

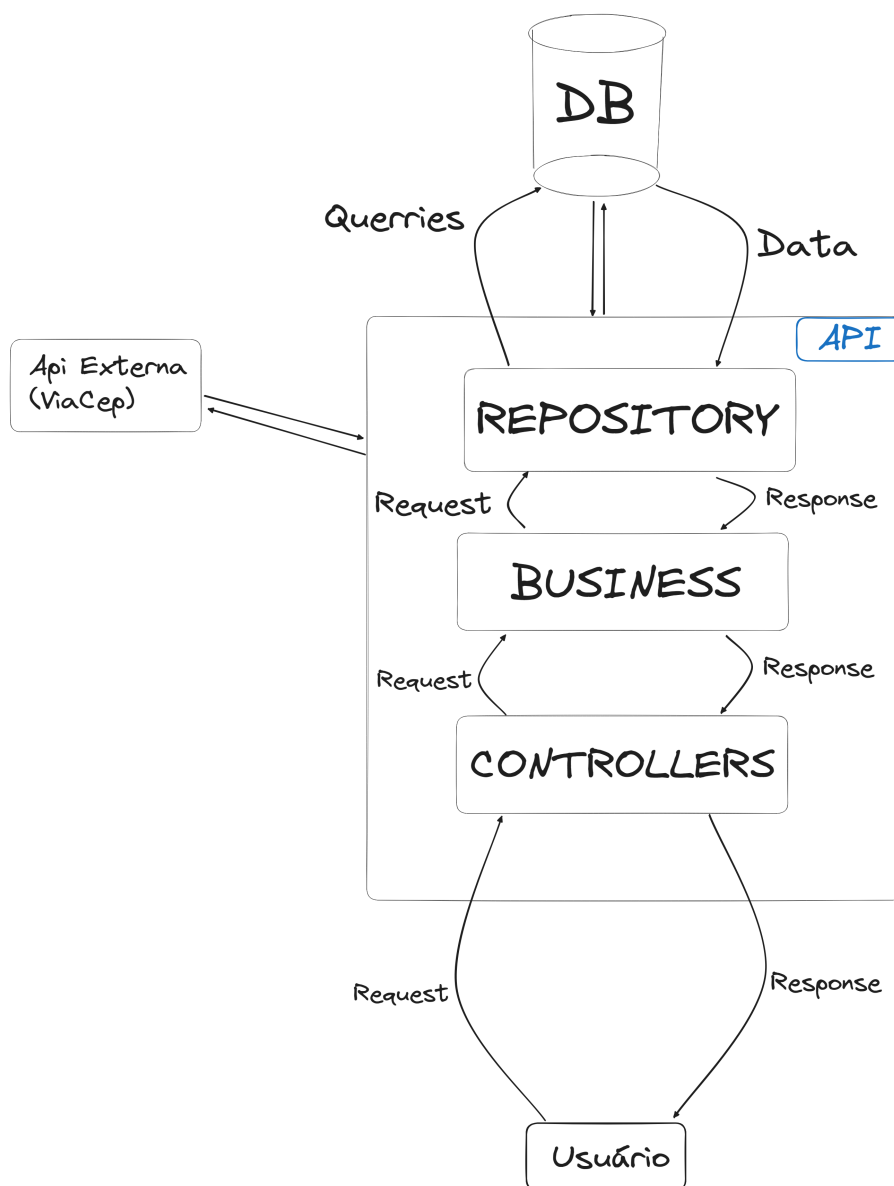
Documentação API

Api para CRUD de Usuarios e Perfis.

Para o desenvolvimento da nossa api focamos em uma arquitetura de 3 camadas simples. Usando as camadas de repositório, business e controller. Cada uma com duas definições.

- **Controller** → Responsável por receber as solicitações do usuário e enviá-las para a camada de negócio para serem processadas. Também é responsável por retornar a resposta da camada de negócio para o usuário.
- **Business** → Responsável por implementar as regras de negócio da aplicação. As regras de negócio são as regras que definem como a aplicação funciona. Por exemplo, uma regra de negócio pode definir como um usuário deve se registrar na aplicação.
- **Repository** → Responsável por acessar e armazenar os dados da aplicação. Para acessar os dados usamos o micro-ORM Dapper e o ORM completo Entity Framework Core. Dapper para leitura e EF Core para escrita.

