

**ANAXIMANDRO RODRIGUES LEITE N154JG-0**

**RAFAEL DE OLIVEIRA SANTANA D3014C-7**

**WASHIGTON THIAGO ALVES CORREA DOS SANTOS D41640-5**

**ESTAÇÃO METEOROLÓGICA:**

Temperatura, humidade do ar e atmosfera.

**SOROCABA - SP**

**2017**

**ANAXIMANDRO RODRIGUES LEITE N154JG-0**

**RAFAEL DE OLIVEIRA SANTANA D3014C-7**

**WASHIGTON THIAGO ALVES CORREA DOS SANTOS D41640-5**

**ESTAÇÃO METEOROLÓGICA:**

Temperatura, humidade do ar e atmosfera.

Atividades Práticas Supervisionadas – trabalho apresentado como exigência para a avaliação do 3º semestre, do curso de Ciência da Computação da Universidade Paulista, sob orientação dos professores do semestre.

**SOROCABA - SP**

**2017**

**RESUMO**

Nos dias de hoje, e possível identificarmos vários aspectos do nosso ambiente em dados coletados de sensores, tais como Temperatura do ambiente, humidade do ar e/ou presença ou não de precipitação atmosférica, vemos com este projeto apresentar o desenvolvimento, analise, coleta de dados e distribuição destes dados, em sua forma técnica e por vez em seu aspecto genérico para qualquer usuário.

Palavras-Chave: Ambiente, Temperatura, Projeto, Dados, Distribuição.

**ABSTRACT**

Nowadays, it is possible to identify several aspects of our environment in data collected from sensors, such as ambient temperature, air humidity and / or the presence or not of atmospheric precipitation, with this project we present the development, analysis, collection of data and distribution of this data, in its technical form and in its generic aspect for any user.

Keywords: Environment, Temperature, Design, Data, Distribution.

Sumário

[1 OBJETIVO 6](#_Toc514406562)

[2 INTRODUÇÃO 7](#_Toc514406563)

[3 ESTAÇÃO METEOROLIGICA 9](#_Toc514406564)

[3.1 Hardware 9](#_Toc514406565)

[3.1.1 Arduino Uno 9](#_Toc514406566)

[3.1.2 Sensor de temperatura e pressão BMP180 10](#_Toc514406567)

[3.1.3 Sensor de temperatura e umidade DHT11 11](#_Toc514406568)

[3.1.4 Sensor de chuva YL-83 12](#_Toc514406569)

[3.2 Software 13](#_Toc514406570)

[3.2.1 Linguagem de Programação JAVA 13](#_Toc514406571)

# 1 OBJETIVO

Este trabalho visa esclarecer e desenvolver um sistema de estação meteorológica para monitoramento de variáveis ambientais, tais quais utilizando linguagens de programação mais adequadas e com melhor funcionamento para seu funcionamento. Apresentamos o conceito da linguagem e dispositivo deste projeto, tal seu funcionamento.

Abordamos o intuito de programação orientada a objeto, sua aplicação neste projeto, vantagens e desvantagens, assim como a estruturação da coleta de dados da estação, apresentação dos dados ao usuário e desenvolvimento.

# 2 INTRODUÇÃO

Uma estação meteorológica consiste por uma unidade de memória central, que por sua vez esta ligava a vários sensores para captação de parâmetros meteorológicos, sendo estes os mais variados tipos, pressão atmosférica, temperatura, umidade relativa do ar, entre vários outros, a definição de quais sensores serão ligados depende da necessidade de cada estação.

A importância da meteorologia e datada destes tempos antigos, já que a previsão de tempo e de suma importância para varias áreas da sociedade.

A história da meteorologia começa na era primitiva, quando o homem já tinha uma preocupação com as condições de tempo. Elas eram fatores importantes para sua sobrevivência, tanto na questão da alimentação e abrigo, quanto na escolha dos locais para repousar. (Bem Paraná, Meteorologia: sua origem e evolução no tempo, 28/02/2018).

Com o passar dos anos, a tecnologia foi evoluindo e estação com mais capacitação e precisão foram feitas, hoje já e possível obter parâmetros de meteorologia com pequenos equipamentos portáteis, existe atualmente vários dispositivos que podem satisfazer as necessidades mais simples como as mais avançadas, entre essas escolhemos o Arduino, um dispositivo surgido em meados de 2005, possui um o conceito de hardware livre, ou seja, qualquer um com conhecimento pode-se modificar, montar e melhorar as funcionalidades e aspectos do hardware. (Adilson Thomsem, O que é Arduino, 02/09/2014).

