

DESENVOLVIMENTO DE APLICATIVO ANDROID PARA CONTROLE E MONITORAMENTO DE DISPOSITIVOS PARA IRRIGAÇÃO

Arthur Breno Rocha Mariano¹, Daniela Andreska da Silva², Alan Bernard Oliveira de Sousa³

RESUMO: A crescente utilização de smartphones no mundo aliado a expansão da rede de internet móvel, tem ocasionado interesse em diversas áreas do conhecimento perante tecnologias atreladas à internet das coisas. Nesse sentido, objetivou-se desenvolver um aplicativo para smartphones com sistema operacional Android, para o controle e o monitoramento de dispositivos para irrigação. Para o desenvolvimento do aplicativo utilizou-se a interface de desenvolvimento integrado Android Studio 4.2.1, sendo aplicado ao desenvolvimento lógico a linguagem JAVA e na escrita do componente visual, a linguagem XML. Na execução do aplicativo desenvolvido, foram empregues um módulo relé conectado a um ESP8266-01 para o controle de uma eletrobomba e um sensor medidor de fluxo associado a uma placa responsável pela coleta e envio de dados do sensor. Na avaliação de desempenho do aplicativo, foram instalados uma eletrobomba e um sensor de vazão, dando-se início aos testes de acionamento e desligamento da bomba, assim como, o acionamento programado, aquisição, requisição e exposição dos dados do sensor de vazão, obtidos em tempo real. Finalizados os testes, pode-se concluir que, o aplicativo foi capaz de estabelecer uma conexão estável com o servidor, tanto no controle da eletrobomba, quanto no monitoramento do sensor de fluxo.

PALAVRAS-CHAVE: ESP8266, internet das coisas, java

DEVELOPMENT OF AN ANDROID APPLICATION FOR CONTROL AND MONITORING OF IRRIGATION DEVICES

ABSTRACT: The growing use of smartphones in the world, together with the expansion of the mobile internet network, has caused exorbitant interest in various areas of knowledge of technologies linked to the internet of things. In this sense, the objective was to develop an

¹ Estudante de Agronomia, Universidade Federal do Ceará - UFC. Fortaleza, Ceará. Fone (85) 994335302. e-mail: arthursrbreno@gmail.com

² Mestranda em Engenharia Agrícola, Universidade Federal do Ceará - UFC. Fortaleza, Ceará

³ Prof. Doutor, Depto de Engenharia Agrícola, Universidade Federal do Ceará – UFC. Fortaleza, CE

application for smartphones with the Android operating system, capable of controlling and monitoring irrigation devices. For the application development it uses Android Studio 4.2.1 integrated development environment, JAVA language was applied to the logical development, and in the writing of the visual components, the XML language was used. In the execution of the developed application, a related module connected to an ESP8266 was used to control an electric pump and a board responsible for collecting and sending data from the flow sensor. In the application performance evaluation, an electric pump and a flow sensor were installed, starting the pump activation and shutdown tests, as well as the programmed activation, acquisition, request and display of flow sensor data, obtained in real time. Once the tests are completed, it can be concluded that, the application was able to establish a stable connection with the ThingSpeak website server, both in pump control and sensor monitoring.

KEYWORDS: ESP8266, internet of things, java

INTRODUÇÃO

O acesso à telefonia móvel vem se popularizando ao passo que se amplia a difusão da internet. Segundo o Instituto Brasileiro de Geografia e Estatística (2021) em 2019 cerca de 98,6% da população brasileira são utilizátrias desses dispositivos móveis. Outro importante dado é que em 2019, 100% da população acima de 10 anos possuía acesso ao uso de smartphones para uso pessoal (IBGE, 2021). Isto deve-se à facilidade na aquisição e portabilidade dos dispositivos, bem como, na acessibilidade de acesso a redes sem fio.

Considerando que os dispositivos móveis podem caber na palma da mão, a portabilidade tem tornado viável a comunicação destes com diversos outros equipamentos, proporcionando ao usuário a interação desta ferramenta com eventos do mundo físico (SANTANA et al., 2012). Além da autonomia, Trinta et al. (2021) revela que sistemas inteligentes podem associar-se a tecnologias de transferência de informações e gerenciamento remoto para monitoramento, sendo um destes, a Internet das Coisas (Internet Of Things - IOT).

