**Trabalho Prático de Engenharia de Software I**

**Profa.:** Monalessa Perini Barcellos (*monalessa@inf.ufes.br*) **Período:** 2023/2

**Grupo:** <<número do grupo>>

**Alunos:** <<nomes dos membros do grupo >>

**E-mails dos alunos:** <<e-mails dos membros do grupo – listar um após o outro, separados por vírgula >>

**Documentação do Sistema <<nome do sistema>>**

1. **Introdução**

Este documento apresenta alguns dos resultados produzidos durante a fase de Especificação e Análise de Requisitos do sistema <<*nome do sistema*>>. A seção 2 apresentada o propósito do sistema. Na seção 3 é apresentada a uma visão geral do minimundo. A seção 4 apresenta os requisitos do usuário definidos no formato de *user stories* (seção 4.1), juntamente com algumas telas do protótipo construído para o sistema como forma de materializar as *user stories*, e os requisitos não funcionais (seção 4.2). A seção 5 apresenta a modelagem conceitual do sistema, incluindo os subsistemas representados na forma de um diagrama de pacotes (seção 5.1), os diagramas de casos de uso (adaptados) (seção 5.2) e os modelos de classes (seção 5.3) correspondentes a cada subsistema.

1. **Propósito do Sistema**

<<Incluir texto de apenas um parágrafo, descrevendo o propósito geral do sistema. O propósito do sistema difere do objetivo de negócio que motivou o desenvolvimento ou a aquisição do sistema. Aqui o foco deve ser no que se espera que o sistema seja capaz de realizar e que seja passível de ser verificado quando o sistema for entregue (e.g., “apoiar a realização de matrículas de alunos, lançamento de notas e emissão de históricos”), ao invés de nos benefícios que o cliente pretende alcançar a partir do uso do sistema (e.g., “aumentar a quantidade de alunos e o lucro da escola”). >>

1. **Visão Geral do Minimundo do Sistema**

<<Incluir texto livre dando uma visão geral do domínio, do problema a ser resolvido e dos processos de negócio apoiados. Também podem ser apresentadas algumas ideias principais sobre o sistema a ser desenvolvido. Os textos apresentados nas questões das atividades sobre levantamento de requisitos e sobre *user stories* podem servir como exemplo para o que deve ser incluso nesta seção. >>

1. **Requisitos de Usuário**
   1. ***User Stories***

Nesta seção, os requisitos do usuário são representados no formato de *user stories* e, para cada uma delas, são apresentados protótipos de telas para sua materialização. Para uma das *user stories* é apresentada uma descrição de “como demonstrar”, que indica os passos que devem ser seguidos (e os resultados esperados) para demonstrar a materialização da *user story* no sistema após sua implementação (ou seja, para demonstrar que a implementação da *user story* atende aos critérios estabelecidos e, assim, está “pronta”).

**US<<id**>>: Como *<<ator>>* quero *<<funcionalidade>>* para *<<finalidade>>*

**Prioridade:**<< Alta, Média ou Baixa>> **Depende de:** <<ids das *user stories* das quais esta depende>>

**Critérios de aceitação:**

CA<<id da US>>.1: <<critério de aceitação>>

CA<<id da US>>.2: <<critério de aceitação>>

CA...

*<<* Nota: para cadastros e consultas, usar o padrão de critérios indicado nos slides – fazendo os ajustes pertinentes (por exemplo, incluindo novos critérios se necessário). *>>*

**Tela(s):**

*<<imagem da tela>>*

Figura X – Tela <<nome da tela/funcionalidade>>

**Como demonstrar:** *<<*descrever como um desenvolvedor deverá demonstrar que a *user story* foi materializada adequadamente (i.e., foi implementada corretamente). Caso haja mais de uma tela para a US, o texto pode ser intercalado com as imagens das telas. Ver exemplo disponibilizado no Classroom. Importante: a descrição de como demonstrar deve ser feita para apenas uma *user story* e ela não deve ser referente a um cadastro (CRUD) nem a uma consulta. Selecionar uma *user story* que identifique uma funcionalidade significativa no sistema a ser desenvolvido. *>>*

**Classes relacionadas:** <<nomes das classes que devem ser usadas para materializar a *user story*>>

**4.2 Requisitos Não Funcionais do Sistema**

*<<*Apresentar na tabela abaixo apenas os requisitos não funcionais cujo escopo é o sistema como um todo. Requisitos não funcionais relacionados a funcionalidades específicas devem ser identificados como critérios de aceitação das *user stories* que tratam as referidas funcionalidades*.>>*

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador** | **Descrição** | **Categoria** | **Prioridade** |
| <<RNFXX>> | <<descrição >> | <<alguns dos valores possíveis: Interoperabilidade, Segurança de Acesso, Facilidade de Aprendizado, Facilidade de Operação, Atratividade, Eficiência em relação ao tempo, Disponibilidade, Manutenibilidade ou Portabilidade, dentre outros>> | <<valores possíveis: Alta, Média ou Baixa>> |

1. **Modelagem Conceitual do Sistema**

Esta seção apresenta a modelagem conceitual do sistema, incluindo os subsistemas identificados e representados na forma de um diagrama de pacotes (seção 5.1), os modelos de casos de uso de cada pacote (seção 5.2) e os modelos de classes correspondentes (seção 5.3).

