**Documento de Visão de Projeto**



**Realm of Cards**

**Histórico de Versões**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** | **Revisor** |
| 09/05/20 | 1.0 | Modelagem e desenvolvimento | Lucas de A. R. Cascão | - |

|  |  |
| --- | --- |
| **Cliente** | Rodrigo Rocha - Interno |
| **Documento** | Documento de Visão de Projeto: *Realm of Cards* |
| **Data** | 09 de maio de 2020 |
| **Autor** | **Lucas De Almeida Rama Cascão**  lucas.cascao@fatec.sp.gov.br |

**Página de Assinaturas**

|  |  |  |
| --- | --- | --- |
| Revisado e Aprovado por: |  |  |
|  |  | 09/05/20 |

**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 do sistema E-Commerce Realm of Cards. O intuito é salientar diferentes aspectos deste produto, obtidos a partir de decisões arquiteturais realizadas no âmbito do sistema E-Commerce Realm of Cards.

* 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 informais com representante do Realm of Cards.

Este documento influencia os seguintes documentos:

* Documento de Requisitos;
* Posposta Técnica Comercial.

Necessidades de Negócio

Um sistema informatizado de vendas online é necessário para que a Realm of Cards possa ganhar mais espaço no mercado, vendendo para todo do Brasil, podendo gerenciar seus produtos, clientes e vendas. O sistema possibilita um controle total sobre as transações de vendas, administração dos produtos e monitoramento dos clientes.

Objetivo do Projeto

Desenvolver uma plataforma para soluções web capaz de:

* Armazenar informações em uma base de dados;
* Utilizar o protocolo HTTP;
* Ser executado em qualquer navegador;

* Controlar as cartas presentes no estoque da loja e entrada de novas cartas;
* Gerenciamento das operações realizadas no e-commerce: venda, troca e devolução de cartas.

Declaração Preliminar de Escopo

Esta seção descreve, em alto nível, o escopo do projeto. Os requisitos serão descritos mais detalhadamente no documento de Requisitos.

* 1. Descrição

Cliente:

* Acesse o sistema;
* Faça login com usuário e senhas validas. Caso não possua, faça um cadastro clicando no botão ‘Cadastre-se’;
* Escolha o produto desejado clicando em comprar;
* Escolha a quantidade;
  1. Produtos a serem entregues

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

* Sistema E-Commerce, implementado de acordo com a especificação feita na fase de análise.
* Documentos de especificação do sistema, concebido na fase de elaboração.
  1. Requisitos

É importante salientar que o documento de visão estará sendo feito no momento de iniciação do projeto. No qual, ainda não se tem todo o detalhamento. Portanto, os requisitos devem descritos em alto nível e depois detalhados em outro documento, caso o projeto seja aprovado.

* + 1. Requisitos Funcionais
* O sistema deve ser capaz de efetuar o cadastro, exclusão, alteração e consulta de cartas do estoque da loja;
* O sistema deve ser capaz de efetuar o controle das vendas;
* O sistema deve ser capaz de fazer o controle da devolução dos produtos comprados;
* O sistema deve ser capaz de produzir uma listagem com os pedidos;
* O sistema deve ser capaz de produzir relatórios referentes as operações do sistema;
* O sistema deve ser capaz de realizar consulta a históricos;
* O sistema deve ser capaz de realizar cadastro, exclusão, alteração e consulta de clientes, bem como seus cartões e endereços.

* + 1. Requisitos Não Funcionais
* Utilizar linguagem Java;
* Utilizar o banco de dados PostgreSQL;
* O sistema deve obedecer o design pattern MVC.

