**Guia de Desenvolvimento - API de Cadastro de Clientes**

**1. Introdução**

Este guia fornece diretrizes para a implementação da API de Cadastro de Clientes, com foco nas melhores práticas de codificação, padrões de design, e estrutura do projeto.

**2. Diretrizes de Codificação**

* **Padrão da API**: Manter os endpoints consistentes com as operações HTTP (GET, POST, PUT, DELETE).
* **Clean Code**: Seguir princípios de código limpo (funções com nomes descritivos, SRP - Princípio da Responsabilidade Única).
* **Tratamento de Exceções**: Implementar tratamento robusto de erros, garantindo que exceções sejam gerenciadas adequadamente e que mensagens de erro sejam claras.

**3. Padrões de Design**

* **Repository Pattern**: Usar classes de serviço para separar a lógica de negócios da camada de acesso a dados.
* **AutoMapper**: Usado para simplificar a conversão entre DTOs e entidades.
* **JWT para Autenticação**: Usar JSON Web Tokens para gerenciar sessões de usuários.

**4. Estrutura de Pastas**

* **Controllers**: Armazenar todos os controllers da API.
* **Services**: Serviços que contêm lógica de negócios.
* **DTOs**: Objetos de transferência de dados entre a API e as requisições.
* **Models**: Modelos de dados correspondentes às tabelas no banco de dados.

**5. Boas Práticas**

* **Validação de Dados**: Implementar validações no nível do modelo para garantir a integridade dos dados.
* **Limitação de Logotipo**: No GET, limitar o retorno do logotipo para evitar sobrecarga de tráfego de rede e garantir melhor visualização.

**6. Configuração de Banco de Dados**

* Utilizar índices em chaves estrangeiras para melhorar o desempenho de JOINs.
* **Script para criação do banco deverá ser feita através dos comandos SQL do arquivo “Scripts de Criação do banco”.**

**7. Ferramentas**

* **Swagger**: Usar Swagger para documentação e teste dos endpoints.
* **Postman**: Recomenda-se utilizar o Postman para testes de APIs durante o desenvolvimento.