Projeto [Validador de CPF/CNPJ]

Documento de Visão do Produto

Aluno: [Carlos Eduardo Bertoglio e João Pedro Fortunato]

[Março /2021]

## Projeto [Validador de CPF/CNPJ]

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

***Documento de Visão do Produto***

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

# INTRODUÇÃO

**Link para o trello:**

[**https://trello.com/invite/b/tv3fGvvy/0b345693c58db345c6fdd924327a672f/projeto**](https://trello.com/invite/b/tv3fGvvy/0b345693c58db345c6fdd924327a672f/projeto)

**Link para o GitHub:**

[**https://github.com/JFortunat/EngSoftII.git**](https://github.com/JFortunat/EngSoftII.git)

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 28/08/2020 | 1.0.0 | Versão inicial do documento | Carlos Eduardo |
| 15/09/2020 | 1.0.0 | Início da Prototipação | Carlos Eduardo |
| 25/11/2020 | 1.0.1 | Versão atualizada do documento | Carlos Eduardo |
| 09/12/2020 | 1.0.2 | Versão atualizada do documento | Carlos Eduardo |
| 09/03/2021 | 1.0.3 | Versão atualizada do documento | Carlos e Joao |
| 09/03/2021 | 1.0.1 | Versão Atualizada da Prototipação | Carlos e Joao |
| 15/06/2021 | 1.0.2 | Versão Atualizada da Prototipação | Carlos e Joao |
| 27/06/2021 | 1.0.3 | Versão Final da Prototipação | Carlos e Joao |
| 27/06/2021 | 1.0.4 | Versão Final da Documentação | Carlos e Joao |

## Propósito do documento de visão do produto

*O propósito deste documento é apresentar a descrição dos serviços e funções que o Aplicativo 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.*

## Características do produto

* O projeto consiste na construção de uma ferramenta de validação de CPF/CNPJ.
* A solução deverá permitir que os registros de agendamento sejam persistidos;
* Para que um usuário (cliente) possa acessar o sistema, ele deve preencher um auto cadastro. Por questão de segurança, para acessar o sistema, o usuário deve efetuar um login para garantir o controle dos CPF/CNPJ testados.
* Também será necessária uma rotina de criptografia de dados para os caracteres de senha dos usuários, caracteres estes que poderão ser gravados nos registros do cliente;
* O aplicativo deve ser homologado para uso em qualquer sistema operacional.
* O Banco de dados usado para o controle de login será o PostgresSQL.

## 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”.*

Foram usados três métodos para que pudessem ser obtidos os requisitos do sistema:

* Entrevista com usuários
* Prototipação das telas

# REQUISITOS

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

## Requisitos Funcionais

*“Os requisitos funcionais descrevem a funcionalidade ou os serviços que se espera que o aplicativo realize em benefício dos usuários. Eles variam de acordo com o tipo de software em desenvolvimento, com usuários e com o tipo de aplicativo 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.”*

**RF-01**: O sistema deve validar a numeração do CPF/CNPJ

**Importância:** [X] crítico [ ] importante [ ] útil

**SOLUÇÃO:**

Criar uma tela onde o usuário possa digitar CPF/CNPJ para poder efetuar o teste.

## Requisitos Não-Funcionais

*“Os requisitos não funcionais são aqueles que não dizem respeito diretamente às funcionalidades fornecidas pelo aplicativo. 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”.*

* **Requisitos de Segurança:**

**RNF/SEG-01:** O usuário autorizado deverá efetuar *login* no sistema para poder realizar as operações de testes.

* **Requisitos de interface:**

**RNF/INT-01:** O sistema deve ter uma interface visual simples.

**RNF/INT-02:** O sistema deve manter uma interface que proporcione alta produtividade, podendo testar um CPF e um CNPJ ao mesmo tempo.

* **Requisitos Operacionais:**

**RNF/OPE-01:** O sistema deve ser desenvolvido em Java.

**RNF/OPE-02:** O Banco de dados utilizado deve ser o PostgresSQL.

# ESPECIFICAÇÃO

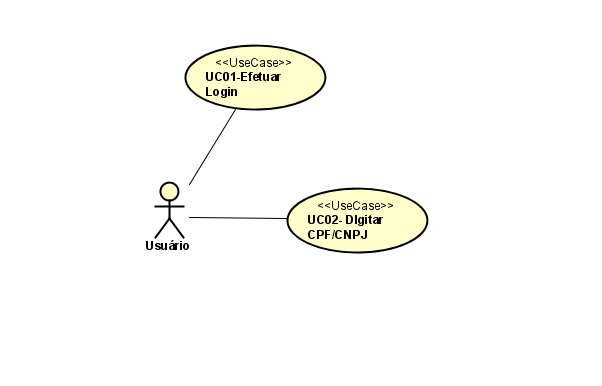
## Diagrama de Casos de Uso (UC)

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



|  |  |  |
| --- | --- | --- |
| **[UC01]** | | |
| Nome: | Efetuar 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 o login cadastrado. 2. O usuário informa a senha. 3. O usuário executa a ação de entrar, a partir do botão Efetuar Login. **[FE01] [FE02] [FE03]** 4. O sistema permite ao usuário entrar no sistema. | |
| Fluxo alternativo: | **FA01 – Realizar cadastro**  Caso o usuário não tenha cadastro e resolva se cadastrar:   1. O usuário executa o cadastro **[UC01]**. | |
| Fluxo de exceção: | **FE01 – Login incorreto**  Caso o dado informado não esteja cadastrado, o sistema exibe a mensagem “Usuário ou senha inválidos” e mantém o usuário na tela de login.  **FE02 – Senha incorreta**  Se a senha não for informada corretamente, o sistema exibe a mensagem “Usuário ou senha inválidos” e mantém o usuário na tela de login. | |
| Esboço de tela: |  | |
| Cenários: | **CEN01 – Usuário com dados corretos para entrar**  O usuário informa os dados corretos para entrar no sistema.  **CEN02 – Usuário com Login correto e senha incorreta**  O usuário informa seu login corretamente e a senha incorreta.  **CEN03 – Usuário com Login 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: |  | |

