**Documento de Visão de Projeto**



**ImoX**

**Histórico de Versões**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** | **Revisor** |
| 24/09/2025 | 1.0 | Modelagem e desenvolvimento | Kleberson dos ss | - |

|  |  |
| --- | --- |
| **Cliente** | FATEC - Interno |
| **Documento** | Documento de Visão de Projeto: Hatchards |
| **Data** | 24 de setembro de 2025 |
| **Autor** | **Kleberson dos santos silva** |

**Página de Assinaturas**

|  |  |  |
| --- | --- | --- |
| Revisado e Aprovado por: |  |  |
|  |  |  |

**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

Desenvolver um e-commerce de produtos diversificados que utilize dados dos clientes para personalizar recomendações de produtos e emitir cupons promocionais, aumentando a fidelização e as vendas, por meio de integração com sistemas de IA e gestão eficiente das informações, utilizando arquitetura limpa, DDD e MySQL.

* 1. Escopo

Este documento descreve o desenvolvimento de um módulo de e-commerce que atende às necessidades de gerenciamento de produtos e clientes, abrangendo funcionalidades como cadastro de produtos, cadastro de clientes, gestão de estoque, carrinho de compras e painel administrativo. Este módulo é parte de um sistema maior e se integra às demais funcionalidades do e-commerce. Do ponto de vista arquitetural, o módulo será estruturado utilizando **arquitetura limpa (Clean Architecture)** em um **monólito**, com divisão em pacotes principais: **Cliente**, **Produto**, **Pedido** e **Administração (Adm)**. O documento detalha a organização em subsistemas, pacotes, classes e utilitários de classe significativos para o design e funcionamento do módulo.

* 1. Referências

Para a construção deste documento foram utilizadas as seguintes referências:

* Reuniões informais com o cliente
* Documentos elaborados pela equipe e cliente:

Documento de requisitos,

Casos de teste do sistema,

DVP,

Casos de teste E2e Cliente,

Estimativas de projeto

Este documento influencia os seguintes documentos:

* Documento de Requisitos
* WBS

Necessidades de Negócio

O principal objetivo do projeto é fornecer um módulo de e-commerce que permita à empresa obter maior controle sobre os dados dos clientes e utilizá-los para aumentar as vendas. As necessidades incluem:

* **Personalização de ofertas e recomendações:** possibilitar que o sistema sugira produtos com base no histórico de compras do cliente ou em interações com a IA, aumentando a relevância das ofertas.
* **Emissão de cupons promocionais:** gerar campanhas de incentivo que fidelizem clientes e estimulem novas compras.
* **Gestão eficiente de vendas e estoque:** controlar produtos, pedidos e clientes de forma integrada, garantindo operação consistente e confiável do e-commerce.
* **Integração com sistemas de IA:** permitir análises inteligentes e geração de recomendações automatizadas, alinhando tecnologia à estratégia de vendas.

A implementação destas funcionalidades atende às expectativas do cliente, promove fidelização, aumenta receita e reduz risco de perda de valor frente à concorrência.

Objetivo do Projeto

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

* Criar propagas e recomendações baseada nos interesses do cliente
* Integrar com IA
* Painel de Administrador

Maximizando o Lucro e aumentando o AVL da empresa através da recomendação e fidelização de cliente através dos cupons

Declaração Preliminar de Escopo

* 1. Descrição

Desenvolver um módulo de e-commerce que integre gestão de produtos, clientes e pedidos, utilizando dados e inteligência artificial para personalizar recomendações, emitir cupons promocionais e aumentar a fidelização e as vendas

* 1. Produtos a serem entregues

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

* Sistema do módulo do e-commerce, 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 (Casos de teste do sistema, protótipo do site, testes e2e cliente e estimativas do projeto).
  1. Requisitos
     1. Requisitos Funcionais

