

FÁBRICA DE DESENVOLVIMENTO

FÁBRICA DE DESENVOLVIMENTO - 2018

Prof. Thiago T. I. Yamamoto

#10 - MODEL - ORM DAPPER



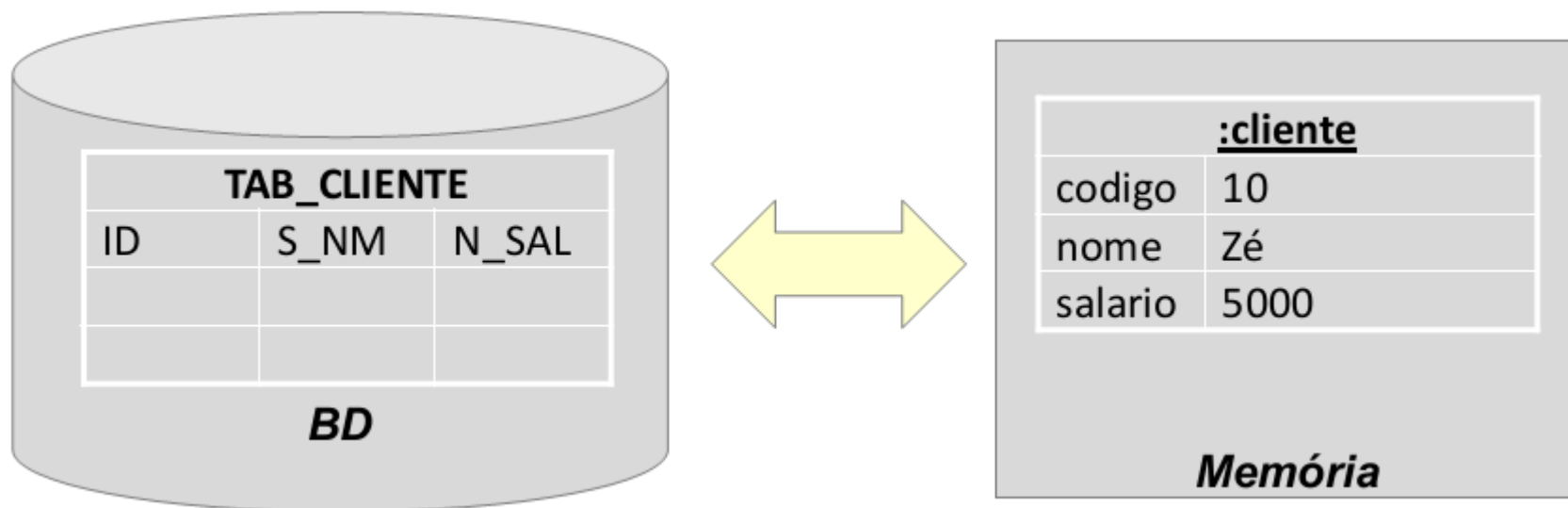
thiagoyama

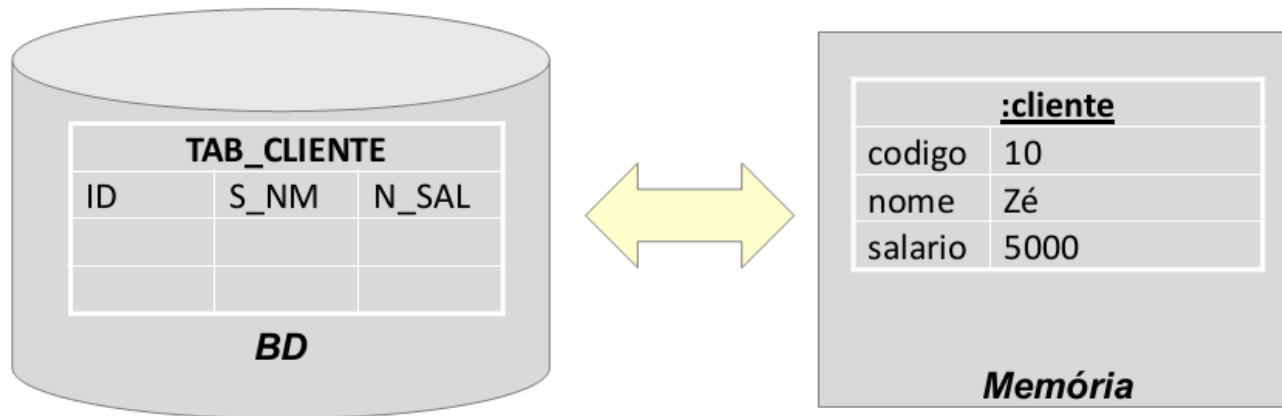


thiagoyama@gmail.com

#10 – MODEL – ORM DAPPER

- Framework ORM
- Dapper
- Tabelas e Banco de Dados
- Aplicação web e Banco de dados
- Manipulando Entidades
- Pesquisas





Mapeamento:

Cliente -> TAB_CLIENTE

codigo -> ID

nome -> S_NM

salario -> N_SAL

ORM - DAPPER

- Frameworks ORM possuem o objetivo de simplificar a conversão de informações estruturadas em objetos, permitindo uma maior produtividade na construção de aplicações.
- Na plataforma .NET temos o **EntityFramework** e o **Nhibernate** como os mais populares.
- Os frameworks de ORM podem ser úteis em um grande número de situações, porém podem apresentar limitações em alguns cenários específicos, como consultas envolvendo tabelas com grande volume de dados.
- Uma forma de solucionar esse problema é utilizar **ADO.NET**.



