



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
FACULDADE DE TECNOLOGIA
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

SUMAÚMA: ESTAÇÃO DE MONITORAMENTO DE
PLANTAS DOMÉSTICAS UTILIZANDO ANDROID
EMBARCADO E ESP32

ARIEL LUANE PEREIRA BENTES

Manaus - AM
JUNHO DE 2023

ARIEL LUANE PEREIRA BENTES

SUMAÚMA: ESTAÇÃO DE MONITORAMENTO DE
PLANTAS DOMÉSTICAS UTILIZANDO ANDROID
EMBARCADO E ESP32

Monografia de Graduação apresentada a Faculdade de Tecnologia da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de bacharel em Engenharia da Computação.

Orientador(a)

EDUARDO JAMES PEREIRA SOUTO, Dr.

Universidade Federal do Amazonas - UFAM

FACULDADE DE TECNOLOGIA

Manaus - AM

JUNHO DE 2023

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

B475s Bentes, Ariel Luane Pereira
Sumaúma: estação de monitoramento de plantas domésticas
utilizando android e esp32 / Ariel Luane Pereira Bentes . 2023
45 f.: il. color; 31 cm.

Orientador: Eduardo James Pereira Souto
TCC de Graduação (Engenharia da Computação) - Universidade
Federal do Amazonas.

1. Internet das Coisas. 2. Android. 3. Esp32. 4. Monitoramento. I.
Souto, Eduardo James Pereira. II. Universidade Federal do
Amazonas III. Título

Monografia de Graduação sob o título *SUMAÚMA: ESTAÇÃO DE MONITORAMENTO DE PLANTAS DOMÉSTICAS UTILIZANDO ANDROID EMBARCADO E ESP32* apresentada por Ariel Luane Pereira Bentes e aceita pela Faculdade de Tecnologia da Universidade Federal do Amazonas, sendo aprovada por todos os membros da banca examinadora abaixo especificada:




Prof. Dr. Eduardo James Pereira Souto

Orientador(a)

Instituto de Computação - Icomp

Universidade Federal do Amazonas



Prof. Dr. Raimundo da Silva Barreto

Instituto de Computação - Icomp

Universidade Federal do Amazonas



Prof. Dr. Eduardo Luzeiro Feitosa

Instituto de Computação

Universidade Federal do Amazonas

Manaus - AM, 30 de Junho de 2023.

Dedico este trabalho aos meus pais e agradeço pelo apoio incondicional em todos os momentos difíceis da minha trajetória acadêmica.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer à minha família, que esteve ao meu lado durante todo o percurso. A vocês pais, Janete Vieira e Luiz Moraes, meus padrinhos Andréa Bentes e Erivan Glaucio e demais familiares, minha gratidão por serem minha base sólida, meu porto seguro. Obrigado por acreditarem em mim, me apoiarem incondicionalmente e me motivarem a buscar sempre o melhor. Seu amor e suporte foram essenciais para minha trajetória acadêmica.

Ao meu noivo Anderson Santos, agradeço por ser uma presença constante e por me apoiar em todos os aspectos da minha vida. Sua compreensão, paciência e incentivo foram cruciais para que eu pudesse me dedicar a este trabalho. Obrigado por me dar forças, acreditar em mim e compartilhar minha alegria e frustrações ao longo dessa jornada. Sua presença é meu conforto e inspiração.

Aos meus amigos, quero expressar minha imensa gratidão. Vocês estiveram presentes nos momentos mais desafiadores e nas conquistas mais significativas. Obrigado por serem minha fonte de alegria, descontração e apoio. Vocês me deram forças quando eu mais precisei e tornaram essa jornada muito mais memorável. Suas palavras de encorajamento, apoio e as risadas compartilhadas fizeram toda a diferença.

Também gostaria de agradecer aos meus orientadores e professores, cuja orientação e conhecimento foram fundamentais para o desenvolvimento deste trabalho. Obrigado por compartilharem sua sabedorias e por investirem seu tempo e dedicação em minha formação. Suas contribuições foram essenciais para a conclusão deste projeto.

Por fim, gostaria de expressar meu agradecimento a todos aqueles que, de alguma forma, contribuíram para o meu crescimento acadêmico e pessoal ao longo

dessa jornada. Seja através de palavras de encorajamento, discussões enriquecedoras, debates em sala de aula ou qualquer forma de suporte, seu impacto em minha vida é significativo e digno de gratidão.

A todos vocês, minha família, amigos e meu noivo, agradeço do fundo do coração. Vocês são parte fundamental do meu caminho e sua presença torna minha vida mais rica e significativa. Sem vocês, essa conquista não seria possível. Obrigado por fazerem parte da minha história e por serem meu apoio constante.

Com sincera gratidão,

Ariel Luane Pereira Bentes

Todas as vitórias ocultam uma abdicação.

Simone de Beauvoir

SUMAÚMA: ESTAÇÃO DE MONITORAMENTO DE PLANTAS DOMÉSTICAS UTILIZANDO ANDROID EMBARCADO E ESP32

Autor: ARIEL LUANE PEREIRA BENTES

Orientador: EDUARDO JAMES PEREIRA SOUTO, Dr.

Resumo

O monitoramento de plantas domésticas por meio de aplicativos em dispositivos Android tem se tornado uma tendência crescente entre os entusiastas de jardinagem. Esses aplicativos oferecem uma solução prática e conveniente para auxiliar no cuidado das plantas, combinando a paixão pela natureza com os avanços da tecnologia móvel. Este trabalho apresenta o estágio inicial de desenvolvimento de uma ferramenta que tem como objetivo facilitar o controle, monitoramento e automação do cultivo de plantas em residências. Neste contexto, foi desenvolvida uma aplicação móvel para Android, que permite a coleta em tempo real de dados provenientes de sensores de um dispositivo embarcado (*Esp32*) associado a uma planta. Esses sensores monitoram a luminosidade, umidade, temperatura e umidade do solo. A aplicação Android exibe de forma simples e intuitiva os dados coletados, possibilitando aos usuários visualizá-los e analisá-los.

Palavras-chave: Internet das Coisas, Android, Esp32.

SUMAÚMA: ESTAÇÃO DE MONITORAMENTO DE PLANTAS DOMÉSTICAS UTILIZANDO ANDROID EMBARCADO E ESP32

Autor: ARIEL LUANE PEREIRA BENTES

Orientador: EDUARDO JAMES PEREIRA SOUTO, Dr.

Abstract

Monitoring of houseplants through apps on Android devices has become a growing trend among gardening enthusiasts. These applications offer a practical and convenient solution to assist in plant care, combining a passion for nature with advances in mobile technology. This work presents the initial stage of development of a tool aimed at facilitating control, monitoring, and automation of plant cultivation in residences. In this context, a mobile application for Android was developed, which allows real-time data collection from sensors of an embedded device (*Esp32*) associated with a plant. These sensors monitor brightness, humidity, temperature, and soil moisture. The Android application displays the collected data in a simple and intuitive manner, enabling users to visualize and analyze them.