Em síntese, o IoT é a conexão entre objetos do cotidiano com a internet, proporcionando a evolução tecnológica através do uso de nanotecnologia, wirelles e sensores (CARVALHO & SOUZA, 2016). Entretanto, para estabelecimento da conexão entre servidores remotos, aplicações e sensores é necessário o desenvolvimento de uma interface de programação de aplicação (API), a fim de possibilitar comunicação entre duas diferentes aplicações, sistemas ou bancos de dados. O sítio ThingSpeak disponibiliza API's gratuitos permitindo requisitar e adicionar dados aos servidores para posterior análise.

Dentre as tecnologias emergentes, a Internet das coisas tem garantido a maximização da produção rural, tanto em produtividade quanto em qualidade, contribuindo para uma melhor gestão da propriedade, visto o monitoramento ser em tempo real (ROSSONI et al., 2019), elevando a assertividade na tomada de decisões e tornando ágil a resolução de problemas (VASCONVELOS, 2018).

De acordo com Santos et al. (2020), desperdícios recorrentes no campo, como, lâmina de irrigação aplicada em excesso, adubação ou aplicação de agrotóxicos de forma prescindível, entre outros processos, poderiam ser evitados se adotado sistemas inteligentes. Isso torna os processos mais simplificados, eficientes e rentáveis para o produtor agrícola (SANTOS, et al., 2020). Visando contribuir na agricultura digital de baixo custo, objetivou-se desenvolver um aplicativo para smartphones com sistema operacional Android, para o controle e o monitoramento de dispositivos para a irrigação.

MATERIAL E MÉTODOS

O aplicativo desenvolvido denominado IoTirrig, objetiva auxiliar no monitoramento de sensores e controle de módulos, realizando o recebimento de dados, exibição e envio a servidores para o acionamento ou desativação de dispositivos. Para seu desenvolvimento, utilizou-se a interface de desenvolvimento integrado (Interface development Environment - IDE) Android Studio 4.2, adotando-se a linguagem de programação Java no desenvolvimento lógico e linguagem de marcação XML (Extensible Markup Language) para o desenvolvimento visual do aplicativo. Na comunicação com o servidor remoto, foi utilizado o módulo ESP8266-01, configurado com o auxílio de um módulo adaptador gravador.

Para monitoramento dos dados trafegados foi desenvolvido uma placa IoT acessível, com o custo de 18,32 R\$. Esta contém duas portas associadas a bornes, tornando possível a comunicação de até dois sensores digitais. A placa é controlada pelo ESP8266-01, possibilitando o controle da eletrobomba e envio de dados coletados pelos sensores de vazão para servidores remotos. A alimentação da placa era proveniente de um módulo conversor serial micro USB.

Para efetuar corretamente o controle da eletrobomba, o módulo ESP8266-01 foi programado para fazer solicitações periódicas ao servidor, coletando os dados enviados pelo aplicativo e definindo o tempo de acionamento e desligamento da eletrobomba (Figura 1).

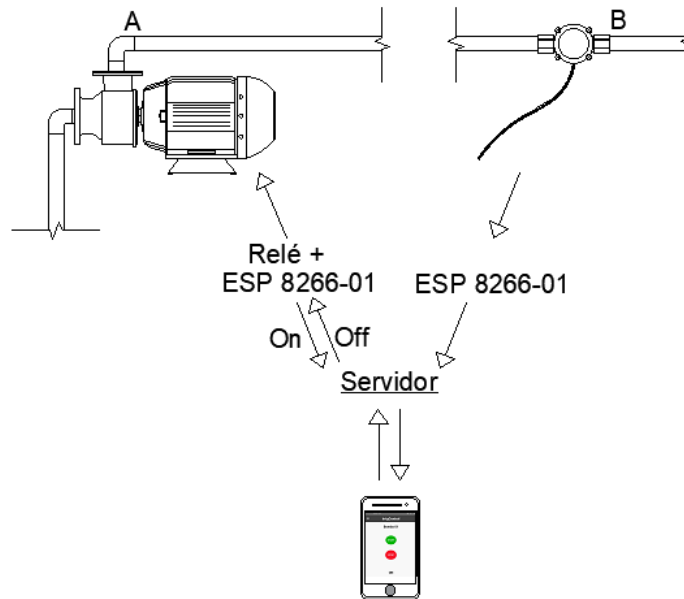


Imagem: Autor.