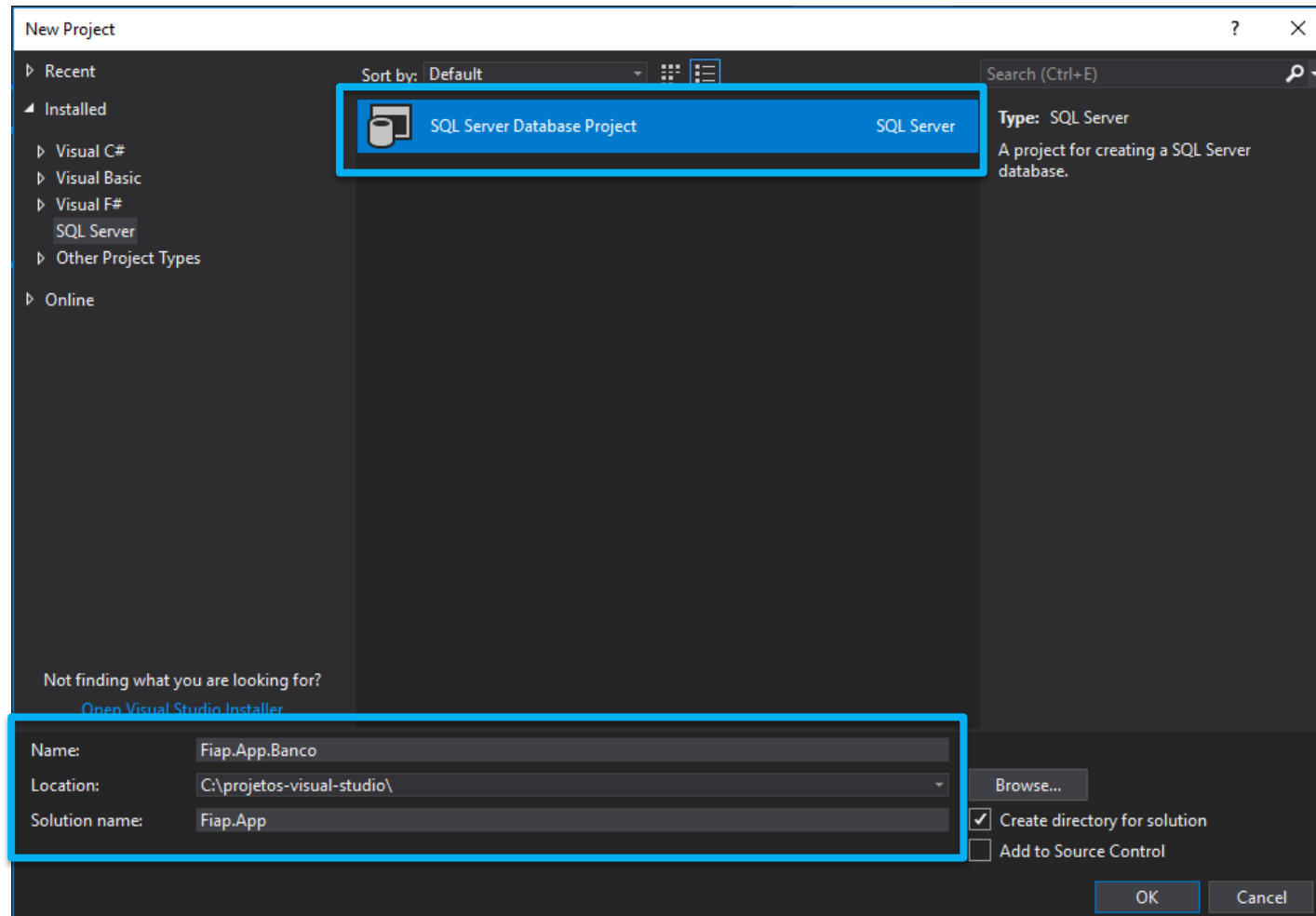
▪ DAPPER

- É um micro-ORM que disponibiliza **Extension Methods** que simplificam o trabalho com os objetos de conexão do ADO.NET.
- Com a integração com ADO.NET podemos trabalhar com qualquer tecnologia relacional compatível com o .NET (SQL Server, Oracle, PostgreSQL, MySQL e etc.)



BANCO DE DADOS - TABELAS

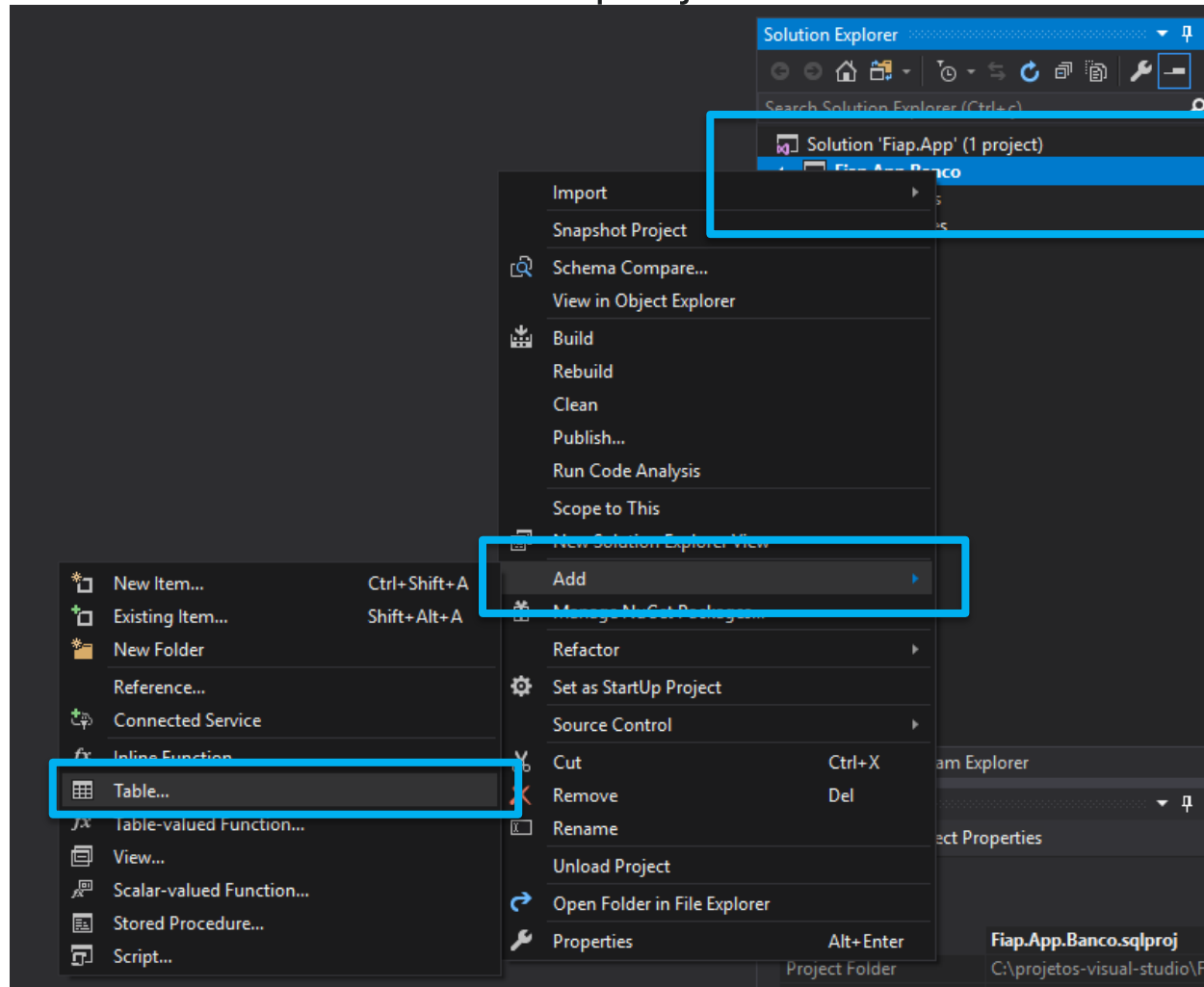
CRIANDO UM NOVO PROJETO



CRIANDO UMA TABELA

Vamos criar uma tabela!