|  |
| --- |
| O sistema deve manter um cadastro único para livros. |
| O sistema deve possibilitar que livros sejam inativados. |
| O sistema deve inativar livros sem estoque e que não possuem venda com valor inferior a parâmetro predefinido no sistema. |
| O sistema deve possibilitar a alteração de dados cadastrais para os livros. |
| O sistema deve possibilitar que um livro seja consulta  do com base em um filtro definido pelo usuário. Todos os campos utilizados para identificação do livro podem ser utilizados como filtro, tanto de forma combinada como de forma isolada. |
| Deve ser possível ativar o cadastro de um livro. |
|  | |
| Grupo: Cadastro de Clientes | |
| O sistema deve possibilitar o cadastro de clientes. |
| O sistema deve possibilitar a alteração de dados cadastrais de clientes. |
| O sistema deve possibilitar que clientes sejam inativados. |
| 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. |
| O sistema deve disponibilizar no cadastro de clientes a consulta de todas as transações já realizadas pelo mesmo. |
| 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. |
| Deve ser possível associar diversos cartões de crédito ao cadastro de um cliente. Deve haver um cartão de crédito configurado como preferencial. |
| 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 | |
| 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. |
| 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. |
| Deve ser possível a partir de um carrinho de compra realizar uma compra. |
| O sistema deve calcular o frete da compra com base nos itens selecionados e o endereço apontado pelo cliente. |
| 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 o mesmo seja incorporado ao perfil do cliente. |
| 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 o mesmo 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. |
| 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. |
| 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. |
| O sistema deve possibilitar que um usuário com perfil de administrador confirme entrega de uma compra. Assim o status deve ficar ENTREGUE. |
| O sistema deve possibilitar que um item de uma compra seja trocado por um cliente através da visualização de pedidos do mesmo. |
| O sistema deverá possibilitar que o administrador autorize pedidos ou compra com status EM TROCA. Assim o pedido passa ficar com status TROCA AUTORIZADA. |
| O sistema deverá possibilitar que o administrador visualize todos pedidos de troca ou compra com status EM 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. |
| 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 | |
| O sistema deve permitir que seja possível realizar entrada de itens de livros em estoque.  No registro de cada item, deve ser indicado o livro já previamente cadastrado e a quantidade de itens do livro. |
| 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 ao livro. |
| Para cada venda realizada deve-se dar baixa no estoque do total de itens vendidos. |
| O sistema deve realizar a reentrada de um item em estoque a partir da troca de um produto. |
|  |
| Grupo: Análise | |
| O sistema deve possibilitar que o usuário consulte o histórico de vendas comparando produtos ou categorias de produtos a partir de uma busca por período, considerando uma data de início e uma de fim. |

* + 1. Requisitos Não Funcionais

Utilizar linguagem Java

Utilizar o banco de dados Mysql

A arquitetura Limpa.

O sistema deve rodar nos seguintes browsers:

* Google Chrome
* Firefox 
  + 1. Regras de Negócio

|  |
| --- |
| Para todo cliente cadastrado é obrigatório o registro de ao menos um endereço de cobrança. |
| Para todo cliente cadastrado é obrigatório o registro de ao menos um endereço de entrega. |
| Todo cadastro de endereços associados a clientes deve ser composto dos seguintes dados: Tipo de residência (Casa, Apartamento, etc), Tipo Logradouro, Logradouro, Número, Bairro, CEP, Cidade, Estado e País. Todos os campos anteriores são de preenchimento obrigatório. Opcionalmente pode ser preenchido um campo observações. |
| 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. |
| Todo cartão de crédito associado a um cliente deverá ser de alguma bandeira registrada no sistema. |
| 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. |

Premissas

* O projeto será orientado pelo professor Rodrigo Rocha.

Influência das Partes Interessadas

Rodinei Perassol Isquierdo: diretor da Houter e patrocinador do projeto. Interessado em minimizar o custo operacional de atendimento da Houter, assim como, acompanhar a tendência dos seus concorrentes.

Pedro Mendes Junior: gerente de marketing da Houter e da Premium. Interessado em expandir os canais de comunicação com os clientes, assim como, fixar a marca de ambas as empresas como inovadoras em termos de tecnologia.

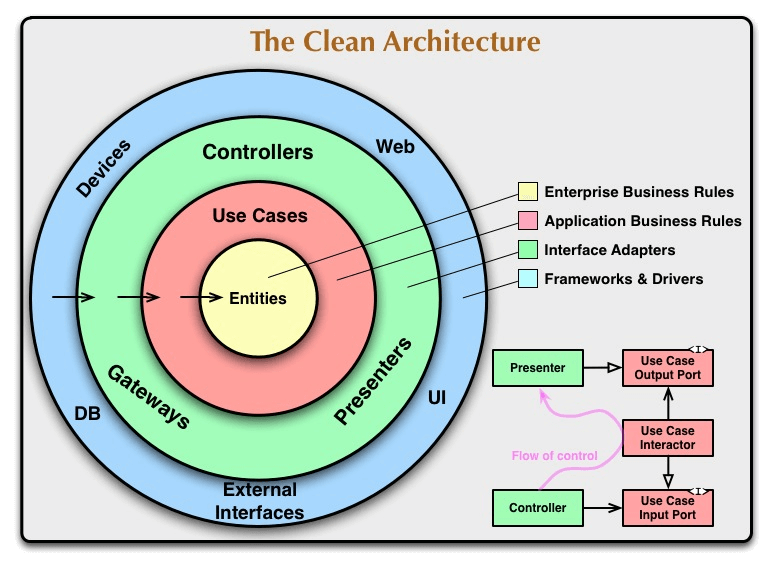
Afonso Santos Filho: gerente da Premium. Interessado em expandir os canais de venda.

