**UNIVERSIDADE PAULISTA**

**ANAXIMANDRO RODRIGUES LEITE N154JG-0**

**RAFAEL DE OLIVEIRA SANTANA D3014C-7**

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

**ESTAÇÃO METEOROLÓGICA:**

Temperatura, umidade 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, umidade do ar e atmosfera.

Atividades Práticas Supervisionadas-trabalho apresentado como exigência para 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, umidade 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](#_Toc530531606)

[2 INTRODUÇÃO 6](#_Toc530531607)

[3 ESTAÇÃO METEOROLIGICA 8](#_Toc530531608)

[3.1 Hardware 8](#_Toc530531609)

[3.1.1 Arduino Uno 8](#_Toc530531610)

[3.1.2 Sensor de temperatura e pressão BMP180 9](#_Toc530531611)

[3.1.3 Sensor de temperatura e umidade DHT11 11](#_Toc530531612)

[3.1.4 Sensor de chuva YL-83 12](#_Toc530531613)

[3.2 Software 13](#_Toc530531614)

[3.2.1 Linguagem de Programação JAVA 13](#_Toc530531615)

[3.2.2 Programação Orientada a Objeto 14](#_Toc530531616)

[3.2.3 Aplicação do JAVA no projeto 15](#_Toc530531617)

[3.3 Página Web 16](#_Toc530531618)

[3.3.1 Linguagem de programação PHP e HTML 16](#_Toc530531619)

[3.3.2 Linguagem de Programação CSS 17](#_Toc530531620)

[4 LINHAS DE CÓDIGO 19](#_Toc530531621)

[4.1 Linhas de Código página WEB 19](#_Toc530531622)

[4.2 Linhas de Códigos JAVA 19](#_Toc530531623)

[5 BIBLIOGRAFIA 19](#_Toc530531624)

# 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, umidade 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 máquina, 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 página Web, utilizando HTML, criado em 1990 por Tim Berners-Lee, se tornou uma das maiores linguagens de programação para página 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 início 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 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, MassimoBanzi, 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.

Começou-se a 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).

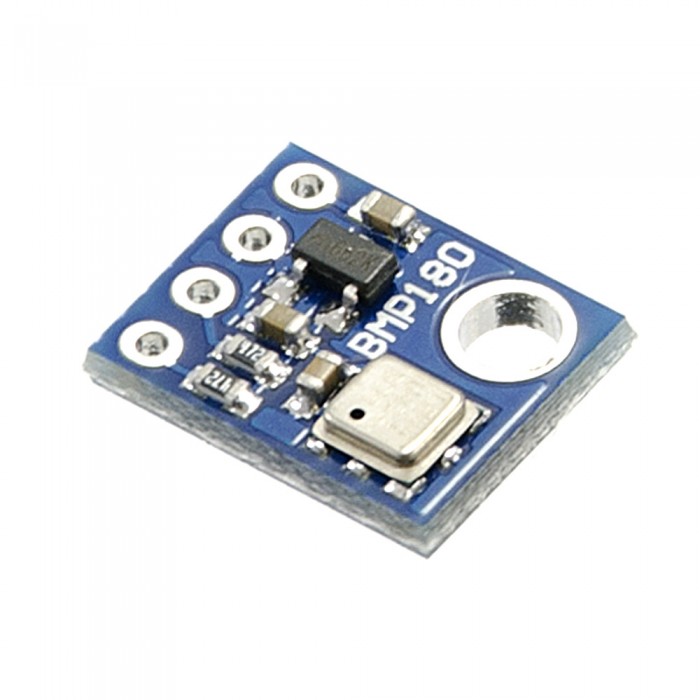
Figura 1 – Arduino UNO



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 precisão é 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.

