



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

RENAN GUERIN

ARDUINO: AUTOMAÇÃO RESIDENCIAL

Assis/SP

2018



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

RENAN GUERIN

ARDUINO: AUTOMAÇÃO RESIDENCIAL

Projeto de pesquisa apresentado ao curso de Análise e desenvolvimento de sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Renan Guerin

Orientador(a): Diomara Martins Reigato
Barros

**Assis/SP
2018**

FICHA CATALOGRÁFICA

Guerin, Renan.

Arduino:Automação Residencial /Renan Guerin. Fundação Educacional do Município de Assis –FEMA – Assis, 2018.

Número de páginas: 45

Orientador: Ms. Diomara Martins Reigato Barros

1. Arduino. 2. Automação

CDD:005.133

ARDUINO: AUTOMAÇÃO RESIDENCIAL

RENAN GUERIN

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____ Diomara Martins Reigato Barros _____

Examinador: _____ Alex Sandro Romeo de Souza Poletto _____

Assis/SP

2018

DEDICATÓRIA

Dedico esse trabalho primeiramente à minha família, pois eles me deram a condição e para a realização desse trabalho, e por fim, amigos e colegas que me apoiaram.

AGRADECIMENTOS

Agradeço à minha família que me deu condições e me apoiou, além de me proporcionar bons conselhos para realizar esse trabalho.

Aos meus amigos e colegas que ajudaram e me apoiaram para a realização deste trabalho, especialmente Matheus de Freitas Pires, que me emprestou o notebook para a apresentação

À Ma. Diomara Martins Reigato Barros, minha orientadora, pela sua fundamental contribuição no desenvolvimento do trabalho desde o início.

RESUMO

Esse projeto visa o desenvolvimento de uma automação residencial de baixo custo para todos os tipos de classe, por meio de um sistema de gerenciamento web, e utilizando um Arduino BlackBoard V1.0 e um Ethernet Shield.

Será utilizada a linguagem ARDUINO IDE para a programação do Arduino, a linguagem PHP será utilizada para o sistema de gerenciamento web e a automatização será mostrada em uma maquete.

Para a modelagem serão utilizadas ferramentas como Freemind e Astah, sendo de fácil acesso e gratuito, desenvolvendo assim, diagramas e o mapa mental.

ABSTRACT

This project aims to develop a low-cost residential automation through a web management system and using an Arduino BlackBoard V1.0 and an Ethernet Shield.

The ARDUINO IDE language will be used for Arduino programming, the PHP language will be used for the web management system and the automation will be shown in a mockup.

For the modeling will be used tools like Freemind and Astah, being easy to access and free, thus developing diagrams and the mental map.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1: IoT, Internet das coisas (Foto: Reprodução/IoT)..... | 13 |
| Figura 2: Smarth House, Casa Automatizada..... | 13 |
| Figura 3: Mapa Mental..... | 20 |
| Figura 4: Diagrama Caso de Uso..... | 22 |
| Figura 5: Diagrama Caso de Uso Admin..... | 22 |
| Figura 6: Caso de Uso Efetuar Login..... | 23 |
| Figura 7: Caso de Uso Ligar Luz..... | 24 |
| Figura 8: Caso de Uso Desligar Luz..... | 25 |
| Figura 9: Caso de Uso Ligar Portão Elétrico..... | 26 |
| Figura 10: Caso de Uso Desligar Portão Elétrico..... | 27 |
| Figura 11: Caso de Uso Ligar Ventilação..... | 28 |
| Figura 12: Caso de Uso Desligar Ventilação..... | 29 |
| Figura 13: Caso de Uso Ligar Irrigação..... | 29 |
| Figura 14: Caso de uso Desligar Irrigação..... | 30 |
| Figura 15: Caso de Uso Admin Efetuar Login..... | 31 |
| Figura 16: Caso de Uso Admin Cadastrar Cliente..... | 32 |
| Figura 17: Caso de Uso Admin Editar Cliente..... | 32 |
| Figura 18: Caso de Uso Admin: Remover Cliente..... | 33 |
| Figura 19: Diagrama de Classe..... | 35 |
| Figura 20: Diagrama de Classe Admin..... | 35 |
| Figura 21: Diagrama de Atividade..... | 36 |
| Figura 22: Diagrama de Atividade Admin..... | 36 |
| Figura 23: Diagrama de Sequência..... | 37 |
| Figura 24: Diagrama de Sequência Admin..... | 37 |
| Figura 25: Diagrama ER..... | 38 |
| Figura 26: Diagrama ER Admin..... | 38 |
| Figura 27: EAP..... | 39 |
| Figura 28: Código Arduino e Ethernet Shield..... | 40 |
| Figura 29: Conexão Arduino e PHP..... | 41 |
| Figura 30: Conexão Arduino e PHP 2..... | 41 |
| Figura 31: Função md5..... | 42 |

SUMÁRIO

| | |
|---|-----------|
| 1. INTRODUÇÃO | 12 |
| 1.1. Objetivos | 14 |
| 1.2. Público alvo | 14 |
| 1.3. Justificativa | 14 |
| 2. DESENVOLVIMENTO DO PROJETO | 15 |
| 2.1. Uml | 15 |
| 2.1.1. Diagrama de caso de uso..... | 15 |
| 2.1.2. Diagrama de classes | 16 |
| 2.1.3. Diagrama de atividade..... | 16 |
| 2.1.4. Diagrama de sequência..... | 16 |
| 2.1.5. Diagrama de entidade e relacionamento | 16 |
| 2.2. Tecnologias usadas no desenvolvimento | 17 |
| 2.2.1. Arduino IDE | 17 |
| 2.2.2. PHP | 17 |
| 2.2.3. HTML 5..... | 18 |
| 2.2.3. CSS | 19 |
| 2.2.4. Javascript | 19 |
| 2.2.5. Brackets | 19 |
| 3. ANÁLISE, ESPECIFICAÇÃO DO SISTEMA E PROJETO DO ESTUDO DE CASO | 20 |
| 3.1. Mapa Mental..... | 20 |
| 3.2. Lista de eventos | 21 |
| 3.3. Diagrama caso de uso..... | 22 |
| 3.4. Especificações de caso de uso | 23 |
| 3.4.1. Caso de uso Efetuar Login | 23 |
| 3.4.2. Caso de uso ligar Luz..... | 24 |
| 3.4.3. Caso de uso desligar Luz | 25 |
| 3.4.4. Caso de Uso ligar Portão Elétrico | 26 |
| 3.4.5. Caso de uso desligar Portão Elétrico..... | 26 |
| 3.4.6 Caso de Uso Ligar Ventilação | 28 |
| 3.4.7. Caso de uso desligar Ventilação | 29 |
| 3.4.8. Caso de Uso Ligar Irrigação..... | 29 |

| | |
|---|-----------|
| 3.4.9. Caso de Uso Desligar Irrigação..... | 30 |
| 3.5. Caso de uso Admin: Efetuar Login..... | 31 |
| 3.5.1. Caso de uso Admin: Cadastrar Cliente | 32 |
| 3.5.2 Caso de Uso Admin: Editar Cliente | 32 |
| 3.5.3. Caso de uso Admin: Remover Cliente | 33 |
| 3.6. Diagrama de classe..... | 35 |
| 3.7. Diagrama de atividade..... | 36 |
| 3.8. Diagrama de sequência..... | 37 |
| 3.9. Diagrama de entidade e relacionamento..... | 38 |
| 4. ESTRUTURA ANALÍTICA DO PROJETO (EAP)..... | 39 |
| 5. IMPLEMENTAÇÃO..... | 40 |
| 5.1. Configuração Arduino e ethernet Shield..... | 40 |
| 5.2. Conexão Arduino e PHP | 41 |
| 5.3. Segurança | 42 |
| 6. CONCLUSÃO | 43 |
| 7. REFERÊNCIAS..... | 44 |

1. INTRODUÇÃO

Com o grande crescimento da tecnologia, as pessoas continuam cada vez mais exigentes quando o assunto é conforto, comodidade e segurança residencial. É neste sentido que a automação residencial ganha espaço atualmente, pois ela possibilita ao público controlar vários equipamentos remotamente, atendendo suas necessidades.

