

AUTOMAÇÃO DE ESTUFA AGRÍCOLA UTILIZANDO NODEMCU E BLYNK

GREENHOUSE AUTOMATION UTILIZING NODEMCU AND BLYNK

SILVA VIEIRA, José Eduardo da¹
DEVOS, Cacildo José²

RESUMO

Utilizando recursos de IoT e Cloud Computing, é possível criar uma solução simples e de baixo custo para o que hoje se chama de Smart Farming. O módulo de desenvolvimento NodeMCU ESP-12F facilita a conexão de objetos à Internet e o aplicativo Blynk disponibiliza uma maneira intuitiva de criar dashboards customizadas para controlar qualquer dispositivo que esteja conectado à sua Cloud. Combinando, tanto o dispositivo quanto o aplicativo, concluiu-se com sucesso o objetivo principal deste projeto.

Palavras-chave: IoT; NodeMCU; Blynk; Smart Farming.

ABSTRACT

Utilizing IoT and Cloud Computing resources, it's possible to create a simple and low cost solution to what is called today Smart Farming. The NodeMCU ESP-12F Development Module makes it easier to connect objects to the internet and the Blynk App provides an intuitive way to create custom dashboards to control any device connected to its cloud. Combining, both device and app, it was successfully concluded the primary objective of this project.

Keywords: IoT, NodeMCU, Blynk, Smart Farming.

¹Graduando do Curso de Ciência da Computação da Universidade de Franca-joseedsv@hotmail.com;

²Professor orientador e coordenador do Curso de Ciência da Computação da Universidade de Franca – cacildo.devos@unifran.edu.br

LISTA DE ABREVIações

ADC	Analog-Digital Converter
CSV	Comma Separated Value
GPIO	General Purpose Input Output
HW	Hardware
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IOT	Internet Of Things
LDR	Light Dependent Resistor
RTC	Real Time Clock
OTA	Over The Air
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/ Transmitter

LISTA DE FIGURAS

Figura 1 – NodeMCU ESP 826612F.....	4
Figura 2 – Módulo Relé 4cns.....	4
Figura 3 – ADS1115.....	5
Figura 4 – Solenoide 12V NF.....	5
Figura 5 – DHT11 e Higrômetro.....	6
Figura 6 – Aplicativo Blynk.....	6
Figura 7 – Configuração de pinos dentro do aplicativo.....	7
Figura 8 – Receber informações dentro do HW.....	8
Figura 9 – Enviar informações ao aplicativo.....	8
Figura 10 – Função que cuida de manusear o Time Input no HW.....	9
Figura 11 – Configuração do Time Input no aplicativo.....	10
Figura 12 – Fluxo principal do protótipo.....	11
Figura 13 – History Graph Widget.....	13
Figura 14 – Estufa Montada.....	14
Figura 15 – Dashboard final.....	15

LISTA DE TABELAS

Tabela 1 – Custos (Eletrônicos).....	12
Tabela 2 – Custos (Widgets).....	13

SUMÁRIO

Introdução.....	3
Materiais e Métodos.....	4
Desenvolvimento.....	12
Resultados e Discussão.....	14
Considerações Finais.....	16
Trabalhos Futuros.....	16

INTRODUÇÃO

Como apontado em estudos realizados pela Organização de Alimentos e Agricultura das Nações Unidas, a população mundial deve ver um crescimento de cerca de 2,3 bilhões de pessoas até 2050 e como consequência, a produção de alimentos deverá ser pelo menos 70% maior em 2050 do que foi em 2006 (Food and Agriculture Organization, 2009).

Dados de uma pesquisa realizada pelo BI Intelligence mostram que os Estados Unidos são líderes em compras e instalações de dispositivos IoT na área de agricultura de precisão. Eles têm uma produção média de grãos quase 100% maior que outros países, 7,34 Toneladas por hectare, enquanto que a média mundial fica em torno de 3,85 Toneladas por hectare (BI Intelligence, 2016).

