**Documento de Visão de Projeto**



**NewStation**

**Histórico de Versões**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor(es)** | **Revisor** |
| 15/09/20 | 1.0 | Modelagem e desenvolvimento | Breno Gabriel e Marcelo de Almeida | Rodrigo |
| 05/11/20 | 1.1 | Readequações de Diagramas | Breno Gabriel e Marcelo de Almeida | Rodrigo |

|  |  |
| --- | --- |
| **Cliente** | Rodrigo Rocha Silva |
| **Documento** | Documento de Visão de Projeto |
| **Data** | 18 de maio de 2020 |
| **Autores** | **Breno Gabriel Rodrigues da Silva**  [brenogabriel19@gmail.com](mailto:brenogabriel19@gmail.com)  **Marcelo de Almeida Filho**  marcelo.a\_f@outlook.com |

**Página de Assinaturas**

|  |  |  |
| --- | --- | --- |
| Revisado e Aprovado por: |  |  |
|  |  | 18/05/21 |

**Índice**

Camada de Apresentação 7

Camada de Persistência 11

Diagrama de caso de uso geral do sistema 5

Escopo 4

Objetivo 4

Pacote Model 10

Qualidade 13

Realização dos Casos de Uso Significativos 11

Representação Arquitetural 4

Tamanho e Performance 13

Visão de Dados 13

Visão de Implementação 13

Objetivo

Este documento trata principalmente da documentação das necessidades de negócios, da justificativa do projeto, do entendimento atual das necessidades do cliente e descreve resumidamente o novo produto, serviço ou resultado que deve satisfazer esses requisitos.

Tem o objetivo de alinhar as expectativas dos interessados para formalizar o início do projeto. Apresentar uma visão arquitetural dos sistemas da linha **E-commerce.**

* 1. Escopo

O escopo deste documento trata do desenvolvimento de um sistema que atenda todas as necessidades de um E-commerce.

O escopo deste documento é documentar as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes. Além disso, mostra sua divisão em classes e utilitários de classe.

* 1. Referências

Listar documentos relacionados a este documento de visão. Podem ser tanto documentos que foram utilizados para confeccionar este documento ou fazem parte deste documento, como documentos que estendem este documento.

Para a construção deste documento foram utilizadas as seguintes referências:

* Reuniões formais entre o grupo e Rodrigo Rocha.
* Documentos elaborados pelo grupo

Este documento influencia os seguintes documentos:

* Documento de Requisitos

Necessidades de Negócio

Necessidade 1: Um sistema informatizado para compra de Cartas, no qual terá cadastro, atualização de cadastro, exclusão e listagem de clientes, produtos e vendas.

Necessidade 2: Um sistema informatizado que a partir dos dados armazenados referentes as operações de venda de produtos, deverá ser criado um gráfico que apresente quais são os produtos mais vendidos da loja.

Objetivo do Projeto

Objetivo 1: Desenvolver uma plataforma para soluções web capaz de:

* armazenar informações em uma base de dados
* utilizar o protocolo HTTP
* ser executado em qualquer navegador

Objetivo 2: Desenvolver um sistema para venda que permita:

* Controlar as Cartas presentes no acervo da Loja e entrada de novas Cartas;
* Gerenciamento das operações realizadas na Loja: compra de Carta, Cadastro de Carta e Cliente.

Declaração Preliminar de Escopo

Esta seção descreve, em alto nível, o escopo do projeto. Os requisitos serão mais bem detalhados nos documentos de Requisitos.

* 1. Descrição

Descrição narrativa do produto do projeto.

* 1. Produtos a serem entregues

Os seguintes itens são considerados produtos do projeto, na sua etapa 1.

* Sistema do módulo da biblioteca, etapa 1, implementado de acordo com a especificação feita na fase de análise. (código objeto e código fonte).
* Documentos de especificação do sistema, concebido na fase de elaboração.
* Hospedagem do sistema em ambiente 24 x 7.
  1. Requisitos
     1. Requisitos Funcionais

Grupo: Cadastro de Cartas

• RF0011 Cadastrar Carta: O sistema deve manter um cadastro único para Cartas.

• RF0012 Inativar cadastro de Carta: O sistema deve possibilitar que Cartas sejam inativados.

• RF0013 Inativar Carta de forma automática: O sistema deve inativar Cartas sem estoque e que não possuem venda com valor inferior a parâmetro predefinido no sistema.