Regina: responsável pelo sistema ERP de ambas as empresas. Interessada que a solução de e-Commerce seja integrada ao sistema interno atual para facilitar o fluxo das operações.

ClickNow: empresa responsável pela especificação da navegabilidade, definição do design, definição do conteúdo, desenvolvimento do site estático e campanhas de marketing do e-Commerce.

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 sendo executados dentro de um Servidor de Aplicações.



**Figura 1 - Modelo Arquitetural Genérico**

O sistema será desenvolvido seguindo os princípios da **Clean Architecture**, estruturado em camadas que se comunicam de forma unidirecional, garantindo modularidade e fácil manutenção.

**Camadas principais:**

* **Interface / API:**  
  Responsável pela comunicação com o usuário ou outros sistemas, fornecendo endpoints REST para operações do e-commerce e recebendo dados de entrada e saída.
* **Application / Service:**  
  Contém os casos de uso do sistema, orquestrando as operações e regras de negócio de aplicação, sem depender de detalhes de persistência ou infraestrutura.
* **Domain / Modelo:**  
  Representa o núcleo do negócio, incluindo entidades, objetos de valor e agregados, encapsulando as regras de negócio puras do e-commerce.
* **Infrastructure / Persistence:**  
  Gerencia a persistência dos dados e integrações externas, utilizando tecnologias como Spring Data JPA e Hibernate para armazenar informações no banco de dados MySQL.

O sistema será monólito, permitindo que cada camada evolua de forma independente, mantendo baixo acoplamento e alta coesão, enquanto integra funcionalidades como gestão de clientes, produtos, pedidos e administração de forma consistente e escalável. Em uma futura evolução poderá ser usado a arquitetura, microsserviso modular.