Mesmo com essa grande diferença de produção, alimentar todo o planeta não pode e nem deve ser tarefa de uma só nação. Buscar este aumento de produção não é tarefa fácil e, portanto, cada vez mais, produtores tem buscado soluções em IoT seja para aumentar a produtividade, para ter uma base de dados melhor para o auxílio na tomada de decisões, ou simplesmente para livrar a mão-de-obra de tarefas que podem perfeitamente ser realizadas por dispositivos eletrônicos.

O objetivo principal deste projeto é o desenvolvimento de uma solução em IoT para ser usada também por produtores de pequeno porte devido ao seu baixo custo de produção e manutenção, e a consequente integração dos mesmo às novas tecnologias.

MATERIAL E MÉTODOS

No desenvolvimento do protótipo foram levados em consideração dois aspectos: custo e eficiência. Todos os componentes elétricos e eletrônicos do projeto foram escolhidos com base nessa premissa.

Como plataforma escolheu-se o NodeMCU ESP-12F (Figura 1), microcontrolador que conta com 9 pinos GPIO, dos quais, 8 podem ser usados como PWM, 1 pino com ADC embutido e suporte à varios protocolos de comunicação, como por exemplo: UART, I2C e SPI. Possui regulador de tensão embutido e pode ser alimentado de 3.3V à até 20V. Foi escolhido para este projeto por sua facilidade de integração com projetos IoT e por ser bastante compacto.



Figura 1 – NodeMCU ESP12F

Fonte: (www.robocore.net)

Para o manuseio de cargas de voltagem superior às suportadas pela placa, foi utilizado um módulo relé de 4 canais (figura 2). Com ele é possível controlar o acionamento de qualquer carga até 250V. O módulo já traz em seu setup todos os componentes necessários para uso em micro controladores.

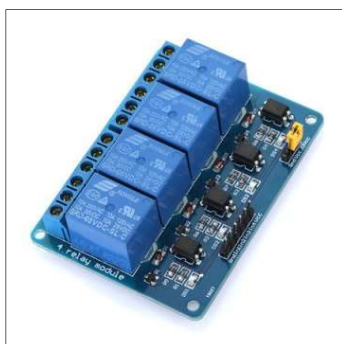


Figura 2 – Módulo relé 4cn

Fonte: (www.filipeflop.com)

Para contornar a deficiência de pinos analógicos do NodeMCU, foi utilizado no protótipo um conversor Analógico – Digital, ADS1115 (figura 3) da AdaFruit. Oferece através do protocolo I2C uma interface de 16bits para leitura de sensores analógicos, com a interface de 16 bits é possível realizar leituras analógicas de mais 4 sensores no NodeMCU.

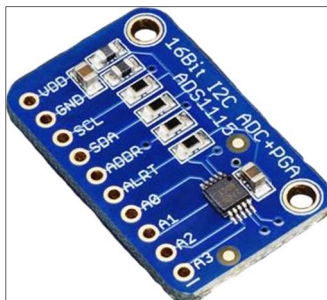


Figura 3 –ADS1115

Fonte: (www.adafruit.com)

Para realizar o controle da irrigação podemos utilizar vários mecanismos, mas o que apresenta o melhor custo x benefício sem dúvidas é utilizar para tal tarefa uma válvula solenoide (figura 4). O solenoide tem funcionamento parecido com o de um relé, onde ao se colocar uma carga de 12V nos terminais da bobina, ela libera o êmbolo que impede a passagem de líquidos. E quando se tira a carga, a bobina fecha o êmbolo e volta a bloquear os líquidos. Para evitar o uso de reservatórios foi utilizado o solenóide ligado diretamente a uma torneira de água.



Figura 4 – Solenóide 12V NF

Fonte: (www.filipeflop.com)

Conseguir a leitura de vários tipos de dados relacionados ao clima dentro do protótipo é uma tarefa simples dado ao grande número de sensores de toda sorte. Para o projeto foram utilizados o DHT11, um higrômetro (figura 5), e um LDR. O DHT11 é utilizado para medir tanto temperatura quando umidade relativa do ar, possui

uma faixa de medição ampla e uma precisão razoável. O higrômetro pode ser adquirido em lojas de jardinagem ou de eletrônica, e serve para medir a resistência elétrica no solo, o que indica a umidade do solo. E o LDR serve para medir a iluminação do projeto.

