Projeto Web IR Remote Control

Documento de Visão do Produto

Aluno: Johannes Dos Reis

03/2020

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 20/03/2020 | 1.0.0 | Versão inicial do documento. | Johannes Dos Reis |
| 03/03/2020 | 1.0.1 | Adicionados casos de uso, alteração de requisitos relacionados ao gerenciamento de dispositivos e adição do requisito executar ações. | Johannes Dos Reis |
| 16/04/2020 | 1.0.2 | Detalhamento melhorado dos requisitos e regras de negócio | Johannes Dos Reis |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Atenção! Para cada atualização no documento coloque a data e nova versão. Fique livre para escolher o “número” da versão.. Aqui é apenas uma sugestão!**

ÍNDICE

[Projeto [nome do projeto] 5](#_Toc522343592)

[1 INTRODUÇÃO 5](#_Toc522343593)

[Propósito do documento de visão do produto 5](#_Toc522343594)

[Características do produto 5](#_Toc522343595)

[Exclusões do projeto 6](#_Toc522343596)

[Necessidades identificadas 6](#_Toc522343597)

[Principais Stakeholders 6](#_Toc522343598)

[Concepção do sistema 7](#_Toc522343599)

[Identificação dos requisitos 7](#_Toc522343600)

[Prioridade dos requisitos 7](#_Toc522343601)

[2 REQUISITOS 8](#_Toc522343602)

[Requisitos Funcionais 8](#_Toc522343603)

[RF01 – Manter Pessoa 8](#_Toc522343604)

[Requisitos Não-Funcionais 9](#_Toc522343605)

[Requisitos de Segurança 9](#_Toc522343606)

[Requisitos de Interface 9](#_Toc522343607)

[Requisitos de Operacionais 9](#_Toc522343608)

[Requisitos de Confiabilidade 10](#_Toc522343609)

[3 ESPECIFICAÇÃO 11](#_Toc522343610)

[Diagrama de Casos de Uso (UC) 11](#_Toc522343611)

[Diagrama de casos de uso geral 11](#_Toc522343612)

[Usuários do sistema 11](#_Toc522343613)

[Especificação do caso de uso 12](#_Toc522343614)

[Manter Pessoa 12](#_Toc522343615)

[Realizar Login 14](#_Toc522343616)

[Diagrama de atividades 16](#_Toc522343617)

[Manter pessoa 16](#_Toc522343618)

[Manter pessoa tipo 17](#_Toc522343619)

[Manter pessoa login 17](#_Toc522343620)

[4 REGRAS DE NEGÓCIO 19](#_Toc522343622)

[4.1      RN001 - ..... 19](#_Toc522343623)

[5 PROJETO 19](#_Toc522343624)

[Diagrama de classes 19](#_Toc522343625)

[Modelo Entidade Relacionamento (MER) 20](#_Toc522343626)

[6 REFERÊNCIAS 22](#_Toc522343627)

[Referências ao modelo do documento 22](#_Toc522343628)

## Projeto [nome do projeto]

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Documento de Visão do Produto***

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# INTRODUÇÃO

## Propósito do documento de visão do produto

*O propósito deste documento é apresentar a descrição dos serviços e funções que o sistema a ser desenvolvido deve prover, bem como as suas restrições de operação e propriedades gerais, a fim de ilustrar uma descrição detalhada do sistema para um auxílio durante as etapas de análise, projeto e testes. O documento especifica todos os requisitos funcionais e não funcionais do sistema e foi preparado levando-se em conta as funcionalidades levantadas durante a fase de concepção do sistema. Também serão apresentados alguns diagramas da UML, referente à parte de projeto do sistema.*

**Atenção: as características do produto a seguir apresentadas são apenas para exemplo! Fique livre para adaptações!**

## Características do produto

* O projeto consiste na construção de um sistema de monitoramento e controle de dispositivos de ar condicionado remotamente;
* A solução possuirá uma página web onde será possível ver a temperatura e humidade do ambiente monitorado;
* A interface gráfica web será desenvolvida com Nuxt um Framework Vue.js.
* Possibilidade de executar as ações instantaneamente ou agenda-las para serem executados posteriormente;
* A solução também permitirá que os eventos agendados sejam persistidos.
* O backend será desenvolvido em Node.js, e estará rodando em uma Raspberry Pi 2.
* O ar condicionado será controla através de um componente emissor de Infravermelho, o qual está conectada a Raspberry através das portas GPIO.
* A temperatura e humidade do ambiente será medida através de um sensor DHT11, que estará conectado a Raspberry pelas portas GPIO.
* O acesso ao sistema será feito através de um login.
* O sistema deve funcionar via internet;
* O banco de dados a ser utilizado deve será o Firebase;

## Exclusões do projeto

*Às vezes a equipe de gerenciamento de projetos declara como exclusões do escopo produtos ou serviços que a equipe do projeto não fornecerá, mas que é necessária para o projeto e que o Patrocinador ou alguma outra parte interessada precisa prover.*

