

Projeto PROJ01/2020

Documento de Requisitos e Plano de Testes

Sistema de cadastro de alunos, professores e administradores em uma escola

Versão 1.0

Responsável(is)	Lucas Sandro Rotermele Franco, Vinícius Hiago Martins
Versão	1.0
E-mails	lucas.rotermel.franco@gmail.com viniciushiagomartins00@gmail.com
Acesso	Somente leitura para clientes Escrita para membros da equipe técnica

Importante: documento com propósitos educacionais.

Controle de versionamento do documento

Data	Versão	Descrição	Autor
18/06/2021	1.0	Criação do documento de requisitos.	Lucas Sandro
19/06/2021	1.0	Criação dos casos de teste.	Vinícius H. Martins

Sumário

<u>1 Introdução.....</u>	<u>4</u>
<u>1.1 Propósito do documento.....</u>	<u>4</u>
<u>1.2 Visão geral da aplicação.....</u>	<u>4</u>
<u>1.3 Definição do escopo.....</u>	<u>4</u>
<u>2 Documento de Requisitos.....</u>	<u>4</u>
<u>2.1 Requisitos funcionais.....</u>	<u>4</u>
<u>2.2 Requisitos não funcionais.....</u>	<u>4</u>
<u>2.3 Diagrama de Casos de Uso Geral.....</u>	<u>5</u>
<u>2.4 Detalhamento de Casos de Uso.....</u>	<u>5</u>
<u>3 Plano de Testes.....</u>	<u>6</u>
<u>3.1 Necessidades de hardware.....</u>	<u>6</u>
<u>3.2 Necessidades de software.....</u>	<u>6</u>
<u>3.3 Necessidades de pessoas.....</u>	<u>6</u>
<u>3.4 Casos de teste.....</u>	<u>7</u>
<u>3.5 Execução dos casos de teste.....</u>	<u>8</u>
<u>3.6 Log de teste.....</u>	<u>8</u>
<u>4 Conclusões.....</u>	<u>8</u>

1 Introdução

Este documento tem como teor apresentar um sistema modelado previamente e com esse sistema aplicar os métodos de planos de testes apresentados na UC de Teste de Software, usando como principais métodos os testes unitários, teste de aceitação, teste estático e teste caixa branca.

1.1 Propósito do documento

O propósito desse documento serve para apresentar todas as informações do sistema criado e com essas informações também definir requisitos para que o sistema seja aprovado com base em testes reproduzidos para encontrar possíveis falhas ou pontos de melhoria.

Logo o documento também é uma apresentação de um exemplo de rotina em um ambiente de desenvolvimento, educando os desenvolvedores a buscar montar um sistema que tenha seus requisitos claros para assim haver testadores presentes que possam realizar testes no sistema para a validação do mesmo. Assim buscando a evolução constante da área de desenvolvimento de sistemas.

1.2 Visão geral da aplicação

A aplicação é apresentada como um sistema básico de cadastro de uma escola e junto com isso o cadastro de seus alunos, professores e administradores, permitindo uma organização e facilidade maior no momento do cadastro dos mesmos. Junto com o cadastro também é possível validar as informações do sistema, acessando todos os registros presentes divididos pelas funções atribuídas a cada registro(aluno, professor e administrador).

1.3 Definição do escopo

Possui as funcionalidades de registrar aluno, registrar professor, registrar administrador, imprimir alunos, imprimir professores e imprimir administradores

2 Documento de Requisitos

Esta seção apresenta os requisitos funcionais, não funcionais, o diagrama de Casos de Uso e a descrição textual de Casos Uso selecionados.

