

Desafio BC

Documentação dos Projetos  
*Documento visão, com a finalidade de instruir mediante o problema e a solução adotada.*

Irei abordar o problema proposto, sobre a criação do banco de dados, arquitetura e tecnologias utilizadas.

# Problema

A empresa possui o cadastro via APP, e nessa onda, habilitamos o cadastro via API para Onboard de nossos clientes.Tendo esse problema, você deverá criar uma API para o cadastro destes clientes.

# Arquitetura

Pensando em desacoplamento de responsabilidades, até mesmo uma possível falha na obtenção do endereço por um fornecedor externo, foi desenvolvido duas APIs, com responsabilidades únicas, conforme mostra a imagem a seguir, Api.RegisterCustomer e Api.PostalCode.

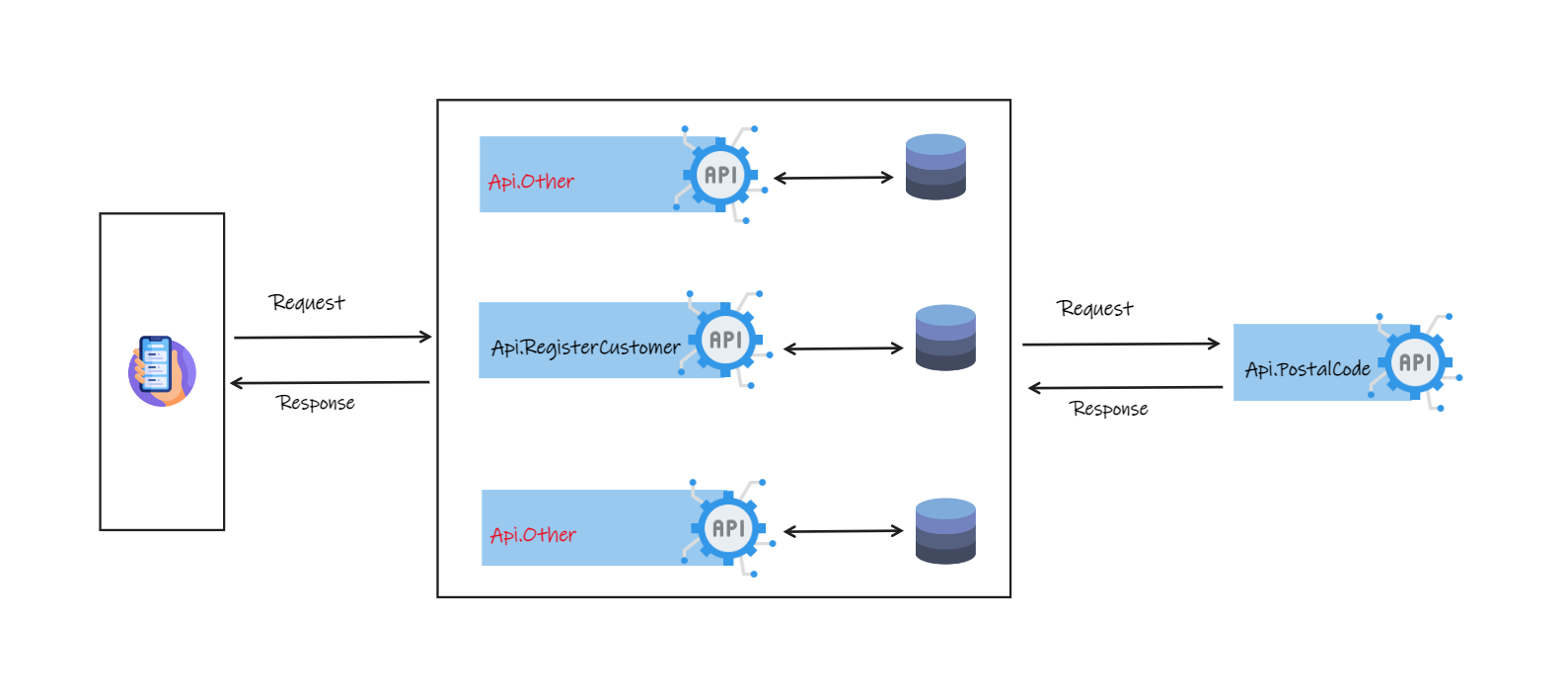


Figura 1: brainstorm de uma possível arquitetura.

