ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 7.327

Volume 6, Issue 1, January 2018

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

IoT: NodeMCU 12e X Arduino Uno, Results of an experimental and comparative survey

Dr. Antonio Carlos Bento
Technologies of Intelligence and Digital Design
Pontifícia Universidade Católica de São Paulo
São Paulo – Brazil

Abstract: ***This study presents the results obtained in an experimental and comparative research involving the devices Nodemcu 12e and Arduino UNO, carried out in the period of 2017, focusing mainly on the positive and negative points presented by the different devices when using a WiFi network. the data collection was an experimental research with the devices, applying the heuristic classification of William J. Clancey, considering personal experience when using these types of equipment. After obtaining the data, proceeded to consolidate, categorize and analyze the quantitative and qualitative results. There was a perception about the results that point out the pros and cons on the different technologies, among these are: ease of configuration; the centralization of resources; ease and difficulties. In a counterpoint, the Arduino Uno device was demonstrated with little capacity and need to add new devices for communication via WiFi, this way Nodemcu 12e, has almost the same market value, but already has the WiFi feature internally, in addition to has greater capacity and ease of use.***

Keywords: internet; arduino; nodemcu; automation; IoT.

I. INTRODUCTION

The motivation for the development of this work had the origin of personal and group experiences, in which the main advance and use of resources for residential automation was observed, mainly because it is a trend that is being applied in different areas such as robotics, residential and industrial automation, as shown by the studies presented by Zabadal and Castro [19]. As general objectives, the use of Arduino Uno and Nodemcu 12e devices should be analyzed, with the specific objectives being to carry out WiFi connection tests and their configurations.

The results should be of the quantitative and qualitative type, as a comparison between the consolidated values is necessary, as well as a detailed analysis of the use of the devices, pointing out their strengths and weaknesses, focusing on the development tools and method of application and use.

II. APPLICATION OF METHOLOGIE

The methodology used follows the models presented by Silva [16] and Marconi and Lakatos [12], as Silva presents [16]: during an experimental research using the observational and experimental method, which may somehow reflect its feeling about the problem, the applicable hypotheses are observed with real suggestions, which may somehow contribute with data to be analyzed and compared, thus having a greater focus on the object of study.

According to Marconi and Lakatos [12], field research is the one used to obtain information and / or knowledge about a problem, for which a response or hypothesis is sought, or, still discover new phenomena or the relations between them.

Field research involves university-level students who have had access to assessments and disciplines in ODL at the institution. This field research compared the different results and experiences of the individuals, through a consolidation of results, which can provide a more detailed analysis and familiarity with the problem, considering the standard use of the solutions with the presented proposal.

Questions were used to evaluate the opinion and understanding of the interviewees with some types of questions, as presented by Marconi and Lakatos [12]:

- Questions of the true and false type: where the respondent selects a response that is considered true or false, such as: Have you ever taken distance courses? (Yes No).
- Inquiries: In this type of question the respondent responds to some real fact, being descriptive or by selection, for example, the questions about the interviewee's schooling and work area were used: What is your profession: descriptive field in which the interviewee enters his profession; What is your age? Descriptive field in which the respondent types his / her age.
- Question of opinion: In which the interviewee expresses his opinion on some subject, such as: Have you seen anything like this? (Yes No); What are the good points? Descriptive; What are the bad points? Descriptive.

The requirements that served as a basis for the collection of information and subsequent consolidation and analysis were developed in a form that, according to Marconi and Lakatos [12]: "The form is one of the essential instruments for social research, whose collection system is to obtain information directly from the interviewee. "

The method used to validate the results was developed based on the Clancey Heuristic Classification [6] to solve problems involving data abstraction for evaluations, taking into account the data obtained during the tests performed with the analyzed devices.

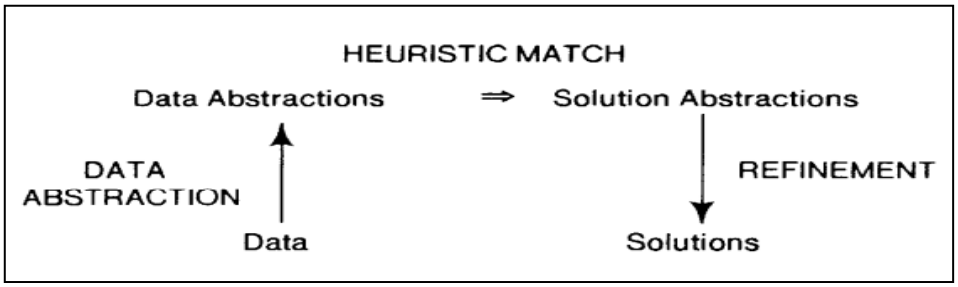

Figure 1: Inference structure for classification and problem solving. Clancey[6]