Apesar da automação residencial estar associada, hoje, a um público de classe alta, nada impede, que soluções sejam implementadas em médias e pequenas residências, além disso, a automação residencial tem um custo variável, alternando de acordo com o projeto desejado.

Perante o exposto, esse projeto visa o desenvolvimento de uma automação residencial de baixo custo feita no Arduino IDE, por meio de um sistema de gerenciamento web, e utilizando um Arduino BlackBoard V1.0 e um Ethernet Shield. Cumprindo assim a proposta de atender as necessidades do usuário.

Internet das Coisas: Entenda o conceito

Quando falamos da evolução na tecnológica, a noção da Internet das Coisas, é um dos principais assuntos.

Villarino diz que a IoT é o modo como as coisas estão conectadas e se comunicam entre si e com o usuário, através de sensores inteligentes e softwares que transmitem dados para uma rede. Como se fosse um grande sistema nervoso que possibilita a troca de informações entre dois ou mais pontos.

Esse conceito de automação vem no seguimento de encontrar soluções que à necessidade do homem de querer realizar o mínimo esforço em suas atividades. Apesar de ainda ser pouco conhecida e divulgada, a domótica pode se tornar popular em um futuro próximo.

1.1. Objetivos

O objetivo desse projeto é o desenvolvimento de um sistema de baixo custo, utilizando um sistema de gerenciamento via web que interage com diversos equipamentos e dispositivos automatizados em uma maquete, onde vai ser demonstrando o controle sobre sistemas de iluminação, alarme de segurança inteligente, irrigação, ventilador e portão elétrico podendo ser gerenciado por qualquer dispositivo conectado à internet, além de estar conectado a um banco de dados.

1.2. Público alvo

A automatização foi desenvolvida para os consumidores que procuram mais conforto, comodidade e facilidade em suas tarefas, não importando a sua classe, pois, a automação residencial tem um custo variável, alternando de acordo com o projeto desejado.

1.3. Justificativa

Devido à crescente busca por tempo, comodidade, praticidade, segurança, além de um bom custo-benefício, esse sistema busca atender as necessidades solicitadas pelo cliente, atendendo a qualquer tipo de público.

2. DESENVOLVIMENTO DO PROJETO

Para o desenvolvimento deste trabalho será utilizado como ferramenta de análise a linguagem UML (UNIFIED MODELING LANGUAGE) com seus devidos diagramas. Para o desenvolvimento do sistema, será utilizada a linguagem do Arduino IDE.

Para a modelagem serão utilizadas ferramentas como Freemind e Astah, sendo de fácil acesso e gratuito, desenvolvendo assim, diagramas e o mapa mental.

O banco de dados utilizado será o MySql.

2.1. Uml

UML (Unified Modeling Language) é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos, cujo o objetivo é auxiliar na definição das características do sistema, tais como seus requisitos, seu comportamento, estrutura lógica, a dinâmica de seus processos até mesmo suas necessidades físicas em relação ao equipamento sobre a qual o sistema deve ser implantado (GUEDES,2011, p.19).

2.1.1. Diagrama de caso de uso

O Diagrama de Caso de Uso é utilizado normalmente nas fases de análise e levantamento de requisitos do sistema, embora venha ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar. Procura identificar os atores (usuários, outros sistemas ou até mesmo algum hardware especial) que utilizarão de alguma forma o software, bem como os serviços, ou seja, as funcionalidades que o sistema disponibilizará aos atores, conhecidos como caso de uso. (GUEDES,2011, p.30).

2.1.2. Diagrama de classes

O Diagrama de Classes, define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si. É considerado um dos mais importantes da UML e serve de apoio para a maioria dos demais diagramas. (GUEDES,2011, p.31).

2.1.3. Diagrama de atividade

O Diagrama de Atividade descreve os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. Concentra-se na representação do fluxo de controle de uma atividade. (GUEDES,2011, p.36).

2.1.4. Diagrama de sequência

O Diagrama de Sequência é um diagrama comportamental que se preocupa com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Costuma identificar o evento gerador do processo modelado, bem como o autor responsável por esse evento, e determina como o processo deve se desenrolar a ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos. (GUEDES,2011, p.36).

2.1.5. Diagrama de entidade e relacionamento

Segundo GUEDES, O modelo de entidade e relacionamento (MER) é usado para representar o modelo conceitual do negócio e de alto nível, ou seja, é um conjunto de conceitos e elementos de modelagem que o projetista do banco de dados precisa conhecer. (GUEDES,2011)

2.2. Tecnologias usadas no desenvolvimento

Nessa seção serão apresentadas as tecnologias que foram utilizados para o desenvolvimento do projeto, sendo elas, Arduino IDE, PHP, HTML 5, CSS, JAVASCRIPT e o editor de texto Brackets.

2.2.1. Arduino IDE

O ambiente de desenvolvimento do Arduino torna fácil escrever o código e enviar para a placa de entrada e saída. Ele funciona em Windows, Mac OS X e Linux. O ambiente de programação é escrito em Java e baseado em Processing, avr-gcc e outros softwares de código livre.

2.2.2. PHP

O PHP é uma linguagem open-source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web. O site becode mostra as seguintes características do PHP:

Custo Reduzido

Uma das principais vantagens do PHP é o fato dele necessitar de menos investimento em sua infraestrutura. Isso significa que os gastos com servidor e implementação normalmente são mais baixos do que em outras linguagens, dessa forma, manutenções em sites feitos em PHP, tendem a ser mais baratos do que em outras linguagens.

Linguagem estabelecida

PHP tem a vantagem de estar no mercado a muito tempo e existir a mais de 20 anos. Isso faz com que se tenha bastante desenvolvedores no mercado, além de ter muito material e soluções disponíveis.

Facilidade de aprendizado

PHP é uma das linguagens mais indicadas para iniciantes do mercado de desenvolvimento de software, sendo fácil, e não tão difícil de aprender.

2.2.3. HTML 5

Segundo o site developer mozilla, onde se encontra o artigo “**HTML5**”, escrito pelo usuário FLR e outros colaboradores, o HTML5 se define e é caracterizada por:

HTML5 é a mais recente evolução do padrão que define o HTML. O termo representa dois conceitos diferentes:

- É uma nova versão da linguagem HTML, com novos elementos, atributos e comportamentos.
- e um conjunto maior de tecnologias que permite o desenvolvimento de aplicações e web sites mais diversos e poderosos. Este conjunto é chamado HTML5 & friends e muitas vezes abreviado apenas como HTML5.