Juntamente com o Arduino, utilizamos uma Protoboard de 400 pontos, interligando ele esta os sensores de temperatura ambiente, humidade relativa do ar, presença ou não de precipitação atmosférica e pressão atmosférica.

A coleta de dados é feita através cabo USB diretamente para um computador, para captação dos dados é preciso elaborar um software, sendo assim é mais comumente utilizado a linguagem de programação em C ou C++, respectivamente surgida nos anos de 1970 e 1980, o C++ tem a vantagem de utilizar o conceito de Programação orientada a objeto, para configuração do Arduino se é utilizado à linguagem C, porém para fazer o software que ira apresentar os dados e os armazenas para outros fins, utilizamos o JAVA.

Surgida após 1990, este tipo de linguagem garante um independência de plataforma, e nos garante uma melhor funcionalidade em qualquer tipo maquina, dado que quando compilado existe um emulador conhecido como JVM (Java Virtual Machine) que ajuda a rodar este programa em qualquer computador (Thiago Vinícius, Java: história e principais conceitos, 2018).

Após coleta de dados do Arduino, temos que interpretar esses dados via software e passar ele para o usuário de forma que ele compreenda, para isso, elaboramos uma pagina Web, utilizando HTML, criado em 1990 por Tim Berners-Lee, se tornou uma das maiores linguagens de programação para pagina Web (David William, Front End Brasil – A história do HTML, 21/04/2012), atualmente já e possível integrar outros tipos de linguagens junto ao HTML ou o mesmo em outras linguagens, assim utilizamos toda a base em PHP, linguagem esta de inicio em 1994 e com suas modificações até os dias atuais, se tornou uma excelente plataforma de trabalho.

# 3 ESTAÇÃO METEOROLIGICA

## 3.1 Hardware

Existem vários tipos de estações meteorológicas portáteis, neste projeto, elaboramos está estação desde sua placa principal até seus sensores, a grande diferença parte do desenvolvimento interpessoal para realização do mesmo. Para realização inicial do projeto escolhemos, escolhemos a placa que ira gerenciar os sensores da estação, existe varias opções no mercado como a BlackBoard e Teensy, porém a única que possui um grande número de artigos, tutoriais e afins e a Arduino, por este motivo a escolhemos entre as outras.

### 3.1.1 Arduino Uno

Criado em 2005 por um grupo de pesquisadores (Tom Igoe, Gianluca Martino, Massimo Banzi, David Cuartielles e David Mellis), nesta época (2005) o custo de uma placa para desenvolvimento era muito caro para os estudantes, a ideia inicial era disponibilizar uma plataforma livre, na qual todos podem modificar, montar e melhorar, e por sua vez fosse acessível, assim abrangendo um maior número de pessoas.

A principio se começou vender em porta de faculdades, com o intuito de ajudar os estudantes, era comercializado sem fins lucrativos, já que o valor cobrado era o mesmo para confecção da placa, após uns meses o Arduino passou por algumas modificações, sua infraestrutura agora aceitava conexão por USB, com este diferencial a escala de comercialização aumentou, se tornando uma placa popular.

Por ser uma plataforma aberta e conforte o tempo foi passando, vários projetos, modificações e melhorias para a placa foram surgindo, feitas principalmente pelos próprios usuários, e por vez algumas adotadas direto para a próxima atualização da placa, seu portfólio de utilidade e gigantesco, desde controle de luzes num ambiente domiciliar ou então uma impressora 3D (Arduino the Documentary, 2010). Nos dias de hoje e possível encontrar varias versões da placa Arduino, entre elas, Arduino UNO, Arduino Leonardo, Arduino MEGA 2560 entre outras tantas, “... existem também os chamados Shields, que são placas que você encaixa no Arduino para expandir suas funcionalidades...” (Adilson Thomsen, O que é Arduino?, 02/09/2014).

Escolhemos para este projeto o Arduino UNO (segue figura 1), escolhemos está por ser a placa inicial para qualquer amador, e também por sua simplicidade de programar como grande atrativo, todos os periféricos necessários para estação meteorológica desde projeto são compatíveis com está placa, para isso utilizamos uma Protoboard de 400 pontos, nele acrescentamos os sensores e o ligamos no Arduino. A versão UNO consta com conectividade via USB, assim não é preciso adquirir um Arduino Internet Shield, um Shield que permite acesso ao Arduino via remoto (WiFi).

