

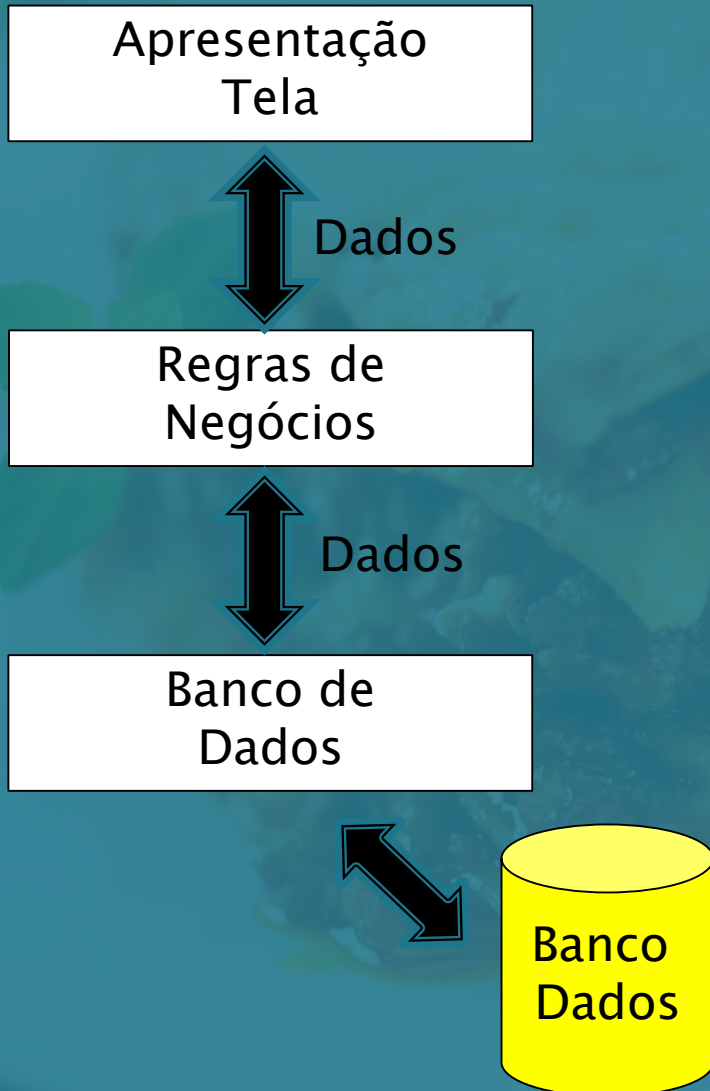


Framework PXC

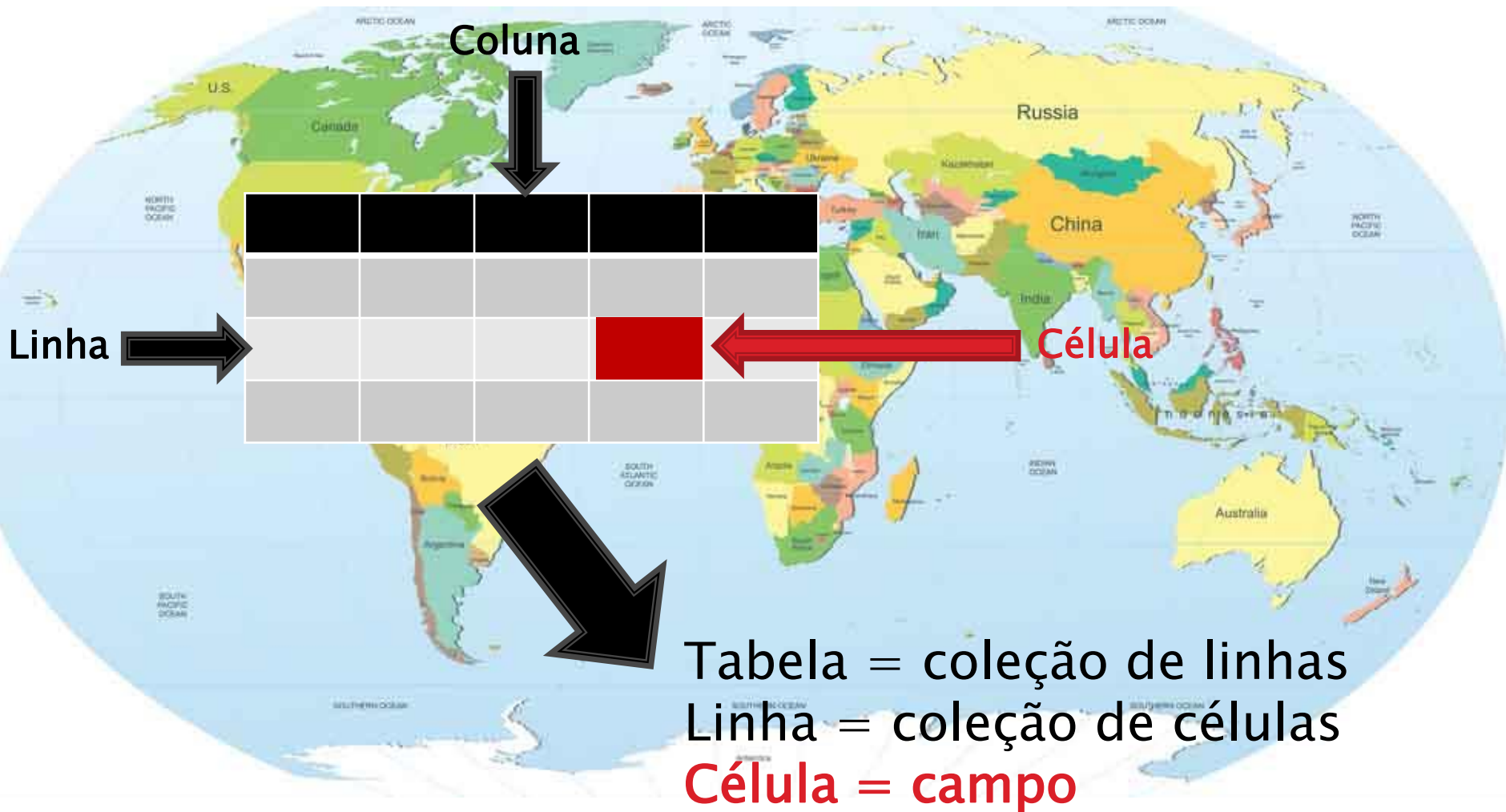
Nelio Maraschin – outubro/2018



Estrutura em camadas



Mapeamento de tabela



O que é um campo?

É a representação de uma informação
Exemplo: tamanho do terreno

```
Int32 tamanhoTerreno;
```

Como representar um campo nulo?



- ▶ NULL é WHERE TAM_TERRENO IS NULL?
- ▶ Como não considerar o campo no WHERE?

Int32? tamanhoTerreno;

Como representar um campo nulo?



- ▶ Foi setado?
- ▶ Tem conteúdo?

`CampoTabela` < `Int32` > `tamanhoTerreno`;

Como representar um campo nulo?

```
public struct CampoTabela<T> : ICampo, IXmlSerializable
{
    public CampoTabela(T conteudo);
    public object Conteudo { get; set; }
    public bool FoiSetado { get; }
    public bool TemConteudo { get; }
    public T LerConteudoOuPadrao();
    public override string ToString();
    //...
}
```

`CampoTabela<Int32> tamanhoTerreno;`



CampoTabela<T>

```
public object Conteudo { get; set; }
```

Retorna o conteúdo do campo,
pode ser **null** ou o **objeto** atribuído.



CampoTabela<T>

```
public bool FoiSetado { get; }
```

Retorna **true** se foi atribuído qualquer conteúdo ao campo, isso inclui **null**.



CampoTabela<T>

```
public bool TemConteudo { get; }
```

Retorna **true** se tiver sido atribuído um conteúdo diferente de **null**.



CampoTabela<T>

```
public T LerConteudoOuPadrao();
```


Se TemConteudo for **true**, então retorna o conteúdo do campo convertido para o tipo T;

Caso contrário, retorna **default(T)** (mesmo que FoiSetado seja **false**).





