DESCRIÇÃO DO MVP – MARKETPEIXE

**1. VISÃO GERAL**

O MVP (Minimum Viable Product) do Marketpeixe é uma plataforma digital que conecta pescadores, indústrias e peixarias diretamente a consumidores finais e restaurantes, permitindo a compra e venda de pescado de forma rastreável, segura e eficiente.

O sistema foi desenhado para testar e validar o modelo de negócios em condições reais, com todas as funcionalidades essenciais para operação, monetização inicial e controle administrativo.

O Marketpeixe MVP entrega um marketplace verticalizado, centrado na autonomia do vendedor artesanal e na experiência simplificada do comprador, apoiado por infraestrutura tecnológica robusta e escalável.

**2. FUNCIONALIDADES PRINCIPAIS ENTREGUES NO MVP**

**2.1 Cadastro de Usuários**

Cadastro de Vendedores (Pescadores, Indústrias e Peixarias) com upload de documentação e validação automática de CNPJ.

Cadastro de Compradores (Consumidores e Restaurantes) de maneira simplificada.

Aprovação Manual de Vendedores para casos de documentação incompleta ou em regularização.

**2.2 Cadastro e Anúncio de Produtos**

Anúncio de produtos frescos e subprodutos (filés, postas, camarões processados).

Upload de fotos reais ou seleção de imagens de catálogo.

Definição obrigatória de forma de entrega (entrega própria ou retirada).

**2.3 Sistema de Compra e Checkout**

Compra de produtos baseada em unidade, com peso médio vinculado.

Tela de checkout integrada ao fluxo de pedido, agendamento de entrega/retirada.

Escolha entre pagamento via PIX ou Cartão de Crédito.

**2.4 Sistema de Pagamentos e Segurança**

Implementação de pagamento escrow: valor pago é retido até confirmação de entrega pelo comprador.

Split automático do pagamento: 85% para o vendedor e 15% de comissão para o marketplace.

**2.5 Comunicação e Atendimento**

Chat interno comprador ↔ vendedor funcional e seguro.

Mensagens automáticas de status do pedido (confirmação, pagamento, envio, entrega).

**2.6 Avaliações e Reputação**

Sistema de avaliação pós-compra (estrelas e comentários).

Monitoramento automático de vendedores mal avaliados e sistema de alertas para administração.

Penalização e bloqueio de vendedores com recorrência de problemas.

**2.7 Painel Administrativo**

Painel de gestão de usuários, produtos, pedidos e avaliações.

Controle manual de aprovação/suspensão/bloqueio de vendedores.

Dashboard inicial de KPIs (vendas realizadas, produtos mais vendidos, cadastros semanais).

**2.8 Funcionalidades Premium (Monetização do Marketplace)**

Sistema de compra de créditos para destaque de anúncios.

Aplicação de créditos para prioridade de exibição nos resultados de busca.

Sistema de emissão de cupons promocionais por vendedores.

**3. FUNCIONALIDADES FUTURAS INICIADAS NO MVP**

**3.1 Rastreabilidade Blockchain**

Implantação de rastreabilidade para produtos premium, registrando capturas, processamentos e vendas em blockchain permissionada.

Geração automática de QR Codes para consulta pública de rastreabilidade.

**3.2 Integração com Operadores Logísticos**

Implementação inicial de contratação de transportadoras parceiras para entrega dos produtos.

**3.3 Programa de Benefícios e Fidelidade**

Implantação de cashback e distribuição automática de cupons de desconto para compradores recorrentes.

**4. ARQUITETURA TÉCNICA**

Frontend: Sondar stacks mobile direcionadas. React.js ou Vue.js, mobile-first, componentes.

Backend: Node.js (Express) ou Python (FastAPI), APIs RESTful, autenticação via JWT.

Banco de Dados: PostgreSQL com modelagem otimizada, criptografia de dados sensíveis. Vessel, Firebase como opção

Armazenamento de Imagens: VPS Hostinger, AWS S3 ou serviço compatível.

Pagamentos: Integração com Gateway (ex.: Pagar.me, Stripe) com suporte a split e escrow.

Mensageria: WebSocket Server para chat em tempo real.

Monitoramento: Logs estruturados, alertas básicos de uptime.

**5. Estratégia de Validação**

O MVP do Marketpeixe permitirá validar:

| **Aspecto** | **Como o MVP testa** |
| --- | --- |
| Adesão dos vendedores | Facilita cadastro e permite testar escala de entrada de pescadores e peixarias. |
| Aderência dos compradores | Sistema simples de compra com checkout fluido. |
| Fluxo financeiro | Pagamento integrado, split de receita e segurança via escrow. |
| Comunicação e Confiança | Chat interno e rastreabilidade de transações reforçam a confiança. |
| Monetização da plataforma | Destaques pagos, cupons e venda de créditos ativam receitas logo no início. |
| Aceitação da rastreabilidade | Produtos premium rastreados via Blockchain testam diferenciação de mercado. |

**BACKLOG RESUMIDO - MARKETPEIXE**

**1. Cadastro e Perfis de Usuários**

[P0] Cadastro de vendedores (Pescadores, Indústrias, Peixarias) com upload de CNPJ e documentos.