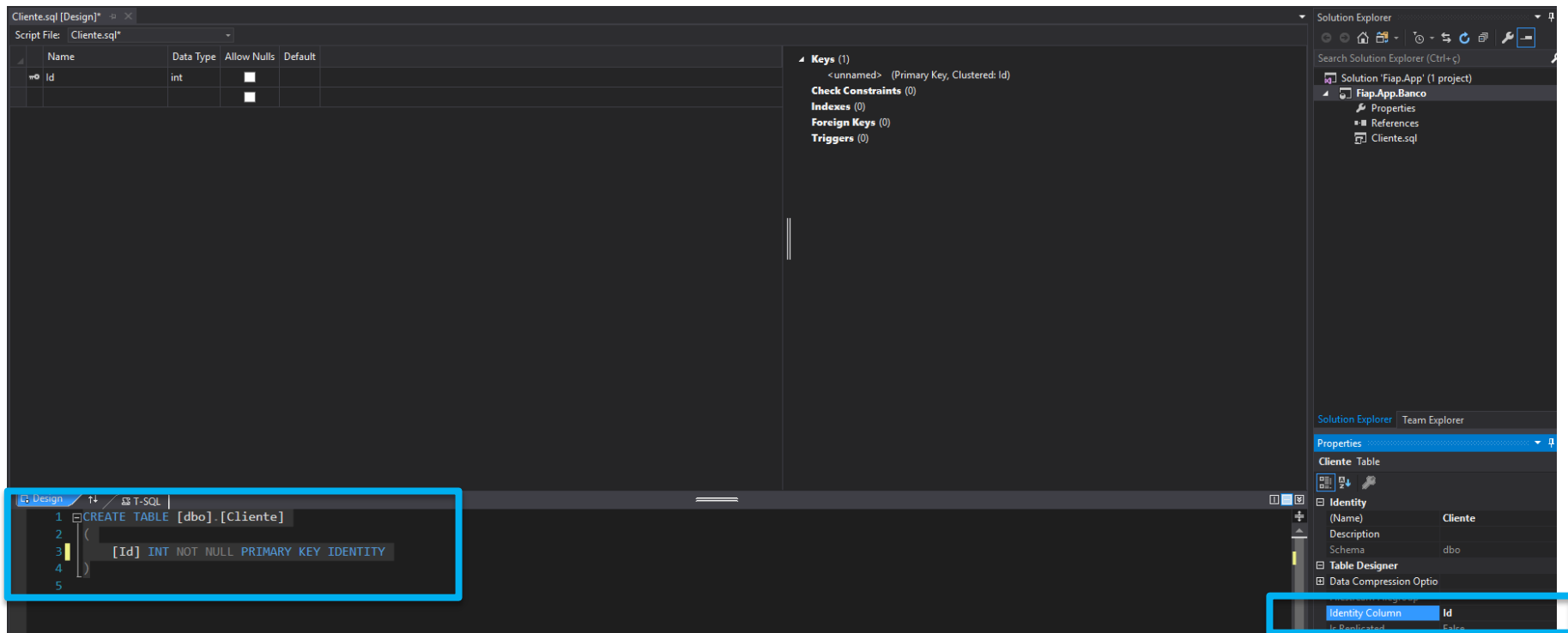
Clique com o botão direito no projeto e escolha: Add -> Table



- **INT:** Armazena valores numéricos inteiros, variando de -2.147.483.648 a 2.147.483.647;
- **FLOAT:** Armazena valores numéricos aproximados com precisão de ponto flutuante, variando de $-1.79E + 308$ a $1.79E + 308$;
- **NUMERIC(18,0):** Armazena valores numéricos com casas decimais, utilizando precisão.
- **DECIMAL(18,0):** Tem as mesmas funcionalidades do tipo NUMERIC, a diferença é que o DECIMAL faz parte do padrão ANSI e NUMERIC é mantido por compatibilidade.
- **DATETIME:** Armazena data e hora, com precisão de centésimos de segundos;
- **CHAR(N):** Armazena N caracteres fixos (até 8.000) no formato não Unicode.
- **VARCHAR(N):** Armazena N caracteres (até 8.000) no formato não Unicode;
- **BIT:** Armazena bits ou seja somente poderá conter os valores lógicos 0 ou 1.

[https://msdn.microsoft.com/en-us/library/ms710150\(v=vs](https://msdn.microsoft.com/en-us/library/ms710150(v=vs)

A propriedade **IDENTITY** é utilizada para atributos (campos/colunas) das tabelas nas funções CREATE TABLE e ALTER TABLE, e tem como finalidade incrementar um valor a cada nova inserção



Vamos adicionar mais colunas na tabela!

Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
Nome	varchar(100)	<input type="checkbox"/>	
DataNascimento	datetime	<input checked="" type="checkbox"/>	
Especial	bit	<input type="checkbox"/>	
Saldo	decimal(18,2)	<input checked="" type="checkbox"/>	

Script File: Cliente.sql*

Keys (1)
<unnamed> (Primary Key, Clustered: Id)

Check Constraints (0)

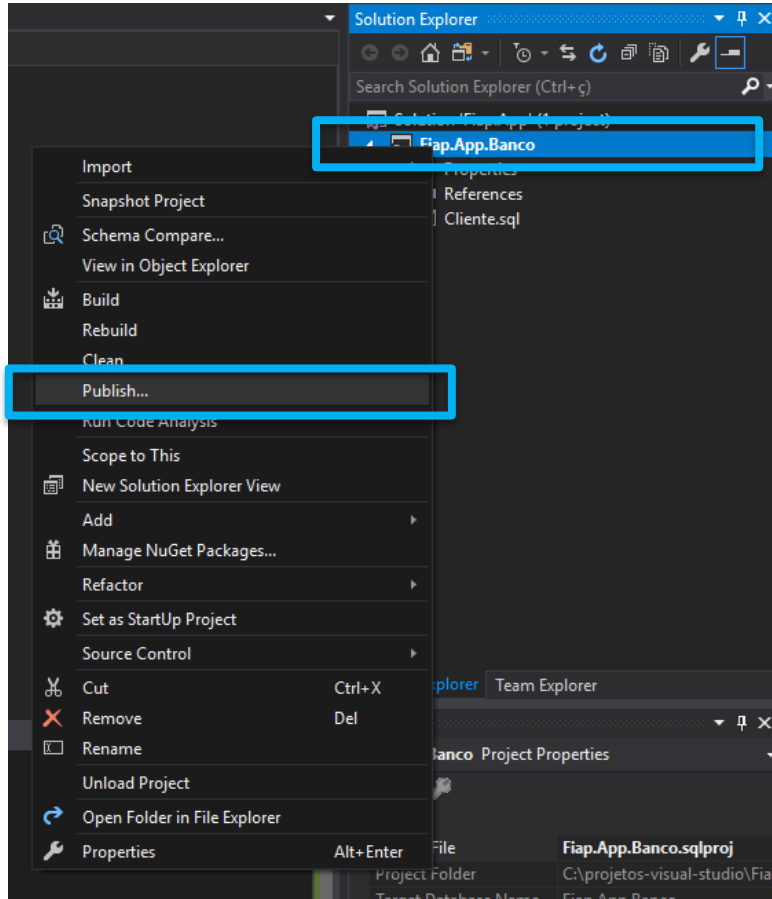
Indexes (0)

Foreign Keys (0)

Triggers (0)

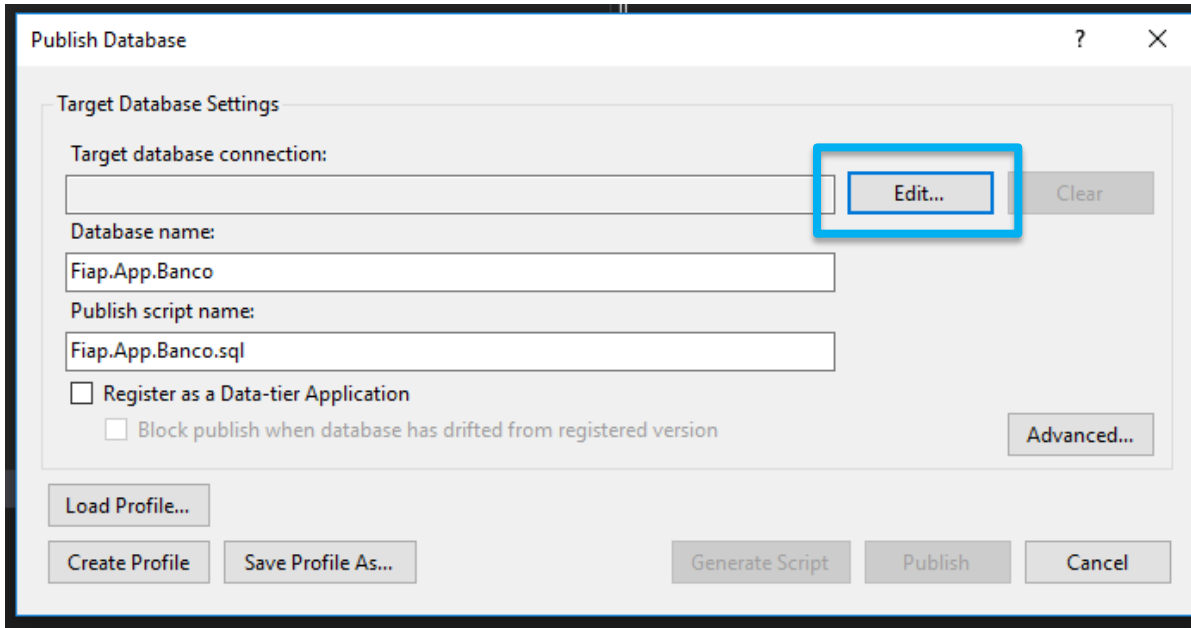
```
1 CREATE TABLE [dbo].[Cliente]
2 (
3     [Id] INT NOT NULL PRIMARY KEY IDENTITY,
4     [Nome] VARCHAR(100) NOT NULL,
5     [DataNascimento] DATETIME NULL,
6     [Especial] BIT NOT NULL,
7     [Saldo] DECIMAL(18, 2) NULL
8 )
9
```

