

Especificação de Requisitos e Casos de Uso: RubinMarket

1. Visão Geral

Plataforma de comércio de itens (Marketplace), focada na interação entre jogadores com notificações externas via Bot do Discord. O sistema visa facilitar a negociação segura de itens in-game, com ferramentas de moderação e conformidade.

Atores

- **Visitante:** Usuário não autenticado (Apenas leitura). Pode buscar e visualizar itens.
- **Player (Usuário Logado):** Usuário autenticado via E-mail/Senha. Pode anunciar, iniciar negociações, enviar mensagens e gerenciar seu perfil.
- **Sistema:** Processos automáticos (Cron Jobs, validações, expiração de anúncios).
- **Bot de Notificação:** Serviço (Sidecar) que consome filas para enviar mensagens diretas (DMs) no Discord.
- **Moderador:** Administrador com poderes de auditoria, banimento e exclusão de conteúdo.

2. Resumo dos Casos de Uso

Grupo A: Acesso e Perfil

- **UC01:** Autenticar-se (Login/Cadastro via E-mail e Senha)
- **UC02:** Gerenciar Perfil (Configuração de Discord Manual)

Grupo B: Mercado (Core)

- **UC03:** Consultar/Buscar Itens
- **UC04:** Criar Anúncio (Dataset Typeahead + Seleção de Moeda)
- **UC05:** Gerenciar Meus Anúncios (Edição, Exclusão e Expiração Automática)

Grupo C: Interação e Notificação

- **UC06:** Iniciar Negociação (Chat Interno com Alerta RMT)
- **UC07:** Receber Notificações (Fila -> Discord)
- **UC08:** Denúncia (Anúncio ou Usuário com Log)

Grupo D: Administração

- **UC09:** Moderar Conteúdo (Auditoria e Logs Criptografados)

3. Detalhamento dos Casos de Uso

UC01: Autenticar-se

- **Autor:** Visitante
- **Fluxo:** Login tradicional via E-mail e Senha (armazenada com hash seguro).

UC02: Gerenciar Perfil (Configuração de Notificações)

- **Autor:** Player
- **Descrição:** Vínculo manual para notificações.
- **Fluxo Principal:**
 1. Player acessa configurações.
 2. Player insere seu Nome de Usuário do Discord.
 3. Sistema exibe alerta sobre a necessidade de permitir DMs no servidor oficial.
 4. Player salva.

UC03: Consultar/Buscar Itens

- **Autor:** Visitante, Player
- **Fluxo:** Busca com filtros, ordenação e visualização de detalhes.

UC04: Criar Anúncio

- **Autor:** Player
- **Fluxo Principal:**
 1. Player clica em "Anunciar".
 2. **Seleção de Item:** Campo com Typeahead (autocompletar) conectado ao Dataset oficial. O usuário seleciona um ID existente.
 3. **Atributos Dinâmicos:** Sistema carrega campos específicos (Refino, Slots, Atributos Mágicos) baseados no item.
 4. **Definição de Preço e Moeda:**
 - O Player insere o valor numérico.
 - O Player seleciona a moeda em uma lista fixa: **Gold Coins** ou **Rubin Coins**.
 5. Player publica o anúncio.

UC05: Gerenciar Meus Anúncios

- **Autor:** Player, Sistema
- **Fluxo A: Ações do Player:**
 1. **Editar:** Player altera preço ou descrição (Item não pode ser alterado).
 2. **Excluir:** Player remove o anúncio permanentemente (Soft Delete).
 3. **Marcar como Vendido:** Remove da lista pública, mantendo histórico pessoal.
- **Fluxo B: Expiração Automática (Sistema/Cron Job):**
 1. Rotina diária verifica anúncios sem atualização/interação.

2. **Regra de 15 dias:** Envia alerta sugerindo renovação.
3. **Regra de 30 dias:** O Sistema arquiva o anúncio automaticamente (status: Arquivado).
4. O anúncio deixa de aparecer nas buscas públicas.