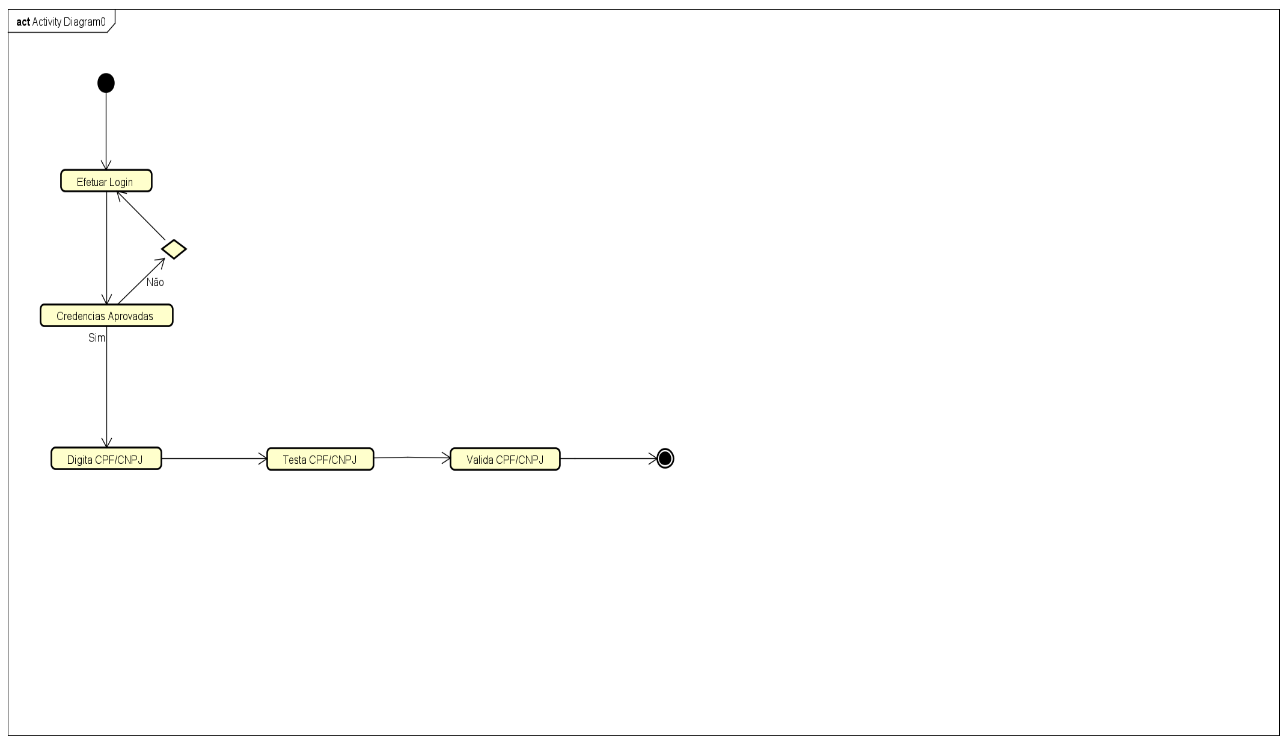
|  |  |  |  |
| --- | --- | --- | --- |
| **[UC02]** | | | |
| Nome: | | Validar CPF/CNPJ | |
| Descrição: | | Este caso de uso descreve a ação para o usuário realizar a validação de CPF/CNPJ | |
| Atores: | | Usuário | |
| Prioridade: | | [ 1 ] | |
| Entradas e pré-condições: | | | * Acesso ao sistema. |
| Saídas e pós-condições: | | | * Usuário cadastrado. |
| Fluxos de eventos | | | |
| Fluxo principal: | 1. O usuário informa CPF/CNPJ. 2. O sistema valida CPF/CNPJ | | |
| Fluxo alternativo: | **FA04 – Cancelar operação de Validação**  Caso o usuário selecione a opção de cancelar alguma operação.   1. O sistema volta para o fluxo principal. | | |
| Fluxo de exceção: |  | | |
| Esboço de tela: |  | | |
| Cenários: | **CEN01 – Usuário digita o CPF/CNPJ.**  O usuário informa CPF/CNPJ necessários para realizar o teste. | | |
| Regras de Interação: |  | | |

# ATIVIDADES

## 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.

Na figura abaixo mostramos a representação gráfica em UML de algumas atividades do sistema.



# 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.

|  |
| --- |
| CPF/CNPJ |
| -CPF/CNPJ: String |
| +CPF/CNPJ (arg: String): void |
| +validaCPF ():bollean  +validaCNPJ ():bollean |
|  |
|  |

# MATRIZ DE RASTREABILIDADE

## Referências ao modelo do documento

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

* Referências da Disciplina Engenharia de Software II UPF/2021 - https://ead.upf.br/course/view.php?id=7467
* Experiência na construção de outros documentos.