*Dentre as saídas do processo "Definir o escopo" encontra-se: "Exclusões do projeto. Identifica de modo geral o que é excluído do projeto. Declarar explicitamente o que está fora do escopo do projeto ajuda no gerenciamento das expectativas das partes interessadas.".*

*PROJECT MANAGEMENT INSTITUTE.* ***Um Guia de Conhecimentos em Gerenciamento de Projetos - Guia Pmbok****. 5ª. ed. [S.l.]: Saraiva, 2014. Pg. 115x*

Não fazem parte do escopo do projeto:

* Instalação e configuração do ambiente tecnológico do cliente.
* Treinamento de instalação, configuração, administração e utilização do sistema;
* Não é garantido o controle de dispositivos que não estão na lista de dispositivos suportados.

**Atenção: as necessidades e stakeholders apresentadas a seguir são APENAS exemplos! Fique livre para adaptações.**

## Necessidades identificadas

*Abaixo serão descritas as necessidades dos usuários para o desenvolvimento do sistema e listadas as funcionalidades relacionadas.*

|  |  |  |
| --- | --- | --- |
| **Necessidade** | **Descrição** | **Requisitos Funcionais** |
| Gerenciar agendamento de ações | Necessidade utilizada para manter o agendamento de ações. O agendamento deverá possuir um cliente, uma ação e uma data e um horário. | - Manter agendamento  - Manter ação  - Manter horário e data |
| Gerenciar dispositivos | Necessidade utilizada para manter os dispositivos e suas ações. | - Manter Dispositivo |
| Mostrar temperatura e humidade | Necessidade utilizada para exibir para o usuário a temperatura e humidade em tempo real. | - Mostrar temperatura e humidade |
| Gerenciar Cliente | Necessidade utilizada para manter as informações dos clientes e usuários do sistema. | - Manter cliente  - Criar o auto cadastro |
| Realizar Login | Necessidade utilizada para controlar o acesso ao sistema. | - Efetuar login no sistema  - Efetuar logout |

## Principais Stakeholders

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificação** | **Stakeholder** | **Responsabilidade** | **No projeto** |
| Desenvolvedor de Software e Hardware | Johannes | Desenvolver o software e hardware | Responsável por codificar e montar a solução. |

## Concepção do sistema

*“As técnicas de levantamento de requisitos têm por objetivo superar as dificuldades relativas a esta fase. Todas as técnicas possuem um conceito próprio e suas respectivas vantagens e desvantagens, que podem ser utilizadas em conjunto pelo analista”.*

*Disponível: < http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/9151>. Acesso 02 ago. 2017.*

Não foram usados cinco métodos para que pudessem ser obtidos os requisitos do sistema.

**Atenção: Escolher um ou mais métodos que você utilizou para a obtenção dos requisitos do sistema.**

## Identificação dos requisitos

*Cada requisito será unicamente identificado no formato [tipoRequisito.numero]. Para requisitos funcionais, o código do tipo de requisito será RF, e para requisitos não funcionais, RNF. Um número será assinalado a cada requisito de forma incremental, na ordem que forem mencionados neste documento.*

## Prioridade dos requisitos

*Para estabelecer a prioridade do requisito, será utilizado uma escala que inicia com numero [1] até [5]. Quanto menor o número, maior a prioridade do requisito.*

# REQUISITOS

*“Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços oferecem e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações.”*

*SOMMERVILLE, I. Engenharia de Software. 9ª. ed. São Paulo: Person Education do Brasil, 2011 P. 57.*

## Requisitos Funcionais

*“Os requisitos funcionais descrevem a funcionalidade ou os serviços que se espera que o sistema realize em benefício dos usuários. Eles variam de acordo com o tipo de software em desenvolvimento, com usuários e com o tipo de sistema que está sendo desenvolvido. Requisitos funcionais podem ser expressos de diversas maneiras e, como já foi dito acima, em diferentes níveis de detalhamento. Os requisitos funcionais de usuários definem recursos específicos que devem ser fornecidos pelo sistema.”*

*SOMMERVILLE, I.* ***Engenharia de Software****. 9ª. ed. São Paulo: Person Education do Brasil, 2011. p. 59.*

**Atenção: Cada requisito funcional descrito a seguir deverá ser implementado no software**

**Os requisitos que descrevem os aspectos funcionais do sistema são apresentados a seguir: SÃO APENAS EXEMPLOS ORIENTATIVOS!**

### RF01 – Manter usuário

**Importância:** [ X ] crítico [ ] importante [ ] útil**Dependência do(s) requisito(s):** []  
**Relacionamento com outro(s) requisito(s):** []  
**Priorização:** [ 1 ]

**PROBLEMAS/NECESSIDADES IDENTIFICADAS**

Para acessar ao sistema, o usuário deve realizar um cadastro, passando suas informações pessoais.

