

# **DESENVOLVIMENTO DE UM APLICATIVO MÓVEL, UTILIZANDO-SE DA PLACA NODEMCU ESP8266, PARA O MONITORAMENTO DO CONSUMO DE ÁGUA.**

**Igor Galindo Paiva**

Graduado em Bacharelado em Sistemas de Informação pela Faculdade Sete de Setembro – FASETE.  
igorgalindopaiva@gmail.com

**Denise Xavier Fortes**

Docente da Faculdade Sete de Setembro - Mestranda em Ciência da Computação. Programa de Pós-Graduação em Ciência da Computação (PROCC) Universidade Federal de Sergipe (UFS) – São Cristóvão, SE – Brasil. Especialista em Banco de Dados pela Universidade Gama Filho (2010). Graduada em Sistemas de Informação pela Faculdade Sete de Setembro – FASETE (2010) - Paulo Afonso – BA, Brasil. denise.fortes@fase.edu.br

**Ricardo Azevedo Porto**

Docente da Faculdade Sete de Setembro – FASETE. Mestre em Sistemas e Computação pela Universidade Salvador- UNIFACS. Pós-graduado em Sistemas de Informação com Ênfase em Banco de Dados pela Faculdade Ruy Barbosa. Graduado em Bacharelado em Ciência da Computação pela Universidade Tiradentes – UNIT. ric.porto@gmail.com

## **RESUMO**

Atualmente, o desperdício de água é um dos fatores mais preocupantes em se tratando das questões ambientais. Sendo assim, o presente estudo desenvolveu um protótipo que possibilita o monitoramento do consumo de água, em tempo real, através de um dispositivo móvel. Com isso, os consumidores podem gerenciar os gastos de água, bem como realizar o acompanhamento dos horários de picos deste consumo, afim de deixá-los mais atentos quanto ao que está sendo desperdiçado. Para a elaboração deste estudo, utilizou-se a pesquisa bibliográfica, como também, a prototipação. Além disso, utilizou-se a placa NodeMCU ESP8266 para a interação com sensores, atuadores e conexão com a internet via wireless, o Firebase Realtime Database para armazenamento dos dados e o Thunkable como ambiente de desenvolvimento Android.

**Palavras-chave:** Tecnologia da Informação, NodeMCU ESP8266, Sistemas Embarcados, Microcontroladores.

## **ABSTRACT**

Nowadays, the waste of water is one of the most worrying problems about environmental subjects. Thus, this paper developed a prototype that makes possible to monitor the water consumption in real time through a mobile device. With this, consumers may manage the use of water as well as keep track of the times when the consumption is more intense, in order to make them more careful about the waste. To carry out this study, it was conducted a bibliographical research and prototyping. Besides this, we made use of the NodeMCU ESP8266 platform for the interaction with the sensors, actuators

and wireless internet connection, the Realtime Database firebase for data storage and the Thunkable as development environment for Android.

**Keywords:** Information Technology. NodeMCU esp8266. Embedded systems. Microcontrollers

## 1 INTRODUÇÃO

Constantemente, os meios de comunicações exibem campanhas de conscientização a respeito do desperdício da água e a importância em economizá-la devido à falta de chuva e esvaziamentos dos reservatórios. De acordo com o relatório da ONU BRASIL (2015), o planeta enfrentará um déficit de água de 40% até o ano de 2030, desta forma, torna-se imprescindível o consumo deste recurso de forma consciente.

De acordo com Capriles (2004), o Brasil desperdiça 40% da água potável destinada ao consumo humano. Existem alguns fatores que contribuem de forma significativa para o desperdício, tais como: banhos demorados, torneiras mal fechadas, falta de saneamento básico e desperdício nos sistemas de distribuição.

Devido a esses fatores, existe uma preocupação com relação a falta de água potável para as gerações futuras. Sendo assim, economizar água, bem como o seu uso de forma consciente, tornou-se um fator primordial. Desta forma, o presente estudo visa apresentar conceitos, definições e resultados, após o desenvolvimento e a utilização de um protótipo capaz de realizar o monitoramento do consumo de água em tempo real.

