Sistema Verificador de Matrícula

Plano de Testes

Versão 0.1

[O nome do módulo deverá ser informado somente quando aplicável. Caso contrário, deverá ser excluído da capa do documento]

Histórico de Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 22/07/2018 | 0.1 | Criação dos Casos de Teste | Carlos Vilela |
|  |  |  |  |
|  |  |  |  |

Sumário

[1. Introdução 4](#_Toc350247348)

[1.1. Escopo 4](#_Toc350247349)

[2. Estágios de Teste 4](#_Toc350247350)

[3. Tipos de Testes 4](#_Toc350247351)

[4. Recursos necessários 5](#_Toc350247352)

[4.1. Recursos Humanos 5](#_Toc350247353)

[4.2. Recursos Computacionais 5](#_Toc350247354)

[5. Riscos e Restrições 6](#_Toc350247355)

[6. Produtos Gerados 6](#_Toc350247356)

[7. Caso de Teste 7](#_Toc350247357)

Plano de Testes

1. Introdução

Este documento relaciona os casos de uso a serem testados, os estágios de testes, método de qualificação, detalhamento dos tipos de testes, alvos de testes, a estratégia adotada para a execução dos testes, os recursos humanos necessários, bem como os produtos que serão gerados.

* 1. Escopo

Este documento descreve o Plano de Testes a ser usado pelo projeto SysMatricula v.0.1 para avaliar a qualidade funcional, confiabilidade e performance. O teste que será coberto por este documento será: Utilização de Seleção de Arquivos, Geração de Resultados.

1. Estágios de Teste

Definem o momento do ciclo de vida do software em que são realizados testes por pessoas diferentes daquelas que o programaram. Entretanto, considerando a divisão das tarefas de teste em quatro níveis relacionados ao escopo do software, estão previstos para o projeto **SysMatricula - Sistema Verificador de Matrícula** os seguintes estágios de teste:

* Teste de Integração: são realizados para verificar basicamente se as unidades testadas de forma individual executam corretamente quando colocadas juntas, isto é, quando integradas. Os testes são realizados pelo Analista de Testes.
* Teste de Sistema: são realizados pelo Analista de Testes, visando a execução do sistema, dentro de um ambiente operacional controlado, para validar a exatidão e perfeição na execução de suas funções.
* Teste de Aceitação ou Homologação: são os testes finais de execução do sistema, realizados pelos usuários, visando verificar se a solução atende aos objetivos do negócio e a seus requisitos, no que diz respeito à funcionalidade e usabilidade, antes da utilização no ambiente de produção.

1. Tipos de Testes

Seguem abaixo os tipos de testes a serem aplicados ao projeto **SysMatricula:**

* **Configuração:** verifica se o software está apto a rodar em diferentes versões ou configurações de ambientes (hardware e software), como, por exemplo, em diferentes browsers.
* **Funcional:** grupos de testes que avaliam se o que foi especificado foi implementado.
* **Integridade de dados:** verificar se os dados do sistema foram incluídos, alterados, excluídos e pesquisados corretamente no banco de dados. Além de validar conteúdos de campos.
* **Performance:** mede e avalia o tempo de resposta de cada transação dos requisitos sensíveis ao tempo.
* **Usabilidade:** verificam o nível de facilidade de uso do software pelos usuários.
* **Regressão:** verifica a ocorrência de novos defeitos após a resolução de defeitos.
* **Acessibilidade**: verifica se a interface do usuário fornece o acesso apropriado às funções do sistema e a navegação adequada. Além disso, estes testes garantem que os objetos dentro da interface do usuário funcionem de acordo com os padrões definidos pelo cliente.
* **Disponibilidade**: avaliam a capacidade do software em continuar operando mesmo quando algum elemento (software ou hardware) fica inoperante ou para de funcionar.

1. Recursos necessários
   1. Recursos Humanos

Esse teste é conduzido pelo analista de negócio juntamente com os usuários finais do sistema (Homologação do sistema), a fim de simular operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.

* 1. Recursos Computacionais

Recursos computacionais necessários para a execução dos testes, sejam eles de hardware ou software.

## Necessidades de Hardware

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo de Hardware | Detalhamento | Quantidade | Forma de Disponibilização | Data Limite |
| Processador | -32 bits: Processador de 1 gigahertz (GHz) ou mais rápido de 32 bits (x86)  -64 bits: Processador de 1 gigahertz (GHz) ou mais rápido de 64 bits (x64) | 1 | Corporativo | 22/07/2018 |
| Memória RAM | -32 bits: 2 GB (4 GB recomendados)  -64 bits: 4 GB (8 GB recomendados) | 1 | Corporativo | 22/07/2018 |

## Necessidades de Software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo de Software | Detalhamento | Quantidade | Forma de Disponibilização | Data Limite |
| Java | versão 6 ou posterior | 1 | Corporativo | 22/07/2018 |

## Necessidade de Pessoas

|  |  |  |  |
| --- | --- | --- | --- |
| Papel | Envolvimento Estimado | Quantidade | Período de Envolvimento no Projeto |
| Testador | 1 hora | 1 | 22/07/2018 à 22/07/2018 |

## Necessidade de Capacitação

|  |  |  |
| --- | --- | --- |
| Treinamento | Duração | Data de Realização |
|  |  |  |

1. Riscos e Restrições

Ambiente de testes não suporta testes de carga.

Equipe de testes com tempo limitado para entregar projeto.

1. Produtos Gerados

O Sistema de Verificação de Matricula, doravante denominado simplesmente SYSMATRICULA, tem por finalidade verificar uma serie de matrículas.

