

# Acesso a Dados

ASP.NET possui várias opções para armazenamento e acesso a dados:

- Database Management Systems (DBMS)
- SQL Server
- ADO.NET
- ORM (Entity Framework)
- Web Services e APIs
- Files
- Storages, NoSQL, Messaging Services
- Etc...

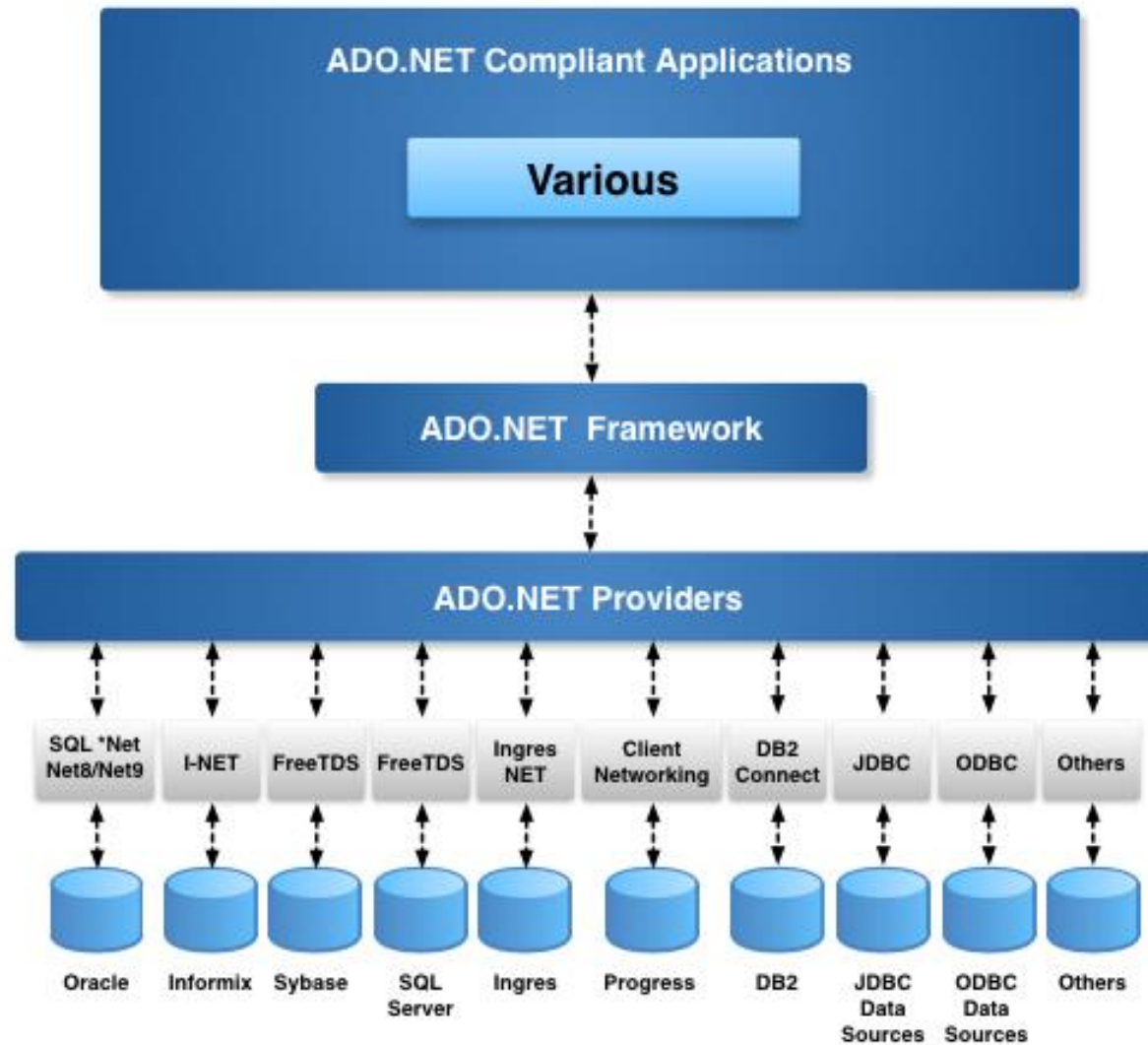


# ADO.NET

Tecnologia de acesso a dados sucessora da tecnologia **ADO**

- Permite acesso a diversas fontes de dados
  - Banco de dados
  - XML
  - Excel
- Executa a conexão através de Providers
- Providers permitem a conexão em servidores e estruturas diferentes.
- Exemplo: SqlServer, Oracle, Firebase, MySql, etc...