O sistema deve rodar nos seguintes browsers:

* IE;
* Google Chrome;
* Firefox;
* Cronograma de Marcos Sumariado;
  + 1. Regras de Negócio
* Para toda carta cadastrada é obrigatório o cadastro dos seguintes dados: título, categoria da carta, preço de compra, preço de venda, código de barras, descrição, uma imagem da carta. Todos os campos anteriores são de preenchimento obrigatório.
* Para todo cliente cadastrado é obrigatório o cadastro dos seguintes dados: Gênero, Nome, Data de Nascimento, CPF, e-mail, senha, endereço residencial. Todos os campos anteriores são de preenchimento obrigatório.
* 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.
* Para todo cliente cadastrado é obrigatório o registro de ao menos um endereço de cobrança e entrega.
* 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. Todos os campos anteriores são de preenchimento obrigatório.
* 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.
* Ao solicitar a compra de itens que estejam em um carrinho deve-se garantir que tais itens ainda permanecem disponíveis em estoque.
* Apenas um cupom promocional pode ser utilizado por compra.
* 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.

Premissas

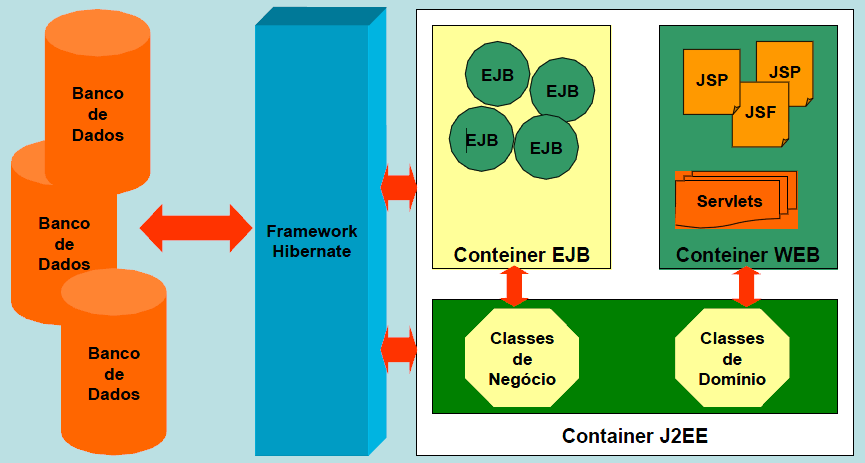
* O projeto será orientado pelo professor Rodrigo Rocha.
* Será necessário internet banda larga para que o projeto seja utilizado.

Influência das Partes Interessadas

* Rodrigo Rocha: Professor da FATEC e orientador do projeto. Interessado em acompanhar as entregas do projeto.
* Lucas De Almeida Rama Cascão: Analista desenvolvedor do sistema. Interessado em realizar o desenvolvimento do projeto.

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.



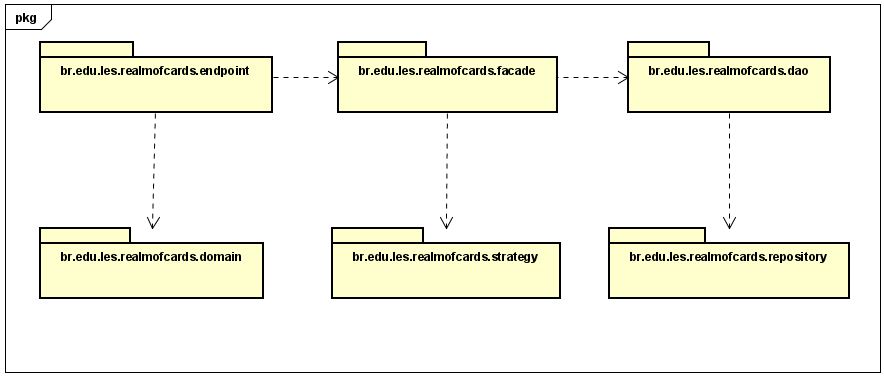
**Figura 1 - Modelo Arquitetural Genérico**

O View abrigará os componentes da arquitetura responsáveis pela camada de apresentação

As Classes de Domínio são as classes que representam os Value Object, contendo somente os atributos e os métodos getters/setters.

