

# Sistema de monitoramento de umidade de solo via web

Humberto A. F. Pereira, Rodolfo F. Oliveira

Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas  
Instituto Federal de São Paulo – Campus Hortolândia (IFSP)

humberto.ferris@gmail.com, rodolfo.oliveira@ifsp.edu.br

**Abstract.** *Recent research shows that the population is growing fast, with 9 billion people living in 2050. This will cause a growth in the urban population, making cities smarter. The Internet of Things aims to connect physical things to the Internet and, from this, generate insights that lead to decision making. The purpose of this paper is to develop a prototype that will measure soil moisture. For this, a web application will be developed to manage the sensors, while principles of the Internet of Things will be used in all the processes of the proposal, starting from obtaining the data of the physical phenomenon until the display of this data on the screen, through a web application. The final system will consist of a soil moisture sensor connected to a NodeMCU, a web system for control that will contain the standards HTML 5, CSS 3, will be used the framework AJAX, PHP language that will make the communication between the application, the data server and the communication of the sensors with the application.*

**Resumo.** *Pesquisas recentes mostram que a população está crescendo rapidamente, podendo passar os 9 bilhões de pessoas vivendo na terra em 2050. Isto gerará um aumento na população urbana, obrigando que as cidades fiquem mais inteligentes. A Internet das Coisas tem como objetivo conectar coisas físicas a Internet e, a partir disso, gerar insights que levem à tomada de decisão. A proposta deste artigo é desenvolver um protótipo que medirá a umidade do solo. Para isso, será desenvolvido um sistema web para gerenciar os sensores, enquanto princípios da Internet das Coisas serão utilizados em todos os processos da proposta, partindo desde a obtenção dos dados do fenômeno físico até a exibição desses dados na tela. O sistema final será composto por um sensor de umidade ligado a um NodeMCU, um sistema web para controle que conterá os padrões HTML 5, CSS 3, será utilizado o framework AJAX, linguagem PHP que fará a comunicação entre a aplicação, o servidor de dados e a comunicação dos sensores com a aplicação.*

## 1. Introdução

O crescimento populacional é um fenômeno que não pode ser ignorado. Segundo a ONU, a quantidade de pessoas no mundo deverá ultrapassar nove bilhões de pessoas em 2050 [ONU, 2013]. Por conta desse crescimento, é necessário a criação de novas tecnologias que repensem o modo de controlar recursos, sejam eles humanos, financeiros ou naturais.

O crescimento da população é um fato inevitável e a Tecnologia da Informação e Comunicação (TIC) é aliada principal para a inovação da gestão de recursos. Uma Cidade Inteligente pode ser definida como uma comunidade onde governo, negócios e população entendem o potencial da tecnologia da informação e tomam decisões conscientes de usá-

la em prol da melhoria do estilo de vida e trabalho de forma significativa na região [Lindskog 2004]. Em resumo, se transformar em uma Cidade Inteligente é usar a TIC para integrar os serviços prestados para aumentar a precisão de relatórios e fazer com que a manutenção dos mesmos seja feita de forma ágil e sem gastos desnecessários.

Para que esses dados sejam gerados, processados, analisados e apresentados, é necessário que equipamentos gerem essas informações. Para isso, o conceito de Internet das Coisas (do inglês *Internet of Things* - IoT) deve ser amplamente adotado pelos governantes. A IoT possibilita que objetos do ambiente tornem-se participantes ativos, ou seja, eles compartilham informações com outros membros da rede, ou com algum gerenciador e são capazes de reconhecer eventos e mudanças ao seu redor, atuando e reagindo de forma autônoma e apropriada a cada situação [Lerner 2012]. A conexão entre as coisas deve ser realizada de forma independente e que garanta a escalabilidade da rede, por isso as tecnologias de Redes de Sensores Sem Fio (RSSF) são essenciais para a viabilidade do projeto. Uma RSSF pode ser definida como uma categoria de rede móvel ad hoc (MANET - *Mobile Ad hoc Network*), já que a comunicação de todos os elementos é feita através de tecnologias que dispensam o uso de fios [Loureiro et al. [S.d.]]. Uma RSSF funciona de forma idêntica à MANET, pois os nós se comunicam entre si através de enlaces de comunicação sem fio. A principal diferença é que a MANET é usada para sustentar a comunicação entre os nós, já uma RSSF é utilizada para o tráfego de dados.

Diante deste cenário, é explícito que a melhora da gestão de recursos naturais demanda o desenvolvimento de novas tecnologias. A agricultura é o setor econômico que mais utiliza recursos naturais, principalmente hídricos. Para exemplificar, a produção de soja, em 2005, utilizou em torno de 50 bilhões de litros cúbicos [Carmo et al. 2007]. Com a utilização de uma rede de sensores de umidade do solo é possível identificar o nível de necessidade de cada tipo de cultura, possibilitando, então, o mapeamento de calor e entendimento da necessidade de irrigação em regiões específicas. Todos os dados gerados por essa rede podem ser utilizados para a criação de processos que visem a diminuição de utilização de recursos hídricos.

O presente trabalho propõe um sistema de monitoramento de umidade de solo baseado em uma rede de sensores que se comunicam através da Internet. Nesta proposta será utilizado um sensor de umidade de solo conectado a um módulo capaz de realizar a comunicação. Esse módulo envia os dados obtidos pelo sensor via HTTP para o servidor de aplicação. O conjunto de todos esses componentes é denominado Nó Sensor. Seguindo o tráfego da informação, o Servidor é o responsável por definir o intervalo de tempo entre uma coleta e a próxima, armazenar os dados coletados, interpretá-los a fim de definir se é necessário tomar uma ação ou não e hospedar e servir como servidor de acesso para a aplicação web.

O restante do trabalho é organizado como se descreve a seguir. Na Seção 2 será apresentado o conceito de Internet das Coisas. Na Seção 3, serão apresentados alguns trabalhos semelhantes e suas linhas de pesquisa. Na Seção 4, a proposta será detalhada e, a seguir, na Seção 5, será apresentado o processo de implementação do protótipo e, em seguida, na Seção 6, os testes realizados. Por fim, na Seção 7 será apresentada a conclusão e propostas de trabalhos futuros para o protótipo.

