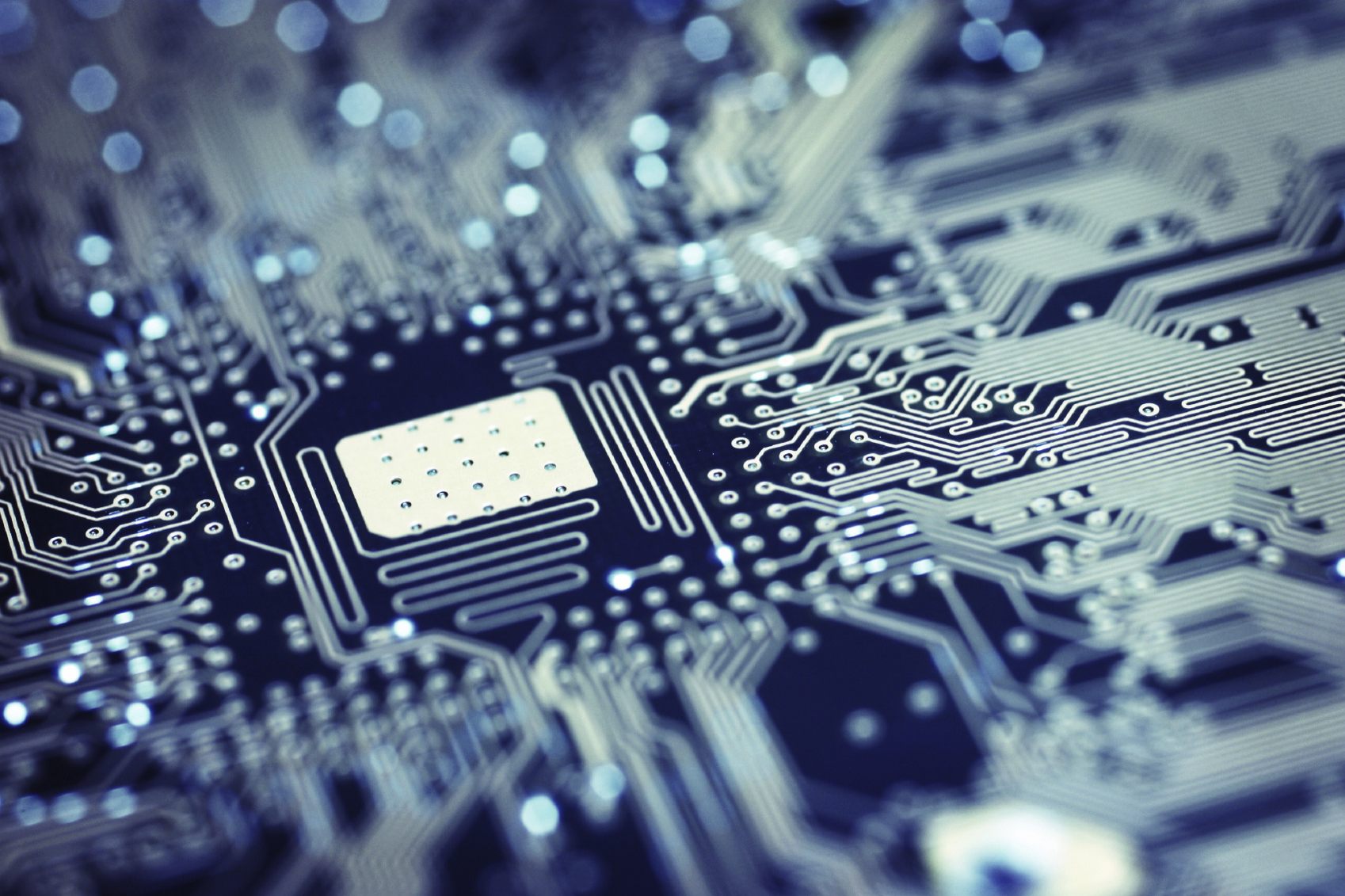
|  |
| --- |
| dOCUMENTO DE REQUiSITOS  Projeto SmartSense |
| Lucas Henrique Costa Araújo |