**SOLUÇÃO**

Criar manutenção de usuários, com possibilidade de inserção, alteração e exclusão.  
Dados que deverão ser considerados para o cadastro:

→ Nome   
→ E-mail  
→ Senha

**RESTRIÇÕES / EXCEÇÕES**

- Um e-mail pode ser cadastrado apenas uma vez.  
- Os campos e-mail, senha e nome deverão ser obrigatoriamente informados.

### RF02 – Manter dispositivo

**Importância:** [ X ] crítico [ ] importante [ ] útil**Dependência do(s) requisito(s):** []  
**Relacionamento com outro(s) requisito(s):** []  
**Priorização:** [ 1 ]

**PROBLEMAS/NECESSIDADES IDENTIFICADAS**

Para controlar diferentes tipos de equipamentos o sistema deve ter uma lista de dispositivos compatíveis com o sistema, os quais podem ser selecionados pelo usuário.

**SOLUÇÃO**

Criar uma lista de dispositivos suportados com suas ações já configuradas, e persisti-la em banco de dados.

O frontend exibira uma lista dos dispositivos compatíveis permitindo ao usuário selecionar qual dispositivo que deseja controlar.

**RESTRIÇÕES / EXCEÇÕES**

### RF03 – Mostrar a temperatura e umidade

**Importância:** [ ] crítico [ X ] importante [ ] útil**Dependência do(s) requisito(s):** []  
**Relacionamento com outro(s) requisito(s):** [RF02]  
**Priorização:** [ 2 ]

**PROBLEMAS/NECESSIDADES IDENTIFICADAS**

O usuário deve ter acesso aos valores em tempo real de temperatura e umidade do ambiente controlado.

**SOLUÇÃO**

O frontend deve mostrar em tempo real os valores de temperatura e umidade presentes no banco de dados em uma página web, e atualiza-los caso haja alguma mudança no banco de dados sem a necessidade de intervenção do usuário para que os valores sejam atualizados.

**RESTRIÇÕES / EXCEÇÕES**

### RF04 – Executar ações

**Importância:** [ ] crítico [ X ] importante [ ] útil**Dependência do(s) requisito(s):** [RF02]  
**Relacionamento com outro(s) requisito(s):** []  
**Priorização:** [ 2 ]

**PROBLEMAS/NECESSIDADES IDENTIFICADAS**

O usuário deve ser capaz executar as ações de controle sobre um dispositivo escolhido.

**SOLUÇÃO**

O frontend deverá mostrar uma página de controle, onde por meio de botões será possível enviar ações, de um dispositivo pré-selecionado anteriormente pelo usuário, para o backend, e este ao receber estas ações deverá mandar um sinal IR através do LED emissor IR conectado com o código corresponde a ação requisitada, assim executando a ação no dispositivo de ar condicionado desejado.

**RESTRIÇÕES / EXCEÇÕES**

### RF05 – Manter agendamento

**Importância:** [ ] crítico [ ] importante [ x ] útil**Dependência do(s) requisito(s):** [RF04, RF02]  
**Relacionamento com outro(s) requisito(s):** []  
**Priorização:** [ 3 ]

**PROBLEMAS/NECESSIDADES IDENTIFICADAS**

O usuário deve ter acesso aos eventos agendados para os dispositivos cadastrados, sendo possível adicionar, excluir e alterar estes eventos.

**SOLUÇÃO**

O frontend deve criar uma lista dos eventos agendados em uma página web, onde será possível visualizar, adicionar, alterar ou excluir os eventos agendados para o dispositivo selecionado previamente estas ações serão enviadas para o backend.

O backend ao receber estar requisições deve atualizar o banco de dados, e agendar a execução das ações no o horário e data especificados.

Dados que deverão ser considerados para o agendamento:

→ horário  
→ data  
→ ação

**RESTRIÇÕES / EXCEÇÕES**

## Requisitos Não-Funcionais

*“Os requisitos não funcionais são aqueles que não dizem respeito diretamente às funcionalidades fornecidas pelo sistema. Podem estar relacionados a propriedades de sistemas emergentes, como confiabilidade, tempo de resposta, espaço em disco, desempenho e outros atributos de qualidade do produto. Às vezes podem dizer respeito ao sistema como um todo. Isso significa que na maioria das vezes eles são mais importantes que os requisitos funcionais individuais. Se uma falha em cumprir um requisito funcional pode comprometer parte do sistema, uma falha em cumprir um requisito não funcional pode tornar todo o sistema inútil”.*

*SOMMERVILLE, I.* ***Engenharia de Software****. 9ª. ed. São Paulo: Person Education do Brasil, 2011. p. 60.*

**Atenção: Cada requisito não funcional descrito a seguir deverá ser implementado no software. Pense em como fará a comprovação do requisito implementado.**

x

Os requisitos que descrevem os aspectos não-funcionais do sistema são apresentados a seguir: ATENÇÃO: OS REQUISITOS A SEGUIR SÃO APENAS SUGESTÕES! VOCÊ PODE INCLUIR OU RETIRAR QUALQUER UM AO SEU LIVRE CRITÉRIO!