Criada para ser utilizável por todos os desenvolvedores da Web Aberta, essa página de referências faz ligações a inúmeros recursos do HTML5, classificados em diversos grupos, baseando-se em suas funções;

- Semântica: permite você descrever mais precisamente o seu conteúdo.
- Conectividade: permite uma comunicação com o servidor de formas modernas e inovadoras.
- Offline e armazenamento: Permite que páginas web armazenem dados localmente do lado do cliente e opere de forma offline mais eficientemente.
- Multimídia: Viabiliza a utilização de áudio e vídeo de forma primorosa na Web Aberta.
- Gráficos e efeitos 2D/3D: viabiliza um leque diversificado de opções de representação gráfica.
- Performance e integração: fornece grande otimização de velocidade e melhor utilização do hardware do computador.
- Acesso ao dispositivo: viabiliza a utilização de diversos métodos e dispositivos de entrada e saída.
- Estilização: permite aos autores a escrita de temas mais sofisticados.

2.2.3. CSS

Segundo Pereira, o CSS é uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML). O CSS define como serão exibidos os elementos contidos no código de uma página da internet e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento.

2.2.4. Javascript

O JavaScript é uma poderosa linguagem que deve ser dominada por quem deseja criar páginas Web dinâmicas e interativas.

Marcondes diz que "Com o JavaScript podemos criar efeitos especiais para nossas páginas na Web, além de podermos proporcionar uma maior interatividade com os usuários. O JavaScript é uma linguagem orientada a objetos, ou seja, ela trata todos os elementos da página como objetos distintos, facilitando a tarefa da programação".

2.2.5. Brackets

Brackets é um software para criação e edição de arquivos (IDE) HTML, CSS e JS. Através de configuração também é possível trabalhar com PHP.

Com ele você pode criar o seu site e testar o que estiver fazendo. O Brackets ainda possui um instalador de extensões e com ele você poderá aumentar e personalizar os recursos.

O Brackets é um open source da Adobe e pode ser baixado e utilizado livremente, e em português.

3. ANÁLISE, ESPECIFICAÇÃO DO SISTEMA E PROJETO DO ESTUDO DE CASO

Para o desenvolvimento do sistema proposto, foi utilizado a tecnologia UML (Linguagem de Modelagem Unificada), toda a parte gráfica foi criada com base nas ferramentas Astah e FreeMind.

3.1. Mapa Mental

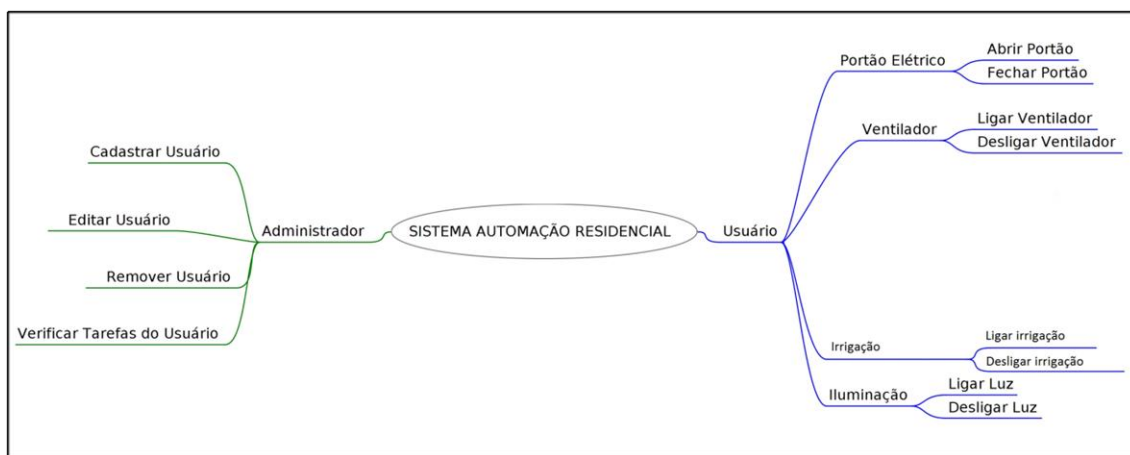


Figura 3: Mapa Mental

Fonte: Autoria própria.

3.2. Lista de eventos

Efetuar o Login;

Ligar iluminação;

Desligar iluminação;

Ligar ventilação;

Desligar ventilação;

Abrir portão-elétrico;

Fechar portão-elétrico;

Ligar alarme;

Desligar alarme;

Cadastrar Cliente;

Verificar tarefas do cliente;

Editar Clientes;

Remover Clientes;

3.3. Diagrama caso de uso

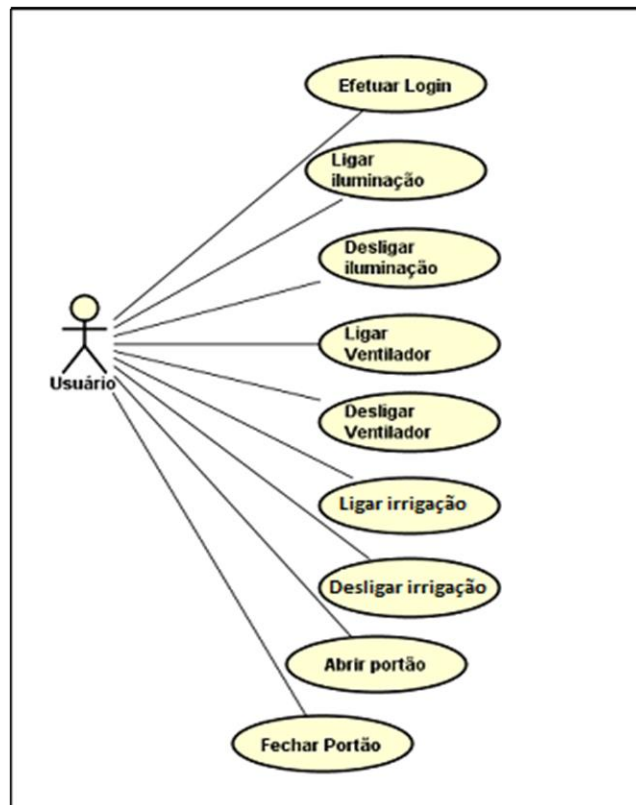


Figura 4: Diagrama Caso de Uso

Fonte: Autoria própria.

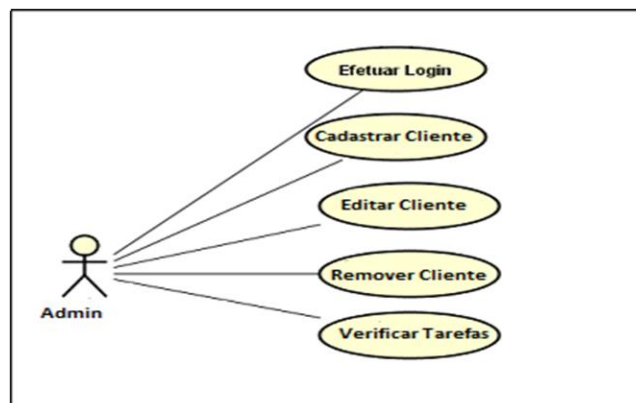


Figura 5: Diagrama Caso de Uso Admin

Fonte: Autoria própria.

3.4. Especificações de caso de uso

Especificações de casos de uso são narrativas em texto, e são muito utilizados para representar requisitos funcionais nos sistemas. Os diagramas de Casos de Uso são representações gráficas dos Casos de Uso e seus relacionamentos com outros casos de uso e atores.

A seguir são apresentados os diagramas de caso de uso individuais e suas especificações.

3.4.1. Caso de uso Efetuar Login

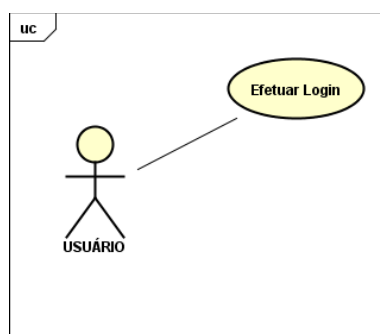


Figura 6: Caso de Uso Efetuar Login

Fonte: Autoria própria.

