

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA  
FACULDADE DE TECNOLOGIA DE BOTUCATU  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E  
DESENVOLVIMENTO DE SISTEMAS**

**JEFERSON DOS SANTOS SOUZA**

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

Botucatu-SP  
Julho – 2018



**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**  
**FACULDADE DE TECNOLOGIA DE BOTUCATU**  
**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E**  
**DESENVOLVIMENTO DE SISTEMAS**

**JEFERSON DOS SANTOS SOUZA**

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

Orientadora: Prof<sup>a</sup>. M<sup>a</sup> Vivian Toledo Santos Gambarato

Artigo Científico apresentado à FATEC -  
Faculdade de Tecnologia de Botucatu, para  
obtenção do título de Tecnólogo no Curso  
Superior de Análise e Desenvolvimento de  
Sistemas.

Botucatu-SP  
Julho – 2018

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

**INTEGRATION OF MOBILE SYSTEMS WITH SYSTEMS USING ARDUÍNO  
PLATFORM APPLIED TO AGRICULTURE**

**Jeferson dos Santos Souza<sup>1</sup>, Vivian Toledo Santos Gambarato<sup>2</sup>**

**RESUMO**

Com o rápido avanço da tecnologia, o surgimento de novas necessidades é praticamente inevitável, devido ao fato de a mesma influenciar no cotidiano das pessoas, e ainda as buscarem, o desenvolvimento de soluções que facilite a interação do indivíduo com o mundo externo, tornou-se um desafio para as empresas de tecnologia. Levando em consideração essa necessidade, foram realizadas pesquisas para o levantamento de requisitos necessários para criação e implementação de um sistema que interliga as três plataformas, *Arduino* com a adição de um módulo *wifi* ESP8266, *Android* e *Firestore 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 *Firestore 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 *Firestore*, eram quase que instantâneas e fiéis ao valor real do sensor.

**PALAVRAS-CHAVE:** Linguagem C. Linguagem de programação. Java. Módulo. Sensor Ph. Tecnologia da Informação.

**ABSTRACT**

With the rapid advance of technology, the emergence of new needs is almost inevitable, due to the fact that it influences the daily life of people, and still seek them, the development of solutions that facilitate the interaction of the individual with the external world, has become a challenge for technology companies. Taking into account this need, research was carried out to determine the necessary requirements for the

creation and implementation of a system that is interconnected in three platforms, Arduino with the addition of an ESP8266, Android and Firebase Database wifi ESP8266 module, which in turn receives the information from a pH sensor in an automated system on the Arduino platform, and handles this information in real time. Java and C programming languages were used for the development, and as a database the Firebase Realtime Database was used for interaction between the Android application and the automated system. The system in question proved to be efficient for the proposed problem and the information manipulated in the Arduino and sent by the wireless module to the server created in Firebase were almost instantaneous and faithful to the real

**KEY WORDS:** Language C. Programming language. Java. Module. Sensor Ph. Information Technology.

---

<sup>1</sup> Graduando em Análise e Desenvolvimento de Sistemas na Faculdade de Tecnologia de Botucatu, jeferson169087@gmail.com

<sup>2</sup> Mestre e Docente na Faculdade de Tecnologia de Botucatu, vsantos@fatecbt.edu.br

## 1 INTRODUÇÃO

As tecnologias de Agricultura de Precisão já são uma realidade no campo para os técnicos e produtores rurais (BERNARDI et al., 2014). De acordo com essa afirmação, 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 de toda a história humana, afirma (EVANS, 2011) e sua utilidade vem sendo mais aprimorada em todo o mundo. O conceito de (Internet *of things* ou Internet das coisas) IoT, segundo o *Cisco Internet Business Solutions Group* (IBSG) citado por Evans (2011), aborda a IoT como o momento exato em que foram conectados à Internet mais "coisas ou objetos" do que pessoas. De acordo com essa afirmação, pode-se dizer que a Internet das coisas é a conexão e controle a coisas que normalmente não necessitam da Internet para funcionar.

Uma das plataformas que vem tendo um papel significativo em *IoT*, é a plataforma Arduíno, que tem a capacidade de automatizar processos rotineiros. “Em termos práticos, um Arduíno é um pequeno computador que se pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele” (MCROBERTS, 2011). Um Arduíno pode controlar diversos componentes que emitam dados ou possam ser controlados, tais como, bombas, relés acionadores, *led's*, *displays*, sensores, motores, dispositivos de GPS, ou até mesmo enviar dados a Internet.