2.1 Requisitos funcionais

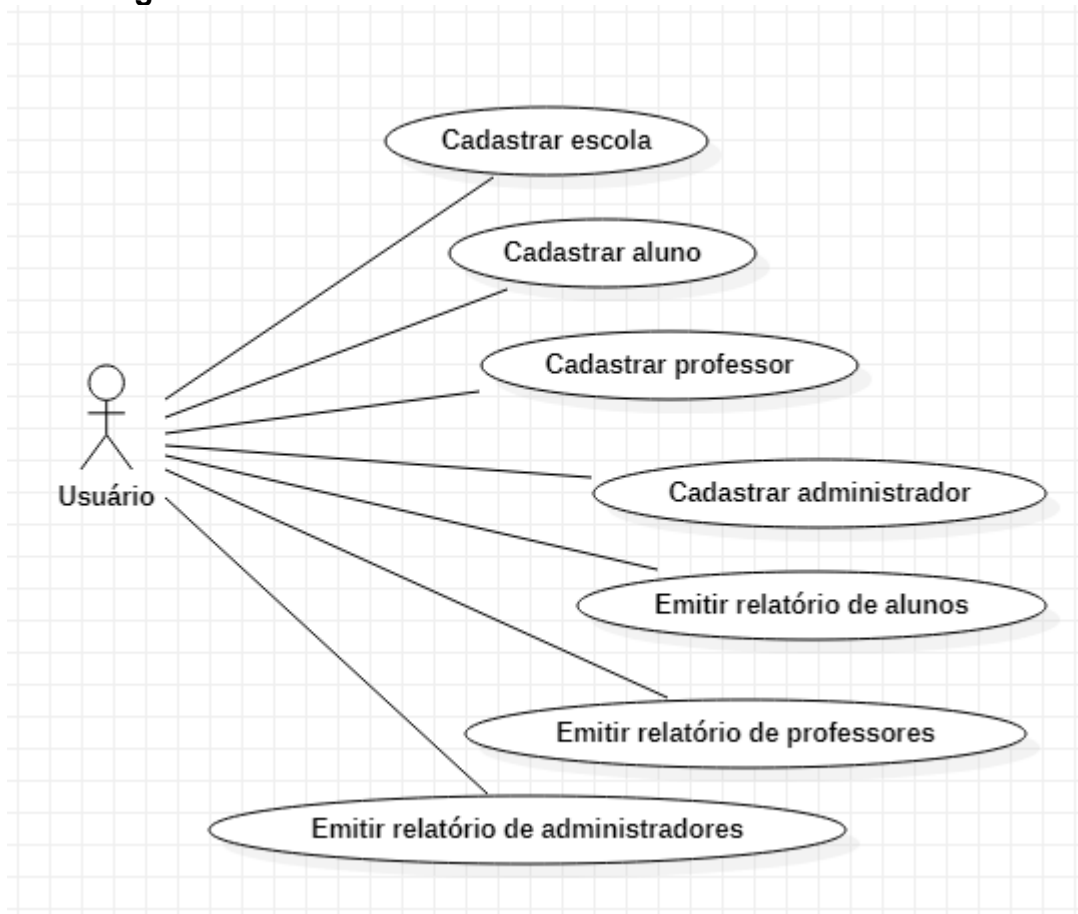
A seguir estão apresentados os requisitos funcionais do sistema

Requisito	Prioridade
RF001 – O sistema deve permitir o cadastro da escola	Alta
RF002 – O sistema deve permitir o cadastro de alunos	Alta
RF003 – O sistema deve permitir o cadastro de professores	Alta
RF004 – O sistema deve permitir o cadastro de administradores	Alta
RF005 – O sistema deve emitir um relatório com todos os alunos cadastrados	Média
RF006 – O sistema deve emitir um relatório com todos os professores cadastrados	Média
RF007 – O sistema deve emitir um relatório com todos os administradores cadastrados	Média

2.2 Requisitos não funcionais

Requisito	Prioridade
RNF001 – O sistema deve ser acessível por meio do sistema operacional Windows	Alta
RNF002 – O sistema deve ter acesso ao teclado do usuário	Alta

