Texto

Descrição gerada automaticamente

**Documentação de um**

**Produto de Software**

**TCC – Trabalho de Conclusão de Curso**

**Nome dos Alunos**

**2024**

ÍNDICE DETALHADO

3. Requisitos do Sistema de Software 3

3.1. Requisitos Funcionais 4

3.2. Modelagem Funcional 4

3.3. Requisitos Não-Funcionais 6

3.4. Protótipo – FAZER POSTERIORMENTE 8

4. Análise 9

4.1. Diagrama de Classes de Análise (Visão de Negócio) 9

5. Projeto 10

5.1. Arquitetura do Sistema 10

5.2. Diagrama de Classes de Projeto por Caso de Uso 10

5.3. Diagrama de atividades 11

5.4. Diagrama de estados 12

6. Testes 13

6.1. Plano de Testes 13

6.2. Roteiro de Testes 13

1. Requisitos do Sistema de Software

Este capítulo tem como objetivo especificar os requisitos funcionais, não funcionais e a regras de negócio, bem como apresentar o protótipo de telas e o cronograma de atividades do desenvolvimento do software.

O texto a seguir tem como objetivo relembrar conceitos e padrões de especificação dos requisitos de software.

**Identificação dos requisitos**

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

* **Requisitos funcionais** devem ser identificados por: [RFxxx] – nome e descrição e serem descritos na seção requisitos funcionais.
* **Regras de negócio** devem ser identificadas por: [RNxxx] – nome e descrição e serem descritas na seção requisitos funcionais, subitem regras de negócio.
* **Requisitos não funcionais** devem ser identificados por: [RNFxxx] – nome e descrição e serem descritos na seção requisitos não funcionais.

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001], [RN001] ou [RNF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos ou regras.

Por convenção, a referência aos casos de uso é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do caso de uso, de acordo com a especificação a seguir:

Casos de Uso devem ser identificados por: CSUXXX – nome a ser descrito na função modelagem funcional.

Os casos de uso devem ser identificador com um identificador único. A numeração inicia com o identificador CSU001 e prossegue sendo incrementada à medida que forem surgindo novos casos de uso.

**Prioridades dos requisitos**

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações “essencial”, “importante” e “desejável”.

* **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
* **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
* **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

1. Requisitos Funcionais

Neste item devem ser descritos os requisitos funcionais que especificam ações que um sistema deve ser capaz de executar, ou seja, os objetivos do sistema, incluindo prioridade e regras de negócio. A seguir são apresentados exemplos.

**[RF001] – Consultar catálogo de produtos**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: Este requisito permite que o vendedor consulte o produto solicitado pelo cliente obtendo informações referentes ao fabricante, preço e quantidade disponíveis.

**[RF002] – Alterar ficha de clientes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ◼ | Essencial | 🞎 | Importante | 🞎 | Desejável |

**Descrição**: Este requisito permite que o vendedor altere os dados de um cliente. Os dados à serem alterados são: endereço completo e telefones para contato.

## Regras de Negócio

**[RN001] – Alteração dos dados do cliente**

**Descrição:** Um vendedor poderá alterar os seguintes dados dos clientes: endereço completo e telefones para contato. As demais informações devem ser disponibilizadas no formato “somente leitura”.

**[RN002] – Produtos em falta**

**Descrição:** Ao consultar o catálogo de produtos o sistema deverá apresentar os produtos em falta (sem estoque) com tarja vermelha, indicando a inexistência de estoque para estes produtos.

1. Modelagem Funcional

Neste item devem ser descritos os requisitos a serem atendidos funcionalmente pelo sistema de uma forma simples, possibilitando a compreensão do comportamento do sistema pela perspectiva do usuário. Devem ser descritos os atores e o diagrama de caso de uso. A seguir um exemplo de especificação de atores, do diagrama de caso de uso e da especificação de caso de uso.

**DIAGRAMA DE CASO DE USO**

A seguir é apresentada a notação básica de um diagrama de caso de uso.