## **2. Internet das Coisas (IoT)**

Inicialmente, o conceito de Internet das Coisas não era definido com clareza. Antigamente, a Internet era uma rede de redes de computadores interligados ao redor do mundo formada, basicamente, pela conexão de servidores e computadores de mesa,

conhecidos como *desktops*. Hoje, além destes elementos, a Internet também é composta por elementos como sensores, lâmpadas, alarmes, geladeiras, entre outros. As pesquisas dessa área buscam resolver problemas de conectividade, ou seja, fazer com que cada coisa possa se comunicar com todas as outras. Mark Hung, autor da pesquisa “*Leading the IoT*”, pelo instituto Gartner, define IoT como “uma rede dedicada a objetos (coisas) que contém tecnologia suficiente para se comunicar e interagir com o ambiente externo”. Segundo o instituto de pesquisas Gartner, a previsão é de que em 2020 o número de coisas conectadas seja de 4 coisas a cada 1 pessoa [Mark Hung 2017].

Um exemplo de aplicação da IoT é a empresa *Nest Learning Thermostat*, adquirida pela Google em abril de 2014. Ela criou um termostato que aprende com as escolhas dos usuários. Através de um aplicativo mobile, o usuário controla a temperatura da casa e essas escolhas são armazenadas e analisadas pelo programa afim de descobrir qual a temperatura ideal para o usuário em determinado momento do dia de determinada estação [Tilley [S.d.]]. Outra aplicação dos conceitos de IoT apresentados é o conceito do mercado inteligente, aplicado pela Amazon, no serviço chamado “Amazon Go” onde todos os produtos utilizam *tags* inteligentes (RFID). Scanners realizam leituras afim de identificar qual produto saiu da prateleira, e isso é feito de forma automática, afim de eliminar filas nos caixas, uma vez que, com o cartão cadastrado, o débito da compra será feito de forma automática assim que o cliente sair do estabelecimento [“Amazon Go” 2018].

### **3. Trabalhos correlatos**

A seguir, serão apresentados trabalhos feitos no Brasil e em outros países do mundo, cujo objetivo é integrar a tecnologia de IoT com a agricultura. Em um trabalho publicado no *International Journal of Management, Technology and Engineering*, os autores Radrunath Karmakar e Rajeshkar C. Biradar desenvolvem uma aplicação que detecta a umidade do solo e, de forma automatizada inicia se necessário, um sistema de irrigação. A proposta utiliza-se de uma tecnologia chamado de SMPSInSYS, que, de acordo com a definição do artigo, é a combinação de duas tecnologias, onde parte dos componentes é relacionada a IoT e a outra é embutida. Os componentes embutidos são conectados ao Arduino Uno, enquanto o componente de IoT é conectado a um modulo ESP8266 [Karmakar and Biradar 2018].

Um segundo trabalho correlato é o de Pernapati (2018), intitulado de “Sistema de irrigação inteligente de baixo custo baseado em IoT”. Os autores construíram uma aplicação que automatiza o processo de irrigação através da implementação de um NodeMCU como micro-controlador responsável pela leitura da umidade e controle da bomba d’água para irrigação. A comunicação é feita através do protocolo MQTT e os dados são disponibilizados através de uma aplicação web e móvel. Eles concluem o artigo afirmando que o processo de irrigação atual é feito de forma manual e as vezes exige que o fazendeiro tenha que iniciar ou parar o processo de irrigação, mesmo durante a madrugada. A implementação desse tipo de aplicação para as fazendas aumentaria o nível de conforto e de controle sobre a quantidade de água que cada cultura necessita, além do baixo custo de implementação, uma vez que todos os *hardwares* são *open project* e extremamente adaptáveis [Pernapati 2018].

A utilização de sistemas web está cada vez mais comum, pois sua facilidade de visualização, além da capacidade de ajustes de tela e acessibilidade, melhora a interação do usuário com o sistema. Baseado nisso, Dyego Cantú escreveu um artigo onde é descrito o processo de construção de uma aplicação web para a visualização de dados obtidos através de sensores de temperatura e umidade. Construído utilizando o

Framework Django e a linguagem de programação Python, o autor justifica sua escolha afirmando que os microcontroladores sob a forma de sensores e atuadores podem se relacionar de maneiras distintas com os objetos do ambiente. Afirma também que a interface web pode ajudar na visualização dos dados e facilitação do acesso, já que, para acessar uma aplicação web, são necessários, apenas, conexão com a internet e um navegador [CANTÚ 2013].

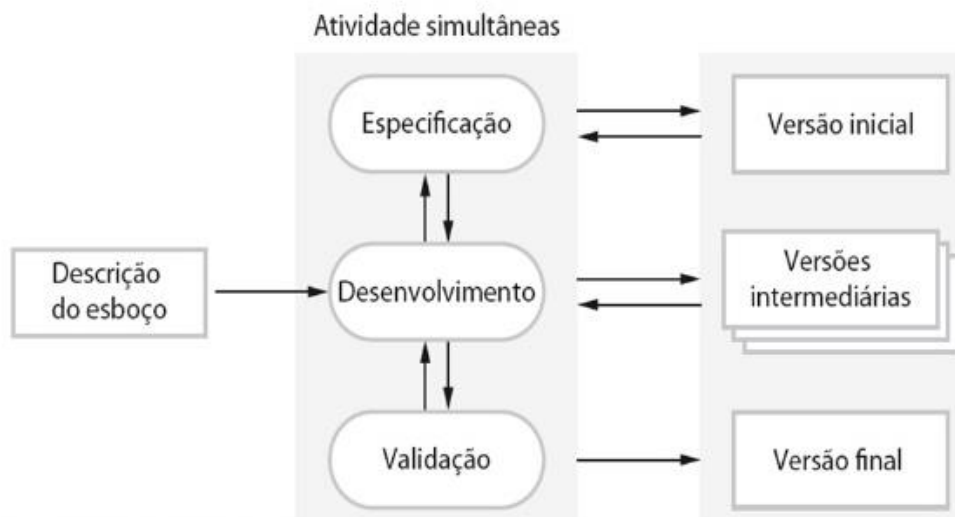
O protótipo proposto neste trabalho utiliza-se de um microcontrolador e um sensor de umidade de solo, o que contribui para a sua simplicidade de implementação, além dos componentes de *softwares*. Tornando, assim, sua arquitetura simplificada de baixo custo, pois une os benefícios dos *softwares* e *hardwares* livres e gratuitos. Por fim, a escolha de exibir os dados através de um *browser* foi para simplificar a interação do usuário com o sistema.

#### 4. Protótipo

Nesta seção será apresentado a proposta deste trabalho bem como a arquitetura e os diagramas utilizados na análise de requisitos, além da metodologia usada no desenvolvimento.