[P0] Cadastro de compradores (Consumidores finais e Restaurantes).

[P1] Validação automática de CNPJ para vendedores.

[P1] Aprovação manual de vendedores sem validação automática.

**2. Anúncio de Produtos**

[P0] Sistema de criação de anúncios manuais.

[P0] Upload de fotos (opção de foto real ou foto de catálogo).

[P0] Campos obrigatórios: Espécie, Peso por unidade, Quantidade, Data de captura.

[P1] Cadastro de subprodutos (filé, posta, camarão, etc.) com campos adicionais (tipo, validade, descrição).

**3. Sistema de Compra**

[P0] Sistema de compra direta pelo app (Checkout).

[P0] Vendas por unidades (kg vinculado à unidade no cadastro).

[P1] Opção de negociação para compras grandes (unidade mínima para negociação no cadastro do produto).

**4. Pagamento**

[P0] Integração com PIX e Cartão de Crédito.

[P0] Sistema de pagamento escrow (liberação após entrega confirmada).

[P0] Split de pagamento automático: 85% vendedor / 15% Marketplace.

**5. Entrega e Logística**

[P0] Cadastro de forma de entrega por produto (Entrega disponível ou Retirada no local).

[P0] Agendamento de entrega/retirada (data e horário combinados).

[P2] (Futuro) Integração com parceiro logístico.

**6. Comunicação**

[P0] Chat interno comprador ↔ vendedor.

[P0] Mensagens automáticas de status do pedido (Confirmado, Em Produção, Enviado, Entregue).

**7. Avaliação e Reputação**

[P0] Sistema de avaliação com estrelas e comentários (para vendedores e compradores).

[P1] Sistema de alerta de qualidade: vendedores mal avaliados são sinalizados.

[P1] Penalização/banimento de vendedores ruins após análise do Admin.

**8. Administração da Plataforma**

[P0] Painel de administração (Gestão de usuários, anúncios e pedidos).

[P1] Dashboard de métricas básicas: vendas por período, ticket médio, produtos mais vendidos, usuários ativos.

**9. Funcionalidades Premium**

[P1] Sistema de recarga de créditos para destaque de anúncios.

[P1] Diferentes níveis de destaque de produto (mais créditos = maior prioridade nas listas).

[P2] Emissão automática de cupons de desconto pelos vendedores.

**10. Funcionalidades Futuras**

[P2] Integração de rastreabilidade Blockchain + QR Code no cadastro dos produtos premium.

[P2] Parceria com operadores logísticos externos (integração API).

[P2] Programa de benefícios para usuários frequentes (Cashback, Cupons Exclusivos).

**PRIORIZAÇÃO DOS ITENS (LEGENDA)**

[P0] Essencial para o MVP (tem que estar no lançamento)

[P1] Muito importante, mas pode entrar no próximo ciclo (1-3 meses após MVP)

[P2] Funcionalidades avançadas (6 meses após MVP ou depois de escalar)

**RESUMO VISUAL DO FLUXO DE USUÁRIOS (SIMPLIFICADO)**

Cadastro > Validação > Publicar Produto > Comprar Produto > Pagamento Seguro > Entrega > Avaliação

BACKLOG DETALHADO - MARKETPEIXE

1. CADASTRO E PERFIS DE USUÁRIOS

**[P0] 1.1 Cadastro de Vendedores (Pescadores, Indústrias, Peixarias) com Upload de CNPJ e Documentos**

**Descrição**: Implementar um fluxo de cadastro específico para vendedores (pescadores, indústrias de pescado e peixarias), solicitando obrigatoriamente os seguintes dados: razão social, nome de contato, CNPJ ativo, endereço completo, telefone, e-mail e upload de documentos comprobatórios (comprovante de CNPJ, certificação sanitária, quando aplicável). O sistema deverá validar a integridade dos dados fornecidos, armazená-los em banco de dados seguro e manter a documentação disponível para conferência administrativa. Esse fluxo visa garantir que apenas vendedores regularizados operem na plataforma, fortalecendo a confiabilidade do marketplace.

**Indicação Técnica**:

1. Backend: Node.js ou Python (Django/FastAPI) com banco de dados PostgreSQL.
2. Frontend: Formulário responsivo em React.js ou Vue.js.
3. Upload de arquivos: Armazenamento seguro em AWS S3 ou serviço equivalente, com controle de acesso.
4. Implementar criptografia de documentos e compliance com LGPD para dados sensíveis.

**Previsão de Tempo**: 5 dias úteis.

**[P0] 1.2 Cadastro de Compradores (Consumidores Finais e Restaurantes)**

**Descrição**: Desenvolver um processo de cadastro simplificado para compradores, sejam consumidores finais ou restaurantes, exigindo apenas CPF ou CNPJ, nome completo ou razão social, telefone, e-mail e endereço de entrega. A proposta é garantir rapidez na entrada de novos usuários, diminuindo fricções no processo de registro e incentivando a adesão à plataforma.