Segundo a Google (2018), o *Android* é um sistema operacional e uma plataforma de programação para *Smartphones* e outros dispositivos que suportam a plataforma, a mesma empresa afirma que, o motivo pelo qual os desenvolvedores optam por desenvolver aplicativos em *Android* são 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 afirma que o sistema operacional *Android* 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* (VAL, 2010).

As informações geradas pela automação podem ser inacessíveis, devido ao fato de os 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 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*, com um sistema automatizado que utilize a plataforma *Arduino*, realizando a integração dos dados coletados, entre *Arduino* e o aplicativo *Android*, para isso, utilizando um servidor em nuvem, permitindo um *feed-back* das informações processadas pelo *Arduino* para o aplicativo *Android*, em tempo real o que dá autonomia ao aplicativo sobre a automação, ou seja, permite a que o aplicativo tenha ação efetiva relacionada às ações que a automação possa ter, alterando o estado de um componente físico da automação remotamente, via aplicativo. O aplicativo em questão teve como enfoque um sistema automatizado voltado à agricultura.

## **2 MATERIAL E MÉTODOS**

### **2.1 Material utilizado**

O presente trabalho foi resultado de um trabalho desenvolvido em equipe, para a matéria do 6º ciclo denominada Tópicos Especiais em Informática, do curso de Análise e desenvolvimento de sistemas da Faculdade de Tecnologia de Botucatu (Fatec), onde também foi criado outro trabalho cujo título é “Desenvolvimento de protótipo para automação de processo de medição de pH em plataforma *Arduino*”, que teve como parte do trabalho o sistema de medição de pH, como também a calibração do eletrodo e suas respectivas funcionalidades, cujo autor foi o aluno Bruno Marques de Almeida.

Para levantamento de informações relevantes para o projeto foram realizadas pesquisas em artigos de revistas, livros e via Internet, nos sites proprietários das plataformas *Android*, *Arduino* e *Firebase*, que por sua vez disponibilizam documentações sobre desenvolvimento, facilitando a criação do projeto. O sistema conta com a integração das 3 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. Para criação do projeto foram utilizados componentes

de hardware e alguns softwares para desenvolvimento das aplicações utilizadas no projeto, como recursos de hardware os seguintes componentes foram utilizados: 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), e por fim 1 módulo *Wifi* ESP8266.

Figura 1 - pH eletrodo sonda BNC Arduino + módulo sensor



O módulo em questão utilizado foi o modelo ESP8266-01, que contém 8 pinos apenas, sendo 2 pinos para alimentação do módulo que trabalha na tensão de 3.3v, 2 pinos 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 4 pinos de barramento periférico, utilizado para conexão entre outros dispositivos, sendo 1 TX que emite sinal para conexão serial e 1 RX que recebe sinal de conexão serial, e 2 (*General Purpose Input/Output Interface*) GPIO que são portas de propósito geral de entrada e saída. Além dos pinos ele contém uma antena embutida, e dois *leds* um que indica se o módulo está ligado e outro que indica 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

