

FACULDADE: PUC GOIAS EAD

CURSO: Análise e Desenvolvimento de Software

DISCIPLINA: Projeto Integrador III-A

PROFESSOR(A): José Ricardo Cosme Lérias Ribeiro

## DOCUMENTAÇÃO TÉCNICA DE PROJETO DE SOFTWARE

### Migração de Arquitetura Monolítica para Microsserviços

Alunos: João Vitor Ferreira da Silva, Pedro Nunes Marques Junior, Victor Hugo Batista Pereira, Ariel Jorge da Silva, Leandro Batista de Sousa Galdido.

Cidade: GOIANIA

Data: 01/11/2025

#### 1. Objetivo Geral

Desenvolver uma documentação técnica que apresente a proposta de migração da aplicação monolítica da empresa TechStore para uma arquitetura distribuída baseada em microsserviços, com foco em escalabilidade, desempenho e segurança da informação.

##### 1.1 Objetivos Específicos

Analisar as limitações e riscos da atual arquitetura monolítica.

Propor uma nova arquitetura distribuída, definindo os microsserviços e suas responsabilidades.

Elaborar uma estratégia gradual e segura de migração.

Detalhar os mecanismos de segurança aplicados, assegurando Confidencialidade, Integridade e Disponibilidade (CID).

Apresentar diagramas e justificativas técnicas que validem as decisões de arquitetura.

#### 2. Desenvolvimento

##### 2.1 Situação Atual – Arquitetura Monolítica

O sistema atual da TechStore foi desenvolvido de forma monolítica, centralizando em um único código todas as funcionalidades do e-commerce: autenticação de usuários, gerenciamento de produtos, controle de pedidos e processamento de pagamentos. Essa estrutura apresenta as seguintes limitações:

Baixa escalabilidade: qualquer aumento de carga afeta todo o sistema.

Dificuldade de manutenção: pequenas alterações exigem o redeploy completo da aplicação.

Dependência tecnológica: não é possível usar linguagens ou frameworks diferentes em partes do sistema.

Riscos de disponibilidade: uma falha em um módulo pode derrubar todo o serviço.

## 2.2 Motivações para a Migração

Reduzir o tempo de implantação de novas features;

Aumentar a resiliência e disponibilidade do sistema;

Melhorar o desempenho e a escalabilidade horizontal;

Implementar autonomia entre equipes de desenvolvimento;

Garantir maior segurança e rastreabilidade dos dados e transações.

## 2.3 Identificação dos Módulos Principais

Autenticação e Usuários – controle de login, cadastro, perfis e tokens JWT.

Catálogo de Produtos – CRUD de produtos, categorias e estoque.

Pedidos – gerenciamento de carrinho, status e histórico de pedidos.

Pagamentos – integração com gateways (Pix, cartão, boleto).

Notificações – envio de e-mails e alertas sobre pedidos e promoções.

## 2.4 Proposta de Arquitetura em Microsserviços

Cada módulo será convertido em um microsserviço independente, comunicando-se via API REST (com JSON) e, em alguns casos, mensageria assíncrona (RabbitMQ/Kafka) para integração desacoplada.

## 2.5 Estratégia de Migração

Etapa 1: Criação de microsserviços Auth e Product em paralelo ao monólito.

Etapa 2: Redirecionamento das requisições via API Gateway (por exemplo, Spring Cloud Gateway).

Etapa 3: Migração dos módulos de Pedidos e Pagamentos.

Etapa 4: Desativação gradual das rotas monolíticas e monitoramento dos logs.

Etapa 5: Migração completa e containerização (Docker + Kubernetes).

## 2.6 Segurança da Informação (CID)

### Resumo

Este documento técnico apresenta o processo de migração de uma aplicação monolítica para uma arquitetura baseada em microsserviços, adotando boas práticas de escalabilidade, segurança e integração entre sistemas. A proposta visa demonstrar como a decomposição de um sistema tradicional pode aumentar a eficiência, a resiliência e a manutenção da aplicação. O estudo de caso considera uma empresa fictícia chamada

TechStore, que realiza vendas online e busca modernizar sua infraestrutura de software.

## 2.7 Conclusão

A migração da TechStore para uma arquitetura baseada em microsserviços trará ganhos significativos de desempenho, segurança e escalabilidade, permitindo evolução contínua do sistema e autonomia das equipes de desenvolvimento. O uso de práticas modernas como mensageria, autenticação distribuída e observabilidade garantem estabilidade e confiabilidade em ambiente corporativo.

## Referências Bibliográficas

Newman, S. (2015). Building Microservices. O'Reilly Media.

Richardson, C. (2018). Microservices Patterns. Manning Publications.

Fowler, M. (2019). Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith. O'Reilly Media.

ISO/IEC 27001:2013 – Information Security Management Systems Requirements.