Figura 1. Esquema do teste em campo A: Eletrobomba. B: Medidor de Fluxo.

Para isso, foi elaborado o código de controle manual da bomba através do aplicativo e outro código para agendamento do controle automático. No início da função loop(), o esp8266 foi programado para realizar uma requisição ao canal e ao campo especificado pelo usuário, em seguida, o dado coletado foi analisado pelo algoritmo.

Caso o dado requisitado pelo ESP8266 seja zero, a bomba permanecerá desligada. Apenas havendo envio de outro comando pelo usuário no aplicativo, a bomba será ativada (Figura 2).

```

if(rele == "1" || rele == "0"){
  if (rele=="1") {
    digitalWrite(portaEsp8266 , LOW);
    Serial.println("ligado");
  }
  if (rele=="0") {
    digitalWrite(portaEsp8266 , HIGH);
    Serial.println("desligado");
  }
  delay(4000);
}

```

Imagem: Autor.

Figura 2. Código fonte para o acionamento manual da bomba pelo aplicativo IoTIrrig.

Dessarte, caso o primeiro caractere seja “a”, a bomba será ligada no modo Timer, caso o caractere seja “b”, a bomba será ligada no modo loop. Para o modo Timer a bomba será ligada somente durante o tempo que o usuário estipulou no aplicativo, sendo desligada e enviada uma informação para o canal selecionado, saindo do modo Timer, para o modo loop a bomba será ligada e desligada conforme tempo estipulado pelo usuário do aplicativo (Figura 3).

```

void loop(){
  while (WiFi.status() != WL_CONNECTED) { delay(500); }
  String rele = ThingSpeak.readStringField(idCanal, campoThingSpeak );

  if(rele.substring(0,1) == "a"){
    String subRele = rele.substring(1);
    int subIntRele = subRele.toInt();
    relógio = subIntRele * 1000;
    digitalWrite(portaEsp8266, LOW);
    delay(relógio);
    ThingSpeak.writeField(idCanal, campoThingSpeak, 0, apiKeywrite);
  }
  if(rele.substring(0,1) == "b"){
    int posicaoC = rele.indexOf("c");
    String primeiraParte = rele.substring(1, posicaoC);
    String segundaParte = rele.substring(posicaoC+1);
    int subIntReleParte2 = segundaParte.toInt();
    int subIntReleParte1 = primeiraParte.toInt();
    loop1 = subIntReleParte1 * 1000;
    loop2 = subIntReleParte2 * 1000;
    digitalWrite(portaEsp8266, LOW);
    delay(loop1);
    digitalWrite(portaEsp8266, HIGH);
    delay(loop2);
  }
}

```

Imagem: Autor.

Figura 3. Código para agendamento do controle automático para acionamento da bomba.

Foi utilizada a função `attachInterrupt`, a fim de que o ESP8266 contabilizasse o número de rotações do sensor de fluxo. A cada cinco segundos, o número de revoluções era armazenado e enviado para o campo no referido canal escolhido pelo usuário (Figura 4).

```

void IRAM_ATTR pulseCounter(){
  pulse_freq++;
}
void setup()
{
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  WiFi.begin(RedeWiFi, Senha);
  while (WiFi.status() != WL_CONNECTED) {delay(500);}
  ThingSpeak.begin(client);
  pinMode(2, INPUT_PULLUP); // habilita o attachInterrupt
  attachInterrupt(digitalPinToInterrupt(2), pulseCounter, FALLING);
}

void loop ()
{
  flowRate = ((5000.0 / (millis() - previousMillis)) * pulse_freq);
  pulse_freq = 0;
  previousMillis = millis();
  while (WiFi.status() != WL_CONNECTED) {delay(500);}
  ThingSpeak.writeField(idCanal, campoThingSpeak, flowRate, apiKeywrite);
  Serial.println(i++);
  delay(5000);
}

```

Imagem: Autor.

Figura 4. Código para monitoramento de recebimento de dados pelo aplicativo.

No seguimento de monitoramento (Figura 5), o módulo ESP8266-01 foi programado apenas para ler as informações do sensor de vazão e enviar para o servidor remoto do ThingSpeak a cada 5 segundos.