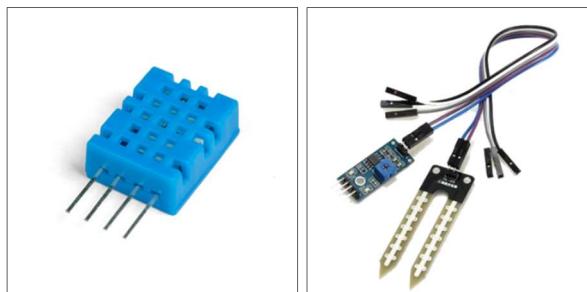


Figura 5 – DHT11 e Higrômetro

Fonte: (www.filipeflop.com)

E unindo o projeto ao aplicativo Blynk (figura 6), foi possível realizar de forma simples e intuitiva uma Dashboard para controlar toda a aplicação através de um smartphone.



Figura 6 – Aplicativo Blynk

Fonte: (www.blynk.cc)

Blynk é um aplicativo para iOS e Android que oferece uma gama variada de widgets ao usuário para que ele monte, como bem entender, uma dashboard para controlar seus projetos IoT. O aplicativo possui uma Cloud pública onde o usuário pode armazenar dados do projeto e os mesmos ficam disponíveis para consulta através do aplicativo por até três meses. É possível também que o usuário monte um servidor local para armazenar seus dados em uma Cloud própria e é fácil encontrar no Git Hub exemplos de como o fazer, mas para fins de facilitar a montagem do protótipo foi utilizado a Cloud pública do aplicativo que além de extremamente robusta, apresenta um Uptime de 99,9% desde seu lançamento em 2015. (Blynk, 2017)

O NodeMCU pode ser programado tanto em LUA, ou em C++. Neste projeto a linguagem escolhida foi C++ por sua facilidade de abstração e comodidade de

programar o módulo utilizando a IDE do Arduino usando, também, os mesmo comandos que são utilizados com as placas Arduino.

Para se utilizar o aplicativo Blynk é necessário antes instalar as bibliotecas disponibilizadas no Git Hub. O processo de instalação é bem simples e não será tratado aqui pelo fato de já ser muito bem explicado na documentação do aplicativo.

Uma vez instaladas as bibliotecas na IDE e o aplicativo no smartphone é só criar uma conta e transferir para o hardware um código com o Auth Token que é fornecido pelo aplicativo quando se cria um projeto novo. O aplicativo dá a possibilidade de controlar, no hardware, pinos analógicos e digitais (figura 7) sem nenhum código extra no hardware. Basta apenas que o usuário adicione à dashboard um Button Widget, por exemplo, e o atribua a algum pino analógico ou digital.

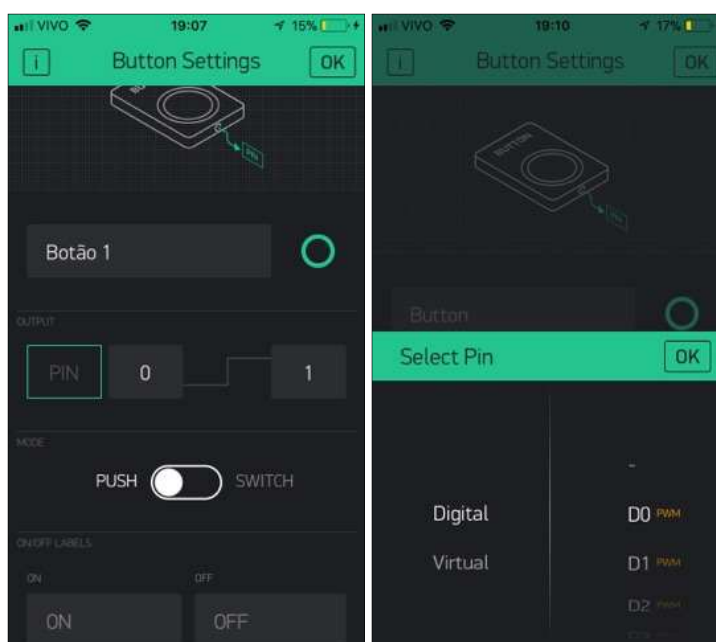


Figura 7 – Configuração de Pinos dentro do aplicativo

Fonte: (Aplicativo Blynk)

