



ALISSON BOEING

Projeto integrador 3 - Relatório de atividade
Controle de sementes

São José - SC
Outubro/2020

1. Motivação

Os cuidados no armazenamento das sementes em uma fazenda ou sementeira é muito importante para assegurar a qualidade e a produtividade da lavoura. Os fatores que mais influenciam neste cuidado são a umidade, temperatura e ventilação do local.

O Brasil possui regiões com variações muito grandes de temperatura entre inverno e verão. Essas alterações interferem no processo de germinação das plantas, faz com que seja mais acelerado ou mais lento dependendo das condições.

O principal dos fatores é a temperatura, em geral, as sementes precisam ser armazenadas a uma temperatura de 25°C. Durante o manejo, por sua vez, podem demandar uma redução para 20°C, como é o caso do milho e da soja.

A umidade excessiva ou a falta dela podem ser muito prejudiciais para as sementes. Estudos indicam que o aumento de 1% na umidade, por exemplo, pode representar uma perda de produtividade de até 50%. A circulação de ar também se mostra muito importante para manter as sementes, dependendo dos volumes armazenados.

2. Proposta

Tendo em vista a problemática citada acima, a proposta é fazer um sistema de monitoramento da temperatura e da umidade das sementes. Com armazenamento dos valores em banco de dados, uma dashboard para visualização online e uma redundância na coleta dos dados quando o dispositivo estiver offline, por um aplicativo em um smartphone via bluetooth low energy. Alguns benefícios que o sistema levaria às sementeiras:

- Monitoramento em tempo real.
- Possibilidade de automatizar uma ação para corrigir a temperatura/umidade, ligando ventiladores por exemplo.
- Diminuição da mão de obra, visto que ações poderiam ser tomadas remotamente.
- Diminuição indireta do controle de pragas como insetos e fungos, sendo que as sementes estarão secas e mantidas em condições controladas.

A divisão de tarefas ficou da seguinte maneira:

- Infraestrutura (nuvem), código ESP32 e Dashboard: Alisson.
- Aplicativo: Rafael.
- Conexão LoraWan: Fabiano.
- Serviço web e banco de dados: Suyan.
- Modelar, fazer o layout e orçar o sistema embarcado para comportar a solução: Kleiton.

3. Solução

A proposta do sistema abaixo foi o modelado para a solução:

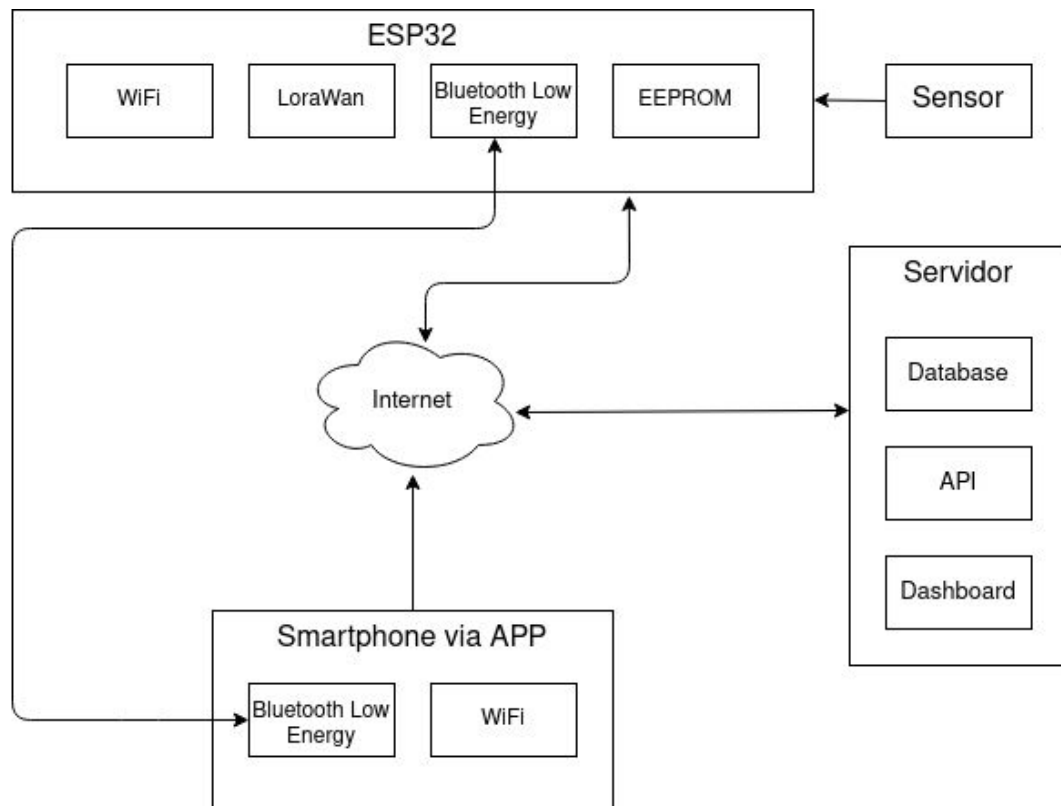


Imagem 1 - Diagrama do sistema.

O sistema conta com os seguintes componentes:

- Esp 32, conjunto de microcontroladores programáveis, na linguagem C++, contando com módulos LoraWan, WiFi, BLE e também vários pinos GPIO's. Um sensor foi conectado à ESP para monitorar a temperatura (o sensor não monitorou a umidade).
- Um aplicativo foi desenvolvido utilizando a ferramenta MIT AppInventor, para coletar os dados da ESP via BLE e enviar pro servidor via internet.
- Um servidor VPS (virtual private server) na nuvem foi contratado para hospedar a API, banco de dados e dashboard do sistema.

4. Desenvolvimento

A imagem abaixo indica o fluxograma do programa desenvolvido para a ESP32:

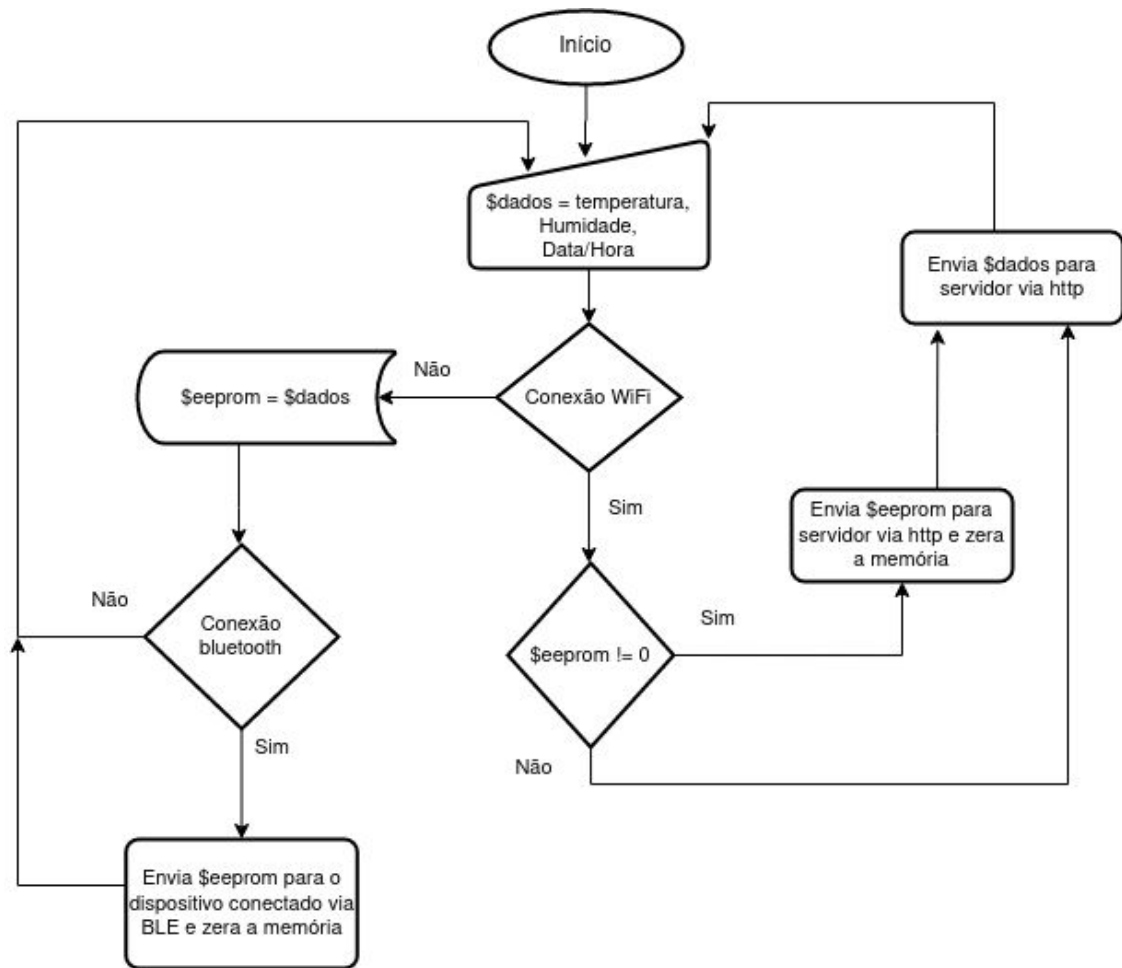


Imagem 2 - Fluxograma do funcionamento da ESP.

Primeiramente a ESP lê os dados do sensor de 10 em 10 segundos (em produção 3 em 3 horas) e verifica se há conexão com a internet, se não houver ela armazenará os dados em uma memória EEPROM, e então verificará se há conexão bluetooth, caso o resultado seja positivo, ela enviará todos os dados para o dispositivo conectado e escreverá zeros na memória, caso negativo voltará para ler mais um dado. E então verificará se há conexão com a internet novamente, se desta vez for positivo, verificará também a EEPROM, se houver dados, enviará os dados da EEPROM primeiro, escreverá zeros na memória e então enviará o dado lido no momento, caso não haja dados armazenados, enviará direto o dado atual.

O servidor em nuvem contratado, possui 1GB de memória e 85GB de armazenamento em disco. Foi instalado e configurado um servidor web Nginx para responder pelo domínio <http://projetodoalisson.serverdo.in>, um gerenciador de banco de dados Mysql e um PHP 7.0 para rodar a API desenvolvida.

Para enviar os dados quando conectado no WiFi é utilizado o serviço http, especificamente o método POST, a API, desenvolvida em PHP e Javascript, monitora as requisições POST's recebidas pelo servidor na URL <http://projetodoalisson.serverdo.in/post-data.php>, quando recebe uma string neste formato: `apikey+temp+umidade+data`, ela se conecta ao banco e insere os valores. O banco de dados

por sua vez, possui apenas uma tabela que armazena os 3 valores e um ID, como exibido na imagem 3. A dashboard está disponível em <http://projetoalisson.serverdo.in/esp-chart.php>, e utiliza uma biblioteca JavaScript chamada Highchart para exibir os dados em forma de gráfico, como exibe a imagem 4.

A parte do BLE, acabei fazendo com um aplicativo teste, que apenas recebia e lia os valores porém, não armazenava encontrei uma referência que fazia este processo e apenas adaptei para o nosso caso de uso. Posteriormente o Rafael ajustou o aplicativo de referência e validou a comunicação.

id	value1	value2	value3
8629	24.87	24.87	2020-10-20T03:02:36Z
8630	26.50	26.50	2020-10-20T03:01:07Z
8631	24.44	24.44	2020-10-20T03:03:27Z
8632	24.50	24.50	2020-10-20T03:03:15Z
8633	24.75	24.75	2020-10-20T03:02:36Z
8634	25.69	25.69	2020-10-20T03:02:11Z
8635	25.19	25.19	2020-10-20T03:02:24Z
8636	26.31	26.31	2020-10-20T03:01:20Z
8637	24.63	24.63	2020-10-20T03:03:02Z
8638	26.06	26.06	2020-10-20T03:01:45Z
8639	25.88	25.88	2020-10-20T03:01:58Z

Imagem 3 - Select no banco de dados.