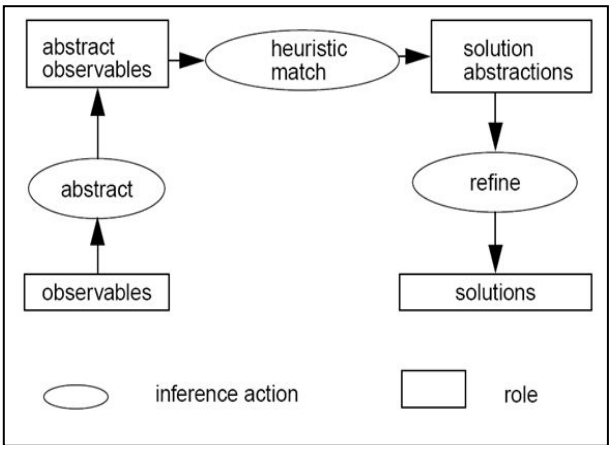

Figure 2: Representation of the heuristic classification of Clancey [6].

Data (knowledge action, observables): following the Clancey model [6], the data were selected through questions that were asked based on the project needs for an internet connection.

Abstracting (inference action): Analyzes and comparisons were developed with data provided by manufacturers as well as empirical tests.

Data abstraction (knowledge action, abstract): Data were selected according to the characteristics of each device, making a comparison between their main positive and negative points.

Confronting heuristics (inference action): the values are confronted with the different sets collected during the exploratory research, relating the positives and negatives to a decision making, adding up the values of each item.

Abstract solution or hypothesis (knowledge action): the hypotheses are used to identify and evidence the problem, for this are presented quantitative values referring to the sum of the results of the research, thus allowing a more detailed analysis using the statistics.

To specify (inference action, heuristic match): with the results of data analysis, using comparisons and working with the hypothesis and presentation of abstract solutions, one can contribute to the direction of some action to contour problems perceived during the study, or even, present a proposal for treatment of the results as decision making, on the choice of the most appropriate device.

Solutions abstractions (knowledge action): After obtaining the results of the evaluations, the evidence of the problem, using the hypotheses to compare the results with the devices Arduino Uno and NodeMCU 12e, the results of these studies are intended to present a validation of the proposal for the exposure of the problem and possible decisions of the points that stood out.

III. Bibliography Review

There are different explanations about the concept of IoT (Internet of Things), as presented by the survey carried out by Zabadal and Castro [19], citing authors such as: Ashton [2]; Kranenburg et al. [11]; European Research Projects on the Internet of Things (CERP IoT,) [5]; Atzori et al. [3]; Oracle [15]; CASAGRAS [4]; Friedewald and Raabe [7].

The group that effectively studies IoT-related issues is called the Telecommunication Standardization Sector (ITU-T) [10], which defines IoT as follows:

"The Internet of Things (IoT) has been defined in ITU-T Recommendation Y.2060 (06/2012) as a global infrastructure for the information society, enabling advanced services interconnecting things (physical and virtual) based on existing interoperable information and constantly evolving and communication technologies. " ITU-T (2016). Translated by the author.

Vermesan et al. [18], explain that IoT is changing the market for industries and consumers. Robots, automated vehicles, augmented reality, virtual reality, drones, among other devices with virtual intelligence, should be part of the development of applications to use the Internet resources of Things. In this way, with the evolution of artificial intelligence, you apply machines like robots to provide smarter resources for decision making or even problem solving, providing innovative experiences for your consumers.

Vermesan et al. [18] also points out that IoT is the missing bridge between the virtual, digital and physical world, uniting people, processes, data and devices while generating knowledge about the internet applications of things and their platforms.

IoT technology is creating the next generation of intelligent homes and buildings, intelligent vehicles and enabling applications in manufacturing through intelligent automation, using artificial intelligence and learning machines, integrating these devices into the components of an IoT-type infrastructure.

Making an analogy to the explanation about Arduino [1], one has: Arduino is an open source electronic platform based on easy-to-use hardware and software. Arduino boards are capable of reading inputs - lights on a sensor, touching a button, or sending and receiving a message via Twitter - and turning it into an output - activating a motor, lighting an LED, posting

something online, sending a set instructions for the Arduino microcontroller. To do this, Arduino uses its own programming language with Arduino software (IDE), based on processing.