**Histórico de Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Autor | Descrição |
| 01/09/2019 | 0.1 | Lucas Henrique | Criação do documento de requisitos funcionais. |
| 18/09/2019 | 0.2 | Lucas Henrique | Remoção do aplicativo mobile |

**Sumário**

|  |  |  |
| --- | --- | --- |
| **1** | **O que é?** | **4** |
| **2** | **Descrição do projeto** | **4** |
| **3** | **Requisitos de Software** | **6** |
| **3.1** | **Requisitos funcionais** | **6** |
| **3.2** | **Requisitos não funcionais** | **7** |
| **4** | **Requisitos de Hardware** | **8** |
| **4.1** | **Requisitos funcionais** | **8** |
| **4.2** | **Requisitos não funcionais** | **9** |
| **5** | **Modelo de Caso de Uso** | **10** |
| **6** | **Protótipo** | **12** |
| **7** | **Técnológias utilizadas** | **14** |
| **8** | **Componetes de hardwares utilizados** | **14** |
| **9** | **Cronograma de desenvolvimento** | **15** |
| **10** | **Marcos de entregas** | **16** |

1. **O que é ?**

O documento de requisitos tem a finalidade de descrever as funcionalidades do software a ser criado para simplificar o entendimento do cliente, usuários finais e dos desenvolvedores tanto para a sua criação e manutenção.

Este documento contém todos os requisitos e funcionalidades que o sistema deve ter, tanto os funcionais quanto os não funcionais e também suas restrições.

Ao ser finalizado e validado, este documento tem a serventia de um contrato formal entre o cliente e a empresa desenvolvedora, gerando a obrigação de que o produto final obedeça às especificações definidas, porém podendo haver alteração em qualquer momento no desenvolvimento desde que ambas as partes entre em acordo, a depender do tipo de alteração, um custo extra pode ser adicionado ao valor final do software.

1. **Descrição do Projeto**

O projeto SmartSense visa oferecer um sistema de monitoramento de dispostivos que estejam concetados com o mesmo. Ele consiste em alguns componentes, estes são: Uma camada de apresentação, composta por uma página web, responsável por realizar a conexão com os dispositivos selecionados previamente pelo usuário e implementar uma interface com as funcionalidades disponiveis no módulo embarcado, sendo que estas podem ser de apenas monitoramento de dados ou de envio de comandos para o dispositivo; Um servidor *backend* resposável por realizar a comunicação direta com os dispositivos embarcados e enviar os dados coletados para a camada de apresentação e ; Uma biblioteca responsável por implementar um protocolo de comunicação que seja capaz de expor suas funcionalidades existentes no dispositivo embarcado e também gerêncie a segurança, não permitido que servidores desconhecidos se conectem com o mesmo.

Uma descrição mais completa sobre alguns componentes existentes no sistema será apresentado a seguir.

**Página Web** - O usuário deverá ter uma página web que possua uma tela de login, uma em que o usuário selecione os dispositivos disponíveis conectados ao sistema e outra para a exibição dos dados monitorados dos dispositivos e também forneça um mecanismo de envio de comandos quando o módulo embarcado for compatível.

**Página Configuração Web -** Nesta página o usuário irá selecionar quais são os dispositivos e dados que deseja exibir e quais interfaces de comandos deseja um fácil acesso. O sistema deverá apresentar componentes de interface de modo que estes sejam utilizados para tipos de exibições de dados distintos, podendo ser gráficos em tempo de execução ou campo de entrada de dados para serem enviados para os dispositivos.

**Servidor Backend** - O sistema deve possuir um servidor que intermediei a comunicação com os dispositivos embarcados, este deve ser capaz de identificar os métodos aceitos e os dados transmitidos por cada dispositivo conectado.