As Classes de Negócio representam as classes responsáveis por aplicar as regras de negócio do sistema como, por exemplo, cadastrar jogo. Constarão também nas classes de negócio o relacionamento com os DAOs responsáveis por persistir e recuperar os objetos no banco de dados.

O diagrama a seguir, Figura 2, representa a organização das classes dentro dos pacotes.



**Figura 2 - Diagrama em pacotes**

* 1. Restrições Arquiteturais

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

* Utilização do JDK 1.8 do Java;
* Utilização do framework Spring Boot;
* Utilização do framework Bootstrap 4 para desenvolvimento da camada de apresentação;
* Utilização do framework Angular 8;
* Utilização do Banco de Dados PostgresSQL.
  1. Objetivos e Restrições Arquiteturais

Considerando premissas definidas para os Sistema E-Commerce Zero1Games pode-se citar as seguintes restrições:

* Utilização de Linguagem orientada a objetos.
* Utilização de Banco de dados relacional.

1. Visão de Use Case

Esta seção apresenta os Casos de Uso arquiteturalmente significativos, que foram selecionados considerando-se o pacote do Modelo de Casos de Uso que representa o sistema E-Commerce Realm of Cards.

Diagrama de Caso de Manter Dados de Atributos

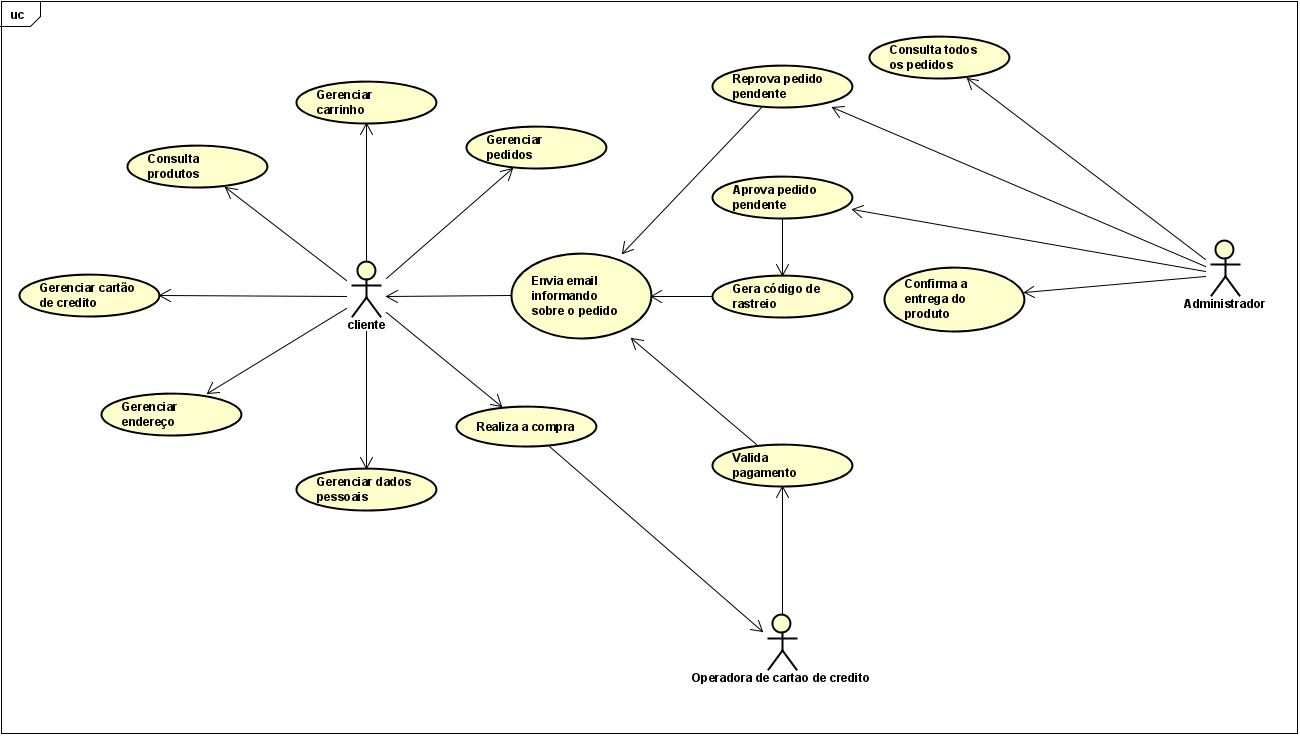


Figura 4.1 Diagrama de Caso de Uso de Venda (E-Commerce Realm of Cards)

* 1. Descrição dos Casos de Uso Arquiteturalmente Significativos

**Gerenciar Carrinho**

Esse caso de uso se trata da possibilidade de o cliente poder inserir produtos, retirar produtos ou alterar a quantidade de unidades do produto escolhido dentro do carrinho.

**Enviar Email Informando Sobre o Pedido**

Esse caso de uso se trata de quando o pedido é atualizado, é enviado um e-mail para o cliente informando sobre a atualização do pedido.

1. Visão de Lógica

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

O Diagrama com as camadas do sistema XXX é ilustrado na figura 5.1.



Figura 5.1 – Diagrama de camadas do XXXX

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

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