Efetuar Login

1.Finalidade/objetivo:

- Permite ao usuário efetuar o login no sistema.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter um login e uma senha de usuário, além de estar conectado a rede.

4.Fluxo Principal:

- O usuário solicita para que o sistema efetue o login na página.
- O sistema verifica os dados e efetua o login na página.

3.4.2. Caso de uso ligar Luz

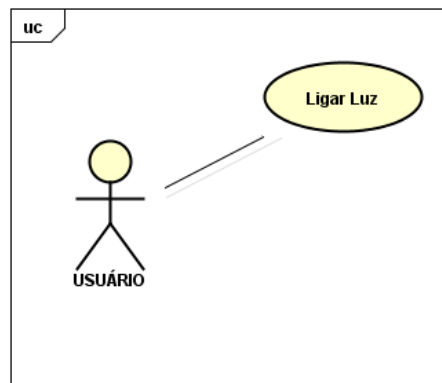


Figura 7: Caso de Uso Ligar Luz

Fonte: Autoria própria

Ligar Luz

1.Finalidade/objetivo:

- Permite ao usuário ligar a luz da residência.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter conexão com a internet e ter efetuado o login no sistema.

4.Fluxo Principal:

- O usuário solicita para que se ligue a luz da residência.
- O sistema liga a luz.

3.4.3. Caso de uso desligar Luz

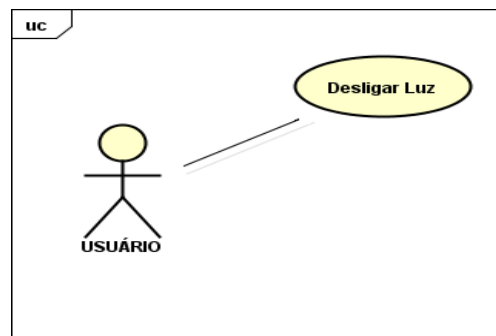


Figura 8: Caso de Uso Desligar Luz

Fonte: Autoria própria.

Desligar Luz

1.Finalidade/objetivo:

- Permite ao usuário desligar a luz da residência.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter conexão com a internet e ter efetuado o login no sistema.

4.Fluxo Principal:

- O usuário solicita para que se desligue a luz da residência.
- O sistema desliga a luz.

3.4.4. Caso de Uso ligar Portão Elétrico

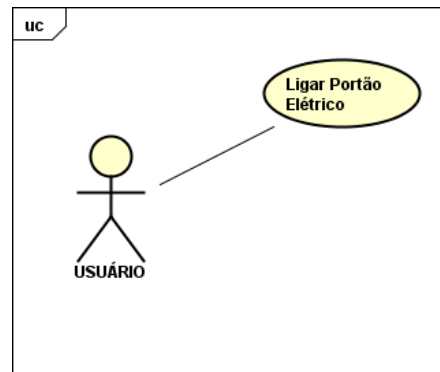


Figura 9: Caso de Uso Ligar Portão Elétrico

Fonte: Autoria própria.

Ligar Portão Elétrico

1.Finalidade/objetivo:

- Permite ao usuário ligar o Portão Elétrico da residência.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter conexão com a internet e ter efetuado o login no sistema.

4.Fluxo Principal:

- O usuário solicita para que se ligue o Portão Elétrico da residência.
- O sistema liga o Portão Elétrico.

3.4.5. Caso de uso desligar Portão Elétrico

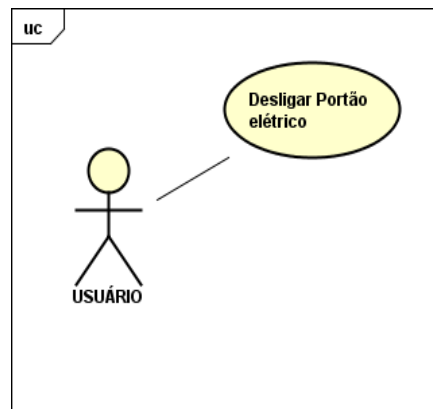


Figura 10: Caso de Uso Desligar Portão Elétrico

Fonte: Autoria própria.

Desligar Portão Elétrico

1.Finalidade/objetivo:

- Permite ao usuário desligar o Portão Elétrico da residência.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter conexão com a internet e ter efetuado o login no sistema.

4.Fluxo Principal:

- O usuário solicita para que se desligue o Portão Elétrico da residência.
- O sistema desliga o Portão Elétrico.

3.4.6 Caso de Uso Ligar Ventilação

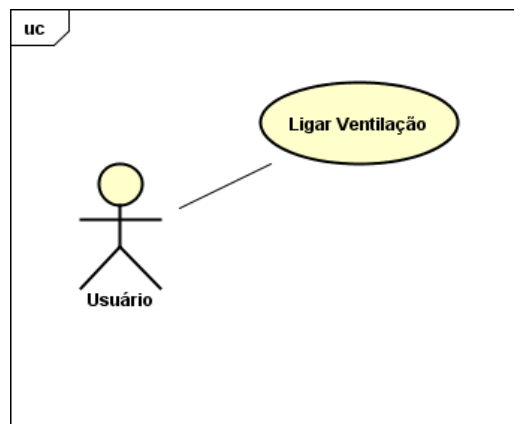


Figura 11: Caso de Uso Ligar Ventilação

Fonte: Autoria própria.

Ligar Ventilação

1.Finalidade/objetivo:

- Permite ao usuário Ligar a Ventilação da residência.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter conexão com a internet e ter efetuado o login no sistema.

4.Fluxo Principal:

- O usuário solicita para que se ligue a ventilação da residência.
- O sistema liga a ventilação.

3.4.7. Caso de uso desligar Ventilação

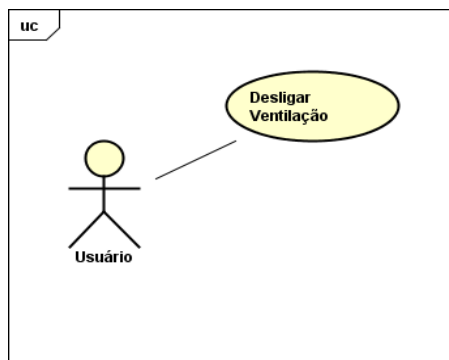


Figura 12: Caso de Uso Desligar Ventilação

Fonte: Autoria própria.

Desligar Ventilação

1.Finalidade/objetivo:

- Permite ao usuário Desligar a Ventilação da residência.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter conexão com a internet e ter efetuado o login no sistema.

4.Fluxo Principal:

- O usuário solicita para que se desligue a ventilação da residência.

- O sistema desliga a ventilação.

3.4.8. Caso de Uso Ligar Irrigação

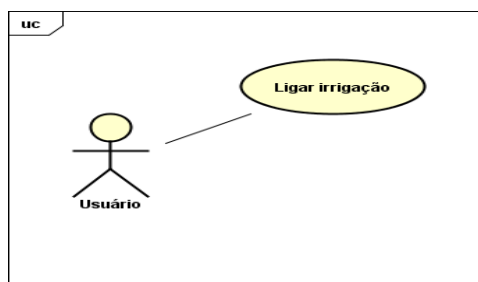


Figura 13: Caso de Uso Ligar Irrigação

Fonte: Autoria própria.