### Requisitos de Usabilidade

|  |  |  |
| --- | --- | --- |
| Ident. | Descrição | Prioridade |
| RNF/INT-01 | O sistema terá versões de layout para diferentes tamanhos de tela. | Baixa |

### Requisitos de Interoperabilidade

|  |  |  |
| --- | --- | --- |
| Ident. | Descrição | Prioridade |
| RNF/INT-01 | O sistema deverá permitir que o usuário utilize uma conta do Google para obter acesso ao sistema. | Média |

### Requisitos de Segurança

|  |  |  |
| --- | --- | --- |
| Ident. | Descrição | Prioridade |
| RNF/SEG-01 | O usuário só poderá ter acesso e utilizar o sistema após a realização de ser autenticado através de um login. | Alta |

# REGRAS DE NEGÓCIO

*“As regras de negócio são****restrições/premissas****necessárias para o negócio “acontecer”.*

*Ex:*

*– O produto somente será entregue ao cliente que realizar o pagamento.*

*– Somente serão aceitos pagamentos em dinheiro. Não serão aceitos cartões de crédito, cartão de débito ou cheque.*

*– Clientes que comprarem 4 produtos ganharão um brinde.*

*Regras responsáveis pela execução do negócio da aplicação.*

|  |  |  |
| --- | --- | --- |
| Identificador | Descrição | Prioridade |
| RN001 | Os de temperatura e humidade devem ser lidos pelo sensor a cada cinco minutos e o backend deve atualizar estes dados no banco de dados. | Alta |
| RN002 | Remover automaticamente do banco de dados um agendamento cuja ação já foi concluída a mais de uma semana. | Média |
| RN003 | O sistema garante o funcionamento apenas com os dispositivos compatíveis presentes na lista. | Alta |
| RN004 | Tentativas de agendamento com dados de hora e data inferiores a data e horário em que a tentativa foi executada não serão executadas. | Médio |

**Atenção: Cada regra de negócio descrita deverá ser implementada no software**

# ESPECIFICAÇÃO

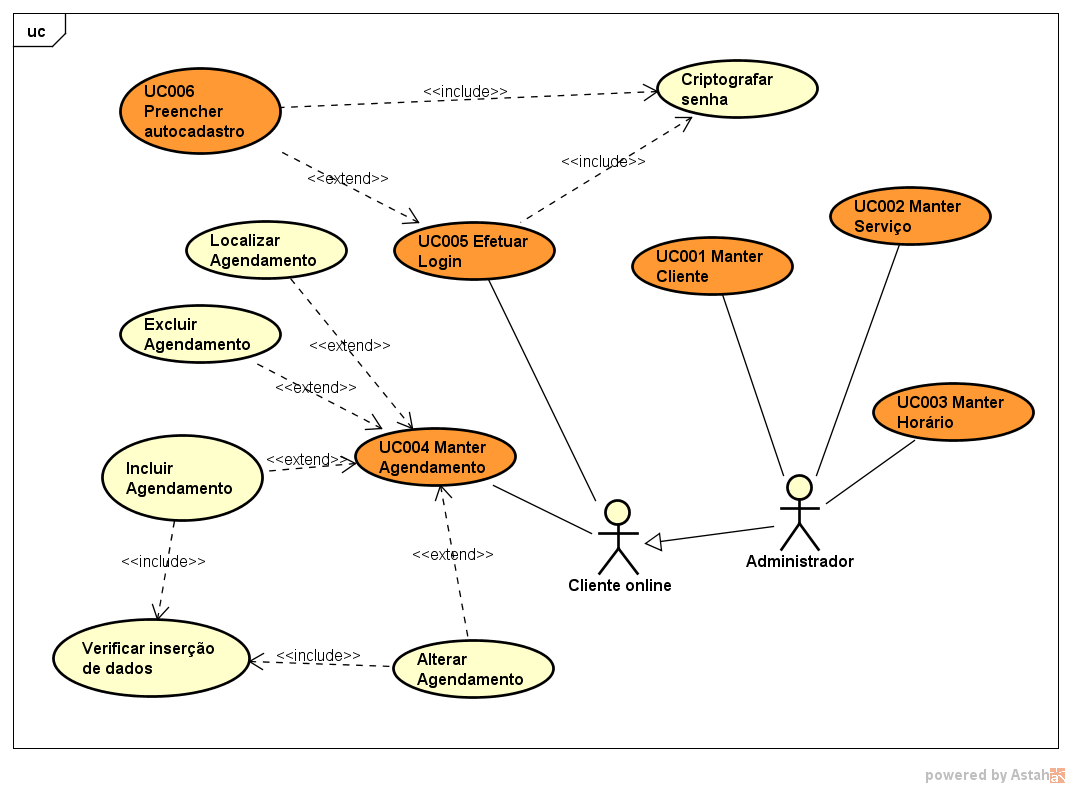
## Diagrama de Casos de Uso (UC)