**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/jsp de resposta e controlam o fluxo de interação com o usuário. Assim, envolve interpretação de páginas jsp, utilização de helpers 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 o preenchimento de campos obrigatórios), solicitação de páginas via ajax, de modo a tornar mais dinâmica a interação com o usuário, dentre outros.



Figura 5.2: Camadas do xxx 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.



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 model, contém as classes que representam o modelo, ou seja, aquelas que contém as informações sobre o Sistema XXX. O pacote controller contém as classes de controle do negócio. O pacote complementador contém classes que auxiliam na complementação de informações e o pacote filtro contém as classes que auxiliam na filtragem de informações.

A imagem 4.4 ilustra os pacotes descritos.



Figura 5.4: Camada de Negócios

* + 1. Pacote Controller

A figura 5.5 ilustra as principais classes de controle.



Figura 5.5: Classes de controle

* + 1. Pacote Model

A figura 5.6. ilustra as principais classes do modelo.



Figura 5.6: Classes do Modelo

## Camada de Persistência

Nesta camada temos o pacote dao que contém as classes e interfaces responsáveis por persistir as informações do XXXX no BD relacional. O pacote hibernate contido em dao, possui as classes que dependem diretamente do Hibernate, que é o framework utilizado para realizar o mapeamento objeto relacional.



Figura 5.7: Camada de Persistência.

## Realização dos Casos de Uso Significativos



1. Visão de Implantação

Esta seção descreve as configurações da rede física (hardware) na qual o XXX 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 XXXX, e as respectivas interconexões (barramento, LAN, etc). A figura 6 ilustra o modelo de implantação para o XXXX.



Figura 6: Visão de Implantação do XXXX

Na Figura 6 observa-se os seguintes nós físicos:

* **Web Client Application (WCA):** Aplicativos com interface de usuário via navegador, construídos com base no Framework Struts.
* **Servidor DB**: Nó que contém o BD Central do Sistema XXX.

