Sistema de Irrigação Automatizado Com Interface de Controle e Monitoramento Online

1Alex Manoel Coelho,2 Wellington Schütz, 3Wesley Bezerra

1Aluno da 8ª fase do Curso de Ciência da Computação do Instituto Federal Catarinense – Campus Rio do Sul, e-mail: alexma\_coelho@hotmail.com; 2 Aluno da 4ª fase do Curso de Ciência da Computação do Instituto Federal Catarinense – Campus Rio do Sul, e-mail: [wellington.schu@live.com](mailto:wellington.schu@live.com); 3Professor de Ensino Superior, Técnico e Tecnólogo do Instituto Federal Catarinense – Campus Rio do Sul, e-mail: wesley@ifc-riodosul.edu.br.

Resumo

*Atualmente encontramos muitas soluções de irrigação para o setor agrícola, mas, no entanto, muitas dessas tecnologias apresentam um custo muito elevado, ou até por serem soluções para grandes plantações, as tornando inviável para o uso dos pequenos produtores. Os métodos de irrigação que se aplicam a esses pequenos agricultores, apesar de sua eficiência e custo benefício, não apresentam soluções tecnológicas atuais, ainda mantendo características primitivas em sua aplicação, para tal problema, buscamos apresentar uma solução de baixo custo e com uso de recursos tecnológicos atuais.*

Palavras-chave: Arduino; Automação; Agricultura

1. INTRODUÇÃO

Será tratado nesse trabalho de um sistema de irrigação automatizado, controlado por uma interface *web*, que tem como finalidade criar um sistema de controle da irrigação acessível e simples para os pequenos agricultores.

Tendo em vista o fato que a região do Alto Vale do Itajaí é extremamente agrícola e é dominada pela agricultura familiar, foi vista à necessidade de criar um sistema para a irrigação dessas propriedades, que seja acessível á essas famílias de agricultores, e ainda possibilite mais agilidade no gerenciamento e ativação da irrigação, não importando o local onde o operador da irrigação se encontre, sendo somente necessária uma conexão com a internet.

O sistema tem como objetivo, verificar as condições ambientais através de sensores e a partir da combinação dos dados climáticos obtidos, definir o melhor momento para iniciar a irrigação, além de armazenar todas as configurações e histórico de monitoramento e irrigação em seu banco de dados.

Ele permitirá uma fácil adaptação a irrigação de várias culturas distintas, a partir da inserção dos dados climáticos necessários para a irrigação.

* 1. CONTEXTO GERAL

Atualmente encontramos muitas soluções de irrigação para o setor agrícola, mas, no entanto, muitas dessas tecnologias apresentam um custo muito elevado, ou até por serem soluções para grandes plantações, as tornando inviável para o uso dos pequenos produtores.

* 1. CONTEXTO ESPECIFICO DA SUB-ÁREA

Hoje em dia existem muitas soluções de irrigação de baixo custo para os pequenos produtores, mas algo que se vê nessas soluções é seu baixo nível tecnológico, sendo técnicas muito manuais, ou até tendo sua maior complexidade tecnológica com o uso de bombas elétricas.

* 1. PROBLEMÁTICA DA SUB-ÁREA

Para realização deste trabalho foi utilizado um conjunto de sensores, sendo monitorados e controlados pelo Arduino, que com a combinação gerada dos dados gerados pelos sensores, ativa ou não um relé que controla o acionamento do maquinário responsável pela irrigação.

* 1. SOLUÇÕES EXISTENTE

Podemos encontrar várias soluções já existentes em sistemas de irrigação de baixo custo, no entanto algo que encontramos em comum em todas elas, é o baixo nível tecnológico das mesmas.

Hoje programas governamentais voltados a apoio à agricultura, fornecem cartilhas com técnicas de irrigação de baixo custo. Desses documentos o mais facilmente encontrado é uma cartilha sobre um Sistema de Manejo de Irrigação para a Agricultura Familiar, feito pela Embrapa.

Notamos que a cartilha da Embrapa, tem foco para a irrigação em localidades com recursos hídricos escassos e para agricultura de subsistência. Outro ponto que notamos é que as técnicas apresentadas se resumem somente ao maquinário utilizado e as técnicas são principalmente manuais, sem uso de tecnologia computacional, vemos também um foco em utilização de matérias reaproveitados, cremos que esse sistema possa ser complementado por nossa proposta apresentada. (COELHO et al., 2014)

* 1. PROPOSTA

Para tentar solucionar essa problemática nos sistemas de irrigação de baixo custo, utilizamos nossos conhecimentos adquiridos dentro de nossa graduação para criar uma proposta.

Utilizamos conhecimentos adquiridos na matéria de Sistemas Digitais em sistemas de microcontroladores, utilizamos o conhecimento em programação orientada a objetos obtidas na matéria de Linguagem de Programação I, conhecimento em banco de dados adquiridos nas disciplinas de Banco de Dados e por fim os conhecimentos em programação WEB obtidos na matéria de Linguagem de Programação II.

Nossa proposta se trata de um sistema de irrigação, utilizando um conjunto de sensores controlados por um Arduino, esse conjunto de sensores é posto na área onde se deseja realizar o monitoramento, esses conjuntos de sensores são cadastrados em um sistema *WEB*, onde o usuário poderá informar a sua localização, bem como as configurações com os requisitos para iniciar uma irrigação, além de o usuário poder acompanhar a situação de sua irrigação não somente localmente, mas também de forma remota. O sistema ainda armazena todo o histórico de irrigação e de configurações em um banco de dados.

1. PROJETO
   1. OBJETIVO GERAL

O principal objetivo é apresentar uma proposta de sistema de irrigação de baixo custo, para pequenos agricultores com recursos escassos.

Temos também como objetivo poder demonstrar de forma prática os conhecimentos apresentados na disciplina de Linguagem de Programação II.

* 1. OBJETIVOS ESPECIFICOS

Traçamos alguns objetivos parciais, onde sua conclusão acaba por formar o sistema completo.

O primeiro objetivo parcial é a montagem e configuração dos sensores com o Arduino, tarefa essa simples, visto que esse objetivo já foi atingido na disciplina de Sistemas Digitais.

Outro objetivo foi a criação de uma página web visualmente agradável e com boa usabilidade, para a criação de tal foi utilizado o *framework* Bootstrap, que possibilita a criação de uma página bonita, funcional e ainda responsiva.