* 1. Restrições Arquiteturais

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

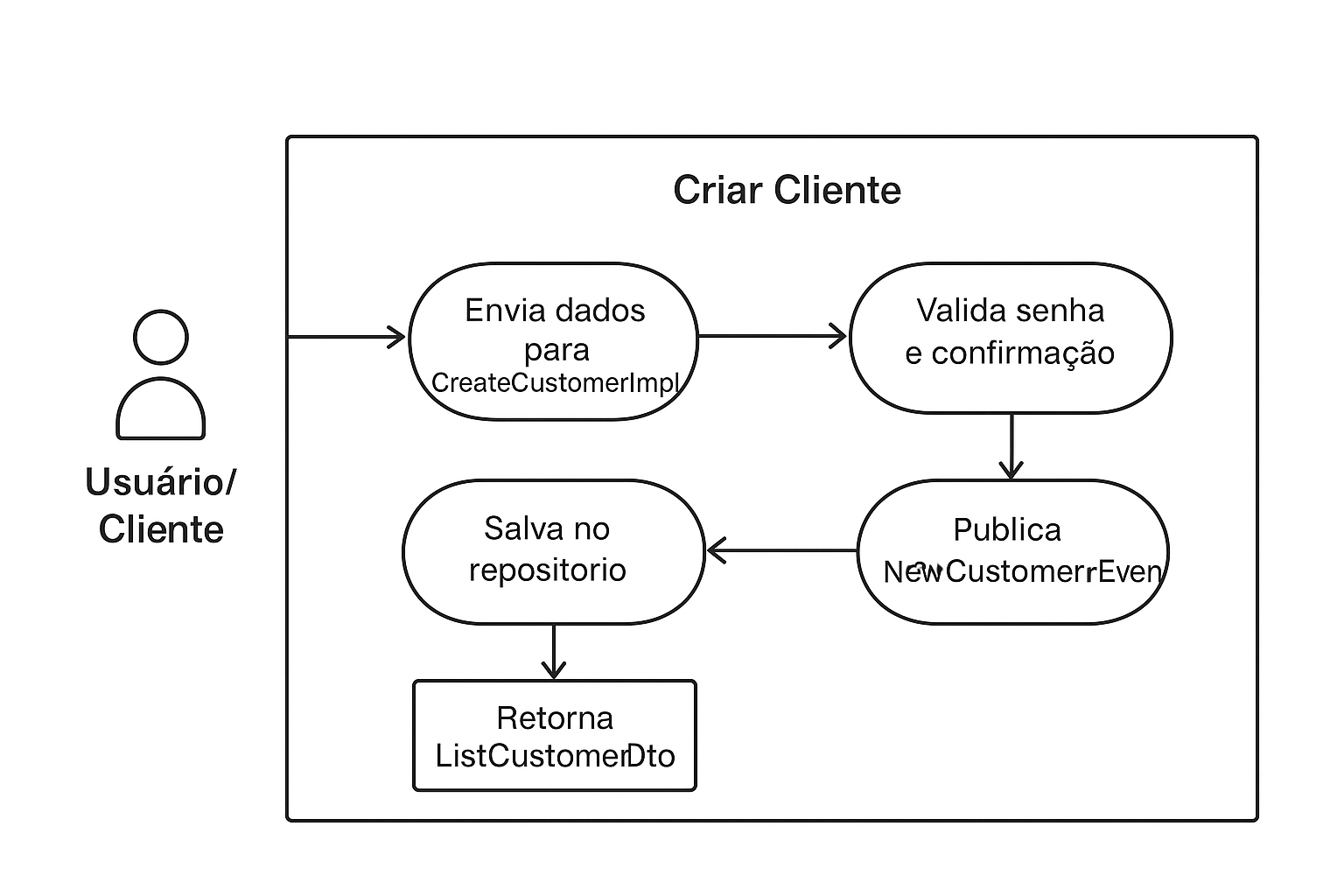
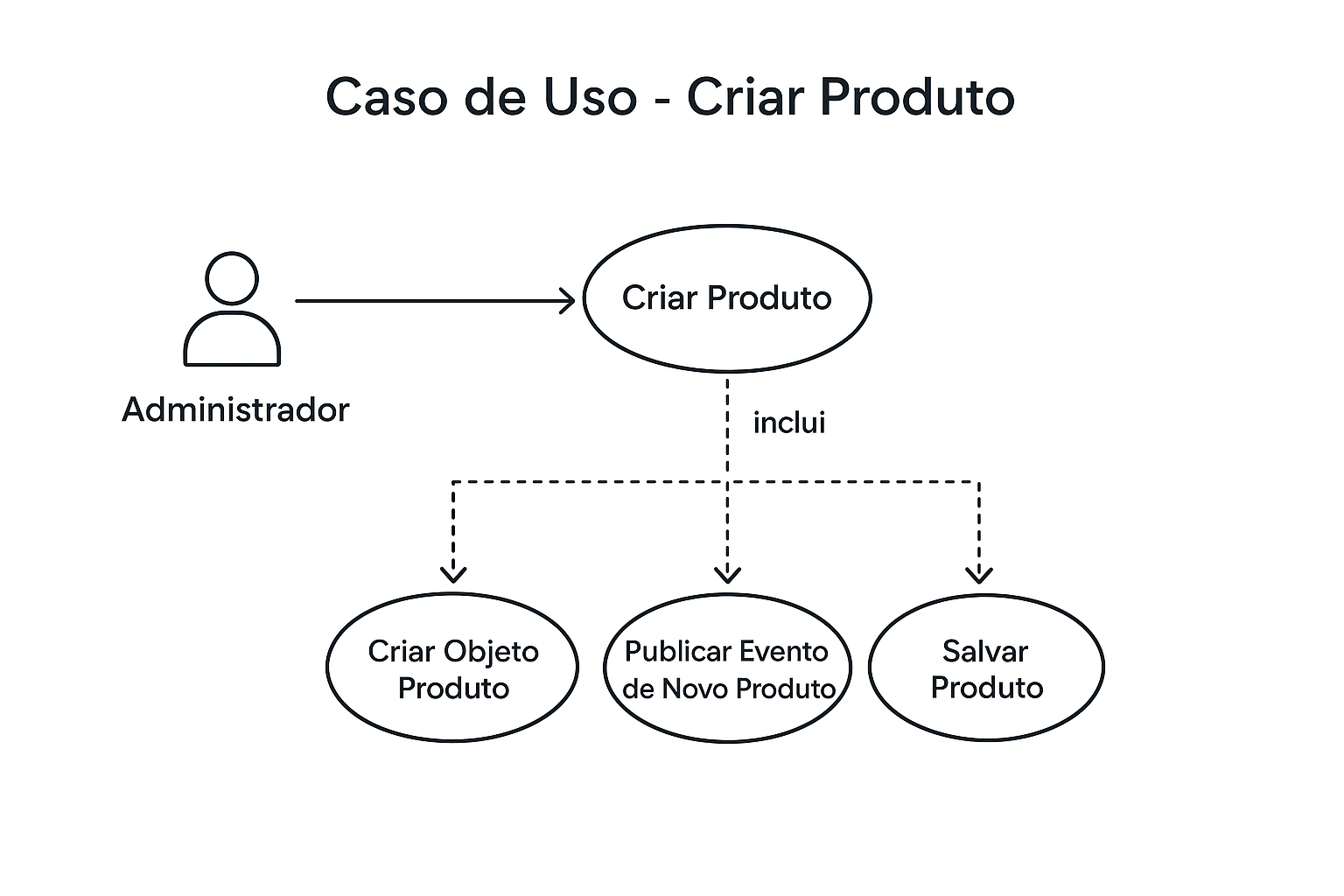
* Utilização do JDK 17 do Java;
* Utilização de Html, Css e JS para o Front End;
* Utilização do framework Spring para desenvolvimento da camada de negócio e controle de transações;
* Utilização do framework de mapeamento objeto-relacional Hibernate;
* Utilização do SGBD Mysql.