##### 4.1 Metodologia Utilizada

O desenvolvimento incremental será utilizado na parte do desenvolvimento do sistema, onde cada funcionalidade será desenvolvida e disponibilizada afim de garantir que a entrega seja feita de forma gradual e contínua, evitando que a disponibilidade do sistema seja comprometida por causa do cronograma. O modelo incremental combina elementos de fluxo de processos lineares e paralelos.[Pressman, 2011]. A Figura 1 apresenta as atividades do modelo incremental.



**Figura 1: Esquema das Atividades do Modelo Incremental. Fonte: Sommerville, 2011**

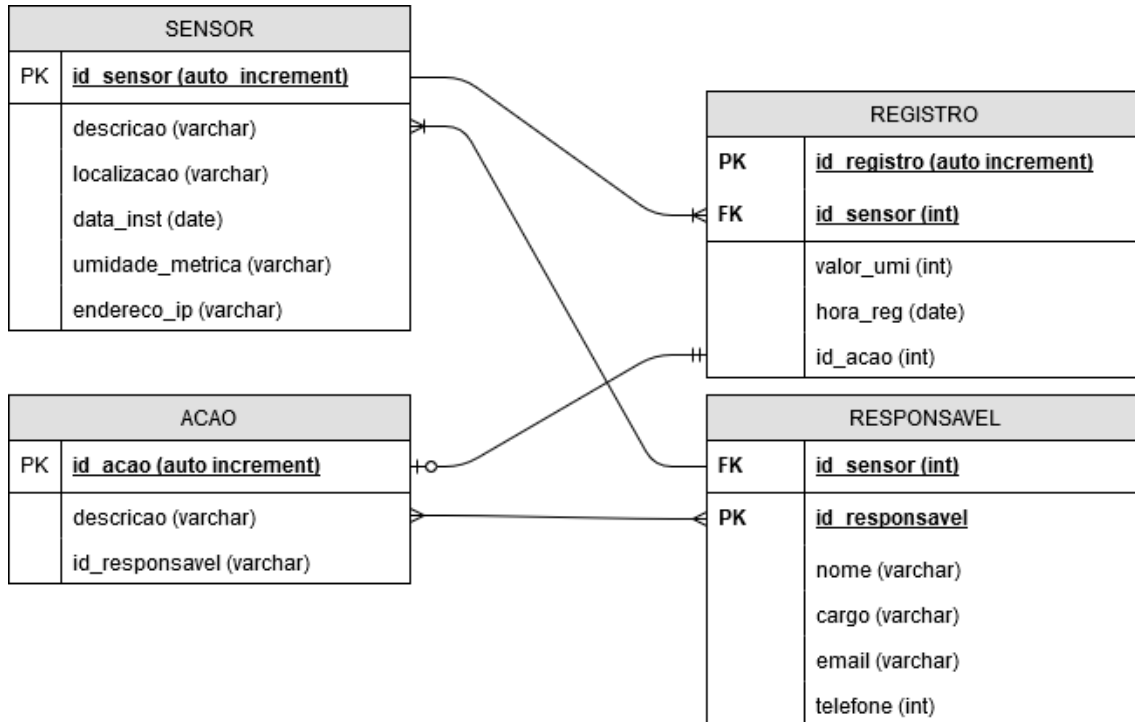
Seguindo os passos do modelo incremental, primeiramente temos a análise e a especificação dos requisitos. Os requisitos do sistema proposto são: o sistema deve ser acessado e gerenciado através da web; adicionar e remover sensores; informações coletadas devem ser exibidas em forma de gráficos; manter histórico das informações

coletadas; envio de alertas via e-mail, quando o valor da umidade for abaixo do valor mínimo

## 4.2 Análise de requisitos

A partir desse tópico, serão apresentados os diagramas utilizados para a elaboração da proposta e construção da aplicação.

### 4.2.1 Diagrama Entidade Relacionamento (DER)



**Figura 2: Representação gráfica do Diagrama Entidade Relacionamento (DER). Fonte: elaboração própria**

A técnica de modelagem de dados mais difundida e utilizada é a abordagem Entidade-Relacionamento (ER). Nesta técnica, o modelo de dados é representado através de um *Modelo Entidade-Relacionamento (Modelo ER)* [Heuser 2008]. Carlos Alberto Heuser define entidade como “conjunto de objetos da realidade modelada sobre os quais deseje-se manter informações no banco de dados”. O Banco de Dados é composto por 4 entidades: *Sensor*, *Registro*, *Acao* e *Responsavel*. A Figura 2 representa as entidades e seus relacionamentos da proposta em questão.

Na entidade *Sensor* são armazenadas as informações relacionadas à localização e descrição do sensor. Destaca-se o campo “*umidade\_metrica*” que é do tipo *varchar*. Esse valor é o responsável por determinar se o solo está seco ou não, ou seja, esse valor é uma referência para o sistema na hora de definir se o solo está seco ou úmido.

Já na entidade *Registro* são armazenadas as leituras que cada sensor enviou para o servidor. Dessa forma, é possível elaborar um histórico e, com base nessas informações, tomar alguma ação ou até mesmo prevê-las. É importante destacar que nessa entidade, o valor coletado, a hora e a tomada de decisão do sistema são registradas.

Na entidade *Acao* temos todas as ações que o sistema pode realizar de acordo com cada valor recebido e analisado. A princípio, a única ação armazenada é a de enviar e-mail. No entanto, é possível implementar ações mais complexas no futuro, como por

exemplo um controle de um sistema de irrigação autônomo. Nela é possível ver dois campos, um responsável pela descrição da ação cadastrada e o campo “id\_responsável”, utilizado para informar quem foi a pessoa que recebeu o alerta enviado através do e-mail.

Por fim, na entidade *Responsavel*, são armazenadas informações dos usuários cadastrados no sistema. Destaca-se o campo *email*, pois é através do valor cadastrado nesse campo que o sistema enviará o *email* informando sobre a situação atual do solo onde está localizado.

Os relacionamentos entre as entidades também são observados no modelo. A entidade sensor tem um relacionamento de 1 para N com a entidade registro e de N para 1 com a entidade *responsavel*.