Ligar Irrigação

1.Finalidade/objetivo:

- Permite ao usuário Ligar a Irrigação da residência.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter conexão com a internet e ter efetuado o login no sistema.

4.Fluxo Principal:

- O usuário solicita para que se ligue a Irrigação da residência.
- O sistema liga a irrigação.

3.4.9. Caso de Uso Desligar Irrigação

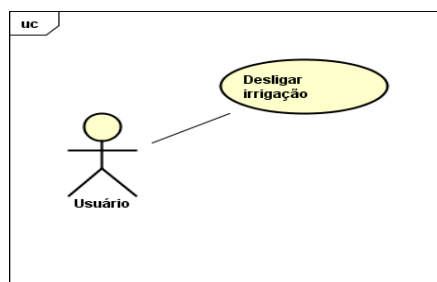


Figura 14: Caso de uso Desligar Irrigação

Fonte: Autoria própria.

Desligar Irrigação

1.Finalidade/objetivo:

- Permite ao usuário Desligar a Irrigação da residência.

2.Ator:

- Usuário.

3.Pré-condições:

- O usuário precisa ter conexão com a internet e ter efetuado o login no sistema.

4.Fluxo Principal:

- O usuário solicita para que se desligue a irrigação da residência.
- O sistema desliga a irrigação.

3.5. Caso de uso Admin: Efetuar Login

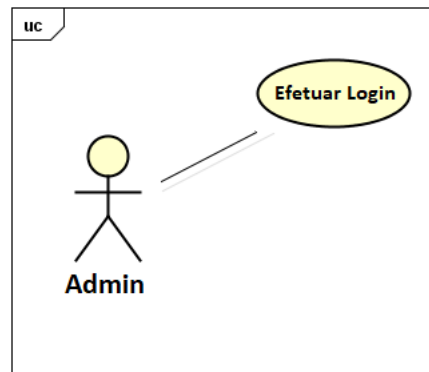


Figura 15: Caso de Uso Admin Efetuar Login

Fonte: Autoria própria.

Efetuar Login

1.Finalidade/objetivo:

- Permite ao administrador efetuar o login no sistema administrador

2.Ator:

- Administrador.

3.Pré-condições:

- O administrador precisa ter conexão com a internet e ter um login e uma senha de administrador

4.Fluxo Principal:

- O administrador solicita para que o sistema efetue o login na página.
- O sistema verifica os dados e efetua o login na página.

3.5.1. Caso de uso Admin: Cadastrar Cliente

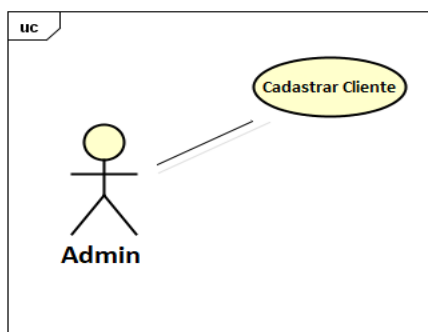


Figura 16: Caso de Uso Admin Cadastrar Cliente

Fonte: Autoria própria

Cadastrar Cliente

1.Finalidade/objetivo:

- Permite ao administrador cadastrar um cliente no sistema administrador

2.Ator:

- Administrador.

3.Pré-condições:

- O administrador precisa ter feito o login no sistema e ter uma conexão a rede.

4.Fluxo Principal:

- O administrador solicita para que o sistema cadastre um usuário.
- O sistema cadastra o usuário solicitado.

3.5.2 Caso de Uso Admin: Editar Cliente

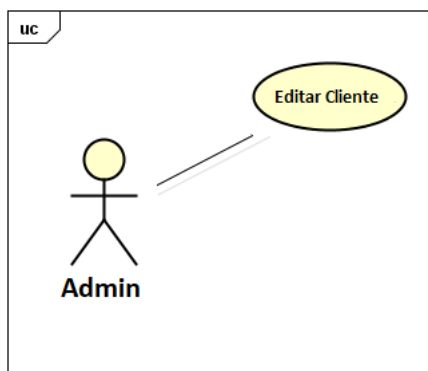


Figura 17: Caso de Uso Admin Editar Cliente

Fonte: Autoria própria.

Editar Cliente

1.Finalidade/objetivo:

- Permite ao administrador editar um cliente já cadastrado no sistema administrador

2.Ator:

- Administrador.

3.Pré-condições:

- O administrador precisa fazer login no sistema, ter pelo menos um cliente cadastrado e estar conectado à rede.

4.Fluxo Principal:

- O administrador solicita para que o sistema edite um usuário do sistema.
- O sistema edita os dados do cliente solicitado.

3.5.3. Caso de uso Admin: Remover Cliente

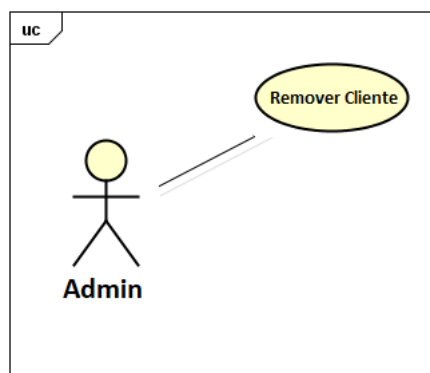


Figura 18: Caso de Uso Admin: Remover Cliente

Fonte: Autoria própria

Remover Cliente

1.Finalidade/objetivo:

- Permite ao administrador remover um cliente no sistema administrador.

2.Ator:

- Administrador.

3.Pré-condições:

- O administrador precisa fazer login no sistema, ter pelo menos um cliente cadastrado e estar conectado à rede.

4.Fluxo Principal:

- O administrador solicita para que se remova um cliente do sistema.
- O sistema remove o cliente solicitado.

3.6. Diagrama de classe

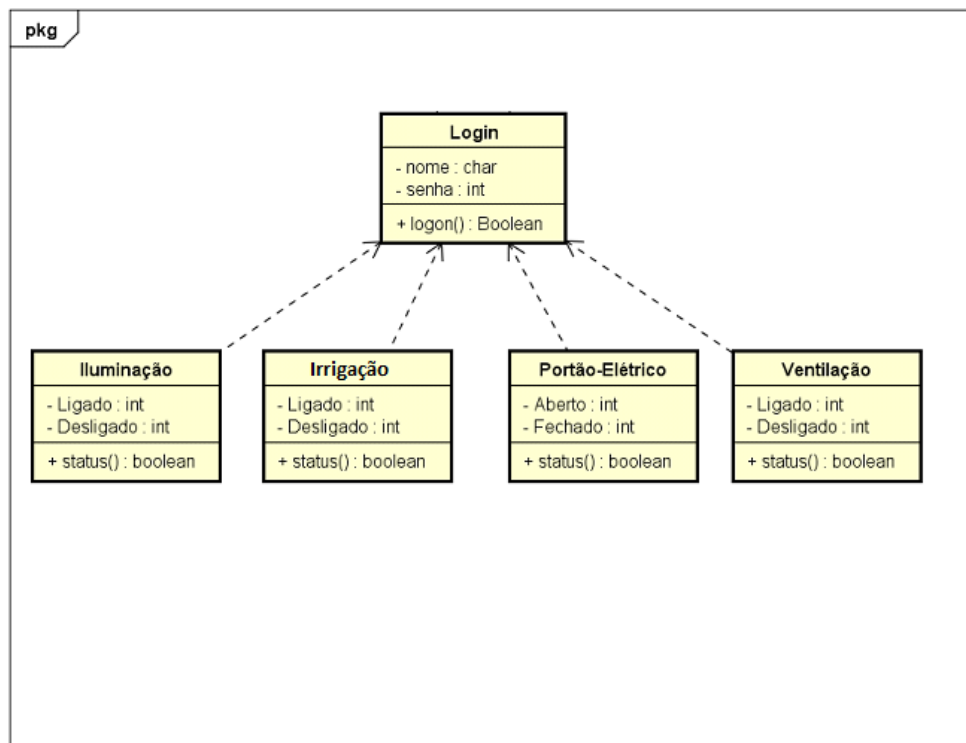


Figura 19: Diagrama de Classe

Fonte: Autoria própria.