PUBLIQUE O PROJETO!



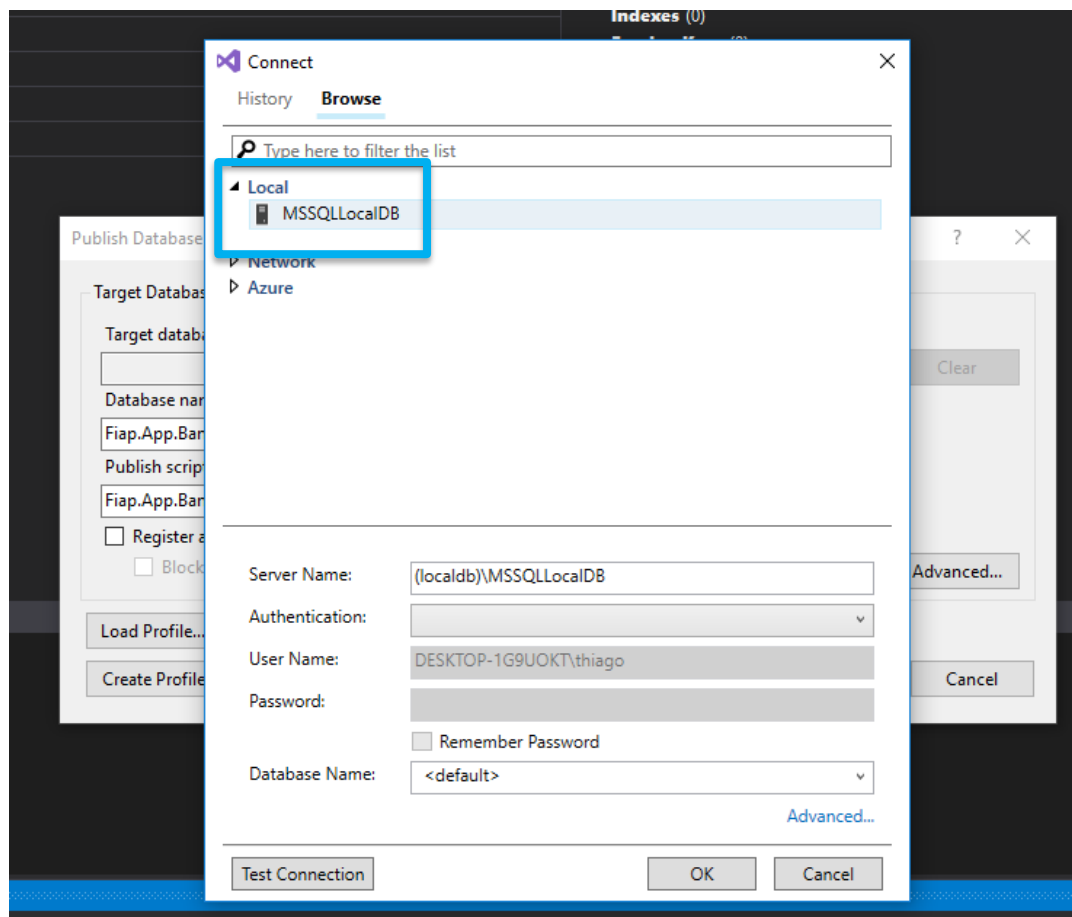
Para **criar as tabelas** precisamos publicar o projeto, para isso, clique com o botão direito do mouse no projeto e escolha **“Publish..”**

CONFIGURE O BANCO DE DADOS!



Precisamos escolher a base de dados para trabalhar com o projeto.

CONFIGURE O BANCO DE DADOS!



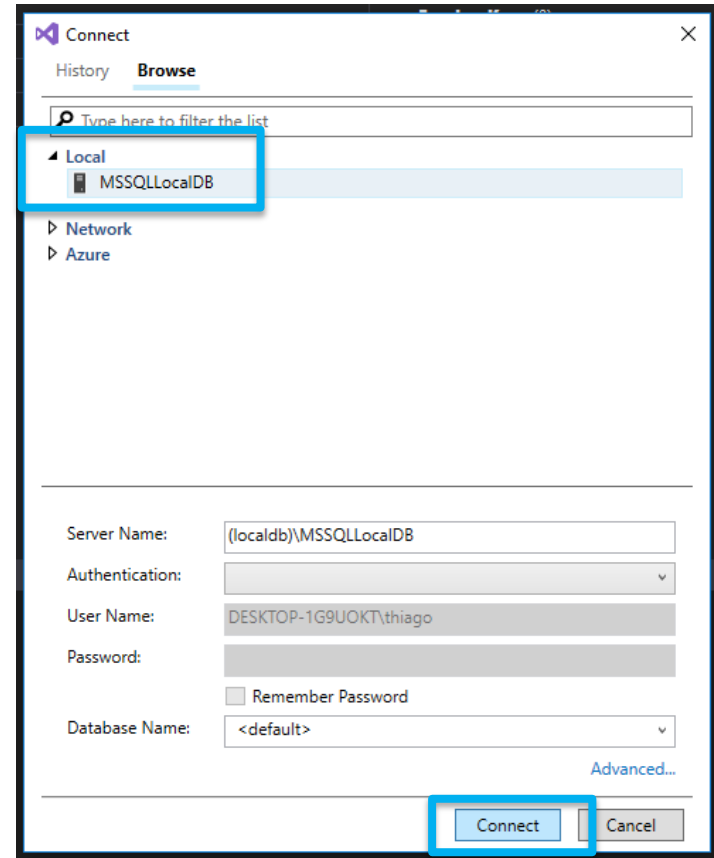
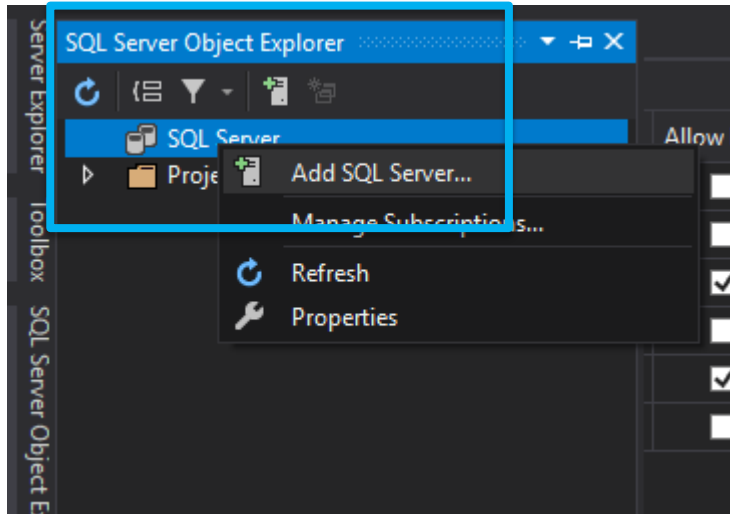
Vamos trabalhar com o **SQL Server LocalDB**.

