



Documentação e Escopo

ChainBuy

Plataforma E-commerce com Blockchain

Apresentação dos requisitos, histórias de usuário e arquitetura do sistema

Grupo 4

Professor:

Gilmar Ferreira Arantes

Alunos:

Abraão Santiago Moreira - 202201679

Felipe Moreira Silva - 202201689

Frederico Garcez Rodrigues - 202201690

Joseppe Pedro Cunha Fellini - 202300194

Mauro Sérgio do Nascimento Junior - 202204842

Introdução



Segurança e Transparência

Plataforma de e-commerce que utiliza blockchain para garantir a integridade e rastreabilidade das transações.



Arquitetura Moderna

Sistema projetado para oferecer experiência de compra segura com recursos avançados de rastreabilidade e pagamentos em criptomoedas.

Processo de Definição de Requisitos



Análise e transformação de requisitos iniciais em requisitos funcionais estruturados e histórias de usuário

Processo de Definição de Requisitos



Importância da Separação

- ✓ **Regras de negócio:** Políticas e diretrizes que governam a operação
- ✓ **Requisitos funcionais:** Comportamentos específicos do sistema
- ✓ **Atributos de qualidade:** Características não-funcionais

A separação clara permite melhor rastreabilidade, priorização e testabilidade dos requisitos.



Processo de Derivação

1. Identificação das regras de negócio fundamentais
2. Análise de impacto no comportamento do sistema
3. Tradução em comportamentos específicos
4. Especificação de interfaces e interações
5. Validação com stakeholders

Este processo iterativo garante alinhamento entre necessidades de negócio e implementação técnica.

Implicações Não-Funcionais



Identificação de Atributos

- ✓ **Segurança:** Proteção contra ameaças, autenticação, confidencialidade
- ✓ **Desempenho:** Tempo de resposta, throughput, utilização de recursos
- ✓ **Confiabilidade:** Disponibilidade, tolerância a falhas, recuperação
- ✓ **Usabilidade:** Facilidade de uso, eficiência, satisfação do usuário



Importância na Arquitetura

Os requisitos não-funcionais são frequentemente implícitos e precisam ser identificados proativamente, pois impactam diretamente:

- Escolhas de tecnologia e infraestrutura
- Decisões de arquitetura e design
- Estratégias de teste e validação
- Experiência final do usuário

Histórias de Usuário



Estrutura Eficaz

Formato padrão:

*Como [papel/persona],
Quero [funcionalidade/ação],
Para [benefício/valor de negócio].*

As histórias de usuário são eficazes porque:

- ✓ Mantêm o foco no valor para o usuário
- ✓ São concisas e compreensíveis
- ✓ Facilitam a priorização



CrITÉRIOS de Aceitação

CrITÉRIOS de aceitação eficazes devem ser:

- ✓ **Específicos e testáveis:** Verificáveis objetivamente
- ✓ **Abrangentes:** Cobrem cenários positivos e negativos
- ✓ **Independentes:** Não dependem da implementação
- ✓ **Automatizáveis:** Verificáveis por testes automatizados

Exemplo Prático: Desmembramento

"O pagamento fica retido no contrato inteligente até que o produto seja entregue."



Regra de Negócio

Política de liberação de pagamento: O pagamento só deve ser liberado ao vendedor após a confirmação da entrega do produto ao comprador.

Esta regra define uma política organizacional independente da implementação tecnológica.



Requisito Funcional

- ✓ O sistema deve reter o valor do pagamento em estado intermediário
- ✓ O sistema deve verificar o status de entrega antes de liberar
- ✓ O sistema deve liberar automaticamente o pagamento após confirmação



Atributo de Qualidade

- ✓ **Segurança:** Uso de contrato inteligente para garantir que o pagamento não possa ser liberado sem a confirmação adequada
- ✓ **Confiabilidade:** Garantia de processamento correto mesmo em caso de falhas



Implicações Não-Funcionais

- ✓ **Desempenho:** Verificação em tempo real, liberação em menos de 5 minutos
- ✓ **Transparência:** Visibilidade do status para todas as partes

Exemplo Prático: História de Usuário

"O pagamento fica retido no contrato inteligente até que o produto seja entregue."



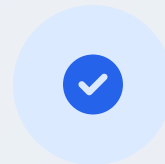
História de Usuário Resultante

Como comprador,
Quero que o pagamento seja liberado somente após a confirmação da entrega,
Para garantir que só pagarei quando receber o produto.

Critérios de Aceitação:

- ✓ O valor da compra deve ser bloqueado no contrato inteligente no momento da confirmação do pedido
- ✓ O comprador deve poder visualizar o status do pagamento como "Em espera" até a confirmação da entrega
- ✓ O pagamento deve ser liberado automaticamente para o vendedor em até 5 minutos após a confirmação de recebimento
- ✓ Se o comprador não confirmar em 7 dias, o sistema deve iniciar um processo de resolução de disputas

Requisitos e Histórias de Usuário



Detalhamento dos requisitos e histórias de usuário para a plataforma de e-commerce com blockchain

HU-01



Requisito 1: Avaliações em Blockchain

Avaliações e comentários armazenados em blockchain para evitar manipulação.

Implementação na primeira iteração do projeto.