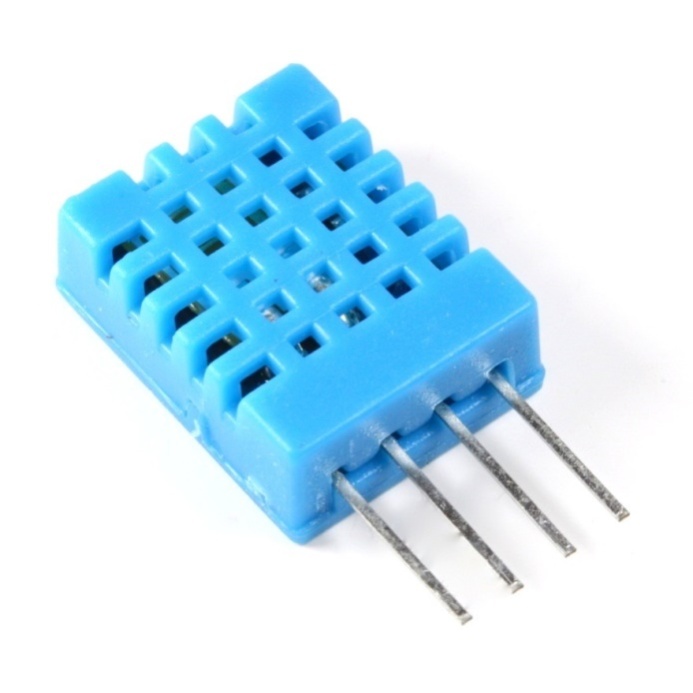
O sensor funciona com alimentação 1.8 a 3.6V, e possui um regulador de tensão embutido que permite que você o conecte normalmente às placas Arduino com nível de sinal de 5V, como o Arduino Uno, a conexão ao Arduino utiliza a interface I2C, por meio dos pinos analógicos 4 (SDA) e 5 (SCL). No módulo temos somente 4 pinos: Vin (1,8 à 3.6V), GND, SCL e DAS, para este projeto conectaram 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 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 vão 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 conectaram o pino de dados do DHT11 ao pino 2 do Arduino Uno, e os outros pinos nos GND e VCC do Arduino.



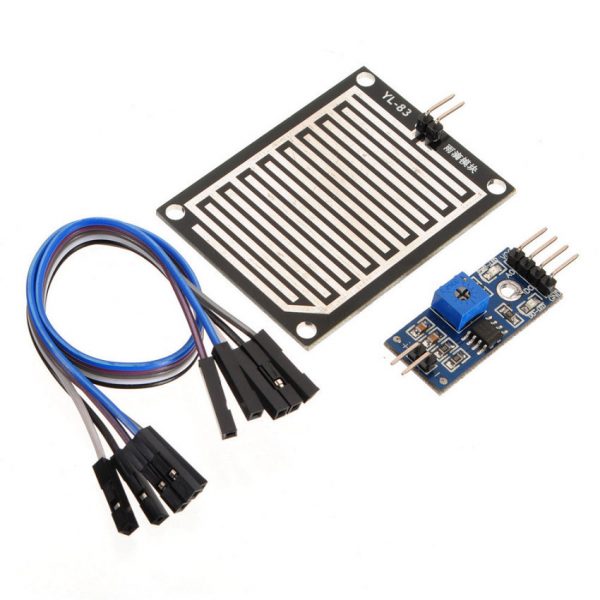
Fonte: Filipeflop, https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-dht11/, 17/05/2018.

### 3.1.4 Sensor de chuva YL-83

O sensor de chuva YL-83 propriamente dito, não prevê se vai 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 microcontrolador, 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 armazená-lo e utilizarmos em uma melhor escala, para isso utilizamos o JAVA, e para transferir esses dados ao usuário utilizamos uma página 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 tentou 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 desempenho e dinamismo, entre outras qualidades. Dentro destas características está e independência de plataforma.

Hoje as maiorias 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, pode 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

Primeiramente adicionamos a biblioteca RXTX que faz a manipulação das portas seriais, esta foi implementada em todas as classes Java. O projeto foi divido em pacotes, sendo um para o Back-end, um para o Front-end e outro com a Classe Main, servindo de controlador dos dados. Dentro do pacote Front-end adicionamos um formulário principal, criado como FormJDialog.

Criamos primeira uma classe abstrata contendo somente as principais variáveis do sistema, chamada absPropriedades, com os devidos construtores para o tratamento correto no decorrer do programa. Assim adotamos o conceito de herança em nosso projeto.

Após isso criamos uma classe onde é estabelecida a conexão com a porta Serial onde foi conectado o Arduino, enviando através do formulário principal o nome da porta com seu devido Timeout e seu Baudrate. Essa classe recebe as ondas de entrada da porta serial, que são tratadas pelo pacote RXTX e convertidas em textos.

Também criamos a classe onde é iniciada a leitura de dados vindo da porta Serial, nesta são aplicados eventos através da porta serial para estabelecer conexão corretamente com a porta e não permitir possíveis falhas na transferência de dados, toda essa conexão é feita em tempo real, ou seja, via buffer. Estas duas (abertura e leitura) foram instanciadas no formulário principal, aplicadas na função de MouseClicked do botão Conectar.