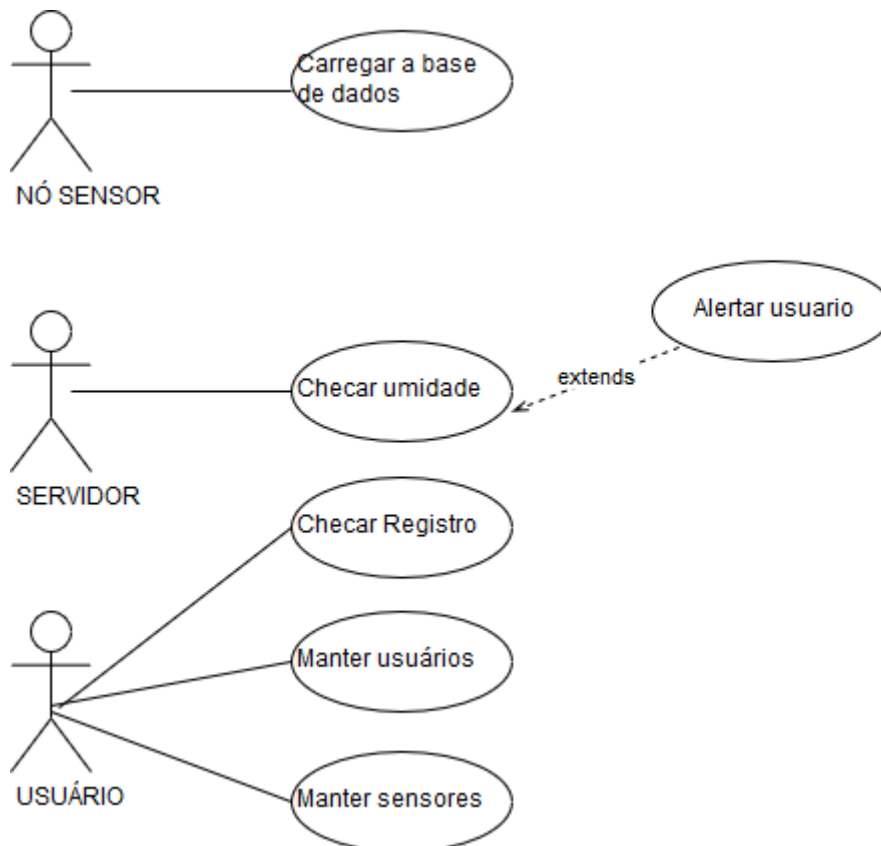
A entidade registro também se relaciona com a entidade *acao*, onde o relacionamento é de 1 para 1 opcional, isso significa que 1 registro pode ter, ou não, uma ação.

O último relacionamento observável é entre a entidade *acao* com a entidade *responsavel*, onde temos um relacionamento de vários para vários, ou seja, um responsável pode ter várias ações e vice-versa.

#### 4.2.2 Diagrama de Caso de Uso

Segundo Roger Pressman, “um caso de uso captura as interações que ocorrem entre produtores e consumidores de informação e o sistema em si” [Pressman 2011].

Na Figura 3 é representado o diagrama de caso de uso e a descrição textual do diagrama é apresentada logo em seguida.



**Figura 3: Diagrama de Caso de Uso da proposta. Fonte: elaboração própria**

As Tabelas 1, 2, 3, 4 e 5 apresentam os fluxos principais e alternativos de cada atividade apresentada no diagrama de caso de uso da Figura 3.

**Tabela 1: Caso de uso – Manter Sensores**

|                 |   |
|-----------------|---|
| Caso de Uso     | Manter sensores   |
| Descrição       | O usuário poderá adicionar, remover, editar sensores e ler os dados gerados.  |
| Pré-condição    | O usuário deve ter acesso a Interface no Servidor Web   |
| Fluxo principal | <ol style="list-style-type: none"> <li>1. Adicionar sensor: clicar em “Adicionar Sensor”; Preencher os campos com as informações solicitadas; Concluir a ação clicando em “Concluir”.</li> <li>2. Listar sensores: clicar em “Listar Sensores” e definir a ordenação de apresentação, podendo escolher “Por Nome” ou “Data de inserção”</li> <li>3. Editar sensor: após listar os sensores, clicar no ícone “Editar” para alterar as informações do sensor.</li> <li>4. Remover sensor: após listar os usuários, clicar no ícone “Excluir” o sensor.</li> </ol> |

**Tabela 2: Caso de Uso – Manter usuários**

|                 |  |
|-----------------|--|
| Caso de Uso     | Manter usuários  |
| Descrição       | O usuário pode adicionar, remover, editar e listar todos os usuários cadastrados.  |
| Pré-condição    | O usuário deve ter acesso a Interface no Servidor Web  |
| Fluxo principal | <ol style="list-style-type: none"> <li>1. Adicionar usuário: clicar em “Adicionar Usuário”; Preencher os campos com as informações solicitadas; definir nível de permissão do usuário; Concluir a ação clicando em “Concluir”.</li> <li>2. Listar usuário: clicar em “Listar Usuário” e definir a ordenação de apresentação, podendo escolher “Por Nome” ou “Data de inserção”</li> <li>3. Editar usuário: após listar os usuários, clicar no ícone “Editar” para alterar as informações do usuário.</li> <li>4. Remover usuário: após listar os usuários, clicar no ícone “Excluir” o usuário.</li> </ol> |

**Tabela 3: Caso de Uso – Checar registros**

|                 |   |
|-----------------|---|
| Caso de Uso     | Checar registros  |
| Descrição       | O usuário pode verificar os valores obtidos pelo sensor e qual a situação da umidade do solo  |
| Pré-condição    | O usuário deve ter acesso a Interface no Servidor Web   |
| Fluxo principal | <ol style="list-style-type: none"> <li>1. Clicar no botão “Consultar” presente no menu lateral.</li> <li>2. Clicar em Registros</li> <li>3. O sistema deve mostrar os registros obtidos e o fundo da linha deve estar sinalizado de acordo com o</li> </ol> |

|  |   |
|--|---|
|  | estado atual da umidade do solo. Sendo da cor vermelha quando abaixo ou igual a umidade de controle e da cor verde quando a cima da umidade de controle |
|--|---|