Projeto integrador III - Controle de sementes

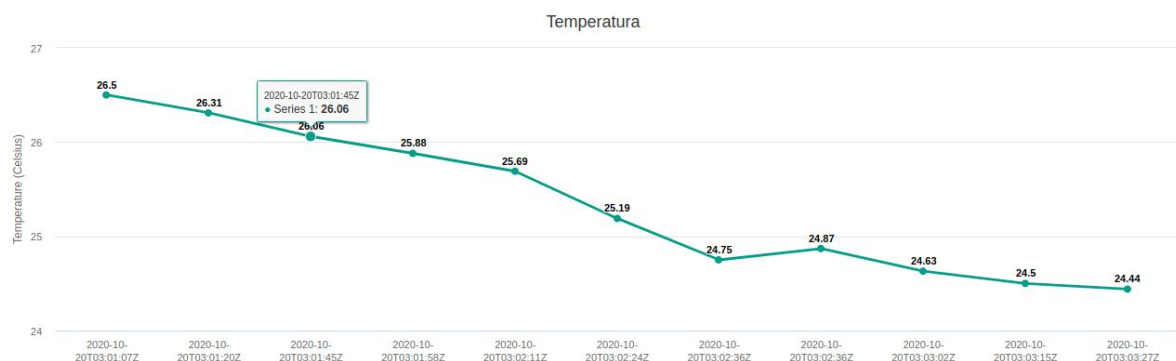


Imagem 4 - Imagem dashboard.

Os códigos utilizados na API, dashboard e ESP32 e estão disponibilizados no GitHub: <https://github.com/AlissonBoeing/ProjetoIntegrador-III>

5. Conclusão

As tarefas que haviam sido destinadas a mim, foram a infraestrutura que seria armazenado a aplicação, banco de dados e serviço web. Outro membro do grupo acabou fazendo a parte do serviço web com netcore em C# e o banco de dados. Porém eu não consegui fazer comunicar com a ESP, então refiz tudo do zero de maneira mais simples utilizando um servidor Nginx, PHP e javascript, encontrei algumas coisas prontas nas quais deixei a referência dentro dos códigos.

Algumas dificuldades foram utilizar o BLE, devido aos aplicativos de referência terem sido um pouco complicado de entender, demorei também para identificar o tamanho do vetor de char que a biblioteca bluetooth envia, acabei quebrando a string de valores em duas para enviar de forma segura.

Juntamente com o Rafael Teles conseguimos fazer funcionar o aplicativo, acredito que ele vai explicar melhor no relatório dele esta parte.

Outro ponto foi o manuseio da EEPROM, também enfrentei um pouco de dificuldade. O relógio também foi um desafio, visto que deve-se manter contando as horas quando o dispositivo estiver sem internet (sem acesso ao NTPclient), então resolvi com um contador interno onde ao iniciar o programa é preciso inserir o timestamp atual.

Não foi possível implementar a comunicação LoraWan, que foi substituída pelo WiFi.

Por fim, acredito que a minha parte do projeto atendeu a solução proposta e poderá ser utilizada no futuro caso haja o interesse de implementar o LoraWan e ou orçar o sistema embarcado para comportar esta solução.

6. Referências:

<https://www.packid.com.br/como-o-armazenamento-de-sementes-interfere-na-produtividade-no-campo/>

<http://www.cdrrs.sp.gov.br/portal/imprensa/noticia/termometria-digital-cati-adota-tecnologia-de-monitoramento-digital-na-secagem-e-armazenagem-de-sementes-na-fazenda-atiliba-leonel>

<https://www.instructables.com/ESP32-BLE-Android-App-Arduino-IDE-AWESOME/>

<https://www.filipeflop.com/blog/bluetooth-low-energy-com-esp32-e-dht11/>

<https://www.hackster.io/botletics/esp32-ble-android-arduino-ide-awesome-81c67d>