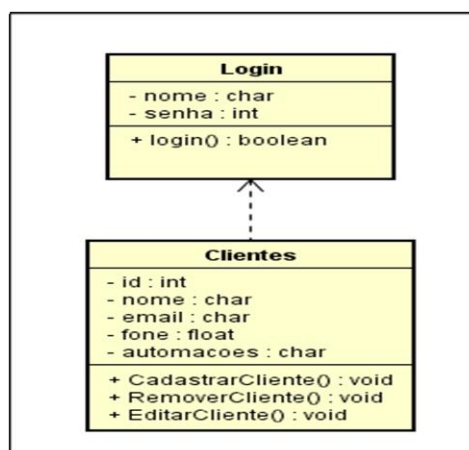


Figura 20: Diagrama de Classe Admin

Fonte: Autoria própria.

3.7. Diagrama de atividade

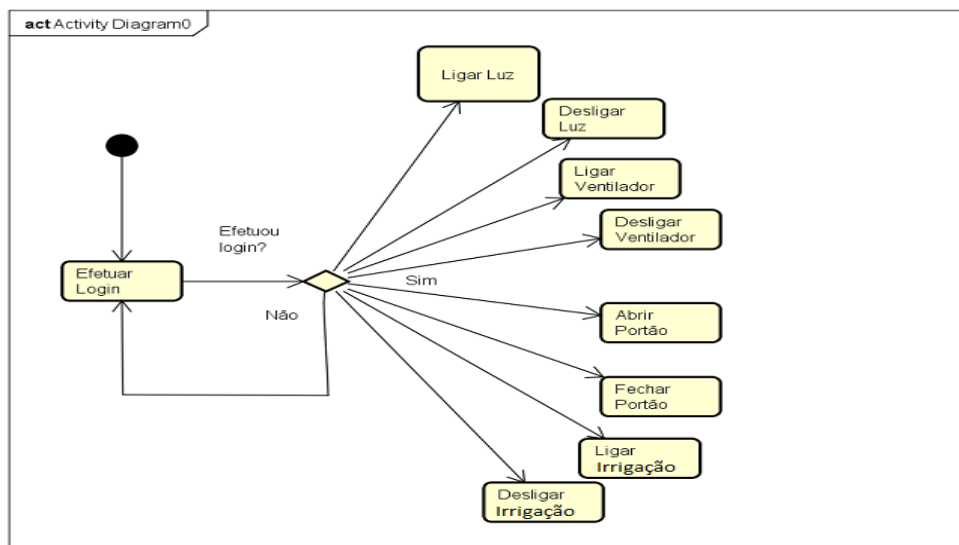


Figura 21: Diagrama de Atividade

Fonte: Autoria própria.

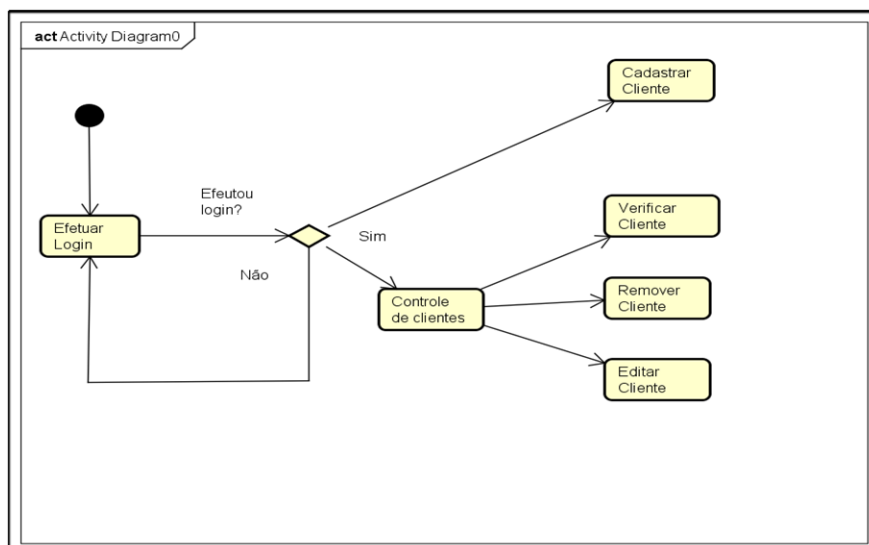


Figura 22: Diagrama de Atividade Admin

Fonte: Autoria própria.

3.8. Diagrama de sequência

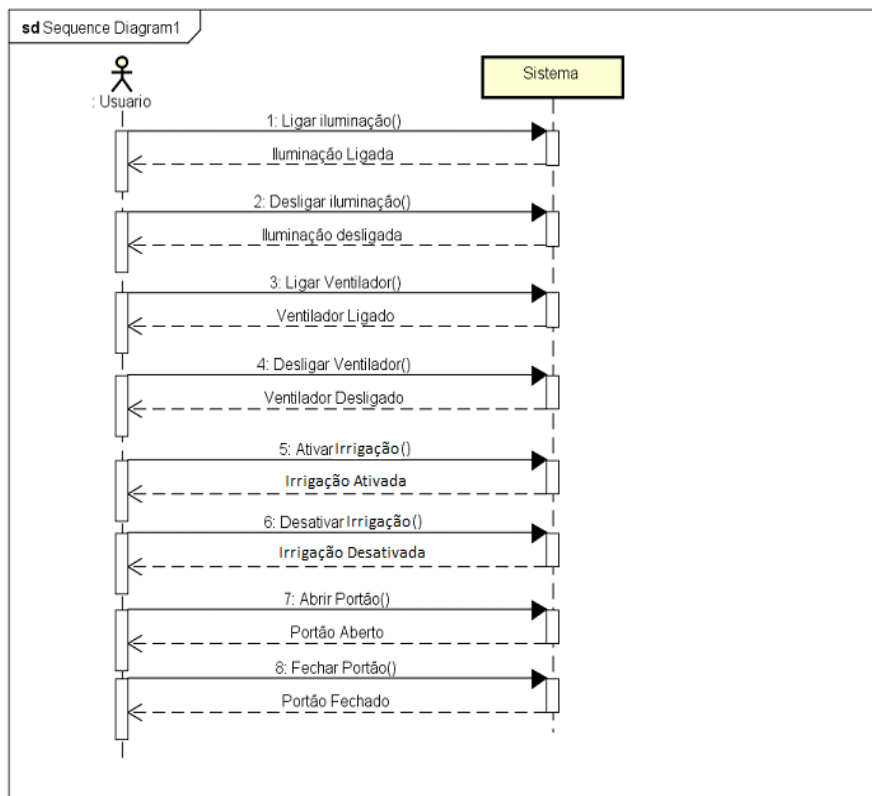


Figura 23: Diagrama de Sequência

Fonte: Autoria própria.