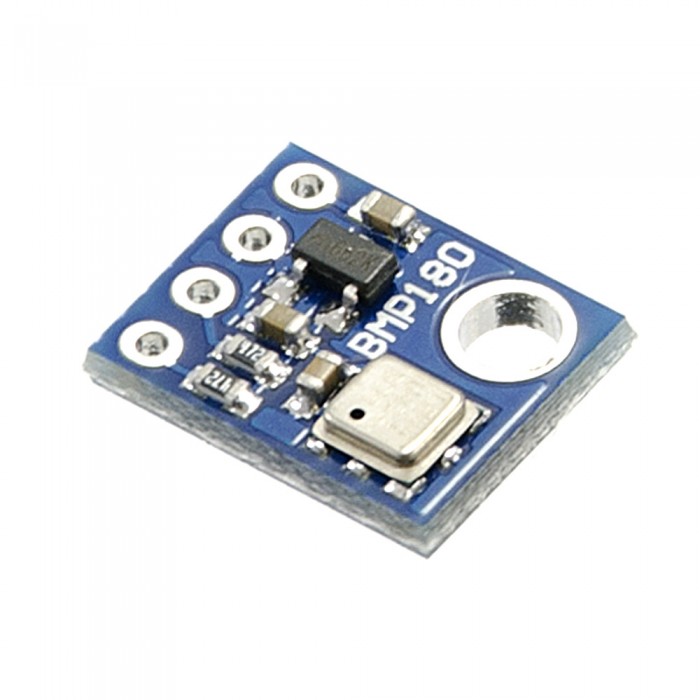
Figura 1 – Arduino UNO



Fonte: http://store.arduino.cc/usa/arduino-uno-rev3, 18/05/2018.

### 3.1.2 Sensor de temperatura e pressão BMP180

O Sensor BMP 180 tem como finalidade medir a Pressão atmosférica do ar, que basicamente é a pressão que o ar da atmosfera exerce sobre o planeta, tal pressão sofre drásticas mudanças conforme também a diferença de altitude em que o sensor se encontra. Este mesmo sensor faz a medição da Temperatura ambiente, ele fornece a temperatura em Celsius, como é o próprio sensor que faz essa medição de temperatura, para melhor acuracidade e importante deixar o mesmo em local arejado e em ambiente externo.



Fonte: Filipeflop, <https://www.filipeflop.com/blog/temperatura-pressao-bmp180-arduino/>, 17/05/2018.

O sensor BMP180 é o sucessor direto do BMP085, embora valha destacar que ele seja totalmente compatível com sua versão anterior. Apesar de medir a temperatura este sensor tem como finalidade medir a pressão atmosférica e com este dado determinar a altitude.

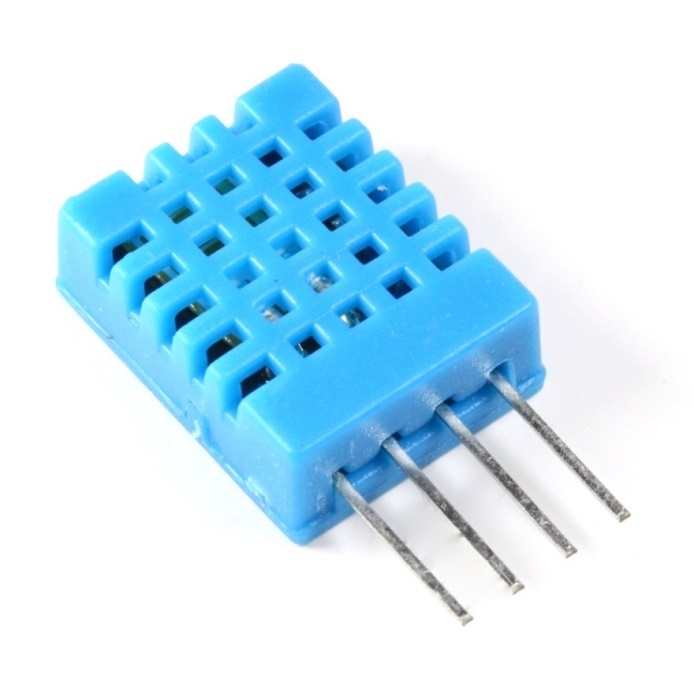
O sensor funciona com alimentação 1.8 a 3.6V, e possui um regulador de tensão embutido que permite que você o conecte normalmente às placas Arduino com nível de sinal de 5V, como o Arduino Uno, a conexão ao Arduino utiliza a interface I2C, por meio dos pinos analógicos 4 (SDA) e 5 (SCL). No módulo temos somente 4 pinos : Vin (1,8 à 3.6V), GND, SCL e DAS, para este projeto conectamos o Vin do módulo ao pino 3.3V do Arduino.