CONFIGURE O BANCO DE DADOS!

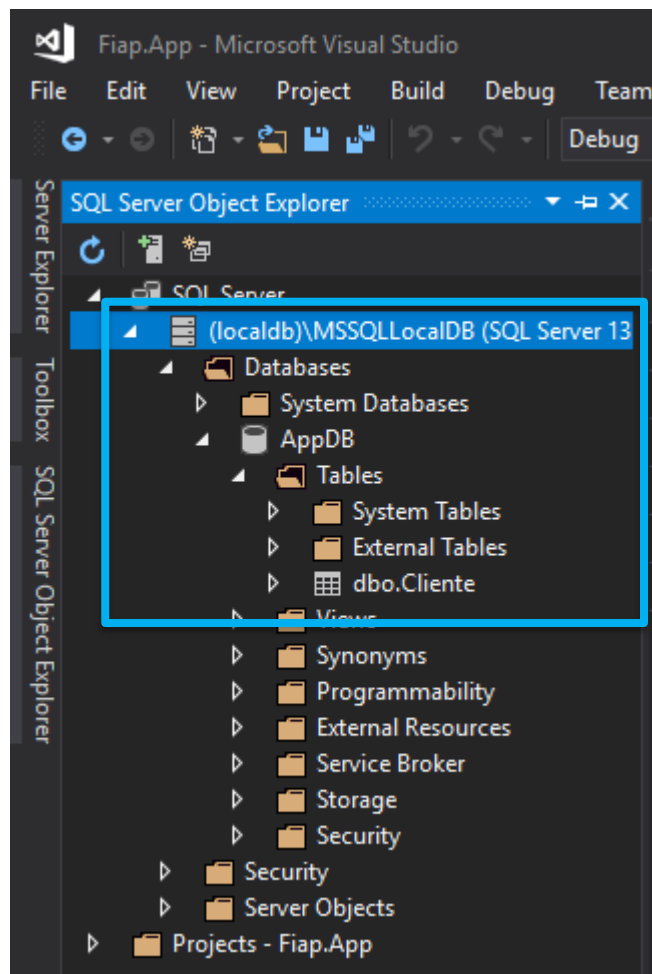
The screenshot shows the 'Publish Database' dialog box. The 'Target Database Settings' section is expanded. The 'Target database connection' field contains 'Data Source=(localdb)\MSSQLLocalDB;Integrated Security=True;Persist Security Info'. The 'Database name' field is highlighted with a blue box and contains 'AppDB'. The 'Publish script name' field contains 'Fiap.App.Banco.sql'. There are checkboxes for 'Register as a Data-tier Application' and 'Block publish when database has drifted from registered version'. At the bottom, there are buttons for 'Load Profile...', 'Create Profile', 'Save Profile As...', 'Generate Script', 'Publish', and 'Cancel'.

Configure o nome do banco de dados.

PARA VER O BANCO...



No **SQL Server Object Explorer** crie uma nova conexão.

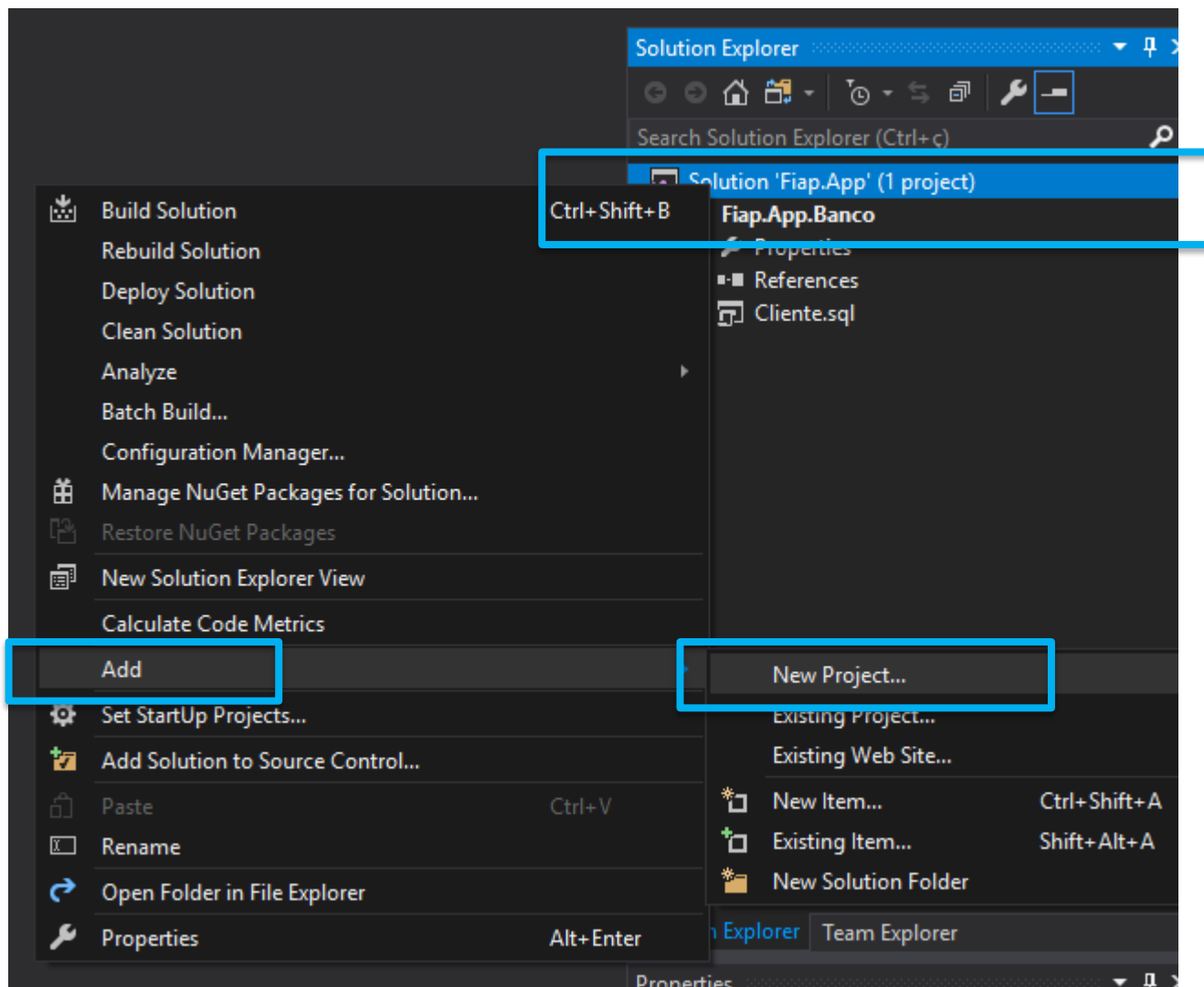


O banco e as tabelas foram criadas!