**Indicação Técnica**:

1. Backend: Utilizar as mesmas bases de autenticação do cadastro de vendedores.
2. Frontend: Formulário simples, validado com máscaras de input (CPF/CNPJ, telefone).
3. Banco de dados: Normalizar registros de usuários por tipo (comprador/vendedor).
4. Opção futura: permitir login social via Google ou Facebook para compradores.

**Previsão de Tempo**: 3 dias úteis.

**[P0] 1.3 Validação Automática de CNPJ para Vendedores**

**Descrição:** Integrar sistema de verificação automática de CNPJ durante o cadastro de vendedores. O sistema deverá consultar a base pública da Receita Federal ou um serviço equivalente (como ReceitaWS) para validar se o CNPJ está ativo, confirmar razão social e verificar eventuais irregularidades. Caso o CNPJ esteja inativo ou irregular, o cadastro deverá ser automaticamente bloqueado até regularização.

**Indicação Técnica:**

1. Consumo de API pública de validação de CNPJ (ex: ReceitaWS).
2. Backend: Implementar chamadas automáticas via token seguro e tratar respostas assíncronas.
3. Frontend: Mostrar status do CNPJ em tempo real ao usuário durante o cadastro.

**Previsão de Tempo**: 4 dias úteis.

**[P1] 1.4 Aprovação Manual de Vendedores sem Validação Automática**

**Descrição**: Para casos em que a validação automática falhar ou não for aplicável (ex.: pequenos produtores em regularização), será necessário um painel interno para análise manual de documentos enviados. O administrador poderá revisar, aprovar, solicitar complementações ou recusar o cadastro. A comunicação do status será automática para o vendedor via e-mail ou notificação no sistema.

**Indicação Técnica:**

1. Backend: Implementar painel administrativo seguro (acesso restrito).
2. Frontend: Interface de visualização de documentos, botões de ação (Aprovar/Recusar/Pedir Complemento).
3. Banco de dados: Estados de cadastro "Pendente", "Aprovado", "Recusado", "Complementar".
4. Integração com envio automático de e-mails via SMTP autenticado ou serviço (ex: SendGrid).

**Previsão de Tempo:** 5 dias úteis.

2. CADASTRO E ANÚNCIO DE PRODUTOS

**[P0] 2.1 Sistema de Criação de Anúncios Manuais**

**Descrição:** Permitir que vendedores (pescadores, indústrias e peixarias) criem anúncios de forma manual e intuitiva, preenchendo um formulário detalhado. Os campos obrigatórios deverão incluir: tipo de produto (peixe inteiro, filé, posta, subproduto), espécie do pescado, forma de apresentação (fresco, congelado), peso por unidade, quantidade disponível, e data da captura ou processamento. O sistema deve validar todos os campos e garantir que nenhum anúncio seja publicado com informações incompletas.

**Indicação Técnica:**

1. Frontend: Formulário responsivo em React.js ou Vue.js, com validação de campos obrigatórios e máscaras adequadas.
2. Backend: APIs de criação e atualização de produtos; banco de dados relacional PostgreSQL para gestão de produtos.
3. Implementar upload parcial (salvar rascunho) e finalização posterior.
4. Garantir compatibilidade mobile-first.

**Previsão de Tempo**: 4 dias úteis.

**[P0] 2.2 Upload de Fotos (Foto Real ou Foto de Catálogo)**

**Descrição:** Habilitar o envio de fotos reais do produto diretamente pelo vendedor ou a seleção de imagens padronizadas de um catálogo oficial de espécies disponibilizado pela plataforma. O sistema deverá permitir múltiplas fotos por anúncio e definir uma imagem principal. As imagens devem ser otimizadas automaticamente para web, sem perda significativa de qualidade.

**Indicação Técnica:**

1. Sistema de upload com compressão automática (ex: TinyPNG API ou biblioteca própria).
2. Armazenamento seguro em AWS S3 ou sistema equivalente com controle de acesso.
3. Banco de dados para indexação das imagens com os produtos.
4. Implementar marca d'água automática para proteção das imagens, se desejado.

**Previsão de Tempo**: 3 dias úteis.

**[P0] 2.3 Cadastro de Campos Obrigatórios (Espécie, Peso, Quantidade, Data de Captura)**

**Descrição:** Definir no cadastro de produtos campos obrigatórios: espécie do pescado, peso unitário ou peso médio, quantidade disponível e data de captura (para pescado in natura) ou data de processamento (para subprodutos). A ausência de qualquer um desses dados deve impedir a publicação do anúncio.

**Indicação Técnica:**

1. Frontend: Validação de formulário em tempo real.
2. Backend: Verificação de obrigatoriedade no endpoint de criação de produto, com retorno de erro amigável em caso de ausência.
3. Banco de dados: Definir constraints para garantir integridade.