Testando CampoTabela

```
CampoTabela<Int32> tamanhoTerreno;
```

- ▶ .FoiSetado? 
- ▶ .TemConteudo? 
- ▶ .Conteudo => `null`
- ▶ .LerConteudoOuPadrao() => `0` em `Int32`

Testando CampoTabela

```
CampoTabela<Int32> tamanhoTerreno;  
tamanhoTerreno = 25;
```

- ▶ .FoiSetado? 
- ▶ .TemConteudo? 
- ▶ .Conteudo => 25
- ▶ .LerConteudoOuPadrao() => 25 em Int32

Testando CampoTabela

```
CampoTabela<Int32> tamanhoTerreno;  
tamanhoTerreno = new CampoTabela<Int32>(null);
```

- ▶ .FoiSetado? 
- ▶ .TemConteudo? 
- ▶ .Conteudo => null
- ▶ .LerConteudoOuPadrao() => 0 em Int32

Classe TOTabela

A - B - C -

Todas propriedades públicas do tipo
`CampoTabela<T>`

Método “PopularRetorno” para atribuir às variáveis privadas o conteúdo do banco de dados

Classe TOTabela – PopularRetorno

```
public override void PopularRetorno(Linha linha)
{
    foreach (Campo campo in linha.Campos)
    {
        switch (campo.Nome)
        {
            case "CODIGO":
                this.codigo = this.LeCampoTabela<Int32>(campo.Conteudo);
                break;
            case "DATA":
                this.data = this.LeCampoTabela<DateTime>(campo.Conteudo);
                break;
            case "TEXTO":
                this.texto = this.LeCampoTabela<String>(campo.Conteudo);
                break;
            case "VALOR":
                this.valor = this.LeCampoTabela<Double>(campo.Conteudo);
                break;
            default:
                break;
        }
    }
}
```

Camada de dados



Comandos

INSERT INTO <tabela> (<lista de campos>) VALUES (<lista de valores>)