UC06: Iniciar Negociação (Chat Interno)

- **Autor:** Player (Comprador)
- **Pré-condição:** O Player **não** pode ser o dono do anúncio.
- **Fluxo Principal:**
 1. Comprador clica em "Negociar" no anúncio.
 2. Sistema abre uma sala de chat dedicada entre Comprador e Vendedor.
 3. **Alerta de Segurança:** O chat exibe uma mensagem fixa e destacada no topo:
"ATENÇÃO: RMT (Real Money Trading) é estritamente proibido. Negocie apenas usando moedas do jogo. Transações em dinheiro real resultarão em banimento permanente."
 4. Players trocam mensagens para combinar local de encontro in-game.

UC07: Receber Notificações

- **Autor:** Sistema, Bot
- **Gatilho:** Nova mensagem no chat (UC06) ou Venda.
- **Fluxo Técnico:** Backend publica evento na Fila -> Bot consome -> Envia DM no Discord se usuário localizado -> Fallback para notificação no site.

UC08: Denúncia

- **Autor:** Player
- **Fluxo A: Denunciar Anúncio:**
 1. Player visualiza um anúncio suspeito.
 2. Seleciona opção "Denunciar Anúncio".
 3. Escolhe motivo: *Spam, Golpe, Preço Irreal ou Ofensa*.
 4. Envia para moderação.
- **Fluxo B: Denunciar Usuário (Via Chat):**
 1. Dentro da negociação (UC06), Player identifica comportamento tóxico ou tentativa de RMT.
 2. Clica em "Denunciar Usuário".
 3. Sistema captura um snapshot dos últimos registros da conversa.
 4. Sistema anexa o **log criptografado** à denúncia.
 5. Player confirma o envio.

UC09: Moderar Conteúdo

- **Autor:** Moderador
- **Fluxo Principal:**
 1. Moderador acessa o Painel Administrativo (/admin/denuncias).
 2. Visualiza fila de reports (Anúncios e Usuários).
 3. **Auditória:**
 - Se denúncia de Anúncio: Vê os dados do item.
 - Se denúncia de Chat: Sistema descriptografa e exibe o Log de Chat associado para que o moderador verifique o contexto (provas de RMT ou ofensa).
 4. **Ações:** Descartar, Remover Anúncio ou Banir Usuário.

4. Requisitos Não-Funcionais (RNF) - Grupo V2.0

4.1 Confiabilidade e Integração

- **RNF-01 (Resiliência API Discord):** O sistema deve operar suas funções críticas (Busca, Criação de Anúncios e Chat Interno) normalmente, mesmo que a API do Discord esteja indisponível ou instável. As notificações devem ser enfileiradas para tentativa posterior.
- **RNF-02 (Rate Limiting):** O serviço de notificações deve implementar filas inteligentes (Throttling) para respeitar estritamente os limites de requisição (Rate Limits) da API do Discord, evitando bloqueio do Bot.

4.2 Usabilidade e Interface (UX)

- **RNF-03 (Mobile First):** As interfaces de Chat e Busca devem ser totalmente responsivas, garantindo usabilidade plena em dispositivos móveis.
- **RNF-04 (UX Dataset):** O componente de busca e criação de anúncios (Typeahead) deve responder às entradas do usuário em menos de 100ms, utilizando cache local ou indexação otimizada.

4.3 Performance

- **RNF-05 (Tempo de Carregamento):** O feed principal de anúncios deve carregar (Time to Interactive) em menos de 2 segundos em conexões 4G padrão.
- **RNF-06 (Latência do Chat):** O sistema de chat deve utilizar WebSockets (ou tecnologia similar) para garantir a entrega de mensagens em tempo quase real (latência < 1s).

4.4 Manutenibilidade

- **RNF-07 (Gestão de Dataset):** A arquitetura deve permitir a atualização do catálogo de itens (IDs, Nomes, Imagens) através de upload de JSON ou script externo, sem a necessidade de reiniciar a aplicação ou derrubar o serviço (Zero Downtime)

Deployment para dados estáticos).

4.5 Segurança e Compliance

- **RNF-08 (Compliance RMT):** O sistema não deve possuir, em nenhuma interface, campos numéricos formatados para moeda real (BRL, USD, EUR). Todos os campos de valor devem referir-se exclusivamente às moedas do jogo (Gold/Rubin).
- **RNF-09 (Auditoria de Chat):** Os logs de conversa devem ser armazenados de forma criptografada no banco de dados. A chave de descriptografia deve ser acessível apenas pelo módulo de Moderação mediante uma denúncia ativa. Retenção dos logs: 90 dias.
 - 1.
 - 1.