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 xxxxx.

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 e
* Casos de uso que acessam sistemas externos
  1. Diagrama de Caso de Uso : Criar Cliente

* 1. Descrição dos Casos de Uso Arquiteturalmente Significativos

**Manter Produtos**

**Criar Cliente**

**1) Usuário/Cliente** — envia CreateCustomerDto (nome, e-mail, senha, confirmação).

**2) Valida senha e confirmação** — checa igualdade e formato aceitável.

**3) Cria objeto** Customer — converte DTO em entidade de domínio.

4) **Salva no repositório** — persiste Customer.

**5) Publica** NewCustomerEvent — notifica outros subsistemas (ex.: envio de e-mail).

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 E-commerce é ilustrado na figura 5.1.



Figura 5.1 – Diagrama de camadas E-commerce

**Apresentação**: Na visão lógica do sistema **e-commerce**, a **camada de apresentação** é a porta de entrada para todas as interações externas via **REST API**.  
Seu objetivo é expor endpoints HTTP, validar dados de entrada básicos e traduzir as requisições em comandos que a aplicação entende, sem incorporar regras de negócio ou detalhes de persistência.

**Negócio**: Na visão lógica do sistema **e-commerce**, a **camada de negócio** representa o **coração do domínio**.  
Aqui residem todas as **regras e políticas que regem o funcionamento do comércio eletrônico**, isoladas de detalhes de infraestrutura ou de apresentação.  
Ela define **o que** o sistema faz, não **como** a tecnologia executa..

**Persistência**: Na visão lógica do sistema **e-commerce**, a **camada de persistência** é responsável por **armazenar e recuperar dados** de forma transparente para as demais camadas.  
Ela implementa os contratos definidos no domínio (interfaces de repositório), garantindo que a lógica de negócio permaneça independente de detalhes de banco de dados ou de frameworks.

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

No front-end, a camada de apresentação é implementada em **HTML, CSS e JavaScript**, executando diretamente no navegador do usuário. Ela é responsável por montar a interface, consumir a REST API por meio de requisições AJAX ou fetch e atualizar dinamicamente a página sem necessidade de recarregamentos completos. Além disso, pode realizar validações preliminares, como a checagem de campos obrigatórios e a formatação de dados, antes de enviar as requisições para o back-end.



Figura 5.2: Camadas do e-commerce com as dependências de tecnologia

* 1. Camada de Apresentação

Nesta camada, o pacote **form** reúne todos os arquivos relacionados à apresentação de informações ao usuário, incluindo páginas, HTML, JavaScript, entre outros. Já o pacote de **controle** da camada abriga as controllers, DTOs e casos de uso responsáveis por gerenciar a comunicação com as classes da camada de negócio. Esses pacotes estão ilustrados 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.

* + 1. Pacote Model

A figura 5.6. ilustra as principais classes do modelo.