Além do controle de pinos digitais e analógicos, também é possível controlar os VirtualPins que a aplicação disponibiliza.

O conceito dos VirtualPins é bem interessante e acrescenta ao aplicativo uma imensa versatilidade. Com pinos digitais é possível apenas enviar ao hardware sinais lógicos de HIGH e LOW. Com os pinos analógicos consegue-se enviar valores de 4bits, entre 0 e 1023. Já com os VirtualPins é possível enviar desde Integers à Strings desde que no lado do hardware seja implementada uma pequena função para processar a informação que será recebida.

```

BLYNK_WRITE(V0){
  int pinValue = param.asInt();
  if(pinValue == 1){
    minhafuncao();
  }
  /*
   * também é possível receber dados com:
   * param.asFloat();
   * param.asDouble();
   * param.asStr();
   *
   * e para widgets que enviam mais de uma informação ao mesmo tempo
   * é possível ler os dados assim:
   *
   * int x = param[1].asInt();
   * int y = param[2].asInt();
   */
}

```

Figura 8 – Receber informações dentro no HW

Fonte: (Autor)

É possível utilizar até 128 VirtualPins dependendo do hardware utilizado, e como mencionado acima, eles podem ser usados para enviar do aplicativo qualquer tipo de informação ao hardware e vice-versa. Pode-se, por exemplo, utilizá-los para enviar as leituras de um algum sensor para um gráfico no aplicativo. Estes dados estarão disponíveis dentro do aplicativo por até três meses e podem ser exportados para um arquivo externo em CSV. Para enviar dados do hardware (figura 9) ao aplicativo é necessário somente uma linha de comando.

```
Blynk.virtualWrite(V1, meuSensor);
```

Figura 9 – Enviar informações ao aplicativo

Fonte: (Autor)