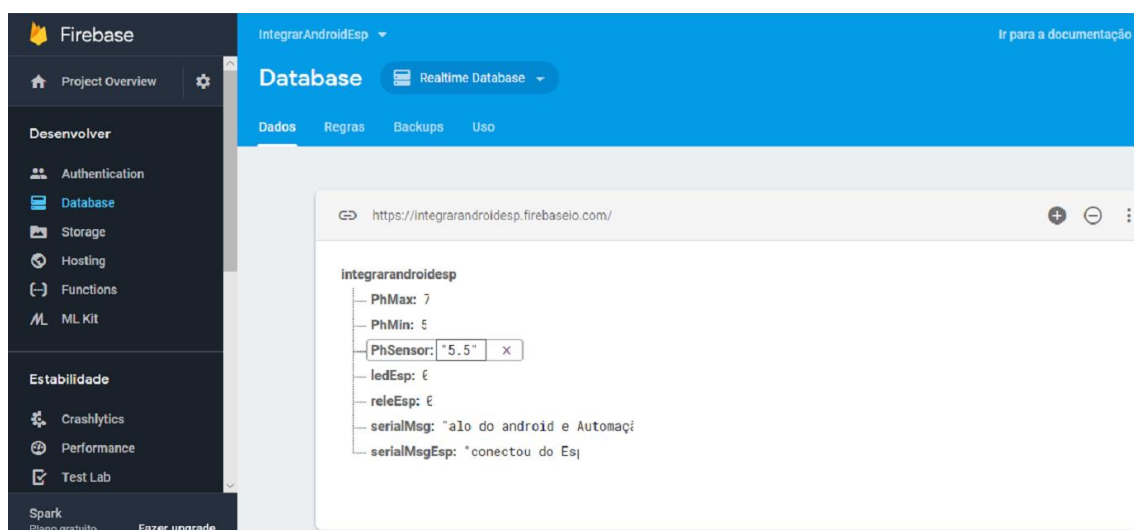


Fonte: Adaptado de FILIPEFLOP (2018).

O microcontrolador do módulo Wifi Esp8266 possui 17 pinos GPIO, podendo atribuí-los a várias funções por programação, os pinos de entrada e saída são bidirecionais, esses pinos podem ser multiplexados com outras funções, como PWM, 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 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

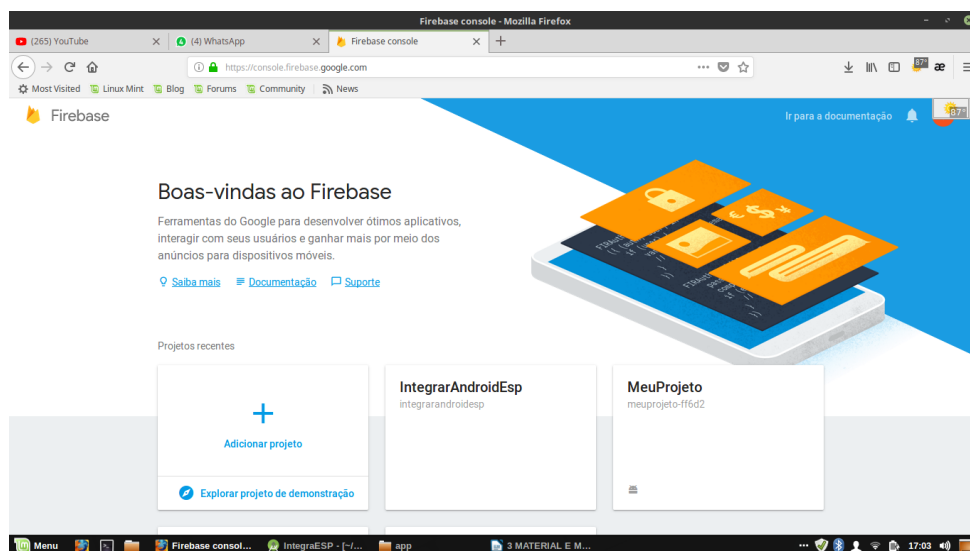


Outros recursos também foram utilizados como, acesso a rede de Internet, 1 roteador *Wireless* para conexão do módulo wifi ESP8266 à Internet e algumas bibliotecas para codificação dos sistemas. Para a IDE *Arduino*, tem-se as bibliotecas, *Firebase-Arduino* e *SoftwareSerial*, e para a IDE *Android 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*, entrando no site com login e senha de e-mail da *Google*, e após entrada no console do sistema, foi adicionado um novo projeto no *Firebase*, conforme demonstrado na Figura 4.

Figura 4 – Console do *Firebase*



Com um novo projeto criado, bastou adicionar um projeto Android a ele seguindo um passo a passo demonstrado pelo *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*, e ainda no projeto

*Firebase* também foi necessária a adição do nome do pacote do projeto *Android*, e o código SHA-1 que é uma das funções mais utilizadas dos sumários de mensagens, que é um esquema de autenticação que não exige criptografia da mensagem inteira, 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 de códigos, no projeto do *Android Studio*, além disso o mesmo site disponibiliza um arquivo denominado '*google-services.Json*' que adiciona nele informações referentes a vinculação dos projetos *Firebase* e *Android*. Para cada adição diferente, o *console* do gera um arquivo JSON diferente, este arquivo é responsável por conter informações de requisição 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 uma biblioteca de serviço utilizada pela aplicação, neste caso o *real time database*, essa vinculação foi feita através do *Android Studio*, clicando em *tools/Firebase*, no menu *Assistance* escolhendo a opção *Realtime Database* e clicando 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 realização da comunicação serial entre o módulo e IDE *Arduino* foi utilizado 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 EPROM do módulo ESP8266, na IDE do *Arduino* que, segundo Cristo et al (2013), EPROM (*Erasable Programable Read Only Memory*) 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 existente no *Arduino*, para