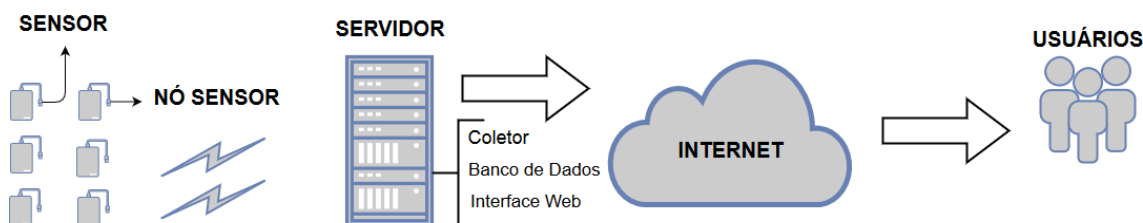
**Tabela 4: Caso de Uso – Carregar base de dados**

|                 |  |
|-----------------|--|
| Caso de Uso     | Carregar base de dados   |
| Descrição       | O sistema fará uma análise a respeito das marcações realizadas pelos sensores.   |
| Pré-condição    | Não há   |
| Fluxo Principal | O sistema irá analisar os dados gerados pelos sensores comparando o nível de umidade dos solos cadastrados e comparando com a umidade de controle determinado para aquele tipo de solo, caso a média da medição seja superior à recomendada, o sistema considerará que há um vazamento, caso contrário, indicará que o nível é considerado normal. |

**Tabela 5: Caso de uso – Checar umidade**

|                   |   |
|-------------------|---|
| Caso de Uso       | Checar umidade  |
| Descrição         | O sistema recebe o valor da umidade atual e compara com o valor mínimo definido pelo usuário                                      |
| Pré-condição      | Não há.   |
| Fluxo Principal   | O sistema recebe o valor da umidade e compara com o valor de controle pré-definido  |
| Fluxo alternativo | O sistema envia um e-mail alertando o(s) usuário(os) a respeito dos sensores que tiveram a medição menor que o valor de controle. |

### 4.3 Arquitetura do Protótipo



**Figura 4: Arquitetura do protótipo proposto. Fonte: elaboração própria**

O sistema tem como objetivo otimizar a gestão dos recursos hídricos utilizados em fazendas, estufas, plantações ou até mesmo em hortas residenciais, através dos conceitos de IoT. Para isso, é necessário a integração de *hardware* específicos e um *software* capaz de coletar, analisar, interpretar e tomar uma decisão. O sensor utilizado é um sensor de



umidade de solo, que envia o valor atual da leitura a cada 30 minutos. Esse valor é coletado, armazenado, interpretado e, baseando-se no valor de controle definido previamente, é possível alertar ou não o responsável por e-mail.

A Figura 4 é uma representação da arquitetura do sistema proposto. O primeiro elemento a ser destacado é o Sensor, responsável pela coleta de dados, ou seja, responsável por realizar a leitura do valor da umidade do solo e enviar para o Nó Sensor. Por sua vez, o Nó Sensor é responsável pela intermediação da comunicação entre o Servidor e o Sensor. O Nó Sensor recebe o valor obtido pelo Sensor e envia-o para o Servidor.

O Servidor é responsável por controlar a aplicação. Ele constitui-se de três principais elementos: Coletor, Banco de Dados e Interface Web. O Coletor é responsável por coletar a informação enviada pelo Nó Sensor. Assim que essa informação é recebida, ela é analisada e um registro é criado e salvo no Banco de Dados. Além dos dados coletados pelo Sensor, o Banco de Dados também armazena informações de usuários, valores chaves para a comparação dos dados, além do registro de cada sensor. Os Dados são disponibilizados em forma de gráficos através de uma Interface Web, onde o usuário pode ter acesso às informações de forma gráfica através de um navegador.

Caso algum ponto a ser monitorado pelo Nó Sensor apresente uma umidade abaixo de um determinado liminar, o sistema enviará, de forma automatizada, um e-mail alertando o responsável para que este possa tomar providências.

## 5. Implementação do Protótipo

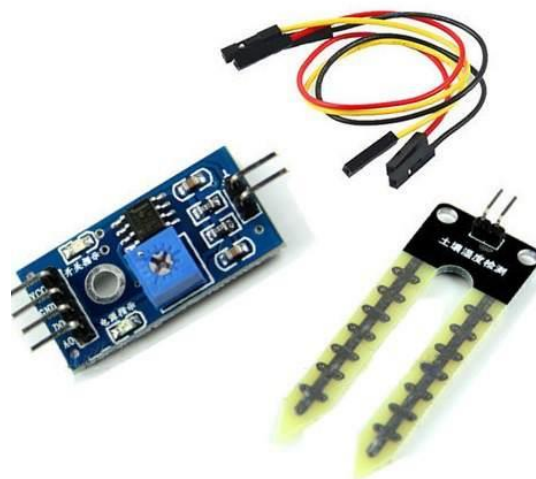
Os tópicos a seguir descrevem em detalhes as tecnologias e equipamentos utilizados para o desenvolvimento e integração dos componentes físico (*hardware*) e os componentes lógicos (*software*) da proposta em questão.

### 5.1 Implementação dos Nós Sensores

Primeiramente, para que a observação de um fenômeno natural seja possível, é preciso de *hardware* construído especificamente para isso. Para a implementação do Nó Sensor, foi utilizado a plataforma NodeMCU. A implementação do Sensor ficou a cargo do sensor de umidade do solo FC-28. Os componentes são representados pelas Figuras 5 e 6, respectivamente.



Figura 5: NodeMCU. Fonte: <https://www.nodemcu.com>



**Figura 6: Sensor de umidade do solo FC-28. Fonte [http://www.tato.ind.br/product\\_images/f/853/Umidade\\_\\_66906\\_zoom.jpg](http://www.tato.ind.br/product_images/f/853/Umidade__66906_zoom.jpg)**

O NodeMCU é uma plataforma aberta de Internet das Coisas construída na base ESP8266. Ele é amplamente utilizado em projetos relacionados à IoT, pois já vem equipado com uma antena Wi-Fi, o que facilita a comunicação e dispensa o uso de fios [Kodali and Sahu 2016]. Seu *firmware* foi desenvolvido baseado na linguagem de programação LUA e está disponível para download no GitHub do projeto, outro fator muito importante para a popularização deste módulo nos projetos de IoT [“nodemcu-firmware” [S.d.]]. Outro fator favorável é seu custo-benefício, já que seu preço é menor que US\$2,00 e dispensa o uso de componentes adicionais como Arduino ou Raspberry PI [“NodeMCU” [S.d.]]. O desenvolvimento dos códigos foi feito utilizando a IDE de desenvolvimento do Arduino, também de código aberto.