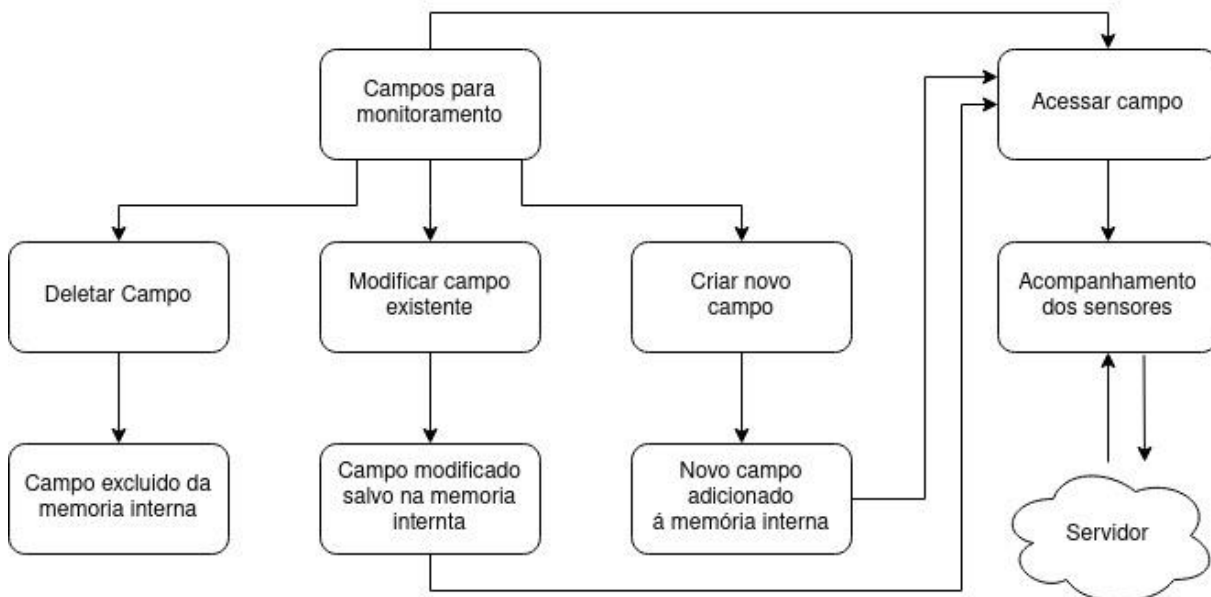


Imagem: Autor.

Figura 5. Esquema de seguimento para monitoramento de sensores.

O servidor ThingSpeak conectado ao aplicativo IoTirrig é uma plataforma analítica IoT, que permite armazenar, visualizar em tempo real e analisar dados na nuvem. A comunicação pode ser realizada através de computadores, smartphones e placas simples com acesso a internet, sendo utilizado no presente estudo o módulo ESP8266-01. O acesso para a inserção e leitura de dados é realizado através de API's disponibilizados pelo próprio site, facilitando o desenvolvimento de programas para a comunicação automatizada.

Dessa forma, para utilizar o aplicativo, deve-se acessar o sítio <<https://thingspeak.com/>>, abrir conta de usuário no referido sítio do ThingSpeak e em seguida gerar um canal definindo o nome e quantidade de campos conforme interesse do usuário.

Gerado o canal, é necessário acessar as configurações de compartilhamento do canal na aba "Sharing" modificando a privacidade do mesmo, de privado para público. Essa alteração se faz necessária, pois possibilita a comunicação do aplicativo IoTirrig ao servidor ThingSpeak. No acesso a aba "API Keys", o usuário poderá copiar as API's para aquisição e envio de dados, as quais serão utilizadas pelo aplicativo IoTirrig e para a configuração do ESP8266-01 através da IDE do Arduino.

RESULTADOS E DISCUSSÃO:

O principal resultado deste trabalho tange o desenvolvimento do aplicativo IoTirrig, sendo este dividido em dois segmentos: o primeiro é dedicado ao acionamento/ desligamento manual ou agendado da eletrobomba e, o segundo é referente ao monitoramento de recebimento de dados pelo aplicativo. Para o acionamento manual o usuário deve habilitar o botão “START”, ao passo que para cessar o funcionamento o mesmo deve ser realizado no botão “STOP” (Figura 6A). Para acionamento e desligamento agendado, as telas referentes as figuras 6B e 6C dispõem de campos para definição do tempo de funcionamento da eletrobomba, pelo usuário do aplicativo.

