

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

ESTAÇÃO METEOROLÓGICA COMPACTA COM DATA ANALYTICS – ENGENHARIA ELÉTRICA USF

André Luis de Souza Melo¹
Fabio Henrique Simões¹
Henrique Savoy Pellizzer¹
Débora Meyhofer Ferreira²
andre.melo@mail.usf.edu.br

¹Aluno do Curso de Engenharia Elétrica, Universidade São Francisco; Campus Itatiba.

²Professor Orientador Débora Meyhofer Ferreira, Curso de Engenharia Elétrica,
Universidade São Francisco; Campus Itatiba.

Resumo

O trabalho apresentado refere-se a uma estação meteorológica automática (EMA), a qual tem seu *Hardware* constituído por um *datalogger* e sensores eletromecânicos/eletrônicos para a medição das variáveis de temperatura, umidade relativa do ar, pressão atmosférica, índice pluviométrico e velocidade do vento. Por se tratar de uma EMA não há necessidade de observações locais para registro dos dados, pois a coleta será realizada através dos sensores e automaticamente registrada pelo *datalogger* que posteriormente irá gravá-la na plataforma *IoT ThingSpeak*. Os dados serão coletados diariamente da plataforma *IoT* via aplicação automática desenvolvida em Linguagem *Python* para *Data Analytics*, com a função de tratar os dados, realizar a previsão de chuva e salvar o resultado em um Banco de Dados na Nuvem da *Microsoft Azure*. Esses dados, uma vez no Banco de Dados, são utilizados para análises e relatórios por plataformas analíticas como o *Power BI*, permitindo melhor visualização das leituras meteorológicas.

Palavras-chave: estação meteorológica, EMA, plataforma *cloud*, *data analytics*.

1. Introdução

A aquisição e análise de dados meteorológicos foi por muito tempo realizada de forma manual, mas com o avanço da tecnologia tornou-se possível realizar a aquisição desses dados de forma automática. No entanto, devido ao alto custo das estações meteorológicas e a dificuldade na análise dos dados coletados, o número de estações operantes e que contribuem para o sistema nacional de meteorologia, hoje regulamentado pelo INMET (Instituto Nacional de Meteorologia), é relativamente baixo para um país de extensões continentais. Desta forma, muitas previsões e análises são realizadas de forma amostral.

Uma aplicação sucinta e útil de uma EMA é o que aconteceu em Março de 2016, na cidade de Itatiba - SP, com diversos problemas de enchentes devido ao alto volume de precipitação naquele período. Para analisar os dados meteorológicos que causaram tal catástrofe, foi utilizada uma estação local operada por um professor da Universidade São Francisco, se essa estação não estivesse em Itatiba, não seria possível obter dados fiéis para estudar as condições climáticas que ocasionaram tais desastres. Com esses dados analisados, é possível detectar um padrão quando uma nova catástrofe semelhante poderá ocorrer na região, comparando os dados coletados e analisados na data do ocorrido, com as previsões disponibilizadas pelo INMET.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

A trabalho tem como foco principal o desenvolvimento de um protótipo de uma estação meteorológica automática de baixo custo e de fácil implementação, que realize a análise dos dados coletados através das mais recentes ferramentas de *Data Analytics* e *Data Science*. O desenvolvimento desta pesquisa se dá devido a necessidade de novas ferramentas mais acessíveis financeiramente para análise de dados meteorológicos, tornando possível aumentar a acessibilidade da análise de dados para qualquer pessoa ou instituição, caso seja necessário fornecer os dados para estudo de desastres naturais ou relacionados.

Os maiores desafios no campo da meteorologia se devem ao alto custo de implementação dos sistemas, embora já existam modelos com custo reduzido em relação às EMA profissionais, o valor destas continua sendo elevado e necessitando de um alto investimento, tornando as mesmas não acessíveis para grande parte da população e instituições.

Inicialmente foi realizada uma pesquisa sobre o funcionamento e composição de estações meteorológicas, levantando assim quais as necessidades que devem ser atendidas por uma estação e quais sensores que devem ser utilizados na mesma. Em seguida, fez-se necessário desenvolver um protótipo com os sensores selecionados que se possibilita o envio dos dados coletados para uma plataforma IoT, onde esses são exibidos e posteriormente analisados por ferramentas analíticas. Os dados são coletados com intervalos de 20 segundos este valor foi definido durante o período de testes do protótipo. Fez-se necessário adquirir uma base de dados históricos do INMET para que fosse possível realizar a análise desses e com isso, criar um modelo preditivo para implantação.

A revisão Bibliográfica considera, primeiramente, o desenvolvimento do hardware, onde foi definido a montagem do circuito eletrônico a ser utilizado, considerando os sensores e os equipamentos necessários para que os dados coletados estejam disponíveis em um ambiente *cloud*. Em seguida, foi levado em consideração a arquitetura do *software*, o qual é necessário para que os dados coletados pelos sensores estejam disponibilizados no ambiente *cloud*. Como última etapa da Revisão, está o desenvolvimento das ferramentas de *Data Analytics* para acessar e utilizar os dados disponíveis na nuvem, de forma que sejam capazes de realizar análises desenvolvidas em Linguagem *Python* e entregar o resultado para uma plataforma visual e interativa com o usuário final.

2. Referencial Teórico

A coleta de dados meteorológicos, de acordo com Vianello (2011), sempre apresentou grande importância no cotidiano do ser humano, representando uma das principais ciências, que é a Meteorologia. Segundo o mesmo, essa ciência não depende exclusivamente de crenças, nações, culturas e idiomas específicos para que possa ser compreendida, mas sim, de um esforço em conjunto internacional para poder coletar, transmitir e tratar as variáveis meteorológicas, sendo essas regidas e coordenadas pela Organização Meteorológica Mundial (OMM) ou *World Meteorological Organization* (WMO).

Ter conhecimento do tempo meteorológico pode beneficiar não apenas o dia a dia de um cidadão comum, como também os setores mais importantes da economia, por exemplo, a agricultura e o transporte, conforme pode ser observado no artigo de Costa (2008), o qual cita a vital importância da meteorologia para o planejamento de voos, interferindo diretamente na segurança e manobras de aeronaves.

Outro estudo realizado por Pereira et al. (1997, apud Strassburger et al., 2008) aponta as inferências que podem ser realizadas com base em dados coletados da temperatura do ar, a qual, quando apresenta elevados valores pode diminuir a taxa de absorção de CO₂ para alguns tipos específicos de plantas, sendo essa uma informação de extrema importância para profissionais da área.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

Para desenvolvimento do projeto, o mesmo foi dividido em cinco etapas principais para melhor organização e controle das atividades, conforme observado na Figura 1.

Figura 1 - Diagrama de Blocos do Projeto



(Fonte: Própria)

2.1 Aquisição de Dados

Uma vez que, a importância em se ter a coleta de dados meteorológicos seja primordial para o ser humano, uma estação meteorológica pode se fazer presente para atender essa necessidade. Diversos elementos podem ser analisados através das observações realizadas por uma estação meteorológica, como por exemplo, temperatura, umidade relativa do ar, velocidade do vento, pressão atmosférica, precipitação, dentre muitos outros (WMO, 2008).

Os dados meteorológicos podem ser coletados de diversas formas, sendo as mais comuns por meio de uma estação meteorológica convencional ou automática. A diferença entre elas está na forma de transmissão dos dados, onde a estação automática coleta e transmite esses dados para uma unidade central de processamento, otimizando o trabalho do observador meteorológico, no entanto, a estação meteorológica não possui o mesmo critério para todas as aplicações, isso irá se diferenciar de acordo com o objetivo e local de uso (WMO, 2008).