### 3.1.3 Sensor de temperatura e umidade DHT11

O sensor DHT11 e um sensor para captar a temperatura ambiente e a umidade do ar, o utilizamos principalmente pelo sensor de umidade, pois o sensor BMP180 já possui um captor de temperatura ambiente. Seu sucessor o DHT12 tem como finalidade os mesmo conceitos, porém os fazem com uma maior taxa de precisão, o DHT11 possui a taxa de umidade do ar entre 20% e 80% com precisão de 5%, e a medição de temperatura vai de 0ºC a 50ºC, com precisão de 2ºC, já seu sucessor respectivamente mede a umidade entre 0% e 100% com precisão de 2% a 5%, e entre -40ºC e 125ºC com precisão de 0,5ºC.

Optamos por escolher o DHT11 por não precisarmos de uma grande precisão nos dados coletados, e nem de medição de temperaturas tão bruscas, como por exemplo, graus Celsius negativos ou muito elevados.

Este sensor possui 4 terminais sendo que somente 3 são usados: GND, VCC e Dados, neste projeto conectamos o pino de dados do DHT11 ao pino 2 do Arduino Uno, e os outros pinos nos GND e VCC do Arduino,



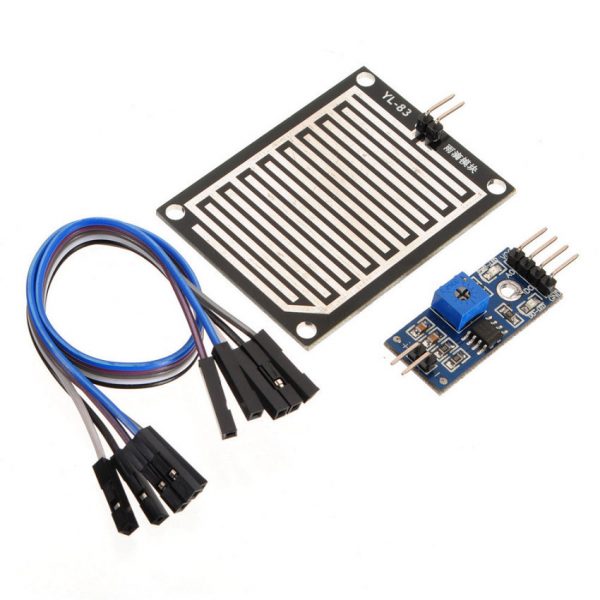
Fonte: Filipeflop, https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-dht11/, 17/05/2018.

### 3.1.4 Sensor de chuva YL-83

O sensor de chuva YL-83 propriamente dito, não prevê se ira chover ou não, este tipo de previsão só e possível por meio de satélites, seu dispositivo é composto por uma placa que é o próprio sensor de chuva/umidade do ambiente, ela e inoxidável e detecta a quantidade de umidade (no caso está seria a chuva), ele consta com duas funções em seu software.

Um sensor de umidade para que seja configurado, assim você estipula uma quantidade de umidade para que ele fale se esta chovendo ou não, e outro você categoriza esses valores, colocando em parâmetros de pouca chuva, moderada, alta ou outros.

Em conjunto deste sensor temos um módulo principal, este que contem todo o circuito de controle e ira se comunicar com o micro-controlador, no caso deste projeto o Arduino. O módulo de controle, tem 2 pinos que vão se comunicar com a placa do sensor, e na outra extremidade, 4 pinos de sinal e alimentação : A0 (sinal analógico), D0 (sinal digital), GND e Vcc. A alimentação vai de 3.3 a 5 volts.



Fonte: Filipeflop, <https://www.filipeflop.com/produto/sensor-de-chuva/>, 18/05/2018.