Keywords: Internet of Things, Android, Esp32.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ambiente de Desenvolvimento Android.	21
Figura 2 – Estrutura do AOSP.	22
Figura 3 – Prototipo idealizado no início do projeto.	24
Figura 4 – Microcontrolador - Esp 32.	27
Figura 5 – Sensor DHT11.	28
Figura 6 – Sensor de Luminosidade - LDR.	28
Figura 7 – Sensor de Umidade do Solo.	30
Figura 8 – Dispositivo de Monitoramento de Plantas com Esp32.	33
Figura 9 – Codificação no Arduino IDE.	34
Figura 10 – Desenvolvimento da HAL.	36
Figura 11 – Desenvolvimento da Manager.	36
Figura 12 – Desenvolvimento do Driver.	36
Figura 13 – Desenvolvimento da Biblioteca.	37
Figura 14 – Aplicação da Biblioteca.	37
Figura 15 – Primeira Tela Desenvolvida em Ambiente Android.	38
Figura 16 – Visão Geral das Camadas.	39
Figura 17 – Dispositivo Final em Execução.	40
Figura 18 – Solo igual a 0 - O solo precisa de água.	40
Figura 19 – Umidade do solo abaixo de 100 - O solo precisa de atenção.	41
Figura 20 – Umidade do solo acima de 100 - Tudo certo.	41

LISTA DE ABREVIATURAS E SIGLAS

AOSP *Android Open Source Project*)

API *Application Programming Interface*

Esp32 *é um microcontrolador desenvolvido pela empresa chinesa Espressif Systems.*

HAL *Hardware Abstraction Layer*)

HTTP *HyperTextTransferProtocol*

IoT *Internet Of Things*

MQTT *Message Queue Telemetry Transport*

MVC *Model-View-Controller*)

NodeMCU *é um firmware de código aberto, para o qual projetos abertos de placas de prototipação estão disponíveis.*

PIR *Parent in Room*

RFID *Radio Frequency Identification*

ROM *Read-Only Memory*

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contextualização e Definição do Problema	14
1.2	Objetivos	15
1.3	Organização do Documento	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Internet das Coisas - <i>IoT</i>	18
2.2	Android	19
2.2.1	ANDROID OPEN SOURCE PROJECT - AOSP	21
2.3	Automação - Utilizando o ESP	22
3	ESTAÇÃO DE MONITORAMENTO DE PLANTAS	24
3.1	Visão Geral do Protótipo	24
3.2	Definição dos Componentes	25
3.2.1	ESP32:	25
3.2.2	Sensores	27
3.2.2.1	Sensor de Umidade do ar e Temperatura	27
3.2.2.2	Sensor de Luminosidade	28
3.2.2.3	Sensor de umidade do solo	29
3.3	Comunicação entre <i>Esp32</i> e Android:	30
4	RESULTADOS E DISCUSSÕES	32
4.1	HARDWARE	32
4.1.1	Desenvolvimento do Hardware	32
4.1.2	Desenvolvimento do HAL e Drivers do AOSP	35
4.2	SOFTWARE	37

4.2.1	Aplicativo	37
4.2.2	Unificação do Sistema de Monitoramento de Plantas	39
5	CONSIDERAÇÕES FINAIS	42
	Referências	44

1

INTRODUÇÃO

Com o avanço da tecnologia e o crescente interesse pelo cultivo e cuidado de plantas domésticas, o monitoramento dessas plantas por meio de aplicativos móveis tem se tornado uma prática cada vez mais popular. Essa tendência reflete a sinergia entre a paixão pela natureza e a conveniência proporcionada pela tecnologia móvel.

Nos últimos anos, a tecnologia desempenha um papel crucial no desenvolvimento de soluções inovadoras para auxiliar entusiastas de plantas domésticas a cuidarem de suas espécies de forma mais eficiente e eficaz. O monitoramento de plantas domésticas por meio de aplicativos no sistema operacional Android surge como uma solução prática e acessível, permitindo que os amantes de plantas monitorem e cuidem de suas vegetações com facilidade (SOUSA, 2019).

Esses aplicativos disponibilizam uma ampla gama de recursos, que vão desde a monitoração de fatores ambientais, como temperatura e umidade do ar, até a programação de lembretes para regar as plantas. Além disso, muitos desses aplicativos oferecem uma extensa biblioteca de informações sobre diversas espécies de plantas, fornecendo orientações detalhadas sobre os cuidados específicos necessários para cada uma delas (RAHIM; ZAKI; NOOR, 2020).

A integração com outros dispositivos inteligentes, como sistemas de irrigação automatizados, proporciona uma experiência ainda mais conveniente, permitindo que os usuários automatizem o processo de cuidado de suas plantas com base nos dados coletados pelo aplicativo (BRANDÃO et al., 2022).

Essas funcionalidades e benefícios tornaram o monitoramento de plantas do-

mésticas por meio de aplicativos no sistema operacional Android uma ferramenta essencial para aqueles que desejam cultivar suas plantas de maneira mais eficiente, proporcionando uma combinação perfeita entre a natureza e a tecnologia, garantindo que os amantes de plantas desfrutem de uma experiência aprimorada em seu cultivo.

1.1 Contextualização e Definição do Problema

À medida que a preocupação com o cultivo e cuidado de plantas domésticas aumenta, surge a necessidade de soluções inovadoras que facilitem o monitoramento e o manejo adequado dessas plantas. A criação de uma estação de monitoramento de plantas utilizando um dispositivo embarcado, como o *NodeMCU*, surge como uma abordagem promissora para atender a essa demanda.

O problema abordado neste trabalho consiste em desenvolver uma estação de monitoramento de plantas capaz de coletar dados relevantes para o crescimento saudável das plantas em um ambiente doméstico. Essa estação será equipada com sensores que medirão parâmetros essenciais, como temperatura, umidade, luminosidade e umidade do solo.

O objetivo principal dessa estação de monitoramento é fornecer informações precisas sobre as condições ambientais e do solo em que as plantas estão situadas. Esses dados coletados pelos sensores serão utilizados para monitorar o ambiente e auxiliar os usuários no cuidado adequado das plantas.

Ao coletar informações sobre a temperatura, a estação de monitoramento ajudará a garantir que as plantas estejam em um ambiente com condições ideais, evitando extremos de calor ou frio que possam afetar negativamente seu crescimento. A umidade é outro fator crítico, pois indica se as plantas estão recebendo a quantidade adequada de água para seu desenvolvimento saudável.

A luminosidade desempenha um papel crucial no crescimento das plantas. Portanto, a estação de monitoramento fornecerá dados sobre a quantidade de luz disponível, permitindo aos usuários avaliar se suas plantas estão recebendo a iluminação necessária para a fotossíntese.

Além disso, o monitoramento da umidade do solo é fundamental para o sucesso do cultivo de plantas domésticas. A estação de monitoramento coletará informações sobre a umidade do solo, permitindo que os usuários determinem o momento adequado para a rega das plantas, evitando assim a falta ou o excesso de água.

Ao utilizar o *NodeMCU* como dispositivo embarcado, será possível coletar de forma automatizada e contínua todos esses dados. Essas informações serão enviadas para um sistema centralizado, como um aplicativo móvel ou plataforma online, permitindo que os usuários acompanhem o estado de suas plantas e tomem ações apropriadas com base nos dados coletados.

Em resumo, o problema consiste em desenvolver uma estação de monitoramento de plantas utilizando o *NodeMCU*, com o objetivo de coletar dados como temperatura, umidade, luminosidade e umidade do solo. Essa estação visa fornecer informações precisas para auxiliar os usuários no cuidado adequado de suas plantas.

1.2 Objetivos

O objetivo geral é criar um dispositivo embarcado utilizando o *NodeMCU* para desenvolver uma estação de monitoramento de plantas. Essa estação será equipada com sensores capazes de coletar dados como temperatura, umidade, luminosidade e umidade do solo, a fim de fornecer informações essenciais para o cuidado e crescimento adequado das plantas.

Para atingir o objetivo geral, os seguintes objetivos específicos foram definidos:

1. Configurar o *NodeMCU*: Preparar o *NodeMCU* para a integração dos sensores, estabelecendo a conexão e comunicação entre o dispositivo e os sensores. Isso pode envolver a utilização de bibliotecas específicas ou códigos personalizados para a plataforma.
2. Montar o circuito: Realizar a conexão dos sensores ao *NodeMCU*, garantindo uma conexão estável e adequada. Será verificada a correta conexão dos pinos e a utilização de componentes adicionais, como resistores ou capacitores, se necessário.

3. Programar o dispositivo: Desenvolver o código necessário para ler os dados dos sensores e transmiti-los por meio do *NodeMCU*. Isso inclui a leitura dos valores de temperatura, umidade, luminosidade e umidade do solo, bem como a formatação e envio dos dados para um destino específico, como uma plataforma online ou um aplicativo.
4. Testar e calibrar: Realizar testes para verificar a precisão e confiabilidade dos dados coletados pelos sensores. Se necessário, serão realizadas calibrações para garantir medições mais precisas.
5. Implementar a funcionalidade de monitoramento: Integrar os dados coletados em uma interface ou plataforma que permita o monitoramento contínuo das condições das plantas. Isso pode envolver o desenvolvimento de um aplicativo móvel, uma interface web ou a integração com sistemas existentes.
6. Documentar o projeto: Registrar todo o processo de criação do dispositivo embarcado, incluindo diagramas de circuito, código-fonte, configurações e instruções de uso. Essa documentação será essencial para a reprodução do projeto e para compartilhá-lo com outras pessoas interessadas.