Dentro do aplicativo, o widget que mais contribuiu para as funcionalidades do projeto foi o Time Input (figura 11). Com ele é possível criar rotinas para o hardware, especificar horários de ligar e desligar para os relés e até mesmo os dias da semana que cada relé vai funcionar. O RTC que o aplicativo traz como um widget torna a tarefa de criar rotinas um pouco mais fácil, pois ele permite que o hardware tenha acesso ao horário do servidor e se ajuste ao fuso em que estivermos.


```

BLYNK_WRITE(V2){ //TimeInput IRRIGAÇÃO
  long segundosAgora = 0;
  long segundosStart = 0;
  long segundosStop = 0;

  if(manual == 0){ //verifica que o modo manual está ligado
    TimeInputParam t(param);

    int ajusteDia = -1; ////////////////////////////////////////////////////
    if(weekday() == 1) ajusteDia = 6; //aqui é necessário um pequeno ajuste de dias
    if(t.isWeekdaySelected(weekday() + ajusteDia)){ //da semana, pois biblioteca do rtc começa
      terminal.println("Irrigação Ativara Hoje."); //no domingo e a do timer na segunda-feira
      terminal.flush(); ////////////////////////////////////////////

      if(t.hasStartTime()){ // se tiver tempo de inicio definido
        segundosAgora = (hour() * 3600) + (minute() * 60 + second()); // converte o horario atual em segundos
        segundosStart = (t.getStartHour() * 3600) + (t.getStartMinute() * 60); //converte horario de inicio em segundos
        if(segundosAgora >= segundosStart && segundosAgora <= segundosStop && digitalRead(releAqua) == 0){
          terminal.println(String("Começa: ") + t.getStartHour() + ":" + t.getStartMinute());
          terminal.println("Irrigacao Iniciando"); terminal.flush();
          digitalWrite(releAqua, HIGH); // liga o rele no hardware
          flagTIW1 = true; //sinaliza o hardware que o aplicativo esta enviando um comando
          Blynk.virtualWrite(V11, 1); // muda o estado do botão no app
        }
      }
    }
  }
}

BLYNK_WRITE(V1){ //TimeInput IRRIGAÇÃO
  long segundosAgora = 0;
  long segundosStart = 0;
  long segundosStop = 0;

  if(manual == 0){ //verifica que o modo manual está ligado
    TimeInputParam t(param);

    int ajusteDia = -1; ////////////////////////////////////////////////////
    if(weekday() == 1) ajusteDia = 6; //aqui é necessário um pequeno ajuste de dias
    if(t.isWeekdaySelected(weekday() + ajusteDia)){ //da semana, pois biblioteca do rtc começa
      terminal.println("Irrigação Ativara Hoje."); //no domingo e a do timer na segunda-feira
      terminal.flush(); ////////////////////////////////////////////

      if(t.hasStartTime()){ // se tiver tempo de inicio definido
        segundosAgora = (hour() * 3600) + (minute() * 60 + second()); // converte o horario atual em segundos
        segundosStart = (t.getStartHour() * 3600) + (t.getStartMinute() * 60); //converte horario de inicio em segundos
        if(segundosAgora >= segundosStart && segundosAgora <= segundosStop && digitalRead(releAqua) == 0){
          terminal.println(String("Começa: ") + t.getStartHour() + ":" + t.getStartMinute());
          terminal.println("Irrigacao Iniciando"); terminal.flush();
          digitalWrite(releAqua, HIGH); // liga o rele no hardware
          flagTIW1 = true; //sinaliza o hardware que o aplicativo esta enviando um comando
          Blynk.virtualWrite(V11, 1); // muda o estado do botão no app
        }
      }
    }
  }
}

```

Figura 10 – Função que cuida de manusear o Time Input no HW
Fonte: (Autor)

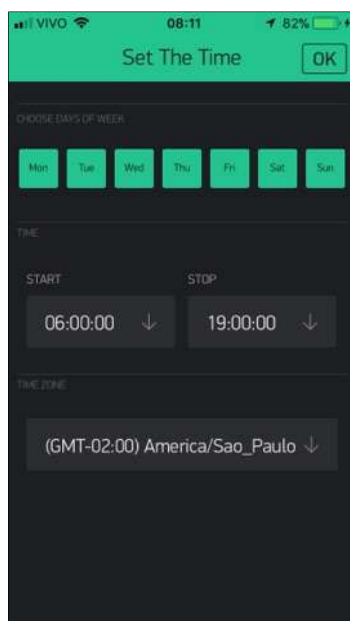


Figura 11 – Time Input
Fonte: (Aplicativo Blynk)

E utilizando a opção de Sharing do aplicativo é possível deixar outras pessoas terem acesso ao hardware através da dashboard criada, elas podem controlar tudo que o autor permitir, mas não podem editar a dashboard e as modificações feitas pelo autor são automaticamente transmitidas para os outros usuários que compartilham o projeto. Esta função do aplicativo só está disponível, até a data deste artigo, para módulos wifi baseados no chip ESP-8266, este fator foi determinante na escolha da plataforma de desenvolvimento.

Para melhorar o entendimento sobre o projeto e compreensão do funcionamento, será apresentado abaixo, de maneira sucinta, o fluxograma do protótipo (figura 12).