**Previsão de Tempo:** 2 dias úteis.

**[P1] 2.4 Cadastro de Subprodutos (Filé, Posta, Camarão Processado, etc.)**

**Descrição:** Permitir o cadastro específico de subprodutos derivados do pescado, como filés, postas, camarões descascados ou qualquer outro item processado. Para subprodutos, além dos campos padrões (espécie, peso, quantidade, data de captura/processamento), devem ser adicionados: descrição do tipo de processamento, data de validade e lote de fabricação, quando aplicável.

**Indicação Técnica:**

1. Frontend: Formulário adaptativo baseado no tipo de produto (pescado inteiro ou subproduto).
2. Backend: Modelagem de dados separada para produtos primários e subprodutos, mantendo rastreabilidade clara.
3. Banco de dados: Inclusão de campos de data de validade e tipo de processamento.

**Previsão de Tempo:** 4 dias úteis.

3. SISTEMA DE COMPRA

**[P0] 3.1 Sistema de Compra Direta pelo App (Checkout)**

**Descrição:** Implementar um sistema de compra direta dentro da plataforma, permitindo que o comprador finalize o pedido integralmente sem sair do app ou site. O fluxo deverá incluir: seleção de produto, escolha da quantidade, resumo do pedido com frete ou retirada, escolha da forma de pagamento (PIX ou Cartão de Crédito) e confirmação final. Após a confirmação, o pedido deve ser registrado e gerenciado pelo sistema até a conclusão da entrega ou retirada.

**Indicação Técnica:**

1. Frontend: Implementar fluxo de checkout em React.js ou Vue.js, responsivo para mobile e desktop.
2. Backend: Serviços para criação de pedido, cálculo de valor total (produto + entrega se aplicável), associação com status inicial "Aguardando Pagamento".
3. Integração com sistema de pagamentos descrito no Item 4.
4. Banco de dados: Tabelas específicas para pedidos, itens de pedido e histórico de status.

**Previsão de Tempo**: 6 dias úteis.

**[P0] 3.2 Vendas por Unidades (kg vinculado)**

**Descrição:** Estruturar a venda de produtos por unidade de venda (exemplo: caixa, peça, quilo) com a informação de peso vinculada à unidade para efeito de transparência e controle de estoque. O vendedor deverá informar, no cadastro do produto, o peso médio relacionado a cada unidade.

Indicação Técnica:

1. Backend: Definir modelo de produto com campos "unidade de venda" e "peso médio por unidade".
2. Frontend: Exibir na página do produto a correspondência entre unidade e peso.
3. Banco de dados: Atualizar entidade Produto para contemplar esses novos atributos.

**Previsão de Tempo**: 3 dias úteis.

**[P1] 3.3 Opção de Negociação para Grandes Compras**

**Descrição:** Oferecer uma funcionalidade onde o vendedor poderá configurar, no cadastro de produto, uma quantidade mínima para abertura de negociação personalizada. Quando um comprador desejar adquirir uma quantidade acima do mínimo configurado, poderá abrir uma negociação direta para tratar preço, prazo ou condições especiais.

**Indicação Técnica:**

1. Frontend: Botão "Negociar" visível ao atingir quantidade mínima configurada.
2. Backend: Serviços de abertura de "solicitação de negociação" vinculado ao produto e ao pedido.
3. Integração direta com o sistema de mensagens internas (chat) entre comprador e vendedor.

**Previsão de Tempo**: 4 dias úteis.

**4. PAGAMENTO**

**[P0] 4.1 Integração com PIX e Cartão de Crédito**

**Descrição:** Integrar meios de pagamento via PIX e Cartão de Crédito diretamente na plataforma, permitindo que o comprador escolha a opção desejada no checkout. O pagamento deverá ser processado de forma segura, com confirmação automática. O sistema deve gerar QR Code dinâmico para pagamentos via PIX e capturar pagamentos via cartão em ambiente seguro, obedecendo às normas PCI-DSS.

**Indicação Técnica:**

1. Integração com gateway de pagamento confiável (ex.: Pagar.me, MercadoPago, Stripe).
2. Backend: Implementar APIs para criação de cobranças e recepção de confirmações (webhooks).
3. Frontend: Fluxo de pagamento seguro, com redirecionamento ou modal integrado.
4. Proteção de dados sensíveis conforme PCI Compliance.

**Previsão de Tempo**: 7 dias úteis.

**[P0] 4.2 Sistema de Pagamento Escrow (Liberação após Entrega Confirmada)**

**Descrição:** Implementar um sistema de pagamento em modalidade escrow: o valor pago pelo comprador fica retido até que a entrega seja confirmada. Após a confirmação manual do recebimento pelo comprador, o valor é liberado automaticamente ao vendedor, descontando a comissão do marketplace.

**Indicação Técnica:**

1. Backend: Controle de status financeiro dos pedidos ("Pagamento Retido", "Entrega Confirmada", "Pagamento Liberado").
2. Integração com gateway que suporte split + escrow (Pagar.me, Stripe Connect, etc.).
3. Frontend: Opção para comprador confirmar entrega no painel de pedidos.

