesmo em função do conteúdo escolar de geografia, biologia e química, onde poderia ser trabalhado a questão dos fertilizantes em hortas, de doenças que podem prejudicar determinada planta, dentre outros. E por fim, no contexto de inteligência artificial, este sistema poderia estar conectado a uma base de dados meteorológica que informaria a previsão de chuva no local e utilizaria esses dados para auxiliar na tomada de decisão de irrigação de solo seco ou não (podendo irrigar um pouco e aguardar a chuva, ou simplesmente não irrigar e aguardar).

Referências Bibliográficas

- Android Studio Website. Conheça o android studio, 2018. URL <https://developer.android.com/studio/intro/?hl=pt-br>. Data de acesso: dez. 2019.
- Arduino Website. Arduino 1.8.6, 2018. URL <https://www.arduino.cc/en/Main/Software>. Data de acesso: dez. 2019.
- Arduino Website. Arduino - introduction, 2019. URL <https://www.arduino.cc/en/guide/introduction>. Data de acesso: mar. 2020.
- K. Ashton. That 'internet of things' thing. *RFID Journal*, 2009.
- Banggood. Geekcreit nodemcu lua esp8266 placa de desenvolvimento esp-12e wifi, 2014. URL https://br.banggood.com/Geekcreit-Doit-NodeMcu-Lua-ESP8266-ESP-12F-WIFI-Development-Board-p-985891.html?cur_warehouse=USA. Data de acesso: fev. 2020.
- S. Bhardwaj, L. Jain, and S. Jain. Cloud computing: A study of infrastructure as a service (iaas). *International Journal of Engineering and Information Technology*, 2010.
- A. Bhatt and J. Patoliya. Cost effective digitization of home appliances for home automation with low-power wifi devices. *International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB16)*, 2016.
- Rachel Cardell-Oliver, Mark Kranz, Keith Smettem, and Kevin Mayer. A reactive soil moisture sensor network: Design and field evaluation. *International journal of distributed sensor networks*, 1(2):149–162, 2005.
- L. Corral, A. Janel, and T. Remencius. Potential advantages and disadvantages of multiplatform development frameworks-a vision on mobile environments. *Procedia Computer Science*, 1:1202–1207, 2012.
- Cultura Mix. Avenca, 2011. URL <https://flores.culturamix.com/flores/avenca>. Data de acesso: mar. 2020.

- Edward Curry, Souleiman Hasan, Christos Kouroupetroglou, Willem Fabritius, Umair ul Hassan, and Wassim Derguech. Internet of things enhanced user experience for smart water and energy management. *IEEE Internet Computing*, 22(1):18–28, 2018.
- I. Dalmasso, S. Datta, C. Bonnet, and N. Nikaein. Survey, comparison and evaluation of cross platform mobile application development tools. In IEEE Xplore press, editor, *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 323–328, 2012.
- C. Kouroupetroglou W. Fabritius U. Hassan E. Curry, S. Hasan and W. Derguech. Internet of things enhanced user experience for smart water and energy management. 22:18 – 28, 2018.
- Espressif. Esp8266 power consumption, 2015. URL <https://bbs.espressif.com/viewtopic.php?t=133>. Data de acesso: mar. 2020.
- Filipe Flop. Sensor de umidade e temperatura dht11, 2014. URL <https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-dht11/>. Data de acesso: mar. 2020.
- Firestore Website. Página principal da firestore, 2010. URL <https://firebase.google.com/?hl=pt-br>. Data de acesso: jan. 2020.
- N. Freitas and F. Marin. Educação ambiental e água: concepções e práticas educativas em escolas municipais. 2015.
- Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.
- Judika Herianto Gultom, Maruf Harsono, Tubagus Dhika Khameswara, and Handri Santoso. Smart iot water sprinkle and monitoring system for chili plant. In *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 212–216. IEEE, 2017.
- Kutilla Gunasekera, Armando Navas Borrero, Fabian Vasuian, and Kim P Bryceson. Experiences in building an iot infrastructure for agriculture education. *Procedia Computer Science*, 135:155–162, 2018.
- G. Bertolini J. Tugoz and L. Brandalise. Captação e aproveitamento da Água das chuvas: O caminho para uma escola sustentável. 2017.
- S. Shah S. Delwadkar K. Kansara, V. Zaveri and K. Jani. Sensor based automated irrigation system with iot: A technical review. *International Journal of Computer Science and Information Technologies*, 6(1):5331–5333, 2015.

