Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

INTEGRAÇÃO DE SISTEMA *MOBILE* COM SISTEMAS QUE UTILIZAM A PLATAFORMA ARDUINO APLICADOS À AGRICULTURA

MOBILE SYSTEM INTEGRATED TO ARDUINO PLATFORM SYSTEMS APPLIED TO AGRICULTURE

Bruno Marques de Almeida[1] Jeferson dos Santos Souza[1] José Ruy Sanches[1] Junior[3]Henrique Victor Oliveira[1] Vivian Toledo Santos Gambarato[2]

RESUMO

Com o rápido avanço da tecnologia, o surgimento de novas demandas é praticamente inevitável. O cotidiano das pessoas tem mudado também nos últimos tempos com relação ao aumenta da quantidade de tarefas realizadas e pouco tempo para fazê-las. Assim, buscam o desenvolvimento de soluções que facilitem a interação do indivíduo com o mundo, o que se tornou um desafio para as empresas de desenvolvimento de sistemas. Levando-se em consideração essa necessidade, foi realizado o levantamento de requisitos necessários para a criação e implementação de um sistema que interliga as plataformas, Arduino *e Android,* com a utilização do módulo *Wifi* ESP8266 e *Firebase Database*, que por sua vez recebe as informações de um sensor de pH em um sistema automatizado na plataforma Arduino e que trata essas informações em tempo real. Para o desenvolvimento, foram utilizadas as linguagens de programação Java e C, e como base de dados foi utilizado o *Firebase Realtime Database*, para interação entre o aplicativo *Android* e o sistema automatizado. O Sistema em questão se mostrou eficiente para o problema proposto e as informações manipuladas no Arduino e enviadas pelo módulo *Wifi* ESP8266 ao servidor criado no *Firebase*, foram encaminhadas quase que instantaneamente e fiéis ao valor real do sensor.

Palavras-chave: Java. Linguagem C. Linguagem de programação. Módulo. Sensor Ph. Tecnologia da Informação.

ABSTRACT

With rapid advance of technology, the emergence of new demands is almost inevitable, for it influences people's daily life. Thus, seeking for development of solutions for such demands and in order to facilitate the integration of the individual and the world, system developing companies are being challenged to fulfill such demands. Considering this necessity, a survey of needed requirements was carried out for creating and implementing as system that connects Arduino and Android platforms, using Wifi ESP8266 module and Firebase Database, which in turn receives the information from a pH sensor in an automated system on the Arduino platform treating this information in real time. For this study, programming languages Java and C were used and as a database it was used the Firebase Realtime Database for interaction between the Android app and the automated system. Results showed that the system was efficient for the proposed problem and manipulated information in Arduino sent through Wifi module ESP8266 to created server in Firebase were sent almost instantly and were as well faithful to the real value of the sensor.

Key words: Information Technology. Java. Language C. Module. Programming language. Sensor Ph.

[1]Aluno de Graduação da Fatec Botucatu
[2] Docente FATEC Botucatu. Av. Ítalo Bachi, s/n. jardim Aeroporto. Email: vsantos@fatecbt.edu.br

Recebido em agosto de 2018 e aceito em agosto de 2019.

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

1. INTRODUÇÃO

As tecnologias de Agricultura de Precisão já são uma realidade no campo para os técnicos e produtores rurais (BERNARDI et al., 2014) e a introdução de processos automatizados de cultivo com o uso da tecnologia é um grande avanço para auxiliar na sustentabilidade da produção e controle de informações relevantes para a agricultura.

A Internet é uma das criações mais importantes em toda a história humana, e sua utilidade vem sendo constantemente aprimorada em todo o mundo (EVANS, 2011). O conceito de "Internet *of things*" (IoT) - (Internet das coisas, segundo *o Cisco Internet Business Solutions Group* (IBSG), aborda como o momento exato em que foram conectados à Internet mais "coisas ou objetos" do que pessoas (Evans, 2011). De acordo com essa afirmação, pode-se dizer que a Internet das coisas é a conexão e controle de coisas que normalmente não necessitam da Internet para funcionar.

Uma das plataformas que vem tendo um papel significativo em *IoT* é a plataforma Arduino, que tem a capacidade de automatizar processos rotineiros. O Arduino é um pequeno microcontrolador programável que processa entradas e saídas entre o dispositivo e os componentes externos conectados a ele (MCROBERTS, 2011). Um Arduino pode controlar diversos componentes que emitem dados ou podem ser controlados, como, bombas, relés acionadores, *led's*, *displays*, sensores, motores, dispositivos de GPS e até mesmo enviar dados à Internet.