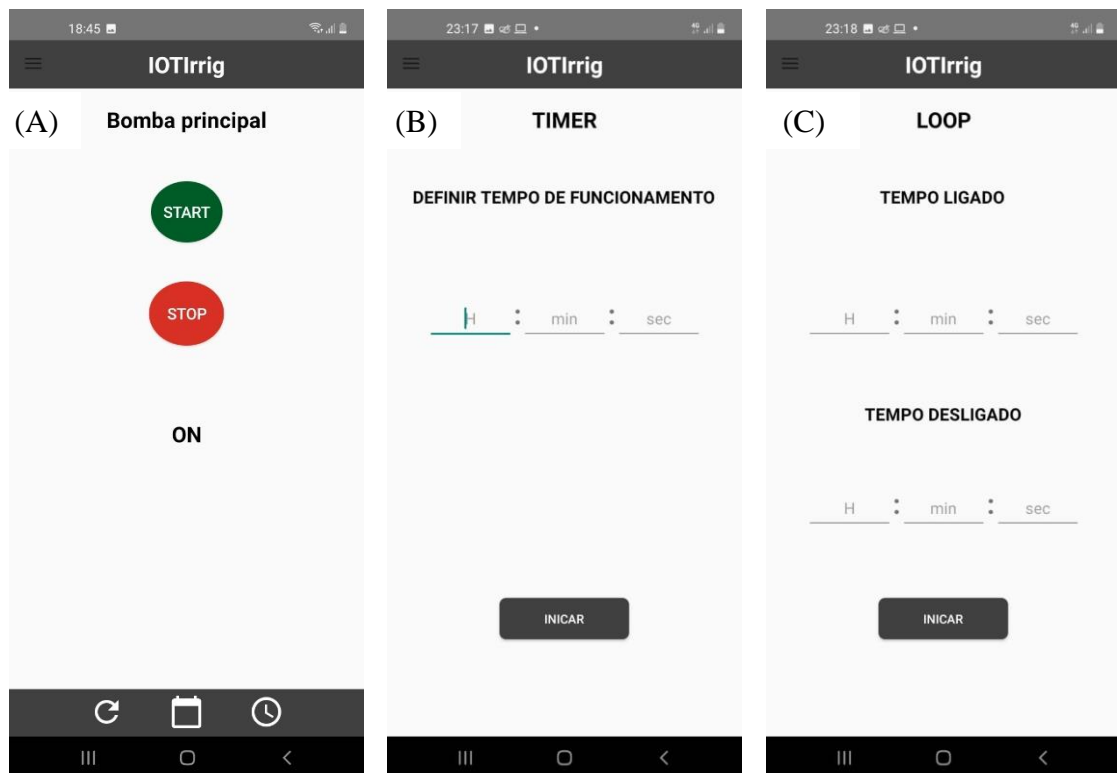


Imagem: Autor.

Figura 6. Guias do aplicativo: (A) tela de acionamento e desligamento manual, (B e C) acionamento e desligamento agendado.

Na guia de monitoramento dos sensores (Figura 7A), é possível verificar a temperatura do ar adquirida pelo sensor dht (35,96°C) e abaixo a umidade relativa do ar (65,0%). Vale ressaltar que o aplicativo possibilita o monitoramento de até cinco sensores com distintas funcionalidades, simultaneamente. A ferramenta encontra-se disponível na loja da Google Play Story no seguinte link: <https://play.google.com/store/apps/details?id=com.Getai.IOTirrig> (Figura 7B).