2.3 Diagrama de Casos de Uso Geral



2.4 Detalhamento de Casos de Uso

Caso de uso	UC001 – Cadastrar alunos
Descrição	O sistema deve permitir o cadastro de alunos. As informações necessárias para o cadastro são nome, idade, endereço, semestre, matrícula e matérias.
Requisitos	RF002
Pré-condições	Usuário do sistema deve estar logado e ter cadastrado a escola.
Pós-condições	O sistema deve apresentar mensagens de falha ou de sucesso no cadastro do aluno. O aluno recém-cadastrado deve aparecer junto com os outros alunos da escola.
Fluxo base	1 – O caso de uso inicia quando o usuário deseja cadastrar um aluno 2 – O usuário insere as informações do aluno 3 – O sistema direciona as informações para as validações necessárias 4 – Se as informações estiverem válidas o aluno é adicionado no cadastro da escola 5 – O sistema retorna para a tela inicial.
Fluxos alternativos	Se as informações estiverem incorretas o sistema proíbe de inserir o aluno no cadastro.
Fluxos de exceção	

Caso de uso	UC002 – Emitir relatório de alunos
Descrição	O sistema deve permitir emitir o relatório de alunos. As informações apresentadas são número de identificação, nome, idade, endereço, semestre, matrícula e matérias.
Requisitos	RF005
Pré-condições	Usuário do sistema deve estar logado, ter cadastrado a escola e no mínimo um aluno.
Pós-condições	O sistema deve apresentar todos os alunos cadastrados no sistema.
Fluxo base	1 – O caso de uso inicia quando o usuário deseja emitir o relatório dos alunos do sistema 2 – O sistema direciona o comando para acessar todos os alunos do cadastro 3 – O usuário recebe do sistema um relatório com todos os alunos 4 – O sistema retorna para a tela inicial.
Fluxos alternativos	Se não possui nenhum aluno no sistema, então não será apresentado o relatório.
Fluxos de exceção	

Caso de uso	UC003 – Cadastrar professores
Descrição	O sistema deve permitir o cadastro de professores. As informações necessárias para o cadastro são nome, idade, endereço, matéria que leciona e salário.
Requisitos	RF003
Pré-condições	Usuário do sistema deve estar logado e ter cadastrado a escola.
Pós-condições	O sistema deve apresentar mensagens de falha ou de sucesso no cadastro do professor. O professor recém-cadastrado deve aparecer junto com os outros professores da escola.
Fluxo base	1 – O caso de uso inicia quando o usuário deseja cadastrar um professor 2 – O usuário insere as informações do professor 3 – O sistema direciona as informações para as validações necessárias 4 – Se as informações estiverem válidas o professor é adicionado no cadastro da escola 5 – O sistema retorna para a tela inicial.
Fluxos alternativos	Se as informações estiverem incorretas o sistema proíbe de inserir o professor no cadastro.
Fluxos de exceção	

3 Plano de Testes

3.1 Necessidades de hardware

Tipo de hardware	Detalhamento	Quantidade	Forma de disponibilização	Data limite
Computador	Memória, processador, espaço em disco	1	Nuvem	19/06/2021

3.2 Necessidades de software

Tipo de software	Detalhamento	Quantidade	Forma de disponibilização	Data limite
Sistema Operacional - Windows	Versão 10	1	Nuvem	19/06/2021
Eclipse	Versão 2019-12	1	Nuvem	19/06/2021
JUnit	Versão 4.12	1	Nuvem	19/06/2021
FindBugs	Versão 3.0.1	1	Nuvem	19/06/2021

3.3 Necessidades de pessoas

Papel	Envolvimento estimado	Quantidade
Testador	2 horas	2

3.4 Casos de teste

Test Case #	TC001
Tipo do teste	Funcional
Criticidade	Baixa
Passos	Acessar a Tela 07 – Cadastrar Cliente Preencher o campo “CPF”
Dado de Teste	Fornecer um CPF conhecido (válido)
CrITÉrios de êxito (resultado esperado)	Após o preenchimento do campo CPF com um CPF válido, deve ser apresentado ao lado do campo o ícone ✓
CrITÉrio de teste utilizado	Particionamento em classes de equivalência (classe de CPFs válidos)

