

Tomorrow's Devs

(Arquitetura da Aplicação)



JrJunior

Sumário

| 1. | Stacks de Desenvolvimento | 3 |
|----|------------------------------|-----|
| 2. | Arquitetura Macro | 4 |
| 3. | Entidades | 4 |
| 4. | Camada Lógica do Backend | 5 |
| 5. | Camada Lógica do Frontend | 5 |
| 6. | Diagrama de Classe Backend | 6/7 |
| 7. | Diagrama de Classe Frontend | 8 |
| 8 | Padrões de Proieto Utilizado | 8 |

1. Stacks do Desenvolvimento

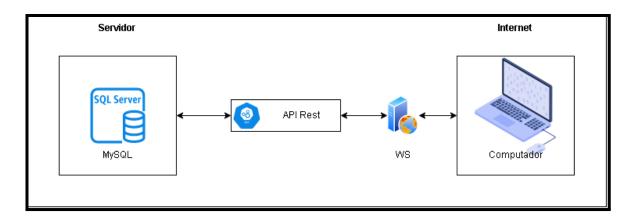
Abaixo segue as ferramentas utilizadas para o desenvolvimento do projeto.



- IDE utilizada para desenvolvimento: RAD Studio Delphi 10.4 Community Edition
- Framework API REST: Horse
- Banco de Dados: MySQL
- Ferramenta para Teste de API: Insomnia
- Postman
- Figma
- BR Modelo WEB
- Driagrams.Net
- Firemonkey

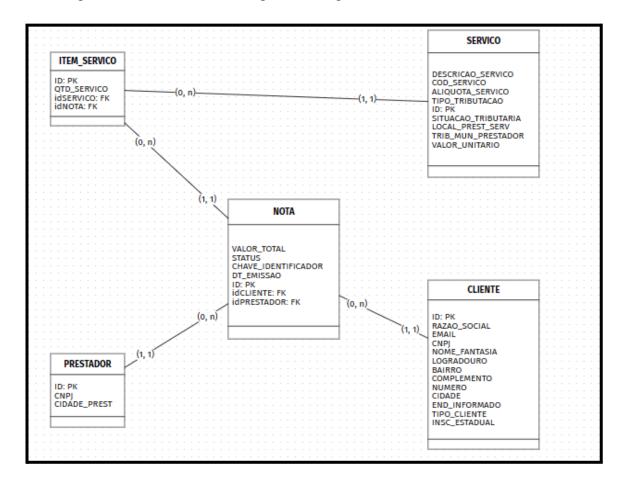
2. Arquitetura Macro

Abaixo podemos visualizar nossa arquitetura.

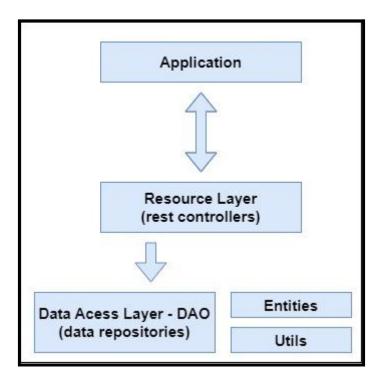


3. Entidades

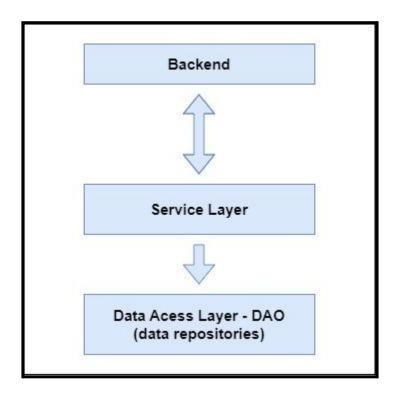
Segue abaixo nossa Modelagem de Lógica de Dados.