Descrever demais elmentos

1. Visão de Implementação

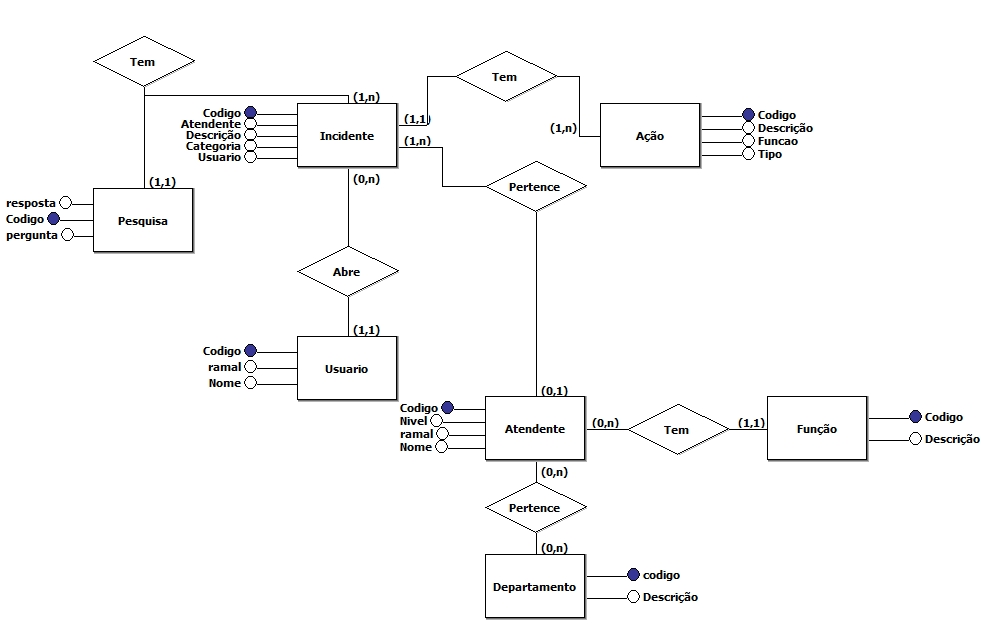
Esta visão descreve a estrutura geral de implementação, a decomposição do software em camadas de implementação.

A estrutura geral de implementação para o SISCAD 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.