O diagrama de casos de uso, expresso em UML (*Unified Modeling Language*), expressa os requisitos funcionais do sistema na forma de casos de uso. Segundo o RUP (*Rational Unified Process*), para cada requisito funcional tem-se um caso de uso. A descrição textual detalhada dos requisitos funcionais, seus fluxos de atividades e requisitos não funcionais associados pode ser encontrada na próxima seção.

Na figura abaixo mostramos a representação gráfica em UML dos casos de uso do sistema.

## Diagrama de casos de uso geral

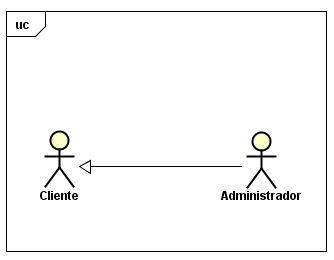
Abaixo é apenas um exemplo!!!



## Usuários do sistema

**Administrador:** realiza manutenção nas entidades que abastecem o uso do sistema;

**Cliente:** realiza atividades relacionadas ao uso do sistema pelo cliente;



## Especificação do caso de uso

*“Uma especificação de caso de uso é um documento funcional, ou seja, ele descreve as funcionalidades que um sistema deve ter. No caso de uso é apresentado como o sistema deve reagir ao conjunto de entradas que recebe qual o comportamento desse sistema e as saídas que serão geradas.”*

*Disponível: < http://www.devmedia.com.br/elementos-da-especificacao-de-casos-de-uso/34391>. Acesso 02 ago. 2017.*

Dentre os casos de uso do sistema mostrados no diagrama de casos de uso, foram escolhidos cinco para serem detalhados e trabalhados nas fases de análise e projeto do sistema.

### Manter Usuário

|  |  |  |  |
| --- | --- | --- | --- |
| **[UC01]** | | | |
| Nome: | | Manter Usuário | |
| Descrição: | | Este caso de uso descreve a ação para o usuário criar seu usuário. | |
| Atores: | | Usuário | |
| Prioridade: | | [ 1 ] | |
| Requisito associado: | | | RF01 |
| Entradas e pré-condições: | | | * Acesso ao sistema. |
| Saídas e pós-condições: | | | * Usuário cadastrado. |
| Fluxos de eventos | | | |
| Fluxo principal: | 1. O usuário informa nome. 2. O usuário informa e-mail. 3. O usuário informa senha. 4. O usuário executa a ação de cadastrar, clicando no botão “cadastrar”. **[FE01] [FE02]** 5. O sistema realiza a ação de cadastrar usuário. 6. O sistema redireciona o usuário a página inicial. | | |
| Fluxo de exceção: | **FE01 – Campo obrigatório não informado**  O sistema verifica se os campos nome, e-mail, senha e saldo inicial foram preenchidos, caso contrário ele exibe a mensagem com o campo obrigatório que não foi informado.  **FE02 – E-mail já cadastrado**  O sistema verifica se o e-mail informado já está cadastrado, se estiver cadastrado ele exibe a mensagem que o e-mail já está cadastrado. | | |
| Esboço de tela: |  | | |
| Cenários: | **CEN01 – Usuário completa todos os dados corretamente**  O usuário informa todos os dados necessários corretamente para realizar o cadastro.  **CEN02 – Usuário completa os dados parcialmente**  O usuário informa os dados parcialmente. | | |
| Regras de Interação: |  | | |