**Previsão de Tempo**: 5 dias úteis.

**[P0] 4.3 Split de Pagamento Automático (85% Vendedor / 15% Marketplace)**

**Descrição:** No momento da liberação do pagamento, o sistema deverá automaticamente dividir o valor: 85% para o vendedor e 15% para o marketplace (comissão). Essa divisão precisa ocorrer no ato da liquidação, garantindo repasse automático e transparente, sem necessidade de intervenção manual.

**Indicação Técnica:**

1. Utilizar funcionalidade nativa de split em gateways modernos (Pagar.me, Stripe, etc.).
2. Backend: Lógica de definição de percentuais variáveis no ato da criação do pedido.
3. Relatórios de repasse e extratos de pagamento para vendedores no painel.

**Previsão de Tempo**: 4 dias úteis.

5. ENTREGA E LOGÍSTICA

**[P0] 5.1 Cadastro de Forma de Entrega por Produto (Entrega Disponível ou Retirada no Local)**

**Descrição:** Permitir que o vendedor, no momento do cadastro do produto, escolha a modalidade de entrega:

Entrega Própria (o vendedor entrega diretamente ao comprador) ou

Retirada Local (o comprador deve buscar o produto no endereço informado pelo vendedor).

Essa informação deve ser exibida de forma clara na página de cada produto e ser capturada no pedido no momento do checkout.

**Indicação Técnica:**

1. Frontend: Campo obrigatório no formulário de cadastro do produto (radio button: “Entrega” ou “Retirada”).
2. Backend: Atualizar endpoint de criação/edição de produto (POST /api/products) para aceitar novo campo: delivery\_type ("delivery" ou "pickup").
3. Banco de dados: Adicionar campo delivery\_type na tabela de produtos (VARCHAR 10).
4. Interface do Usuário: Mostrar no resumo do pedido a modalidade selecionada.
5. Validação: Impedir criação de produto sem definição do tipo de entrega.

**Previsão de Tempo**: 3 dias úteis.

**[P0] 5.2 Agendamento de Entrega/Retirada (Data e Horário Combinados)**

**Descrição:** Após a confirmação do pedido, o comprador deverá escolher a data e o horário desejados para a retirada ou entrega do produto (conforme disponibilidade informada pelo vendedor). O sistema deve registrar essa escolha vinculada ao pedido.

**Indicação Técnica Específica:**

FrontEnd

1. Após o checkout, abrir modal ou etapa adicional para agendamento.
2. Componente de calendário com seleção de datas (lib: react-datepicker ou equivalente).
3. Validação de escolha mínima: 24 horas após o pedido (para produção/entrega).

Backend:

1. Endpoint PATCH /api/orders/{id}/schedule para salvar delivery\_date e delivery\_time.
2. Banco de dados: Criar colunas delivery\_date (DATE) e delivery\_time (TIME) na tabela de pedidos.
3. Notificações:Enviar confirmação de agendamento por e-mail ou push notification.
4. Administração: Painel administrativo com visualização dos agendamentos futuros por vendedor.

**Previsão de Tempo**: 4 dias úteis.

**[P0] 6.2 Mensagens Automáticas de Status do Pedido**

**Descrição:** Automatizar o envio de mensagens de sistema ao comprador e ao vendedor a cada atualização importante do pedido, como: pedido confirmado, pagamento aprovado, pedido enviado, pedido entregue.

**Indicação Técnica:**

Backend:

1. Implementar eventos assíncronos (pub/sub) para disparo de mensagens no backend em Node.js (exemplo: biblioteca bull com Redis para filas de eventos).
2. Exemplo de evento: Quando o pedido mudar de status → gerar mensagem automática.
3. APIs: Criar serviço interno NotificationService para centralizar geração de mensagens.

Frontend: Mostrar mensagens automáticas no mesmo painel do chat, mas diferenciadas visualmente (ex.: cinza ou texto system message).

1. Mensagens principais:
2. "Seu pedido foi confirmado!"
3. "Seu pagamento foi aprovado!"
4. "O vendedor despachou seu pedido."
5. "Pedido entregue! Avalie seu vendedor."

Banco de dados

Registrar mensagens automáticas na mesma tabela de chat\_messages, com flag is\_system\_message = true.

**Previsão de Tempo**: 3 dias úteis.

6. COMUNICAÇÃO

**[P0] 6.1 Chat Interno Comprador ↔ Vendedor**

**Descrição:** Desenvolver um sistema de comunicação interna entre comprador e vendedor, permitindo o envio de mensagens antes e depois da conclusão do pedido. As mensagens precisam ser armazenadas de forma segura, com histórico vinculado ao pedido ou à negociação.

**Indicação Técnica Específica:**

Arquitetura:

1. Utilizar WebSocket para mensagens em tempo real (exemplo: Socket.IO para Node.js).
2. Estruturar o chat em “salas” individuais por pedido ou por negociação.

Frontend:

1. Criar componente de chat por pedido (ChatBox), carregando histórico automaticamente.
2. Exibir bolhas de mensagem com timestamps.
3. Biblioteca sugerida para WebSocket cliente: socket.io-client (React.js ou Vue.js).