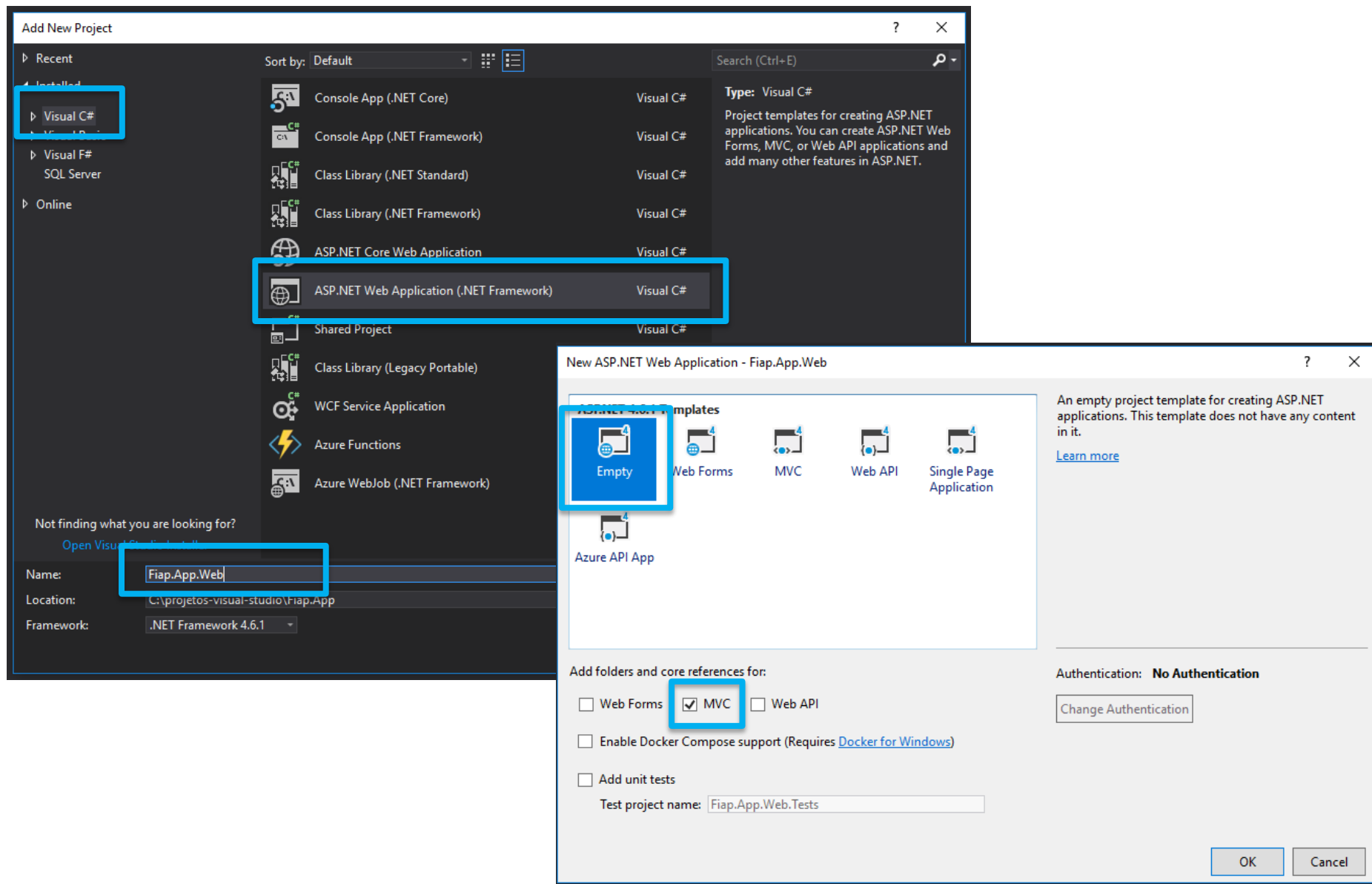
APLICAÇÃO WEB

CRIE UM PROJETO ASP.NET MVC

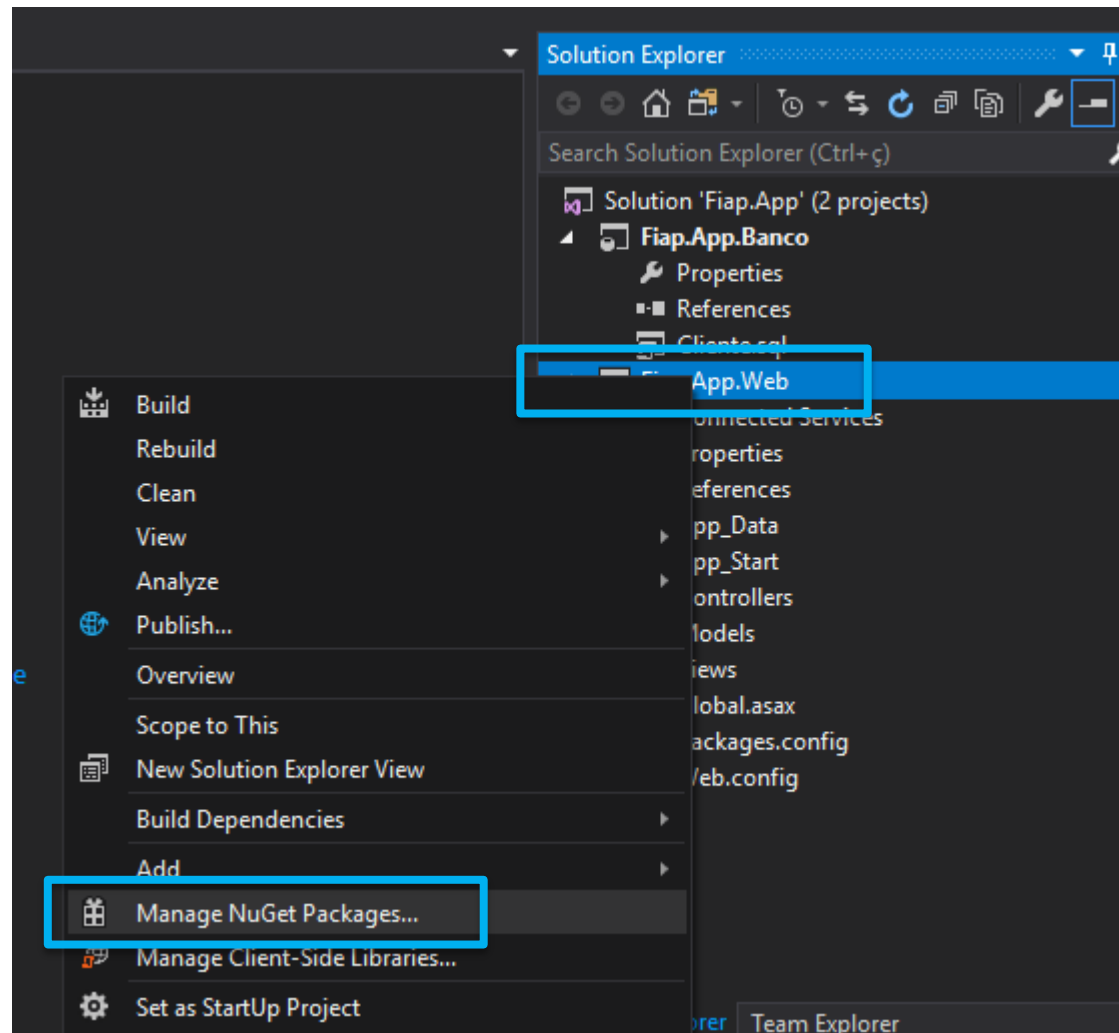
Vamos adicionar mais um projeto na solução!