Notação básica do diagrama de caso de uso.

**ATORES**

A seguir é apresentado um exemplo da especificação de atores.

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Cliente | Usuário do sistema responsável por pesquisar livros disponíveis para venda. |

**ESPECIFICAÇÃO DO CASO DE USO**

A seguir é apresentado um exemplo da especificação de casos de uso.

|  |  |
| --- | --- |
| **CSU001 – Pesquisar Livro** | |
| Sumário: | Este caso de uso tem como objetivo pesquisar livros vendidos pela Web. |
| Pré-condição:O livro deve estar cadastrado. | |
| **Fluxo Principal** Este caso de uso se inicia quando o ator cliente escolhe a opção de “Pesquisar Livros”.O ator cliente informa nome do livro. [RN1]O nome do livro é enviado para o Sistema de Estoque que retorna os dados do livro: Código, Nome do livro, Preço e Disponibilidade/Indisponibilidade. [FE1]O sistema exibe os dados do livro que são: Código, Nome do Livro, Preço e Disponibilidade/Indisponibilidade. | |
| Fluxos AlternativosNão se aplica. | |
| Fluxos de Exceção[FE1] Fluxo Exceção 1: Livro não encontradoEste fluxo alternativo ocorre quando o livro não é encontrado no Sistema de Estoque.O sistema exibe uma mensagem que o livro pesquisado não foi encontrado.O fluxo retorna ao passo 1 do fluxo básico. | |
| Pós-condições:Não se aplica. | |
| Regras de Negócio:RN1 | |

1. Requisitos Não-Funcionais

**[RNF001] - Portabilidade**

**Descrição:** O sistema deve ser capaz de rodar tanto em computadores quanto em celulares, mantendo uma interface amigável em ambos os cenários.

**[RNF002] - Manutenção**

**Descrição:** O sistema deve ser fácil de manter e atualizar.

**[RNF003] - Disponibilidade**

**Descrição:** O sistema deve estar sempre disponível para que qualquer um possa utilizá-lo.

**[RNF004] - Conformidade**

**Descrição:** O sistema deve seguir a LGPD e demais leis de regulamentação digital.

**[RNF005] - Segurança**

**Descrição:** O sistema deve estar protegido de acessos de usuários não permitidos.

**[RNF006] - Usabilidade**

**Descrição:** A interface do sistema deve ser simples e amigável para o usuário, permitindo que qualquer ação dentro dele seja feita em menos de 5 minutos.

**[RNF008] - Arquitetura**

**Descrição:** A arquitetura utilizada seguira o modelo MVC junto com o padrão DAO.

**[RNF009] - Tecnologia**

**Descrição:** A tecnologia utilizada para o back-end será Python com o micro framework flask; para o front-end será utilizado a biblioteca REACT; e para o banco de banco de dados, MySQL.

1. Protótipo – FAZER POSTERIORMENTE

Neste item deve ser apresentado o protótipo do sistema que consiste na interface preliminar contendo um subconjunto de funcionalidades e telas. O protótipo deve ser incrementalmente evoluído até a concordância completa dos requisitos previstos para o sistema, de comum acordo com o usuário. O protótipo é um recurso que deve ser adotado como estratégia para levantamento, detalhamento, validação de requisitos e modelagem de interface com o usuário (usabilidade).

Sugere-se que cada tela possua uma descrição detalhada do seu funcionamento. Alguns itens importantes na descrição são:

* Objetivo da tela;
* De onde é chamada e que outras telas podem chamar;
* Regras:
* Domínio (tamanho de campo, tipo de dados que aceita valor default);
* Tipo de usuários que podem acessar;
* Lógica de negócio (campos obrigatórios, validade entre datas, preenchimento anterior de um campo para efetuar uma operação, etc).

A descrição detalhada das telas deve registrar informações que possam ser consultadas na implementação do sistema, facilitando, agilizando e minimizando erros de implementação e na execução de testes.

