Guia de Desenvolvimento - Projeto Kraken

Este documento foi preparado para orientar os desenvolvedores que implementarão a solução do projeto Kraken. Ele contém diretrizes de codificação, padrões de design a serem seguidos, e outras informações relevantes para garantir a consistência e a qualidade do código.

# Diretrizes Gerais de Codificação

1. \*\*Nomenclatura\*\*: Utilize nomes significativos para classes, métodos e variáveis, de acordo com as convenções C#.  
2. \*\*Organização de Código\*\*: Separe as responsabilidades de cada camada de maneira clara, evitando acoplamento excessivo.  
3. \*\*Tratamento de Exceções\*\*: Capture exceções de forma adequada e forneça mensagens de erro informativas.  
4. \*\*Comentários\*\*: Adicione comentários claros e concisos onde necessário, sem excessos.  
5. \*\*Padrão de formatação\*\*: Siga as regras de formatação padrão do .NET Core (uso de camelCase, PascalCase, etc.).

# Padrões de Design a Seguir

1. \*\*Repository Pattern\*\*: Para a camada de infraestrutura, garantindo que o acesso a dados seja abstraído da lógica de negócio.  
2. \*\*Unit of Work\*\*: Implementar o Unit of Work onde apropriado para gerenciar transações entre diferentes repositórios.  
3. \*\*Dependency Injection\*\*: Utilizar a injeção de dependências para promover a testabilidade e flexibilidade no código.  
4. \*\*DTO (Data Transfer Object)\*\*: Para transferir dados entre as camadas de forma eficiente, utilize objetos DTO.  
5. \*\*Factory Pattern\*\*: Utilize o padrão de fábrica para criar instâncias de objetos complexos.

# Estrutura do Projeto

O projeto está organizado nas seguintes camadas principais:  
1. \*\*Kraken.Api\*\*: Contém os controladores da API que interagem com os serviços da aplicação. Certifique-se de que as rotas sigam o padrão REST.  
2. \*\*Kraken.Aplicacao\*\*: Lida com a lógica de negócio e coordena as interações entre a API e a camada de domínio.  
3. \*\*Kraken.CrossCutting\*\*: Contém funcionalidades que são reutilizadas em várias partes do projeto, como logging, tratamento de exceções e injeção de dependências.  
4. \*\*Kraken.Dominio\*\*: Define as entidades de domínio. Certifique-se de que as entidades que refletem as tabelas estejam centralizadas aqui.  
5. \*\*Kraken.InfraEstrutura\*\*: Responsável pela comunicação com o banco de dados e outras fontes externas. Use o Repository Pattern para abstrair o acesso a dados.

# Informações Adicionais

1. \*\*Segurança\*\*: Siga boas práticas de segurança, como validação de entrada de dados e proteção contra vulnerabilidades como SQL Injection e XSS.  
   2. \*\*Logs\*\*: Implemente logs consistentes em toda a aplicação para facilitar a depuração e o monitoramento do sistema.  
   3. \*\*Manutenção\*\*: Sempre documente qualquer novo recurso ou modificação significativa no código.

# Telas do front-end