## 2 SMART CITIES

O uso das Tecnologias de Informação e da Comunicação (TICs) para a resolução de problemas, bem como para tornar os serviços providos aos cidadãos mais eficientes, estão genericamente associadas às cidades inteligentes (*Smart Cities*). O termo *Smart* surgiu em 2010, na União Europeia, para representar projetos sustentáveis e ações no espaço urbano.

Batty et al. (2012, apud Silva, 2017), afirmam que não é somente pela automação de serviços rotineiros que atendem pessoas, edificações e sistemas de gestão, entre outros, que as cidades estão se tornando inteligentes, mas porque permitem monitorar, compreender,

analisar e planejar seu desenvolvimento, aumentando a eficiência, os investimentos e a qualidade de vida dos cidadãos em tempo real.

Dameri e Cocchia (2013), conforme citados por Silva (2017), afirmam que não é possível saber se os vocábulos inteligente e digital são utilizados com o mesmo significado ou se estes definem tecnologias, estratégias ou cidades distintas.

Para Cocchia (2014, apud Silva, 2017), as cidades digitais possuem ênfase na melhoria da qualidade dos serviços e das informações providas aos cidadãos por meio da TICs. Enquanto as cidades inteligentes envolvem diferentes aspectos da vida urbana. Já Batty et al. (2012, apud Silva, 2017), as cidades digitais possuem enfoque na infraestrutura, enquanto que as cidades inteligentes tratam especialmente da forma como os recursos estão sendo utilizados. Logo, de acordo com os autores acima, pode-se afirmar que as cidades digitais são um subconjunto das cidades inteligentes, destacando-se a utilização da tecnologia e inovação, dos requisitos ambientais e do desenvolvimento econômico.

### **3 SISTEMAS EMBARCADOS**

De acordo com Chase (2007), diferentemente dos computadores que executam sistemas operacionais e softwares para as mais variadas aplicações, os sistemas embarcados são construídos para executar uma tarefa pré-determinada. Sendo assim, na maioria dos projetos, não há flexibilidade de software ou hardware que lhes permita realizar outras tarefas que não sejam aquelas para os quais foram desenhados e desenvolvidos.

De acordo com a Embedded Architects (2009):

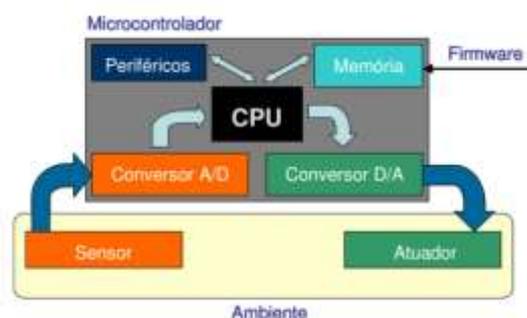
Sistemas embarcados são desenvolvidos para uma tarefa específica. Por questões como segurança e usabilidade, alguns inclusive possuem restrições para computação em tempo real. O software escrito para sistemas embarcados é muitas vezes chamado firmware, e armazenado em uma memória ROM ou memória flash ao invés de um disco rígido. Por vezes o sistema também é executado com recursos computacionais limitados: sem teclado, sem tela e com pouca memória. Todos estes fatores também podem ser traduzidos em custo reduzido.

Portanto, pode-se afirmar que são computadores com sistemas digitais micro controlados para aplicações específicas e, geralmente, são completamente encapsulados pelo dispositivo que controla. Além disso, os sistemas embarcados podem interagir com seu ambiente por meio de hardware e software. Em geral, a sua capacidade de processamento é reduzida quando comparada aos computadores desktops, pois utilizam microcontroladores.

A explicação acima corrobora com Delai (2013), “Sistemas embarcados estão relacionados ao uso de hardware (eletrônica) e software (instruções) incorporados em um dispositivo com um objetivo pré-definido. A diferença entre um sistema embarcado e um computador de propósito geral está justamente na objetividade”.

Os sistemas embarcados interagem continuamente com o ambiente externo por meio de sensores e atuadores. A Figura 1 abaixo representa um sistema embarcado simples, incluindo dispositivos de entrada (sensores), saída (atuadores) e sistema interno de processamento (microcontrolador).

