ADR: Arquitetura do Auth Service

📌 **Data**: 22/02/2025
📌 **Status**: Aprovado
📌 **Autor**: Clever Santoro Lopes

---

1. Contexto

O **Auth Service** é responsável por autenticação e autorização de usuários no e-commerce. Ele deve garantir:

✅ **Autenticação segura** (tokens JWT ou OAuth 2.0).
✅ **Autorização baseada em papéis (RBAC)** para controle de acesso.
✅ **Integração com provedores externos** (Google, Facebook, Apple).
✅ **Alta disponibilidade** e **resiliência contra ataques** (ex: força bruta).

---

2. Decisão

Optamos por um **serviço de autenticação centralizado** utilizando **OAuth 2.0 + JWT**, garantindo escalabilidade e segurança.

📌 Tecnologias Escolhidas

Componente	Tecnologia	Justificativa
Linguagem	Node.js (NestJS) ou GoLang	Performance e suporte a APIs
Banco de Dados	PostgreSQL	Armazena usuários e permissões
Cache	Redis	Armazena tokens para revogação rápida
Autenticação	JWT + OAuth 2.0	Segurança e suporte a terceiros
Mensageria	Kafka / RabbitMQ	Eventos de login/logout
Monitoramento	Prometheus + Grafana	Logs e métricas de segurança

---

📌 Arquitetura do Auth Service

1️⃣ **API Gateway** → Direciona requisições para o Auth Service.
2️⃣ **Auth Service** → Gera tokens JWT, valida credenciais e gerencia permissões.
3️⃣ **Banco de Dados** → Armazena informações de usuários e sessões.
4️⃣ **Integração com OAuth** → Permite login com Google, Facebook, Apple.
5️⃣ **Monitoramento e Segurança** → Protege contra ataques e acessos suspeitos.

---

3. Alternativas Consideradas

3.1 Autenticação com Sessões

🔴 **Rejeitado** – Depende de sticky sessions e não escala bem em múltiplas instâncias.

3.2 OpenID Connect com Keycloak/Auth0

🟡 **Parcialmente Considerado** – Boa opção para reduzir complexidade, mas pode gerar custos adicionais.

3.3 OAuth 2.0 com JWT

✅ **Aprovado** – Permite autenticação distribuída, segura e escalável.

---

4. Consequências

✅ Benefícios

✔ **Autenticação segura** com tokens assinados.
✔ **Escalabilidade** sem depender de sessões no servidor.
✔ **Integração com provedores externos** (SSO).
✔ **Monitoramento e rastreabilidade** de logins e acessos.

⚠️ Desafios

⚠ **Revogação de tokens** → Necessário cache distribuído (Redis) para blacklisting.
⚠ **Gerenciamento de OAuth** → Pode aumentar a complexidade inicial da implementação.

---

5. Próximos Passos

🚀 Implementação de **MFA (autenticação multifator)** para maior segurança.
🔍 **Monitoramento** com OpenTelemetry para rastrear tentativas de login.
📈 **Testes de carga e segurança** para validar resistência contra ataques.

🎯 **Conclusão**
O **Auth Service** fornecerá autenticação segura, escalável e integrada com provedores externos, garantindo a segurança do e-commerce.