• RF0014 Alterar cadastro de Carta: O sistema deve possibilitar a alteração de dados cadastrais para as Cartas.

• RF0015 Consulta de Cartas: O sistema deve possibilitar que uma Carta seja consultada com base em um filtro definido pelo usuário. Todos os campos utilizados para identificação da Carta podem ser utilizados como filtro, tanto de forma combinada como de forma isolada.

• RF0016 Ativar cadastro de Cartas: Deve ser possível ativar o cadastro de uma Carta.

Grupo: Cadastro de Clientes

• RF0021 Cadastrar cliente: O sistema deve possibilitar o cadastro de clientes.

• RF0022 Alterar cliente: O sistema deve possibilitar a alteração de dados cadastrais de clientes.

• RF0023 Inativar cadastro de cliente: O sistema deve possibilitar que clientes sejam inativados.

• RF0024 Consulta de clientes: O sistema deve possibilitar que um cliente seja consultado com base em um filtro definido pelo usuário. Todos os campos utilizados para identificação do cliente podem ser utilizados como filtro, tanto de forma combinada como de forma isolada.

• RF0025 Consulta de transações: O sistema deve disponibilizar no cadastro de clientes a consulta de todas as transações já realizadas por ele.

• RF0026 Cadastro de endereços de entrega: Deve ser possível associar diversos endereços de entrega ao cadastro de um cliente. Cada cadastro de endereço deve ser identificado com um nome composto de uma frase curta.

• RF0027 Cadastro de cartões de crédito: Deve ser possível associar diversos cartões de crédito ao cadastro de um cliente. Deve haver um cartão de crédito configurado como preferencial.

• RF0028 Alteração apenas de senha: O sistema deve possibilitar que a senha do usuário seja alterada sem que seja necessária a alteração de todos os dados cadastrais.

Grupo: Gerenciar Vendas Eletrônicas

• RF0031 Gerenciar carrinho de compra: O sistema deve permitir que produtos sejam colocados em um repositório temporário para futura compra (carrinho de compra). Deve ser possível adicionar, alterar e excluir itens de compra no carrinho. Também deve ser possível visualizar os itens no carrinho.

• RF0032 Definir quantidade de itens no carrinho: Deve ser possível editar a quantidade de cada item ao adicionar um produto no carrinho. Também deve ser possível editar a quantidade de itens de um carrinho na visualização dos itens já adicionados.

• RF0033 Realizar compra: Deve ser possível a partir de um carrinho de compra realizar uma compra.

• RF0034 Calcular frete: O sistema deve calcular o frete da compra com base nos itens selecionados e o endereço apontado pelo cliente.

• RF0035 Selecionar endereço de entrega: O cliente pode selecionar qualquer endereço de entrega previamente cadastrado em seu perfil ou um novo endereço de entrega pode ser cadastrado. Caso um novo endereço de entrega seja inserido, deve-se dar a possibilidade que este seja incorporado ao perfil do cliente.

• RF0036 Selecionar forma de pagamento: O cliente pode selecionar qualquer cartão de crédito previamente cadastrado em seu perfil ou um novo cartão de crédito pode ser cadastrado. Caso um novo cartão de crédito seja cadastrado, deve-se dar a possibilidade que este seja incorporado ao perfil do cliente. O cliente também poderá utilizar um cupom de troca ou um cupom promocional válido. Deve-se possibilitar que o pagamento seja feito utilizando tanto cupons de troca, promocionais e cartão de crédito.

• RF0037 Finalizar Compra: Uma compra deve ser finalizada após a seleção da forma de pagamento e endereço de entrega. Após a finalização o status da compra deve ser EM PROCESSAMENTO.

• RF0038 Despachar produtos para entrega: O sistema deve possibilitar que um usuário com perfil de administrador selecione vendas já aprovadas para serem entregues. Assim o status deve ficar EM TRÂNSITO.

• RF0039 Produtos entregues: O sistema deve possibilitar que um usuário com perfil de administrador confirme entrega de uma compra. Assim o status deve ficar ENTREGUE.

• RF0040 Solicitar troca: O sistema deve possibilitar que um item de uma compra seja trocado por um cliente através da visualização de pedidos dele.

• RF0041 Autorizar trocas: O sistema deverá possibilitar que o administrador autorize pedidos ou compra com status EM TROCA. Assim o pedido passa ficar com status TROCA AUTORIZADA.