Backend:

1. Criar servidor de WebSocket separado ou integrado via gateway de eventos.
2. APIs REST auxiliares para recuperação de histórico (GET /api/chats/{orderId}) e envio offline (POST /api/chats/send).
3. Banco de dados: Tabela chat\_messages com campos id, sender\_id, receiver\_id, order\_id, message, created\_at.

Segurança:

1. Autenticação no WebSocket usando token JWT.
2. Validação de permissão: comprador e vendedor só podem acessar chats de pedidos dos quais participam.

**Previsão de Tempo**: 6 dias úteis.

**[P0] 6.2 Mensagens Automáticas de Status do Pedido**

**Descrição:** Automatizar o envio de mensagens de sistema ao comprador e ao vendedor a cada atualização importante do pedido, como: pedido confirmado, pagamento aprovado, pedido enviado, pedido entregue.

**Indicação Técnica Específica:**

Backend:

1. Implementar eventos assíncronos (pub/sub) para disparo de mensagens no backend em Node.js (exemplo: biblioteca bull com Redis para filas de eventos).

Ex. de evento: Quando o pedido mudar status → gerar mensagem automática.

1. APIs: Criar serviço interno NotificationService para centralizar geração de mensagens.

Frontend:

1. Mostrar mensagens automáticas no mesmo painel do chat, mas diferenciadas visualmente (ex.: cinza ou texto system message).

Mensagens principais:

"Seu pedido foi confirmado!"

"Seu pagamento foi aprovado!"

"O vendedor despachou seu pedido."

"Pedido entregue! Avalie seu vendedor."

Banco de dados:

1. Registrar mensagens automáticas na mesma tabela de chat\_messages, com flag is\_system\_message = true.

**Previsão de Tempo**: 3 dias úteis.

7. AVALIAÇÃO E REPUTAÇÃO

**[P0] 7.1 Sistema de Avaliação com Estrelas e Comentários**

**Descrição**: Após a finalização de um pedido (entrega confirmada), o comprador poderá avaliar o vendedor e o produto adquirido. A avaliação será feita com uma nota de 1 a 5 estrelas e um comentário opcional. Essa avaliação será vinculada tanto ao pedido quanto ao perfil público do vendedor.

**Indicação Técnica**:

Frontend:

1. Componente de avaliação visual (exemplo: biblioteca react-star-ratings ou vue-star-rating).
2. Formulário simples pós-entrega no histórico de pedidos ("Avalie sua experiência").

Backend:

1. Endpoint POST /api/reviews para envio de avaliação, incluindo: order\_id, rating (1–5), comment (texto livre), reviewer\_id, reviewee\_id.
2. Regras de integridade: só pode avaliar quem efetivamente comprou e recebeu o pedido.

Banco de dados:

1. Tabela reviews com campos: id, order\_id, reviewer\_id, reviewee\_id, rating, comment, created\_at.

Exibição:

1. Mostrar média de avaliações no perfil dos vendedores (exemplo: “4.7 ★ com base em 45 avaliações”).
2. Mostrar avaliações recentes no perfil do vendedor.

**Previsão de Tempo: 4** dias úteis.

**[P1] 7.2 Sistema de Alerta de Qualidade (Vendedores Mal Avaliados Sinalizados)**

**Descrição**: Implementar um mecanismo que monitore automaticamente a média de avaliação dos vendedores. Caso a média fique abaixo de um limiar definido (ex.: 3.0 estrelas), o sistema deverá sinalizar esse vendedor para revisão manual da administração.

**Indicação Técnica Específica:**

Backend:

1. Serviço de monitoramento periódico (cron job diário) para calcular a média de avaliação de cada vendedor.
2. Caso a média esteja abaixo do limite por mais de 3 avaliações válidas, criar uma entrada de alerta em uma tabela quality\_alerts.

Painel Administrativo:

1. Criar seção específica para “Vendedores em Alerta” no Admin Panel.
2. Permitir ao admin visualizar avaliações recentes e decidir ações (notificação, suspensão, bloqueio).

Notificação Interna:

1. Sistema envia aviso automático ao admin quando novos alertas forem gerados.

**Previsão de Tempo**: 3 dias úteis.

**[P1] 7.3 Penalização/Banimento de Vendedores Após Análise Administrativa**

**Descrição:** Após análise manual do administrador sobre vendedores mal avaliados ou com condutas inadequadas, será possível realizar o bloqueio, suspensão temporária ou remoção definitiva do vendedor da plataforma.

**Indicação Técnica Específica:**

Frontend Administrativo:

1. No painel de gestão de usuários, adicionar botões de ação: “Suspender”, “Bloquear”, “Excluir”.
2. Exigir justificativa obrigatória ao aplicar penalização.

Backend:

1. Atualizar status do usuário na base (status = “active”, “suspended”, “banned”).
2. Endpoint PATCH /api/users/{id}/status para administradores.