Segundo a Google (2018), o *Android* é um sistema operacional e uma plataforma de programação para *smartphones* e outros dispositivos que suportam essa plataforma. A mesma empresa afirmou que o motivo pelo qual os desenvolvedores optaram por desenvolver aplicativos em *Android* foi atender requisitos de negócios, criando novos serviços e negócios e também fornecer jogos e outros tipos de conteúdo para os usuários da plataforma, visando alcançar a maioria dos usuários de dispositivos móveis, já que a própria empresa afirmou que esse sistema operacional é o mais utilizado no mundo, na plataforma mobile. O *Android* é um sistema operacional baseado no *Kernel Linux*, o *kernel* ou (*GNU* / *linux*) é o componente central da maioria dos sistemas operacionais e serve de ponte entre o sistema operacional e o *hardware*, garantindo assim o funcionamento e a comunicação entre ambos, durante a operação do sistema operacional (VAL, 2010).

As informações geradas pela automação podem ser inacessíveis, devido ao fato dos dados serem tratados de forma restrita, o que torna o levantamento de informações escasso ou inexistente. Tendo um sistema que receba essas informações e as gerencie com a utilização

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

dos recursos que os *Smartphones* proporcionam, sendo o acesso à Internet um deles, torna o processo automatizado mais completo, pois as informações se tornam acessíveis ao usuário.

Assim, o presente trabalho teve como objetivo a criação e integração de um aplicativo *Android*, para a medição do pH do solo, com um sistema automatizado que utilize a plataforma Arduino, realizando a integração dos dados coletados, entre o Arduino e o aplicativo *Android*, utilizando um servidor em nuvem, que permita um *feedback* das informações processadas pelo Arduino para o aplicativo *Android*, em tempo real, dando autonomia ao aplicativo sobre a automação, ou seja, permitindo que o aplicativo tenha ação efetiva relacionada a ações que a automação possa ter, alterando o estado de um componente físico da automação remotamente, via aplicativo.

2. MATERIAL E MÉTODOS

2.1 Material utilizado

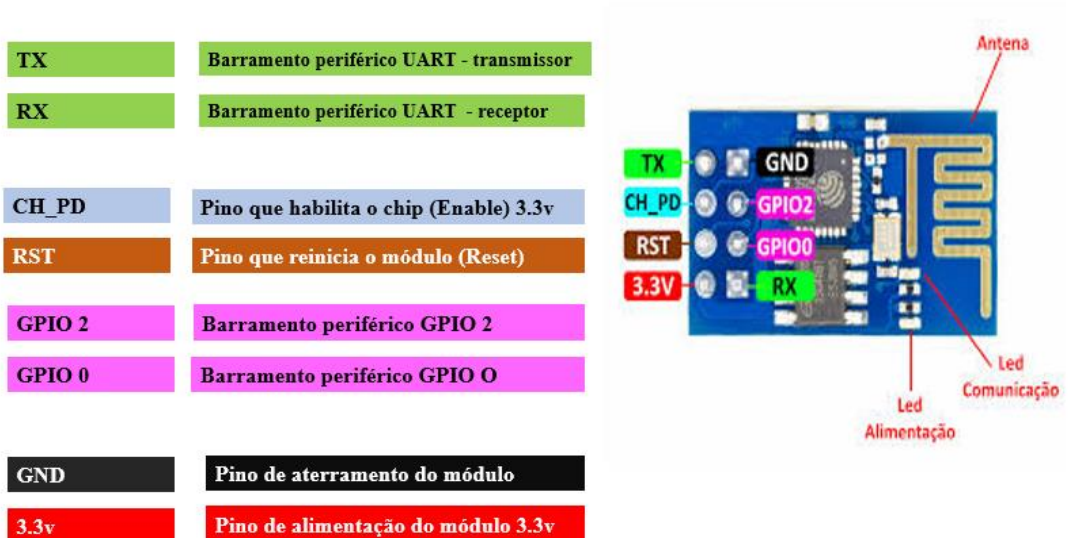
O sistema contou com a integração de três plataformas, sendo um projeto *Firebase*, utilizando o recurso *realtime database*, que é um elo entre a aplicação *Android* e o Módulo *Wifi* ESP8266 que, por sua vez, foi o responsável por receber os dados da automação via Arduino e enviá-los ao servidor. Nesse projeto, foram utilizados como recursos de hardware os seguintes componentes: 1 *protoboard* de 400 pinos para as conexões do circuito, 1 kit de *jumpers* para fazer as ligações do circuito, 1 módulo relé com dois canais, 2 resistores, sendo um de 100 ohms e outro de 220 ohms, *leds*, um notebook Acer Aspire ES 15, 1 pH eletrodo sonda BNC Arduino com módulo sensor (FIGURA 1) um módulo *Wifi* ESP8266.