4. Camada Lógica Backend

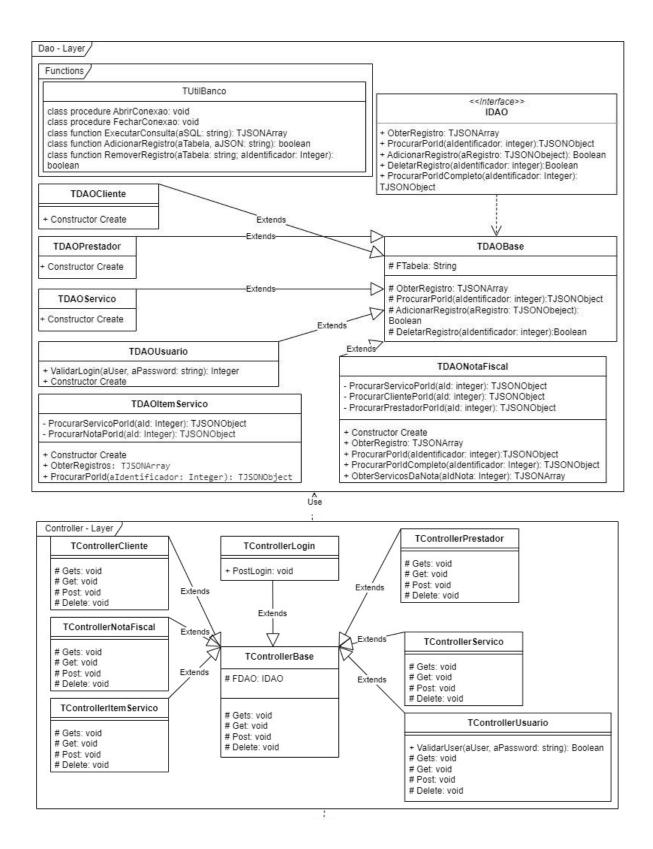


5. Camada Lógica Frontend





6. Diagrama de Classe Backend



Entity - Layer

TCliente

- Fld: Integer
- FRazaoSocial : String
- FEmail : String FCNPJ : String
- FNomeFantasia : String
- FLogradouro : String
- FBairro : String

- FComplemento : String FCumero : Integer FCidade : String FEnderecoInformado : String
- FTipoCliente : string FInscricaoEstadual : String
- FJSON: TJSONObject

- + Constructor Create + Destructor Destroy + Constructor Create(ald : Integer)

TServico

- Fld: Integer
- FDescricaoServico: String
- FCod_Servico : Integer
- FAliquotaServico : Double
- FSituacao_Tributaria : Integer
- FLocal_Prest_Servico : String FTrib_mun_Prestado : String
- FValorUnitario : Currency FJSON : TJSONObject

- + Constructor Create + Destructor Destroy + Constructor Create(ald : Integer)

TPrestador

- FId: Integer FCNPJ : String
- FCidadePrestador : String
- FJSON: TJSONObject
- + Constructor Create
- + Destructor Destroy + Constructor Create(ald : Integer)

TItem Servico

- Fld: Integer
- FQuantidadeServicos : Integer
- FNota : TNota
- FServico : TServico FJSON : TJSONObject
- + Constructor Create
- + Destructor Destroy
- + Constructor Create(ald : Integer)

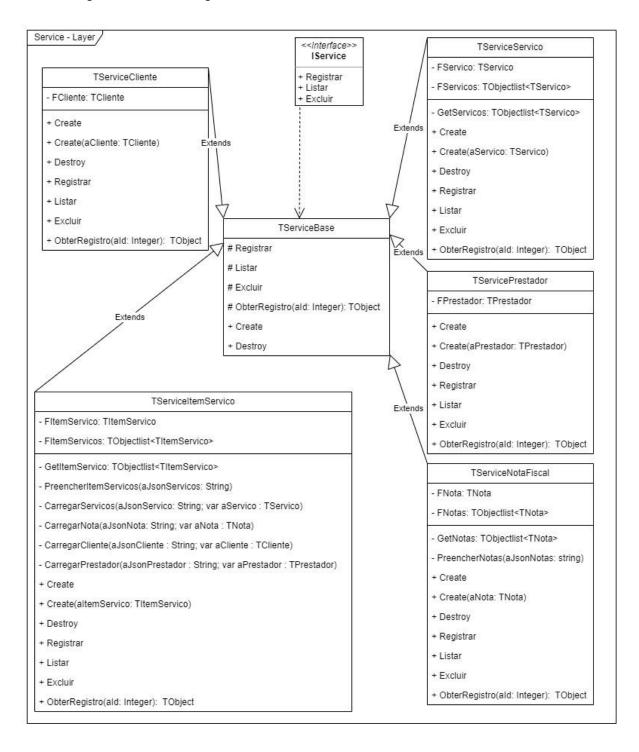
TNotaFiscal

- Fld: Integer
- FDataEmissao : TDate
- FChaveIndentificador : String
- FStatus : String

- FValorTotal : Currency FCliente : TCliente FPrestador : TPrestador FJSON : TJSONObject
- + Constructor Create
- + Destructor Destroy
- + Constructor Create(ald : Integer)

7. Diagrama de Classe Frontend

Segue abaixo no Diagrama de Classe do Frontend.



8. Padrões de Projeto Utilizado

Padrão Singleton