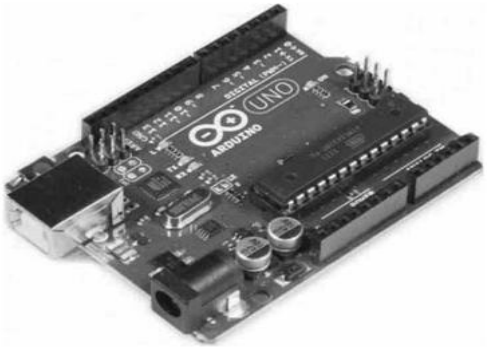

Figure 3: Arduino Uno presented by McRoberts [13].

Veloso et al. [17], explains that the NodeMCU is from the ESP8266 family, being one of the easiest to use, and it is not necessary to use another device of the Arduino type, since it already has the necessary processing capacity to execute its applications, it still has the direct connection to WiFi, without the need to install new devices, unlike the Arduino Uno, which does not have this capability and needs other connections.

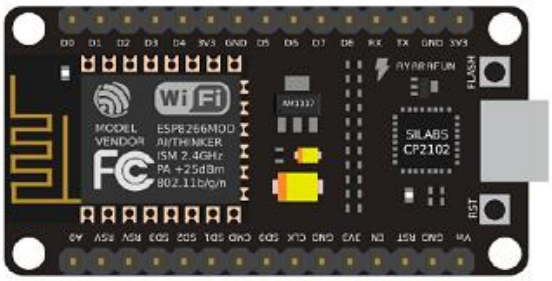

Figure 4: NodeMCU 12e device, presented by Veloso et al. [17].

IV. Results and Discussions

The experiments were applied to the 2 types of devices using the bibliographic references of Arduino UNO [1], NodeMCU 12e [17] and the manufacturer of NodeMCU [14], and the libraries and documents in GitHub [9] of libraries to work with different devices for the internet of things.

In addition to the manual tests performed directly on the devices, considering a score between 1 and 0 (binary, has or does not have) for each question, considering value 1 as relevant, that is, has the resource evaluated, and 0 for irrelevant, considering situations in which the resource does not exist, comparing relevant points of use such as: speed, capacity and processing, in this way the following items were selected and evaluated:

- Price: The value of the devices varies according to the places searched, were found very low prices, as very high, for example, for the NodeMCU prices were found between $ 19.00 and $ 60.00 for the Arduino One, values between R $ 18.00 and R $ 70.00 were found, considering the lower prices, the two equipments were demonstrated with equivalent values, but it must be taken into account that the Arduino Uno, needs another device to to connect to the internet, in the case would be the ESP8266, which costs around R $ 25, thus taking into consideration the requirements for internet access, NodeMCU 12e has greater advantages ..

- Integrated WiFi: the WiFi requirement is important for saving parts, also involves connecting to other devices, as previously reported, the NodeMCU 12e, already has the integrated WiFi, the same does not occur with the Arduino Uno, this way the Arduino Uno stands at a disadvantage during comparison.

- Expansion board: In order to increase the number of connection pins, it is necessary to use expansion boards, they are called Shields, in this case, both devices have expansion cards, but it does not come with the devices, in this case if it is necessary to purchase them separately, the price is around $ 25.00, for each model.

- Smaller size: The size requirement was considered for better coupling in boxes or small devices, besides considering the weight, in this case the NodeMCU 12e has the measures of 2.5cm X 5.1cm, while the Arduino Uno has 5 , 3cm X 7.9cm, in this requirement the NodeMCU 12e presented the smaller size, which should be considered for the projects.

- Increased processing: The NodeMCU 12e has a processing speed of 160mhz with an ATMega 328P processor, while the Arduino Uno has a speed of 16mhz with a Tensilica LX2016 processor, so the Arduino Uno has shown a slower processing compared to the NodeMCU.

- Better documentation: During the tests, it was possible to identify few references on the NodeMCU 12e, due to the fact that it is a new product, just launched in the market, the Arduino Uno, has references older and with more quantity.

- Simple encoding: Both devices can be used on the same platform, such as Arduino's own IDE, but in this case NodeMCU uses direct references to its libraries, without the need to use additional libraries, the way of connection and use is the you need only select the device type on the platform. Another advantage of Nodemcu12e is the possibility of working with the language C, LUA, C ++, and also has an exclusive development platform.