Figura 1 - pH eletrodo sonda BNC Arduino e módulo sensor

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

O módulo em questão foi o modelo ESP8266-01, que contém 8 pinos, sendo dois deles para a alimentação do módulo que trabalha na tensão de 3.3v, outros dois para as funcionalidades do módulo sendo 1 CH_PD ou *chip enabled* que quando recebe a tensão de +3.3v ativa o *chip* do módulo e 1 RESET que reinicia o módulo quando necessário e os quatro pinos restantes foram de barramento periférico, utilizado para conexão entre outros dispositivos, sendo um TX que emite sinal para conexão serial e um RX que recebe sinal de conexão serial eos dois sendo "*General Purpose Input/Output Interface"* (GPIO) que são portas de propósito geral de entrada e saída. Além dos pinos, o módulo apresentava uma antena embutida e dois *leds* um que indicavam se o módulo estava ligado e outro que indicava a comunicação do módulo com outros dispositivos ou com a rede de Internet. A Figura 2 ilustra o módulo e suas respectivas portas.

Figura 2 - Módulo Wifi ESP8266



Fonte: Adaptado de FILIPEFLOP (2018).

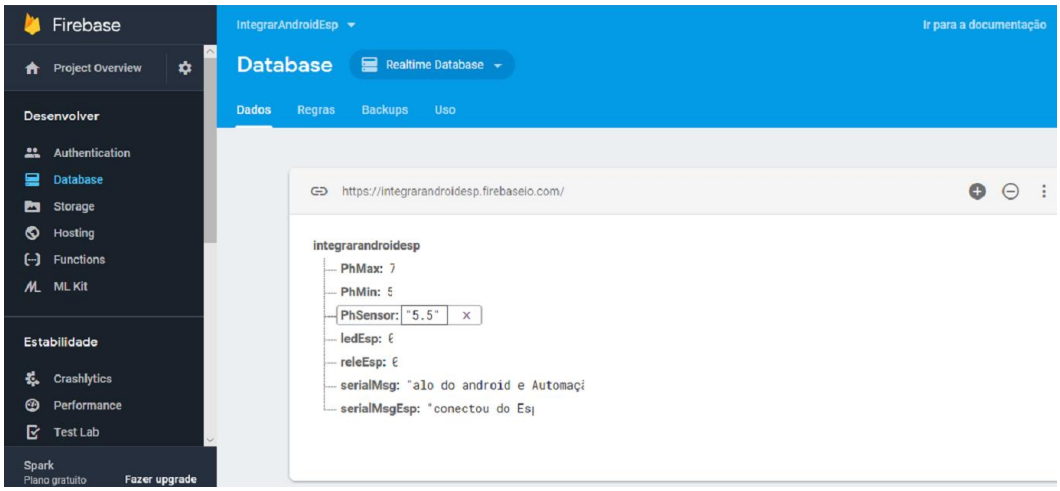
O microcontrolador do módulo Wifi Esp8266 possui 17 pinos GPIO, aos quais são atribuídas várias funções por programação. Os pinos de entrada e saída são bidirecionais, podendo ser multiplexados com outras funções, como modulação por largura de pulso (PWM), Transmissão e Recepção Assíncrona Universal (UART), etc. (ESPRESSIF, 2018).

Os recursos de software utilizados foram IDE Arduino para criação do código para o módulo Wifi ESP8266. Também foi utilizada a IDE *Android Studio* para criação do aplicativo, acesso ao site do *Firebase* para criação do banco de dados (FIGURA 3), para isto

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

basta obter uma conta de e-mail do Google e o software *Fritzing* para criação dos desenhos esquemáticos de ligações dos circuitos.

Figura 3 – Estrutura do Banco de Dados do Firebase



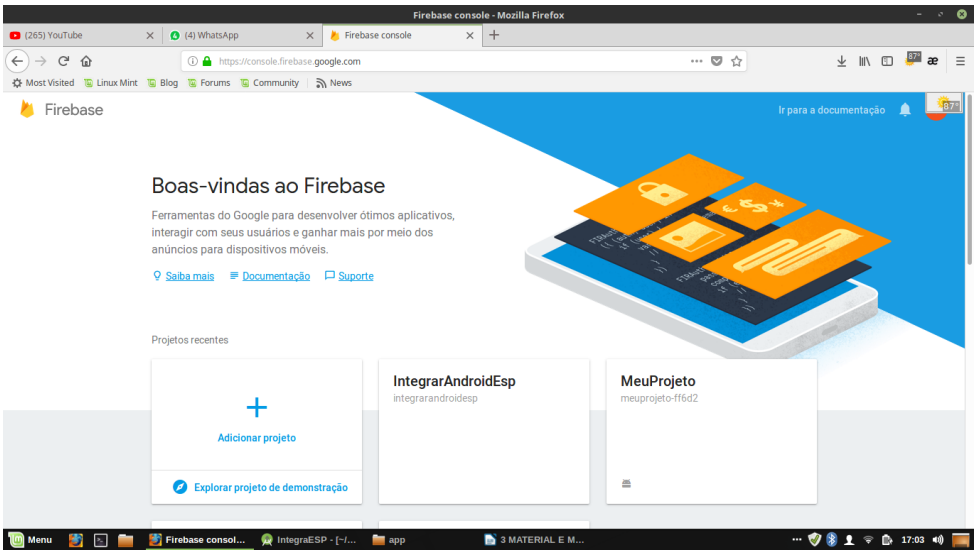
Outros recursos também foram utilizados como o acesso à rede de Internet, um roteador *Wirelles* para conexão do módulo *Wifi* ESP8266 à Internet e algumas bibliotecas para codificação dos sistemas. Para a IDE Arduino, usou-se as bibliotecas, *Firebase*-Arduino e *SoftwareSerial*, e para a IDE *Android Studio* o SDK do *Firebase* e a biblioteca *Realtime database*.