### Realizar Login

|  |  |  |
| --- | --- | --- |
| **[UC02]** | | |
| Nome: | Realizar Login | |
| Descrição: | Este caso de uso descreve a ação para o usuário entrar no sistema. | |
| Atores: | Usuário | |
| Prioridade: | [ 1 ] | |
| Entradas e pré-condições: | | * Usuário deve estar cadastrado no sistema. |
| Saídas e pós-condições: | | * Usuário logado. |
| Fluxos de eventos | | |
| Fluxo principal: | 1. O usuário informa e-mail cadastrado. **[FA02]** 2. O usuário informa a senha. 3. O usuário executa a ação de entrar, a partir do botão Entrar. **[FE01] [FE02] [FE03] [FA01]** 4. O sistema permite ao usuário entrar no sistema. | |
| Fluxo alternativo: | **FA01 – Cancelar operação**  Caso o usuário selecione a opção de cancelar alguma operação.  O sistema volta para o fluxo principal.  **FA01 – Login com Google**   1. O usuário clica no botão Login com Google 2. O sistema redireciona o usuário a uma pagina de login do Google 3. Após o usuário fazer login através do Google o sistema permite o usuário entrar no sistema. **[FA01]** | |
| Fluxo de exceção: | **FE01 – Campo obrigatório não informado**  O sistema verifica se os campos login e senha foram preenchidos, caso contrário ele exibe a mensagem com o campo obrigatório que não foi informado.  **FE02 – E-mail incorreto**  Caso o dado informado não esteja cadastrado, o sistema exibe a mensagem “E-mail incorreto” e mantém o usuário na tela de login. **[FA03]**  **FE03 – Senha incorreta**  Se a senha não for informada corretamente, o sistema exibe a mensagem “Digite sua senha novamente” e mantém o usuário na tela de login.  **FE04 – Login incorreto**  Caso o sistema de login do Google retorne erro, o sistema exibe a mensagem “Dados inválidos”. | |
| Esboço de tela: |  | |
| Cenários: | **CEN01 – Usuário com dados corretos para entrar**  O usuário informa os dados corretos para entrar no sistema.  **CEN02 – Usuário com e-mail correto e senha incorreta**  O usuário informa seu e-mail corretamente e a senha incorreta.  **CEN03 – Usuário com e-mail e a senha incorreta**  O usuário informa seu e-mail e senha incorreta.  **CEN04 – Usuário não informa nenhum dado**  O usuário não informa nenhum dado. | |
| Regras de Interação: |  | |

### Alterar dispositivo

|  |  |  |  |
| --- | --- | --- | --- |
| **[UC01]** | | | |
| Nome: | | Alterar dispositivo | |
| Descrição: | | Este caso de uso descreve a ação para o usuário alterar o dispositivo controlado. | |
| Atores: | | Usuário | |
| Prioridade: | | [ 1 ] | |
| Requisito associado: | | | RF02 |
| Entradas e pré-condições: | | | * Acesso ao sistema, * Usuário autenticado. |
| Saídas e pós-condições: | | |  |
| Fluxos de eventos | | | |
| Fluxo principal: | 1. Usuário clica no na seta para baixo na parte superior da tela. 2. Na lista que o sistema retorna o usuário escolhe um dispositivo. **[FA01]** 3. O sistema altera o dispositivo controlado. | | |
| Fluxo alternativo: | **FA01 – Cancelar operação**  Caso o usuário selecione a opção de cancelar alguma operação.  O sistema volta para o fluxo principal. | | |
| Esboço de tela: |  | | |
| Cenários: | **CEN01 – O dispositivo é alterado**  O usuário informa o dispositivo desejado e o sistema altera o dispositivo controlado.  **CEN01 – O dispositivo não é alterado**  Caso o usuário não informar o dispositivo a ser controlado o sistema não altera o dispositivo controlado . | | |
| Regras de Interação: |  | | |

### Manter Agendamento

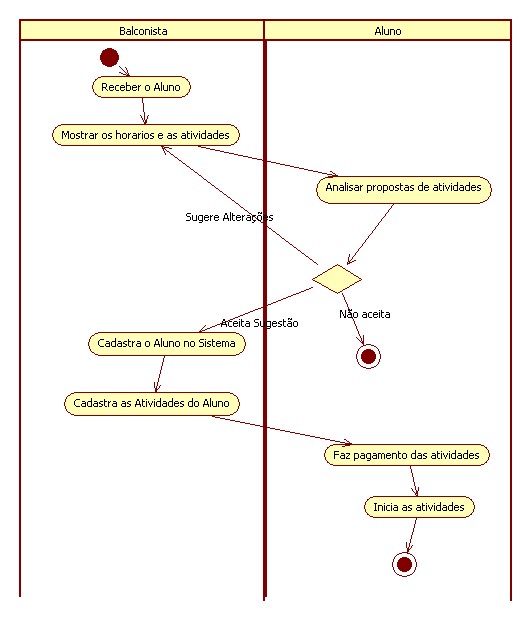
|  |  |  |  |
| --- | --- | --- | --- |
| **[UC01]** | | | |
| Nome: | | Manter Agendamento | |
| Descrição: | | Este caso de uso descreve a ação para o usuário realizar manutenções em seus agendamentos. | |
| Atores: | | Usuário | |
| Prioridade: | | [ 1 ] | |
| Requisito associado: | | | RF04 |
| Entradas e pré-condições: | | | * Acesso ao sistema, * Usuário autenticado. |
| Saídas e pós-condições: | | |  |
| Fluxos de eventos | | | |
| Fluxo principal: | 1. Usuário clica no botão ‘+’ mostrado na parte superior da lista de agendamentos. **[FE01], [FE02]** 2. O sistema abre um formulário para preenchimento dos dados do agendamento. 3. O usuário informa a data 4. O usuário informa o horário 5. O usuário informa a ação. 6. O usuário executa a ação gerar agendamento, clicando no botão “Agendar”. **[FE01]** | | |
| Fluxo de exceção: | **FE01 – Campo obrigatório não informado**  O sistema verifica se os campos data, horário e ação foram preenchidos, caso contrário ele exibe a mensagem com o campo obrigatório que não foi informado. | | |
| Fluxo alternativo: | **FA01 – Alterar Agendamento**  Caso o usuário deseje alterar um agendamento, o mesmo poderá alterá-lo utilizando os passos a seguir:   1. O usuário seleciona o botão editar (representado por um ícone de lápis) na lista de agendamentos mostrada pelo sistema. 2. O sistema abre o formulário do agendamento. 3. O usuário altera a data se o mesmo desejar. 4. O usuário altera o horário se o mesmo desejar. 5. O usuário altera a ação se o mesmo desejar. 6. O usuário executa a ação alterar agendamento, clicando no botão” Confirmar”. **[FA03]**   **FA02 – Deletar Agendamento**  Caso o usuário deseje deletar um agendamento, o mesmo poderá alterá-lo utilizando os passos a seguir:   1. O usuário seleciona o botão excluir (representado por um ícone de lixeira) na lista de agendamentos mostrada pelo sistema. 2. O sistema pede confirmação se o usuário deseja realizar essa ação. **[FA03]** 3. O sistema exclui o agendamento.   **FA03 – Cancelar operação**  Caso o usuário selecione a opção de cancelar alguma operação.   1. O sistema volta para o fluxo principal. | | |
| Esboço de tela: |  | | |
| Cenários: | **CEN01 – Usuário completa todos os dados corretamente**  O usuário informa todos os dados necessários corretamente para realizar o cadastro.  **CEN02 – Usuário completa os dados parcialmente**  O usuário informa os dados parcialmente. | | |
| Regras de Interação: |  | | |