- Analog Outputs: In this requirement, the NodeMCU 12e has only one analog port, compared to six analog ports on the Arduino Uno, reinforcing that the use of analog pins will depend on other types of devices, this to meet the different designs, for this test not it was necessary to use no analogue pins.

- Digital outputs: The Arduino Uno has thirteen digital ports, against 11 pins of the NodeMCU 12e, these pins are also used for connections with different types of devices, in the case of this research, no digital pins of the NodeMCU were used.

- Number of pins: In this case, as previously informed the number of pins of the NodeMCU 12e is much smaller than the number of ports of the Arduino Uno, this requirement is necessary to be considered for the connection of different types of devices, depending on the project in progress , but to increase the capacity of ports is used the Shield, or expansion board.

- Uses 3V: The 3V input allows the connection of devices that use this type of voltage, in case the two devices use and make available inputs for devices with 3V.

- Uses 5V: The 5V input allows the connection of different types of devices that use this type of voltage, in case the two devices use and make available inputs for devices with 5V.

- Reset button: This requirement is required for the times when the code load is being sent to the device, which may lock or otherwise be blocked, in this case the Reset button, allows the devices can be reset, the two devices have reset button.

- Memory RAM: This requirement involves the memory in which the variables are stored, among other values during the execution of the code, ie, those data that are necessary for the execution of the program, in this case the Arduino Uno has 2kbytes of RAM, being that the NodeMCU 12e has 20Kbytes, the latter has a capacity far superior to the Arduino Uno.

- Flash Memory: This requirement involves the memory used to record the programs to be used with the devices, especially for the increased capacity presented by NodeMCU 12e, with 4Megabytes versus 32Kbytes of Arduino Uno, this requirement is very important because it involves the size of the program that should be loaded on the devices, in this case, a program developed for NodeMCU, may not be loaded on an Arduino Uno, because it would not have its size in bytes.

- MicroUSB: The MicroUSB connection allows both the transfer of data from a computer to the device, as well as the power source to connect them as needed, in this requirement the two devices have a Micro USB connection.

- WiFi Speed: As the Arduino Uno does not have internet connections, it is necessary to use external devices such as an ESP8266 that allows 802.11b and g connections (11MBps and 56MBps speed limit, respectively). The NodeMCU 12e has the same connections besides the 802.11n that has a limit 300MBps of speed and has a range up to twice as much, so the NodeMCU presented better capacity of speed for WiFi.

Table 1: Comparative table with the devices Nodemcu 12e and Arduino Uno. Developted by author.

Requirement	NODEMCU	ARDUINO
Price	1	1
WiFi integrated	1	0
Expansion Shield	1	1
Minor size	1	0
High process	1	0
Best documentation	0	1
Simple code	1	0
Analogic output	1	1
Digital output	1	1
Pins available	0	1
Use 3v	1	1
Use 5v	1	1
Reset button	1	1
Flash memory	1	0
RAM memory	1	0
MicroUSB	1	1
Best WiFi speed	1	0
TOTAL	15	10

Analyzing the results presented on the consolidation of the desired requirements to develop an automation project, in which it is only necessary to send or access information from a WiFi network, we have: a total of 17 items (88.24%), 15 being considered relevant for the NodeMCU 12e device, 10 items being relevant to the Arduino Uno, or 58.82% that this device meets a design for automation.

The difference between the two devices shows 29.41% of variation, considering a probability of 5%, the result represents a great relevance, in this way the use of the NodeMCU 12e, demonstrated that this equipment is superior in different aspects in comparison with the Arduino One.