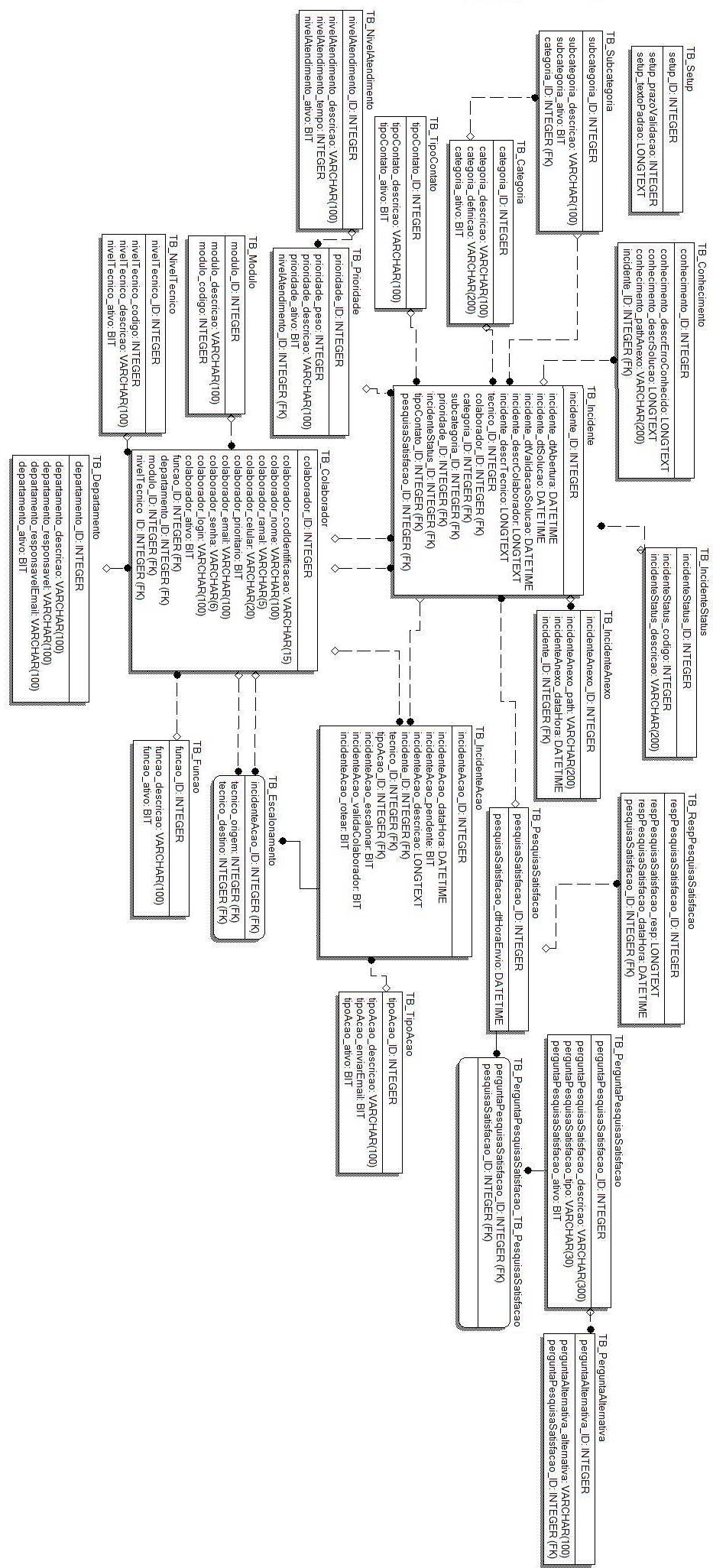
1. Visão de Dados

O mecanismo de persistência utilizado no sistema SISCAD utiliza-se o banco de dados Relacional Oracle juntamente com o framework para mapeamento objeto-relacional, Hibernate. O controle de transações adotado envolve a utilização do Spring Framework em conjunto com o Hibernate.

As figuras XXX e YY, apresentam a visão lógica e física da base de dados do SISCAD.

****

Figrua XXX – Modelo Lógico



Figrua XXX – Modelo Físico

A Tabela 2 define o mapeamento das principais classes de modelo para entidades do modelo lógico do BD Oracle.

Note que existem alguns campos nas entidades lógicas do BD que não estão mapeadas diretamente com as classes de modelo da Visão Lógica contidas neste documento:

Os campos que começam com “IN” são campos definidos através de domínios do BD. As classes de modelo que possuem os atributos que representam tais campos não constam neste documento, mas podem ser consultadas no documento [REF – Modelo de Design].

Os campos #NRSQ e DT REGISTRO presentes em todas as entidades lógicas do BD, correspondem a atributos presentes em superclasses das classes de modelo, que não constam neste documento, mas podem ser consultadas no documento [REF – Modelo de Design].

Tabela 2: Mapeamento Objeto-Relacional

|  |  |
| --- | --- |
| **Classe** | **Entidade** |
|  |  |
|  |  |

1. Tamanho e Performance

O sistema XXXX será usado para o controle do comércio exterior e conseqüentemente terá uma grande base.

Seus servidores provavelmente irão passar por períodos de picos de utilização (por exemplo, próximo a finalização de prazos de entrega de documentos e/ou procedimentos fiscais, tal como ocorre com os servidores que recebem declarações de imposto de renda).

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 XXX podem ser obtidas no documento de requisitos não funcionais.

1. Qualidade

O sistema XXX será usado para o controle do comércio exterior brasileiro, conseqüentemente tratando de altos volumes financeiros e um grande número de operações de importação/ exportação diariamente.

Eventuais erros e/ou falhas na sua operação podem levar a prejuízos significativos tanto em termos financeiros quanto na imagem da UMC, portanto na fase de design deve-se levar em consideração como fatores prioritários a confiabilidade e robustez do sistema.

Adicionalmente, o sistema XXX 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 infra-estrutura de segurança deve ser especificada e projetada.

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

1. Cronograma Macro.

Os prazos podem ser dados em semanas ou meses dentro do projeto ou ate mesmo data.

|  |  |
| --- | --- |
| **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.