• RF0042 Visualização de trocas: O sistema deverá possibilitar que o administrador visualize todos os pedidos de troca ou compra com status EM TROCA.

• RF0043 Confirmar recebimento de itens para troca: O sistema deverá possibilitar que o administrador confirme o recebimento de pedidos de troca ou compra com status EM TROCA. Nesta confirmação o administrador deverá informar se os itens trocados deverão retornar ao estoque. Em caso positivo deve-se dar entrada no estoque dos respectivos itens.

• RF0044 Gerar cupom de troca após recebimento de itens: O sistema deverá gerar um cupom de troca quando o administrador informar que os itens a serem trocados chegaram. Este cupom deverá ser disponibilizado para o cliente para ser utilizado em futuras compras.

Grupo: Controle de estoque

• RF0051 Realizar entrada em estoque: O sistema deve permitir que seja possível realizar entrada de itens de Cartas em estoque. No registro de cada item, deve ser indicado a Carta já previamente cadastrado e a quantidade de itens do Carta.

• RF0052 Calcular valor de venda: O sistema deve calcular o valor de venda com base no valor de custo e o grupo de precificação. Sendo que o valor de venda será o valor de compra mais o percentual definido no grupo de precificação relacionado a Carta.

• RF0053 Dar baixa em estoque: Para cada venda realizada deve-se dar baixa no estoque do total de itens vendidos.

• RF0054 Realizar reentrada em estoque: O sistema deve realizar a reentrada de um item em estoque a partir da troca de um produto.

* + 1. Requisitos Não Funcionais

Grupo: Geral

Utilizar linguagem Java

Utilizar o banco de dados MySQL

A arquitetura da solução obedecer ao design pattern MVC.

O sistema deve rodar nos seguintes browsers:

* IE
* Google Chrome
* Firefox
* Cronograma de Marcos Sumariado

**Grupo: Cadastro de Cartas**

• RNF0021 Código de Carta: Todo Carta cadastrado deve receber um código único no sistema.

• RNF0013 Cadastro de domínios: Deve haver um script de implantação do sistema que insere todos os registros de tabelas de domínio necessárias por exemplo: grupo de precificação, autor, editora, fornecedor etc.

**Grupo: Cadastro de Clientes**

• RNF0031 Senha forte: A senha cadastrada pelo usuário deve ser composta de pelo menos 8 caracteres, ter letras maiúsculas e minúsculas além de conter caracteres especiais.

• RNF0032 Confirmação de senha: O usuário obrigatoriamente deve digitar duas vezes a mesma senha no momento do registro dela.

• RNF0033 Senha criptografada: A senha deve ser criptografada

• RF0034 Alteração apenas de endereços: O sistema deve possibilitar que endereços de entrega ou cobrança possam ser alterados ou adicionados de forma simples sem a necessidade da edição dos demais dados cadastrais.

• RNF0035 Código de cliente Todo cliente cadastrado deve receber um código único no sistema.

**Grupo: Gerenciar Vendas Eletrônicas**

• RNF0042 Apresentar itens retirados do carrinho Deve ser apresentado na listagem de itens do carrinho os produtos removidos por atingirem o prazo determinado para finalização da compra (apresentar o tempo conforme parâmetro do sistema). Assim a opção comprar deve ser desabilitada e o itens deverão ser adicionados novamente no carrinho.

* + 1. Regras de Negócio

Grupo: Cadastro de Cartas

* RN0011 Dados obrigatórios para o cadastro de uma Carta: Para todo Carta cadastrado é obrigatório o cadastro dos seguintes dados: nome, raridade, preço, descrição.
* RN0012 Associação com categorias: Um Carta pode estar associado com mais de uma categoria.
* RN0013 Definindo valor de venda: Todo Carta após cadastrado deverá ser associado a um grupo de precificação onde o valor deverá ter como base a margem de lucro parametrizado para o grupo definido no cadastro do Carta.
* RN0014 Validar margem de lucro: Um Carta somente pode ter seu valor alterado se estiver dentro da margem de lucro definida pelo critério de grupo de precificação, dado pela sua raridade.