Esses objetivos específicos são fundamentais para a realização do projeto e garantirão o desenvolvimento de uma estação de monitoramento de plantas funcional e eficiente.

1.3 Organização do Documento

Este documento está organizado da seguinte forma:

- Capítulo 1 - INTRODUÇÃO: Este capítulo fornece uma visão geral do trabalho desenvolvido, apresentando os objetivos e justificativas para o projeto.
- Capítulo 2 - FUNDAMENTAÇÃO TEÓRICA: Neste capítulo, será realizada uma revisão da literatura, explorando estudos e trabalhos existentes que serviram como base teórica para o desenvolvimento do sistema proposto.

- Capítulo 3 - METODOLOGIA: Este capítulo apresenta os materiais e métodos utilizados para a construção de cada parte do sistema projetado. Serão descritas as etapas de desenvolvimento, desde a seleção dos componentes até a implementação prática do sistema.
- Capítulo 4 - RESULTADOS E DISCUSSÕES: Neste capítulo, será abordada a montagem do hardware, o desenvolvimento do software e os testes realizados com o protótipo. Serão apresentados os resultados obtidos e realizada uma análise e discussão dos dados coletados.
- Capítulo 5 - CONCLUSÃO: Este capítulo apresenta as conclusões obtidas com base nos resultados e discussões realizadas anteriormente. Também são discutidos os objetivos alcançados e são apresentadas sugestões para trabalhos futuros, com o intuito de aprimorar o sistema e explorar novas possibilidades de pesquisa.

Essa estrutura proporcionará uma organização clara e coerente do trabalho, fornecendo uma introdução adequada, fundamentação teórica, descrição da metodologia, apresentação e discussão dos resultados, além de conclusões relevantes e direcionamentos para trabalhos futuros.

2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos e ferramentas utilizadas na concepção deste trabalho: *Esp32*, Android, os sensores de monitoramento de plantas e internet das coisas. Os trabalhos de conclusão de curso de (SOUSA, 2019), (BRANDÃO et al., 2022) e (PEREIRA, 2019) foram os principais alvos de estudo para o desenvolvimento deste trabalho, entre outros que serão apresentados a seguir.

A estação de monitoramento de plantas domésticas utilizando Android embarcado no ESP32 combina o uso do microcontrolador *Esp32*, que é capaz de se conectar à internet e processar dados, com a plataforma Android, que fornece uma interface amigável para o usuário. A fundamentação teórica para esse projeto envolve os seguintes conceitos.

2.1 Internet das Coisas - *IoT*

A Internet das Coisas (*IoT*) é uma tecnologia que possibilita a conexão e comunicação entre dispositivos físicos por meio da internet. Essa interconexão dos dispositivos na *IoT* permite a coleta, troca e análise de dados em tempo real, possibilitando a automatização de processos, a tomada de decisões mais eficientes e a criação de novos serviços e experiências para os usuários (ASHTON et al., 2009).

A *IoT* tem aplicações em diversos setores, como cidades inteligentes, agricultura, saúde, indústria, transporte, entre outros (ATZORI; IERA; MORABITO, 2010). Por exemplo, na agricultura, sensores conectados podem monitorar a umidade do solo,

temperatura e outros parâmetros para otimizar a irrigação e o cultivo das plantas. Na área de saúde, dispositivos vestíveis podem coletar informações sobre a atividade física e os sinais vitais dos pacientes, permitindo o monitoramento remoto e o cuidado personalizado.

No contexto do projeto da estação de monitoramento de plantas, a *IoT* desempenha um papel fundamental. Através da conexão dos sensores de temperatura, umidade, luminosidade e umidade do solo ao ESP32, os dados coletados podem ser enviados para uma plataforma online ou aplicativo móvel. Isso permite que os usuários monitorem o estado das plantas em tempo real, recebam alertas e tomem ações adequadas com base nos dados coletados.

A utilização da *IoT* nesse projeto proporciona maior eficiência no monitoramento das plantas, permitindo uma resposta rápida e precisa às condições ambientais. Além disso, a integração com outras tecnologias, como o Android, possibilita uma interface amigável para os usuários interagirem com a estação de monitoramento.

Compreender os conceitos e princípios da *IoT* é essencial para o desenvolvimento de soluções inteligentes e conectadas, como a estação de monitoramento de plantas domésticas. Essa fundamentação teórica servirá como base para o desenvolvimento adequado do projeto, considerando as melhores práticas e aproveitando ao máximo as possibilidades oferecidas pela *IoT*.

2.2 Android

O desenvolvimento Android é baseado em um conjunto de fundamentos teóricos que embasam a criação de aplicativos para dispositivos móveis com o sistema operacional Android. Esses fundamentos incluem conceitos de arquitetura de software, padrões de design e princípios de desenvolvimento móvel. Abaixo estão alguns dos principais fundamentos teóricos do Android utilizando as documentações (DEVELOPERS, 2023) e (GUIDELINES., 2023):

- **Arquitetura MVC (Model-View-Controller):** A arquitetura MVC é amplamente utilizada no desenvolvimento de aplicativos Android. Ela separa a lógica de

negócios (Model), a interface do usuário (View) e o controle do fluxo de dados (Controller). Essa abordagem ajuda a melhorar a modularidade, a manutenção e a testabilidade do código.

- **Padrões de Design:** Existem diversos padrões de design que podem ser aplicados no desenvolvimento Android para promover a reutilização de código, a escalabilidade e a legibilidade do código. Alguns exemplos incluem o padrão de Observador (Observer), Injeção de Dependência (Dependency Injection) e Singleton.
- **Ciclo de Vida da Atividade (Activity Lifecycle):** O ciclo de vida da atividade no Android é uma sequência de eventos que ocorrem ao criar, pausar, retomar e destruir uma atividade. Compreender o ciclo de vida da atividade é essencial para gerenciar adequadamente os recursos do aplicativo, como conexões de banco de dados, sensores e gerenciamento de memória.
- **Componentes do Android:** O Android possui uma série de componentes essenciais, como Atividades (Activities), Fragmentos (Fragments), Provedores de Conteúdo (Content Providers) e Serviços (Services). Cada componente tem um propósito específico e permite a construção de aplicativos mais complexos e interativos.
- **Material Design:** O Material Design é um conjunto de diretrizes de design desenvolvido pelo Google para criar interfaces de usuário modernas e atraentes. Essas diretrizes fornecem princípios de design, componentes e padrões de interação que ajudam a criar aplicativos consistentes e visualmente agradáveis.

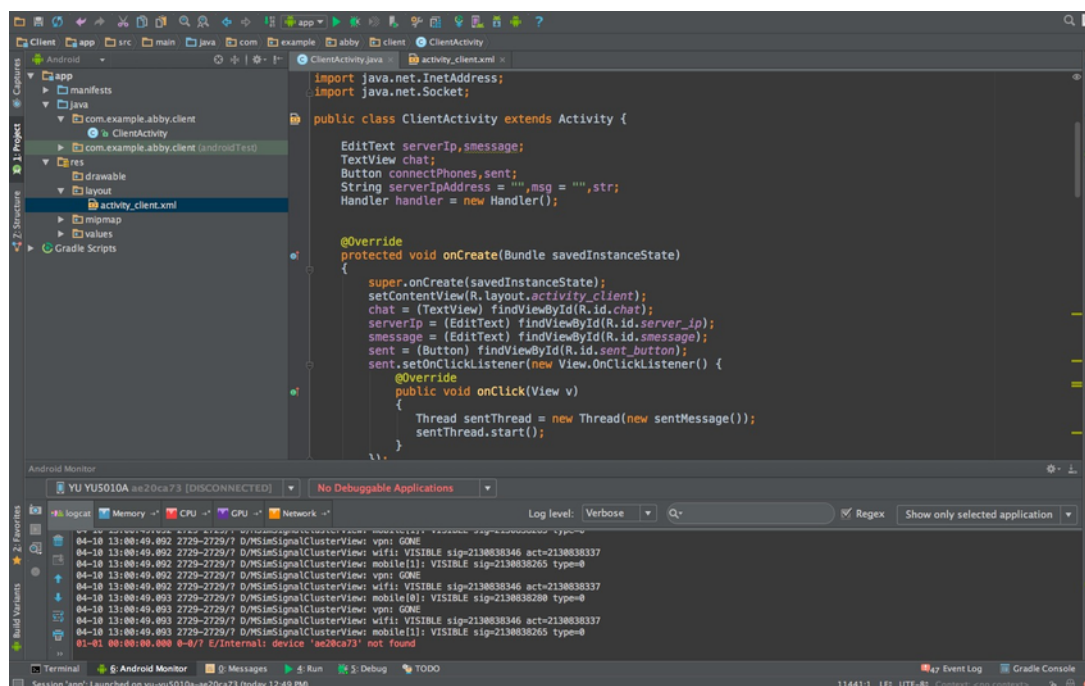


Figura 1 – Ambiente de Desenvolvimento Android.