O trabalho de Strassburger et al. (2011), detalha com clareza os sensores que foram utilizados na estação meteorológica convencional e na automática. Na primeira, sensores mecânicos de base de mercúrio foram utilizados para medição da temperatura, sendo necessário uma observação local da variável para registro dos dados, enquanto na segunda estação, foi utilizado um sensor eletrônico, o qual registrava os valores coletados diretamente em uma central de processamento, no entanto, o autor não ressalta o uso de plataforma cloud e nem de plataformas gráficas para visualização dos dados da EMA, realizando um estudo isolado de cada uma das estações.

A EMA deve ser composta por sensores específicos para a leitura das variáveis meteorológicas, como sensores de temperatura e pressão atmosférica, os quais enviam os dados em tempo real e em intervalos específicos para uma central de processamento, onde estes poderão ser coletados (Fernandes; Araújo; Gomes; 2013). Nesse tipo de estação, podem ser utilizados qualquer sensor que gere um sinal elétrico de saída, podendo esse ser analógico ou digital, independentemente do tipo, é recomendável que esses sensores tenham boas resoluções para amostragem de sinais e baixo erro relativo durante a medição (WMO, 2008).

Uma vez escolhidos os sensores de monitoramento meteorológico, esses devem ser integrados a um sistema de aquisição de dados, o qual geralmente é baseado em um microcontrolador. Nessa etapa é importante observar a compatibilidade da saída elétrica do sensor com a entrada do microcontrolador, caso contrário, poderão ocorrer leituras incorretas dos dados ou falhas de comunicação (Fernandes; Araújo; Gomes; 2013).

A base de tempo para coletar as variáveis através dos sensores deve ser levada em consideração, pois caso essa base de tempo seja muito elevada, parte dos dados poderão ser perdidos, ou seja, mudanças importantes no valor da variável fora do tempo estabelecido na amostragem não serão coletadas, com isso, a leitura não poderá ser confiável para futuras análises (Lathi, 2007). No caso do estudo apresentado por Palmieri et al. (2014), foi adotado um período de amostragem igual a 15 minutos, onde para tal aplicação gerou bons resultados,

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

devido ao local de medição não estar sujeito a alterações bruscas em variáveis meteorológicas, porém poderia gerar diferenças importantes em variáveis como velocidade do vento, podendo não detectar possíveis rajadas de vento no local.

2.2 Coleta de Dados

Com relação ao sistema de aquisição dos dados, Fernandes; Araújo; Gomes (2013) realizaram a comunicação com os sensores utilizando o microcontrolador Arduino Uno R3, porém não somente o Arduino, como outros tipos de microcontroladores podem ser utilizados para coleta dos dados meteorológicos. Prasanna et al. (2019) utilizou por exemplo o microcontrolador NodeMcu para coletar as informações de três sensores diferentes. Prasanna et al. (2019) não realiza a análise de dados e modelagem preditiva em produção, mas sim em ambiente de desenvolvimento, o que não é a realidade do mercado. O sistema embarcado *Raspberry Pi* também é muito explorado e utilizado para coletar medições de sensores meteorológicos, conforme é relatado no artigo de Nallakaruppan; Kumaran (2019). A análise de dados é feita em ambiente de desenvolvimento usando Linguagem R, além da necessidade de um conversor analógico digital para o *Raspberry Pi* utilizado, de forma que o autor tenha conseguido validar sua teoria.

Para esta aplicação foi selecionado o microcontrolador ESP32, pois é possível programá-lo pela plataforma *open source* Arduino IDE, a qual é muito utilizada em meios acadêmicos, e utiliza programação em linguagem C que se classifica como linguagem de alto nível. Além disso, o ESP32 possui módulo de comunicação Wifi integrado, não sendo necessário uma placa de expansão para comunicação com a ferramenta *cloud*.

2.3 Registro dos Dados

Diversas plataformas *cloud* estão disponíveis para integrar dados coletados por um sistema de aquisição. Uma delas é o *ThingSpeak*, selecionada para esse projeto, apresentada por Prasanna et al. (2019), a qual é uma plataforma analítica que recebe dados e permite gerar visualizações gráficas em tempo real, podendo ser facilmente integrada com um sistema embarcado e programada via linguagem de programação, como códigos criados em Matlab da *Mathworks*, além de ser *open source*.

Muitas estações meteorológicas automáticas são construídas já utilizando sistemas *IoT*, que se comunicam com satélites, para coletar informações como imagens térmicas, de forma que funcionem como um protagonista importante para o monitoramento do clima e detecção de mudanças climáticas ao redor do globo (Sun et al, 2015), no entanto, tais estações geram discussões com relação ao custo e dificuldades para implantação (Fernandes; Araújo; Gomes; 2013), sendo de difícil acesso para o cidadão comum.

2.4 Análise e Previsão dos dados

Com a integração de plataformas *IoT* com os sistemas embarcados, torna-se possível aplicar técnicas e análises avançadas utilizando conceito de *Data Science*, dentre esses, *Machine Learning* (Nallakaruppan; Kumaran; 2019). Esses conceitos podem ser empregados por aplicações *web* ou funções em *cloud* para análise de dados.

Nallakaruppan; Kumaran (2019) apresentam em seu projeto um modelo de *machine learning* classificado como Modelo de aprendizagem supervisionada. Na aprendizagem supervisionada, os dados coletados e armazenados na plataforma *cloud* são extraídos por uma ferramenta essencial para *Data Science*, como Linguagem *Python* e Linguagem R

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

(Nallakaruppan; Kumaran; 2019), onde os dados são explorados, tratados, limpos e divididos em duas amostras, uma para treinamento do modelo e outra para teste. O modelo de *machine learning*, que nada mais é que um modelo matemático, irá aprender os padrões e *insights* entre as variáveis preditoras e a variável que será prevista, para que, uma vez treinado, possa gerar a previsão com relação a novos conjuntos de dados (Géron, 2019). Os dados de teste servirão para validação do modelo, de forma que a acurácia das previsões sejam mantidas em um valor aceitável e que os resultados possam ser apresentados para o usuário final (Géron, 2019).

Uma das bases de *Data Science* é a Estatística, a qual se desenvolveu rapidamente e se tornou uma teoria aplicada, sendo altamente relacionada com análise e modelagem de dados, algo muito importante em uma das primeiras etapas de um processo de Ciência de Dados, que é a Análise Exploratória dos Dados (Bruce; Bruce; 2019), onde serão obtidas informações de como os dados estão se comportando, se existe alguma correlação entre variáveis, problemas de escala, valores *missing*, dentre outros fatores que irão influenciar fortemente na entrega do resultado de análise para o usuário. Além disso, o Modelo de *Machine Learning* espera receber dados que estejam estruturados, limpos e que, no caso de uma aprendizagem supervisionada para previsão de valores numéricos, estejam próximos de uma distribuição normal, caso contrário, o padrão dos dados a ser definido pelo modelo não será coerente, podendo gerar previsões incorretas, porém isso varia com a aplicação e o problema de negócio.

Uma estação meteorológica automática em conjunto com um sistema *IoT*, permite aplicar a Ciência de Dados para gerar análises mais detalhadas sobre as variáveis, assim como, previsões de temperatura ou umidade relativa do ar como forma de resultado para o usuário, garantindo maior eficiência e praticidade ao utilizar a estação meteorológica (Prasanna et al. , 2019). A ideia por trás da Ciência dos Dados e da Análise dos Dados é tornar a análise mais rica e conseqüentemente, gerar valor para o usuário (FOREMAN, 2016).