carregar o *firmware* no microcontrolador ESP8266EX, a ligação entre seriais foi através dos pinos RX (receptor) e TX (Emissor), foi necessário conectar o pino RX do módulo ESP8266 no RX do *Arduino*, pois a placa do *Arduino* 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, essa ligação é feita por emissor (TX) e um receptor (RX).

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

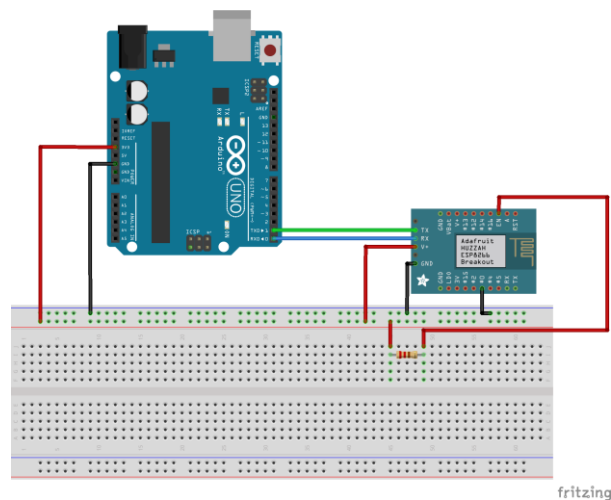
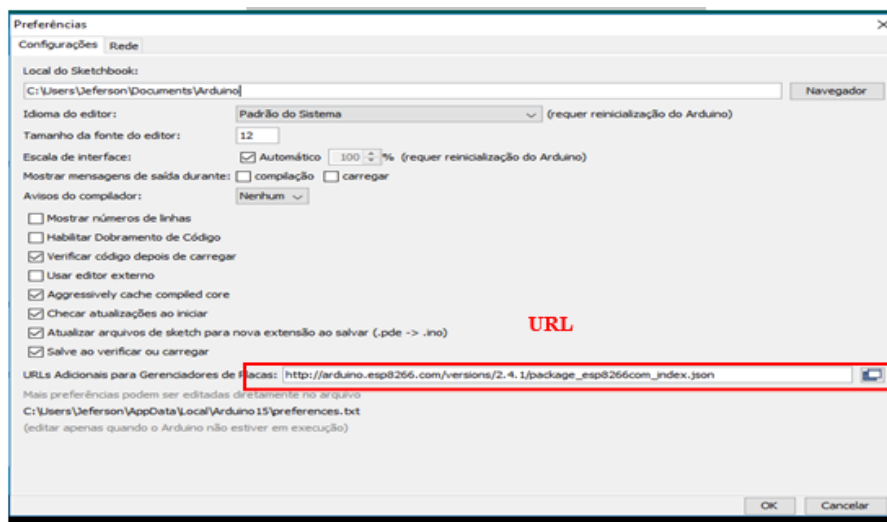