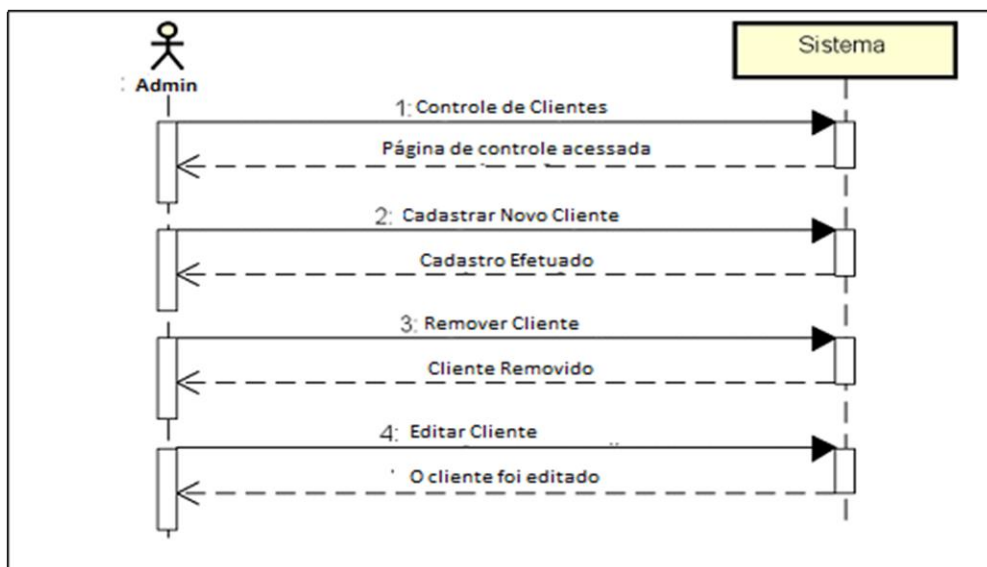


Figura 24: Diagrama de Sequência Admin

Fonte: Autoria própria.

3.9. Diagrama de entidade e relacionamento

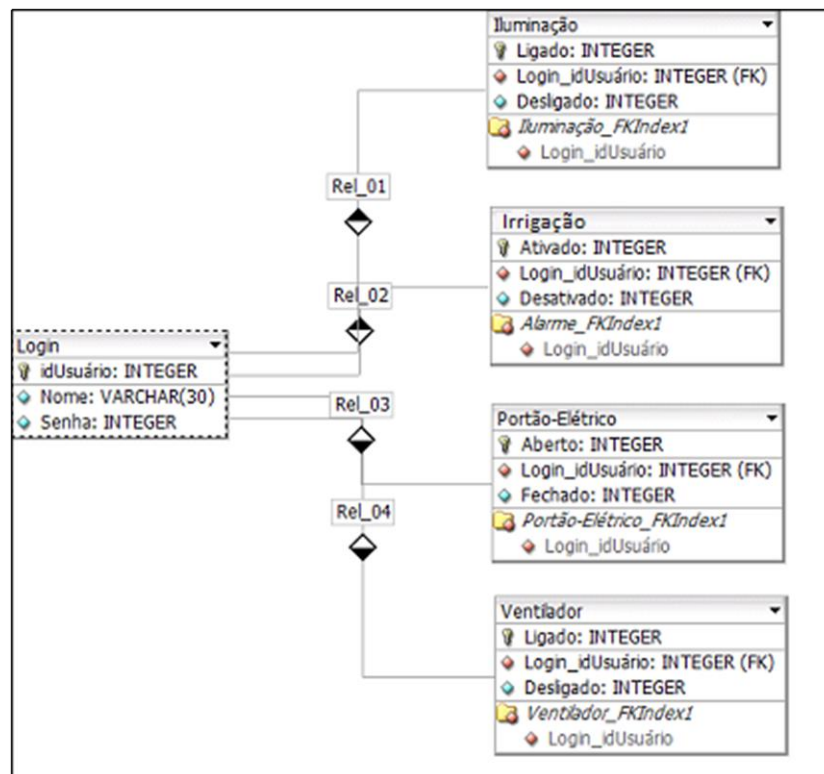


Figura 25: Diagrama ER

Fonte: Autoria própria.

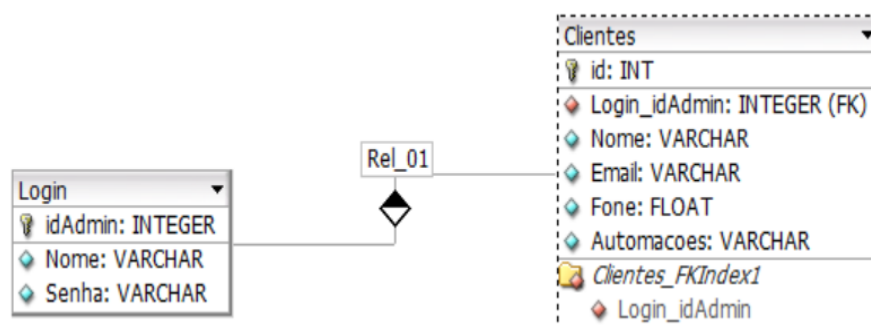


Figura 26: Diagrama ER Admin

Fonte: Autoria própria.

4. ESTRUTURA ANALÍTICA DO PROJETO (EAP)

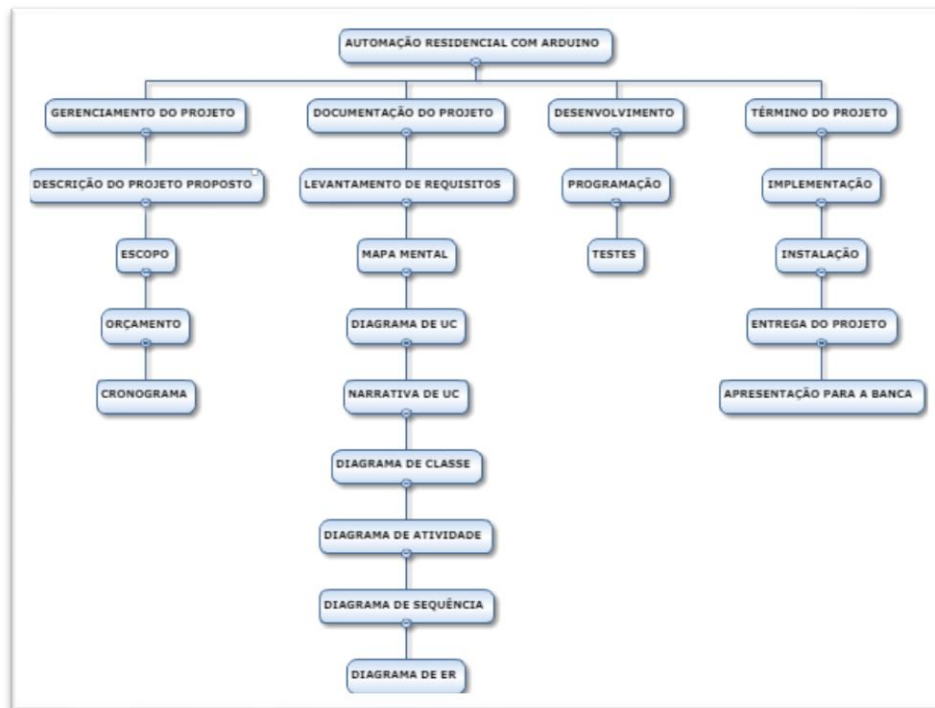


Figura 27: EAP

Fonte: Autoria própria.

5. IMPLEMENTAÇÃO

Para a implementação do sistema foi utilizada o software ARDUINO IDE e a linguagem PHP.