2.2 Metodologia

Para implementação da codificação, foi necessária a configuração das plataformas correlatas, para que o sistema se comunicasse de forma consistente. A primeira configuração foi a criação de um projeto no *Firebase*, cujo acesso é possível utilizando-se o login e senha de e-mail da *Google*. Uma vez na interface do sistema, foi adicionado um novo projeto, conforme demonstrado na Figura 4.

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

Figura 4 – Console do *Firebase*



Com um novo projeto criado, bastou-se adicionar um projeto *Android* a ele seguindo o passo a passo, segundo o *Firebase*. Com o projeto *Android* previamente criado e aberto no *Android Studio*, seguiu-se as instruções dadas pelo *Firebase*, para adição do aplicativo *Android* ao projeto *Firebase*, onde foi necessária a adição de algumas dependências e bibliotecas no aplicativo *Android. Ainda* foi necessária a adição do nome do pacote do projeto *Android* e o código SHA-1, uma das funções mais utilizadas dos sumários de mensagens, que é um esquema de autenticação que não exige criptografia da mensagem inteira e que também funcionam em sistemas de criptografia de chaves públicas (TANENBAUM; WETHERHALL, 2011).

Para adicionar as dependências do *Firebase* ao aplicativo, o site disponibiliza linhas de código e onde devem ser inseridas essas linhas no projeto do *Android Studio*. Além disso, o mesmo site disponibiliza um arquivo denominado '*google-services.Json'*, onde são adicionadas informações referentes à vinculação dos projetos *Firebase* e *Android*. Para cada adição diferente, a interface gera um arquivo JSON diferente, responsável por conter informações de requisição *Hyper Text Transfer Protocol* (HTTP), que segundo Tanenbaum e Wetheral (2011) é um protocolo de aplicação, que constitui a base da *World Wide Web*. Essa requisição HTTP é enviada ao endereço do projeto com as respectivas chaves de acesso e autenticação.

Para vincular o *Android* ao *Firebase,* foi necessário implementar a biblioteca de serviço, a *real time database*. Neste caso, a vinculação foi feita através do *Android Studio*, clicando-se em *tools/Firebase*, no menu *Assistance,* escolhendo a opção *Realtime Database* e

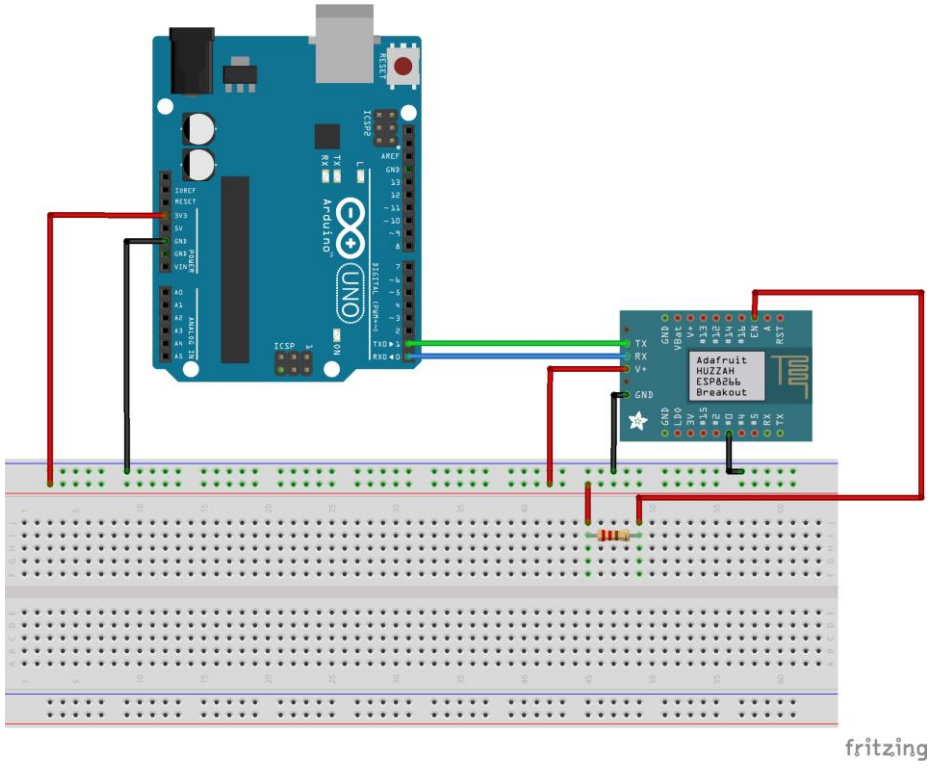
Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