Grupo: Cadastro de Clientes

* RN0021 - Cadastro de endereço de cobrança: Para todo cliente cadastrado é obrigatório o registro de ao menos um endereço de cobrança.
* RN0022 - Cadastro de endereço de entrega: Para todo cliente cadastrado é obrigatório o registro de ao menos um endereço de entrega.
* RN0023 - Composição do registro de endereços: Todo cadastro de endereços associados a clientes deve ser composto dos seguintes dados: Logradouro, Número, Bairro, CEP, Cidade e Estado. Todos os campos anteriores são de preenchimento obrigatório. Opcionalmente pode ser preenchido um campo de Complemento.
* RN0024 - Composição do registro de cartões de crédito: Todo cartão de crédito associado a um cliente deverá ser composto pelos seguintes campos: Nº do Cartão, Nome impresso no Cartão, Bandeira do Cartão e Código de Segurança.
* RN0025 - Bandeiras permitidas para registro de cartões de crédito: Todo cartão de crédito associado a um cliente deverá ser de alguma bandeira registrada no sistema.
* RN0026 - Dados obrigatórios para o cadastro de um cliente: Para todo cliente cadastrado é obrigatório o cadastro dos seguintes dados: Gênero, Nome, Data de Nascimento, CPF, Telefone (deve ser composto pelo tipo, DDD e número), e-mail, senha, endereço residencial.
* RN0027 – Tipo de cliente O cliente deve receber um tipo com base no seu perfil de compra.

Grupo: Gerenciar Vendas Eletrônicas

* RN0031 - Validar estoque para adição de itens no carrinho: Não deve ser permitido adicionar um item no carrinho de compra que não esteja disponível em estoque. Também deve ser validado a quantidade do item adicionado ao carrinho para que não seja adicionado mais itens do que o disponível em estoque.
* RN0032 - Validar estoque para compra: Ao solicitar a compra de itens que estejam em um carrinho deve-se garantir que tais itens ainda permanecem disponíveis em estoque.
* RN0033 - Uso de cupom promocional para pagamento: Apenas um cupom promocional pode ser utilizado por compra.
* RN0034 - Uso de diversos cartões de crédito: Uma compra pode ser paga utilizando mais de um cartão de crédito, porém o valor mínimo para ser pago com cada cartão deve ser R$ 10,00.
* RN0035 - Uso de cupons junto a cartão de crédito: Ao realizar pagamento utilizando cupons e cartões em conjunto, deve-se sempre considerar o valor máximo dos cupons. Somente neste caso é permitido que seja realizado um pagamento de um valor menor que R$ 10,00 no cartão. Exemplo: Uma compra de R$ 35,00 o cliente pode pagar R$ 30,00 utilizando cupons de troca ou cupons promocionais e pagar R$ 5,00 com cartão de crédito.
* RN0036 - Gerar cupom de troca: Um cupom de troca deve ser gerado quando uma compra for paga com outros cupons em que o valor supere o valor da compra. Obs: O sistema não deve possibilitar o uso de cupons que supere a compra desnecessariamente, ex: a venda tem valor total de R$ 50,00 e o cliente possui três cupons, um com valor de R$ 20,00, outro com valor de R$ 40,00 e um terceiro com valor de R$ 35,00 o sistema não deve possibilitar o uso dos três cupons nesta compra, deve ser aceito apenas dois cupons e consequentemente gerar um cupom com a diferença de R$ 5,00, ou R$ 10,00 ou R$ 25,00.
* RN0037 - Validar Forma de Pagamento para finalização de compra: Após a finalização da compra a forma de pagamento deve ser validada. Para tal deve-se validar a validade e veracidade dos cupons de troca e promocionais que porventura foram utilizados. Também deve ser validado o aceite da compra pela respectiva operadora de cartão de crédito.
* RN0038 - Alterar status da compra conforme processo de aprovação de forma de pagamento: Caso as formas de pagamento tenham sido validadas com sucesso, a compra deve passar ter o status APROVADA. Caso contrário deve passar a ter o status REPROVADA.
* RN0039 - Alterar status da compra para transporte: Toda compra selecionada para ser entregue por um administrador deve ter seu status alterado para EM TRANSPORTE.
* RN0040 - Alterar status da compra após entrega: Toda compra selecionada como entregue por um administrador deve ter seu status alterado para ENTREGUE.
* RN0041 - Gerar pedido de troca: Todo item selecionado para troca deve gerar um pedido de troca. Este pedido deverá terá o status EM TROCA. Caso o cliente solicite a troca de toda a compra o status do pedido deverá ser EM TROCA.
* RN0042 - Alterar status do pedido após recebimento de troca: Ao confirmar que os itens de um pedido de troca ou uma compra com status EM TROCA foi recebido o status do pedido ou compra deverá ser TROCADO.
* RN0043 - Validação para solicitar troca: Somente itens de pedidos com status ENTREGUE poderão receber solicitação de troca.
* RN0044 - Bloqueio de produtos: Ao adicionar o item no carrinho, este deverá ser temporiamente bloqueado para que novas compras não sejam solicitadas. Tal bloqueio só deve ser retirado no caso da compra que gerou tal status não ser efetivada ou aprovada em um prazo parametrizado, o prazo deve levar em consideração o momento do bloqueio. Obs.: O prazo parametrizado deve ser relativo ao último item incluído no carrinho.
* RNF0045 - Retirar item do carrinho: Toda vez que um item for desbloqueado todos os itens do mesmo produto deverão ser retirados do carrinho de compra que gerou o prazo de bloqueio.