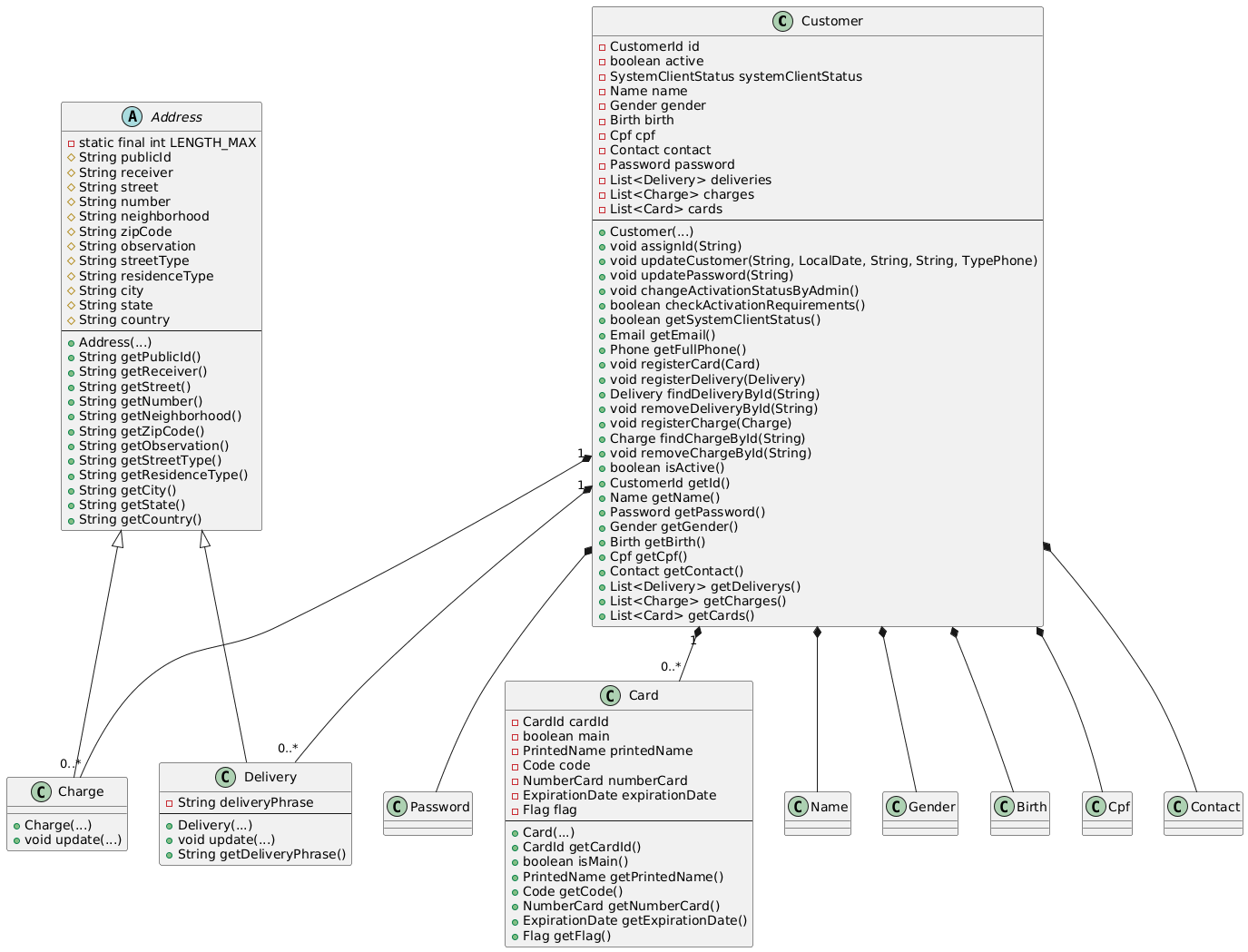


Figura 5.6: Classes do Modelo

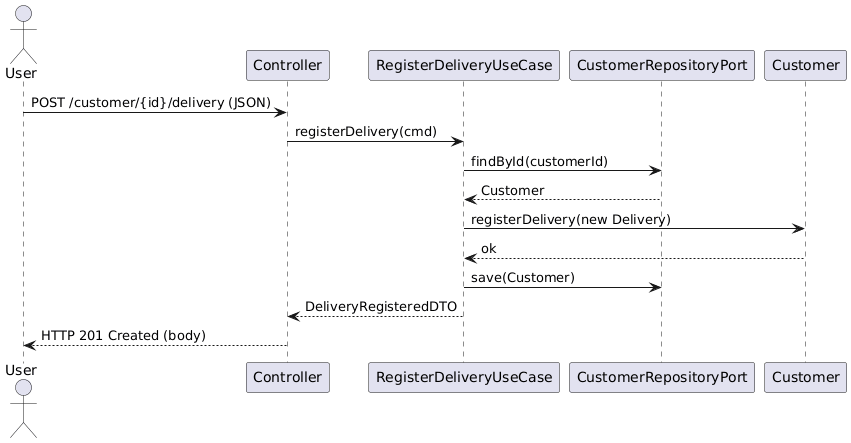
## Camada de Persistência

Nesta camada temos o pacote dao que contém as classes e interfaces responsáveis por persistir as informações do sistema 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 e-commerce 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.

A figura 6 ilustra o modelo de implantação para o XXXX.