Para esse projeto, a escolha dos modelos de aprendizado de máquina apresenta o *Random Forest*, *kNN*, *Extra Tree Classifier* e *Naive Bayes*, devido a estrutura desses algoritmos ser preparada para trabalhar com problemas de classificação, onde a variável alvo possui duas ou mais categorias (Géron, 2019). O resultado dessa análise é então armazenado no Banco de Dados em um ambiente *cloud*, sendo esse o *Cosmos DB*, o qual permite de forma independente e escalável armazenar múltiplos modelos de dados, como documentos, chave-valor e tabelas (Bharadi, 2018).

2.5 Entrega dos resultados

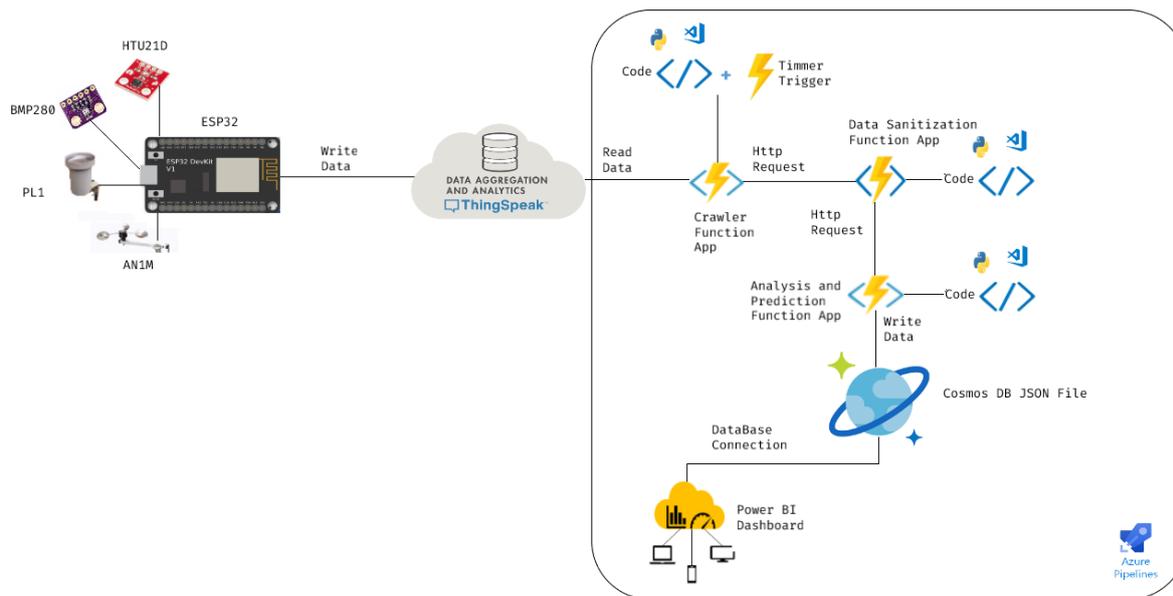
O Banco de Dados em *cloud* serve para armazenar todo o resultado da aplicação analítica de dados, sendo o último passo a visualização desses dados de forma visual e interativa, conforme desenvolvido no trabalho de Fernandes; Araújo; Gomes, 2019, no qual os autores exibiram os dados de uma estação meteorológica em uma página *web* desenvolvida pelos mesmos. Embora essa seja uma alternativa válida, pode-se se apropriar de *frameworks* e plataformas prontas que tragam mais simplicidade e dinamismo para a visualização dos dados, como o *Power BI* da *Microsoft Corporation*. O *Power BI* permite a comunicação com diversas fontes de dados, inclusive plataformas *cloud*, o que faz com que a leitura dos dados ocorra de forma fluída, sendo flexível a possíveis alterações nas formas de visualização (Russo; Ferrari; 2016).

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

3. Metodologia

O projeto consiste em uma arquitetura dividida entre *Hardware* e *Software*, iniciando na leitura e aquisição de dados meteorológicos, posteriormente armazenando os mesmos, e em sequência analisando-os através de ferramentas de Análise de Dados. A Figura 2 a seguir apresenta tal arquitetura de forma visual:

Figura 2 - Arquitetura do Projeto



(Fonte: Própria)

3.1 Estrutura de Montagem

A montagem da estação meteorológica foi realizada em uma estrutura metálica composta por um cano de aço inoxidável ou alumínio como parte central, a qual tem a sua extremidade inferior fixada ao solo através de contato direto, porém a mesma pode ser fixada em uma base retangular ou quadrada de concreto, caso seja necessário. Esta haste central foi utilizada para fixar todos os equipamentos relativos ao *Hardware* do protótipo, ou seja, os sensores que estão fixados em suportes de alumínio, assim como um abrigo para os sensores, caixa de controle e comunicação onde estão alocados o *datalogger*, a fonte de alimentação e os demais itens que não podem sofrer a ação de intempéries (WMO, 2008).

3.2 Aquisição e Registro de Dados

A aquisição de dados e de grandezas meteorológicas foi realizada através de sensores pré-selecionados de forma a apresentar baixo custo de implementação, que forneçam e garantam leituras confiáveis e com alto índice de resolução dos dados, que posteriormente são analisados. A EMA, que significa Estação Meteorológica Automática, é composta por quatro sensores principais, os quais estão descritos na Tabela 1.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

Tabela 1 - Lista de Sensores da Estação

Sensor	Variável	Fabricante	Modelo	Comunicação	Tensão de alimentação	Potência	Resolução	Observações
Higrômetro / Termógrafo	Umidade do ar e Temperatura	MEAS (Measurement Specialties)	HTU21D	Protocolo I2C	3V3	2,7μW	±3% (Umidade Relativa) ±0,3 °C (Temperatura)	Alta Resolução e baixo consumo
Barômetro	Pressão Atmosférica	BOSCH	BMP280	Protocolo I2C	3V3	2,3mW	±100 Pa	Alta Resolução e baixo consumo
Anemômetro	Velocidade do Vento	WRFCOMERCIAL	AN1M	Pulso digital gerado a cada rotação	5V	-	0,9 km/h (mínima) 135 km/h (máxima)	Resistente às condições climáticas
Pluviômetro	Precipitação	WRFCOMERCIAL	PL1	Pulso digital gerado a cada 0,25mm de precipitação	5V	-	0,25mm	Auto esvaziamento e resistente à intempéries

(Fonte: Própria)

Após a coleta dos dados é necessário que estes sejam processados e registrados através de um *data logger*. Para este protótipo foi selecionado o dispositivo ESP32 *Devkit*.

O ESP32 *Devkit* é uma placa de desenvolvimento fabricada pela *Espressif Systems*, composta por um microcontrolador e seus periféricos, como entradas e saídas digitais e analógicas, assim como pinos Rx e Tx. O principal periférico que influenciou a escolha deste dispositivo para o protótipo foi o módulo *Wifi/Bluetooth* integrado, o qual permite a comunicação com a plataforma *IoT* via *Internet*, desta forma é possível realizar o *upload* dos dados registrados de forma automática, através de programação, sem a necessidade de intervenção humana, para que estes fiquem armazenados e posteriormente possam ser analisados.

A programação deste microcontrolador foi escrito via *IDE* do Arduino, sendo essa uma ferramenta *FLOSS* (*Free Library Open Source Software*), ou seja, sem custo, fornecendo um ambiente baseado em Linguagem C para desenvolvimento do *script* de coleta dos dados.