clicando-se em '*Add the RealTime Database to your app', 'Accept changes'* e o próprio *Android Studio* implementa as bibliotecas necessárias. Após essa etapa, finalizou-se as configurações de utilização do aplicativo ao serviço *RealTime Database* do *Firebase*.

Para configurações do Módulo *Wif* ESP8266-01, foi necessário instalar o *firmware* do microcontrolador ESP8266, através do programa *Prolific Software Serial* versão 1.12.0, que atualiza o *firmware* do microcontrolador. Para a realização da comunicação serial entre o módulo e IDE Arduino foi utilizada a placa Arduino modelo UNO, seguindo o diagrama de montagem ilustrado na Figura 5, que também foi utilizado para carregar o código criado para memória *Erasable Programable Read Only Memory* (EPROM) do módulo ESP8266, na IDE do Arduino. Segundo Cristo et al (2013), EPROM () é uma memória só de leitura programável, que pode ser apagada e regravada.

Também foi necessário remover o microcontrolador do Arduino Atmega 328P (FIGURA 6) para que o ESP8266 utilizasse o conversor serial já existente, para carregar o *firmware* no microcontrolador ESP8266EX. A ligação entre seriais foi realizada através dos pinos RX (receptor) e TX (Emissor) e foi necessário conectar o pino RX do módulo ESP8266 no RX do Arduino, pois sua placa inverte a comunicação entre os terminais seriais, sendo que o RX do Arduino se interliga com o TX do conversor serial e vice e versa.

Figura 5 - Diagrama de ligação para atualizar *firmware* e carregar *sketch* Arduino

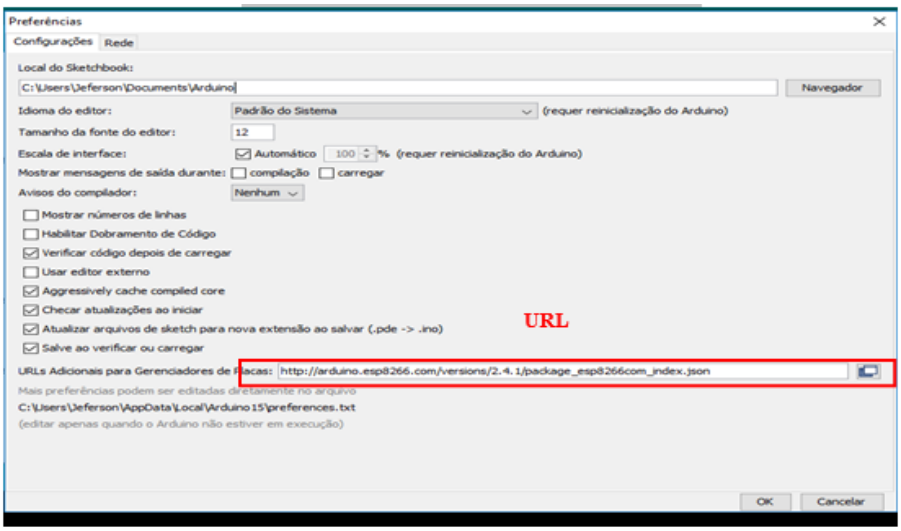


Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

Figura 6 - Microcontrolador ATMega328P

Também foi necessária a URL adicional para o gerenciador de placas da IDE Arduino, uma vez que ela não tem suporte às placas que contém o microcontrolador ESP8266EX (FIGURA 7).