Após isso é iniciado o contato com o Arduino, recebendo os dados oriundos dos sensores e sendo armazenado de uma String chamada Dadoslidos. Esta String é instanciada na área de Texto do formulário principal para melhor visualização e design para o usuário. Essa string então é enviada à classe para salvar em arquivo JSON, pois este é lido pela WebPage em PHP.

Ainda no formulário principal adicionamos o botão DESCONECTAR, onde instancia outra classe criada para somente fechar a comunicação com a porta Serial (Arduino). Os métodos utilizados foram da classe própria do Java e da biblioteca RXTX, contando com abertura de porta serial, leitura de porta serial, identificação de porta serial.

## 3.3 Página Web

### 3.3.1 Linguagem de programação PHP e HTML

Para construção da apresentação dos dados, foi denotado à utilização de uma página WEB, existem linguagens conhecidas globalmente para esta demanda, no entanto esta sendo utilizado o conjunto de PHP com o próprio HTML.

“A linguagem PHP é uma linguagem de script open-source muito utilizada, é especialmente adequada para o desenvolvimento Web e que pode ser embutida dentro do HTML... A melhor coisa em usar o PHP é que ele é extremamente simples para um iniciante, mas oferece muitos recursos avançados para um programador profissional. Não tenha medo de ler a longa lista de recursos do PHP. Pode entrar com tudo, o mais rápido que puder, e começar a escrever scripts simples em poucas horas...”. (The PHP Group, O que é PHP ?, 2001-2018 The PHP Group)

O site em PHP é feito a partir do método de programação Orientado a Objeto. Primeiro o site foi programado e desenvolvido em HTML, seguindo o princípio de organização e limpeza de programação exigida, após essa etapa o código foi migrado para PHP no qual se começou o processo de divisão, que constitui em separar a parte de programação em partes, assim a própria parte programada em HTML e mais aproveitada.

O header, que no caso é o cabeçalho, foi separado devido ao site possuir 2 (duas) páginas com essa separação ele pode ser aproveitado por ambas às páginas sem que precise ser digitado novamente. Dentro do header foi disponibilizado o Nav. (barra de navegação), as instruções ao CSS e os comandos que permitem que o site possa se tornar Responsivo, assim ele se ajustará a qualquer tela permitindo melhor funcionalidade e aproveitamento por parte do usuário. Um dos comandos essências para isso é o “Viewport”, pois ele permite que o seu Browser receba instruções para controlar as dimensões da página, isso junto com a programação utilizando valores em porcentagem no Código em Cascata (CSS) para suas medidas.

O Footer é uma parte do site destinada ao rodapé, não é colocada muita informação nessa parte, porém o copyright e o nome da empresa são disponíveis nele, a cor escolhida tal foi definida a partir do background do index.

Logo passamos para index da página, onde pudemos localizar todo o corpo e o conteúdo chave da página, no nosso caso é onde esta disponibilizada a tabela com os valores recebidos a partir do Arduino (JAVA), e com isso o usuário terá uma previsão climática do dia atual. Um referencial do dia foi colocado ao lado superior direito da página para que caso o usuário queira anotar alguma informação, tudo que ele necessita estará ao seu alcance, data, umidade e a probabilidade de chuva no dia em questão. Ainda no index, estão localizadas as referências que conectam com os demais elementos necessários para o site (header e footer). A tabela foi à solução mais viável para este site, pois todos os dados seriam apenas solicitados e recebidos sem que quaisquer cálculos sejam feitos no mesmo, a mesma tabela pode ser vista dos móbiles, como o site é responsivo a função de dia e hora, permite que o site receba as informações do servidor que hospeda o PHP.

Há página “Sobre”, é estruturalmente parecido ao index, porém foram alteradas algumas variáveis, pois não havia necessidade de contato com a equipe e devido a isso, nesta página esclarecemos alguns aspectos que como foram coletados os dados, e o motivo da criação deste site.