**Biblioteca para Arduino® -** O sistema deve possuir um protocolo de comunicação padrão para todos os dispositivos embarcados, este será útil para a compatibilidade de novos equipamentos.

A biblioteca será útil para facilitar a implementação da comunicação entre os dispositivos embarcados e o servidor backend, esta será capaz de informar quais dados podem ser enviados e quais comando podem ser recebidos.

Esta biblioteca deverá ser compatível com o modulo wifi ESP-01 para Arduino.

1. **Requisitos de Software**

**3.1 Requisitos Funcionais**

Os requisitos funcionais se referem ao comportamento que o sistema deve apresentar em seu uso correto.

|  |  |  |
| --- | --- | --- |
| RFS01 | | |
| Requisito | Descrição | Prioridade |
| Seleção de dispositivos | O sistema deverá apresentar um mecanismo para selecionar quais diversos conectados serão monitorados simultaneamente. | Alta |

|  |  |  |
| --- | --- | --- |
| RFS02 | | |
| Requisito | Descrição | Prioridade |
| Componente de interface - Gráfico | A página web deverá possuir um componente de interface que possua a entrada compatível com a saída dos módulo embarcado e gere um gráfico em tempo de execução exibindo o comportamento da variação dos dados. | Baixa |

|  |  |  |
| --- | --- | --- |
| RFS03 | | |
| Requisito | Descrição | Prioridade |
| Componente de interface – Campo de dados | A página web deverá possuir um mecanismo de exibição de dados, exibindo somente o valor atualizado de um dado oriundo do módulo. | Média |

|  |  |  |
| --- | --- | --- |
| RFS04 | | |
| Requisito | Descrição | Prioridade |
| Componente de interface – Botão de ação | O sistema deverá fornecer um componente para realizar uma ação do tipo *on/off*, sendo que está deverá ser compatível com os dispositivos que ofereçam uma entrada de comandos semelhante. | Média |

|  |  |  |
| --- | --- | --- |
| RFS05 | | |
| Requisito | Descrição | Prioridade |
| Componente de interface – Campo de ação | O sistema deverá fornecer um componente para receber uma entrada de dados que será enviada para o controlador, esta entrada será numérica e irá consistir basicamente no envio de dados como um valor desejado de um *setpoint.* | Média |

|  |  |  |
| --- | --- | --- |
| RFS06 | | |
| Requisito | Descrição | Prioridade |
| Protocolo de comunicação do Arduino® | Deverá sem implementado um protocolo de comunicação padrão entre o módulo embarcado e o servidor *backend*, este deve ser feito de modo que após a conexão o servidor consiga reconhecer todas as entradas e saídas de dados e os métodos aceitos pelo módulo embarcado. | Alta |

|  |  |  |
| --- | --- | --- |
| RFS07 | | |
| Requisito | Descrição | Prioridade |
| Configuração de interface | A página web deverá possuir uma configuração de interface, de modo que seja possível selecionar os componentes e quais dados e comandos os mesmos são responsáveis. | Média |

**3.2 Requisitos não funcionais**

Os requisitos não funcionais são aqueles que não estão diretamente relacionados as funcionalidades do sistema, eles estão relacionados com questões de segurança e usabilidade do sistema.

|  |  |
| --- | --- |
| RNFS01 | |
| Requisito | Descrição |
| Interface simples | O sistema deve apresentar uma interface que apresente os conteúdos essenciais para o usuário. |

|  |  |
| --- | --- |
| RNFS02 | |
| Requisito | Descrição |
| Mecanismo de autenticação | O sistema deve possuir um mecanismo para autenticação de usuários, onde o mesmo irá preencher dois campos, o do usuário e a da senha, e logo após selecionar um botão para confirmar o *login*. |

|  |  |
| --- | --- |
| RNFS03 | |
| Requisito | Descrição |
| Biblioteca de comunicação padrão | Está biblioteca será implementada para ser utilizada como o mecanismo gerenciador de comunicação entre o módulo embarcado e o sistema *backend*, esta deverá possuir a capacidade de informar ao servidor os dados que são tratados pelo módulo e os comandos que são reconhecidos, também será responsável por receber e interpretar comandos do servidor.  Esta biblioteca deverá ser compatível com o modulo *wi-fi* ESP-01 para Arduino. |