Figura 6 - microcontrolador ATmega328P



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



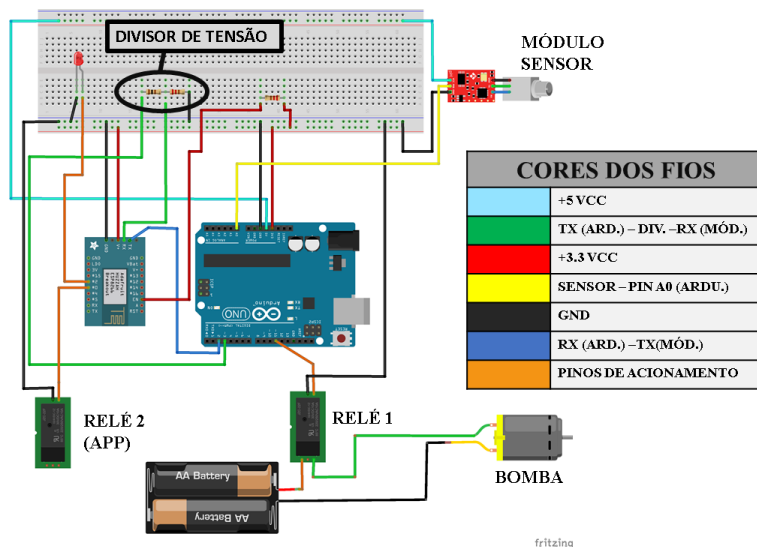
Após inserida a URL no gerenciador de placas, foi selecionada a placa *Generic ESP Module*, foi importada a biblioteca *Firebase Arduino* disponível para download no site *GitHub*, para conexão do módulo com o Projeto *Firebase*, para fazer a conexão com a *Internet* inicializa a conexão pela função *wifi.begin()*, que recebe como parâmetros o 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 endereço de Host que é o nome do projeto que se hospeda no firebase, disponibilizado no menu *Database* do projeto, e outro parâmetro que é a chave de autenticação do banco de dados disponível no diretório *Project Overview/Configurações* do projeto/*Contas de Serviço/ Segredos* do banco Finalizando então a configuração de conexão entre o Módulo ESP8266 e o *firebase*.

Para fazer a comunicação entre o *Arduino* e o Módulo foi incluída a biblioteca *SoftwareSerial* no código do *Arduino*, e declarando 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 *Arduíno* como TX (transmissor) que foi conectado ao pino RX (recptor) do módulo.

O módulo funciona na tensão de 3.3v e o *Arduino* na tensão de 5v, e a tensão emitida no pino 3 TX do *Arduino* é de 5v o que pode ocasionar a queima do módulo, para evitar esse problema foi instalado um divisor de tensão, para converter a tensão de

saída do pino 3 de 5v para 3.3v, na entrada RX do módulo, o divisor de tensão foi desenvolvido com dois resistores um de 100 ohms e outro de 200 ohms, ligados entre si e na extremidades do resistor de 100 ohms a saída da porta digital 3, e na outra extremidade do resistor de 200 ohm o negativo, e no ponto de interligação dos resistores a tensão de saída foi aproximadamente 3.3v, fazendo com que a comunicação serial entre o *Arduino* e o módulo ocorra sem danificar os componentes. Para envio dos dados do *Arduino* ao módulo utilizou-se o nome do *SoftwareSerial* definido no código adicionando a função *print()*, que envia uma mensagem manipulada pelo *Arduino* ao módulo, a Figura 8 demonstra 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 por sua vez recebe essa informação, no caso o valor obtido pelo sensor do sistema, e envia ao *Firestore* a cada dois segundos pela função, *Firestore.set()*, que envia o nome do campo no banco de dados a ser alterado e o valor recebido pelo *Arduino*. Para verificação do estado do *led* e relé, foi utilizada a função *Firestore.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 *Firestore* foi preciso declarar dois objetos um do tipo *FirestoreDatabase* que recebe a instancia do banco de dados do

*Firebase* é 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 automaticamente atualiza o seu valor e exibe para o usuário, para a leitura em tempo real é o método *addValueEventListener()*, que contém dois métodos implementados um *onDataChange()*, que toda vez que alterado o valor no banco esse evento é disparado, e outro  *onCancelled()*, é disparado quando não houver comunicação entre a referência e o banco.

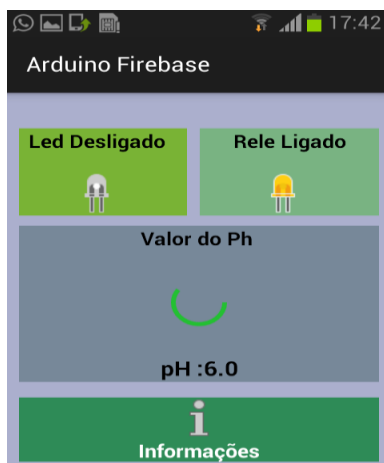
Finalizando esses passos o módulo *wifi* ESP8266, conectou-se e autenticou-se com o servidor em nuvem do *Firebase*, recebendo e enviando dados para o mesmo, e ainda foi conectado via comunicação serial com o *Arduino* que é responsável pelo controle do sensor, e manipulação de seus respectivos valores. E configurada as dependências do aplicativo junto com as bibliotecas o aplicativo está pronto para receber informações do banco de dados e alterar alguns dados do 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**

Algumas discussões e recomendações a respeito do trabalho e sua capacidade de utilização são descritas a seguir, além de descrever as telas do aplicativo e suas funcionalidades, neste capítulo também são citadas algumas vantagens que o projeto apresenta para o agricultor.