Figura 7 -Inserir URL da Placa ESP8266



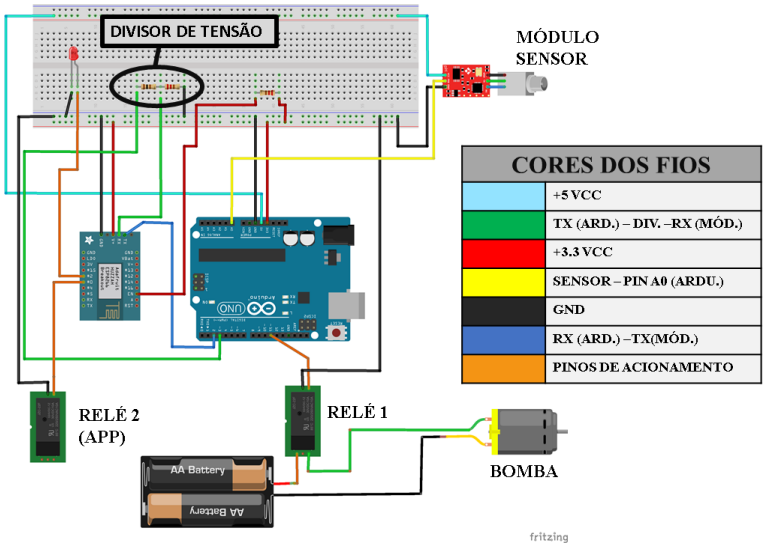
Após a *Uniform Resource Locator* (URL) ser inserida no gerenciador de placas, foi selecionada a placa *Generic ESP Module*. *T*ambém foi importada a biblioteca *Firebase* Arduino, disponível para download no site *GitHub*, para conexão do módulo com o Projeto *Firebase*, para fazer a conexão com a Internet. A conexão se inicializa pela função *wifi.begin( )*, que recebe como parâmetros o Service Set IDentifier (SSID) e a senha do roteador. Após a linha que estabelece a conexão com a internet, foi definida a linha de conexão com o *Firebase* pela função *Firebase.begin()*, que recebe como parâmetro o endereço de Host, que é o nome do projeto que se hospeda no *Firebase*, disponibilizado no menu *Database* do projeto. Outro parâmetro importante é a chave de autenticação do banco de dados disponível no diretório *Project Overview*/Configurações do projeto/Contas de Serviço/ Segredos do banco, finalizando a configuração de conexão entre o Módulo ESP8266 e o *Firebase*.

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

Para fazer a comunicação entre o Arduino e o Módulo, foi incluída a biblioteca *SoftwareSerial* no código do Arduino, declarando-se os pinos digitais 2 e 3 como pinos de comunicação serial, através da linha de comando '*SoftwareSerial* ESP8266(2,3)' onde o pino 2 do Arduino foi declarado como RX (receptor) que foi conectado ao TX (transmissor) do módulo *Wifi* ESP8266 e o pino 3 do Arduino como TX (transmissor) que foi conectado ao pino RX (receptor) do módulo.

O módulo funciona na tensão de 3.3v e o Arduino na tensão de 5v logo a tensão emitida do Arduino ao pino 3 TX também é de 5v, o que poderia queimar o módulo. Para se evitar esse problema, foi instalado um divisor de tensão, que converte a tensão de saída do pino 3 de 5v para 3.3v, na entrada RX do módulo. Esse divisor foi desenvolvido com dois resistores, um de 100 ohms e outro de 200 ohms, ligados entre si. Na extremidade do resistor de 100 ohms, têm-se a saída da porta digital 3, na outra extremidade do resistor de 200 ohm, o negativo e no ponto de interligação dos resistores, a tensão de saída foi de, aproximadamente, 3.3v, fazendo com que a comunicação serial entre o Arduino e o módulo ocorresse sem danificar os componentes. Para envio dos dados do Arduino ao módulo utilizou-se o nome do *Software Serial* definido no código adicionando a função *print*(), que envia uma mensagem manipulada pelo Arduino ao módulo. A Figura 8 apresenta o esquema de ligação para comunicação serial entre o Arduino e o módulo ESP8266 e também o divisor de tensão.

Figura 8– Diagrama para comunicação serial entre Arduino e módulo ESP8266



O módulo recebe essa informação, no caso o valor obtido pelo sensor do sistema e a envia ao *Firebase* a cada dois segundos pela função, *Firebase.set*(), que envia o nome do campo no banco de dados a ser alterado e o valor recebido pelo Arduino. Para verificação do

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

estado do *led* e relé, foi utilizada a função *Firebase.getInt()*, que passa como parâmetro o nome do campo do banco de dados a ser lido, nesse caso, um inteiro e faz uma condição, se o valor for um então o módulo aciona o *led* caso zero o *led* fica apagado.