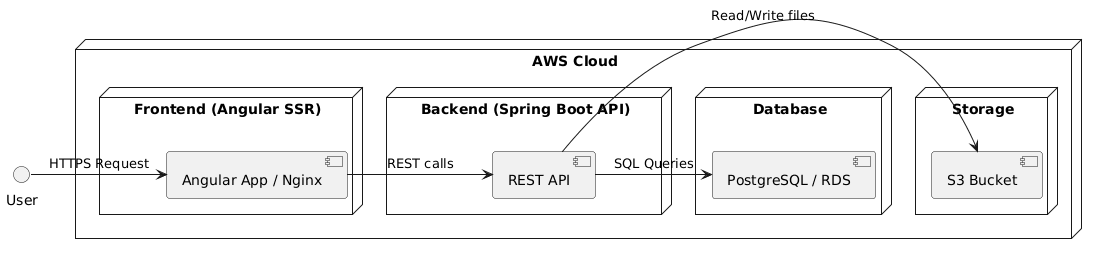


Figura 6: Visão de Implantação do XXXX

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

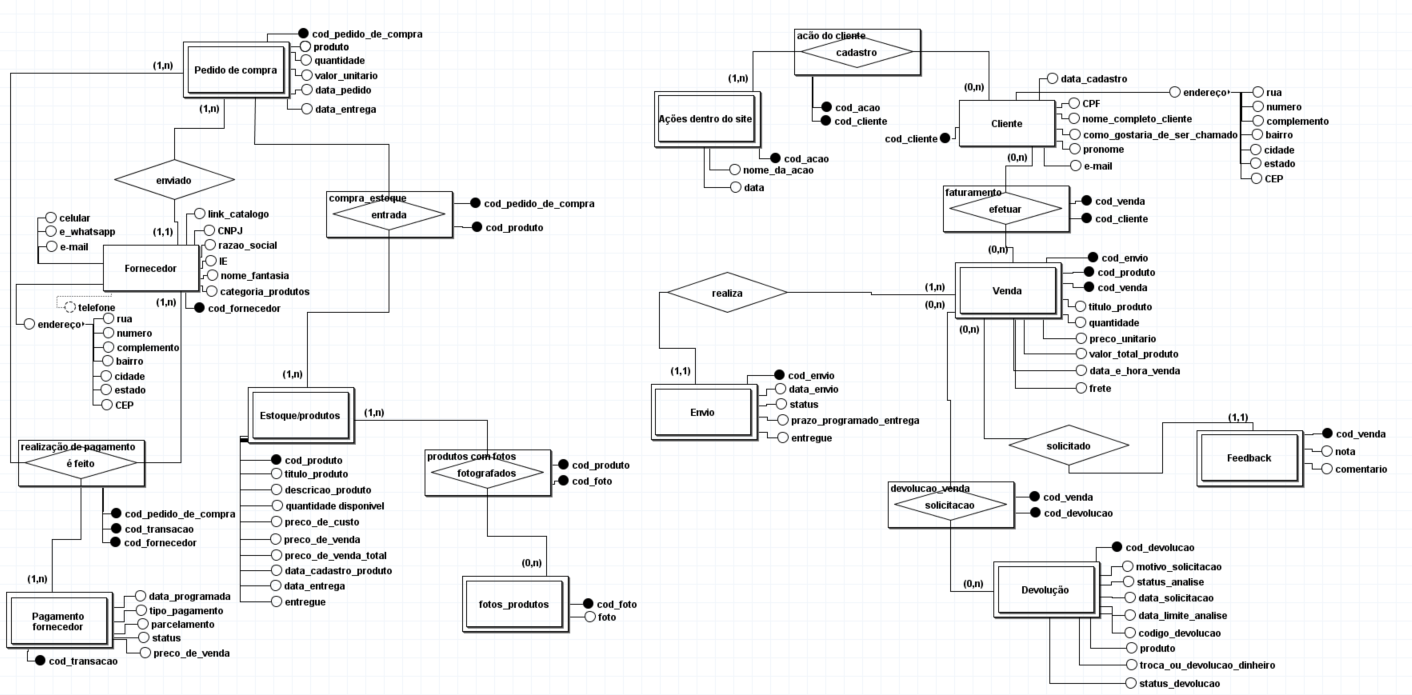
Esta visão descreve a estrutura geral de implementação, a decomposição do software em camadas de implementação.

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.