UPDATE <tabela> SET <lista de campos = valores> WHERE <pk = valor>

DELETE FROM <tabela> WHERE <pk = valor>

SELECT <lista de campos> FROM <tabela>
WHERE <lista de campos = valores>



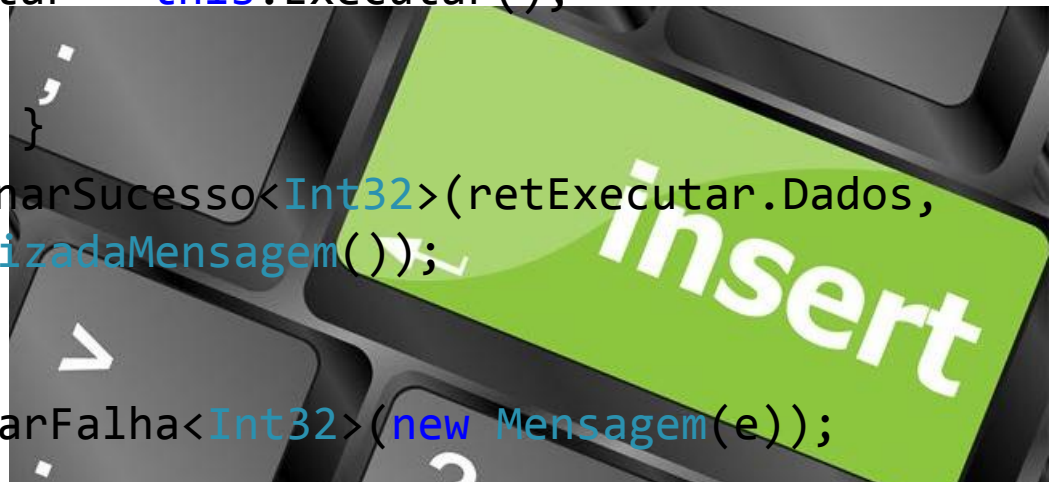
SQL

Banco de dados



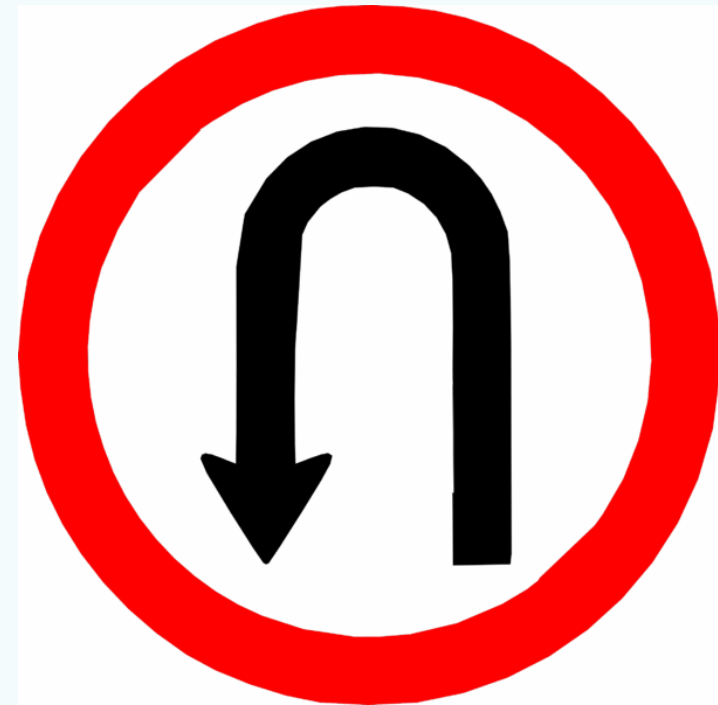
INSERT INTO <tabela> (<lista de campos>) VALUES (<lista de valores>)

```
public Retorno<Int32> Incluir(TOTabela1 toTabela1)
{
    try {//...
        this.Sql.Comando.Append("INSERT INTO (");
        this.MontarCamposChave(this.Sql.MontarCampoInsert, toTabela1);
        this.MontarCampos(this.Sql.MontarCampoInsert, toTabela1);
        this.Sql.Comando.Append(") VALUES (");
        this.Sql.Comando.Append(this.Sql.Temporario.ToString());
        this.Sql.Comando.Append(") ");
        Retorno<Int32> retExecutar = this.Executar();
        if (!retExecutar.Ok) {
            return retExecutar; }
        return this.Infra.RetornarSucesso<Int32>(retExecutar.Dados,
            new OperacaoRealizadaMensagem());
    }
    catch (Exception e) {
        return this.Infra.RetornarFalha<Int32>(new Mensagem(e));
    }
}
```



Classe Retorno

```
public struct Retorno<T>
{
    public readonly T Dados;
    public readonly Mensagem Mensagem;
    public readonly bool Ok;
}
```



Classe Retorno

```
public readonly bool Ok;
```

Retorna **true** se a saída do método for por
return this.Infra.RetornarSucesso;

Retorna **false** se a saída do método for por
return this.Infra.RetornarFalha ou por uma
Exception;