Desenho da nossa arquitetura:



Sumário
DB => Database (SQL Server)

Desenvolvimento

Banco de dados → SQL Server;

Entidades (tabelas) do nosso projeto:

- **Usuario**
 - Id
 - Nome

- Email
 - DataNascimento
 - NumeroCasa
 - Cep
 - Logradouro
 - Complemento
 - Bairro
 - Localidade
 - Uf
 - Ibge
 - Gia
 - Ddd
 - Siafi
 - PerfilId
 - DataCriacao
 - DataAtualizacao
 - Ativo
-
- **Perfis**
 - Id
 - Nome
 - DataCriacao
 - DataAtualizacao
 - Ativo

Rotas do nosso projeto:

- **Usuario**

- Para obter todos os usuários:
 - Método Get
 - <https://ploapi20240117212848.azurewebsites.net/api/Usuario>
- Para obter um usuário específico:
 - Método Get
 - <https://ploapi20240117212848.azurewebsites.net/api/Usuario?id=idInformado>
- Para criar um usuário:
 - Método Post
 - <https://ploapi20240117212848.azurewebsites.net/api/Usuario>
 - Body

```
{
  "nome": "string",
  "email": "string",
  "dataNascimento": "2024-01-18T03:34:44.563Z",
  "numeroCasa": "string",
  "complemento": "string",
  "cep": "string",
  "perfilId": 0
}
```

- Para atualizar um usuário:
 - Atentar-se: id da rota deve ser igual ao do body;
 - Método Patch
 - <https://ploapi20240117212848.azurewebsites.net/api/Usuario?id=idInformado>
 - Body:

```
{
  "id": 0,
  "nome": "string",
  "email": "string",
}
```

```
"dataNascimento": "2024-01-18T03:35:00.973Z",
"numeroCasa": "string",
"complemento": "string",
"cep": "string",
"perfilId": 0
}
```

- Para deletar um usuário:
 - Método Delete
 - <https://ploapi20240117212848.azurewebsites.net/api/Usuario?id=IdInformado>

→ Instruções para manipulação na entidade de usuário:

- Nome não pode ser vazio ou nulo
- No Post e Patch preenchemos as informações de endereço com base no que o usuário inserir no cep, sendo assim é necessário preencher um valor válido para esse campo. Consultamos uma api externa para preencher os campos de endereço mostrados na seção de entidades. Api consultada: via cep - endpoint: viacep.com.br/ws/{cep}/json/
- Data de nascimento não é válida quando é maior que a data atual ou menor que 100 anos
- PerfilId precisa ser um id de um perfil existente

Informações → Campos de data de criação e data atualização são preenchidos automaticamente na inserção ou atualização das entidades.

O campo ativo mostrado nas entidades irá mostrar nas consultas apenas os retornos em que ativo estiver true. Conseguimos manipular os dados ativo false no banco. Serve para realizarmos uma exclusão lógica no delete, onde não perdemos os dados de fato no banco.

- **Perfil**
 - Para obter todos os perfis:

- Método Get
- <https://ploooapi20240117212848.azurewebsites.net/api/Perfil>
- Para obter um perfil específico:
 - Método Get
 - <https://ploooapi20240117212848.azurewebsites.net/api/Perfil?id=IdInformado>
- Para criar um perfil:
 - Método Post
 - <https://ploooapi20240117212848.azurewebsites.net/api/Perfil>
 - Body:

```
{  
  "nome": "string"  
}
```

- Para atualizar um perfil:
 - Método Patch
 - <https://ploooapi20240117212848.azurewebsites.net/api/Perfil?id=InformarId>
 - Body:

```
{  
  "id": 0,  
  "nome": "string"  
}
```

- Para deletar um perfil:
 - Método Delete
 - <https://ploooapi20240117212848.azurewebsites.net/api/Perfil?id=InformarId>

→ Instruções para manipulação na entidade de perfil:

- Nome não pode ser vazio ou nulo

Informações → campos de dataCriacao, dataAtualização e ativo agem da mesma maneira nesse campo.