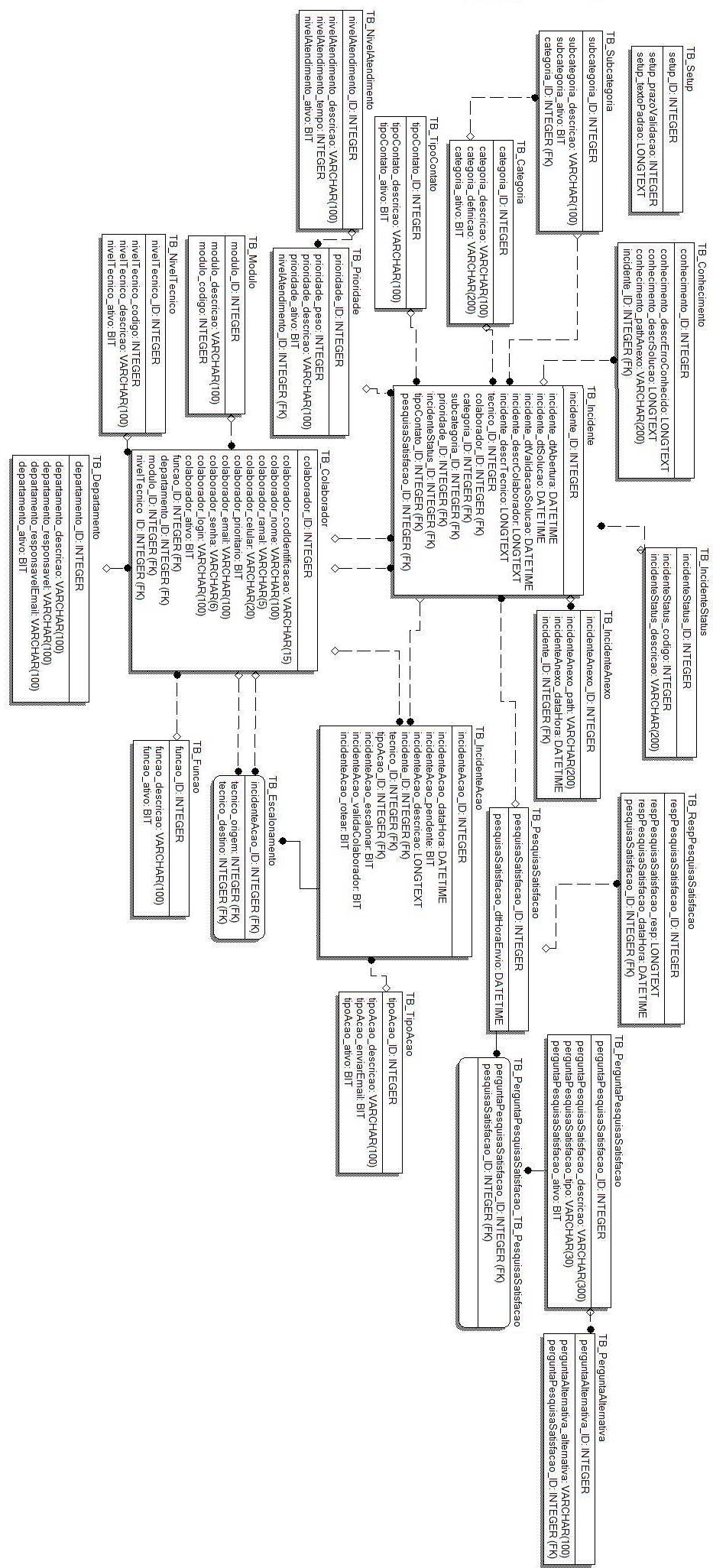
1. Visão de Dados

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 será utilizado para gerenciar o e-commerce, consequentemente mantendo uma grande base de dados.  
Seus servidores provavelmente enfrentarão períodos de pico de utilização, por exemplo, durante campanhas promocionais, lançamentos de produtos ou datas comemorativas de alto movimento, quando a carga de acessos e transações aumenta significativamente.

As estimativas sobre o número de usuários e a carga esperada durante esses períodos de pico, bem como informações detalhadas sobre o desempenho e a escalabilidade do sistema, podem ser consultadas no documento de requisitos não funcionais.

1. Qualidade

O sistema será utilizado para gerenciar o e-commerce, lidando com grandes volumes de transações financeiras e um elevado número de operações de compras e vendas diariamente. Erros ou falhas em sua operação podem causar prejuízos significativos, tanto financeiros quanto à reputação da empresa, portanto, na fase de design, a **confiabilidade** e a **robustez** do sistema devem ser prioridades absolutas. Além disso, como o sistema estará disponível na Internet, ele pode se tornar alvo de ataques cibernéticos, visando roubo ou corrupção de informações. Para minimizar esses riscos, é fundamental que uma infraestrutura de segurança seja cuidadosamente especificada e implementada. Informações detalhadas sobre requisitos de qualidade e segurança do sistema podem ser consultadas 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.