Tivemos o objetivo de criar um *Back-End* robusto o suficiente para gerenciar o sistema online, bem como gerenciar o banco de dados, com adição, remoção e alteração das informações armazenas no mesmo. Para tal foi utilizado o sistema JSP (*Java Server Pages*), um sistema para o *Back-End* baseando na linguagem de programação Java

Outro ponto foi a utilização de um banco de dados para a o armazenamento dos dados de irrigação bem como as configurações da mesma, nesse ponto foi optado por utilizar o sistema de banco de dados MySQL, sistema no qual aprendemos a utilizar nas disciplinas de Banco de Dados e utilizamos no conteúdo didático em Linguagem de Programação II.

O objetivo mais importante é a junção de todos os objetivos anteriores, com a junção de tudo temos nosso sistema de irrigação completo para o uso.

* 1. REQUISITOS

Para realização deste projeto foi utilizado um conjunto de várias ferramentas e equipamentos.

Para a parte de do sistema de monitoramento, foram utilizados um conjunto de sensores gerenciados por um controlador, que com a combinação dos dados informados pelos sensores, ativa ou desativa um relé que controla o acionamento do maquinário responsável pela irrigação:

Para a obtenção de dados foram utilizados um conjunto de sensores que segue abaixo:

* + 1. **Sensor DHT11:**

Usado para realizar a leitura da Temperatura e da Umidade do ar. É um sensor de umidade relativa e temperatura, com saída digital calibrada. Possui uma exclusiva tecnologia para medir a umidade, garantindo a confiabilidade e estabilidade. Possui internamente um microcontrolador de 8 Bits para tratar o sinal. Possui tamanho compacto, baixo consumo, encapsulamento simples com apenas quatro terminais, podendo transmitir o sinal por até 20 metros. Todas essas características permitem utilizar esse sensor nas mais diversas aplicações. (CARVALHO, 2013)

**Parâmetros Técnicos:**

* Tensão de Alimentação: +5V;
* *Range* de temperatura: 0 a 50ºC com erro de +/- 2ºC;
* *Range* de umidade: 20 a 90% RH com erro de +/- 5%;
* Interface Digital.
  + 1. **Sensor Moisture Sample:**

Usado para realizar a leitura da humidade do solo. Este sensor de pode ler a quantidade de umidade presente no solo ao seu redor. Este sensor utiliza as duas sondas para passar corrente através do solo, e, em seguida, a resistência faz uma leitura para obter o nível de umidade. O fator principal é a água, pois conduz eletricidade mais facilmente (menor resistência), e sem água e com o solo seco a condutividade de energia é menor (maior resistência). (CARVALHO, 2013)

**Parâmetros Técnicos:**

* Tensão de Alimentação: 3,3V ou 5V;
* Tensão do sinal de saída: 0 a 4,2V;
* Corrente: 35mA.
  + 1. **Relay:**

Tem a capacidade de controlar diferentes tensões e potências sem o risco de danificar o micros-controlador. (CARVALHO, 2013)

* + 1. **Arduino Uno R3:**

É uma placa de microcontrolador, ele tem 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB, uma entrada de alimentação uma conexão ICSP e um botão de reset. (CARVALHO, 2013)

Para a criação da interface *WEB* usamos basicamente HTML, CSS e JavaScript, para agilizar a confecção e facilitar a criação do código da interface utilizamos o *Framework* Bootstrap.

O “Framework” Bootstrap, é uma biblioteca CSS e HTML que fornece *templates* para a tipografia, botões, navegação e componentes de navegação em geral, onde o usuário basta chamar a classe em uma determinada *tag* HTML para o visual ser aplicado na página. (TWITTER, 2016)

Para o *Back-End* acabamos por utilizar o sistema JSP, que necessita de vários componentes para poder ser utilizado, no caso desse projeto os componentes foram os seguintes (SESHADRI, 1999):

* JDK (*Java Development Kit*);
* Netbeans IDE;
* Apache Tomcat Server.

O JDK é o conjunto de componentes para o desenvolvimento na linguagem de programação Java. Ele é composto pela JVM (*Java Virtual Machine*), que é uma máquina virtual, que compila o código Java para um código que pode ser lido por qualquer máquina, também é composto ainda por um conjunto de bibliotecas com funcionalidades que facilitam a programação em JAVA. (ORACLE, 2013)

O Netbeans IDE é um ambiente de desenvolvimento integrado que possibilita a o desenvolvimento em uma série de linguagens, que nesse caso em especifico foi utilizado para desenvolvimento em Java, ele oferece as ferramentas necessárias para o desenvolvimento tanto de aplicações *Desktop*, como desenvolvimento *WEB*. Em sua instalação é possível instalar junto todos os componentes para o desenvolvimento das aplicações, nesse caso em especifico optamos por instalar o Netbeans com as ferramentas necessárias para o desenvolvimento em JSP, o que inclui em sua instalação o Apache Tomcat. (ORACLE, 2016)

O Apache Tomcat é um servidor WEB Java, que implementa as tecnologias Java Servlet e JavaServerPages. Ele proporciona um ambiente onde os códigos Java puros, podem ser executados para a confecção de sistemas WEB com essa linguagem. (TOMCAT FOUNDATION, 2013)

O último ponto necessário para o sistema é o seu banco de dados, no qual foi utilizado o sistema gerenciador de Banco de Dados MySql. Se trata de um sistema de banco de dados muito difundido mundialmente, principalmente devido a sua fácil integração com a linguagem de programação PHP, essa sendo a linguagem de programação mais utilizada para criação de sites para a internet. É um sistema de banco de dados muito funcional, principalmente por sua portabilidade, pois roda em diversos sistemas operacionais e por sua compatibilidade, por suportar diversas aplicações que rodam em uma grande diversidade de linguagens de programação e numa diversidade de plataformas. (ORACLE, 2016)

* 1. DESENVOLVIMENTO

Para a aplicação do projeto foram criadas uma série de classes para a manipulação das informações do sistema, o diagrama com essas classes pode ser visto abaixo.

