

[**Documento de Requisitos**](#_3znysh7)

### Smart Room: Criação de um quarto inteligente com preço acessível

Matheus Henrique Lopes BADOCO[[1]](#footnote-1)

Centro Universitário Municipal de Franca – lopsbdc@gmail.com.br

Mahezer Carvalho de MELO[[2]](#footnote-2)

Centro Universitário Municipal de Franca – mahezercarvalho@gmail.com.br

Orientador:Leandro BORGES[[3]](#footnote-3)

Centro Universitário Municipal de Franca – leandro.borges@me.com

**Franca**

**2018**

**Sumário**

**1 Introdução 3**

**2 Descrição geral do sistema 3**

**2.1 Problema 3**

**2.2 Justificativa 3**

**2.3 Objetivo 3**

**3Definição dos Processos da Aplicação 4**

**3.1 BPMN 4**

**3.2 Diagrama de Caso de Uso 5**

**4 Matriz de Rastreabilidade 6**

**4.1 Requisitos Funcionais 6**

**4.2 Regra de Negócio 7**

**4.3 Casos de Uso 8**

**4.4 Matriz 9**

**5 Diagrama de Sequência 10**

**6 Máquina de Estado 12**

**7 Prototipação de Telas 15**

**1 Introdução**

Este documento tem como principal função facilitar a compreensão da aplicação desenvolvida através de diagramas utilizados na Engenharia de Software. Ele irá tratar sobre o desenvolvimento, os requisitos necessários para desenvolve-lo, bem como o funcionamento e a aplicabilidade da solução.

# 2 Descrição geral do sistema

A solução desenvolvida visa facilitar a vida das pessoas no aspecto de controlar o próprio quarto. Além de ser possível controlar a umidade, temperatura e as luzes do quarto por um aplicativo de Smartphone, o sistema tem como principal diferencial o modo automático: onde o próprio quarto “se controla”, sem a necessidade de intervenção humana – e mais; tudo isso por um preço acessível.

**2.1 Problema**

Muitas pessoas que foram entrevistadas antes do desenvolvimento dessa solução tinham como problema a falta de automação domiciliar, o que fazia a casa depender totalmente de uma intervenção humana o tempo todo. Além disso, quando realizamos a pesquisa desses produtos no mercado notamos que o valor deles são um pouco elevados (o que leva a maioria das pessoas a desistirem), além de não possuírem um modo automático.

# 2.2 Justificativa

O que nós buscamos nesse projeto não é apenas controlar, mas também automatizar um quarto, visando reduzir o stress das pessoas, e também, melhorar a qualidade de vida; e também, decidimos trabalhar com peças de custo baixo e que possuíssem um bom benefício a fim de reduzir o preço final dessa solução.

# 2.3 Objetivo

# 

O principal objetivo é o desenvolvimento de uma placa que seja responsável por gerenciar o quarto, bem como a criação de um aplicativo para Smartphone para controla-la (a fim dealém de facilitar a vida do usuário); tudo isso utilizando componentes de baixo custo. Para alcançar esse objetivo, escolhemos a placa NodeMCU, que possui os recursos necessários para a realização dessa solução e um preço baixo em relação aos concorrentes.

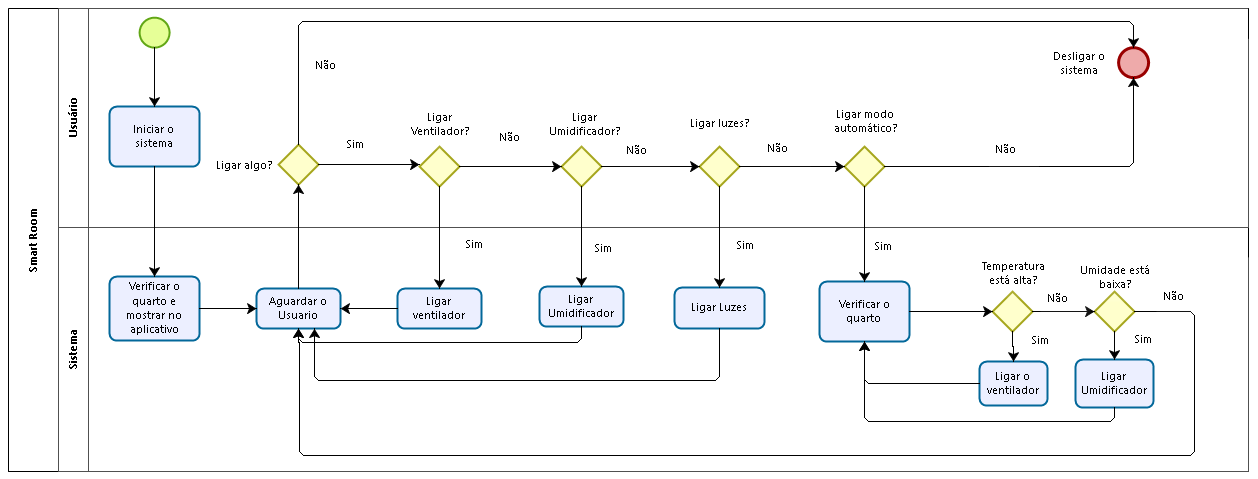
# 3 Definição dos Processos da Aplicação