O sensor Pluviômetro está conectado à uma entrada digital do ESP32 *Devkit* que foi programadas para se comportar como pino de interrupção externa. Uma vez que o sinal deste sensor é emitidos no formato de pulsos, através de *reed switch* mecânico, é possível que seja perdido, caso o programa esteja executando algum outro ponto da rotina de programação no momento de ocorrência do pulso, desta forma fez-se necessário a interrupção da programação, de forma que ocorra o registro do pulso recebido e após a programação retorne a execução de forma automática. O sensor Anemômetro assim como o anterior têm seu sinal emitido no formato de pulsos, através de um *reed switch*, para a leitura do valor apresentado para o mesmo, a cada ciclo de leitura, o programa contabiliza o número de pulsos gerados pelo sensor em um intervalo de 5 segundos, através de uma equação que considera as dimensões estruturais do mesmo, transforma este valor contabilizado em variável de velocidade expressa em m/s

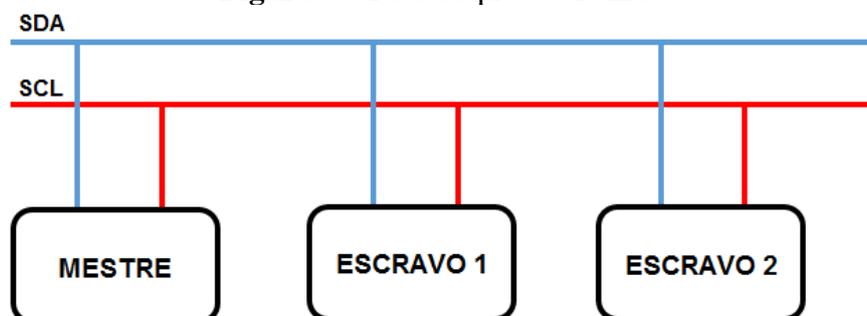
Os sensores Barômetro, Higrógrafo e Termógrafo possuem comunicação do protocolo *I2C*. O protocolo *I2C* possui em sua estrutura de funcionamento a requisição de dados de um dispositivo mestre (microcontrolador) para os dispositivos escravos (sensores) em um barramento de comunicação. Todos os dispositivos possuem seu próprio endereço, onde o mesmo é definido pelo fabricante e pode ser encontrado no datasheet do componente. Usando o protocolo, é possível trocar informações de um dispositivo para o outro através de requisições e envios pelo barramento de comunicação, os quais são: *SDA* e *SCL*. Na prática, o dispositivo mestre faz uma requisição de informações para o endereço *X*, onde todos os dispositivos ligados

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

ao barramento recebe essa requisição, mas apenas o que possui o endereço correto retorna a informação solicitada.

O sensor de pressão BMP 280 possui por definição do fabricante o endereço 0x76, enquanto o HTU21D possui o endereço 0x40.

Figura 3 – Foto do protocolo I2C



(Fonte: <https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinos/>)

Com isso, o ESP32 *Devkit* atua como um *datalogger* da estação meteorológica, uma vez que ele recebe os dados de todos os sensores.

Os dados são recebidos e processados para então ser enviados pelo ESP32 para a plataforma *IoT Thingspeak*, em intervalos de 20 segundos, por se tratar de uma estação meteorológica automática. Esse intervalo de tempo foi escolhido pois não existe a necessidade do envio constante dos dados com período inferiores a este, podendo assim possibilitar o acompanhamento em tempo real dos dados coletados, e ao mesmo tempo não gerar um fluxo excessivo de dados para a plataforma *IoT*.

A comunicação entre o ESP32 e o *Thingspeak* é realizada através da biblioteca “*ThingSpeak*” para a *IDE* do Arduino, a qual, baseada em um cliente *Wifi* permite o envio de dados para a plataforma. Dessa forma, o ESP32 deve estar conectado a uma rede *Wifi* com *Internet* para que os dados sejam enviados periodicamente, caso contrário, não haverá registro de dados.

3.3 Armazenamento de Dados

Os dados enviados pelo ESP32 são exibidos no *ThingSpeak* através de um *channel* ou canal, o qual é composto por *Fields* ou Campos de visualização de dados. Cada um desses *fields* representa uma variável meteorológica, ou seja, um valor captado pelo sensor que é enviado ao ESP32 e salvo na Plataforma *Cloud* a cada 20 segundos. Por se tratar de uma plataforma analítica, cada um desses *Fields* permitem diferentes configurações gráficas como, alteração de escala dos dados, cores de exibição, tipo do gráfico (podendo ser no formato de linhas, barras, colunas, degraus), limites de eixo, dentre outros itens que servem para melhorar a visualização dos dados em tempo real. O que classifica o *ThingSpeak* como sendo uma plataforma *IoT* é que o hardware não precisa estar conectado diretamente em uma porta serial para enviar dados à plataforma, ele envia esses dados via *internet* em tempo real, logo, não existe a necessidade de nenhuma conexão física para o fluxo de dados. Além disso, a visualização do canal pode ser realizada via qualquer dispositivo que esteja conectado à *internet*, seja esse um computador ou um *smartphone*, basta ter o *link* de acesso ao canal criado para a aplicação desejada.

O *ThingSpeak* permite não apenas a escrita de dados nos seus *fields* como também a leitura, que servirá para análises posteriores nesse projeto. Essa leitura pode ser feita através de um endereço *url* fornecido pelo *ThingSpeak* referente ao canal, podendo nessa *url* alterar

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

parâmetros de consulta, como o *field* desejado, a data e horário de início da consulta e a data e horário de término da consulta. Além desse ponto, a plataforma apresenta fácil integração com linguagem *Python*, linguagem *Javascript (node javascript)*, dentre outras, facilitando o trabalho de coleta e armazenamento de dados.

Se for preciso coletar os dados históricos da estação, mas de forma manual, a plataforma *ThingSpeak* também fornece a opção de fazer o *download* do repositório de dados no formato “.csv”, “.json” ou “.xml”, conforme necessidade.

3.4 Análise e Previsão de dados

O processo de análise de dados irá utilizar dados do *ThingSpeak*, cuja saída se dá no formato *JSON*, formato esse de fácil integração e leitura dentro de um sistema computacional, além de apresentar um baixo volume em disco. Com os dados da estação meteorológica disponíveis e armazenados na plataforma *ThingSpeak* torna-se possível iniciar o processo de análise de dados.

A partir desse ponto, todas as demais etapas são baseadas e controladas dentro de um *Azure Pipeline*. *Azure* é a plataforma *Cloud* da empresa *Microsoft Corporation*, a qual permite que uma aplicação ou serviço *web* seja executado sem a necessidade de um servidor, ou seja, permite que seja executado uma aplicação *Serverless*. Para esse projeto, isso significa que não é necessário uma requisição do usuário para poder realizar as análises dos dados meteorológicos do dia anterior, pois isso é feito automaticamente às 00h59min., pela própria aplicação. Embora o portal da *Azure* seja considerado uma ferramenta proprietária, essa aplicação não atinge os requisitos mínimos estipulados pela empresa para dar início à cobrança, logo, custos adicionais não são cobrados para uso do portal.

A infraestrutura da aplicação *Serverless* é dividida em quatro etapas principais, seguindo uma arquitetura de coreografia, onde cada etapa depende uma da outra, tornando o sistema serializado.

Figura 4 - Infraestrutura da aplicação



(Fonte: Própria)

3.4.1. Coleta de Dados

Dentro do *Azure*, existe uma funcionalidade chamada “*Azure functions*” a qual permite executar *scripts* pela aplicação escritos em linguagem *Python*, linguagem *Java*, dentre outras. Para a primeira etapa da infraestrutura de coleta e armazenamento de dados, foi criada uma *Azure Function* denominada “*Crawler*”.