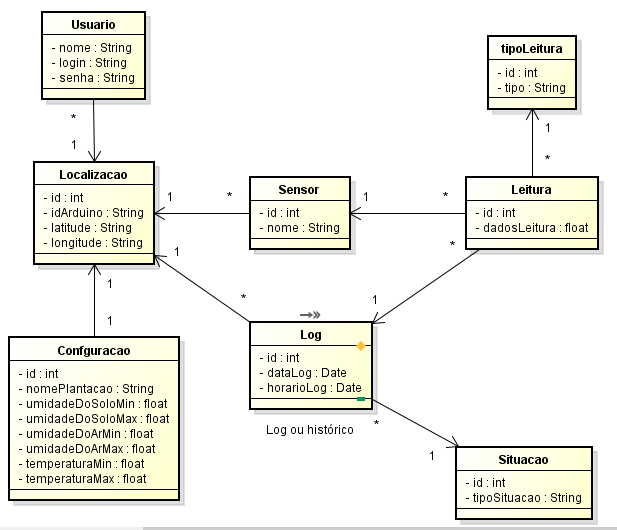


Figura - Diagrama de Classes

Podemos ver uma classe para a manipulação de usuários do sistema, na classe Usuarios, onde o mesmo possui um nome, um *login* e uma senha (Figura 2). Logo em seguida podemos ver uma classe de manipulação das localizações, onde manipulamos a localização de cada um dos conjuntos de sensores, chamada de Localizacao, cada localização recebe uma identificação através do id e suas respectivas informações, que é idArduino, no qual recebe a identificação do sensor em questão e latitude e longitude, que é a posição geográfica desse conjunto de sensores (Figura 3).