1. **Requisitos de Hardware**

**4.1 Requisitos Funcionais**

Os requisitos funcionais de hardware se referem as caracteristicas que os dispositivos deverão apresentar.

|  |  |  |
| --- | --- | --- |
| RFH01 | | |
| Requisito | Descrição | Prioridade |
| Conexão *wi-fi* | O módulo embarcado deverá possuir um mecanismo de comunicação para realizar a conexão via wi-fi com o servidor *backend*. | Média |

|  |  |  |
| --- | --- | --- |
| RFH02 | | |
| Requisito | Descrição | Prioridade |
| Módulo - Controle de luminosidade | O modulo de controle de luminosidade será responsável será responsável por coletar dados de luminosidade de um sensor de luminosidade presente no módulo e a partir desta entrada de dados alterar a intensidade luminosa do seu atuador, o funcionamento deste dispositivo irá depender de um valor de *setpoint* de luminosidade previamente informado pelo usuário.  Este módulo possuirá uma saída de dados que indicará o valor do sensor e a intensidade do atuador e uma entrada de dados com o valor do *setpoint* desejado. | Média |

|  |  |  |
| --- | --- | --- |
| RFH03 | | |
| Requisito | Descrição | Prioridade |
| Módulo - Medição de temperatura | O módulo de medição de temperatura irá conter um sensor de temperatura, que será responsável por coletar os dados referentes a temperatura de um ambiente.  Este módulo possuirá apenas a saída de dados contendo os valores referentes a temperatura. | Média |

|  |  |  |
| --- | --- | --- |
| RFH04 | | |
| Requisito | Descrição | Prioridade |
| Módulo - Controle de trava elétrica | O módulo de controle da trava elétrica será do tipo *on off*, este consiste em um relé que será acionado via transistores e realiza a abertura/fechamento de uma trava elétrica.  Este módulo possuirá apenas uma entrada de dados indicando o acionamento. O desligamento ocorrerá de forma automática 10 segundos após o acionamento da mesma. | Média |

**4.2 Requisitos não funcionais**

Os requisitos não funcionais de hardware são aqueles que não estão diretamente relacionados as funcionalidades do sistema, eles estão relacionados com os componentes necessários para a implementação de todo os dispositivos.

|  |  |
| --- | --- |
| RNFH01 | |
| Requisito | Descrição |
| Módulo embarcado | O módulo embarcado será desenvolvido na plataforma open-hardware Arduino. |

|  |  |
| --- | --- |
| RNFH02 | |
| Requisito | Descrição |
| Módulo *Wi-fi* | O módulo embarcado irá conter uma placa ESP-01, responsável por conectar a uma rede *wi-fi* habilitando a comunicação entre os dispositivos e o servidor. |

|  |  |
| --- | --- |
| RNFH03 | |
| Requisito | Descrição |
| Controle de luminosidade | O módulo para controle de luminosidade deverá possuir um sensor LDR para verificar a luminosidade ambiente e também um LED. Este irá possuir um controlador PD e atuar em uma faixa de aceitação de até 5% de variação de um valor de *setpoint* previamente definido, sendo que este poderá ser alterado em tempo de execução. |

1. **Modelo de Caso de Uso**

O modelo de caso de uso é a forma de representar as funcionalidades do sistema em que os atores possam interagir, os atores de um sistema podem ser definidos como o usuário, outros sistemas, equipamentos em que o mesmo possa se comunicar entre outros.

Para representar o modelo de caso de uso é necessário a implementação dos diagramas de caso de uso onde mostram as funcionalidades externas do sistema e quais são os atores que podem interagir com aquela funcionalidade.