Fonte: (JOBY, 2016)

2.2.1 ANDROID OPEN SOURCE PROJECT - AOSP

O AOSP é um projeto de código aberto desenvolvido pelo Google que fornece a base para o sistema operacional Android. Ele inclui o código-fonte, ferramentas e recursos necessários para construir e modificar o Android. O objetivo do AOSP é promover a transparência e o desenvolvimento aberto do Android, permitindo que desenvolvedores e fabricantes personalizem o sistema de acordo com suas necessidades. Ele fornece as bases do sistema, como o kernel do Linux, o framework de aplicativos e os drivers de hardware (ver Figura 2) (SOURCE, 2023a).

Os desenvolvedores podem criar versões personalizadas, adicionar novos recursos e corrigir bugs. O AOSP é usado para criar ROMs personalizadas e não inclui os serviços e aplicativos do Google, que são licenciados separadamente. Em resumo, o AOSP é a base do sistema Android, oferecendo flexibilidade e oportunidades de personalização para desenvolvedores e fabricantes.

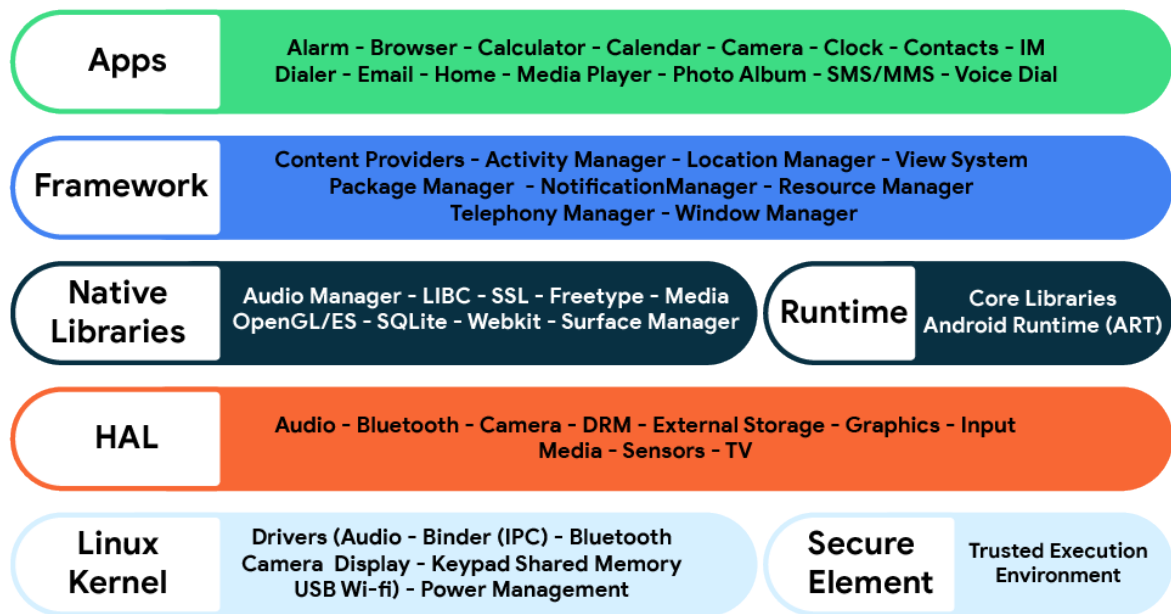


Figura 2 – Estrutura do AOSP.

Fonte: (SOURCE, 2023b)

2.3 Automação - Utilizando o ESP

Em conformidade com a publicação de (TOTVS, 2022), A automação de processos consiste no uso de tecnologia e integração de dados e sistemas para automatizar tarefas que anteriormente eram realizadas manualmente. Seu objetivo é aumentar a produtividade, eficiência e valor gerado pelas atividades.

Essa estratégia de otimização pode ser aplicada em diversas áreas e adaptada para atividades específicas. Embora a automação possa envolver o uso de robôs ou dispositivos mecânicos para executar tarefas no lugar de profissionais, nem sempre significa substituir completamente o trabalho humano pela tecnologia ou equipamento.

Para (STRAUB, 2020), utilizar o *Esp32* para uma automação residencial tem custo reduzido, uma vez que o microcontrolador já possui um sistema de comunicação wifi e bluetooth. A integração desses recursos elimina a necessidade de módulos adicionais e simplifica a conexão do projeto, eliminando a necessidade de fios e outras conexões para os componentes necessários. Isso torna o projeto mais compacto e simples.

(STRAUB, 2020) também cita as automações possíveis com o *Esp32*. Algumas delas já foram mencionadas anteriormente, mas vamos explorar um pouco mais cada uma:

- **Sistema de controle de iluminação:** Através da automação residencial, é possível controlar o liga e desliga de lâmpadas utilizando smartphones e computadores. Isso pode ser feito através de módulos relés conectados ao ESP32.
- **Sistema de monitoramento:** O ESP32 CAM, que é um módulo com uma câmera integrada, pode ser utilizado para monitorar a residência. Através do ESP32, é possível configurar e acessar a câmera de forma simples e direta.
- **Sistema de segurança:** A automação residencial permite o acionamento e a interrupção de sensores de presença e movimento, como o *PIR*, e seus respectivos sistemas de alarme sonoro.
- **Sistema de Acesso:** É possível incluir sistemas de verificação de acesso utilizando leitores biométricos, leitores *RFID*, senhas e outras tecnologias compatíveis, garantindo a segurança contra a entrada de pessoas não autorizadas.
- **Sistema de controle para cortinas e janelas:** Com dispositivos e produtos compatíveis, é possível automatizar a abertura e o fechamento de cortinas e janelas, utilizando sensores de luz e chuva, por exemplo.
- **Sistema de controle de temperatura:** Através da automação residencial, é possível controlar a temperatura do ambiente. Sensores como o DHT11 ou DHT22 e o DS18B20 podem ser integrados ao sistema para controlar o ar condicionado e aquecedor.

3

ESTAÇÃO DE MONITORAMENTO DE PLANTAS

Neste capítulo serão apresentados os componentes e dispositivos necessários para a implementação do sistema de monitoramento de plantas. Os principais elementos incluem sensores, módulos de comunicação e placas de desenvolvimento. Cada componente desempenha um papel fundamental no funcionamento do sistema e na coleta de dados.