**Figura 1** - Exemplo de um Sistema Embarcado



Fonte: Chase, 2007, p.2.

De acordo com a figura acima, tem-se um sistema embarcado básico, controlando uma variável ambiente (temperatura), umidade ou pH do ar de uma estufa. O microcontrolador gerencia todo o funcionamento do sistema, pois tem a capacidade de realizar a leitura dos sinais externos capturados pelo sensor, processar os sinais e enviar os resultados para os atuadores.

### 3.1 Sensores

Os sensores são capazes de converter as grandezas físicas em sinais elétricos correspondentes, que serão enviados para um microcontrolador, que por sua vez será responsável pelo tratamento dos dados recebidos. Segundo Ribeiro (2004), os sensores podem ser classificados como: sensores externos e sensores internos.

Os sensores externos são responsáveis por observar os aspectos do mundo exterior, como por exemplo: sensores de contato, de proximidade, de força, de distância, de laser, de ultrassom, de infravermelhos e sensores químicos. Enquanto que os sensores internos fornecem informação sobre os parâmetros internos do equipamento como, por exemplo, a velocidade ou sentido de rotação de um motor. São exemplos de sensores internos: potenciômetros e os sensores inerciais (acelerômetros, giroscópios, bússolas).

De acordo com os sinais recebidos pelos sensores, os microcontroladores são capazes de executar ações pré-definidas, como ligar e desligar algum equipamento, abrir ou fechar persianas ou, até mesmo, disparar o airbag de um carro. Como afirma Ogata (1998), o elemento controlador trata os sinais recebidos e, posteriormente, são amplificados e enviados aos atuadores do sistema.

### **3.2 Atuadores**

De acordo com Petry (2016), “O atuador assim como um sensor é um transdutor que transforma uma forma de energia em outra, porém este faz o caminho inverso, ele transforma um sinal elétrico em uma grandeza física, movimento, magnetismo, calor entre outros.”

Sendo assim, o atuador é um elemento capaz de produzir movimento, através de comandos manuais, elétricos ou mecânicos recebidos. Portanto, são dispositivos que realizam ações que interferem no processo em controle, como: ventiladores, luzes, aquecedores, motores, etc. Ainda de acordo com o autor, os atuadores podem ser classificados a depender do tipo de energia que utiliza, tais como: atuadores hidráulicos, pneumáticos e eletromagnéticos. Onde, os atuadores hidráulicos utilizam um fluido à pressão para movimentar o braço, sendo mais utilizados em robôs de grande porte. Já os pneumáticos utilizam um gás à pressão para movimentar o braço e são mais baratos que os hidráulicos, sendo usados em robôs de pequeno porte. Por fim, os eletromagnéticos, são os motores elétricos, tais como: motores de passo, servos motores, corrente contínua ou corrente alternada.

## 4 DESENVOLVIMENTO DO PROTÓTIPO

### 4.1 Plataforma Android

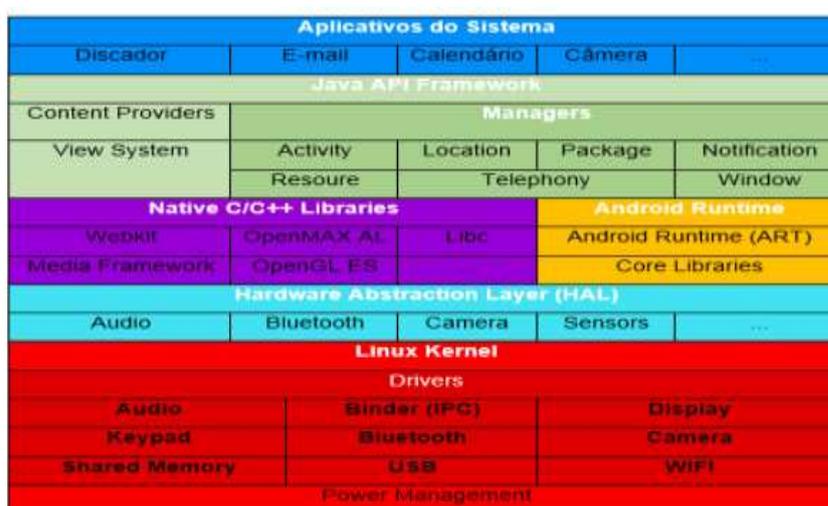
Android é uma plataforma open source voltada para dispositivos móveis. Segundo Cordeiro (2017), o Android possui um SDK que disponibiliza as ferramentas e APIs necessárias para o desenvolvimento na plataforma Android, com a linguagem JAVA.