Esta é a parte da documentação que trata os diagramas de engenharia de software desenvolvidos e necessários para a construção da solução.

# 3.1 BPMN

O BPMN (Business ProcessModelandNotation) visa entender o fluxo de processos de um modelo de negócios, além de mapear e otimizar fluxos de processos no processo de criação de um software.

A imagemabaixo mostra o BPMN desenvolvido para este projeto, mostrando como o sistema funciona, bem como as decisões que podem ser tomadas (tanto pelo sistema, quanto pelo usuário).



# Imagem 1 – BPMN

# 3.2 Diagrama de Caso de Uso

O Diagrama de Caso de Uso pertence a família de diagramas UML (UnifiedModellingLanguage).Diferentemente do BPMN, este diagrama busca mapear as principais funcionalidades do sistema, através do ponto de vista dos atores da aplicação.

A imagemabaixo mostra o Diagrama de Caso de Uso utilizado nesse projeto, modelando os atores e seus respectivos Casos de Uso.

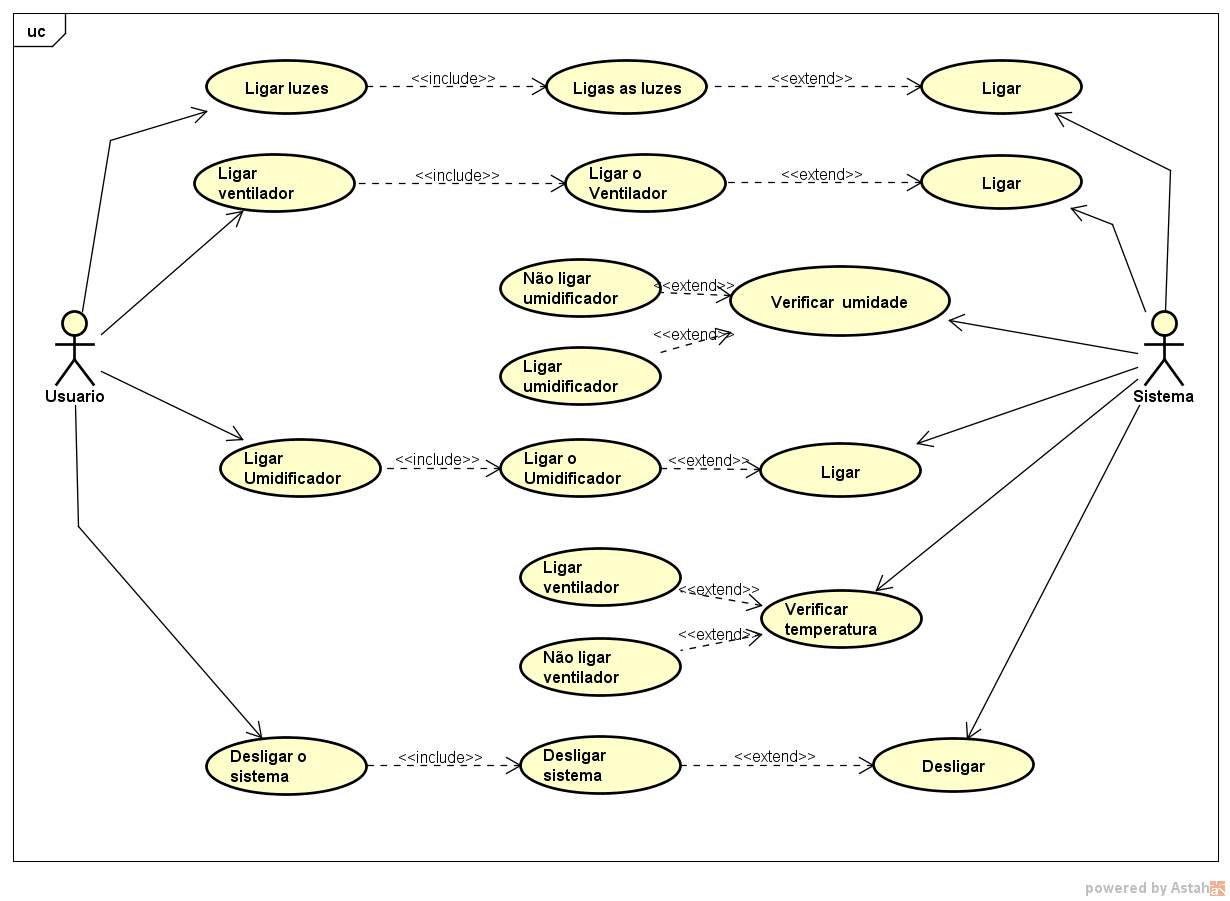


Imagem 2: Diagrama de Caso de Uso

# 4. Matriz de Rastreabilidade

Após mapear o Diagrama de caso de uso e BPMN, é necessário realizar a matriz de rastreabilidade, a fim de facilitar uma futura manutenção no sistema. Para isso, primeiramente, é necessário mapear os Requisitos Funcionais, Casos de Uso e Regras de Negócio.

**4.1 Requisitos Funcionais**

Essa parte da documentação visa mapear quais os requisitos fundamentais para o funcionamento do sistema. Abaixo podemos ver as imagens referentes a cada requisito:

Imagem 3: Requisito Funcional 1



Imagem 4: Requisito Funcional 2