3.1 Visão Geral do Protótipo

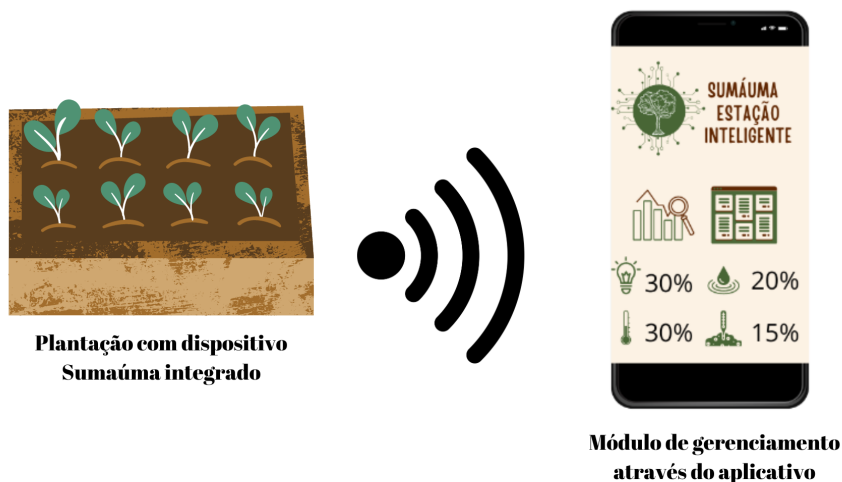


Figura 3 – Protótipo idealizado no início do projeto.

Fonte: (Autor)

A estação de monitoramento de plantas é composta por um dispositivo embarcado ESP32 e um aplicativo Android, que trabalham em conjunto para fornecer aos usuários informações atualizadas sobre as condições ambientais de suas plantas. O protótipo, representado na Figura 3, foi projetado para ser de fácil utilização e fornecer uma experiência intuitiva para os usuários.

O dispositivo embarcado ESP32 desempenha um papel central na coleta e processamento dos dados. Ele é conectado a sensores de luminosidade, umidade, temperatura e umidade do solo, que são responsáveis por coletar informações relevantes sobre as condições ambientais ao redor das plantas. O ESP32 é responsável por processar os dados coletados pelos sensores e enviá-los para o aplicativo Android por meio de uma conexão sem fio.

O aplicativo Android foi desenvolvido para fornecer aos usuários uma interface intuitiva e fácil de usar. Ele se conecta ao dispositivo ESP32 por meio de uma conexão sem fio e exibe os dados coletados pelos sensores de forma clara e organizada. Os usuários podem acompanhar em tempo real os níveis de luminosidade, umidade do solo, temperatura e umidade do ar, permitindo que tomem medidas adequadas para o cuidado das plantas.

Nas seções seguintes são apresentados os detalhes da implementação do sistema, incluindo os componentes utilizados, a conexão dos sensores ao ESP32 e a Comunicação entre o Esp32 e o Android.

3.2 Definição dos Componentes

3.2.1 ESP32:

O *Esp32*, exibido na Figura 4, é um microcontrolador de baixo custo e baixo consumo de energia amplamente utilizado em projetos de IoT devido às suas capacidades de comunicação sem fio, processamento e conectividade com a Internet. Ele é equipado com Wi-Fi e Bluetooth integrados, o que permite a troca de dados e a interação com outros dispositivos.

A arquitetura do ESP32 é baseada em um processador dual-core Xtensa LX6 de 32

bits, com frequência de clock de até 240 MHz. Ele possui memória RAM e memória flash embutidas, o que permite armazenar o código de programação e os dados coletados pelos sensores. Além disso, o ESP32 possui uma variedade de interfaces de comunicação, incluindo UART, SPI, I2C e I2S, que facilitam a conexão com outros dispositivos e periféricos (ESPRESSIF, 2018).

Uma das principais vantagens do ESP32 é a sua capacidade de se comunicar por meio de redes sem fio, como Wi-Fi e Bluetooth. Isso permite que o dispositivo se conecte à Internet e troque dados com servidores remotos, aplicativos móveis e outros dispositivos IoT. O suporte integrado ao protocolo MQTT (Message Queuing Telemetry Transport) também é uma característica importante do ESP32, que facilita a integração com serviços de nuvem e a troca eficiente de mensagens entre dispositivos (COHN et al., 2015).

Para o projeto da estação de monitoramento de plantas, o ESP32 desempenha um papel central na coleta de dados dos sensores e no envio desses dados para o aplicativo Android. Ele recebe as leituras dos sensores de luminosidade, umidade, temperatura e umidade do solo por meio de suas interfaces analógicas e digitais, processa esses dados e os envia para o aplicativo por meio de uma conexão Wi-Fi.

A programação do ESP32 pode ser realizada utilizando diversas linguagens, como C/C++ e MicroPython. Existem várias ferramentas de desenvolvimento disponíveis, incluindo a popular plataforma Arduino, que oferece uma biblioteca específica para o ESP32. Essas ferramentas simplificam o desenvolvimento do código, permitindo o controle dos pinos de E/S, a configuração dos módulos de comunicação sem fio e a integração com os sensores.

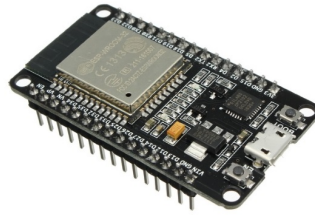


Figura 4 – Microcontrolador - Esp 32.

Fonte: (HERO, 2023)

3.2.2 Sensores

A estação de monitoramento de plantas utiliza diferentes tipos de sensores para coletar dados ambientais e obter informações relevantes sobre o estado das plantas. Os principais sensores incluídos no protótipo são:

3.2.2.1 Sensor de Umidade do ar e Temperatura

O sensor de umidade do ar e temperatura DHT11 (ver Figura 5) é um sensor digital amplamente utilizado para medir a temperatura e a umidade ambiente. Ele possui uma faixa de medição de temperatura de 0° a 50° Celsius e uma faixa de medição de umidade do ar de 20% a 90%. O sensor fornece leituras com uma precisão de aproximadamente 2° Celsius para temperatura e 5% para umidade (OLIVEIRA, 2019).

No sistema de monitoramento de plantas, o DHT11 é conectado ao ESP32 por meio de uma interface digital, como GPIO (General Purpose Input/Output). O ESP32 lê os dados do sensor e os processa para obter os valores de temperatura e umidade. Essas leituras são enviadas para o aplicativo Android para exibição ao usuário .

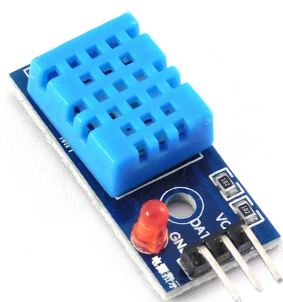


Figura 5 – Sensor DHT11.

Fonte: (ELETRONICOS, 2023)

3.2.2.2 Sensor de Luminosidade

O sensor de luminosidade, também conhecido como LDR (Light Dependent Resistor), é um componente eletrônico que varia sua resistência elétrica de acordo com a intensidade da luz incidente. Quanto maior a intensidade da luz, menor a resistência do sensor (ALVES, 2019).

O LDR (ver Figura 6) é conectado ao ESP32 por meio de uma interface analógica, como uma entrada ADC (Analog-to-Digital Converter). O ESP32 lê a tensão fornecida pelo LDR e a converte em um valor digital correspondente à intensidade da luz. Essa informação é transmitida para o aplicativo Android, permitindo que o usuário visualize a luminosidade do ambiente.



Figura 6 – Sensor de Luminosidade - LDR.

Fonte: (MULTILÓGICA-SHOP, 2023a)

3.2.2.3 Sensor de umidade do solo

O sensor de umidade do solo (ver Figura 7) é utilizado para medir a umidade presente no solo. Ele consiste em dois eletrodos que são enterrados no solo e medem a resistividade elétrica. A umidade do solo afeta a condutividade elétrica, sendo que solos úmidos conduzem eletricidade com mais facilidade do que solos secos (CIRCUITO, 2023).

O sensor de umidade do solo é conectado ao ESP32 por meio de uma interface analógica ou digital, dependendo do modelo do sensor. O ESP32 realiza a leitura da resistência elétrica medida pelo sensor e a converte em um valor que representa a umidade do solo. Esses dados são transmitidos para o aplicativo Android, permitindo que o usuário monitore a umidade do solo das plantas.

A integração dos sensores ao ESP32 e o processamento dos dados são realizados por meio da programação do microcontrolador. O ESP32 lê os valores dos sensores em intervalos regulares e os envia para o aplicativo Android por meio de uma conexão Wi-Fi. O aplicativo exibe os dados coletados em tempo real, permitindo que o usuário acompanhe as condições ambientais e o estado das plantas de forma conveniente.