Na figura 1, o cliente pode solicitar uma requisição para Api.RegisterCustomer, que por sua vez tem a responsabilidade de comunicar-se com Api.PostalCode para obter o endereço, em Seguida Api.RegisterCustomer realiza persistencia dos dados.

Além do mais, Api.Other foram colocadas somente como exemplo, que inclusive poderão utilizar-se da Api.PostalCode.

# Banco de Dados

Para realização da persistência dos dados, foi utilizado o Sql Server, sem a geração do migration pelo Visual Studio, dessa forma seguindo o conceito de database first. Entretanto, tomando os devidos cuidados para trazer as regras de negócio para responsabilida da aplicação. A seguir um dicionário de dados e os scripts para executar no Sql Server.

|  |  |
| --- | --- |
| **Tabela** | **Descrição** |
| TB\_CUSTUMER | Tabela para registrar o novo cliente cadastrado no sistema. |
| TB\_ADDRESS | Tabela para registrar o endereço mediante o cep informado. |
| TB\_STATUS\_ADDRESS | Tabela para registrar o status da TB\_ADDRESS. |
| VW\_FULLDATA\_CUSTUMER | View com os campos essenciais das 3 tabelas. |

Figura 2: Descrição das tabelas e view que foram criadas.

Na figura 2, mostro as tabelas que foram criadas como regra de negócio para aplicação. Inclusive foi criado uma view, com os campos essenciais das três tabelas, ganhando dessa forma ao realizar uma consulta.

|  |  |  |
| --- | --- | --- |
| **Tabela** | **Coluna** | **Descrição** |
| VW\_FULLDATA\_CUSTUMER | name | Nome do cliente |
|  | tax\_id | Cpf ou Cnpj validado na aplicação. |
|  | phone\_number | Telefone do cliente. |
|  | created\_at | Data de criação da conta. |
|  | status | Para o Cep informado vai ser atribuído “Approved” caso contrário “Pending” |
|  | state | Estado do cliente |
|  | city | Cidade do cliente |
|  | district | Bairro do cliente |
|  | address | Endereço do cliente |
|  | code | Cep do cliente |

Figura 3: Campos e descrições da view criada.

A view vai ser útil no momento em que precisar realizar consulta, desta forma ganhando um maior desempenho na aplicação.

Na figura 4, Mostro todos as colunas, tipo, tamanho e o significado de cada coluna para cada tabela.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Coluna** | **Tipo** | **Tamanho Máximo** | **Descrição** |
| TB\_CUSTUMER | id | uniqueidentifier | 16 | Guid gerado na aplicação. |
| TB\_CUSTUMER | name | varchar | 60 | Nome do cliente |
| TB\_CUSTUMER | tax\_id | varchar | 25 | Cpf ou Cnpj validado na aplicação. |
| TB\_CUSTUMER | password | varchar | 30 | Senha mínimo caracters de 8, 1 letra Maíscula, 1 Minúscula, Caracter Especial. Validado na aplicação |
| TB\_CUSTUMER | phone\_number | varchar | -1 | Telefone do cliente. |
| TB\_CUSTUMER | created\_at | datetime | 8 | Data de criação da conta. |
| TB\_STATUS\_ADDRESS | id | uniqueidentifier | 16 | Guid gerado na aplicação. |
| TB\_STATUS\_ADDRESS | status | varchar | 10 | Para o Cep informado vai ser atribuído “Approved” caso contrário “Pending” |
| TB\_ADDRESS | id | uniqueidentifier | 16 | Guid gerado na aplicação. |
| TB\_ADDRESS | id\_custumer | uniqueidentifier | 16 | Chave estrangeira que referência a TB\_CUSTUMER. |
| TB\_ADDRESS | id\_status\_address | uniqueidentifier | 16 | Chave estrangeira que referência a TB\_STATUS\_ADDRESS |
| TB\_ADDRESS | state | varchar | 4 | Estado do cliente |
| TB\_ADDRESS | city | varchar | 200 | Cidade do cliente |
| TB\_ADDRESS | district | varchar | 200 | Bairro do cliente |
| TB\_ADDRESS | address | varchar | 200 | Endereço do cliente |
| TB\_ADDRESS | code | varchar | 200 | Cep do cliente |

Figura : Tabelas, colunas, tipo, tamanho e descrições de cada campo.

# Scripts Sql

Segue passo a passo os scripts a serem executados para utilização na aplicação.