As matrículas armazenadas em lote (arquivo em formato texto) e separadas com quebra de linha, sendo a divisão de matrículas indicada pela mudança de linha, serão verificadas da seguinte maneira:

* As matriculas que não houverem dígito verificador terão este gerado automaticamente.
* As matrículas que possuírem o dígito verificador serão autenticadas, acusando se está correto ou não.

Dada uma matrícula de 0000 a 9999 o cálculo do dígito verificador deverá ser feito multiplicando o primeiro número a esquerda por 5, o segundo por 4, o terceiro por 3 e o quarto por 2. O resultado parcial de cada posição deve ser somado e o resultado final da soma deverá ser dividido por 16. O resto desta divisão deverá ser convertido para hexadecimal, sendo este então o dígito verificador.

Exemplo: matrícula = 9876

Cálculo:

9 \* 5 = 45

8 \* 4 = 32

7 \* 3 = 21

6 \* 2 = 12

Total = 45 + 32 + 21 + 12 = 110

Resto da divisão de 110 por 16 é igual a 14.

O número 14 em base 16 é "E". Este é o dígito verificador da matrícula.

Matrícula completa = 9876-E

1- Ler as matrículas que estão no arquivo matriculasSemDV.txt e gerar um arquivo de saída matriculasComDV.txt com as matrículas completas, conforme regra de formação descrita acima.

Exemplo:

matriculasSemDV.txt

9876

9992

matriculasComDV.txt

9876-E

9992-0

2- Ler as matrículas que estão no arquivo matriculasParaVerificar.txt e gerar um arquivo matriculasVerificadas.txt com as matrículas e um indicador de "verdadeiro" ou "falso" ao lado de cada matrícula, separado por espaço.

Exemplo:

matriculasParaVerificar.txt

9876-E

9992-2

matriculasVerificadas.txt

9876-E verdadeiro

9992-2 falso

1. CASOS DE TESTE

Para a definição dos Casos de Teste foram utilizados os módulos desenvolvidos do SysMatricula.

|  |  |
| --- | --- |
| Caso No | CT001 – Selecionar Arquivo. |
| Objetivo do Teste | Verificar se o usuário consegue Navegar nas pastas e Selecionar Arquivos. |
| Passos | 1. Executar o SysMatricula:   Acessar a pasta onde o arquivo com extensão Jar está guardado 🡪 Clicar 2(duas) vezes no arquivo 🡪 Esperar o sistema abrir.   1. No corpo do programa, Click no botão Selecionar Arquivo. 2. Navegar pelas pastas e selecionar o arquivo matriculasSemDV.txt 3. Verificar as Notificações e prosseguir. |
| Critérios de Êxito | O usuário deve receber a Notificação “Verificação do Arquivo Concluída(...)”. |

|  |  |
| --- | --- |
| Caso No | CT002 – Gerar Dígito Verificador.. |
| Objetivo do Teste | O Sistema deve ser capaz de analisar o arquivo e gerar o arquivo matriculasComDV.txt com as matrículas juntamente com seu respectivo dígito verificador, mais ainda criar o arquivo na Pasta de Resultados conforme inicialmente indicado. |
| Passos | 1. **Selecionar Pasta de Resultados.** 2. Após Selecionar o Arquivo matriculasSemDV.txt, Clicar no botão Gerar Dígito Verificador; 3. Navegar pelas pastas e selecionar a pasta onde deverá ser gerado o arquivo matriculasComDV.txt; 4. **Confirmar a Criação dos Resultados.** 5. Após selecionar a Pasta de Resultados, o usuário deverá confirmar se selecionou a pasta correta; 6. Ao confirmar a pasta, deverá aguardar até que a Notificação “Arquivo Gerado (...)” seja exibida. 7. Confirmar Notificação. |
| Critérios de Êxito | 1. **Selecionar Pasta Resultados.** 2. O usuário deve conseguir selecionar a Pasta de Resultados. 3. **Confirmar Criação dos Resultados.** 4. O Sistema deve conseguir gerar o arquivo matriculasComDV.txt no diretório escolhido como Pasta de Resultados. |

|  |  |
| --- | --- |
| Caso No | CT002 – Verificar Matrículas. |
| Objetivo do Teste | O Sistema deve ser capaz de analisar o arquivo e gerar o arquivo matriculasVerificadas.txt com as matrículas juntamente com seu respectivo dígito verificador, sinalizando se a matrícula possui o dígito verificador correto, mais ainda criar o arquivo na Pasta de Resultados conforme inicialmente indicado. |
| Passos | 1. **Selecionar Pasta de Resultados.** 2. Após Selecionar o Arquivo matriculasParaVerificar.txt, Clicar no botão Verificar Matrícula; 3. Navegar pelas pastas e selecionar a pasta onde deverá ser gerado o arquivo matriculasVerificadas.txt; 4. **Confirmar a Criação dos Resultados.** 5. Após selecionar a Pasta de Resultados, o usuário deverá confirmar se selecionou a pasta correta; 6. Ao confirmar a pasta, deverá aguardar até que a Notificação “Arquivo Gerado (...)” seja exibida. 7. Confirmar Notificação. |
| Critérios de Êxito | 1. **Selecionar Pasta Resultados.** 2. O usuário deve conseguir selecionar a Pasta de Resultados. 3. **Confirmar Criação dos Resultados.** 4. O Sistema deve conseguir gerar o arquivo matriculasVerificadas.txt no diretório escolhido como Pasta de Resultados. |