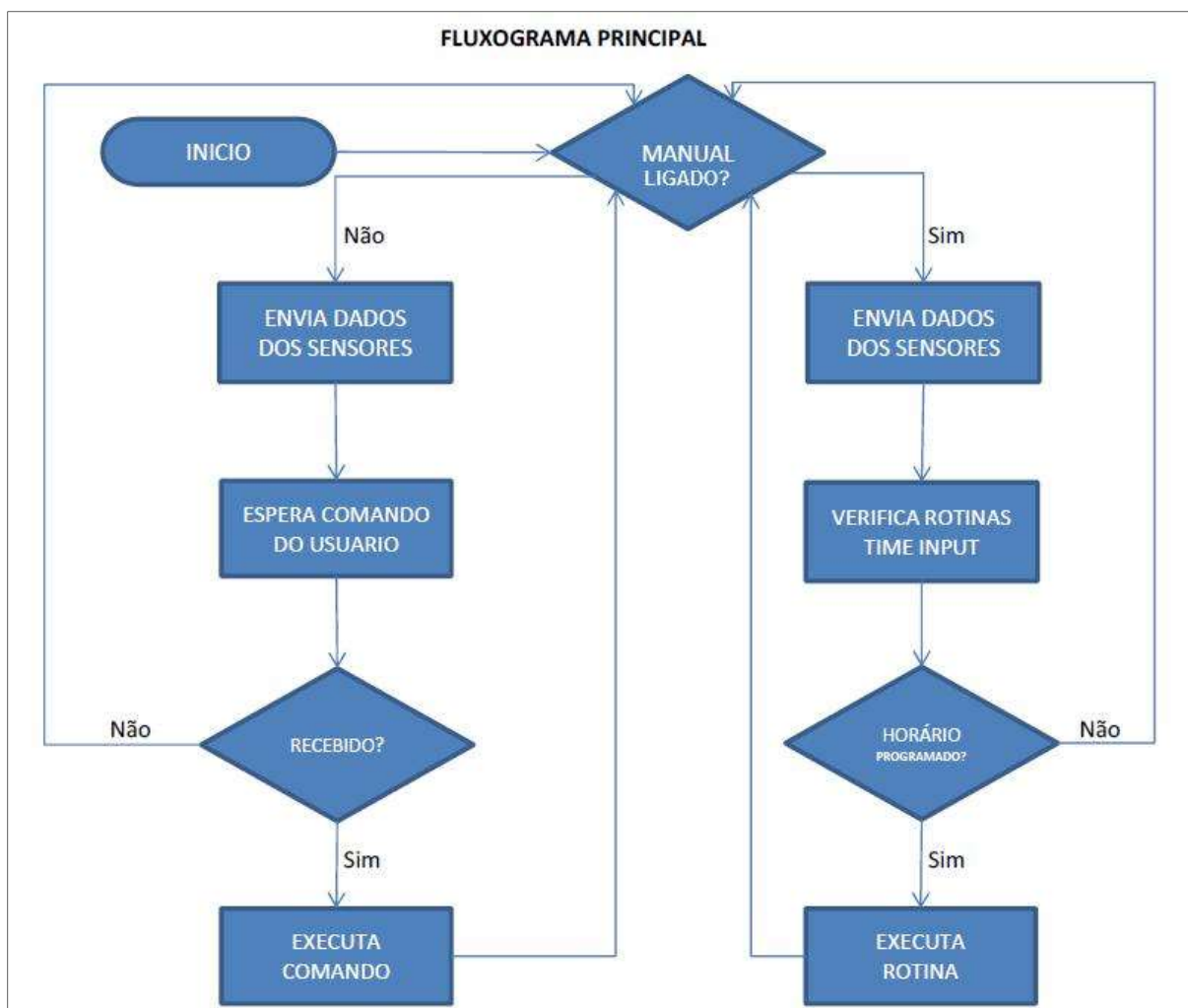


Figura 12 – Fluxo principal do protótipo

Fonte: (Autor)

O funcionamento disposto acima acontece da seguinte maneira: o dispositivo testa constantemente se o modo Manual está ligado ou desligado. Se estiver ligado, o algoritmo funciona basicamente como mostrado no fluxograma acima, mas se estiver desligado, ele envia as informações dos sensores para o aplicativo e faz a checagem em separado de cada Time Input: Irrigação, Iluminação, Aquecimento e Ventilação.

Cada Time Input dá ao usuário a opção de escolher em quais dias da semana ele estará ativo e os horários de ligar e desligar. Deste modo é possível, como descrito anteriormente, criar rotinas a serem seguidas pelo hardware.