Connection tests were performed using the following sample code available in GitHub [9] and Arduino IDE [1]:

```
/*
 WiFi Web Server
A simple web server that shows the value of the analog input pins.
using a WiFi shield.
This example is written for a network using WPA encryption. For
WEP or WPA, change the Wifi.begin() call accordingly.
Circuit:
* WiFi shield attached
* Analog inputs attached to pins A0 through A5 (optional)
created 13 July 2010
by dlf (Metodo2 srl)
modified 31 May 2012
by Tom Igoe
*/
```

```
#include <SPI.h>
#include <WiFi.h>

char ssid[] = "yourNetwork";      // your network SSID (name)
char pass[] = "secretPassword";   // your network password
int keyIndex = 0;                 // your network key Index number (needed only for WEP)

int status = WL_IDLE_STATUS;
WiFiServer server(80);
void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  // check for the presence of the shield:
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue:
    while (true);
  }
  String fv = WiFi.firmwareVersion();
  if (fv != "1.1.0") {
    Serial.println("Please upgrade the firmware");
  }
  // attempt to connect to Wifi network:
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    status = WiFi.begin(ssid, pass);
    // wait 10 seconds for connection:
    delay(10000);
  }
  server.begin();
  // you're connected now, so print out the status:
  printWifiStatus();
}
void loop() {
  // listen for incoming clients
  WiFiClient client = server.available();
  if (client) {
    Serial.println("new client");
```

```
// an http request ends with a blank line
boolean currentLineIsBlank = true;
while (client.connected()) {
 if (client.available()) {
  char c = client.read();
  Serial.write(c);
  // if you've gotten to the end of the line (received a newline
  // character) and the line is blank, the http request has ended,
  // so you can send a reply
  if (c == '\n' && currentLineIsBlank) {
   // send a standard http response header
   client.println("HTTP/1.1 200 OK");
   client.println("Content-Type: text/html");
   client.println("Connection: close");  // the connection will be closed after completion of the response
   client.println("Refresh: 5");  // refresh the page automatically every 5 sec
   client.println();
   client.println("<!DOCTYPE HTML>");
   client.println("<html>");
   // output the value of each analog input pin
   for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
    int sensorReading = analogRead(analogChannel);
    client.print("analog input ");
    client.print(analogChannel);
    client.print(" is ");
    client.print(sensorReading);
    client.println("<br />");
   }
   client.println("</html>");
   break;
  }
  if (c == '\n') {
   // you're starting a new line
   currentLineIsBlank = true;
  } else if (c != '\r') {
   // you've gotten a character on the current line
   currentLineIsBlank = false;
  }
 }
}
// give the web browser time to receive the data
delay(1);
// close the connection:
client.stop();
```



```
    Serial.println("client disonnected");
  }
}
void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  // print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}
```

There were no problems with the NodeMCU 12e device, but for the Arduino Uno, it was necessary to attach another device like the ESP8266, for the WiFi connection, since it does not have this feature as standard, so the use of the Arduino Uno has proved more complex and cost more than the same process with NodeMCU 12e.

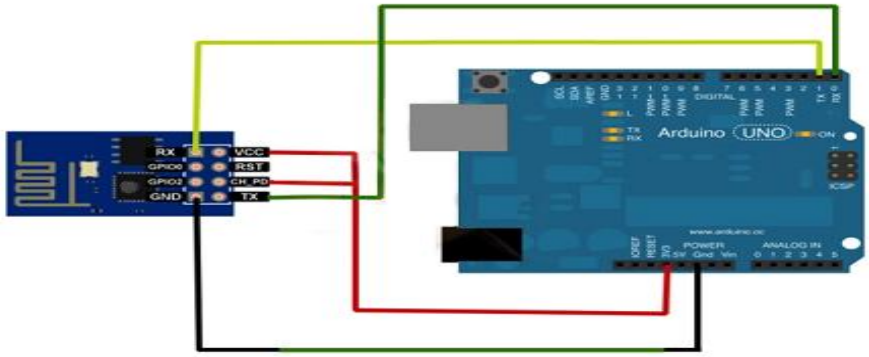

Figure 5:Device schema withArduino Uno connected with the device ESP8266 for WiFi connection. Developted by author

The wiring diagram follows the guidelines given on the manufacturer's website at GitHub [9].

V. Conclusion

With the results obtained and analyzed, it is possible to conclude and observe the advantages of using the NodeMCU 12e device, some of the positive points observed are the large storage and processing capacity compared to the Arduino UNO device, the results presented during the tests and simulation highlight the ease of use of NodeMCU 12e, in addition to the connections that were sufficient for this study.

In this way the objectives of the project were fulfilled in what is compared and analyzes the results obtained, which provided an identification of the main positive and negative points between the two analyzed devices. Noting the negative points, which did not represent any impediment to the tests, such as the smaller number of ports in the NodeMCU compared to the Arduino UNO.

Another important point observed was the ease of connecting to the internet, in addition to that the NodeMCU 12e device can also be used as a Web server, allowing access to an internal page, in which users can configure it or even access devices remotely, allowing them to be used in large environments, but an internet connection is required.

As observed in a large part of the bibliographies, the growth in the use of devices for the Internet of things is increasing, due to the use in residential and industrial automation projects, providing the execution of tasks mechanically or automatically, considering artificial intelligence as support for decision making, for the low price and ease it is possible to notice this increase when talking about the internet of things, in different groups of people.