A *Azure Function* é composta por um código e uma condição para execução. Nesse caso, todos os códigos que a *Azure Function* executa são baseados em Linguagem *Python*, adotando *Python 3* e desenvolvidos na plataforma gratuita de desenvolvimento *Visual Studio Code*. A condição de execução desse código é via um *Timer Trigger*, ou seja, em um determinado tempo específico a função é executada.

O código principal de execução da *Azure Function* invocar dois outros *scripts Python*. O primeiro *script* é composto pelos comandos necessários para requisitar a comunicação com o *ThingSpeak*, enquanto o segundo *script* realiza a serialização dos dados para um formato *JSON (Javascript Object Notation)* estruturado, que se baseia no conceito de “chave-valor”.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

Esse método de execução segue as boas práticas de Arquitetura Limpa de *Software*, onde pode-se ter controle absoluto sobre as dependências do código fonte do sistema.

Uma vez que a *Azure Function* possua o código que deverá ser executado é necessário informar quando a execução deverá ocorrer. A *Azure function* fornece diversos tipos de *triggers* para execução, como o *Http request*, mas para esse projeto está sendo utilizado um *Timer Trigger*, ou seja, a *Azure Function* é executada todos os dias às 00h59min. Para configurar essa *feature* é preciso inserir a configuração no formato *CRON*, que nada mais é que um comando que informa quando determinada tarefa deve ser executada, geralmente é dado na sintaxe de “minuto hora dia_do_mês mês dia_da_semana”, por exemplo, para esse projeto a sintaxe será “59 00 * * *”, o símbolo de “*” significa que a *Azure Function* será executada todos os dias no horário 00h59min. Como saída, essa função envia os dados serializados em *JSON* via *HTTP Request Post* para a próxima função do processo, que no caso, é a de tratamento dos dados ou “*Data Sanitization*”.

Dessa forma, a primeira etapa da infraestrutura executa e armazena os dados conforme solicitado. A *Azure Function* se preocupa, portanto, em apenas executar as principais tarefas para a etapa que ela foi criada.

3.4.2. Tratamento dos dados

A segunda etapa do projeto consiste no tratamento dos dados coletados pela *Crawler Function*, para que o processo de previsão do modelo seja realizado pela próxima função.

Em questão de infraestrutura, foram gerados dois *scripts* em *Python* no *Visual Studio Code*. Um *script* para dar estrutura aos dados e outro principal para execução de processo e envio dos dados para a próxima função via protocolo *Http Request Post*.

Para a estruturação dos dados, foi adotada a Linguagem *Python*, pois essa oferece os pacotes do *Python Open Data Science Stack*, pacotes esses que foram construídos prontamente para construção de um *pipeline* completo de *Machine Learning*. Dentre esses pacotes, o principal é o “*Pandas*”, um excelente pacote para converter os arquivos coletados de *JSON* para *dataframe* e com isso poder realizar avançadas manipulações nos dados. Uma vez que os dados tenham uma estrutura, são eliminadas as variáveis que não serão utilizadas para análises posteriores e previsão do modelo, como por exemplo, o *ID* do registro coletado no *ThingSpeak*.

Uma vez que os dados tenham sido estruturados e manipulados, esses são enviados para a próxima etapa que é a do Modelo Preditivo, o qual irá receber os dados no mesmo formato que ele aprendeu para poder retornar a previsão de chuva.

3.4.3. Modelo Preditivo

Machine Learning é um dos principais temas dentro de *Data Science* ou Ciência de Dados. A possibilidade de ensinar um *software* a aprender com a experiência é algo fascinante e que abre um mundo de possibilidades. É cada vez mais comum *softwares* que aprendem e adquirem novos conhecimentos através da experiência. O pacote “*sklearn*” é provavelmente a biblioteca mais poderosa para se trabalhar com *Machine Learning* em *Python*, podendo ser usados em conjunto com o *NumPy*, *Pandas* e *Matplotlib* para construir ferramentas eficientes para *Machine Learning* e Modelagem Estatística, como Regressão e Classificação.

O primeiro passo é trabalhar na criação do modelo preditivo, para isso, foram utilizados os dados disponibilizados pelo INMET (Instituto Nacional de Meteorologia) com registros de 265 estações meteorológicas, desde 1961 até 2019, instaladas nas cinco regiões do Brasil (Norte, Nordeste, Centro Oeste, Sul e Sudeste), totalizando mais de 12 milhões de registros.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

Basicamente, o processo de análise e previsão de dados pode ser resumido pela Figura 5.

Figura 5 - Machine Learning Workflow



(Fonte: Própria)

Um vez escolhida a Linguagem de Programação, o próximo passo é adotar o processo de *Data Wrangling* ou etapa de preparação dos dados. *Data Wrangling* é simplesmente a manipulação de dados, que consiste praticamente no processo de conversão ou mapeamento dos dados do seu estado bruto em um outro formato que seja mais conveniente para consumo por aplicações e ferramentas de análise, sendo um pré requisito no processo de visualização, agregação, modelagem estatística e *Machine Learning*. Se esse trabalho for negligenciado, todo o restante estará comprometido. Os dados vão sendo transformados à medida que as técnicas de manipulação vão sendo aplicadas. Somente quando os dados estiverem realmente consistentes, será possível realizar a exploração dos dados ou a etapa de análise exploratória e pré processamento dos dados. Atividades como, formatar dados do tipo data, eliminar valores *NA* do conjunto de dados, conversão de unidades, são tarefas comumente realizadas na etapa de preparação dos dados.

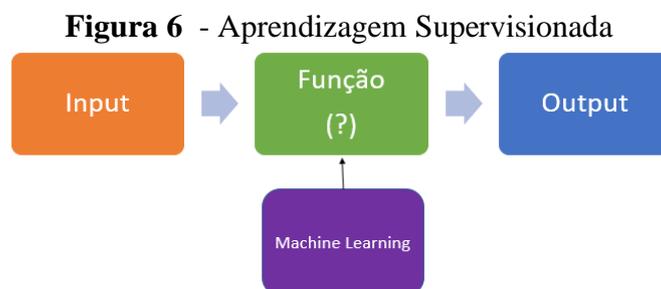
Na Etapa de Exploração dos dados, ou simplesmente Análise Exploratória, foi possível ter uma melhor compreensão dos dados. Uma vez que os dados estejam limpos, pode-se olhar para eles e tentar compreender como estão organizados, buscar *outliers*, buscar padrões, buscar algo que auxilie a melhorar o conjunto de dados para a modelagem preditiva. Os dados, normalmente mostram exatamente as informações, mas para isso é preciso olhar para eles por diferente ângulos, nessa etapa é muito aplicado a Estatística, que é um ramo da ciência responsável por transformar dados em informação, o dado apenas armazenado não tem qualquer valor.

Ao observar um fenômeno sucessivamente é possível notar que, muito raramente, os resultados encontrados serão iguais, isso porque praticamente tudo está sujeito a variação, no entanto, o uso de métodos estatísticos permite que se facilite a compreensão e descrição dessa inconsistência e que ela seja usada de forma a ajudar no processo de tomada de decisão. Ela é utilizada para descrever, resumir e explorar os dados. Ao trabalhar com *Machine Learning*, utiliza-se Estatística para interpretar e avaliar os resultados do modelo.