DESENVOLVIMENTO

Quando o assunto é IoT, plantações e lavouras não são exatamente o primeiro cenário que vem à cabeça, mas mesmo assim o mercado de Agricultura Inteligente espera um crescimento de US\$5,18 bilhões em 2016 para US\$11,23 bilhões em 2022, o que representa um crescimento médio anual de 13,27% entre 2017 e 2022 (GUERRA, 2017).

Utilizar técnicas e tecnologias de IoT na agricultura não é algo exatamente novo, plataformas e sensores já existem no mercado há um bom tempo e já tem visto uso em diversas aplicações nessa área. Seja para coleta de dados para análise posterior, controle de atuadores, etc. e com os desenvolvimentos recentes, tais componentes se tornaram mais acessíveis em relação ao seu custo e disponibilidade. Um projeto, como o deste documento, por exemplo, pode ser tirado do papel com custo muito menor se comparado a projetos similares.

Tabela 1 - Custos (Eletrônicos) - 2017

Item	Filipe Flop	Robocore	Proj. Arduino
NodeMCU ESP-12F	49,99	49,99	41,00
Módulo Relé	24,90	49,00	38,00
Valvula Solenóide NF 3/4"	34,90	49,00	53,00
DHT11	12,90	15,00	x
ADS1115	36,90	x	x
Higrômetro	10,90	x	x
Placa Circuito Impresso 5x7	5,90	x	x

Fonte: (Autor)

E apesar da gratuidade de se utilizar o aplicativo Blynk, seus widgets possuem um custo baseado em uma moeda própria do aplicativo, embora seu custo seja pequeno, eles agregam também algum custo ao projeto. Todo usuário cadastrado recebe gratuitamente duas mil unidades para customizar suas dashboards como bem quiser, mas para se conseguir dashboards maiores e com mais componentes, alguns custos devem ser observados. A seguir, se encontra uma lista de custos relativos baseados nos widgets que foram utilizados no dashboard deste projeto.

Tabela 2 - Custos (Widgets) - 2017

Widget	Qtidade	Custo Virtual	Custo Real (U\$)
Labeled Display	4	400	0,4
Time Input	4	200	0,2
Vertical Slider	2	200	0,2
Push Notification	1	400	0,4
RTC	1	100	0,1
Button	5	200	0,2
History Chart	2	900	0,9
Shared Access	1	1000	0,99
Totais		7100	7,03

Fonte: (Autor)

Dentro do aplicativo, a organização da dashboard fica totalmente a critério de quem a monta, podendo redimensionar os widgets, organizar em diferentes abas, cores, rótulos e grupos. Existe também a opção de adicionar ao mesmo projeto mais de um dispositivo, ou seja, uma mesma dashboard controlando e recebendo dados de mais de um hardware, ficando a cargo do criador adicionar em cada widget o authToken referente a cada um.

Para ter acesso às leituras mais antigas dos sensores, são utilizados history graphs (figura 13) que nos mostram no aplicativo todas as leituras num período de até três meses e dá a opção de exportar os dados salvos em CSV que pode ser utilizado para análises posteriores e ajustes no algoritmo.

**Figura 13 – History Graph Widget**

Fonte: (Aplicativo Blynk)

O baixo custo do projeto dá a pequenos produtores, ou produtores de baixa renda, um viés para aliar à suas lavouras novas tecnologias capazes de aumentar o rendimento na produção, ao mesmo tempo em que deixa livre quem ficaria responsável pelas funções automatizadas neste projeto, tudo na palma da mão através do smartphone.

Integrar tais produtores ao smart farming pode não representar isoladamente o aumento necessário na produção mundial de alimentos, mas, com certeza, eles não devem ser deixados de lado às novas tecnologias. Dados levantados pelo Ministério do Desenvolvimento Agrário mostram que a agricultura familiar no Brasil é responsável cerca de 70% dos alimentos nas mesas dos brasileiros, desempenhando um papel decisivo na cadeia produtiva brasileira (Portal Brasil, 2015)

RESULTADOS E DISCUSSÃO

Com todo o sistema montado, hardware e dashboard, foi observado crescimento acelerado das hortaliças escolhidas para o experimento. Em especial a Salsa e o Orégano que já possibilitavam podas para uso com apenas três semanas.