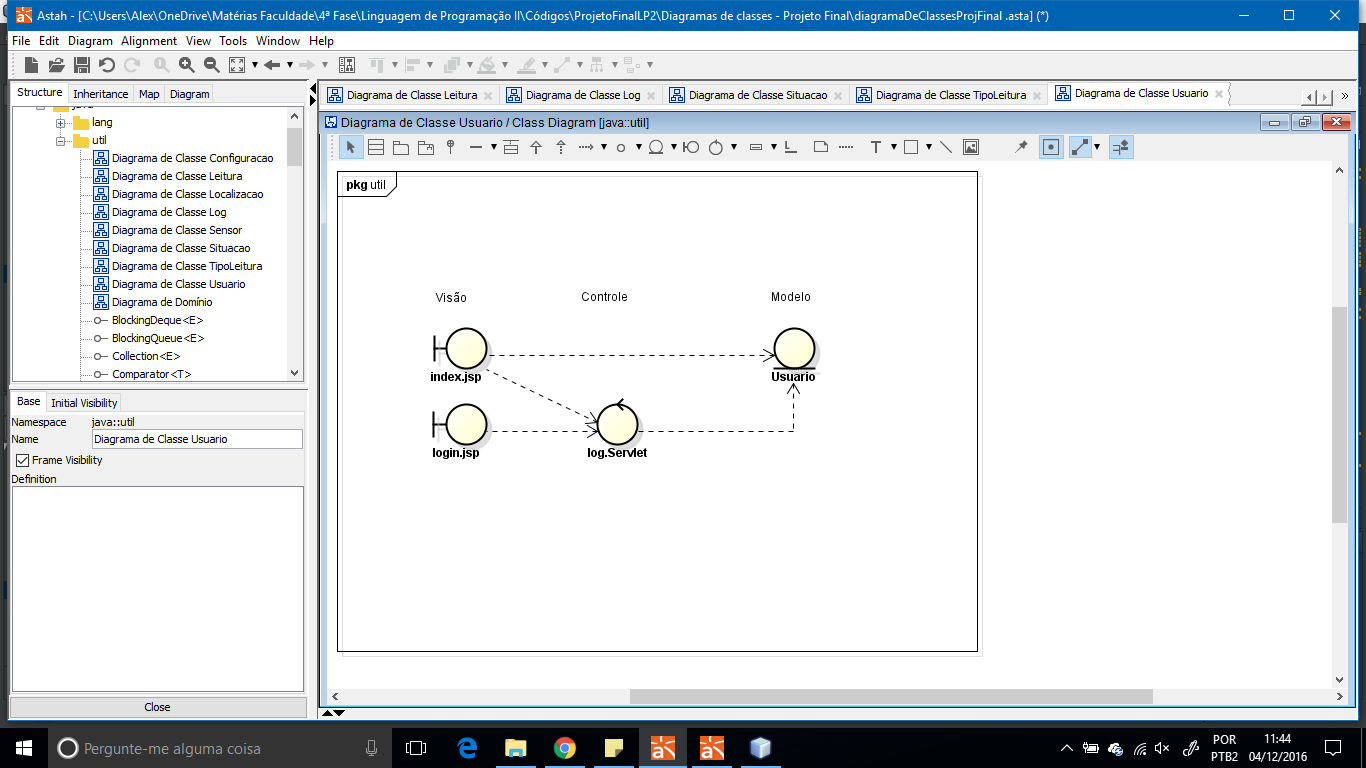


Figura 2 - Diagrama da classe Usuario

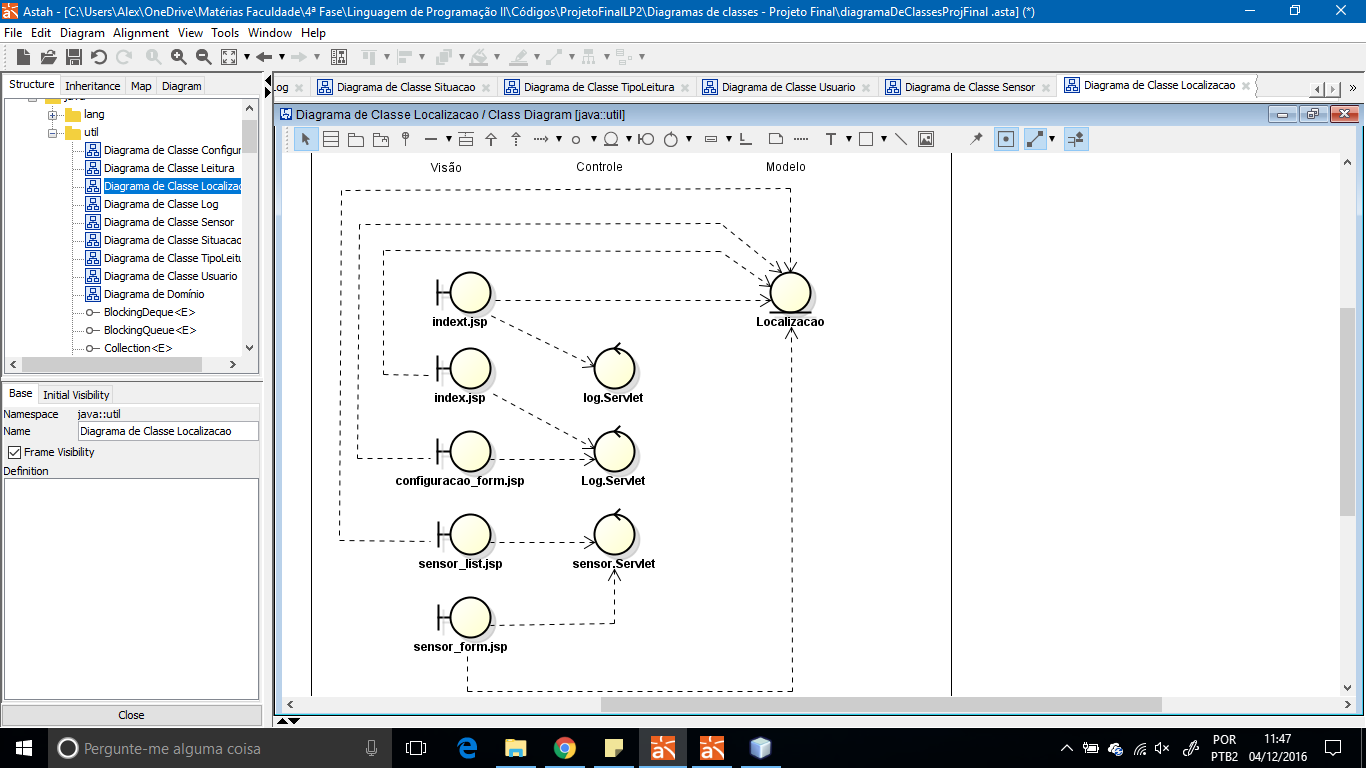


Figura 3 - Diagrama da classe Localizacao

Vemos a classe Configuracao, a qual recebe todas as configurações necessárias para as condições da irrigação, cada uma dessas configurações recebe uma identificação através do id e de nomePlantacao e em seguida as configurações através de umidadeDoSoloMin, umidadeDoSoloMax, umidadeDoArMin, umidadeDoArMax, temperaturaMin e temperaturaMax (Figura 4).

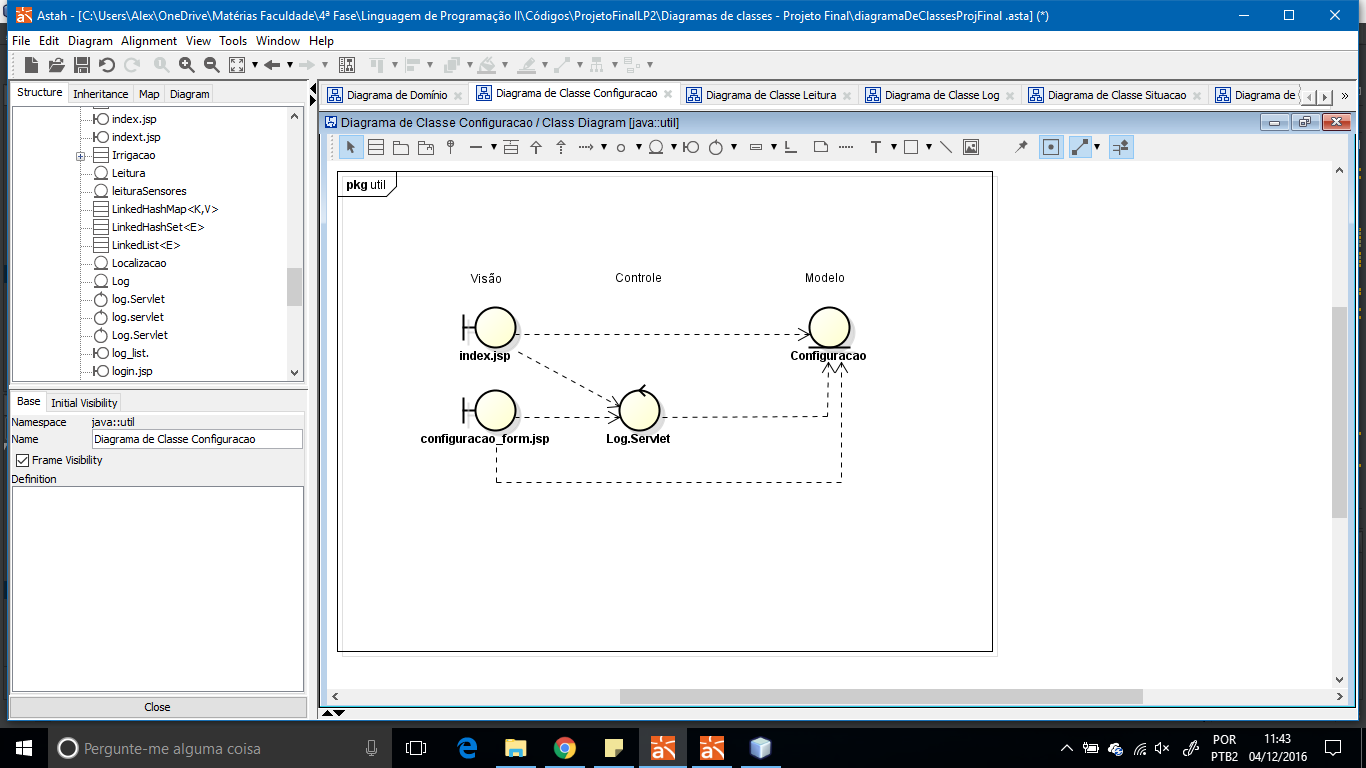


Figura 4 - Diagrama da classe Configuracao

A classe Sensor, somente identifica o sensor os sensores utilizados, no conjunto do Arduino, esses sensores recebem uma identificação através do id e do nome (Figura 5).