De acordo com Lacheta (2010):

O Android é a nova plataforma de desenvolvimento para aplicativos móveis como smartphones e contém um sistema operacional baseado em Linux, uma interface visual rica, GPS, diversas aplicações já instaladas e ainda um ambiente de desenvolvimento bastante poderoso.

A arquitetura da plataforma Android é modular dividida em seis camadas, as quais são denominadas: Linux Kernel; Camada de Abstração de Hardware (HAL); Libraries; Android RunTime; Application Framework e Applications. A seguir, a figura 2 representa a arquitetura da plataforma Android:

**Figura 2 - Arquitetura da plataforma Android**



Fonte: Disponível em: <<https://developer.android.com/guide/platform/index.html?hl=pt-br#api-framework>>.

Acesso em: 15 jun. 2017.

De acordo com a figura 2, a base do Android é uma versão modificada do kernel Linux 2.6, que provê vários serviços essenciais, como: segurança, rede e gerenciamento de memória e processos. Além disso, possui uma camada de abstração ao *hardware* para as outras camadas de *software*. Acima do kernel ficam as bibliotecas C/C++ utilizadas por diversos

componentes do sistema. Já o Android Runtime é composto pela máquina virtual chamada Dalvik VM, onde as aplicações são executadas. Na camada acima, fica o *framework* de aplicações, que fornece todas as funcionalidades necessárias para a construção de aplicativos, através das bibliotecas nativas. No topo é a camada que está visível para os usuários comuns, os aplicativos.

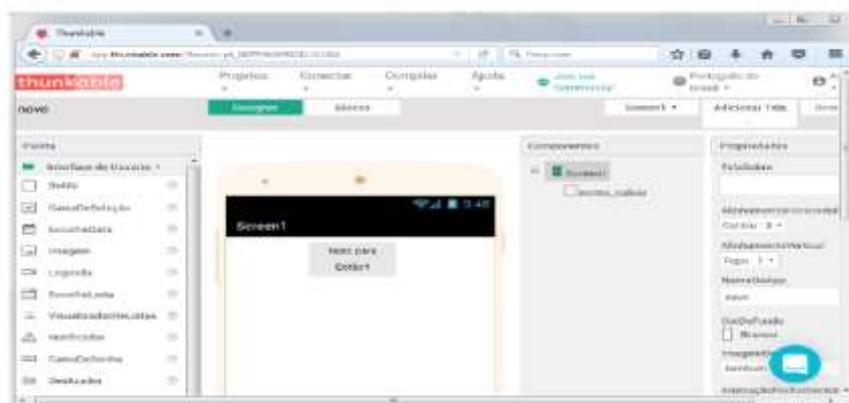
## 4.2 Firebase

O Firebase é um Backend as a Service (BaaS), ou seja, um serviço de computação na nuvem que serve como um mediador entre os softwares. Portanto, ele é responsável por conectar o back-end e o front-end de uma aplicação. De acordo com Batschinski (2016), o BaaS possibilita aos desenvolvedores abstraírem completamente a infraestrutura do lado do servidor, permitindo assim, aos desenvolvedores, se concentrarem na experiência do usuário (Front-end) em vez de lidar com infraestrutura e codificação do back-end.

## 4.3 Thunkable

O Thunkable é um ambiente de desenvolvimento para dispositivos Android que funciona online e de forma gratuita. Este ambiente utiliza o conceito de drag-and-drop (clique e soltar) para criar os aplicativos, não necessitando de conhecimento prévio sobre códigos e linguagens de programação. A seguir, a figura 3 representa a IDE do Thunkable.