A tela principal é composta por 4 botões onde 1 é para o acionamento do relé, outro para acionamento do *led*, o terceiro contém uma barra de progresso que altera seu valor conforme alterado no banco de dados, e um que “chama” a tela de leitura do sensor ainda demonstra o valor do sensor em tempo real (Figura 9).

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 ainda demonstra o valor máximo e mínimo lido pelo sensor, e informações de valor mínimo e máximo aceitável para a automação em específico, onde o valor do sensor deverá estar entre o mínimo e máximo aceitável, e 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, para se obter esta conclusão foram utilizadas soluções com pH 4,0 e outra com pH de 6,8.



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, 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, alterando o valor do dado no campo, os dados no aplicativo era alterado automaticamente. E 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*, surgem várias possibilidades de projetos, uma vez que com a plataforma *Arduino*, já existe um grande campo de exploração, e ainda com os recursos de vinculação de projetos *Android*, *IOS* e *Web*, ao *Firebase*, expande ainda esse campo de atuação para soluções tecnológicas.

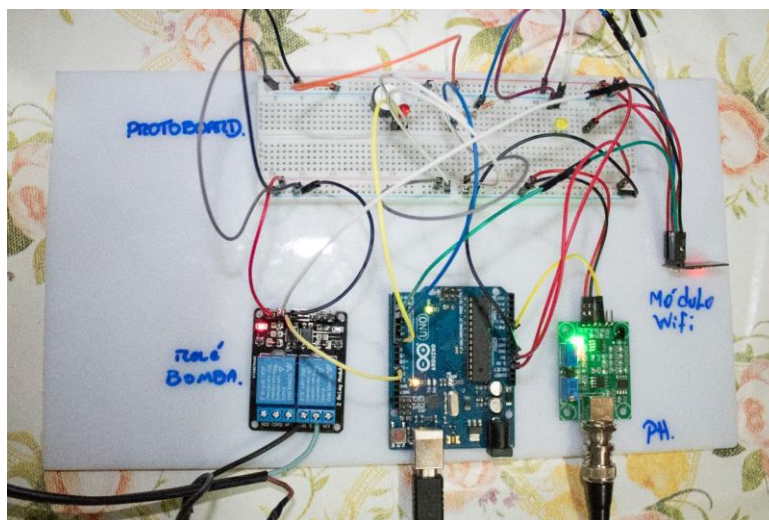
O custo para realização do projeto foi relativamente baixo, para o problema proposto, que foi de aproximadamente 350 Reais, levando em consideração somente os componentes eletrônicos, e para o agricultor que faz controle e medição de sensores automatizados com a plataforma *Arduino*, se torna viável sua aplicação, porém para áreas de pequena escala, para áreas de escalas maiores, 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 que recomenda-se 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 utilizar o sistema, é saber o valor 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 dessas leituras, para que esses dados, de alguma forma possam auxiliar o agricultor, na tomada de decisões e até mesmo ter informações mais detalhadas dessas leituras.

Outra vantagem para utilização do projeto, é a comodidade de verificar o valor do sensor quando e onde quiser, em tempo real, sem a necessidade de estar no local, e

ainda, via aplicativo *Android* com *Smartphones* ou *Tablets*, acionar um componente físico, como um relé, que posteriormente, acionando uma bomba, ou um motor etc., com isso aumentando o tempo do agricultor para outras atividades, e melhorando a sua interação e autonomia, sobre esses componentes automatizados. Temos a seguir na Figura 11 a demonstração do 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*, com fins semelhantes ou distintos. O projeto atingiu os objetivos do problema proposto, e teve eficácia nas suas ações, a respeito das informações veiculadas e manipuladas por ele, e sendo compartilhadas com o servidor em nuvem, 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 em:<[https://www.cisco.com/c/dam/global/pt\\_br/assets/executives/pdf/internet\\_of\\_things\\_iiot\\_ibsg\\_0411final.pdf](https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iiot_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](https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-concepts/content/en/Unit%201/10_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.