## 3.2 Software

Por utilizarmos o dispositivo do Arduino, a própria empresa nos disponibiliza uma ferramenta chamada Arduino IDE, com ele pode fazer a programação dos dados coletado pelos sensores, para tal utilizamos a linguagem de programação C (esta programação também já está disponível na Biblioteca do Arduino, este caso para os sensores utilizados neste projeto), após essa configuração dos dados, temos que transportar esses dados para outra plataforma, para que possamos armazena-lo e utilizarmos em uma melhor escala, para isso utilizamos o JAVA, e para transferir esses dados ao usuário utilizamos uma pagina WEB em PHP.

### 3.2.1 Linguagem de Programação JAVA

O JAVA se deu inicio em 1995, porém já em 1990 a empresa Sun Microsystem estava desenvolvendo um aplicativo set-top box (basicamente um decodificador de sinal), nesta época eles procuravam desenvolver um programa portátil que poderia se rodar em qualquer micro chip, tentaram utilizando a linguagem de programação C/C++, mas encontraram limitações que iriam dificultar o foco principal do projeto, assim eles começaram fazer uma remodelação na linguagem que estava sendo usada e fizeram uma nova, chamada inicialmente de “Oak”, embora fosse uma inovação no mercado, não se fez o sucesso (Olhar Digital, Linguagem Java: Um pouco da historia, 07/07/2013).

Após uns anos, com a popularidade da internet e se tornando uma parte ferramenta indispensável para sociedade, a Sun Microsystem se adiantou e reformulou sua antiga linguagem “Oak”, adotando um novo nome, que se deu inicio o JAVA.

Com isso, seu sucesso foi grande, pois ainda não se tinha uma linguagem tão portátil, fora isto ela constava outras vantagens referente a outras linguagens, sendo seu suporte a programação orientada a objeto, segurança, uma linguagem simples, alta performance e dinamismo, entre outras qualidade. Dentro destas características está e independência de plataforma.

Hoje a maioria das linguagens sofrem na transferência de plataforma quando o sistema desenvolvido tem que migrar para outra plataforma, pois quando compilado um programa a ação do compilador é transformar o arquivo-fonte em código de máquina.

Por exemplo, se o programa desenvolvido for compilado em sistemas Macintosh, mais tarde terão problemas quando forem migrar para plataformas Intel, tendo que transferir o código fonte para a plataforma Intel e fazer a compilação novamente para produzir o código de máquina específico para este sistema. Muitas vezes o programador terá que alterar o código fonte antes de efetuar a compilação para a nova plataforma, esse motivo acontece por possuírem arquiteturas de processador diferenciadas. (Thiago Vinícius, Devmedia Java: História e principais conceitos,17/07/2012).

### 3.2.2 Programação Orientada a Objeto

Enraizado em seu projeto o JAVA possui a programação orientada a objeto, este conceito já existia desde a década de 1970, porém se deu sucesso com o JAVA que se popularizou e tornou uma das maiores linguagens de programação. Programação orientada a objeto, ou POO, consiste em uma dinâmica em trazer a realidade para dentro da programação, como em nosso dia-a-dia temos o costume de tratar as coisas como um Objeto, em POO este conceito evolui, fazendo o uso desse método de programação, podemos transformar tudo em objeto dentro do programa. Fazendo uso do POO e tendo uma boa pratica de programação e possível ter uma grande economia de tempo para futuras manutenções no software, ou um entendimento melhor do mesmo, torno o programa mais limpo e leve, e com isso é possível desenvolver softwares muito mais complexos.

Os principais conceitos do POO são: Classe (Atributos e métodos), Objeto, Encapsulamento, Herança, Sobrecarga e Polimorfismo, sendo eles respectivamente.

Classe é a arquitetura do objeto, para que se possa ter um objeto temos que ter a classe, e é na classe que inserimos as características do Objeto como atributos e métodos. O Objeto é o produto final, com as características inseridas na classe criada, a partir dela podemos instanciar um objeto, são os objetos que dirão o rumo do nosso programa.

Podemos associar o encapsulamento como se fosse um dispositivo de segurança, é uma forma de esconder dados privativos ou protegidos do nosso objeto para que terceiros não tenham acesso, a herança é a habilidade de herdar características de outros objetos, sobrecarga é um meio de termos ações com o mesmo nome em nossas classes, no caso do JAVA ele não permite métodos com o mesmo nome, apenas se usar esse conceito neles. Polimorfismo tem como objetivo diminuir a quantidade de código escrito, aumentando a clareza e a facilidade de manutenção.

### 3.2.3 Aplicação do JAVA no projeto