**Figura 3** - Ambiente de desenvolvimento Android, *Thunkable*

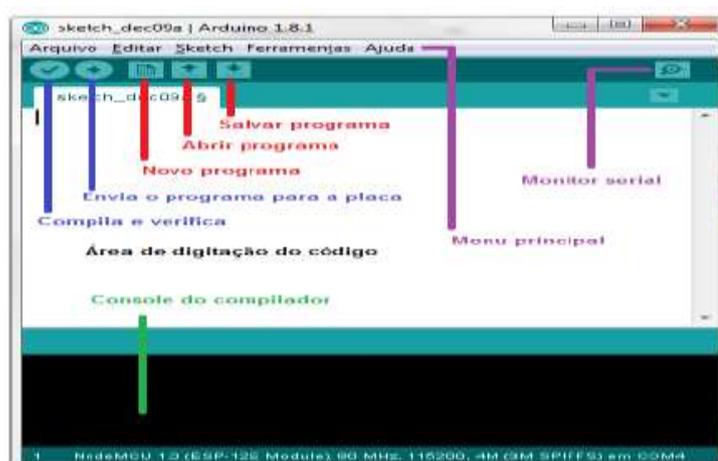


Fonte: Disponível em: < <https://thinkable.com/#/>>. Acesso em: 23 jun. 2017.

## 4.4 IDE do Arduino

A IDE do Arduino é um ambiente de desenvolvimento utilizado para programar e enviar as instruções para o Arduino e outras placas como, por exemplo, a NodeMCU ESP8266. De acordo com McRoberts (2011, p. 24), a IDE do Arduino é um *software* livre no qual é possível escrever códigos na linguagem que o Arduino compreende (baseada na linguagem C). Além disso, possui uma interface simples e de fácil entendimento.

**Figura 4** - Ambiente de desenvolvimento (IDE) para Arduino.



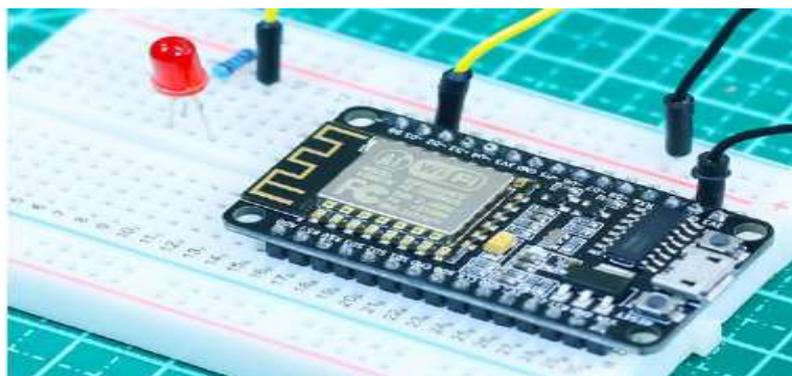
Fonte: Autores, 2017.

#### 4.5 NodeMCU Esp8266

A NodeMCU é uma placa de desenvolvimento, fácil de programar e de baixo custo, construída sobre o SDK ESP8266. Para Thomsen (2016), o módulo Wifi ESP8266 NodeMCU ESP-12E é uma das placas mais interessantes da família ESP8266, já que pode ser facilmente ligada à um computador para ser programada com a linguagem Lua e também através da IDE do Arduino utilizando uma linguagem similar à linguagem C.

A Figura 5 representa a placa NodeMCU, através desta, percebe-se que o seu formato e as dimensões facilitam a integração com diversos componentes como, por exemplo, os *jumpers* e a *protoboard*, normalmente utilizados em pequenos projetos.

**Figura 5.** Placa NodeMCU ESP8266



Fonte: Disponível em: <<https://www.filipeflop.com/blog/esp8266-nodemcu-como-programar/>>. Acesso em: 08 dez. 2017.

#### 4.6 Protótipo

O protótipo desenvolvido tem como objetivo realizar o monitoramento do consumo de água. Seu funcionamento está representado na Figura 6. Através dessa figura, percebe-se que todos os sinais elétricos gerados serão enviados pelo sensor de fluxo de água (2) para a placa NodeMCU ESP8266 (3) que, em seguida, enviará os dados via *wireless* para a base de dados *Firebase Realtime Database* (4), o que tornará possível o monitoramento do consumo através de uma aplicativo Android instalado no smartphone (5), ou em outro dispositivo móvel com a plataforma Android.