Com os requisitos funcionais apresentados é possível criar os seguintes casos de usos:

|  |  |
| --- | --- |
| CSU01 | |
| Realiza *Login* | |
| Pré-condição: | - Acessar a página do sistema responsável para realizar o *login* dos usuários. |
| Fluxo de eventos primário: | 1. O usuário entrará com o seu *login*. 2. O usuário entrará com a senha. 3. O usuário confirmará clicando no botão entrar. 4. O sistema verifica se os dados inseridos estão corretos. 5. O sistema redireciona para a página do usuário. |
| Fluxo de eventos secundário: | 1. Se o usuário inserir dados inválidos o sistema apresenta uma mensagem identificando o problema. |
| Pós-condição: | O usuário passa a ter acesso a tela de configuração e de monitoração. |

|  |  |
| --- | --- |
| CSU02 | |
| Selecionar dispositivo | |
| Pré-condição: | - Estar registrado no sistema (CSU01) |
| Fluxo de eventos primário: | 1. Entrar na parte do sistema referente a configuração. 2. Selecionar o dispositivo que deseja monitorar. |
| Fluxo de eventos secundário: | 1. Se não houverem dispositivos conectados o sistema irá informar. |
| Pós-condição: | O usuário passa a monitorar os dados transmitidos pelo dispositivo. |

|  |  |
| --- | --- |
| CSU03 | |
| Configuração de interface de componentes | |
| Pré-condição: | - Estar registrado no sistema (CSU01)  - Ter selecionado dispositivos (CSU02)  - Estar utilizando a página web. |
| Fluxo de eventos primário: | 1. Entrar na parte do sistema referente a configuração de interfaces. 2. Selecionar o componente de interface. 3. Associar este componente a um comando ou dado de um dispositivos embarcado selecionado previamente. |
| Fluxo de eventos secundário: | 1. Se não houverem dispositivos conectados ou selecionados o sistema irá informar. |
| Pós-condição: | A componente ficará disponível na homepage do usuário. |

1. **Protótipo**

Para o desenvolvimento inicial do projeto é necessário a criação de um protótipo para exibição de funcionamento, este irá consistir com uma versão inicial do projeto um sistema web com funcionalidades parciais implementada, sendo a conexão com dispositivos presentes na rede, a exibição dos dados coletados pelos mesmo e uma implementação inicial de uma interface para envio de comandos para o dispositivo.

O protótipo irá consistir em um modelo físico de conterá quatro modulos embarcados, estes serão: Dois módulos de luminosidade que possuirá uma entrada de dados para o *setpoint*, valor desejado para a luminosidade do ambiente e saidas de dados contendo o valor da medição de um sensor LDR e a intensidade de um LED contido do modulo; Um módulo de medição de temperatura por meio de um termistor, possuirá apenas uma saída de dados indicando a temperatura medida pelo sensor e; Um modulo de acionamento de um relé que simula a abertura de uma trava eletrica.

Este sistema será simulado em um ambiente físico que irá simular uma casa, sendo que os modulos serão distribuidos entres so comodos da residência. A figura apresentada a seguir exibe modelagem de como será o ambiente físico que será utilizado para realizar a simulação. Esta apresenta 5 comodos sendo uma sala, uma cozinha, dois quartos e um banheiro.

Os módulos de controle de luminosidade serão distribuidos nos quartos, o módulo de monitoramento de temperatura será testado na cozinha e o módulo de controle da trava eletrica será testado na sala.



Figura 1: Modelagem 3d do ambiente físico utilizado na simulação

1. **Tecnológias utilizadas**

Para realizar o desenvolvimento dos softwares presentes no sistema serão utilizados a tecnológias que serão apresentadas a seguir.

Para o desenvolvimento da página web será utilizado o framework Angular em sua versão 7, com este é possível implementar um front-end SPA (Single Page Applications) utilizando diversas bibliotecas existentes com componentes diversos que facilitam o seu desenvolvimento. A utilização de frameworks contribui para a diminuição do tempo de desenvolvimento da aplicação.