A importância e usabilidade do NodeMCU é apresentado na pesquisa publicada com o título de “Prototipação com nodeMCU para Internet das Coisas em Smart Cities: Uma Pesquisa”. Segundo o artigo, a característica de ser autossuficiente na gestão da conexão com os sensores, atuadores e transferência de dados através de websockets faz com que ele seja extremamente importante na avaliação de implantação de projetos de IoT [Veloso et al. 2017].

O sensor FC-28 foi escolhido para compor esse protótipo devido ao seu baixo custo de aquisição, em torno de R\$15,00. Porém, conforme o andamento do projeto e dos testes, verificou-se uma grande limitação equipamento, além da imprecisão dos dados obtidos. Esse modelo obtém a medição de forma mais rápida, porém, pode haver variações do valor real para o valor obtido, a variar com a calibragem do mesmo.

O fato dos componentes de *hardwares* serem *open hardware* foi fundamental para a decisão de montar o protótipo baseado neles, pois permite que o projeto possa ser utilizado como base para trabalhos futuros.

## 5.2 Implementação do Servidor

A implementação do servidor foi feita uma máquina virtual GNU/Linux Ubuntu *Desktop*. Nela está hospedada a aplicação web, o servidor Web Apache e o banco de dados, cujo Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado é o MariaDB. Os componentes escolhidos para montar a arquitetura de *software* são de projetos *open*

*source* e gratuitos, os quais permitem que futuros trabalhos possam ser desenvolvidos tendo como base este.

Como descrito anteriormente, o Apache HTTP Server é o responsável por possibilitar o acesso da aplicação através de um navegador web. Desenvolvido em 1995, tem como objetivo ser utilizado pelo maior número de pessoas possível [Apache Software Foundation, 2018]. Foi construído sob a licença de software livre, o que possibilita sua popularização.

Para o banco de dados, o MariaDB foi o Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado neste protótipo. O MariaDB é um dos SGBD's mais populares de mundo e foi desenvolvido pelos desenvolvedores do MySQL com a garantia de ser e continuar um software livre, sob a licença GNU GPL. Sua inclusão no pacote de desenvolvimento deu-se pelo fato de ser escalável, mais rápido, mais robusto, além de uma ampla gama de mecanismos de armazenamento maior que a antiga versão do MySQL [“MariaDB Foundation” 2018].

Para finalizar, foi utilizada a linguagem de programação PHP para escrever os *scripts* que rodam no *back-end* e no *front-end* da aplicação. Além de estar inserido no pacote de desenvolvimento, o PHP é muito robusto e de fácil entendimento. O PHP é uma linguagem de *scripting* popular de propósito genérico que se tornou parte do desenvolvimento web. Rápido, flexível e pragmático, o PHP é utilizado desde blogs até os sites das maiores empresas do mundo [PHP, 2018]. Ele também está presente no *front-end* da aplicação. A integração entre os comandos em PHP e a versatilidade de um *template bootstrap* deu à aplicação um design moderno e responsivo, permitindo a visualização em diversos tipos de telas.

## 6. Testes e resultados

Os testes demonstraram muitos desafios logo no início. Inicialmente, a primeira parte do teste foi obter os valores obtidos pelo sensor utilizando uma rede local sem fios. Primeiramente, é necessário calibrar o sensor para diminuir a variação de valores obtidos. Para isso, o NodeMCU foi programado para pegar os valores retornados pelo sensor e imprimi-los na console do Arduino IDE.

No primeiro teste sensor ficou submerso em água e, em seguida, ficou suspenso no ar, para indicar os valores de 100% e 0% de umidade, respectivamente. Com os valores brutos em mãos, foi feito um novo código onde o NodeMCU, agora, mapeava os valores obtidos e redistribuía-os em valores que variavam de 0 a 100, indicando a porcentagem da umidade. Nesse ponto, já foi possível ver que o sensor perdia a calibragem de forma relativamente fácil. Cada atualização do código que era feita no NodeMCU, percebia-se uma discrepância entre o valor da umidade mínima e máxima. Além disso, foi necessário conectá-lo à rede e receber o IP obtido.

Por fim, o NodeMCU coleta o valor da umidade e encapsula, junto com o IP obtido, em um pacote HTTP e envia, através do método POST para o servidor. Um exemplo de conteúdo enviado pode ser observado na Figura 7.

|  |          |
|--|----------|
| .....Endereço IP obtido: 192.168.0.104                         |          |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=3  | Linha 1  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=3  | Linha 2  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=3  | Linha 3  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=2  | Linha 4  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=3  | Linha 5  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=82 | Linha 6  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=84 | Linha 7  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=88 | Linha 8  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=89 | Linha 9  |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=82 | Linha 10 |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=83 | Linha 11 |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=84 | Linha 12 |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=85 | Linha 13 |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=3  | Linha 14 |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=3  | Linha 15 |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=1  | Linha 16 |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=0  | Linha 17 |
| Connected to the server<br>enderecoIP=192.168.0.104&umidade=0  | Linha 18 |

**Figura 7: Resultado da medição impressa na console da IDE.**  
**Fonte: elaboração própria.**

Na Figura 7, é possível observar que, nas 5 primeiras medições, os valores variaram de 2 a 3. Nesse momento, o sensor encontrava-se em um recipiente vazio. O valor obtido foi diferente do esperado (valor 0). No entanto, o mesmo foi considerado, já que era um valor próximo a 0. Nas linhas de 6 até 9, é possível observar um aumento repentino no valor da umidade. Durante essas medições, o sensor encontrava-se submerso em água. Esperava-se o valor de 100, porém, apesar da diferença de valores e da instabilidade do sensor, considera-se os valores obtidos como aceitáveis. Em seguida, as medições presentes entre as linhas 10 e 13, o sensor foi enterrado em um recipiente com terra bem úmida. Devido ao método de medição do sensor, acredita-se que seu funcionamento seja melhor quando em contato com a terra e, devido às condições do meio em que ele foi inserido, os valores obtidos foram considerados aceitáveis. Por fim, nas medições restantes, o sensor foi removido do recipiente com terra úmida e,

novamente, adicionado ao recipiente vazio, o que explica a queda repentina na medição do sensor. A Figura 8 mostra como foi montado o ambiente de teste.



**Figura 8: ambiente utilizado para teste. Fonte: elaboração própria.**

O próximo passo foi fazer com que os dados enviados pelo NodeMCU fossem salvos no Banco de Dados do Servidor. Para isso, foi escrito um código em PHP que recebe os valores através do POST, insere-os em uma consulta SQL, junto com a hora em que o pacote foi recebido, para então, inserir o novo registro no banco de dados.