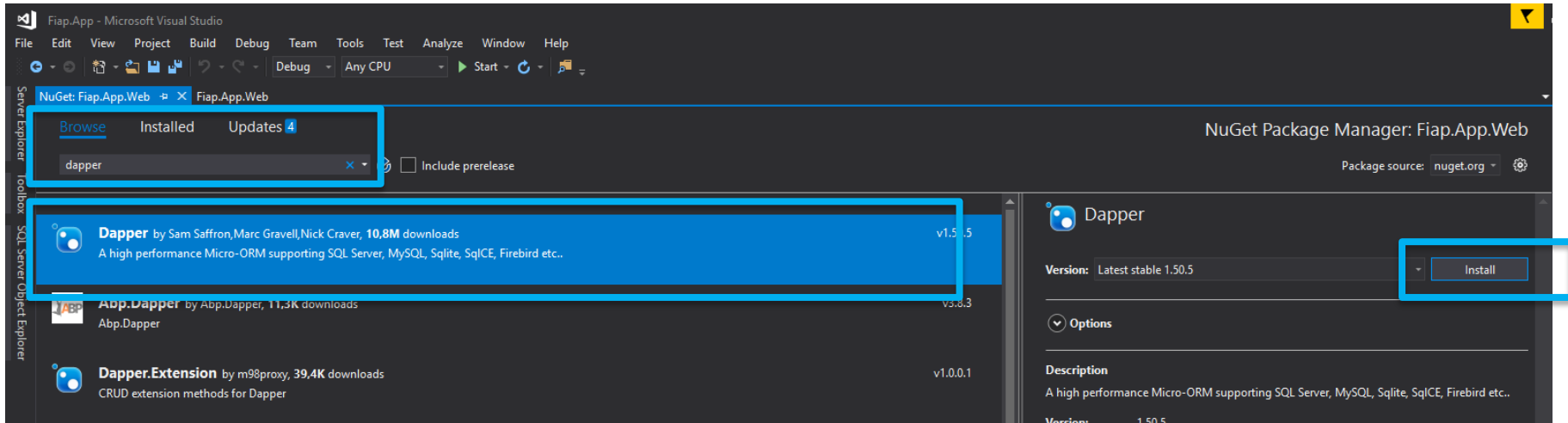
PROJETO ASP.NET MVC



ADICIONANDO O DAPPER



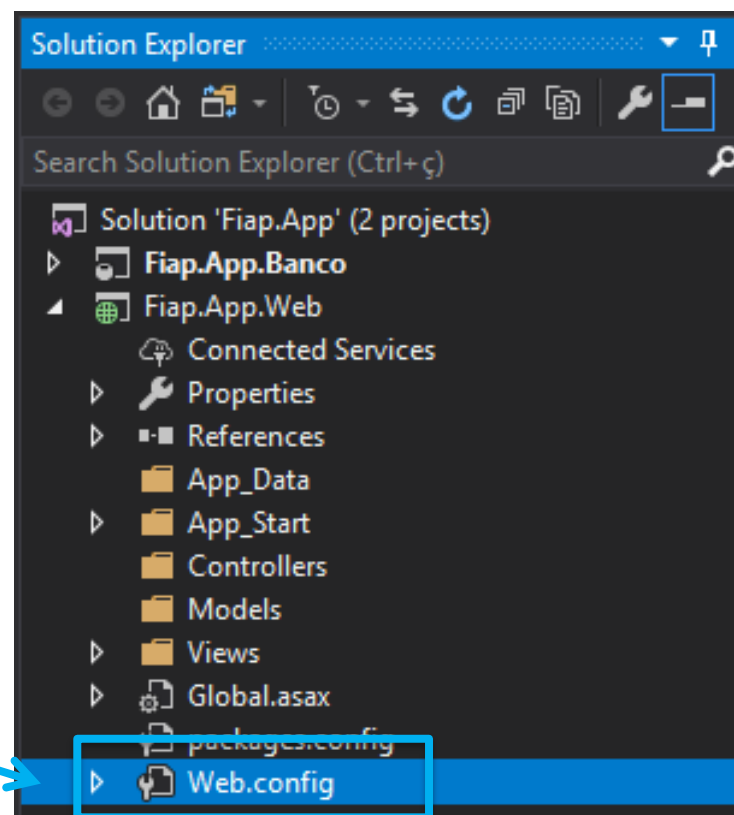
ADICIONANDO O DAPPER



- **SQL Server:** sistema de gerenciamento de banco de dados da Microsoft.
- **SQL Server Local Db:** versão simplificada do SQL Server Express, voltada para os desenvolvedores.

Para configurar o banco de dados existe uma string de conexão no arquivo de configuração do projeto.

Abra o arquivo Web.config que está na raiz do projeto



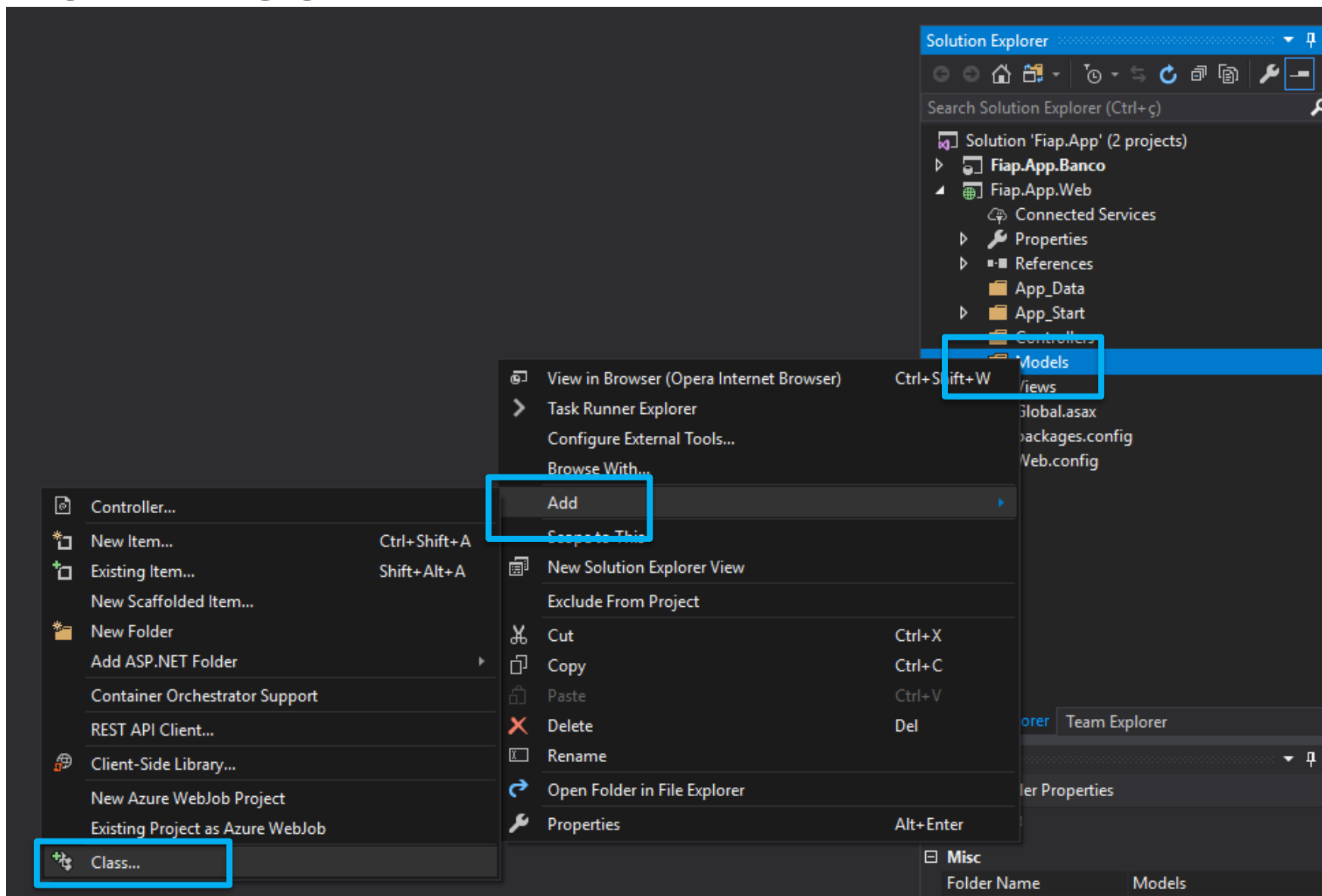
CONFIGURANDO O BANCO DE DADOS

- Adicione a string de conexão dentro da tag <configuration>.
- Ajuste o nome do banco e dados e da string de