Comunicação:

1. Sistema envia automaticamente e-mail ao vendedor penalizado com explicação e prazo, se aplicável.

Segurança:

1. Vendedores banidos não devem conseguir fazer login ou recuperar acesso.

**Previsão de Tempo:** 3 dias úteis.

8. ADMINISTRAÇÃO DA PLATAFORMA

**[P0] 8.1 Painel de Administração (Gestão de Usuários, Anúncios e Pedidos)**

**Descrição:**

Desenvolver um painel administrativo completo para gerenciar usuários (compradores e vendedores), produtos/anúncios ativos, pendentes ou bloqueados, e pedidos em andamento. O painel deve permitir a execução de ações administrativas, como aprovação de vendedores, edição de anúncios, análise de transações, bloqueio de contas e acompanhamento de pedidos.

**Indicação Técnica Específica:**

Frontend (Admin Panel):

1. Construído em React.js ou Vue.js.
2. Organização em módulos: “Usuários”, “Anúncios”, “Pedidos”, “Alertas de Qualidade”.
3. Tabelas dinâmicas usando bibliotecas como react-table ou vue-good-table.
4. Filtros inteligentes por status, datas, nome, e-mail, tipo de usuário.

Backend:

1. Endpoints RESTful separados para cada tipo de recurso:

GET /api/admin/users

GET /api/admin/products

GET /api/admin/orders

PATCH /api/admin/users/{id}/status (suspender/reativar)

PATCH /api/admin/products/{id}/status (publicar/bloquear)

1. Controle de permissão e autenticação via token JWT com role admin.
2. Banco de Dados: Adicionar timestamps para ações administrativas (admin\_action\_log).
3. Segurança: Acesso apenas a administradores autenticados.
4. Logs de auditoria para todas as alterações realizadas no sistema administrativo.

**Previsão de Tempo**: 7 dias úteis.

**[P1] 8.2 Dashboard de Métricas Básicas (Vendas por Período, Ticket Médio, Produtos Mais Vendidos, Usuários Ativos)**

**Descrição:** Criar um dashboard visual no painel administrativo consolidando os principais indicadores de desempenho da plataforma, como volume de vendas no período, ticket médio, produtos mais vendidos, evolução de novos cadastros e usuários ativos semanais.

**Indicação Técnica Específica:**

Frontend:

1. Componentes de gráficos usando bibliotecas como recharts, chart.js ou apexcharts.
2. Layout responsivo com cards resumidos e gráficos detalhados.

Backend:

1. Endpoints de agregação de dados para dashboard:

GET /api/admin/metrics/sales (vendas totais por período)

GET /api/admin/metrics/top-products (produtos mais vendidos)

GET /api/admin/metrics/users-growth (novos usuários/mês)

GET /api/admin/metrics/average-ticket (ticket médio).

1. Query otimizada para grandes volumes de dados (uso de índices e consultas agregadas).

Banco de Dados:

1. Criação de tabelas de apoio para cache de métricas diárias, se necessário.

**Previsão de Tempo**: 5 dias úteis.

9. FUNCIONALIDADES PREMIUM

**[P1] 9.1 Sistema de Recarga de Créditos para Destaque de Anúncios**

**Descrição:** Implementar um sistema interno de créditos onde vendedores possam adquirir pacotes (ex.: 10, 50, 100 créditos) para aplicar em seus anúncios. Quanto mais créditos um vendedor investir no anúncio, maior será sua visibilidade no marketplace, dentro de lógicas específicas de ordenação.

**Indicação Técnica Específica:**

Frontend:

1. Página exclusiva no painel do vendedor: “Comprar Créditos”.
2. Botões de compra de pacotes (ex.: 10 créditos = R$ 20, 50 créditos = R$ 80).
3. Integração direta com sistema de pagamento já existente (PIX, cartão).

Backend:

1. Nova tabela credits\_transactions com os campos: id, user\_id, amount\_purchased, amount\_used, created\_at.

Endpoints:

1. POST /api/credits/purchase (compra de créditos)
2. PATCH /api/credits/use (uso de crédito para destacar anúncio)

Banco de Dados:

1. Campo adicional highlight\_credits em produtos para controle de prioridade.

Regra de Segurança:

1. Vendedor só poderá usar créditos disponíveis. Tentativas de uso acima do saldo serão bloqueadas.

**Previsão de Tempo**: 5 dias úteis.

**[P1] 9.2 Diferentes Níveis de Destaque de Produto (Mais Créditos = Maior Prioridade)**

**Descrição:** Anúncios serão ordenados nas listas do marketplace não apenas por data de postagem, mas também pela quantidade de créditos investidos para destaque. Produtos com mais créditos aparecerão em posições superiores nas buscas e listas.

**Indicação Técnica Específica:**

BeckEnd:

1. Atualizar endpoint de listagem de produtos (GET /api/products) para incluir ordenação customizada: primeiro por créditos, depois por data.
2. Fórmula de ordenação sugerida: ORDER BY highlight\_credits DESC, created\_at DESC.