Essa integração entre os sensores e o ESP32 é uma parte fundamental do sistema de monitoramento de plantas, fornecendo informações valiosas para a tomada de decisões relacionadas ao cuidado das plantas.

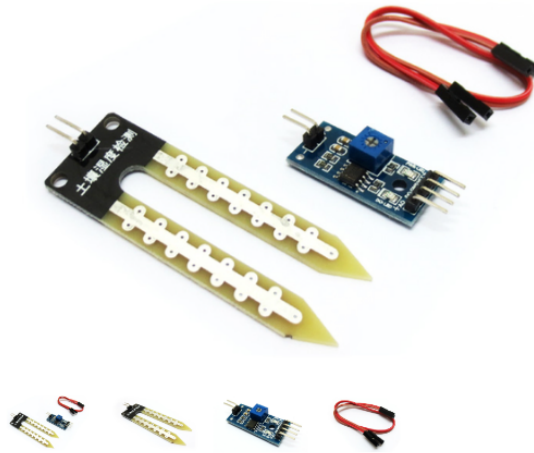


Figura 7 – Sensor de Umidade do Solo.

Fonte: (MULTILÓGICA-SHOP, 2023b)

3.3 Comunicação entre *Esp32* e Android:

Para estabelecer a comunicação entre o ESP32 e o dispositivo Android, é necessário utilizar protocolos e tecnologias apropriados. Isso pode incluir a utilização de Wi-Fi ou Bluetooth para a troca de dados entre os dispositivos. É fundamental entender os conceitos de comunicação sem fio e como configurar a conexão entre o *Esp32* e o dispositivo Android. Abaixo está o passo a passo de uma configuração entre o *Esp32* e o Android, onde foram utilizadas as referências das documentações do *Esp32* (ESPRESSIF, 2023) e do Android Conectividade (DEVELOPERS, 2023) e (TEIXEIRA, 2019).

A comunicação entre o *Esp32* e dispositivos Android pode ser realizada por meio de várias tecnologias, como Wi-Fi, Bluetooth ou até mesmo via USB. Aqui está um passo a passo básico para estabelecer essa comunicação usando a tecnologia Wi-Fi:

- Passo 1: Configurar o *Esp32* Certifique-se de ter o firmware adequado instalado no *Esp32* para suportar a comunicação via Wi-Fi. Configure as credenciais de rede (SSID e senha) no *Esp32* para que ele possa se conectar à mesma rede Wi-Fi que o dispositivo Android.

- Passo 2: Desenvolver o aplicativo Android

Use o Android Studio ou outra ferramenta de desenvolvimento Android para criar um aplicativo para o dispositivo Android. Adicione as permissões necessárias no arquivo de manifesto do aplicativo para acessar a rede Wi-Fi.

- Passo 3: Estabelecer a conexão Wi-Fi

No aplicativo Android, implemente a lógica para procurar e se conectar à mesma rede Wi-Fi que o *Esp32*. Utilize as bibliotecas e APIs do Android para gerenciar a conexão Wi-Fi.

- Passo 4: Trocar dados entre o *Esp32* e o Android

Defina um protocolo de comunicação, como *MQTT* ou *HTTP*, para enviar e receber dados entre os dispositivos. Implemente a lógica de envio e recebimento de dados no *Esp32* e no aplicativo Android.

- Passo 5: Tratar os dados recebidos

No aplicativo Android, implemente a lógica para processar os dados recebidos do *Esp32*. Exiba as informações recebidas na interface do aplicativo ou execute ações com base nesses dados.

4

RESULTADOS E DISCUSSÕES

Nesta seção, será abordada a montagem do hardware e o desenvolvimento do software. A escolha cuidadosa de cada um desses componentes foi fundamental para obter resultados mais eficientes neste projeto. Além disso, serão apresentados os testes e análises realizados com o protótipo desenvolvido, proporcionando uma visão mais completa do trabalho.

4.1 HARDWARE

4.1.1 Desenvolvimento do Hardware

Para a criação de um dispositivo embarcado (*NodeMCU*) para uma estação de monitoramento de plantas baseado em Android, foram necessários sensores específicos para coletar dados cruciais sobre as condições ambientais das plantas. Esses sensores incluem:

- Sensor de temperatura: Responsável por medir a temperatura do ambiente onde as plantas estão localizadas. Isso ajuda a monitorar variações de temperatura que podem afetar o crescimento e a saúde das plantas.
- Sensor de umidade: Utilizado para medir a umidade do ar ao redor das plantas. Essa informação é fundamental para determinar se o ambiente está muito seco ou muito úmido, ajudando a controlar a rega e fornecer as condições ideais para o

desenvolvimento das plantas.

- Sensor de luminosidade: Responsável por medir a intensidade da luz que as plantas estão recebendo. Isso é importante, pois diferentes espécies de plantas possuem necessidades específicas de luz. O sensor de luminosidade ajuda a garantir que as plantas estejam recebendo a quantidade adequada de luz para realizar a fotossíntese e crescer saudáveis.
- Sensor de umidade do solo: Essencial para monitorar a umidade do solo onde as plantas estão enraizadas. Ele fornece informações sobre se o solo está seco ou úmido, permitindo que a rega seja realizada no momento certo e evitando problemas como a falta ou o excesso de água para as plantas.

Esses sensores são integrados ao dispositivo embarcado (*NodeMCU*), que atua como o cérebro do sistema de monitoramento. O (*NodeMCU*) coleta os dados dos sensores e os envia para o aplicativo Android, onde os usuários podem visualizar as informações em tempo real e tomar ações adequadas para garantir o bem-estar das plantas.

A combinação dos sensores de temperatura, umidade, luminosidade e umidade do solo, conforme a Figura 8, permite um monitoramento abrangente das condições ambientais das plantas. Com base nesses dados, os usuários podem ajustar o ambiente, como a rega, a exposição à luz e a ventilação, para criar as condições ideais de crescimento para suas plantas.

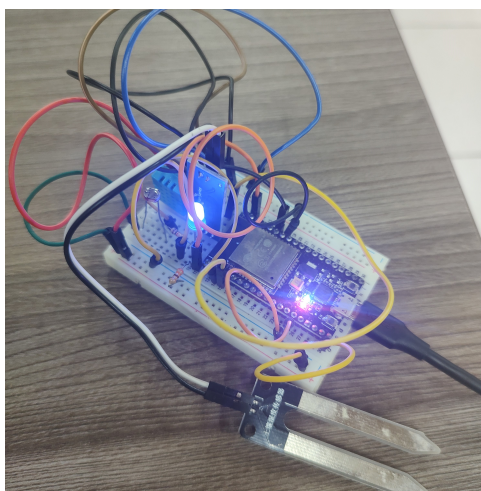


Figura 8 – Dispositivo de Monitoramento de Plantas com Esp32.

Fonte: (Autor)

```
#include "DHT.h"

#define MODE_MANUAL 0
#define MODE_AUTO 1
#define LED_BUILTIN 2

//Added temperature sensor
#define DHT_PIN 21
#define DHTTYPE DHT11

int ledPin = 23;
int ledChannel = 0;
int ledValue = 10;
int ledMode = MODE_MANUAL;

int ldrPin = 36;
int ldrMax = 4000;

int thresholdValue = 50;

//create an instance of DHT sensor
DHT dht(DHT_PIN, DHTTYPE);
```

Figura 9 – Codificação no Arduino IDE.

Fonte: (Autor)

Conforme mostrado na Figura 9, foi utilizada a biblioteca DHT.h, que é uma biblioteca utilizada para facilitar a leitura de sensores de umidade e temperatura da série DHT, como o DHT11, DHT22, AM2302, RHT03, entre outros. Esses sensores são amplamente utilizados em projetos de automação residencial, monitoramento ambiental e outras aplicações onde é necessário obter dados precisos de temperatura e umidade.

A biblioteca DHT.h simplifica a comunicação com os sensores DHT, fornecendo funções que permitem a leitura dos dados de temperatura e umidade com facilidade. Ela abstrai a complexidade de comunicação com o sensor, manipulação de bits e cálculos de conversão, tornando o processo de leitura dos dados mais simples e rápido.