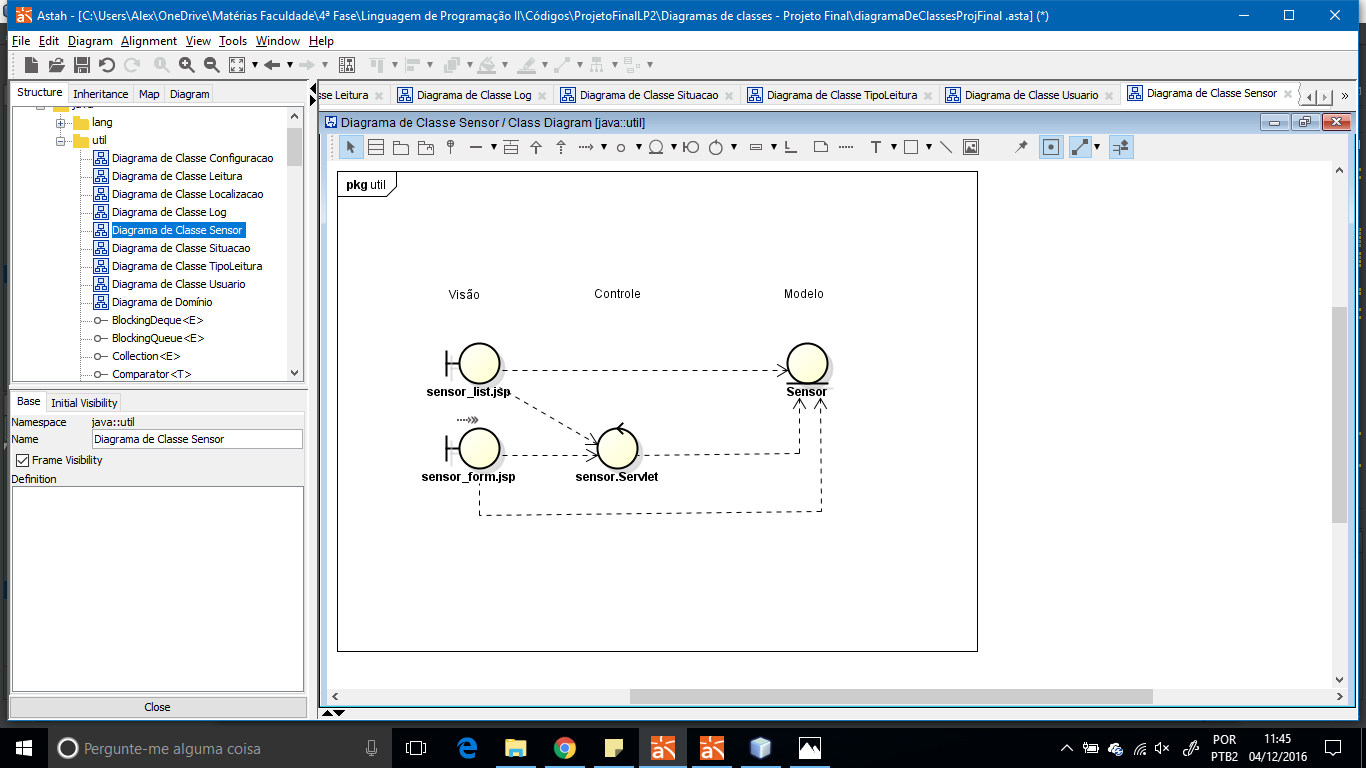


Figura 5 - Diagrama da classe Sensor

A classe tipoLeitura, identifica qual o tipo de leitura efetuada pelo sensor DHT-11, visto que esse sensor identifica tanto humidade do ar como temperatura, então o mesmo estabelece uma identificação para o tipo de leitura e informa qual tipo está sendo realizado pela variável tipo (Figura 6).

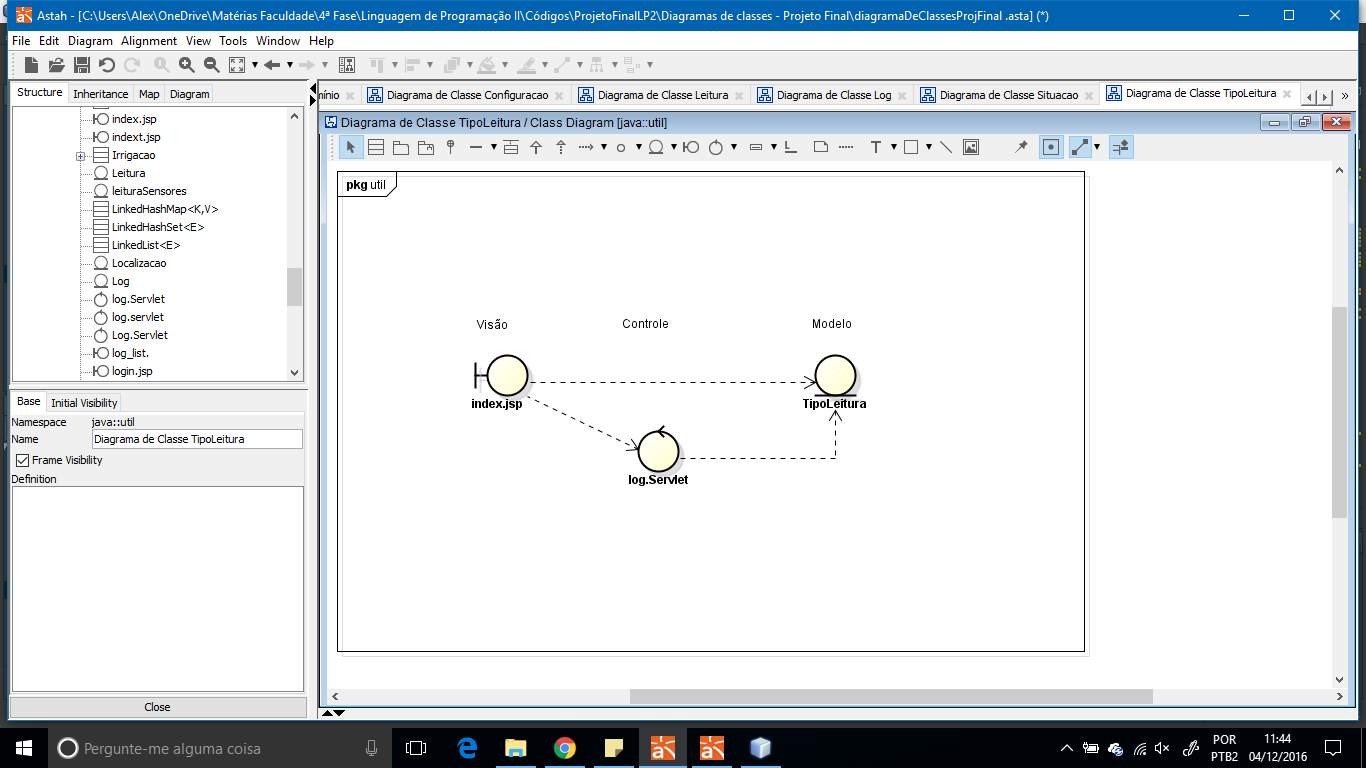


Figura 6 - Diagrama da classe tipoLeitura

A classe Leitura recebe os dados de leitura dos sensores, onde ela identifica as leituras com id e recebe os dados da leitura por dadosLeitura (Figura 7).