Para conexão efetiva do aplicativo com o *Firebase,* foi preciso declarar dois objetos: um do tipo *FirebaseDatabase,* que recebe a instancia do banco de dados do *Firebase* e um objeto do tipo *DatabaseReference,* que recebe a referência do banco de dados do projeto *Firebase* vinculado ao aplicativo, e ainda outros objetos do tipo *DatabaseReference* referenciando cada um dos campos contido no banco de dados individualmente, para que quando um campo alterar seu valor, o aplicativo atualiza automaticamente o seu valor e exibe para o usuário, para a leitura em tempo real, no método *addValueEventListener*(), que contém dois métodos implementados, um *onDataChange*(), que toda vez que o valor é alterado no banco, esse evento é disparado e outro, *onCancelled*(), que é disparado quando não há comunicação entre a referência e o banco.

Finalizando esses passos, o módulo *wifi* ESP8266 conectou-se e autenticou-se com o servidor em nuvem do *Firebase*, recebendo e enviando dados para o mesmo e ainda foi conectado via comunicação serial com o Arduino, que é responsável pelo controle do sensor e manipulação de seus respectivos valores. Depois de configurada as dependências do aplicativo junto com as bibliotecas, o aplicativo estava pronto para receber informações do banco de dados e alterar alguns dados desse banco, como valores do campo *led* e *rele*, que recebem dois valores inteiros apenas, 0 para desligado e 1 para ligado, tanto o aplicativo quanto o módulo podem fazer a leitura em tempo real do banco e interagirem entre si.

3. RESULTADOS E DISCUSSÃO

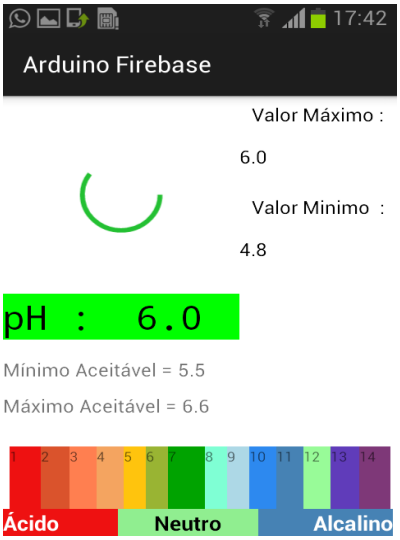
A tela principal é composta por 4 botões onde o primeiro aciona o relé e o segundo, o *led* o terceiro contém uma barra de progresso que altera seu valor conforme alterado no banco de dados e o último acessa à tela de leitura do sensor e demonstra o valor do sensor em tempo real (Figura 9).

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

Figura 9 – Tela principal

A Tela de leitura do sensor (FIGURA 10) também é composta por uma barra de progresso, que é alterada de acordo com a leitura valor do sensor e demonstra os valores máximo e mínimo lidos pelo sensor, além de informações de valores mínimo e máximo aceitáveis para a automação em específico, onde o valor do sensor deverá estar entre o mínimo e máximo aceitável. Ainda conta com uma tabela de coloração com a faixa de valores de pH, que o sensor suporta.

Figura 10 – Tela de Leitura do Sensor



O sensor utilizado apresentou uma leitura eficiente do pH de uma solução, levando um tempo entre 30 segundos e 1 minuto para estabilizar esta leitura, tempo observado com testes de soluções com pH 4,0 e 6,8.

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

Uma dificuldade encontrada foi a leitura dos dados do Arduino para o módulo, já que a comunicação serial envia uma informação de cada vez, ou seja, um caractere seguido do outro. Assim, foi preciso fazer estruturas de decisão tanto no módulo *Wifi* ESP8266, quanto no aplicativo *Android*, para verificar se a informação era a correta e atendia aos valores de leitura do sensor. O sistema trabalhou de forma consistente, pois alterando-se o valor doss dado no campo, os dados no aplicativo eram alterados automaticamente. Clicando no botão que se encontrava no aplicativo, alterava-se o estado do componente no sistema automatizado e vice-versa.

Com a utilização do módulo *Wifi* ESP8266 para enviar informações do Arduino ao *Firebase*, surgiram várias possibilidades de projetos, uma vez que com a plataforma Arduino, já existe um grande campo de exploração, além dos recursos de vinculação de projetos *Android*, IOS e *Web* ao *Firebase*, expandindo ainda esse campo de atuação para soluções tecnológicas.

O custo para realização do projeto foi relativamente baixo, de aproximadamente, R$ 350 reais, levando-se em consideração somente os componentes eletrônicos. Assim, para o agricultor que faz controle e medição de sensores automatizados com a plataforma Arduino, sua aplicação se torna viável, para áreas de pequena escala, como áreas domésticas ou de pequena e média escala de produção que tenha a facilidade no roteamento de sinal de Internet para o módulo *Wifi* ESP8266, já para áreas de escalas maiores de produção e rurais, onde se obtém mais de um reservatório de nutriente, se torna necessário estudar a viabilidade do roteamento do sinal de Internet até o módulo e expandir do projeto, adicionando outros modelos do módulo *Wifi* ESP8266, também disponíveis no mercado com até 11 barramentos periféricos, que aumenta o número de componentes a serem manipulados, pois o módulo utilizado para este projeto conta apenas com dois barramentos periféricos. Outra melhoria recomendada é a implementação de atuadores com bomba dosadora de produtos químicos, dando a possibilidade de tornar o nível do pH mais ácido ou alcalino pelo aplicativo.