SQL Server Local Db

```
<connectionStrings>
  <add name="AppFiapDB"
    connectionString="Data Source=(LocalDB)\MSSQLLocalDB;
    Database=AppDB;
    Integrated Security=True"
    providerName="System.Data.SqlClient"/>
</connectionStrings>
```

CRIANDO A CLASSE PARA A TABELA DO BANCO



- Crie uma classe que representa a tabela do banco de dados:

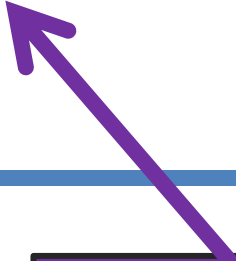
```
public class Cliente
{
    public int Id { get; set; }
    public string Nome { get; set; }
    public DateTime DataNascimento { get; set; }
    public bool Especial { get; set; }
    public decimal Saldo { get; set; }
}
```

OPERAÇÕES BÁSICAS

CONEXÃO COM O BANCO

- Para obter uma conexão com o banco de dados, precisamos passar a configuração da connection string (web.config).

```
IDbConnection db = new SqlConnection(  
    ConfigurationManager.  
    ConnectionStrings["AppFiapDB"].  
    ConnectionString);
```



Configuração do
banco no

CADASTRAR

- Defina o SQL de **insert** com os parâmetros de *values* com os mesmos nomes das propriedades da classe Modelo.

```
string sql = "INSERT INTO Cliente (Nome, DataNascimento, Saldo, Especial) VALUES (@Nome, @DataNascimento, @Saldo, @Especial); SELECT CAST(SCOPE_IDENTITY() as int)";
```

```
int id = db.Query<int>(sql, cliente).Single();
```

ID gerado
pelo banco de

Query para o
Dapper recuperar

- Defina o SQL para realizar a pesquisa e defina o parâmetro com o mesmo nome da query.

```
string sql = "SELECT * FROM Cliente WHERE id = @id";  
var cliente = db.Query<Cliente>(sql,  
    new { id = 1 }).SingleOrDefault();
```



Identificador (id) da

- Defina o SQL para remover um cliente e defina o parâmetro com a chave primária da tabela.

```
db.Execute(  
    "DELETE From Cliente WHERE id = @id", new { id = 1 });
```

Id que será



ATUALIZAR

- Crie o SQL de UPDATE e defina os parâmetros com o mesmo nome das propriedades da classe.

```
string sql = "UPDATE Cliente SET Nome = @Nome,  
DataNascimento = @DataNascimento, Saldo =  
@Saldo, Especial = @Especial WHERE Id = @Id";  
  
db.Execute(sql, cliente);
```

- Para listar todos execute o SQL informando qual a classe será utilizada como retorno.

```
IList<Cliente> lista =  
    db.Query<Cliente>("SELECT * FROM Cliente")  
    .ToList();
```

Classe para
retorno da query.

- É possível realizar pesquisas com **join**, **order by**, **limit** ou qualquer comando SQL.

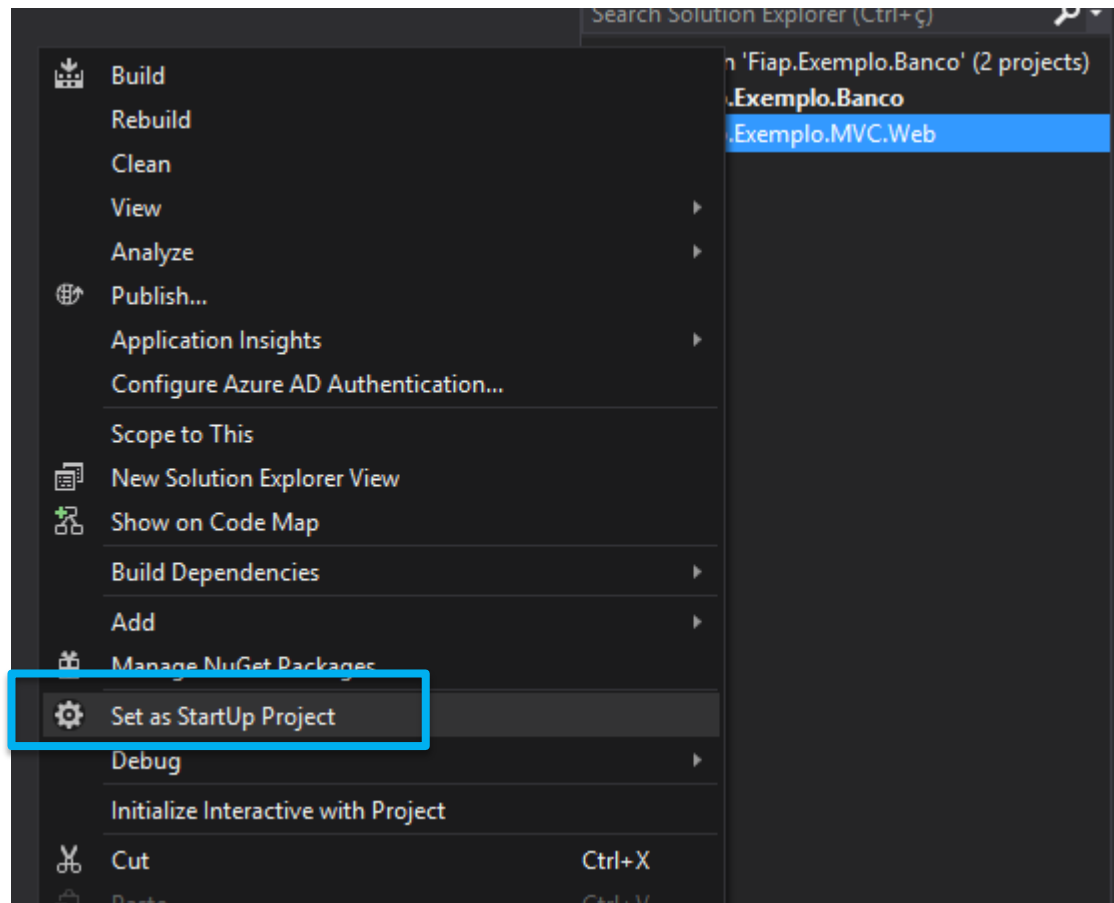
```
var sql = "SELECT * FROM Cliente WHERE nome LIKE @nome  
ORDER BY nome";
```

```
var lista = db.Query<Cliente>(sql, new { nome =  
"%" + nome + "%" }).ToList();
```

<https://www.w3schools.com/sql/>

SET AS STARTUP PROJECT

- Como temos dois projetos na Solução, precisamos marcar um projeto como o inicial (que será executado):



Copyright © 2018 - Prof. Me. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

“Descobri que quanto mais eu estudo, mais sorte eu pareço ter nas provas”