O terceiro desafio foi criar o alerta via e-mail. A proposta inicial era realizar tal tarefa através de um *script* em PHP que executava um comando do serviço de *sendmail*, configurado no servidor. Porém as barreiras de segurança da Google e Yahoo (servidores de e-mail utilizados nos testes) impediam que, segundo definição da política de segurança do servidor, aplicativos menos seguros acessassem a conta. Diversas configurações foram feitas e alteradas, inclusive a liberação de acesso para aplicativos menos seguros, porém sem sucesso.

Outra alternativa escolhida, foi a utilização do serviço *MailGun*, onde o ambiente de envio de e-mails já está configurado de forma que, para fazer o envio, é necessária a utilização da API para o envio, também sem sucesso.

Para contornar esse problema e mostrar a reação do sistema perante uma situação onde a umidade cai, afim de validar o teste, ao exibir os registros na aplicação web, a linha onde a umidade é igual ou menor que a umidade definida como mínima, receberá a cor vermelha como *background*.

|    |    |               |                     |
|----|----|---------------|---------------------|
| 2  | 3  | 192.168.0.104 | 2018-11-27 12:11:12 |
| 3  | 3  | 192.168.0.104 | 2018-11-27 12:11:58 |
| 4  | 3  | 192.168.0.104 | 2018-11-27 12:11:58 |
| 5  | 3  | 192.168.0.104 | 2018-11-27 12:11:58 |
| 6  | 2  | 192.168.0.104 | 2018-11-27 12:11:58 |
| 7  | 3  | 192.168.0.104 | 2018-11-27 12:11:58 |
| 8  | 82 | 192.168.0.104 | 2018-11-27 12:11:58 |
| 9  | 84 | 192.168.0.104 | 2018-11-27 12:11:58 |
| 10 | 88 | 192.168.0.104 | 2018-11-27 12:11:58 |
| 11 | 89 | 192.168.0.104 | 2018-11-27 12:11:58 |
| 12 | 82 | 192.168.0.104 | 2018-11-27 12:11:58 |

**Figura 9: Visualização das medições através do navegador. Fonte: elaboração própria.**

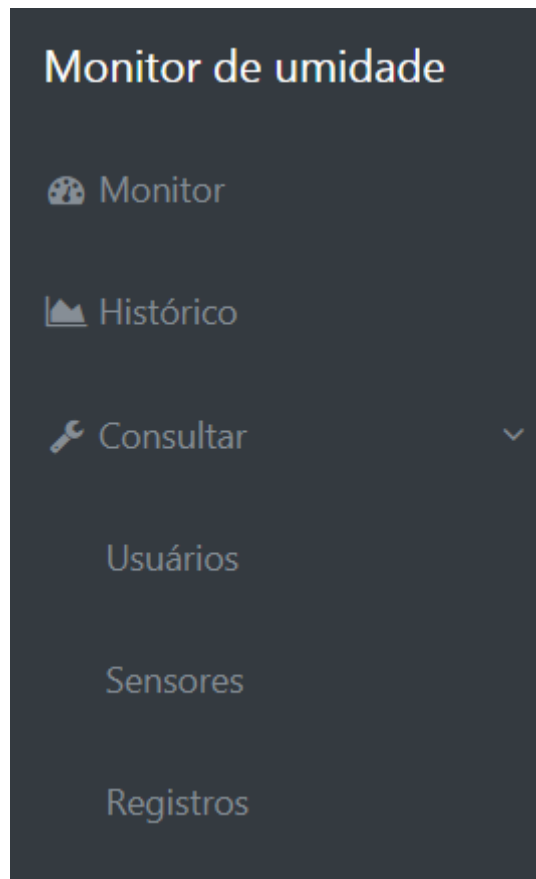
A Figura 9 mostra as informações coletadas diretamente do banco de dados e tratadas através da HTML. A Figura 9 representa a tabela gerada quando acessado no navegador. Para que a validação de dados fosse possível, durante a consulta foi utilizado uma função de *INNER JOIN* entre as tabelas Sensor e Registro. Para que a visualização seja bem sucedida, o valor do IP do sensor armazenado em ambas tabelas, sejam idênticos, caso contrário, o sistema não retornará os valores armazenados na tabela Registro.

Para a construção das páginas web, utilizou-se um template *bootstrap* como base, garantindo, assim, através das técnicas de responsividade, uma visualização agradável, independente da tela. A navegação também foi pensada para que seja algo simples e intuitivo. No topo da página, podemos observar o nome “Monitor de umidade”, que, ao mesmo tempo, é uma forma de voltar para o início da navegação. No lado esquerdo da tela apresenta-se um menu com as opções “Monitor”, “Histórico” e “Consultar”.

Clicando na opção “Monitor”, o usuário será redirecionado para a tela de início, composta pelo gráfico referente às últimas medições.

A opção “Histórico” carrega o gráfico referente às medições mais recentes.

Ao clicar na opção “Consultar” (Figura 10), três opções são oferecidas ao usuário, são elas: “Usuários”, “Sensor”, “Registros”. Ao clicar na opção “Usuários”, o usuário é redirecionado para uma página com uma tabela onde são apresentados os dados da tabela Responsavel. O mesmo acontece com as opções “Sensor” e “Registros”, cada um retornando uma tabela com os dados das tabelas Sensor e Registro, respectivamente.



**Figura 10: Tela da barra de navegação. Fonte: elaboração própria.**

É possível cadastrar um responsável ou um sensor novo, acessando o formulário de cadastro presente na visualização das respectivas consultas, porém, não é possível cadastrar um novo registro. A Figura 11 mostra o formulário de cadastro de sensor e a Figura 12 mostra o cadastro de responsável, tratado, no formulário, como usuário. Figuras retiradas do sistema, através de um *print screen*.

| Cadastrar sensor                                 |  |
|--|--|
| Descrição  | Localização  |
| <input type="text" value="Descrição do sensor"/> | <input type="text" value="Localização do sensor"/> |
| Data de instalação                               |  |
| <input type="text" value="dd/mm/aaaa"/>          |  |
| Umidade de controle                              | IP do sensor                                       |
| <input type="text" value="Umidade de controle"/> | <input type="text" value="Endereço IP do sensor"/> |
| <input type="button" value="Cadastrar"/>         |  |

**Figura 11: Tela do formulário de cadastro de sensor. Fonte: elaboração própria.**

Registrar novo usuário

Nome

Cargo

Email

Telefone

[Cadastrar](#)

[Login Page](#)

[Forgot Password?](#)

**Figura 12: Tela do formulário de cadastro de responsável. Fonte: elaboração própria.**

Os testes foram bem sucedidos porém a limitação do sensor é notável pois, mesmo com o sensor submerso em água, o valor retornado variava entre 82% e 89%. Quando suspenso, o sensor retornava entre 0% e 3%, como esperado. Tal erro poderia ser corrigido com uma nova calibração. No entanto, percebeu-se que as calibrações deveriam ser realizadas de 1 em 1 dia, inviabilizando a utilização de tal sensor.