Grupo: Controle de estoque

* RN0051 - Validar dados de estoque: Para cada entrada em estoque, deve ser obrigatoriamente informado o produto e a quantidade.
* RN0052 - Quantidade de itens: Não deve ser permitido que seja realizado a entrada de itens de Cartas com quantidade igual a zero.
* RN0053 - Data de entrada: Não deve ser permitido que itens sejam registrados sem que uma data de entrada seja registrada.

Premissas

* O projeto será orientado pelo professor Rodrigo Rocha.

Influência das Partes Interessadas

* Marcelo de Almeida Filho: Programador Fullstack. Interessado na conclusão e entrega do projeto.
* Breno Gabriel Rodrigues da Silva: Programador Backend. Interessado na conclusão e entrega do projeto.
* Rodrigo Rocha: Professor responsável por avaliar entregas do projeto e por prestar consultorias.

Representação Arquitetural

Os sistemas serão desenvolvidos tendo como base a arquitetura ilustrada na Figura 1. Toda a arquitetura será baseada nos padrões de projetos tradicionais do GoF e também nos padrões J2EE sendo executados dentro de um Servidor de Aplicações.

Diagrama

Descrição gerada automaticamente

**Figura 1 - Modelo Arquitetural Genérico**

Na camada de View, temos uma classe Controller, as classes de View Helper e Commands. A Controller é uma WebServlet na qual receberá uma Requisição do nosso Servidor Apache Tomcat. As View Helpers vão construir um objeto com base no “action” de operação que foi recebida pela Requisição via Tag “form” da JSP/HTML e devolve esse objeto para Controller. A Command é uma classe que retorna uma operação de CRUD na classe Fachada (Classe de Controle), é invocada devido ao tipo de operação que recebemos via Requisição. Vale ressaltar que as telas de usuário serão construídas em linguagem HTML e utilizando JSP caso seja necessário.

Na Camada de Controle temos a Fachada e a Strategy. A Strategy irá executar as regras de negócio para as operações, e é invocada pela Fachada. A Fachada é invocada por qualquer Command, e nela é onde executamos chamadas para nossas classes de Strategy e Classes de DAO.

No Modelo, temos classe de domínio, classes de DAO e classes de Conexão, sendo as classes de domínio contendo os atributos e métodos getters e setter para encapsulamento. As classes DAO invocam a classe de Conexão para estabelecer uma conexão com nosso Banco de Dados.

* 1. Restrições Arquiteturais

Foram identificadas algumas orientações / restrições pertinentes ao desenvolvimento deste sistema:

* Utilização do JDK 1.8 do Java;
* Utilização do servidor TomCat 9.0;
* Utilização do Banco de Dados MySQL.
  1. Objetivos e Restrições Arquiteturais

Considerando premissas definidas para o Sistema Loja de Carta pode-se citar as seguintes restrições:

* Utilização da Linguagem Java
* Considerar a utilização de software Livre, quando possível
* O Sistema de Gerenciamento de Banco de Dados a ser considerado em implementações de âmbito corporativo será o SGBD MySQLWorkbench ou DBeaver.

1. Visão de Use Case

Esta seção apresenta os Casos de Uso significativos em termos de arquitetura, que foram selecionados considerando-se o pacote do Modelo de Casos de Uso que representa o sistema Loja de Carta.