Com os dados limpos, tratados e filtrados, o próximo passo é a escolha do modelo de *machine learning* para prever a quantidade de chuva do dia anterior. O modelo deve ser baseado na Aprendizagem Supervisionada, que é o termo usado sempre que o algoritmo é “treinado” sobre um conjunto de dados históricos contendo entradas e saídas. Aqui, busca-se dados históricos, dados do passado que serão usados para treinar um algoritmo de forma que esse possa fazer previsões quando receber um novo conjunto de dados. O algoritmo aprendeu o relacionamento entre as características do tempo quando ocorreu chuva e quando não ocorreu. Baseado no treinamento com os dados históricos, o modelo pode tomar decisões precisas quando receber novos dados.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

A aprendizagem supervisionada é basicamente dividida em duas categorias: Regressão e Classificação. Para esse problema, serão usados algoritmos de Classificação, a qual consiste em realizar a previsão de valores categóricos. Logo, o modelo de *machine learning* nada mais é que uma função matemática que representa a relação entre os dados de entrada e os dados de saída, conforme pode ser observado na Figura 6.



(Fonte: Própria)

Para escolha do algoritmo de classificação, foi considerado o algoritmo *Random Forest* ou Floresta Aleatória, o qual é formado por diversos modelos de árvores de decisão em paralelo, esses modelos são todos treinados ao mesmo tempo, ao final é gerada a média das previsões gerando uma acurácia muito maior no resultado do mesmo. Outro modelo escolhido, de forma que a acurácia possa ser comparada foi o *Naive Bayes*, baseado no Teorema Bayesiano, assim como também foram escolhidos os modelos *Extra Tree Classifier* (ETC, mesma família que o *Random Forest*) e *kNN* (*k-nearest neighbors*). Todos esses algoritmos são utilizados para treinar modelos de *machine learning* de aprendizagem supervisionada e voltados para problemas de classificação.

Os dados são uma parte importante durante o processo de aprendizado do Modelo preditivo. Durante esse processo, trabalha-se com dois conjuntos de dados: Dados de Treino e Dados de Teste. Separa-se, de forma aleatória 70% do conjunto de dados original e denomina esse como Dados de Treino, o restante 30% irá compor os Dados de Teste. Essa é uma etapa importante que deve ser realizada antes de treinar o modelo preditivo. Além disso, antes de realizar a divisão dos dados, a distribuição da variável categórica que será prevista não deve estar mal distribuída, ou seja, uma categoria com muito mais ocorrências que a outra, pois isso poderá causar um problema de *overfitting* no modelo.

Com isso, pode-se alimentar o algoritmo, esse já disponibilizado pela biblioteca “*sklearn*” da Linguagem *Python*. Esse algoritmo irá representar os dados sob a forma de um modelo, portanto um modelo nada mais é que uma representação dos dados criada por um algoritmo.

A próxima etapa é a avaliação do modelo. Como o algoritmo pode gerar resultados tendenciosos, em função de diversos fatores, como tamanho e qualidade dos dados de entrada, é importante avaliar o quão bem o algoritmo aprende a partir da sua experiência. Para os algoritmos escolhidos para esse projeto, pode-se usar um conjunto de dados de teste, formados somente pelas variáveis preditoras, para comparar a acurácia do modelo. Nesse caso, o modelo irá gerar o valor da variável de classificação de chuva e comparar com o valor dessa mesma variável que já estava registrada no conjunto de dados de teste, com isso, é possível ter uma ideia da acurácia do modelo. Foi escolhido o modelo que apresentou uma melhor acurácia, porém é válido que o modelo pode ser otimizado ao aplicar outras técnicas durante a etapa de análise exploratória, devido o resultado do modelo preditivo ser altamente influenciado pelo conjunto de dados de treino.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

Todo esse processo até a obtenção do modelo não foi realizado na nuvem, mas sim em um ambiente de desenvolvimento usando a plataforma *open source Jupyter Notebook*, pois a criação do modelo foi realizada uma única vez.

Uma vez que o modelo tenha sido criado foi preciso fazer o *deploy* deste para a plataforma *cloud* da *Azure*. Nesse caso, utilizando a *Azure Function "Analysis and Prediction"*, o modelo é carregado e então recebe os dados que foram enviados via *HTTP request* da função anterior, retornando uma saída que poderá assumir valor igual a "0" (não irá chover) ou "1" (irá chover). A função então irá gravar os dados que ela recebeu como entrada (dados sumarizados pela média diária) e o valor da variável *target* no Banco de Dados *Cosmos DB*, o qual nada mais é que um Banco de Dados dentro da *cloud Azure*, guardando os resultados no formato *JSON* para posteriores análises. O resultado dessa variável pode ser comparado com o *ThingSpeak*, para confirmar se realmente no dia anterior choveu ou não, de forma que valide a previsão do modelo, podendo esse, posteriormente, ser usado em análises em tempo real.

3.4.4. Entrega do resultado

Para a quarta etapa, foi utilizada a plataforma visual da *Microsoft Power BI*, a qual via comunicação *Client* com o Banco de Dados *Cosmos DB* recebe os dados que nele estão armazenados, e a partir de uma configuração prévia do *dashboard*, exibe estes de forma interativa para o usuário, permitindo que esse consiga visualizar as variáveis meteorológicas de temperatura, velocidade do vento, umidade relativa do ar, pressão atmosférica e previsão de chuva para qualquer dia posterior a data da consulta. Esse *Dashboard* pode ser disponibilizado online para ser acessado via *web browser*, *app* em *smartphone* e *notebook*.

4. Resultados

Primeiramente, a entrega primária desse projeto é resultante da montagem do *hardware*, composto pelo sensor HTU21D para medição de temperatura e umidade relativa do ar, sensor BMP280 para medição da pressão atmosférica, sensor PL1 para medição do índice pluviométrico e sensor AN1M para medição da velocidade do vento. Todos esses sensores foram conectados ao ESP32, o qual foi configurado no WiFi do local onde a estação se encontra instalada, nesse caso na cidade de Itupeva-SP, para envio dos dados para o *ThingSpeak*. Os sensores, assim como o ESP32, estão montados em uma estrutura composta por basicamente uma tubulação de aço inoxidável, com duas caixas de proteção com IP55, cada qual para proteger o ESP32 e a fonte de alimentação com bateria da estação, respectivamente, caso a energia elétrica da rede seja interrompida. Os sensores HTU21D e BMP280 estão dentro de uma proteção contra intempéries, conforme pode ser observado nas Figuras 7, 8, 9, 10 e 11.

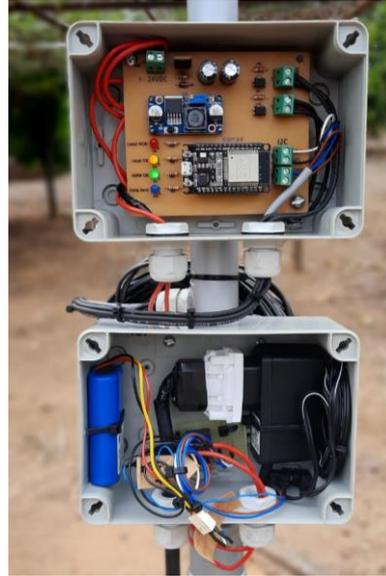
TRABALHO DE GRADUAÇÃO - ENGENHARIAS

Figura 7 - EMA Estrutura



(Fonte: Própria)

Figura 8 - Proteção ESP32 e Bateria



(Fonte: Própria)

Figura 9 - Pluviômetro



(Fonte: Própria)

Figura 10 - Proteção HTU21D e BMP280



(Fonte: Própria)

Figura 11 - Anemômetro



(Fonte: Própria)

Os dados são então enviados para o canal de *ID* 1074545 criado no *ThingSpeak* para visualização dos dados em tempo real, iniciando as medições em ambiente de produção a partir do dia 07/11/2020. Cada *feed* do canal se refere a uma variável meteorológica adquirida pelos sensores da EMA, esses dados são enviados em um intervalo de 20 segundos, com detalhe na variável pluviométrica, a qual é acumulativa, ou seja, diferente das demais variáveis, ela armazena o valor anterior até a próxima mudança de estado, quando ocorre a mudança de um dia para o outro, o ESP32 zera o valor dessa variável.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