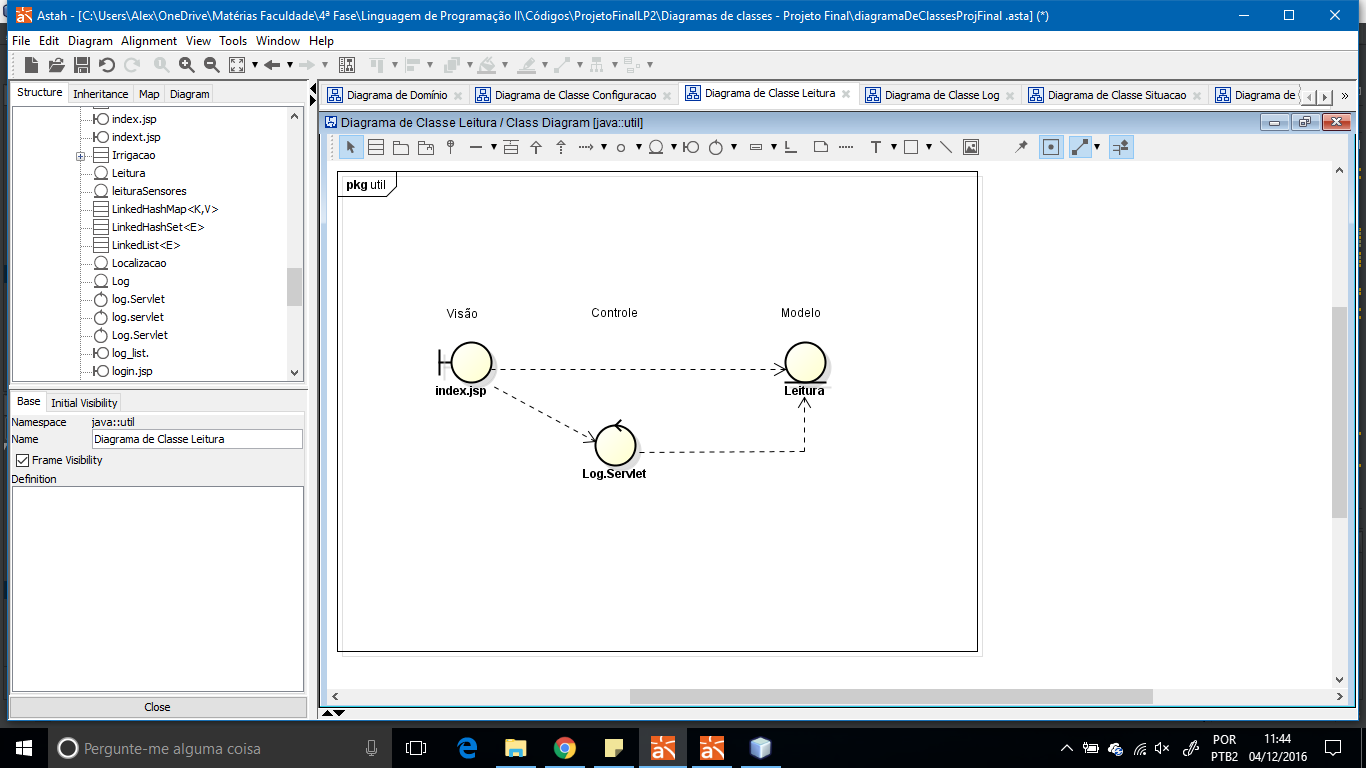


Figura 7 - Diagrama da classe Leitura

A classe Log, fica responsável por manipular o acompanhamento do estado do sistema, recebendo uma identificação de cada log do sistema através do id e ainda atribui a esse log uma data por dataLog e um horário por horarioLog (Figura 8).