A classificação dos casos de uso, em termos de significância, foi realizada com base na observação de pelo menos um dos seguintes critérios:

* Casos de uso que estendem outros Casos de Uso
* Casos de Uso que são incluídos em outros Casos de Uso
  1. Diagrama de Caso de Cadastro Dados de Cliente

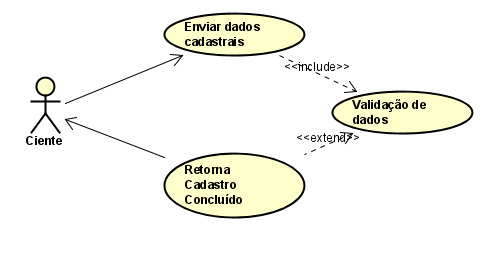


Figura 4.1 Diagrama de Caso de Uso de Cadastro de Cliente

* 1. Descrição dos Casos de Uso Significativos em termos de Arquitetura

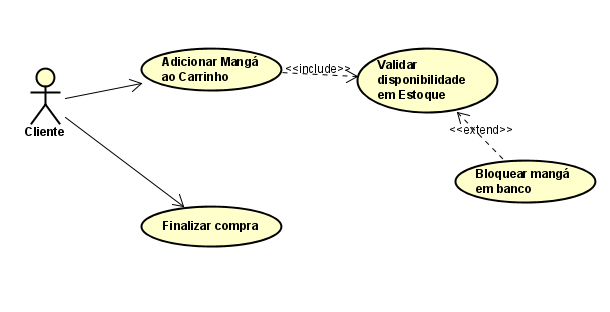
**Manter Produtos**

Diagrama

Descrição gerada automaticamente

Este caso de uso se inicia opcionalmente após a consulta de produtos por pat nunmber e o usuário solicita manutenção (incluir, alterar, excluir logicamente) nos dados de Atributos.

* 1. Descrição dos Casos de Uso de Condução – Venda de Carta



Esse caso de uso de condução é o único que será implementado em sistema e consiste em quando o Cliente adicionar ao carrinho de compras um Carta, o sistema irá bloquear em estoque o Carta para que não haja a perda da venda para esse cliente.

1. Visão de Lógica

Esta visão apresenta elementos de design significativos do ponto de vista da arquitetura, descrevendo a organização do Sistema Loja de Carta em pacotes, bem como a organização desses pacotes em camadas.

O Diagrama com as camadas do sistema Loja de Carta é ilustrado na figura 5.1.

Diagrama

Descrição gerada automaticamente

Figura 5.1 – Diagrama de camadas do Loja de Carta

**Apresentação**: Contém classes para as interfaces gráficas com os usuários (GUI). Através destas interfaces os usuários conseguem interagir com o sistema, com o intuito de incluir, alterar e excluir produtos.

**Negócio**: Contém classes que controlam a execução das funcionalidades do sistema.

**Persistência**: Contém classes responsáveis por persistir as entidades de modelo. Por exemplo,

contém as classes que permitem ler e gravar os objetos no banco de dados relacional.

A Figura 5.2 ilustra o diagrama de camadas com as tecnologias utilizadas no desenvolvimento, já descritas na figura 5.1.

Neste momento, é importante ressaltar que a camada de apresentação envolve componentes que são executados na porção servidora e na porção cliente. Na porção servidora, são executados os componentes que montam as páginas html/xhtml de resposta e controlam o fluxo de interação com o usuário. Assim, envolve interpretação de jsf, utilização de ManageBeans para montagem das páginas de resposta, delegação de solicitação para a camada de negócio, obtenção de respostas da camada de negócio, dentre outros. Já na porção cliente, estão os códigos javascript que são executados nos navegadores dos usuários com o intuito de facilitar a interação. Por exemplo, podem fazer uma pré-verificação dos dados digitados (como atribuir o valor restante a ser pago no segundo cartão após entrada de valor a ser pago no primeiro cartão no cenário de efetuar pagamento usando de 2 cartões de crédito).

Diagrama

Descrição gerada automaticamente

Figura 5.2 – Camadas do sistema com as dependências de tecnologia

* 1. Camada de Apresentação

Nesta camada, temos o pacote form que contém todos os arquivos relacionados à exibição de informações para usuário, o que engloba páginas JSP e HTML, imagens, javascript, dentre outros. Já o pacote de controle desta camada, contém as Actions do Struts responsáveis pela comunicação com as classes da camada de negócio. Estes pacotes podem ser vistos na Figura 5.3.