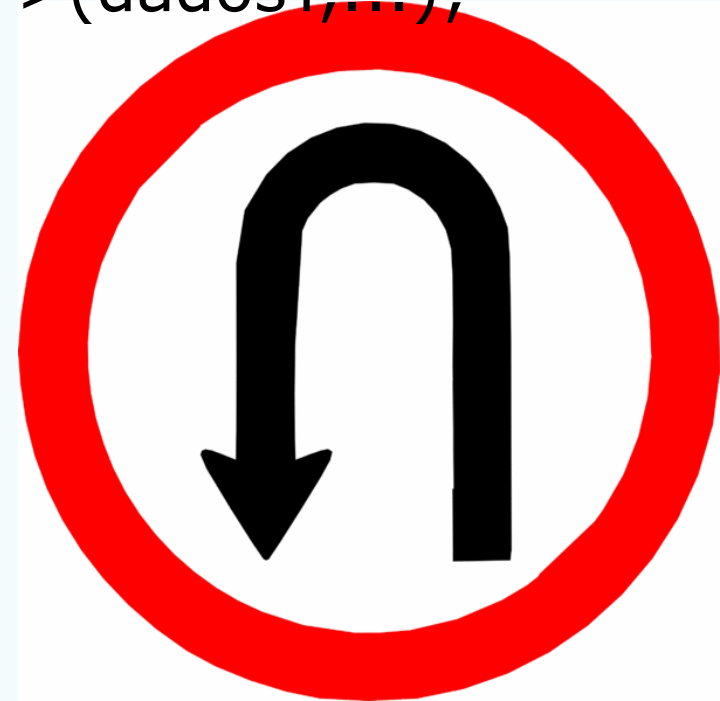


Classe Retorno

```
public readonly T Dados;
```

Só possuí informação em caso de sucesso na execução, seu valor é setado na chamada ao método

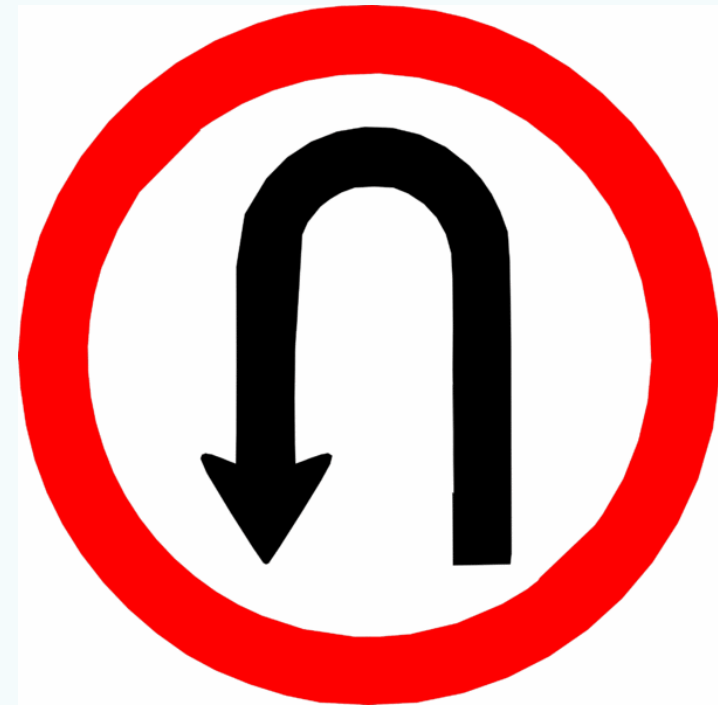
```
this.Infra.RetornarSucesso<T>(dadosT,...);
```



Classe Retorno

```
public readonly Mensagem Mensagem;
```

Possui a mensagem de retorno, tanto em caso de sucesso ou em caso de falha.



Classe Mensagem

```
public class Mensagem
{
    protected string mensagem;
    public Mensagem();
    public Mensagem(Exception excecao);
    public string ParaOperador { get; }
    public override string ToString();
}
```



Banco de dados



UPDATE <tabela> SET <lista de campos = valores> WHERE <pk = valor>

```
public Retorno<Int32> Alterar(TOTabela1 toTabela1)
{
    //...
    this.Sql.Comando.Append("UPDATE <tabela> SET ");
    this.MontarCampos(this.Sql.MontarCampoSet, toTabela1);
    //montar campos da PK
    this.MontarCamposChave(this.Sql.MontarCampoWhere, toTabela1);
    Retorno<Int32> retExecutar = this.Executar();
    if (!retExecutar.Ok) {
        return retExecutar; }
    if (retExecutar.Dados == 0) {
        return this.Infra.RetornarFalha<Int32>(
            new RegistroInexistenteMensagem());
    }
    return this.Infra.RetornarSucesso<Int32>(
        retExecutar.Dados, new OperacaoRealizadaMensagem());
    //...
}
```