O Angular é uma ferramenta de programação multiplataforma focada no front-end da aplicação, desenvolvida pela Google®, este é um dos frameworks mais populares do mercado. Possui quatro conceitos fundamentais, componentes, diretivas,rotas e serviços.

Os componentes permitem que a criação de itens de layout de forma que os mesmos funcionem como um modulo do sistema podendo ser adicionado e retirado a qualquer momento simplesmente com a adição de uma tag. As diretivas permitem a alteração de componentes adicionados às páginas, como o exemplo da diretiva \*ngIf, esta permite uma condicional no arquivo html que pode ser utilizado para exibir ou não um trecho de layout. As rotas são os mecanismos que definem qual página será exibida no sistema. Os serviços são responsáveis por armazenarem os códigos que realizam as requisições.

Para o desenvolvimento do sistema *backend* foi utilizado a tecnologia Node.js, esta possui uma arquitetura que pode ser nomeado com *single thread* com IO não bloqueante, com isso é possível realizar requisições simultâneas de forma que o acesso ao disco para cada requisição irá ocorrer de forma paralela. Cada persistência de dados ocorrerá de forma assíncrona no servidor, sendo que quando o processo for concluído uma chamada será realizada informando a conclusão do processo.

Também foi utilizado o *framework express* no *backend*, com ele é possível modelar uma *api* de forma simples e ágil, utilizando *middlewares* para configurações e seus métodos existentes para a criação de *endpoints* para serem consumidas por seus clientes.

1. **Componetes de hardwares utilizados**

O módulos embarcados serão desenvolvidas na plataforma *open-hardware* Arduino juntamente com um modulo wi-fi Esp 01.

O módulo de controle de luminosidade conterá um sensor do tipo LDR, póssuira uma entrada para definir o *setpoint* do sistema de controle.

O módulo de monitoramento de temperatura possuira um sensor do tipo termistor.

O módulo da trava elétrica possuirá um circuito elétrico contendo transistores e um relé para realizar o chaveamente uma uma tensão externa para o acionamento de uma trava elétrica, possuindo uma entrada referente a um sinal de acionamente da trava.

Estas informações poderão ser alteradas a medida que o sistema seja desenvolvido.

## Cronograma de desenvolvimento

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fases | Atividade | Mês | | | | | | | | | | | | | | | | |
| Setembro | | | | | Outubro | | | | | Novembro | | | | Dezembro | | |
| 1º | 2º | 3º | 4º | 1º | | 2º | 3º | 4º | 1º | | 2º | 3º | 4º | | 1º | 2º |
| Documentação | Análise de requisitos |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Revisão de requisitos |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Desenvolvimento de Software | Página Web |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Servidor *backend* |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Protocolo de comunicação do dispositivo embarcado |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Desenvolvimento de Hardware | Integração da plataforma com o módulo *wi-fi* |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Módulo de controle de luminosidade |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Módulo para monitoramento de temperatura |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Módulo para acionamento da trava elétrica |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Testes de software | Página Web |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Servidor *backend* |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Integração do servidor e da página web |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Integração do servidor e como o módulo embarcado |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Teste de hardware | Sistema de controle de luminosidade |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Mecanismo de monitoramento da temperatura |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |
| Mecanismo de controle da trava elétrica |  |  |  |  |  | |  |  |  |  | |  |  |  | |  |  |

## Datas de entrega

|  |  |
| --- | --- |
| Atividade | Data |
| Documento de requisitos inicial | 06/09/2019 |
| Protótipo inicial da camada de apresentação do sistema | 09/10/2019 |
| Protótipo inicial do módulo de controle de luminosidade | 13/11/2019 |
| Protótipo inicial do módulo de controle de temperatura | 27/11/1019 |
| Versão inicial do sistema completo | 18/12/2019 |