Apesar das limitações técnicas, o *hardware* e o *software* funcionaram bem em para criar este primeiro protótipo.

## 7. Conclusão

Conclui-se que a Internet das Coisas, além de desafiadora, mostra-se como uma opção viável para o monitoramento de eventos físicos. A integração de serviços gratuitos e/ou livres com os *hardwares* de código aberto, que tem custo baixo, testes com uma arquitetura robusta são possíveis. Em uma arquitetura onde haja mais nós e usuários acessando, algumas mudanças devem ser feitas.

Primeiramente, observou-se que o protocolo HTTP é pesado e causa interrupção na comunicação, quando utilizado em conjunto com o NodeMCU.

Em seguida, sugere-se a troca do sensor, pois o utilizado no protótipo apresentou uma vida útil curta, além da imprecisão dos dados obtidos por ele. O sensor FC-28 utiliza um método de medição eletromagnética para obter o valor da umidade do solo. Esse tipo de medição é conhecido como “medição indireta”. Além disso apresenta uma vida útil curta, devido a corrosão sofrida pelas hastes [Típico 2017].



O terceiro ponto que deve ser observado, é adicionar uma interface de controle de acesso e a disponibilização dos dados restrita aos dados do usuário *logado* ao sistema, garantindo a disponibilidade e a confidencialidade das informações armazenadas. Além disso, o alerta através do e-mail também deve ser implementado como trabalhos futuros.

O protótipo exigiu que conhecimentos de diversas áreas fossem colocados em prática afim de entregar um protótipo funcional. Esses conhecimentos navegam desde a parte de engenharia de software, que foi a concepção do sistema e de suas interações e relacionamentos, passando por eletrônica, afim de montar os componentes físicos de forma com o fenômeno fosse observado com sucesso. Em seguida, conhecimentos sobre o suporte a servidores e suas configurações de acesso, como a instalação de um webserver capaz de hospedar uma aplicação web, que exigiu conhecimentos de desenvolvimento web para que se pudesse entender como manipular o template *bootstrap* e o transporte de dados de uma página para outra através de suas tecnologias.

A integração de conhecimentos obtidos durante o percurso com os apresentados durante o curso resultou num projeto que contém todas as fases, desde a concepção até a entrega de um produto final.

## 8. Referências Bibliográficas

- Amazon Go (2018). <https://amazon.com/go>, [accessed on Nov 28].
- Apache HTTP Server ([S.d.]). [https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html), [accessed on Nov 28].
- CANTÚ, D. (2013). Sistema Web Para Monitoramento De Sensores De Temperatura E Umidade. p. 87.
- Carlos Alberto Heuser (2008). *Projeto de Banco de Dados*. 6. ed.
- Carmo, L. R., Ojima, A. L. R. de O., Ojima, R. and Nascimento, T. T. (2007). Água virtual, escassez e gestão: o Brasil como grande exportador de água. *Ambiente e Sociedade*, v. 10, n. 2, p. 83–96.
- Cidades terão mais de 6 bilhões de habitantes em 2050, destaca novo relatório da ONU ([S.d.]). <https://nacoesunidas.org/cidades-terao-mais-de-6-bilhoes-de-habitantes-em-2050-destaca-novo-relatorio-da-onu/>, [accessed on Nov 28].
- Karmakar, R. and Biradar, R. C. (2018). Automated Soil Moisture Detection and Protective Sensing based Smart Irrigation System using IoT ( SMPSInSYS ). v. 8, n. 659, p. 659–663.
- Kodali, R. K. and Sahu, A. (2016). An IoT based soil moisture monitoring on Losant platform. *Proceedings of the 2016 2nd International Conference on Contemporary Computing and Informatics, IC3I 2016*, p. 764–768.
- Lerner, M. (2012). Internet of Things Edited by : Technical Editors :
- Lindskog, H. (2004). Smart communities initiatives. *proceedings of the 3rd ISOneWorld Conference*, n. April, p. 16.
- Loureiro, A. A. F., Nogueira, M. S., Nakamura, E. F. and Maur, C. ([S.d.]). Redes de Sensores Sem Fio. p. 179–226.
- MariaDB Foundation (2018). <https://mariadb.org/about/>.
- Mark Hung (2017). Leading the IoT.

nodemcu-firmware ([S.d.]). <https://github.com/nodemcu/nodemcu-firmware>, [accessed on Nov 25].

NodeMCU ([S.d.]). [http://www.nodemcu.com/index\\_en.html](http://www.nodemcu.com/index_en.html), [accessed on Nov 25].

Pernapati, K. (2018). IoT based low cost smart irrigation system. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, n. Icticct, p. 1312–1315.

PHP ([S.d.]). <http://www.php.net/>, [accessed on Nov 28].

Pressman, R. S. (2011). *Engenharia de software Uma Abordagem Profissional*. 7. ed.

Tilley, A. ([S.d.]). Google Acquires Smart Thermostat Maker Nest For \$3.2 Billion. <https://www.forbes.com/sites/aarontilley/2014/01/13/google-acquires-nest-for-3-2-billion/#759d9d266ee2>, [accessed on Nov 26].

Típico, V. D. (2017). COMPORTAMENTO DO SENSOR DE UMIDADE DO SOLO COM ARDUINO ® EM LATOSSOLO VERMELHO DISTROFÉRICO TÍPICO Jeison Junior Pinto 1 , Marcio Antonio Vilas Boas 2 , Vander Fabio Silveira 3. p. 1–4.

Veloso, A. F. S., Sousa, B. A., Braz, A. R., et al. (2017). Prototipação com nodeMCU para Internet das Coisas em Smart Cities: Uma pesquisa. *Livro Anais - Artigos e Minicursos*, p. 105–110.