## Diagrama de atividades

*“O diagrama de atividade é um diagrama definido pela UML, e representa os fluxos conduzidos por processamentos. É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Comumente isso envolve a modelagem das etapas sequenciais em um processo computacional. Este diagrama representa uma alternativa de detalhamento diagramático de um caso de uso complexo”.*

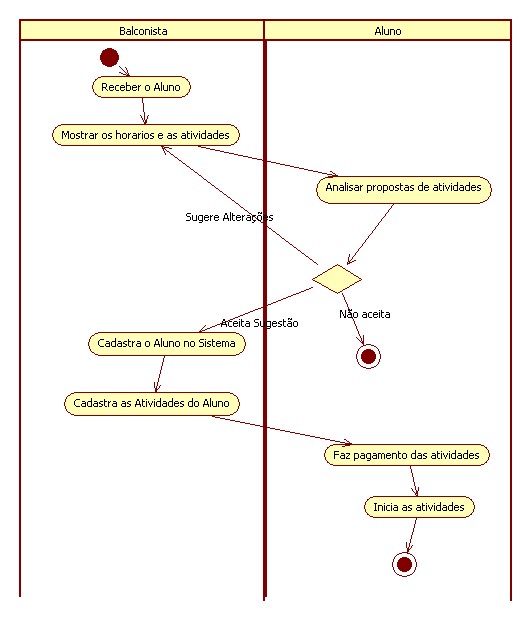
### Manter pessoa

Diagrama que descreve as atividades executadas no caso de uso de referência.



### Manter pessoa tipo

Diagrama que descreve as atividades executadas no caso de uso de referência.



### Manter pessoa login

Diagrama que descreve as atividades executadas no caso de uso de referência.