Other points that stood out during the research were the low cost of assembling a complete project, being able to attend different types of applications and resources, in this way the products became viable for the different types of public, demonstrating capacity of processing and communication. For different types of projects it should be necessary to create different types of storage boxes, in this way 3D printers can support projects.

References

1. Arduino. 2017. Site oficial. Disponível em: <https://www.arduino.cc> acesso em 28/09/2017.
2. Ashton Kevin. (2009). That 'Internet of Things' thing. Publicado no RFID Journal. Disponível em <http://www.rfidjournal.com/article/view/4986>. Acesso em: 13 Mar. 2017.
3. Atzori, Luigi; IERA, Antonio; Morabito, Giacomo. (2010) The internet of things: a survey. Computer Networks.
4. CASAGRAS (Org.). RFID and the Inclusive Model for the Internet of Things. The Old Vicarage, UK: CasaGras An Eu Framework 7 Project, [S.D.].
5. CERP. (2009). IoT – Internet of Thinsgs of European Research Cluster. Internet of things: Strategic Reserach Roadmap, Disponível em: <http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf> Acesso em: 26 set. 2017.
6. Clancey, W. J. (1985). Heuristic classification. Artificial Intelligence, v. 27, n. 3, p. 289–350, dez. 1985. ISSN: 00043702.
7. Friedewald, M.; Raabe, O. (2011). Ubiquitous computing: an overview of technology impacts. Telematics and Informatics, 2011.
8. Gartner. (2017). Internet of Things Definition. Disponível em: <http://www.gartner.com/it-glossary/internet-of-things/ >. Acesso em: 25 set. 2017.
9. GitHub. Site oficial das bibliotecas para os dispositivos. Disponível em: <https://github.com/nodemcu/nodemcu-firmware> Acesso em: 22 set. 2017.
10. ITU-T. (2012). Internet of Things Global Standards Initiative. Disponível em: http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060. Acesso em: 28/09/2017.
11. Kranenburg, R. V. et al. (2011). The internet of things. Paper Prepared for the 1st Berlin Symposium on Internet and Society, October 25-27, 2011.
12. Marconi, M.; Lakatos, E. (2010). Fundamentos de metodologia científica. 5. ed. São Paulo, Brasil: Ed. Atlas, 2010. p. 310 ISBN: 8522433976 ISSN: 9788522457588
13. McRoberts, Michael. (2015). Arduino Básico. 2a Edição. Novatec Editora. São Paulo, SP. ISBN: 978-85-7522-404-5
14. NodeMCU. (2017). Site oficial do dispositivo e suas documentações. Disponível em: < https://nodemcu.readthedocs.io/en/master/ > Acesso em: 18/08/2017.
15. Oracle. (2015). Java and IoT: The Intelligent Platform for the Connected Vehicle. Redwood Shores, Ca: Oracle Java Embedded, 2015.
16. Silva, C. R. D. O. E. (2004). Metodologia e organização do projeto de pesquisa (guia prático). Centro Federal de Educação Tecnológica do Ceará, v. 1, p. 34, 2004.
17. Veloso, Artur F.; Sousa, Beatriz A.; Braz, Arleide R.; Rabelo, Ricardo A. L.; Brito, Elizângela M.; Lima, Erick M. S. (2017). Prototipação com nodeMCU para Internet das Coisas em Smart Cities: Uma pesquisa. III Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos, v. 1, n. 1, p. 105-110, jun, 2017.
18. Vermesan, Ovidiu; Eisenhauer, Markus; Sundmacker, Harald; Guillemin, Patrick; Serrano, Martin; Tragos, Elias Z.; Valiño, Javier; derWees, Arthur; Glubak, Alex; Bahr, Roy. (2017). Internet of Things Cognitive Transformation Technology Research Trends and Applications. IERC - European Research Cluster on the Internet of Things.

AUTHOR(S) PROFILE

Dr. Antonio Carlos Bento, Ph.D. in Cognitive Environments and Digital Design at the PUC-SP Digital Technologies and Intelligent Technologies course. Holds a Master's degree in Information Systems Technology from the Fundação and Instituto de Ensino Para Osasco. He holds an MBA in Information Systems from the University of São Paulo. Graduated in Data Processing from Ibirapuera University and is currently a professor in undergraduate and postgraduate courses. He has experience in the area of Computer Science, with emphasis on Information Technology, Project Management, Database, Software Engineering, Education Software Systems, Intelligent Systems and Specialists, Security, Systems Modeling and Information Technology Management. Acting on WBAN projects, artificial intelligence and information technology.

