

Brainstorm de Requisitos - Sistema de Marketplace Digibuybr (Expandido)

Requisitos Funcionais

- - Cadastro de usuários com autenticação por e-mail, telefone ou redes sociais
- - Login e autenticação com autenticação de dois fatores (2FA) opcional
- - Cadastro e gerenciamento de produtos com controle de estoque e variações (cor, tamanho, etc.)
- - Busca inteligente com filtros por preço, categoria, avaliação e frete grátis
- - Adição de produtos ao carrinho de compras com atualização em tempo real
- - Processamento de pedidos com suporte a múltiplos vendedores no mesmo carrinho
- - Integração com múltiplos gateways de pagamento (ex: Stripe, PagSeguro, Inter, etc.)
- - Cálculo de frete com integração aos Correios, transportadoras e entrega própria
- - Rastreamento de pedidos por etapas (confirmado, enviado, a caminho, entregue)
- - Sistema de avaliações com fotos e comentários por parte do cliente
- - Sistema de denúncia de produtos irregulares
- - Painel de controle detalhado para vendedores (KPIs, gráficos, exportação de vendas)
- - Painel de controle para compradores com histórico de pedidos e preferências
- - Suporte a múltiplos idiomas e moedas (se internacionalização for objetivo futuro)
- - Notificações por e-mail, push e SMS sobre movimentações de pedidos
- - Funcionalidade de favoritar produtos e receber alertas de preço
- - Gestão de campanhas promocionais com regras (Ex: leve 3 pague 2, desconto progressivo)
- - Sistema de devolução automatizado com geração de etiquetas
- - Relatórios gerenciais para administradores do sistema

Requisitos Não Funcionais - (Complexidade e Alto Custo)

- - Alta disponibilidade: sistema deve funcionar 24/7, com downtime < 1% mensal
- - Escalabilidade horizontal para lidar com picos de acesso (ex: Black Friday)
- - Performance: tempo de resposta inferior a 2s para 95% das requisições
- - Segurança de dados sensíveis: criptografia em trânsito (HTTPS) e em repouso (ex: AES)
- - Armazenamento de dados em conformidade com LGPD
- - Compatibilidade com navegadores modernos e responsividade mobile
- - Log centralizado com alertas de falhas críticas e tentativas de acesso indevido
- - Documentação técnica completa e gerada automaticamente com Swagger/OpenAPI
- - Facilidade de manutenção: separação de responsabilidades no código e uso de padrões como MVC ou Clean Architecture
- - Deploy automatizado em ambiente de homologação para validação contínua
- - Teste automatizado com cobertura mínima de 80% para código de backend
- - Uso de containers (Docker) para padronização de ambiente entre devs

- - Ambiente de staging com dados fictícios realistas para testes de aceitação
- - Tempo médio de recuperação em falhas críticas inferior a 1 hora (MTTR)
- - Política de backup com redundância geográfica e restauração validada periodicamente