## 

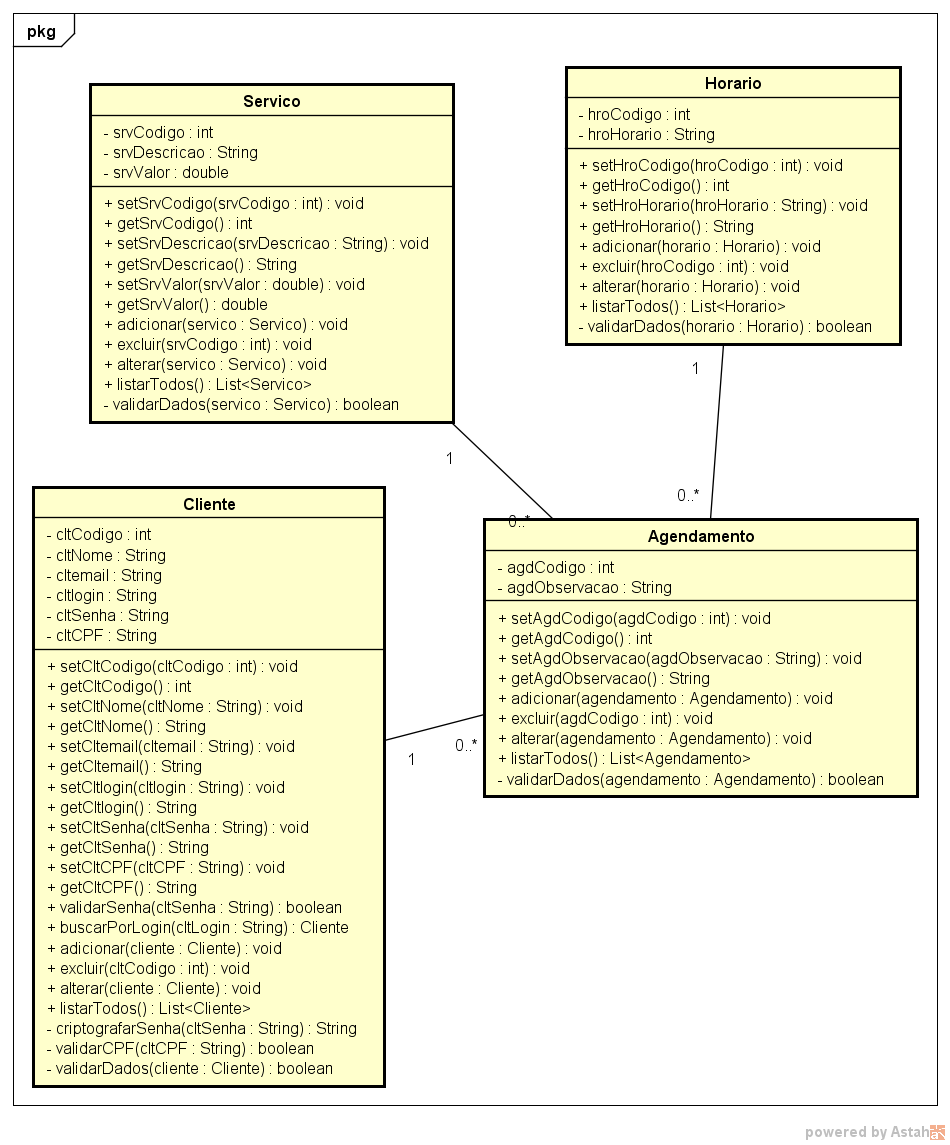
# PROJETO

## Diagrama de classes

“O diagrama de classes demonstra a estrutura estática das classes de um sistema onde estas representam as "coisas" que são gerenciadas pela aplicação modelada.

O diagrama de classes é considerado estático já que a estrutura descrita é sempre válida em qualquer ponto do ciclo de vida do sistema”.

Na figura abaixo mostramos a representação gráfica em UML das classes do sistema.

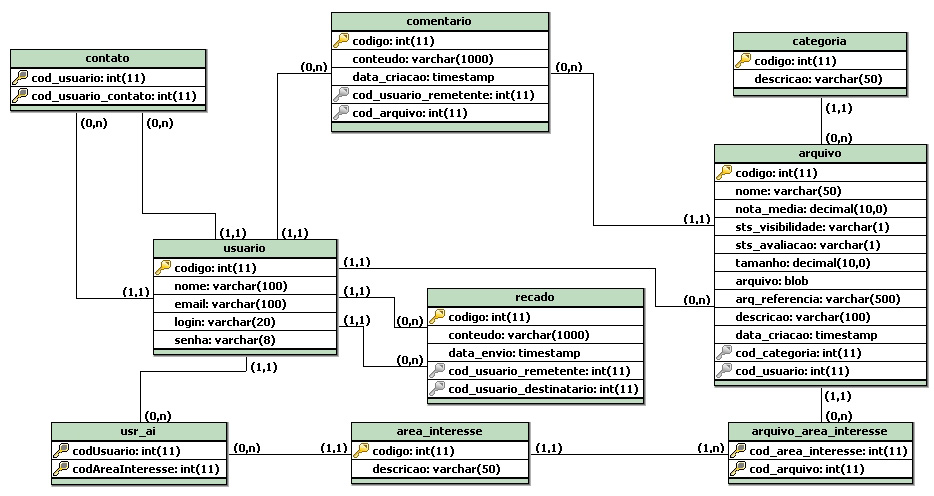


## Modelo Entidade Relacionamento (MER)

*“O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).*

*Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais.”*

Na figura abaixo mostramos a representação gráfica do Modelo ER do sistema.



# REFERÊNCIAS

## Referências ao modelo do documento

Esta subseção apresenta as referências aos documentos que utilizamos no auxílio à construção deste documento de requisitos.

* Periódicos da CAPES - <http://www.periodicos.capes.gov.br/>
* Referências da Disciplina Engenharia de Software Educativo - <http://www.cin.ufpe.br/~asg/nova_pagina_1.htm>
* Página da Disciplina Especificação de Requisitos e Validação de Sistemas- <http://www.cin.ufpe.br/~if716/>
* Experiência na construção de outros documentos.