HU-01: Avaliações em Blockchain

Como administrador da plataforma

Quero que avaliações sejam armazenadas em blockchain

Para garantir a integridade dos dados

Critérios de Aceitação:

- ✓ Registro automático na blockchain
- ✓ Sem impacto no tempo de resposta
- ✓ Mecanismo de retry para falhas

HU-02



Requisito 2: Registro de Transações

A plataforma deve registrar as transações para garantir a rastreabilidade, autenticidade e segurança.

Implementação na primeira iteração do projeto.



HU-02: Registro de Transações

Como administrador da plataforma

Quero que todas as transações sejam registradas automaticamente

Para garantir rastreabilidade, autenticidade e segurança

Critérios de Aceitação:

- ✓ Registro automático de todas transações relevantes
- ✓ Painel de consulta para verificação de status
- ✓ Mecanismo de retry para falhas

HU-03



Requisito 3: Autenticação de Vendedores

O sistema deve oferecer um mecanismo de autenticação para garantir que os vendedores sejam legítimos e reduzir o risco de fraude.

Implementação na segunda iteração do projeto.



HU-03: Autenticação de Vendedores

Como administrador da plataforma

Quero um mecanismo de autenticação para vendedores

Para garantir legitimidade e reduzir fraudes

Critérios de Aceitação:

- ✓ Verificação de identidade dos vendedores
- ✓ Bloqueio e notificação em caso de suspeita
- ✓ Autenticação em dois fatores (2FA)

HU-04



Requisito 4: Rastreamento de Entrega

A plataforma deve permitir que os usuários acompanhem o status da entrega de seus pedidos em tempo real, com atualizações armazenadas diretamente na blockchain.

Implementação na primeira iteração do projeto.



HU-04: Histórico de Entrega

Como cliente do e-commerce

Quero visualizar o status de entrega em tempo real

Para acompanhar o processo logístico

Critérios de Aceitação:

- ✓ Histórico completo de atualizações em ordem cronológica
- ✓ Registro de cada atualização na blockchain
- ✓ Destaque do status atual na interface

HU-05



Requisito 5: Pagamento com Criptomoedas

O usuário deve poder cadastrar sua carteira digital na plataforma e utilizar criptomoedas para realizar pagamentos.

Implementação na segunda iteração do projeto.



HU-05: Pagamento com Criptomoedas

Como cliente do e-commerce

Quero utilizar criptomoedas para pagamentos

Para ter uma alternativa segura e descentralizada

Critérios de Aceitação:

- ✓ Cadastro de carteira digital no perfil
- ✓ Opção de pagamento com criptomoeda no checkout
- ✓ Suporte inicial para Bitcoin e Dogecoin

HU-06



Requisito 6: Controle de Estoque

Bloqueio de vendas para produtos sem estoque disponível, evitando a comercialização de itens indisponíveis.

Implementação na primeira iteração do projeto.



HU-06: Estoque Esgotado

Como vendedor do e-commerce

Quero que produtos sem estoque saiam de comercialização

Para evitar vender produtos indisponíveis

Critérios de Aceitação:

- ✓ Status "indisponível" quando estoque chega a zero
- ✓ Notificação ao vendedor sobre esgotamento
- ✓ Remoção automática dos carrinhos de compra

HU-07



Requisito 7: Histórico de Transações

Visualização do histórico imutável de transações de pedidos registrados na blockchain, garantindo transparência e confiabilidade.

Implementação na segunda iteração do projeto.



HU-07: Histórico Imutável de Pedido

Como comprador do e-commerce

Quero visualizar o histórico imutável de transações

Para ter transparência e confiança no processo

Critérios de Aceitação:

- ✓ Linha do tempo com eventos do pedido
- ✓ Eventos vinculados a registros na blockchain
- ✓ Informações de data/hora e hash da transação

HU-08



Requisito 8: Especificações de Produtos

Exibição detalhada das especificações dos produtos aos compradores, permitindo decisões de compra informadas.

Implementação na primeira iteração do projeto.



HU-08: Especificações do Produto

Como comprador da plataforma

Quero visualizar especificações completas do produto

Para tomar decisões de compra informadas

Critérios de Aceitação:

- ✓ Exibição de informações detalhadas do produto
- ✓ Especificações sempre atualizadas
- ✓ Layout responsivo e acessível

HU-09



Requisito 9: Atualização Automática de Estoque

Atualização automática do estoque conforme as compras são realizadas, garantindo informações precisas sobre disponibilidade.

Implementação na primeira iteração do projeto.



HU-09: Atualização de Estoque

Como cliente do sistema

Quero que o estoque seja atualizado automaticamente

Para evitar venda de produtos indisponíveis

Critérios de Aceitação:

- ✓ Redução do estoque após finalização da compra
- ✓ Bloqueio de compras com estoque insuficiente
- ✓ Atualização imediata na interface

HU-10



Requisito 10: Armazenamento Seguro de Cartões

Armazenamento seguro de informações de cartões bancários para compras futuras, com criptografia forte para dados sensíveis.

Implementação na segunda iteração do projeto.



HU-10: Salvar Cartões

Como cliente do e-commerce

Quero salvar informações de cartões de forma segura

Para facilitar futuras compras

Critérios de Aceitação:

- ✓ Cadastro de múltiplos cartões
- ✓ Exibição apenas dos últimos 4 dígitos
- ✓ Criptografia forte para dados sensíveis

Diagrama de Classes



Estrutura do sistema: representação visual da arquitetura de classes e seus relacionamentos

Diagrama de Classes