**4.2Regra de Negócio**

Essa parte da documentação visa mapear as principais regras para que o sistema funcione da maneira adequada. Abaixo podemos ver as imagens referentes a cada regra:

Imagem 5: Regra de negócio 1



Imagem 6: Regra de negócio 2



Imagem 7: Regra de negócio 3



Imagem 8: Regra de negócio 4



Imagem 9: Regra de negócio 5

**4.3 Casos de Uso**

Essa parte da documentação visa mapear de uma maneira mais detalhada cada parte do caso de uso (apresentado no item 3.2), e tornar mais visual o que esses casos afetam o sistema. Abaixo podemos ver as imagens referentes a cada caso:



Imagem 10: Caso de Uso 1



Imagem 11: Caso de Uso 2



Imagem12: Caso de Uso 3



Imagem 13: Caso de Uso 4



Imagem 14: Caso de Uso 5



Imagem 15: Caso de Uso 5

**4.4Matriz**

A Matriz de rastreabilidade tem como objetivo mapear os riscos de edições ou alterações no sistema, e facilitar a vida do programador em casos de erros ou manutenções; tudo isso com base nos requisitos funcionais, casos de uso e regras de negócio. Como já foram explicados todos os requisitos para a criação da Matriz, segue abaixo as imagens referentes a ela:



Imagem 16: Matriz de Rastreabilidade – Casos de Uso x Requisitos Funcionais



Imagem 17: Matriz de Rastreabilidade – Regras de Negócio x Requisitos Funcionais

# 5 Diagrama de Sequência

O Diagrama de Sequência tem por objetivo compreender a dinâmica dentro do sistema, entendendo como ocorre o fluxo dos processosentre os principais atores e suas interações. Abaixo está a imagem do Diagrama de Sequência:

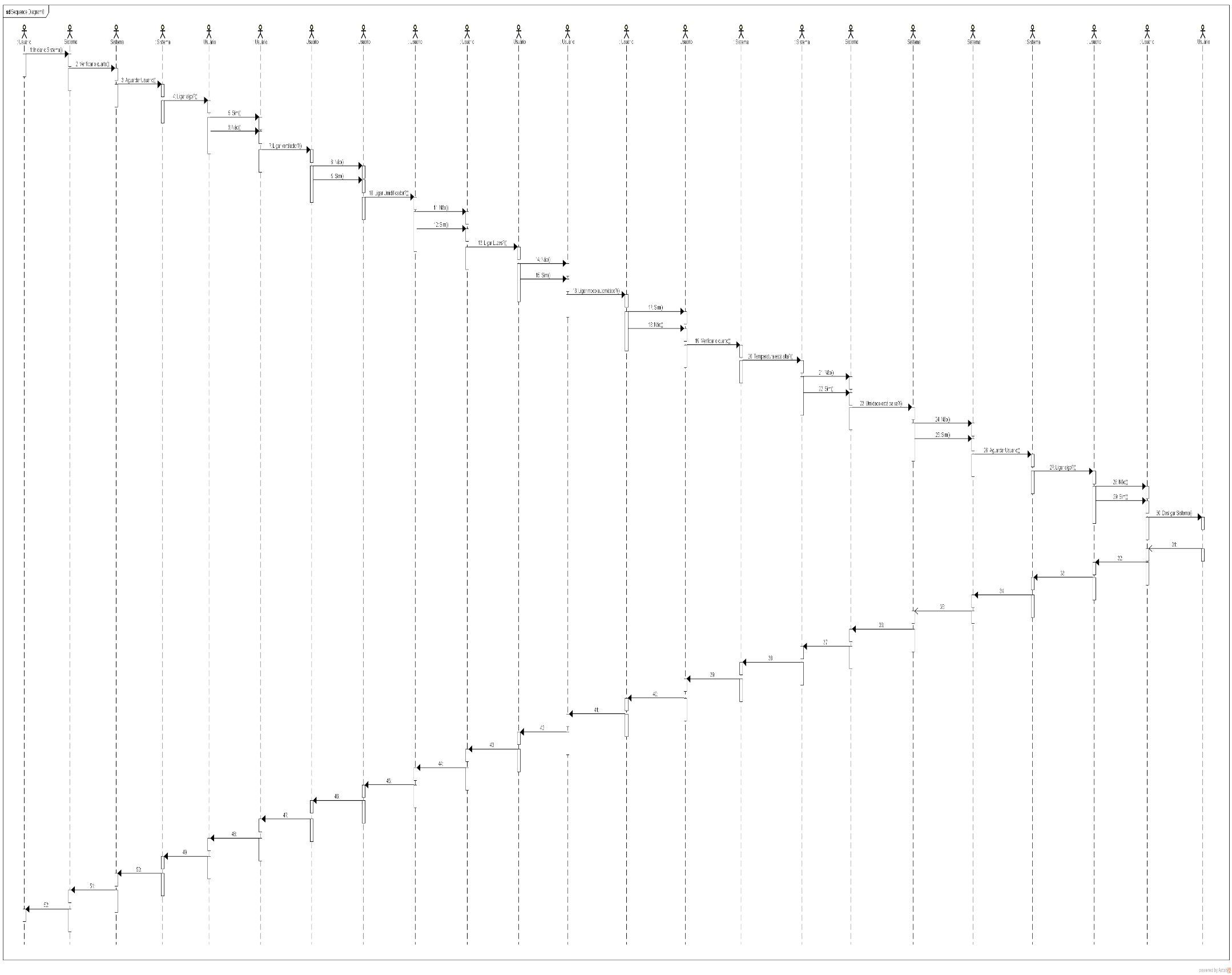


Imagem 18: Diagrama de Sequência

**6 Máquina de Estado**

Os diagramas de Máquina de Estado têm por objetivo compreender os diferentes estados que uma aplicação pode ter em cada uma de suas decisões, a fim de facilitar a compreensãodas informações que fazem um estado transitar para outro. Abaixo as imagens referentes aos diagramas deste projeto:

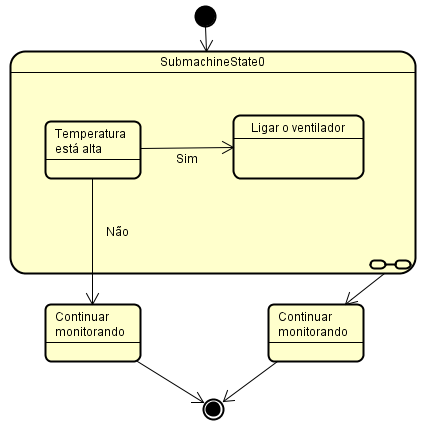


Imagem 19: Diagrama de Máquina de Estado 1

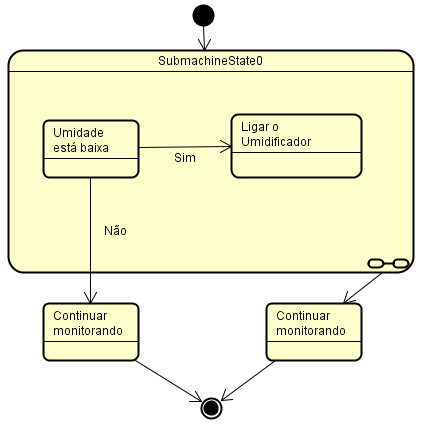


Imagem20: Diagrama de Máquina de Estado 2

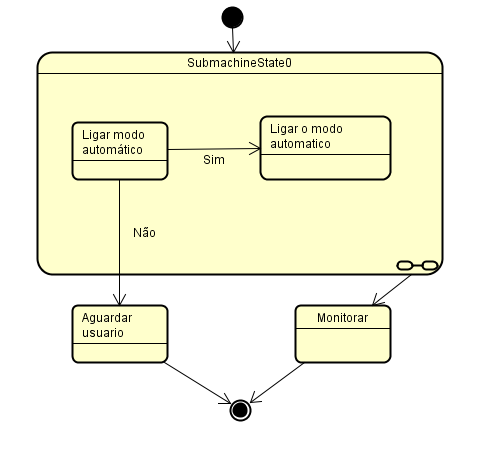


Imagem21: Diagrama de Máquina de Estado 3

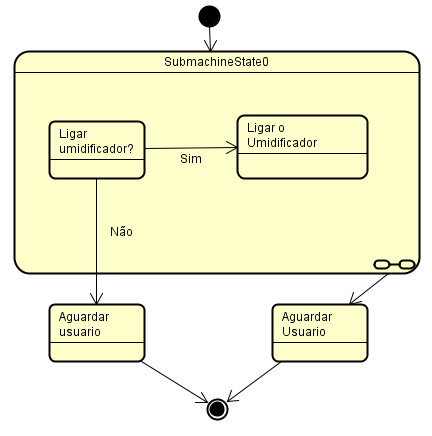


Imagem22: Diagrama de Máquina de Estado 4

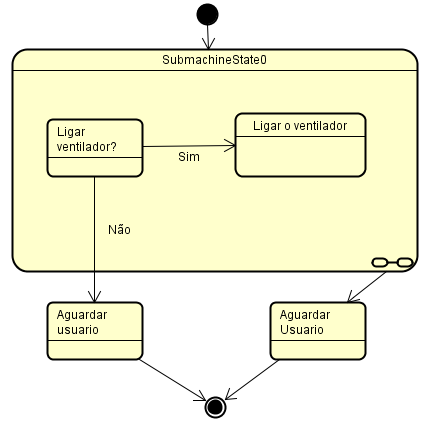


Imagem23: Diagrama de Máquina de Estado 5

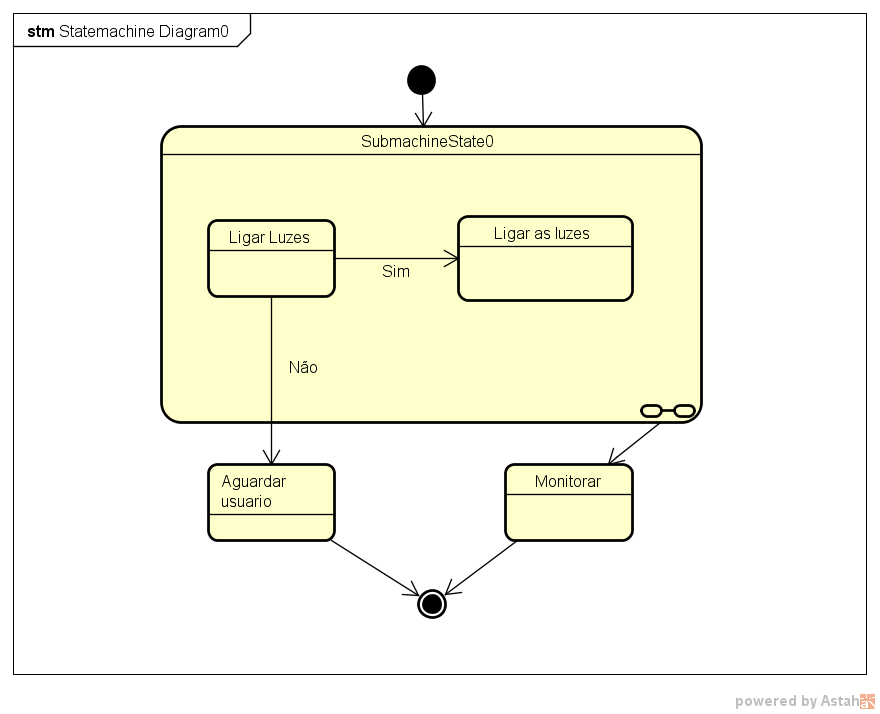


Imagem 24: Diagrama de Máquina de Estado 6

**7 Prototipação de Telas**

A prototipação de telas serve para nortear o programador ao construir o software, e também para validar o que será