**Figura 6 - Exemplo do Funcionamento do Sistema**



Fonte: Autores, 2017.

Inicialmente, nos primeiros testes, o protótipo apresentou um bom comportamento na execução das funções que foram especificadas durante a programação. E realizou o monitoramento do consumo de água, indicando resultados aproximados do consumo real.

Igor Galindo Paiva | Denise Xavier Fortes | Ricardo Azevedo Porto

Os experimentos foram efetuados a partir da utilização de uma maquete que está representada na Figura 7 abaixo:

**Figura 7** - Maquete utilizada durante o experimento



Fonte: Autores, 2017.

A maquete, representada pela figura 7, foi desenvolvida para simular uma residência. Onde, a água sairia da fonte de abastecimento e passaria pelo sensor de fluxo, enviando assim, os dados através da placa NodeMCU ESP8266 para o *Firebase*. Desta forma, o usuário passou a ser informado quanto ao consumo de água em tempo real.

Durante o processo de desenvolvimento, os parâmetros do sensor foram alterados na programação para que o sistema pudesse apresentar valores da medição mais aproximados dos valores de consumo real.

## 5 CONSIDERAÇÕES FINAIS

Com o intuito de permitir ao usuário o acompanhamento dos dados sobre o consumo de água, apresentando algumas conversões básicas como: o consumo em litros, metros cúbicos e valor aproximado em Real (R\$). Desenvolveu-se um sistema elétrico responsável por capturar os dados da medição e gravá-los em uma base de dados online, tornando-se possível a realização da leitura e conversões dos dados, em informações úteis, para os usuários através de um aplicativo Android.

Igor Galindo Paiva | Denise Xavier Fortes | Ricardo Azevedo Porto

---

Portanto, o aplicativo instalado no dispositivo móvel, propôs ao usuário um acompanhamento do consumo por data, com a quantidade de água consumida exibida em litros e em metros cúbicos, bem como informes, através de mensagens, do limite de consumo diário atingido.

Durante o processo de desenvolvimento, foram percebidos alguns problemas, dentre estes, a precisão do sensor de fluxo de água e o tempo que a placa NodeMCU ESP8266 utilizava para gravar os dados na base de dados online. No entanto, os problemas foram parcialmente solucionados, tornando-se capaz a realização do monitoramento do consumo com valores aproximados da quantidade de água real. Por ser um protótipo, com objetivo de apresentação de ideia, o projeto apresentou resultados consideravelmente satisfatórios.

## REFERÊNCIAS

- BATSCHINSKI, George. Backend as a Service: Prós e Contras. Disponível em: <<https://www.infoq.com/br/news/2016/07/backend-pros-e-contras>>. Acesso em: 10 dez. 2017.
- CAPRILES, René. Água e Desastres. Disponível em: <[http://ambientes.ambientebrasil.com.br/agua/artigos\\_agua\\_doce/agua\\_e\\_desastres.html](http://ambientes.ambientebrasil.com.br/agua/artigos_agua_doce/agua_e_desastres.html)>. Acesso em: 15 nov. 2017.
- CHASE, Otávio. Sistemas Embarcados. 2007. 7 p.
- COCCHIA, Annalisa. Smart and digital city: A Systematic Literature Review. Smart City: How to Create Public and Economic Value with High Technology in Urban Space, 2014. 13-43 p.
- CORDEIRO, Fillipe. Android SDK: O que é? Para que Serve? Como Usar?. Disponível em: <<https://www.androidpro.com.br/android-sdk/>>. Acesso em: 12 dez. 2017.
- DA CUNHA, Fulvio Alexandre. Prototipação de Software. Disponível em: <<https://www.webartigos.com/artigos/prototipacao-de-software/1896>>. Acesso em: 19 nov. 2017.
- DELAI, Andre Luiz. **Sistemas Embarcados**: a computação invisível. Disponível em: <<http://www.hardware.com.br/artigos/sistemas-embarcados-computacao-invisivel/conceito.html>>. Acesso em: 01 dez. 2017.
- EMBEDDED ARCHITECTS. **O que é um sistema embarcado**. Disponível em: <<http://www.embarc.com.br/p1600.aspx>>. Acesso em: 02 dez. 2017.