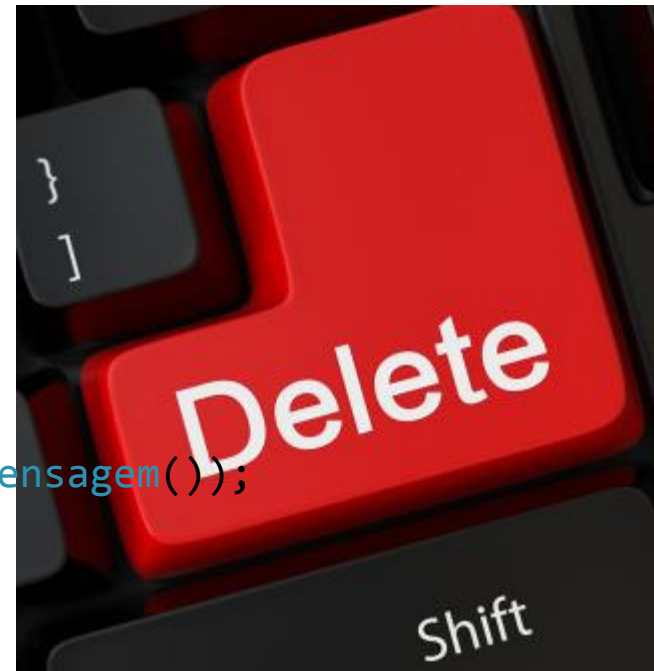


Banco de dados



DELETE FROM <tabela> WHERE <pk = valor>

```
public Retorno<Int32> Excluir(TOTabela1 toTabela1)
{
    //...
    this.Sql.Comando.Append("DELETE FROM <tabela> ");
    //montar campos da PK
    this.MontarCamposChave(this.Sql.MontarCampoWhere, toTabela1);
    Retorno<Int32> retExecutar = this.Executar();
    if (!retExecutar.Ok) {
        return retExecutar; }
    if (retExecutar.Dados == 0) {
        return this.Infra.RetornarFalha<Int32>(
            new RegistroInexistenteMensagem());
    }
    return this.Infra.RetornarSucesso<Int32>(
        retExecutar.Dados, new OperacaoRealizadaMensagem());
    //...
}
```



Banco de dados



SELECT <lista de campos> FROM <tabela> WHERE <lista de campos = valores>

```
public Retorno<List<TOTabela1>> Listar(TOTabela1 toTabela1)
{
    //...
    this.Sql.Comando.Append("SELECT <campos> FROM <tabela> ");
    this.MontarCamposChave(this.Sql.MontarCampoWhere, toTabela1);
    this.MontarCampos(this.Sql.MontarCampoWhere, toTabela1);
    Retorno<Int32> retListar = this.Consultar();
    if (!retListar.Ok) { return retListar; }
    List<TOTabela1> lista = new List<TOTabela1>();
    foreach (Linha linha in retListar.Dados) {
        TOTabela1 toTabela1Linha = new TOTabela1();
        toTabela1Linha.PopularRetorno(linha);
        lista.Add(toTabela1Linha); }
    return this.Infra.RetornarSucesso<List<TOTabela1>>(
        lista, new OperacaoRealizadaMensagem());
    //...
}
```



Regras de negócio

//...

```
if (!toTabela1.Codigo.TemConteudo)
{
    return this.Infra.RetornarFalha<Int32>(
        new CampoObrigatorioMensagem("Código"));
}
if (toTabela1.Codigo.LerConteudoOuPadrao() < 0)
{
    return this.Infra.RetornarFalha<Int32>(
        new MensagemTabela1(
            TipoFalha.FalhaMenorIgualZero, "Código"));
}
```



Regras de negócio

Escopo transacional com efetivar transação

```
using (EscopoTransacional escopo =  
        this.Infra.CriarEscopoTransacional())  
{  
    escopo.EfetivarTransacao();  
}
```



commit

Regras de negócio

```
BDTabela1 bdTabela1 = this.Infra.InstanciarBD<BDTabela1>();
Retorno<Int32> retIncluir;

using (EscopoTransacional escopo =
        this.Infra.CriarEscopoTransacional())
{
    retIncluir = bdTabela1.Incluir(toTabela1);
    if (!retIncluir.Ok)
    {
        return this.Infra.RetornarFalha<Int32>(retIncluir.Mensagem);
    }
    escopo.EfetivarTransacao();
}
return this.RetornarSucesso<Int32>(retIncluir.Dados,
    new OperacaoRealizadaMensagem("Inclusão"));
```