utilizado, o que pode ser otimizado, e o que pode ser retirado. Segue abaixo as imagens referentes a solução apresentada nesse projeto:

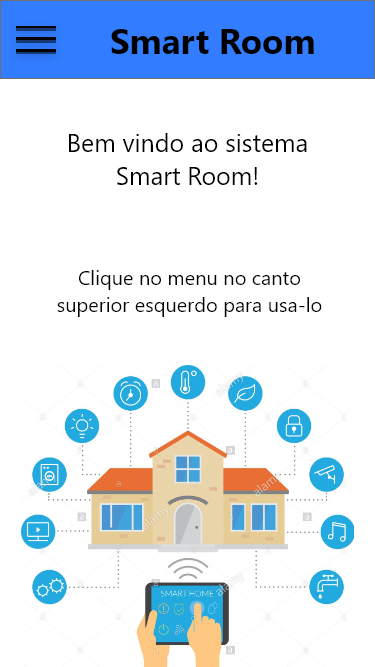


Imagem 25: Prototipação –Tela Inicial

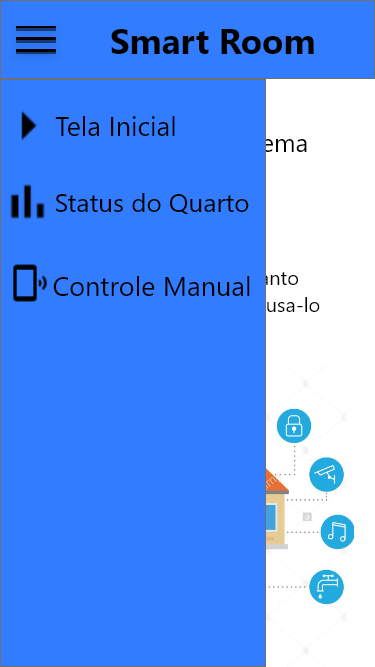


Imagem 26: Prototipação – Menu

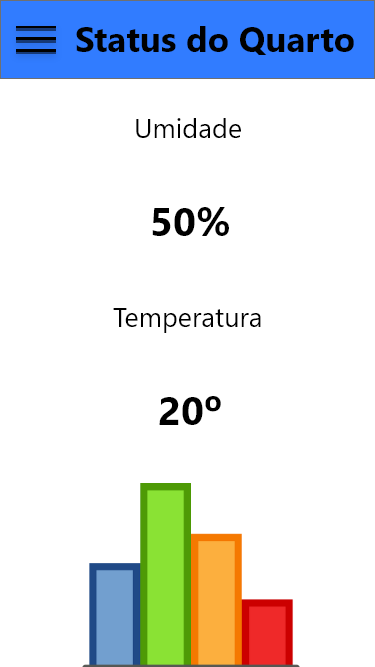


Imagem 27: Prototipação – Status



Imagem 28: Prototipação – Tela de comandos

1. Discente do curso de Bacharelado em Sistemas de Informação [↑](#footnote-ref-1)
2. Discente do curso de Bacharelado em Sistemas de Informação [↑](#footnote-ref-2)
3. Docente do curso de Bacharelado em Sistemas de Informação [↑](#footnote-ref-3)