5.1. Configuração Arduino e ethernet Shield

Configurações da ethernet shield e do Arduino no software Arduino IDE

```
#include <Ethernet.h> //Biblioteca necessária para a utilização do modulo EthernetShield
#include <Servo.h> // Biblioteca necessária para a utilização do servo motor

Servo portao; //Definição da variável portão para o servo.

//Configuracoes do Ethernet Shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192,168,0, 90 }; // ip que o arduino assumira
byte gateway[] = { 192,168,0, 1 }; // ip do roteador
byte subnet[] = { 255, 255, 0, 0 };

//instanciamento do serviço de rede com atribuição de um nome para o servidor(server).
EthernetServer server(80); //porta utilizada 80

//definição das variáveis e dos pinos onde serão conectados os componentes da maquete.
int rele1 = 2;
int rele2 = 3;
int rele3 = 4;
int rele4 = 5;

void setup() { // função setup() é chamada no momento em que o programa começa. É usada para inicializar variáveis, definir
  // os modos de entrada ou saída dos pinos, indicar bibliotecas, etc.
  portao.attach(9); //definição do pino onde o portão elétrico será conectado
  Ethernet.begin(mac, ip, gateway, subnet); //função que envia os endereços mac, ip, gateway, subnet para o módulo de rede
  server.begin(); //função que inicia o módulo de rede de acordo com as atribuições anteriores

  //Aqui foram definidos os pinos como saída(OUTPUT)
  // pinMode. Esta função permite configurar um pino específico para se comportar como um pino de entrada ou de saída.
  pinMode(rele1,OUTPUT);
  pinMode(rele2,OUTPUT);
  pinMode(rele3,OUTPUT);
  pinMode(rele4,OUTPUT);
}

void loop() { //A função loop () será executada para sempre, ou até que você faça upload de um novo código.
  //Ela também pode ser reiniciada resetando o Arduino (através do botão de reset por exemplo).
  EthernetClient client = server.available(); //declaração da variável client que receberá as requisições
  //de server.available();
  if(client){ //estrutura que receberá as requisições do socket criado em arduino.php
    {
      switch(client.read()){ // função da biblioteca Ethernet.h que receberá o valor do socket
        {
          case '0': // "0" = valor recebido do Socket que acionará a função digitalWrite(rele1, HIGH).
            digitalWrite(rele1, HIGH); //digitalWrite Esta função escreve um valor HIGH (alto) ou LOW (baixo) em um pino digital que tenha sido configurado como OUTPUT (saída).
            break;
          case '1':
            digitalWrite(rele2, HIGH);
            break;
          case '2':
            digitalWrite(rele3, HIGH);
            break;
          case '3':
            digitalWrite(rele4, HIGH);
            break;
          case '4':
            digitalWrite(rele1, LOW);
            break;
          case '5':
            digitalWrite(rele2, LOW);
            break;
          case '6':
            digitalWrite(rele3, LOW);
            break;
          case '7':
            digitalWrite(rele4, LOW);
            break;
          case '8':
            portao.write(140); //aqui o portão será acionado e se movimentará em um ângulo de 140 graus no sentido anti horário
            break;
        }
      }
    }
  }
}
```

Figura 28: Código Arduino e Ethernet Shield

Fonte: Autoria própria.

5.2. Conexão Arduino e PHP

```

<?php
$sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
// $sock: variável para criar o socket
// socket_create: função que habilita o novo socket

socket_connect($sock, "192.168.200.90", 80); //(Socket, ip do arduino, Porta).
// socket_connect: Função que conecta o websocket

//Estrutura que receberá as requisições da interface web(menu.php) e associar com o parâmetro
// do $_POST['estadoRele'].
if ($_POST['estadoRele']=="r1_ligado")
{
    socket_write($sock, "0"); //socket_write: função do php que enviará via socket um valor
    // para a estrutura (switch (client.read()) do arduino que servirá
    // para o acionamento do relé.
}
if ($_POST['estadoRele']=="r1_desligado")
{
    socket_write($sock, "1");
}
if ($_POST['estadoRele']=="r2_ligado")
{
    socket_write($sock, "2");
}
}

```

Figura 29: Conexão Arduino e PHP

Fonte: Autoria própria

```

{
    socket_write($sock, "6");
}
if ($_POST['estadoRele']=="r4_desligado")
{
    socket_write($sock, "7");
}

header("Location: menu.php"); //header: retorna o menu.php

socket_close($sock); //socket_close: Encerra a conexão com o socket.
?>

```

Figura 30: Conexão Arduino e PHP 2

Fonte: Autoria própria

5.3. Segurança

A segurança é feita pela função hash md5, que é feita para criptografar a senha cadastrada no banco de dados do sistema

```
<?php
echo "Renan457: " . md5("Renan457");
echo "<BR/><br/>";
echo "Guerin@457: " . md5("Guerin@457");
echo "<BR/><br/>";
echo "Renan Guerin: " . md5("Renan Guerin");
//Algoritmo baseado em hash
//Criptografar a senha no banco...
?>
```

Figura 31: Função md5

Fonte: Autoria própria

6. CONCLUSÃO

Foi muito satisfatório trabalhar em uma tecnologia que eu nunca havia estudado antes, pois além do novo conhecimento da tecnologia, também aprendi o básico de eletrônica. Apesar do sistema ser simples, foram dedicadas muitas horas de estudo, para que os usuários possam ter melhor experiência e facilidade com a tecnologia. Pretendo continuar estudando essa tecnologia, pois ela pode se tornar popular em futuro próximo, vou me manter sempre atualizado para desenvolver novas aplicações e se for possível, comercializá-las

7. REFERÊNCIAS

Arduino, Disponível em: <https://www.arduino.cc/en/main/software/>

Acessado em: 19/03/2018.

Artigo Domótica. Disponível em:

<https://mundoeducacao.bol.uol.com.br/informatica/domotica.htm/>

Acessado em: 17/03/2017.

Brackets. Disponível em: <https://imasters.com.br/box/ferramenta/brackets/>

Acessado em 21/12/2017.

GUEDES, Gilleanes **T.A UML 2: Uma abordagem prática**. 2 Ed. São Paulo: Novatec, 2011.

HTML5. Disponível em:

<https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5>

Acessado em 19/03/2018.

Javascript - Introdução à linguagem Javascript. Disponível em:

<https://br.ccm.net/faq/2680-javascript-introducao-a-linguagem-javascript/>

Acessado em 15/11/2017.

MARCONDES, Rafael **O que é Javascript**. Disponível em:

<https://developer.mozilla.org/pt->

[PT/docs/Web/JavaScript/O_que_%C3%A9_o_JavaScript](https://developer.mozilla.org/pt-PT/docs/Web/JavaScript/O_que_%C3%A9_o_JavaScript)

Acessado em 27/05/2018.

MCROBERTS, Michael **Arduino Básico**. 2 Ed. São Paulo: Novatec, 2015.

O que é PHP e porque você precisa aprender HOJE! Disponível em:
<https://bencode.com.br/o-que-e-php/>

Acessado em: 19/03/2018.

PEREIRA, Ana Paula **O que é CSS?** Disponível em:
<https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm/>

Acessado em 15/02/2017.

SILVA, Júlia Marques Carvalho **Php na Prática**. Elsevier: Campus.

VILLARINO, Julia **INTERNET DAS COISAS: UM DESENHO DO FUTURO**,
Disponível em: <http://www.proof.com.br/blog/internet-das-coisas/>

Acessado em: 25/03/2018.