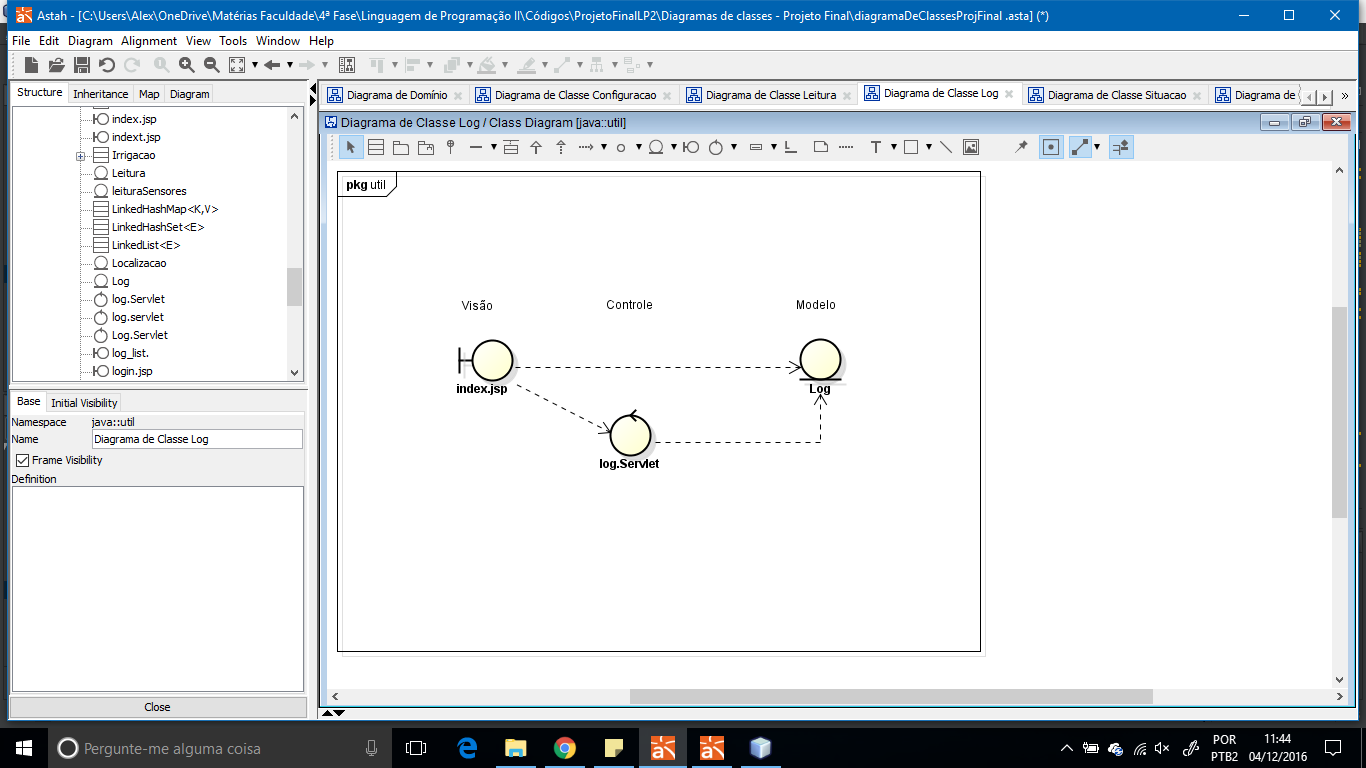


Figura 8 - Diagrama da classe Log

Por fim temos a classe Situação que manipula os dados da situação atual da irrigação, se ela está ocorrendo ou não, cada situação recebe uma identificação por id e em tipoSituação se a irrigação está ocorrendo ou não (Figura 9).

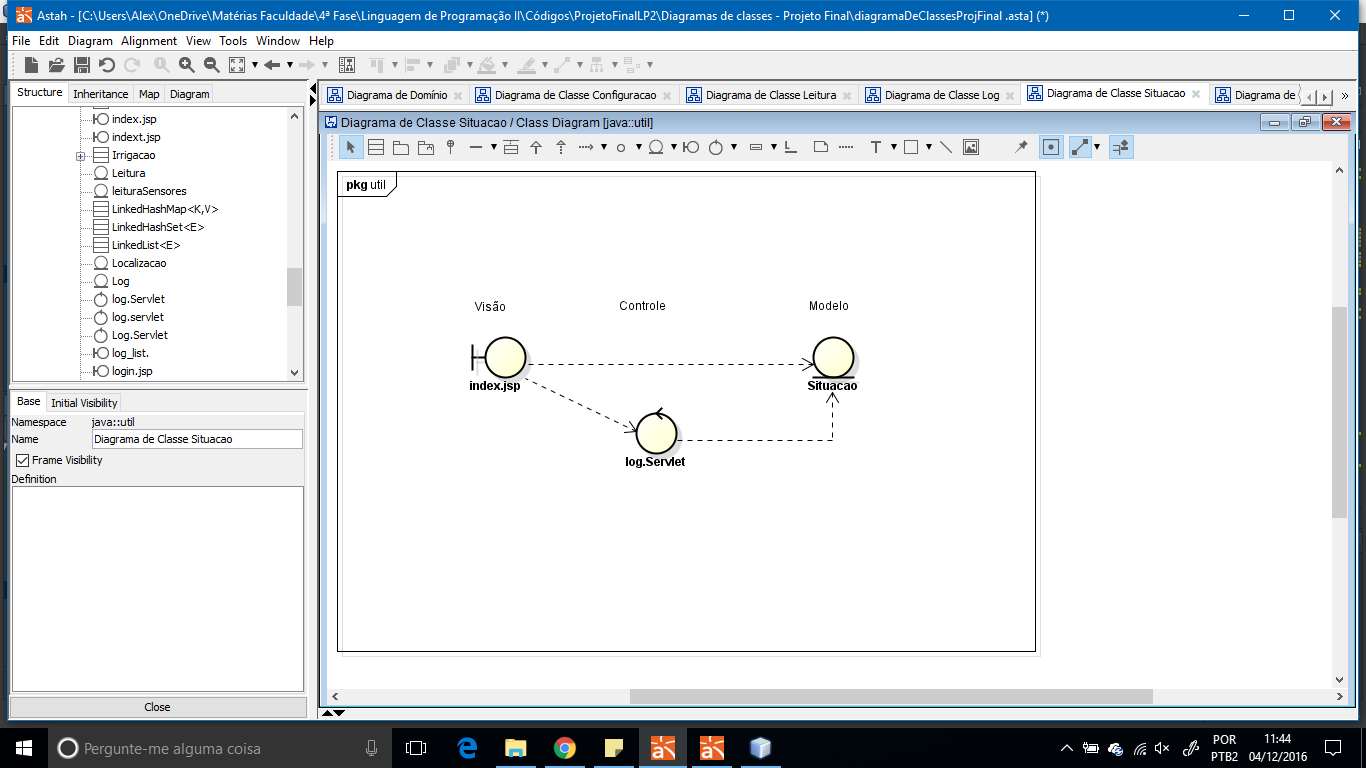


Figura 9 - Diagrama da classe Situacao

Foi criada a classe Arduino, ao qual inicializa a conexão do sistema com o Arduino, essa classe monitora a atividade dos controladores, e suas funções podem ser chamadas por qualquer outra classe do sistema. Nessa classe foram aplicadas as chamadas a biblioteca RXTX, uma biblioteca Java que permite a comunicação serial entre um programa Java e o Arduino através da porta USB da máquina onde roda o programa (DIAS, 2014).

Para o banco de dados do sistema podemos ver as tabelas e utilizadas e suas ligações demonstradas na figura abaixo (Figura 9).