Figura 14 – Estufa Montada

Fonte: (Autor)

Foi desenvolvida uma solução IoT de baixo custo para smart farming, automatizando funções básicas necessárias para plantações, promovendo aumento

no rendimento, aceleração no ciclo de produção e otimização para o tempo de trabalhadores que, de outro jeito, ficariam ocupados com os cuidados que foram tratados e automatizados neste projeto.

A todo o momento o usuário tem na tela de seu smartphone os dados atuais lidos pelos sensores e a possibilidade de controlar os sistemas do protótipo, a qualquer hora e em qualquer lugar do mundo onde ele tenha acesso a internet.



Figura 15 – Dashboard final
Fonte: (Aplicativo Blynk)

CONSIDERAÇÕES FINAIS

Durante toda a duração do projeto foram levantados dados sobre como construir um sistema IoT para smart farming, necessidades básicas de verduras e hortaliças e hardware necessários para tal tarefa. Foi construído um sistema viável para uso em pequenas lavouras propiciando aumento notável de produção em troca de pouco investimento.

O uso do NodeMCU foi uma boa escolha para o projeto pelo fato de seu tamanho e custo reduzidos, capacidade de processamento e facilidade de integração com a plataforma escolhida para desenvolver a dashboard do projeto, o Blynk.

Além de todas as funcionalidades do aplicativo, ele também disponibiliza a possibilidade de criação de um aplicativo StandAlone que pode ser encomendado pelo desenvolvedor e publicado na AppStore ou PlayStore. Toda a parte de infraestrutura do projeto e customização é cuidada pelos próprios colaboradores da Blynk.

TRABALHOS FUTUROS.

Para uma próxima versão deste projeto, tem-se a intenção de implementar a utilização da tecnologia OTA. Tecnologia utilizada por todas as maiores empresas produtoras de soluções em IoT para atualizar o firmware sem a necessidade de nenhum cabo conectado ao hardware, todo o processo é feito pela internet desde que ambos, hardware e computador, estejam conectados à mesma rede wi-fi.

BIBLIOGRAFIA

- BI Intelligence. (20 de Dezembro de 2016). ***Why IoT, big data & smart farming are the future of agriculture***. Acesso em 4 de Outubro de 2017, disponível em Business Insider: <http://www.businessinsider.com/internet-of-things-smart-agriculture-2016-10>
- Blynk. (2017). ***Blynk Docs***. Acesso em 30 de Setembro de 2017, disponível em Blynk Getting Started: <http://docs.blynk.cc/>
- Food and Agriculture Organization. (12 de Outubro de 2009). ***Global agriculture towards 2050***. Acesso em 5 de Outubro de 2017, disponível em Fao: http://www.fao.org/fileadmin/templates/wsfs/docs/Issues_papers/HLEF2050_Global_Agriculture.pdf

GUERRA, M. (14 de Agosto de 2017). **3 Ways the IoT Revolutionizes Farming**. Acesso em 12 de Outubro de 2017, disponível em EletronicDesign: <http://www.electronicdesign.com/analog/3-ways-iot-revolutionizes-farming>

MINATEL, P. (23 de Abril de 2016). **IoT de forma fácil com o Blynk**. Acesso em 28 de Setembro de 2017, disponível em Pedro Minatel: <http://pedrominatel.com.br/pt/esp8266/iot-de-forma-facil-com-o-blynk/>

Portal Brasil. (07 de Julho de 2015). **Agricultura familiar produz 70% dos alimentos consumidos pelo brasileiro**. Acesso em 08 de Outubro de 2017, disponível em Portal Brasil: <http://www.brasil.gov.br/economia-e-emprego/2015/07/agricultura-familiar-produz-70-dos-alimentos-consumidos-por-brasileiro>