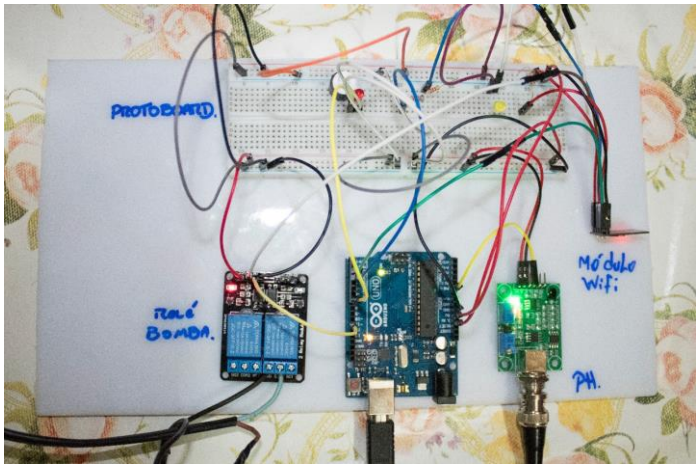
Uma vantagem de se utilizar o sistema, é saber os valores mínimo e máximo de leitura do sensor, porém para melhor eficiência e ações estratégias do agricultor, recomenda-se a melhoria desse recurso, como adicionar horário e data a essas leituras, para que esses dados, de alguma forma auxiliem o agricultor na tomada de decisões e até mesmo te tenha informações mais detalhadas dessas leituras.

Outra vantagem para utilização do projeto é a comodidade de verificação do valor do sensor quando e onde quiser, em tempo real, sem a necessidade de se estar no local, via aplicativo *Android* com *smartphones* ou *tablets*, além de poder acionar um componente físico,

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

como um relé, que poderia acionar uma bomba ou um motor, aumentando o tempo do agricultor para outras atividades e melhorando a sua interação e autonomia sobre esses componentes automatizados. Na Figura 11, observa-se o protótipo eletrônico e os respectivos componentes.

Figura 11 – Protótipo Eletrônico do Circuito



4. CONCLUSÃO

Com a facilidade para encontrar documentações a respeito das plataformas utilizadas, o projeto pode ser usado como base para outros, ou até mesmo adaptado, para outros sistemas automatizados com Arduino. O projeto atingiu os objetivos do problema proposto e teve sua eficácia comprovada. As informações veiculadas e manipuladas por ele se mostraram fiéis no aplicativo, em relação aos valores reais do sensor.

REFERÊNCIAS

BERNARDI, A. C. C, [et al.], editores técnicos. **Agricultura de Precisão:** Resultados de um Novo Olhar. – Brasília, DF: Embrapa, 2014. 596 p.

CRISTO, F.; PREUSS, E.; FRANCISCATTO, R.. **Arquitetura de computadores:** Frederico Westphalen, 2013.

ESPRESSIF. **ESP8266EX Datasheet version 4.3,** Jun, 2015. Disponível em: <https://www.espressif.com/en/support/download/documents?keys=> Acesso em: 29 mar. 2018.

EVANS, D. **A Internet das Coisas:** Como a próxima evolução da Internet está mudando tudo. - San Jose, CA: Cisco IBSG © 2011, Abr. 2011. Disponível

Tekhne e Logos, Botucatu, SP, v.10, n.2, set., 2019.
ISSN 2176 – 4808

em:<https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iot_ibsg_0411final.pdf > Acesso em: 25 mar. 2018.

FILIPEFLOP. **Módulo ESP8266 ESP 01**, 2018. Disponível em:<https://www.filipeflop.com/produto/modulo-wifi-esp8266-esp-01/> Acesso em : 05 abr. 2018.

GOOGLE. **Introduction to Android**, Dez. 2016. Disponível em:<https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-concepts/content/en/Unit 1/10_c_intro_to_android.html> Acesso em: 28 mar. 2018.

MCROBERTS, M. **Arduino básico** ; [tradução Rafael Zanolli]. -- São Paulo: Novatec Editora, 2011.

TANENBAUM, A. S; WETHERAL, D. J. **Redes de Computadores**: 5ª edição – São Paulo: Pearson Prentice Hall, 2011.

VAL, C. E. C. **Ubuntu**: Guia do Iniciante, 1ª ed.; Vitória, ES: Revista espírito Livre, 2010.