**O protótipo de telas deve estar associado ao caso de uso correspondente, podendo ser apresentado conjuntamente com o caso de uso correspondente. A seguir é apresentado um exemplo de protótipo de tela.**

1. Análise

Este capítulo tem como objetivo analisar, detalhar e propor uma solução geral do sistema, sob o ponto de vista de negócio, de acordo com os requisitos levantados e validados no capítulo 3.

* 1. Diagrama de Classes de Análise (Visão de Negócio)

Neste item deve ser apresentado o modelo do domínio, visão de negócio, que representa um primeiro modelo conceitual do diagrama de classes. Posteriormente, esse diagrama deve ser validado, refinado e complementado para compor o diagrama de classes de projeto.

O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e métodos de cada classe, e os relacionamentos entre elas.

A seguir é apresentada a notação básica de um diagrama de classes



Notação básica do diagrama de classes.

1. Projeto

Este capítulo tem como objetivo refinar a proposta de solução geral do sistema, apresentando a solução técnica, incluindo a visão de projeto e implementação, a arquitetura e a tecnologia utilizada.

* 1. Arquitetura do Sistema

Neste item deve ser apresentada a arquitetura de infraestrutura do sistema, demonstrando o tipo de arquitetura que será utilizada (por exemplo, cliente/servidor de n-camadas, MVC, ...), a configuração de hardware, de rede, de software, padrões de projeto, componentes específicos (dll, jar, ...) e componentes externos a serem utilizados, bem como o dimensionamento mínimo de conexões.

Para a representação da arquitetura de infraestrutura pode-se utilizar uma figura ilustrativa ou o diagrama de implantação. A seguir é apresentada a notação básica de um diagrama de implantação.

Notação básica do diagrama de implantação.

* 1. Diagrama de Classes de Projeto por Caso de Uso

Este item tem como objetivo apresentar a realização de caso de uso para cada caso de uso.

A realização de caso de uso deve conter:

* + diagrama de classes para cada caso de uso. Notação básica do diagrama de classes já foi apresentada no item 4.1.
  + diagrama de sequência (para cada cenário do fluxo principal/básico e fluxos alternativos). Neste item devem ser apresentados os diagramas de sequência essenciais ao sistema, identificados através dos casos de uso.

Um diagrama de sequência representa interações de objetos organizadas em uma sequência temporal, apresentando os objetos que participam da interação e a sequência das mensagens trocadas. O diagrama de sequência deve validar o diagrama de classes e vice-versa.

**Obs: Para os casos de uso que representam <<crud>> básico não há necessidade de realização do caso de uso.**

* 1. Diagrama de atividades

O diagrama de atividades representa o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema, geralmente utilizado para os métodos que contém regras de negócio. A seguir é apresentada a notação básica de um diagrama de atividades. **Esse diagrama somente deverá ser elaborado se houver necessidade e agregar valor ao projeto.**



Notação básica do diagrama de atividades.

* 1. Diagrama de estados

O diagrama de estados especifica as sequências de estados pelas quais o objeto pode passar durante seu ciclo de vida em resposta a eventos. A seguir é apresentada a notação básica de um diagrama de estados. **Esse diagrama somente deverá ser elaborado se houver necessidade e agregar valor ao projeto.**



Notação básica do diagrama de estados

6. Testes

Este capítulo tem como objetivo identificar erros no sistema, validar as funções do sistema, verificar se os requisitos foram implementados de forma adequada. Sugere-se a criação de um plano de testes e um roteiro de testes baseando nos casos de uso. Também pode-se utilizar a técnica de TDD (*Test Driven Development*), neste caso os testes também devem ser registrados.

6.1. Plano de Testes

Neste item deve ser criado o plano de testes do sistema, permitindo a validação do sistema por parte do desenvolvedor, através da verificação dos requisitos do sistema desenvolvido.

6.2. Roteiro de Testes

Neste item devem ser registrados os testes realizados no sistema tendo como base o Plano de Testes do Sistema. O roteiro de testes deve ser elaborado com base nos casos de uso.