* 1. **Identificação de Subsistemas**

*<<* Caso o sistema não tenha sido modularizado em subsistemas, esta seção (5.1) deve ser eliminada. Se este for caso, ajustar o texto na introdução do documento, na introdução da seção 5 e a numeração das próximas seções. Obs.: considerando o contexto do trabalho, é provável que todos os sistemas sejam modularizados em subsistemas.*>>*

A Figura X mostra os subsistemas identificados para o sistema, os quais são descritos em seguida, na Tabela 1.

*<<diagrama de pacotes, contendo os subsistemas identificados e suas dependências>>*

Figura X – Diagrama de pacotes representando os subsistemas.

Tabela 1 – Subsistemas

|  |  |
| --- | --- |
| **Subsistema** | **Descrição** |
| <<nome do subsistema>> | <<descrição do subsistema>> |
| <<nome do subsistema>> | <<descrição do subsistema>> |

**5.2 Modelo de Casos de Uso**

O modelo de casos de uso visa representar as funcionalidades que o sistema deve prover. Aqui, ele é usado de forma adaptada para prover uma abstração visual das funcionalidades – i.e., *user stories* – do sistema. A seguir, são apresentados os diagramas de casos de uso de cada um dos subsistemas identificados.

**5.2.1 Subsistema <<nome do subsistema>>**

A Figura X apresenta o diagrama de casos de uso do subsistema <<nome do subsistema>>.

*<<diagrama de casos de uso – indicar no diagrama para cada caso de uso a(s) user history(ies) nele contemplada(s)>>*

Figura X – Diagrama de casos de uso do subsistema <<nome do subsistema>>

**5.2.2 Subsistema <<nome do subsistema>>**

A Figura X apresenta o diagrama de casos de uso do subsistema <<nome do subsistema>>.

*<<diagrama de casos de uso – indicar no diagrama para cada caso de uso a(s) user history(ies) nele contemplada(s)>>*

Figura X – Diagrama de casos de uso do subsistema <<nome do subsistema>>

**5.3 Modelo Conceitual Estrutural**

O modelo conceitual estrutural visa capturar e descrever as informações (classes, associações e atributos) que o sistema deve representar para prover as funcionalidades descritas nas *user stories*. A seguir, são apresentados os diagramas de classes de cada um dos subsistemas identificados.

**5.3.1 Subsistema <<nome do subsistema>>**

A Figura X apresenta o diagrama de classes do subsistema <<nome do subsistema>>.

*<<diagrama de classes do subsistema>>*

Figura X – Diagrama de classes do subsistema <<nome do subsistema>>

As seguintes restrições de integridade devem ser observadas:

*<<*Listar as restrições de integridade do modelo. Caso o grupo prefira, elas podem ser representadas em notas no modelo ao invés de nos tópicos abaixo.*>>*

* RI-1: << texto da RI-1>>
* RI-n: <<texto da RI-n>>

Os seguintes tipos específicos de domínio devem ser considerados:

<<Listar os tipos específicos de domínio identificados no diagrama de classes.>>

* <<nome do tipo específico de domínio>>: <<descrição – indicando se é um tipo de dados ou um tipo enumerado. No primeiro caso, descrever o tipo de dados, no segundo, indicar os valores que o atributo do referido tipo pode receber>>
* ...

**5.3.2 Subsistema <<nome do subsistema>>**

A Figura X apresenta o diagrama de classes do subsistema <<nome do subsistema>>.

*<<diagrama de classes do subsistema>>*

Figura X – Diagrama de classes do subsistema <<nome do subsistema>>

As seguintes restrições de integridade devem ser observadas:

*<<*Listar as restrições de integridade do modelo. Caso o grupo prefira, elas podem ser representadas em notas no modelo ao invés de nos tópicos abaixo.*>>*

* RI-1: << texto da RI-1>>
* RI-n: <<texto da RI-n>>
* ...

Os seguintes tipos específicos de domínio devem ser considerados:

<<Listar os tipos específicos de domínio identificados no diagrama de classes.>>

* <<nome do tipo específico de domínio>>: <<descrição – indicando se é um tipo de dados ou um tipo enumerado. No primeiro caso, descrever o tipo de dados, no segundo, indicar os valores que o atributo do referido tipo pode receber>>
* ...