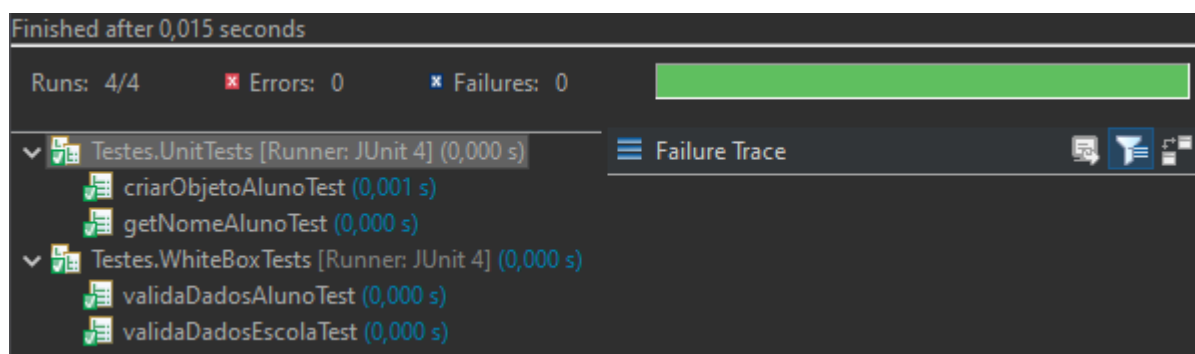
Test Case #	TC002
Tipo do teste	Funcional
Criticidade	Baixa
Passos	Acessar a Tela 07 – Cadastrar Cliente Preencher o campo “CPF”
Dado de Teste	Fornecer um CPF inválido
CrITÉrios de êxito (resultado esperado)	Após o preenchimento do campo CPF com um CPF inválido, deve ser apresentado ao lado do campo o ícone X
CrITÉrio de teste utilizado	Particionamento em classes de equivalência (classe de CPFs inválidos)

Test Case #	TC003
Criticidade	Baixa
Objetivo do teste	Verificar se a validação do CPF do cliente está funcionando adequadamente. A validação do CPF deve acontecer logo após o preenchimento do campo de mesmo nome na tela de cadastro de cliente.
Passos	Acessar a Tela 07 – Cadastrar Cliente Preencher o campo “CPF”
Dado de Teste	Não fornecer um CPF
CrITÉrios de êxito (resultado esperado)	Após o preenchimento do campo CPF com um CPF inválido, deve ser apresentado ao lado do campo o ícone X
CrITÉrio de teste utilizado	Particionamento em classes de equivalência (classe de CPFs inválidos)

3.5 Execução dos casos de teste

Os frameworks utilizados para a realização dos testes foram:

- JUnit: testes unitários e caixa branca (teste estrutural);
- FindBugs: cobertura de código;



3.6 Log de teste

Nos testes unitários e caixa branca, o resultado esperado foi obtido corretamente. Foi validado a criação do objeto “Usuário”, construtor e o envolvimento de validação de campos.

Com o uso do FindBugs também não foi encontrado nenhum erro em todo o código.

4 Conclusões

A utilização do JUnit e do Findbugs para os testes resultou em um bom aproveitamento pois tornaram o desenvolvimento do sistema em algo muito mais focado em possibilitar a execução de testes mais facilmente e também na validação constante do que foi desenvolvido em buscas de possíveis falhas antes de serem realizados os testes.

Ambos frameworks foram de fácil adaptação para o projeto e de fácil utilização, isso foi muito positivo pois permitiu focar em desenvolver mais os testes e menos em como entender a ferramenta.

Referências

Microsoft Windows 10. Microsoft Corporation, 2015. Disponível em:

<<https://www.microsoft.com/pt-br/windows/get-windows-10>>. Acesso em: 19 jun. 2021.

Eclipse. Eclipse Foundation, 2001. Disponível em: <<https://www.eclipse.org/downloads/>>.

Acesso em: 19 jun. 2021.

JUnit. BECK, Kent; GAMMA, Erich; SAFF, David; VASUDEVAN, Kris, 2006. Disponível em:

<<https://github.com/junit-team/junit4/wiki/Download-and-Install>>. Acesso em: 19 jun. 2021.

FindBugs. PUGH, Bill; HOVEMEYER, David, 2006. Disponível em:

<<http://findbugs.sourceforge.net/downloads.html>>. Acesso em: 19 jun. 2021.