Ao incluir a biblioteca DHT.h no código-fonte do projeto, é possível utilizar as funções disponibilizadas para iniciar a comunicação com o sensor, solicitar a leitura dos dados de temperatura e umidade e receber os valores resultantes. A biblioteca também oferece recursos adicionais, como a verificação da integridade dos dados lidos

e a configuração de parâmetros específicos do sensor.

Em resumo, a biblioteca DHT.h simplifica a leitura de sensores de umidade e temperatura da série DHT, proporcionando uma interface conveniente para obter dados precisos de temperatura e umidade em projetos eletrônicos. Com ela, é possível incorporar facilmente o monitoramento dessas variáveis em seus projetos, agilizando o desenvolvimento e aumentando a precisão das medições.

4.1.2 Desenvolvimento do HAL e Drivers do AOSP

Para o desenvolvimento do *AOSP* (Android Open Source Project), foram realizadas modificações em alguns arquivos a fim de converter os dados coletados pelos sensores e disponibilizá-los na camada do usuário, conforme ilustrado nas Figuras 10, 11 e 12, 13 e 14. Cada módulo do *AOSP* requer um arquivo de configuração específico para direcionar o sistema de compilação com informações sobre o módulo em questão.

No contexto do *AOSP*, o *HAL* (Hardware Abstraction Layer) desempenha um papel crucial, pois estabelece uma interface padrão para acessar os componentes do dispositivo. Isso permite que os fabricantes implementem o *HAL* de acordo com as especificidades do seu hardware. Com a definição dos sensores que serão utilizados, a implementação do *HAL* serve como um padrão, garantindo que os sensores sejam acessíveis por qualquer outra aplicação que necessite dessas informações.

Ao adotar o *HAL* como padrão, o desenvolvimento do *AOSP* se beneficia da interoperabilidade entre os diferentes dispositivos Android, proporcionando uma experiência consistente para os usuários, independentemente do hardware específico do dispositivo. Essa abordagem modular e padronizada facilita o desenvolvimento de aplicativos que dependem dos dados dos sensores, pois eles podem ser acessados de maneira uniforme, independentemente do fabricante do dispositivo.

Dessa forma, ao modificar os arquivos e realizar a implementação adequada do *HAL* para os sensores selecionados, os dados coletados pelos sensores se tornam disponíveis para qualquer outra aplicação que precise dessas informações, permitindo uma maior flexibilidade e utilidade do sistema de monitoramento de plantas baseado

em Android.



Figura 10 – Desenvolvimento da HAL.

Fonte: (Autor)

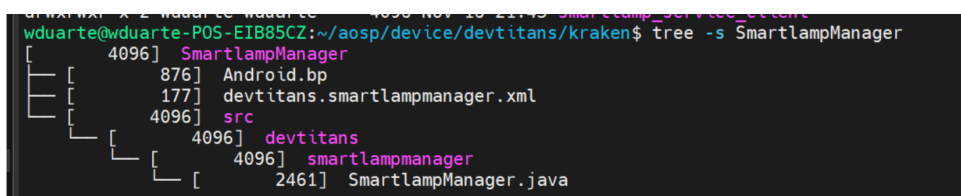


Figura 11 – Desenvolvimento da Manager.

Fonte: (Autor)

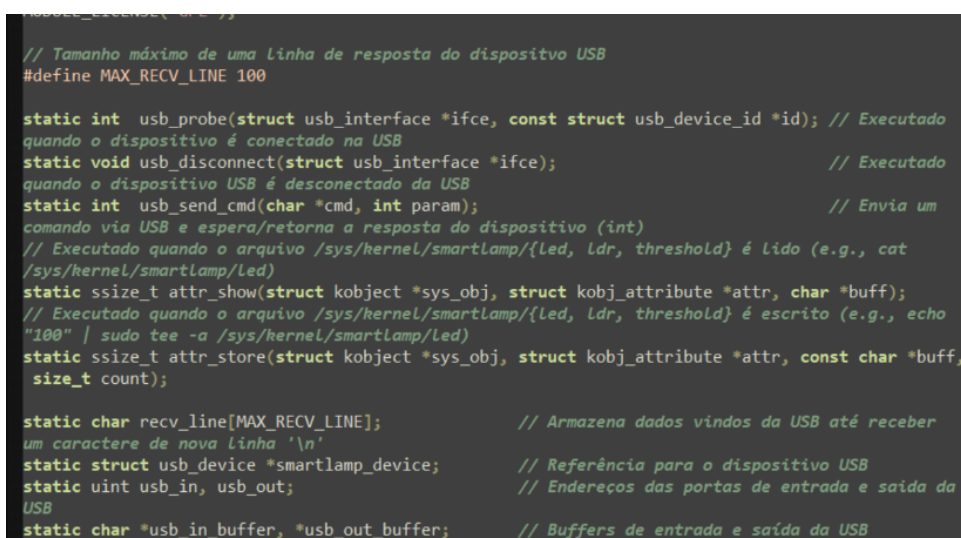


Figura 12 – Desenvolvimento do Driver.

Fonte: (Autor)

```
100: Completamente claro
*/
int getThreshold();
bool setThreshold(int thresholdValue);

//Para leitura da temperatura e umidade
float getTemp();
float getUmi();
```

Figura 13 – Desenvolvimento da Biblioteca.

Fonte: (Autor)

```
#pragma once // Inclui esse cabeçalho apenas uma vez
#include <iostream> // std::cout (char-out) e std::endl (end-line)
#include <string.h> // Função strcmp
#include <stdlib.h> // Função atoi
#include "agromito_lib.h" // Classe Agromito
```

Figura 14 – Aplicação da Biblioteca.

Fonte: (Autor)

4.2 SOFTWARE

4.2.1 Aplicativo

O desenvolvimento do aplicativo de monitoramento de plantas em Android envolveu a codificação de diferentes funcionalidades para permitir a interação do usuário com a estação de monitoramento. O código foi escrito em linguagem Java, que é a linguagem de programação predominante no desenvolvimento de aplicativos Android.

Uma das principais partes do código é a implementação da interface do usuário, que inclui o layout da tela, botões, caixas de texto e outros elementos de interação. O layout da interface foi projetado para ser intuitivo e fácil de usar, proporcionando uma experiência agradável ao usuário, ilustrado na Figura 15.

Além disso, a codificação também inclui a lógica do aplicativo, como a obtenção e exibição dos dados coletados pela estação de monitoramento. Isso envolve a comunicação com o ESP32, que é responsável por enviar os dados para o aplicativo. A codificação adequada dos protocolos de comunicação e o tratamento correto dos dados recebidos são essenciais para garantir a precisão das informações exibidas no aplicativo.

Durante o processo de codificação, foram seguidas as melhores práticas de desenvolvimento Android, como a separação de responsabilidades em classes e métodos, uso de bibliotecas e frameworks relevantes e a aplicação de princípios de design e

arquitetura, como o padrão MVC (Model-View-Controller) ou MVVM (Model-View-ViewModel).

Por fim, testes foram realizados para garantir o bom funcionamento do aplicativo em diferentes dispositivos Android e cenários de uso. Isso envolveu testes de unidade, testes de integração e testes de usabilidade, buscando identificar e corrigir possíveis problemas e melhorar a qualidade do aplicativo.

No geral, o desenvolvimento em Android para o monitoramento de plantas exigiu a habilidade de escrever um código limpo, modular e eficiente, visando fornecer uma experiência satisfatória ao usuário e obter resultados precisos no monitoramento das plantas.



Figura 15 – Primeira Tela Desenvolvida em Ambiente Android.

Fonte: (Autor)

4.2.2 Unificação do Sistema de Monitoramento de Plantas

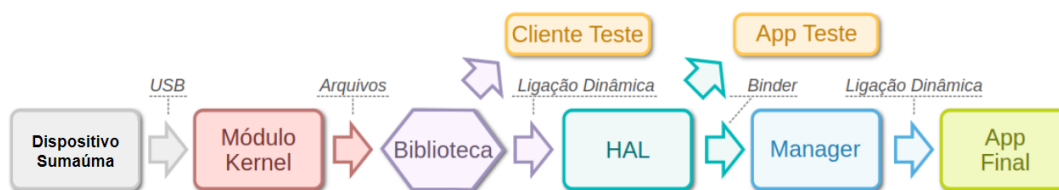


Figura 16 – Visão Geral das Camadas.