Figura 12 - Feeds ThingSpeak



(Fonte: Própria)

A próxima etapa foi a construção do modelo preditivo no ambiente de desenvolvimento *Jupyter Notebook*. Os dados foram disponibilizados pelo INMET, com mais de 12 milhões de registros de 256 estações espalhadas pelo território nacional. Como etapa de desenvolvimento, primeiramente foram eliminados as colunas de dados que não seriam utilizadas, assim como os valores *NA* do conjunto de dados e os *outliers*. Em seguida, foi realizando uma transformação na variável de chuva, onde para qualquer valor positivo diferente de 0, a mesma iria assumir valor categórico igual a 1, caso contrário, seria 0, sendo essa a variável alvo. Com isso, os dados foram divididos em treino e teste, e os modelos foram treinados, sendo a melhor escolha o modelo de *Random Forest Classifier*, com uma precisão aproximada de 88%. Embora o *ETC* tenha apresentado melhor acurácia, esse se mostrou um tipo de arquivo muito grande para que o *deploy* seja realizado, interferindo na eficiência da aplicação.

Figura 13 - Acurácia dos Modelos

```
Acurácia Random Forest: 88.425010
Acurácia Naive Bayes: 67.694320
Acurácia kNN: 74.492031
Acurácia ExtraTreesClassifier: 89.959134
```

(Fonte: Própria)

Uma vez que os dados estejam sendo exibidos no *ThingSpeak*, foi desenvolvido as funções da aplicação analítica de dados, realizando o *deploy* de cada uma delas para a plataforma *cloud Azure* e monitorando a operação dessas de forma sequencial todos os dias às 00h59min., o que pode ser realizado observando os *logs* de operação das três funções dentro do

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

Portal da Azure e dos registros do Banco de Dados *Cosmos DB*. A primeira função, responsável por extrair os dados do *ThingSpeak* foi nomeada como “*tgthingspeakcrawler*” com um *log* de operação descrito na Figura 14. A segunda função, responsável por estruturar e sumarizar os dados é denominada “*tgdatasanitization*”, descrita na Figura 15 e a terceira função, “*tgweatherprediction*”, a qual realiza a previsão de chuva e grava o resultado no Banco de Dados *Cosmos DB* é representada na Figura 16. Todas as três funções operam de forma sequencial.

Figura 14 - Log Azure *tgthingspeakcrawler* **Figura 15 - Log Azure *tgdatasanitization***

Invocation Traces				Invocation Traces			
Date (UTC)	Success	Result Code	Duration (ms)	Date (UTC)	Success	Result Code	Duration (ms)
2020-11-10 00:59:00.005	Success	0	46524	2020-11-10 00:59:01.162	Success	200	45397

(Fonte: Própria)

(Fonte: Própria)

Figura 16 - Log Azure *tgweatherprediction*

Date (UTC)	Success	Result Code	Duration (ms)
2020-11-10 00:59:11.455	Success	200	34982

(Fonte: Própria)

Com a aplicação operando em nuvem de forma automatizada, os dados podem ser visualizados no Banco de Dados *Cosmos DB*, utilizando como exemplo a análise realizada no dia 10/11/2020, referente aos dados meteorológicos do dia 09/11/2020 na cidade de Itupeva-SP, com os registros observados na Figura 17.

Figura 17 - *Cosmos DB* Banco de Dados

id	/pr...
21735d3...	0
f01d0aaf...	0
83484ab...	0
Load more	

```

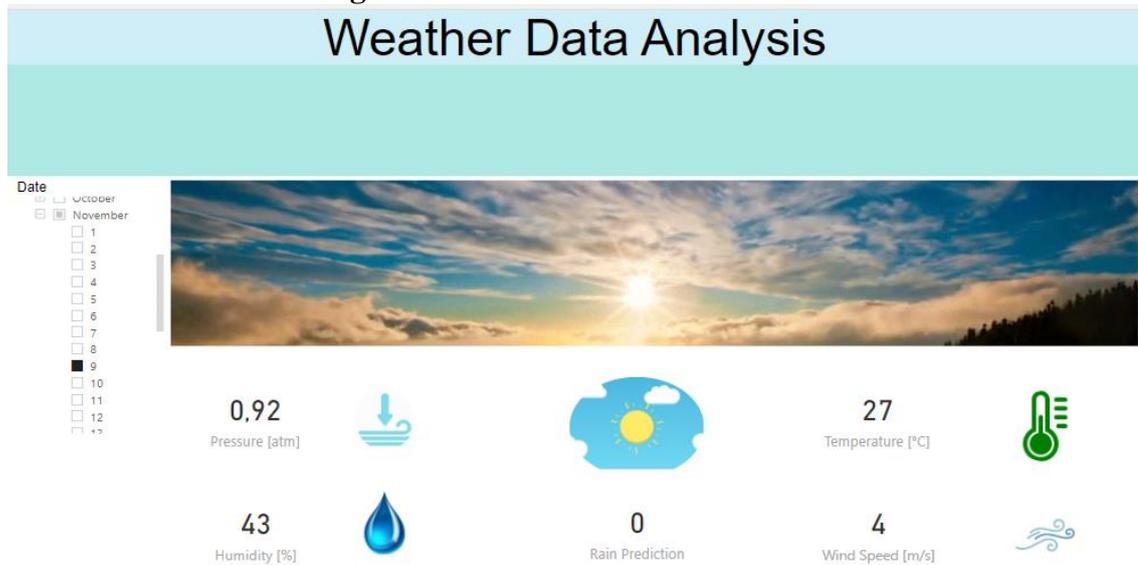
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
"prediction": 0,
"model": {
  "date": "2020-11-10",
  "temperature": 26.694947769668565,
  "humidity": 42.61844929828412,
  "wind_velocity": 3.1938237395512683,
  "pressure": 0.9204275428948541
},
"id": "83484abc-6e4c-4cbb-8a39-772b6933f0f9",
"_rid": "oDKXAPTQbJsdAAAAAAAAA==",
"_self": "dbs/oDKXAA==/colls/oDKXAPTQbJs=/docs/oDKXAPTQbJsdAAAAAAAAA==/",
"_etag": "\"0200c215-0000-0200-0000-5fa9e6020000\"",
"_attachments": "attachments/",
"_ts": 1604969986

```

(Fonte: Própria)

Para a entrega do resultado, a plataforma analítica *Power BI* foi utilizada para acessar os dados no Banco de Dados *Cosmos DB* e retornar os valores sumarizados das variáveis meteorológicas e o resultado da previsão do modelo de forma interativa e visual. Uma pré-configuração foi necessária para remover os campos que não seriam utilizados, assim como transformação do formato dos dados e configuração da área de exibição para atualização automática dos dados.

Figura 18 - Resultado Final *Power BI*



(Fonte: Própria)

5. Discussão

A estrutura desenvolvida para instalação do protótipo apresentou-se eficaz para a fixação dos sensores, proteções e caixas de proteção da bateria fonte de alimentação e placas eletrônicas, garantindo uma boa fixação da mesma no solo. A estação se mostrou resistente a condições adversas de temperatura, umidade e pluviosidade, validando a proteção desenvolvida para os sensores HTU21D e BMP280. Com relação a comunicação WiFi do ESP32, a mesma se mostrou estável, apresentando em alguns momentos falhas de conexão, sendo necessário realizar alterações na sua rotina de programação para correção desse problema. Além disso, em momentos nos quais ocorre a queda de energia, embora a estação permaneça funcionando devido a bateria de *backup*, o roteador WiFi perde a alimentação, fazendo com que a estação perca sua conexão e por consequência deixe de enviar os dados a plataforma IoT, uma forma de resolver esse problema seria realizar a instalação de um *NoBreak* para suprir o roteador.