- Carlos Kamienski, Juha-Pekka Soininen, Markus Taumberger, Ramide Dantas, Attilio Toscano, Tullio Salmon Cinotti, Rodrigo Filev Maia, and André Torre Neto. Smart water management platform: Iot-based precision irrigation for agriculture. *Sensors*, 19(2):276, 2019.
- N. Kolban. Kolban's book on esp8266, an introductory book on esp8266. 2015.
- Mark G Lawrence. The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications. *Bulletin of the American Meteorological Society*, 86(2):225–234, 2005.
- G. Lawton. Developing software online with platform-as-a-service technology. *Computer*, 6(41):15–20, 2008.
- M. LTD. Apple vs android - a comparative study 2017., mar 2017. URL <https://android.jlelse.eu/apple-vs-android-a-comparative-study-2017-c5799a0a1683>.
- M. Lima. Brasil já tem mais de um smarphone ativo por habitante, diz estudo da fgv, 2018. URL <https://link.estadao.com.br/noticias/geral,brasil-ja-tem-mais-de-um-smartphone-ativo-por-habitante-diz-estudo-da-fgv,70002275238>. Data de acesso: mar. 2020.
- M. Rouse. Microcontroller (mcu), 2016. URL <https://internetofthingsagenda.techtarget.com/definition/microcontroller>. Data de acesso: mar. 2020.
- James Manyika. *The Internet of Things: Mapping the value beyond the hype*. McKinsey Global Institute, 2015.
- J. Morgan. *A Simple Explanation Of 'The Internet Of Things'*. Elsevier, 2014.
- E. Oliveira N. Melo, M. Salla and V. Frasson. Consumo de água e percepção dos usuários sobre o uso racional de água em escolas estaduais do triângulo mineiro. 2015.
- I. Orikumhi C. Y. Leow O. Elijah, T. A. Rahman and M. N. Hindia. An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*, 2018.
- T. A. Rahman S. A. Babale O. Elijah, I. Orikumhi and S. I. Orakwue. Enabling smart agriculture in nigeria: Application of iot and data analytics. *Proc. IEEE 3rd Int. Conf. Electro Technol. Nat. Develop. (NIGERCON)*, 2017.
- André Presse. Is water really a scarce resource? initiating entrepreneurship for global clean water supply. In *Conflict Resolution in Water Resources and Environmental Management*, pages 111–131. Springer, 2015.

- P. Rajalakshmi and S. D. Mahalakshmi. Iot based crop-field monitoring and irrigation automation. *IEEE*, 2016.
- RSSF UFRJ. Redes de sensores sem fio, 2018. URL <https://www.gta.ufrj.br/ensino/eel878/redes1-2018-1/trabalhos-vf/rssf/>. Data de acesso: mar. 2020.
- Stuart Russel, Peter Norvig, et al. *Artificial intelligence: a modern approach*. Pearson Education Limited, 2013.
- Samambaias Lindas. Avenca-como cuidar passo a passo [simples e rápido], 2019. URL <https://samambaiaslindas.com/avenca-como-cuidar/>. Data de acesso: jan. 2020.
- J. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé. Vision and challenges for realising the internet of things. *CERP IoT*, 2010.
- É. Marques T. Teixeira and J. Pereira. Educação ambiental em escolas públicas: caminho para adultos mais conscientes. 2017.
- TechTudo. O que é um arduino e o que pode ser feito com ele?, 2013. URL <https://www.techtudo.com.br/noticias/noticia/2013/10/o-que-e-um-arduino-e-o-que-pode-ser-feito-com-ele.html>. Data de acesso: fev. 2020.
- F. TongKe. Smart agriculture based on cloud computing and iot. *Journal of Convergence Information Technology(JCIT)*, 8, 2013.
- Universidade USP. Conceitos fundamentais: Grandezas luminosas, 2018. URL http://www.fau.usp.br/arquivos/disciplinas/au/aut0213/Material_de_Apoio/03_-_Ia._Conceito_Fundamentais_%28grandezas_Luminosas%29.pdf. Data de acesso: mar. 2020.
- Fabrizio Valpreda and Ilaria Zonda. Grüt: A gardening sensor kit for children. *Sensors*, 16(2): 231, 2016.
- Vida de Silício. Sensor de umidade do solo - higrômetro, 2014. URL <https://www.vidadesilicio.com.br/sensor-umidade-solo-higrometro>. Data de acesso: mar. 2020.
- SalesForce Website. Novo em saas? bem-vindo!, 2018. URL <https://www.salesforce.com/br/saas/>. Data de acesso: nov. 2019.
- A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet Things Journal*, 1:22–32, feb 2014.

Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: State-of-the-art and research challenges. *J. Internet Sert. Appl.*, 1(1):7–18, may 2010.