### 3.3.2 Linguagem de Programação CSS

A linguagem de programação CSS ou Códigos em Cascata é considerado o primórdio de todo site Web, nesta parte da programação está contido toda a estrutura do site, bem como definições de tamanhos, letras, cores, backgrounds, reconhecimento de tamanho de telas, dentre outras coisas. Existem 2 (dois) CSS incorporados ao projeto, o primeiro é o CSS padrão, referente ao site acima de 768px (pixels) de largura que são basicamente telas apresentadas em notebooks e desktops, e o segundo é o mediaQuerry, ele analisa a partir das informações do viewport o tamanho da tela e ajuda o site a se adaptar da melhor forma possível a telas menores que o tamanho padrão, para fazer esses ajustes é necessário moldar o site novamente para que ele fique com as proporções certas, porém desta vez com o preview do site em um celular para que houvesse mais precisão e funcionalidade.

# 4 LINHAS DE CÓDIGO

## 4.1 Linhas de Código página WEB

## 4.2 Linhas de Códigos JAVA

# 5 BIBLIOGRAFIA

14/05/2018

<http://www.inmet.gov.br/portal/index.php?r=estacoes/estacoesautomaticas>

<http://meteoropole.com.br/2015/03/qual-a-estacao-meteorologica-mais-antiga-do-mundo/>

<https://www.monolitonimbus.com.br/topicos-de-historia-da-meteorologia/>

<http://www.klementinum.com/prohlidky/astronomical-tower/>

<https://www.bemparana.com.br/noticia/meteorologia-sua-origem-e-evolucao-no-tempo->

<https://www.filipeflop.com/blog/estacao-meteorologica-com-arduino/>

<https://www.embarcados.com.br/estacao-meteorologica-com-arduino/>

<http://blog.mundoclima.com.br/como-funciona-e-quais-as-vantagens-da-estacao-meteorologica/>

16/05/2018

<https://www.filipeflop.com/blog/o-que-e-arduino/>

<https://www.devmedia.com.br/historia-do-c-c/24029>

<https://www.devmedia.com.br/java-historia-e-principais-conceitos/25178>

<https://www.infoescola.com/informatica/html/>

<http://mundoeducacao.bol.uol.com.br/informatica/html.htm>

<https://secure.php.net/manual/pt_BR/history.php.php>

17/05/2018

http://forum.fazedores.com/t/clones-e-outras-alternativas-ao-arduino/239/2

<https://www.embarcados.com.br/arduino-uno/>

Arduino The Documentary (2010)

http://www.globos.com.br/lib/site/o-que-e-pressao-atmosferica/

<https://store.arduino.cc/usa/arduino-uno-rev3>

<https://www.filipeflop.com/blog/temperatura-pressao-bmp180-arduino/>

<https://portal.vidadesilicio.com.br/dht11-dht22-sensor-de-umidade-e-temperatura/>

<https://www.filipeflop.com/blog/monitorando-temperatura-e-umidade-com-o-sensor-dht11/>

<https://www.arduino.cc/en/Main/Software>

18/05/2018

https://www.filipeflop.com/blog/sensor-de-chuva-yl-83/

https://www.embarcados.com.br/estacao-meteorologica-com-arduino/

https://www.devmedia.com.br/classe-inputstream-e-outputstream-em-java/32007

https://www.devmedia.com.br/utilizando-a-api-rxtx-para-manipulacao-da-serial-parte-i/6722

https://pt.stackoverflow.com/questions/81800/rxtx-conex%C3%A3o-em-porta-serial-de-balan%C3%A7a

https://www.devmedia.com.br/trabalhando-com-threads-em-java/28780

<https://www.devmedia.com.br/utilizando-a-api-rxtx-para-manipulacao-da-serial-parte-iii/7171>

http://www.java67.com/2016/10/3-ways-to-convert-string-to-json-object-in-java.html

https://pt.stackoverflow.com/questions/66563/para-que-serve-fun%C3%A7%C3%A3o-super

https://pt.stackoverflow.com/questions/194517/como-armazenar-os-dados-vindo-do-arduino-numa-string-java

https://www.devmedia.com.br/introducao-a-serializacao-de-objetos/3050

https://pt.stackoverflow.com/questions/70230/como-enviar-dados-via-post-e-recuperar-em-um-php

http://www.guj.com.br/t/como-fazer-extends-de-duas-classes/100116

http://www.guj.com.br/t/setar-label/346830/5

http://www.guj.com.br/t/enviar-system-out-println-de-uma-classe-para-um-jlabel/68861

https://www.java-forums.org/awt-swing/19683-system-out-println-jtextarea.html

https://tips4java.wordpress.com/2008/11/08/message-console/

https://www.devmedia.com.br/trabalhando-com-json-em-java-o-pacote-org-json/25480