Igor Galindo Paiva | Denise Xavier Fortes | Ricardo Azevedo Porto

KÖCHE, José Carlos. Fundamentos de Metodologia Científica: Teoria da ciência e iniciação à pesquisa. 26. ed. Rio de Janeiro: Vozes, 2009.

LACHETA, Ricardo R. Google Android: aprenda a criar aplicações para dispositivos móveis com Android SDK. 2. ed. São Paulo: Novatec Editora, 2010.

LEITE, Jair C. Engenharia de Software: O modelo evolutivo. Disponível em: <<http://engenhariadesoftware.blogspot.com.br/2007/03/o-modelo-evolutivo.html>>. Acesso em: 19 nov. 2017.

MCROBERTS, Michael. Arduino Básico. São Paulo: Novatec Editora, 2011. 453 p.

ONU BRASIL. Até 2030 planeta pode enfrentar déficit de água de até 40%, alerta relatório da ONU. Disponível em: <<https://nacoesunidas.org/ate-2030-planeta-pode-enfrentar-deficit-de-agua-de-ate-40-alerta-relatorio-da-onu/>>. Acesso em: 16 nov. 2017.

PETRY, Márcio Luís. Controle Híbrido de um Robô Autônomo Seguidor de Linha. 2016. 83 p. TCC (Engenharia da Computação) - Universidade Tecnológica Federal do Paraná, Pato Branco, 2016.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico. 2. ed. Novo Hamburgo: Feevale, 2013. 271 p.

RIBEIRO, M. Isabel. Sensores em robótica, Enciclopédia Nova Activa Multimédia, Volume de Tecnologias, p.228-229, Portugal, 2004.

SALIBY, Eduardo. Tecnologia de Informação: Uso da Simulação para Obtenção de Melhorias em Operações Logísticas. Disponível em: <<http://www.ilos.com.br/web/tecnologia-de-informacao-uso-da-simulacao-para-obtencao-de-melhorias-em-operacoes-logisticas/>>. Acesso em: 20 nov. 2017.

SHANNON, Robert E. **System Simulation**: The Art and Science. [S.l.]: Prentice Hall, 1975. 368 p.

SILVA, André Koide da. Cidades Inteligentes e sua Relação com a Mobilidade Inteligente. Disponível em: <[https://edisciplinas.usp.br/pluginfile.php/1918002/mod\\_folder/content/0/Artigo%20-%20Mobilidade%20Inteligente.pdf?forcedownload=1](https://edisciplinas.usp.br/pluginfile.php/1918002/mod_folder/content/0/Artigo%20-%20Mobilidade%20Inteligente.pdf?forcedownload=1)>. Acesso em: 29 nov. 2017.

SILVEIRA, Cristiano Bertulucci. O que é PWM e Para que Serve?. Disponível em: <<https://www.citisystems.com.br/pwm/>>. Acesso em: 29 nov. 2017.

SILVEIRA, João Alexandre da. ARDUINO: Um Guia Básico para Iniciantes. Disponível em: <<http://www.paulobrites.com.br/wp-content/uploads/2015/04/ARDUINO-%E2%80%93-Um-Guia-B%C3%A1sico-para-Iniciantes.pdf>>. Acesso em: 02 dez. 2017.

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p.

Igor Galindo Paiva | Denise Xavier Fortes | Ricardo Azevedo Porto

---

SOUZA, Karla Pires de; GASPAROTTO, Angelita Moutin Segoria. A importância da Atividade de Teste no Desenvolvimento de Software. [201?]. 17 f. XXXIII Encontro Nacional de Engenharia de Produção, Salvador, 2013.

THOMSEN, Adilson. Como programar o módulo ESP8266 NodeMCU. Disponível em: <<https://www.filipeflop.com/blog/esp8266-nodemcu-como-programar/>>. Acesso em: 11 dez. 2017.