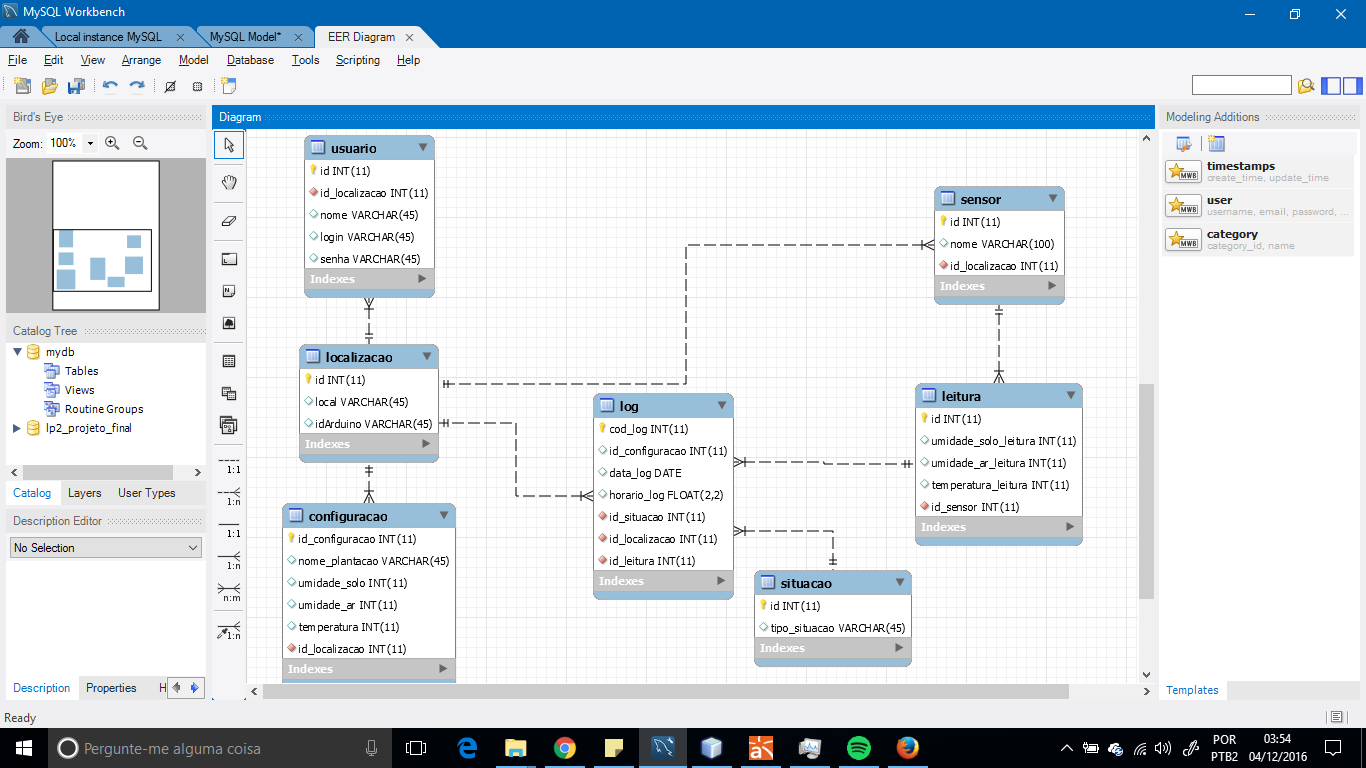


Figura 10 - Digrama de relacionamento de tabelas

1. CONCLUSÕES

A proposta inicial do projeto, foi a confecção de um sistema de irrigação automatizado de baixo custo, utilizando uma controladora Arduino, juntamente com sensores, onde o gerenciamento do sistema seria realizado por uma interface *WEB*, para dessa forma facilitar a atividade dos pequenos agricultores.

De um modo geral, esse trabalho acabou por envolver diversos tópicos visto em todo o curso de Bacharelado em Ciências da Computação, sobre Sistemas Digitais, Banco de Dados e Programação, ainda estendendo os assuntos para a área da agricultura.

Para a conclusão do projeto, houve um grande aprendizado sobre a confecção de páginas para a internet, do funcionamento dos sistemas *WEB*, além de demonstrar a versatilidade da linguagem de programação Java, podemos ver ainda demonstrar o uso de banco de dados em uma situação real.

A criação de um sistema de sensores controlador por Arduino, além da confecção de todo o sistema *WEB*, bem como a utilização de um banco de dados para o armazenamento das informações da irrigação, foram etapas, concluídas com êxito, sem nenhum grande problema que impedisse a sua implementação.

Visto isso, podemos concluir que, é possível criar um sistema de irrigação moderno e com baixo custo em sua confecção e aplicação, graças pelo grande *range* de medições que esses sensores simples possuem, é totalmente possível os adaptar para diversas culturas, em diversas condições, portanto o trabalho foi concluído com êxito.

1. **REFERENCIAS BIBLIOGRAFICAS**

COELHO, Eugênio Ferreira et al. **Sistemas e Manejo de Irrigação de Baixo Custo para Agricultura Familiar Sistemas e Manejo de Irrigação de Baixo Custo para Agricultura Familiar.**2014. Disponível em: <http://ainfo.cnptia.embrapa.br/digital/bitstream/item/133043/1/Cartilha-Manejo-Irrigacao-03-09-2015.pdf>. Acesso em: 02 dez. 2016.

CARVALHO, Renon Steinbach. **Sistema de Irrigação Automatizado.**2013. 12 f. Ifc - Rio do Sul, Rio do Sul, 2013.

TWITTER (Estados Unidos) (Org.). **Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.**2016. Disponível em: <http://getbootstrap.com/>. Acesso em: 02 dez. 2016.

SESHADRI, Govind. **Understanding JavaServer Pages.**1999. Disponível em: <http://www.javaworld.com/article/2076557/java-web-development/understanding-javaserver-pages-model-2-architecture.html>. Acesso em: 02 dez. 2016.

ORACLE (Estados Unidos). **Java SE 7 Features and Enhancements.**2013. Disponível em: <http://www.oracle.com/technetwork/java/javase/jdk7-relnotes-418459.html>. Acesso em: 02 dez. 2016.

ORACLE (Estados Unidos). **NetBeans Release Roadmap.**2016. Disponível em: <https://netbeans.org/community/releases/roadmap.html>. Acesso em: 02 dez. 2016.

TOMCAT FOUNDATION (Estados Unidos). **Apache Tomcat Configuration Reference.**2013. Disponível em: <http://tomcat.apache.org/tomcat-5.5-doc/config/realm.html>. Acesso em: 02 dez. 2016.

ORACLE (Estados Unidos). **MySQL 8.0 Release Notes.**2016. Disponível em: <http://dev.mysql.com/doc/relnotes/mysql/8.0/en/>. Acesso em: 02 dez. 2016.

DIAS, Klauder. **Comunicação Serial Java + Arduino.**2014. Disponível em: <https://www.embarcados.com.br/comunicacao-serial-java-arduino/>. Acesso em: 02 dez. 2016.