A API IoT *ThingSpeak* apresentou boa resposta para representação e registro dos dados obtidos pela EMA, sendo apenas necessário ajustes na escala dos gráficos de cada variável. Para comunicação com a plataforma *Cloud*, as funções operaram com sucesso de forma orquestrada, acessando o *ThingSpeak* no horário estipulado e com duração média de 10 segundos para execução de todo o processo de análise até a gravação do resultado no *Cosmos DB*. A análise dos dados se restringiu apenas a entrega da média e do resultado do modelo preditivo, porém pode-se acrescentar mais parâmetros, como mediana, valor máximo ou valor mínimo a partir de uma alteração no script da função no ambiente cloud, mostrando mais uma vez a fácil escalabilidade e confiabilidade de se trabalhar nesse ambiente.

O *Power BI* foi configurado para exibir os dados armazenados no *Cosmos DB* referente aos seus respectivos dias, seguindo um *Dashboard* padronizado, permitindo ao usuário escolher o dia que deseja visualizar as variáveis climatológicas, inclusive a verificação da variável preditiva de chuva referente a respectiva data e possível comparação com o que realmente foi detectado pelo pluviômetro no *ThingSpeak*.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

6. Conclusão

Com o desenvolvimento deste trabalho foi possível comprovar a possibilidade de efetuar a instalação de uma EMA integrada com uma plataforma *cloud* analítica de dados. Os sensores selecionados apresentaram as características desejadas para a leitura das variáveis meteorológicas com boa precisão, o que tornou possível uma análise confiável do clima do local onde a estação se encontra instalada. Por se tratar de uma EMA de baixo custo, os sensores não são profissionais, porém atendem a necessidade com uma boa precisão. Os sensores de pressão, umidade relativa e temperatura tiveram um custo total de 40 reais, enquanto os sensores anemômetro e pluviômetro um custo de 300 reais, o microcontrolador ESP32 e todos os componentes da placa tiveram um custo médio de 70 reais e a estrutura mecânica da estação saiu pelo valor de 150 reais, a plataforma *ThingSpeak* é *Open Source*, a plataforma *Azure* e *PowerBI* não apresentaram custo devido à baixa necessidade de recursos executados. Desta forma o custo total para desenvolvimento de toda aplicação se restringe apenas ao valor gerado pelo desenvolvimento do *hardware* da estação, sendo igual a 560 reais.

A hospedagem de aplicações em ambientes *cloud* é um tema de muita importância e em processo de rápida expansão na área de engenharia, principalmente em IoT, dessa forma, a aplicação desenvolvida para análise dos dados da EMA forneceu alta confiabilidade e disponibilidade para operação com um custo nulo, ou seja, abaixo dos requisitos mínimos de cobrança do provedor *cloud*, operando de forma orquestrada e automática. Com isso, foi possível integrar de forma ágil uma plataforma IoT com um sistema de aplicação em *Cloud*, gerando visualização interativa de dados para o usuário. Uma aplicação desse tipo pode se restringir desde apenas uma análise simples dos dados meteorológicos, até *deploy* de modelos preditivos, como o que foi apresentado nesse projeto, fornecendo informações importantes em formato de *Dashboard*.

O modelo de integração entre plataforma *cloud* e IoT está em constante crescimento e desenvolvimento a cada ano, com novas tecnologias no mercado e ferramentas de desenvolvimento, porém a exploração de recursos *open source* ainda é um desafio, principalmente quando o objetivo é a produção para fins comerciais. No entanto, para aplicações acadêmicas ou particulares é possível desenvolver usando recursos gratuitos, de acesso simplificado, com bons resultados e qualidade semelhante à de ferramentas proprietárias, conforme foi possível comprovar com o desenvolvimento desse projeto.

7. Referências Bibliográficas

BRUCE, P; BRUCE, A. **Estatística Prática para Cientista de Dados – 50 Conceitos Essenciais**. Rio de Janeiro: Alta Books, 2019. 320 p.

FOREMAN, W.J. **Data Smart – Usando Data Science para Transformar Informação em Insight**. Rio de Janeiro: Alta Books, 2016. 448 p.

LATHI, P.B. **Sinais e Sistemas Lineares**. Rio de Janeiro: Bookman, 2007. 856 p.

MARSLAND, S. **Machine Learning – An Algorithmic Perspective**. New York: CRC Press, 2015. 452 p.

NALLAKARUPPAN, K. M.; KUMARAN, U. S. **IoT based Machine Learning Techniques for Climate Predictive Analysis**. International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, v. 7, Issue – 5S2, January 2019.

STRASSBURGER, S.A. et al. **Comparação da Temperatura do ar obtida por estação meteorológica convencional e automática**. Revista Brasileira de Meteorologia, v. 26, n. 2, p. 273 - 278, 2011.

TRABALHO DE GRADUAÇÃO - ENGENHARIAS

PALMIERI, M. A. et al. **Sistema Automatizado para Coleta de Dados de Umidade Relativa e Temperatura do Ar**. Engenharia Agrícola – ESALQ USP, Jaboticabal, v. 34, n. 4, p. 636-648, jul/ago. 2014.

PRASANNA, M. et al. **An Intelligent Weather Monitoring System using the Internet of Things**. International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, v. 8, Issue – 4, November 2019.

FERNANDES, G. A. G; ARAÚJO, L. H; GOMES, C. C. R. **Sistema Automatizado de Aquisição de Dados Meteorológicos**. 7ª Jornada Acadêmica – Unidade Universitária de Santa Helena de Goiás – Crescimento Regional Inovação e tecnologia no mercado de trabalho, 18 – 23 Novembro, 2013.

INSTITUTO NACIONAL DE METEOROLOGIA DO BRASIL – INMET: A Estação Meteorológica e seu Observador. Consultor ad hoc do Inmet Rubens Leite Vianello, 2011.

COSTA, G. M. M. A Meteorologia Aeronáutica no Aeroporto de Guarulhos. **In: SIMPÓSIO DE TRANSPORTE AÉREO**, 7., 2008, Rio de Janeiro. Empresa Brasileira de Infra-Estrutura Aeroportuária – INFRAERO, 2008, p. 539-550, Tr. 500.

WORLD METEOROLOGICAL ORGANIZATION: Guide to Meteorological Instruments and Methods of Observation. WMO – No. 8. 2008. Geneva.

MAUREIRA, G. A.M; OLDENHOF, D; TEERNSTRA, L. **ThingSpeak – an API and Web Service for the Internet of Things**. Leiden Institute of Advanced Computer Science, 2011.

SUN, B. et al. **Variability and Trends in U.S. Cloud Cover: ISCCP, PARTMOS-x, and CLARA-A1 Compared to Homogeneity-Adjusted Weather Observations**. American Meteorological Society, DOI: 10.11775 / JCLI-D-14-00805.1, p. 4375 – 4389, 1 June 2015.

GÉRON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. Rio de Janeiro: Alta Books, 2019, 555 p.

BHARADI, A.V. et al. **Cloud based NoSQL Database for Iris based Biometric System Azure based Cosmos DB implementation**. Universal Review, Volume 7, Issue X, ISSN NO : 2277-2723, October 2018.

RUSSO, M; FERRARI, A. **Introducing Microsoft Power BI**. Microsoft Press, 2016. ISBN: 978-1-5093-0228-4