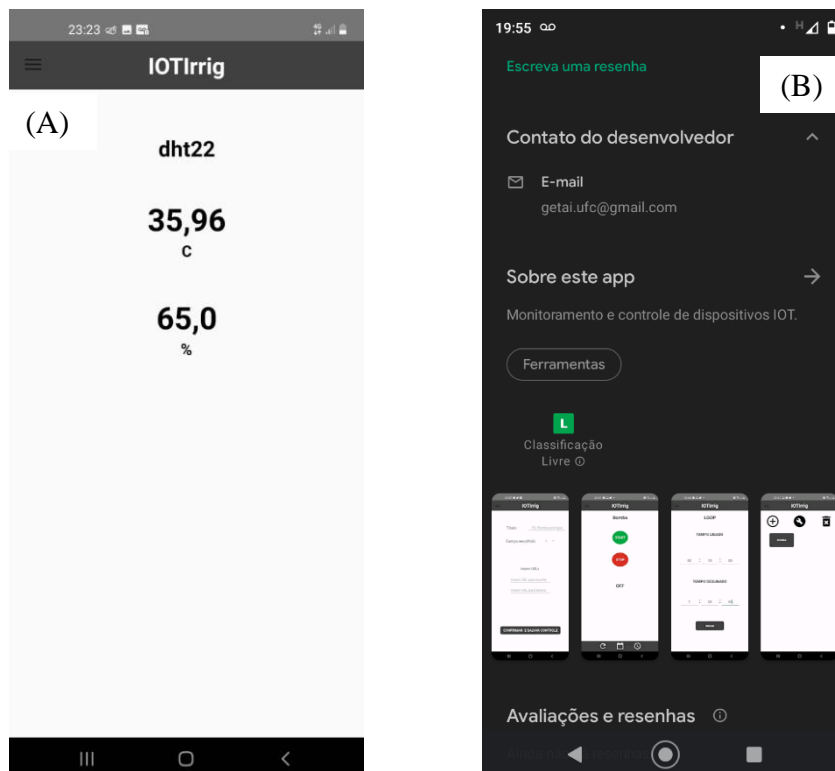


Imagem: Autor.

Figura 7. Guias do aplicativo: (A) Monitoramento dos sensores, (B) apresentação do aplicativo na Play Store.

O aplicativo demonstrou-se capaz em estabelecer conexão estável com o servidor, tanto para o controle da eletrobomba, quanto para o monitoramento dos sensores.

CONCLUSÕES

O aplicativo demonstrou resposta satisfatória no controle remoto da eletrobomba e monitoramento do sensor de vazão YF-S201, através de quaisquer smartphone com sistema operacional Android versão 4.3 ou superior.

A placa IoT de baixo custo desenvolvida, foi capaz de coletar os dados do sensor de forma constante e enviar para o servidor.

REFERÊNCIAS

CARVALHO, T.; SOUZA, T. L. Internet das coisas e sua aplicação em bibliotecas. **Revista Gestão**. v. 13, p. 264-270, 2015.

IBGE - Instituto Brasileiro de Geografia e Estatística. **Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal 2019**. Informativo. 2021.

ROSSONI, R. L. C.; ROSSONI, A. L.; MAIA, J. C. S. A Internet das Coisas e o Agronegócio no Brasil: um olhar sobre a produção científica brasileira na base Scopus. **Cadernos de Prospecção**, v. 12, n. 5, p. 1320, 2019.

SANTANA, C. S.; PONTES, I. G.; NUNES, M. A. S. N.; SILVA, R. X. Aplicando traços de acessibilidade e usabilidade web móvel na Universidade Federal de Sergipe: respeito à cidadania e à inclusão digital. **Revista Gestão Inovação e Tecnologias**. v. 2, n. 5, p. 445-464, 2012.

SANTOS, I. B.; SANDMANN, A.; SOUZA, B. E.; SCHMIDT, C. A. P.; PAULA FILHO, P. L.; MELGES, A. I.; MARCOLIN, J. F. Projeto e implementação de um gateway de internet das coisas (IoT) para otimização e monitoramento de processos do agronegócio. **Brazilian Journal of Development**, v. 6, n. 1, p. 344-369, 2020.

TRINTA, M. A.; CESAR, F.; PANTOJA, C. E. (2021). Gerenciamento Autônomo de uma Agricultura Utilizando Sistema Multiagente em uma Arquitetura IoT. **Revista Eletrônica De Iniciação Científica Em Computação**. v. 19, n. 1, p. 9, 2021.

VASCONCELOS, M. A Era da Agricultura 4.0. **Revista Fonte**. Belo Horizonte, v. 15, n. 20, p. 85-89, 2018.