**Documento de Arquitetura de Software - Sistema de Agendamento**

**1. Introdução**

Este documento descreve a arquitetura proposta para o Sistema de Agendamento, delineando as principais componentes e sua interação. A arquitetura é baseada em uma abordagem distribuída, semelhante a microserviços, utilizando um banco de dados central compartilhado.

**2. Visão Geral da Arquitetura**

A arquitetura do sistema é composta pelos seguintes componentes principais:

* **Frontend**: Aplicação web desenvolvida em ASP.NET MVC, responsável pela interface do usuário.
* **API Gateway**: Componente responsável pelo roteamento de solicitações do frontend para os microserviços apropriados e pela autenticação de usuários.
* **Microserviços**: Componentes independentes que representam as funcionalidades principais do sistema.
* **Banco de Dados**: Banco de dados central compartilhado para armazenamento de informações globais e bancos de dados individuais para cada microserviço.

**3. Componentes da Arquitetura**

**3.1 Frontend:**

* Desenvolvido em ASP.NET MVC.
* Responsável pela apresentação da interface do usuário e interação com o usuário final.
* Comunica-se com os microserviços por meio de chamadas de API RESTful.

**3.2 API Gateway:**

* Responsável pelo roteamento de solicitações do frontend para os microserviços apropriados.
* Realiza a autenticação e autorização de usuários.
* Pode implementar políticas de segurança, como controle de acesso e proteção contra ataques.

**3.3 Microserviços:**

* Cada funcionalidade principal do sistema é representada por um microserviço separado.
* Exemplos de microserviços incluem Autenticação, Agendamento, Cadastro de Serviços e Produtos, Controle Financeiro, entre outros.
* Cada microserviço tem sua própria lógica de negócios e banco de dados associado.

**3.4 Banco de Dados:**

* Um banco de dados central é compartilhado por todos os microserviços para armazenamento de informações globais, como dados de usuários e configurações do sistema.
* Cada microserviço possui seu próprio banco de dados separado para armazenamento de dados específicos da funcionalidade.

**4. Comunicação entre Componentes**

* O frontend se comunica com os microserviços por meio de chamadas de API RESTful.
* Os microserviços podem se comunicar entre si por meio de chamadas de API ou mensagens assíncronas, dependendo dos requisitos específicos de cada funcionalidade.

**5. Considerações de Segurança**

* Devem ser implementadas medidas de segurança em todos os níveis da arquitetura, incluindo autenticação de usuários, proteção contra ataques e controle de acesso.

**6. Considerações de Escalabilidade e Desempenho**

* A arquitetura distribuída permite escalabilidade horizontal dos microserviços, facilitando a adição ou remoção de instâncias conforme necessário para lidar com o aumento da carga.
* A performance do sistema deve ser monitorada de perto e otimizações devem ser realizadas conforme necessário.

**7. Considerações Finais**

A arquitetura proposta oferece uma abordagem flexível e modular para o desenvolvimento do Sistema de Agendamento, permitindo escalabilidade, manutenção e expansão contínuas. A implementação detalhada de cada componente deve levar em consideração os requisitos específicos do projeto e as melhores práticas de arquitetura de software.