Fonte: (Autor)

Após todos os módulos serem desenvolvidos de forma separada, unificamos o sistema para realizar testes de verificação de erros e depuração de bugs. No início, identificamos um defeito no sensor de solo, o que levou à substituição do componente. Após a troca, realizamos o primeiro teste, que foi bem-sucedido, fornecendo as informações esperadas. Isso confirmou o funcionamento adequado do sistema após a resolução do problema com o sensor.

O sistema operacional Android é amplamente utilizado em dispositivos móveis, como smartphones e tablets. No contexto desse projeto, o Android será utilizado como uma plataforma embarcada no ESP32, permitindo a interação do usuário com a estação de monitoramento de plantas. É necessário compreender os princípios básicos de desenvolvimento de aplicativos Android, como layout de interface, interações de usuário e comunicação com outros dispositivos.

O aplicativo final foi modificado durante o processo, para deixar as informações mais claras para o usuário. Os dados retornavam apenas em números, onde as leituras são: Abaixo igual a 0, o solo precisa de água, conforme a Figura 18; Abaixo de 100, o solo precisa de atenção, como mostra a Figura 19; Acima de 100, o solo está umido e está tudo certo, mostrado na Figura 20.

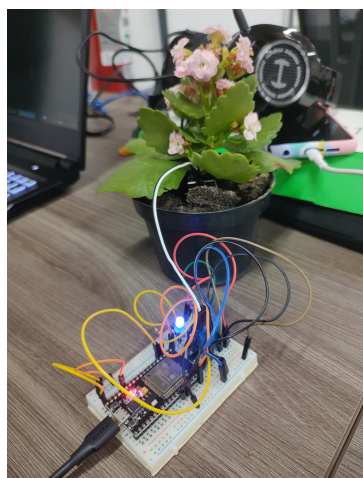


Figura 17 – Dispositivo Final em Execução.

Fonte: (Autor)



Figura 18 – Solo igual a 0 - O solo precisa de água.

Fonte: (Autor)



Figura 19 – Umidade do solo abaixo de 100 - O solo precisa de atenção.

Fonte: (Autor)



Figura 20 – Umidade do solo acima de 100 - Tudo certo.

Fonte: (Autor)

5

CONSIDERAÇÕES FINAIS

Com base na análise e implementação do sistema de monitoramento de plantas utilizando a combinação do AOSP (Android Open Source Project) e do ESP32, chegamos a importantes conclusões.

O monitoramento de plantas utilizando a combinação do AOSP como sistema operacional embarcado no ESP32 mostrou-se uma solução eficiente e versátil. A integração do AOSP proporciona uma interação direta entre o usuário e a estação de monitoramento de plantas, facilitando o acesso e controle dos dados coletados pelos sensores.

O AOSP oferece uma base sólida e personalizável para o desenvolvimento de aplicativos Android voltados para o monitoramento de plantas. Com o acesso ao código-fonte completo, é possível adaptar o sistema operacional e desenvolver recursos específicos para atender às necessidades do monitoramento de plantas, como controle de iluminação, umidade, temperatura e outros parâmetros essenciais.

Por sua vez, o ESP32 é um microcontrolador poderoso e de baixo custo, ideal para projetos de automação e monitoramento. Sua capacidade de conexão com sensores e dispositivos externos, juntamente com a integração com o AOSP, torna-o uma escolha ideal para a implementação de uma estação de monitoramento de plantas.

Através dessa combinação, é possível coletar dados precisos e em tempo real sobre as condições ambientais das plantas, permitindo um acompanhamento detalhado de seu crescimento e saúde. Além disso, a plataforma oferece a possibilidade de integração com outros sistemas de automação residencial, facilitando a criação de ambientes

controlados e a aplicação de ações corretivas quando necessário.

Portanto, a combinação do AOSP e do ESP32 abre um leque de possibilidades no monitoramento de plantas, trazendo avanços significativos para a agricultura inteligente e sustentável. Com o contínuo desenvolvimento dessas tecnologias, espera-se que mais recursos e funcionalidades sejam agregados, tornando o monitoramento de plantas cada vez mais eficiente e acessível a agricultores, pesquisadores e entusiastas.

REFERÊNCIAS

- ALVES, P. Ldr – o que é e como funciona! <https://www.manualdaeletronica.com.br/ldr-o-que-e-como-funciona/>, 2019. 28
- ASHTON, K. et al. That ‘internet of things’ thing. *RFID journal*, Hauppauge, New York, v. 22, n. 7, p. 97–114, 2009. 18
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. 18
- BRANDÃO, R. d. S. et al. Automação do monitoramento de plantas usando o módulo nodemcu v3 esp8266. 2022. 13, 18
- CIRCUITO, C. Sensor de umidade do solo. <https://curtocircuito.com.br/blog/Categoria%20Arduino/sensores-de-umidade-de-solo-tipos-e-diferencas>, 2023. 29
- COHN, R. et al. *MQTT Version 3.1. 1, Specification URIs*. 2015. 26
- DEVELOPERS, A. Android developers. <https://developer.android.com/>, 2023. 19, 30
- ELETRONICOS, C. *DHT11*. 2023.
<https://www.acheicomponentes.com.br/modulos/modulo-sensor-de-temperatura-dht11-p-arduino>. 28
- ESPRESSIF. *ESP32 Technical Reference Manual*. 2018. 26
- ESPRESSIF. Documentation - esp32. <https://docs.espressif.com/projects/espressif/en/latest/esp32/>, 2023. 30
- GUIDELINES., G. M. D. Google material design guidelines. <https://material.io/design>, 2023. 19
- HERO, M. *esp32*. 2023. <https://www.makerhero.com/produto/modulo-wifi-esp32-bluetooth/>. 27
- JOBY, A. P. *Ambiente de Desenvolvimento Android*. 2016.
https://www.researchgate.net/figure/The-Android-Studio-IDE_fig1_303737681. 21
- MULTILÓGICA-SHOP. *LDR*. 2023. <https://multilogica-shop.com/produtos/foto-cc3a9lula-ldr-10mm/>. 28

MULTILÓGICA-SHOP. *Sensor de Umidade de Solo*. 2023. <https://multilogica-shop.com/produtos/modulo-sensor-de-umidade-do-solo/>. 30

OLIVEIRA, E. Como usar com arduino – sensor de umidade e temperatura dht11. <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-de-umidade-e-temperatura-dht11>, 2019. 27

PEREIRA, D. N. d. O. Smart estufa-monitoramento e integração com iot na estufa hidropônica da universidade federal fluminense-campus praia vermelha. Universidade Federal Fluminense, 2019. 18

RAHIM, N. A.; ZAKI, F.; NOOR, A. Smart app for gardening monitoring system using iot technology. *system*, v. 29, n. 04, p. 7375–7384, 2020. 13

SOURCE, A. *Android Projeto de código aberto*. 2023. <https://source.android.com/?hl=pt-br>. 21

SOURCE, A. *Estrutura do AOSP*. 2023.

https://source.android.com/static/docs/setup/about/images/android_framework_details.png?hl=pt-br. 22

SOUSA, D. L. d. A. Design de interface para controle, monitoramento e automação do cultivo doméstico de plantas. 2019. 13, 18

STRAUB, M. G. Automação residencial com esp32 - controle sua casa pela web. <https://www.usinainfo.com.br/blog/automacao-residencial-com-esp32-controle-sua-casa-pela-web/>, 2020. 22

TEIXEIRA, G. Esp32 wifi: Comunicação com a internet. <https://www.usinainfo.com.br/blog/esp32-wifi-comunicacao-com-a-internet/>, 2019. 30

TOTVS, E. Automação de processos: tipos, exemplos e como fazer. <https://www.totvs.com/blog/gestao-para-assinatura-de-documentos/automacao-de-processos/>, 2022. 22