Frontend:

1. Indicar visualmente que um produto é “Anúncio Destaque” (ex.: badge dourado).
2. Ajustar resultados de busca e categorias para refletir a nova lógica.

Banco de Dados:

1. Indexação no campo highlight\_credits para garantir performance em listas grandes.

**Previsão de Tempo**: 4 dias úteis.

**[P2] 9.3 Emissão de Cupons de Desconto pelos Vendedores**

**Descrição:** Permitir que vendedores criem cupons promocionais para seus produtos, oferecendo descontos para compradores (ex.: “10% OFF para os 10 primeiros clientes”). Cada cupom terá regras de validade, limite de uso e tipo de desconto (por valor fixo ou percentual).

**Indicação Técnica Específica:**

Frontend:

1. Painel para vendedores gerenciarem cupons: criar, editar, ativar/desativar.
2. Formulário para configuração: nome do cupom, tipo (valor fixo ou %), valor, validade, limite de usos.

Backend:

1. Nova tabela discount\_coupons com os campos: id, seller\_id, code, discount\_type, discount\_value, max\_usage, valid\_until, used\_count.

Endpoints:

1. POST /api/coupons (criação)
2. PATCH /api/coupons/{id} (edição)
3. GET /api/coupons/validate/{code} (validação no checkout).

Aplicação no Checkout:

1. Validação automática no momento do pagamento.
2. Aplicação do desconto no valor total antes de confirmar o pedido.

**Previsão de Tempo**: 4 dias úteis.

10. FUNCIONALIDADES FUTURAS

**[P2] 10.1 Integração de Rastreabilidade Blockchain + QR Code**

**Descrição:** Implementar um sistema de rastreabilidade para produtos premium. Cada lote de pescado terá suas informações (captura, processamento, transporte) registradas em Blockchain para garantir autenticidade e transparência. Ao gerar um anúncio, será criado um QR Code vinculado a essa cadeia de dados, permitindo ao comprador consultar toda a rastreabilidade apenas escaneando o código.

**Indicação Técnica Específica:**

Arquitetura:

1. Uso de Blockchain permissionada (ex.: Hyperledger Fabric, Amazon Managed Blockchain) ou pública leve (Polygon, para reduzir custo de transação).
2. Cada etapa (captura, processamento, venda) cria um hash transacional.

Frontend:

1. No cadastro de produto premium: formulário extra para inserção de dados de rastreabilidade.
2. Geração automática de QR Code (lib: qrcode.react ou vue-qrcode).
3. QR Code exibido na página do produto.

Backend:

1. Serviço de integração blockchain: API que assina dados, registra hash e retorna ID de rastreio.

Banco de dados:

1. manter espelho local da rastreabilidade para consultas rápidas (blockchain\_reference\_id, batch\_id, capture\_data, process\_data, sale\_data).

Segurança:

1. Criptografia de dados sensíveis antes de envio ao blockchain.

**Previsão de Tempo**: 10 dias úteis.

**[P2] 10.2 Integração com Operadores Logísticos (Entrega Terceirizada)**

**Descrição:** Oferecer a vendedores a opção de contratar entrega de pedidos diretamente com transportadoras parceiras integradas à plataforma. O sistema deve permitir cotação automática de frete, contratação e rastreio básico dos envios.

**Indicação Técnica Específica:**

Frontend:

1. Opção no checkout: “Desejo contratar entrega via parceiro logístico”.
2. Exibir tabela de cotações (ex.: Jadlog, Correios, Transportadora Local).

Backend:

1. Integração via API REST/SOAP com operadores logísticos (dependendo do parceiro):

Exemplo: Correios SIGEP Web, Jadlog API.

Endpoints para:

1. Cotação automática (POST /api/logistics/quote)
2. Geração de etiqueta (POST /api/logistics/label)
3. Rastreio de pedido (GET /api/logistics/track/{tracking\_code}).

Banco de dados:

1. Tabelas auxiliares logistics\_orders, logistics\_tracking.

Painel Administrativo:

1. Monitoramento de todos os envios em andamento.

**Previsão de Tempo**: 8 dias úteis.

**[P2] 10.3 Programa de Benefícios para Usuários Frequentes (Cashback e Cupons Exclusivos)**

Descrição: Criar um programa de fidelidade para premiar usuários recorrentes, com cashback sobre compras e distribuição automática de cupons exclusivos para compras futuras. O objetivo é aumentar a taxa de recompra e fidelização de clientes.

**Indicação Técnica Específica:**

Frontend:

1. Seção "Minhas Recompensas" no painel do comprador.
2. Exibição do saldo de cashback acumulado e lista de cupons disponíveis.

Backend:

1. Lógica de acumulação: ex.: “5% de cashback em cada compra confirmada”.
2. Banco de dados: Tabelas cashback\_balances e exclusive\_coupons.

Endpoints:

1. GET /api/rewards/balance (saldo atual)
2. POST /api/rewards/redeem (uso de cashback/cupom).

Aplicação no Checkout:

1. Permitir que cashback acumulado seja usado como desconto direto.

**Previsão de Tempo**: 6 dias úteis.