Diagrama

Descrição gerada automaticamente

Figura 5.3: Camada de Apresentação

## Camada de Negócio

Nesta camada, temos o pacote controle que contém as classes responsáveis por controlar as regras de negócio da aplicação. O pacote domínio, contém as classes que representam o modelo, ou seja, aquelas que contém as informações sobre o Sistema NewStation.

A imagem 5.4 ilustra os pacotes descritos.

Diagrama

Descrição gerada automaticamente

Figura 5.4: Camada de View

* + 1. Pacote Controle

A figura 5.5 ilustra as principais classes de controle.

Diagrama

Descrição gerada automaticamente

Figura 5.5: Classes de controle

* + 1. Pacote Modelo

A figura 5.6. ilustra as principais classes do modelo.

Diagrama

Descrição gerada automaticamente

Figura 5.6: Classes do Modelo

## Camada de Persistência

Nesta camada temos o pacote “dao” que contém as classes responsáveis por persistir as informações do sistema no BD relacional.

Diagrama

Descrição gerada automaticamente

## Realização dos Casos de Uso Significativos

Diagrama

Descrição gerada automaticamente

1. Visão de Implantação

Esta seção descreve as configurações da rede física (hardware) na qual o NewStation será implantado e executado.

Trata-se de uma visão do Modelo de Implantação que, para a configuração em questão, indica os nós físicos (computadores, CPUs), que executarão o subsistema NewStation, e as respectivas interconexões (barramento, LAN, etc). A figura 6 ilustra o modelo de implantação para o NewStation.

1. Visão de Implementação

Esta visão descreve a estrutura geral de implementação, a decomposição do software em camadas de implementação.

A estrutura geral de implementação para o NewStation é baseada na estrutura da Visão Lógica, assim, não há necessidade de detalhar os diagramas de camadas e pacotes de implementação, uma vez que são fortemente baseados naqueles desenvolvidos para Visão Lógica.

1. Visão de Dados

O mecanismo de persistência utilizado no sistema Loja de Carta utiliza-se o banco de dados Relacional MySQL.

A figura 7, apresenta a visão física da base de dados do sistema Loja de Carta.

Diagrama

Descrição gerada automaticamente

Figura 7 – Modelo Físico

Tabela 2: Mapeamento Objeto-Relacional

1. Tamanho e Performance

O sistema Loja de Carta será usado para o controle do comércio exterior e consequentemente terá uma grande base.

Seus servidores provavelmente irão passar por períodos de picos de utilização (por exemplo, próximo a data da Black Friday).

As estimativas do número de usuários e de carga de utilização em períodos de pico de utilização, bem como maiores informações sobre questões relacionadas ao tamanho e desempenho do sistema Loja de Carta podem ser obtidas no documento de requisitos não funcionais.

1. Qualidade

O sistema Loja de Carta será usado para venda de Cartas, tendo assim um público bastante específico e não tendo grandes quantidades de acessos simultâneos, descartando a necessidade de um servidor de alto desempenho, diminuindo seu custo de manutenção.

Adicionalmente, o sistema Loja de Carta pode ser alvo de ataques de “hackers” para roubar ou simplesmente corromper informações, possibilidade aumentada pela interface do sistema disponível na Internet, para evitar que tais ataques sejam bem-sucedidos uma infraestrutura de segurança deve ser especificada e projetada.

Maiores informações sobre questões relacionadas aos requisitos de qualidade do sistema Loja de Carta podem ser obtidas no documento de requisitos não funcionais.

1. Cronograma Macro.

|  |  |
| --- | --- |
| **Resultado** |  |
| Plano Preliminar | Semana 2 |
| Plano Fase 1 | Semana 3 |
| Especificação Fase 1 | Semana 5 |
| Piloto Fase 1 | Semana 11 |
| Solução Testada Fase 1 | Semana 12 |
| Plano Fase 2 | Semana 13 |
| Especificação Fase 2 | Semana 14 |
| Piloto Fase 2 | Semana 20 |
| Solução Testada Fase 2 | Semana 21 |

Obs: Os prazos apresentados são uma estimativa inicial considerando as informações disponíveis nesta etapa do projeto. Um cronograma detalhado será elaborado na fase de planejamento e, eventualmente, estes prazos podem ser modificados.

1. Referências

Unified Modeling Language: <http://www.omg.org/technology/documents/formal/uml.htm>

RUP. Rational Unified Process.