# ADO.NET



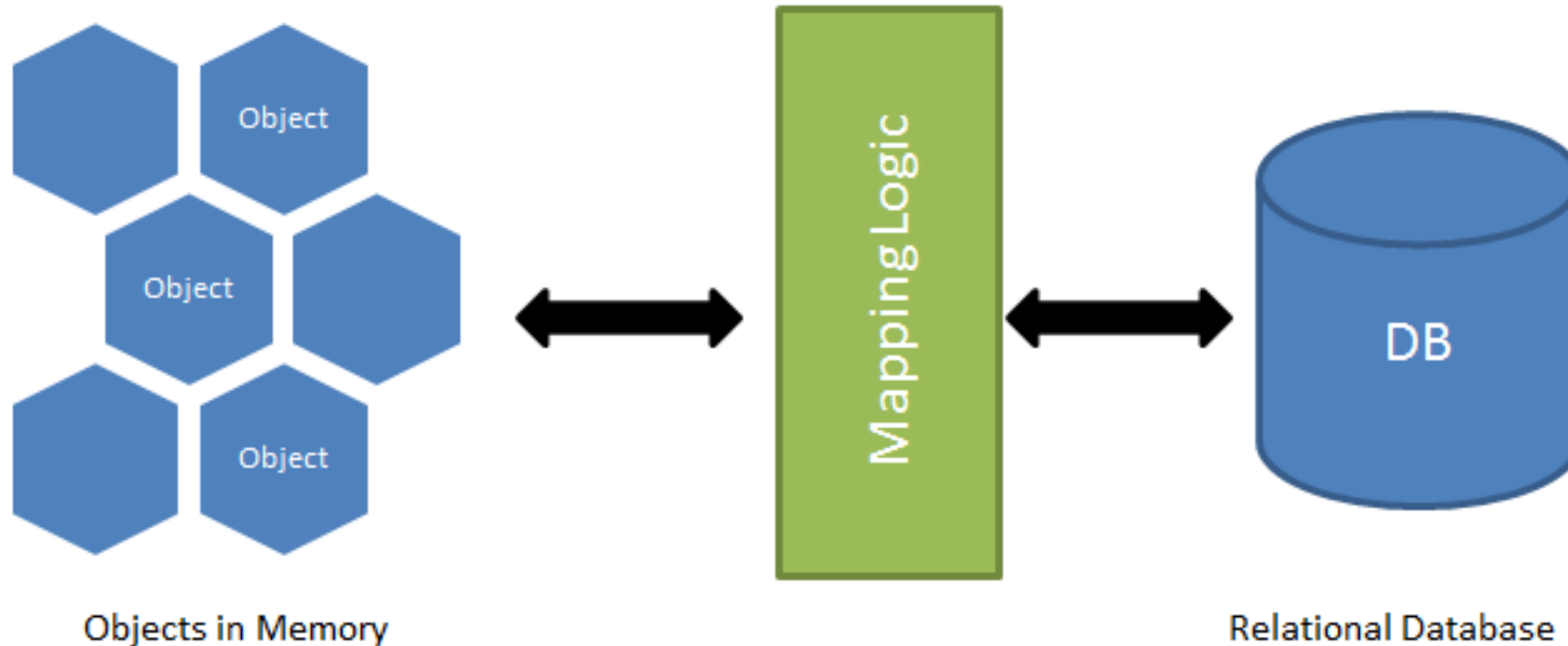
# O que é um ORM?

É uma arquitetura de Mapeamento Objeto Relacional (ORM)

- Abstração do modelo orientado a objetos
- Para o modelo de banco de dados relacional
- Existem diversos ORMs no mercado
- Normalmente abstraem totalmente as interações com o banco
- Tabelas -> Classes
- Colunas -> Propriedades
- Conversão de tipos de dados e manipulação das informações
- Independência do servidor de banco de dados

# O que é um ORM?

## O/R Mapping



# Entity Framework

- É o ORM da Microsoft
- A comunicação ainda acontece através de um Provider ADO.NET
- Permite a conexão a múltiplos bancos (SQL Server, MySql e Oracle)
- Abstrai totalmente o acesso a base de dados
- O “Contexto” do EF é o conector entre a base e a estrutura de classes
- As consultas são realizadas utilizando o LINQ
- LINQ é a linguagem de acesso e consulta a coleções do .NET

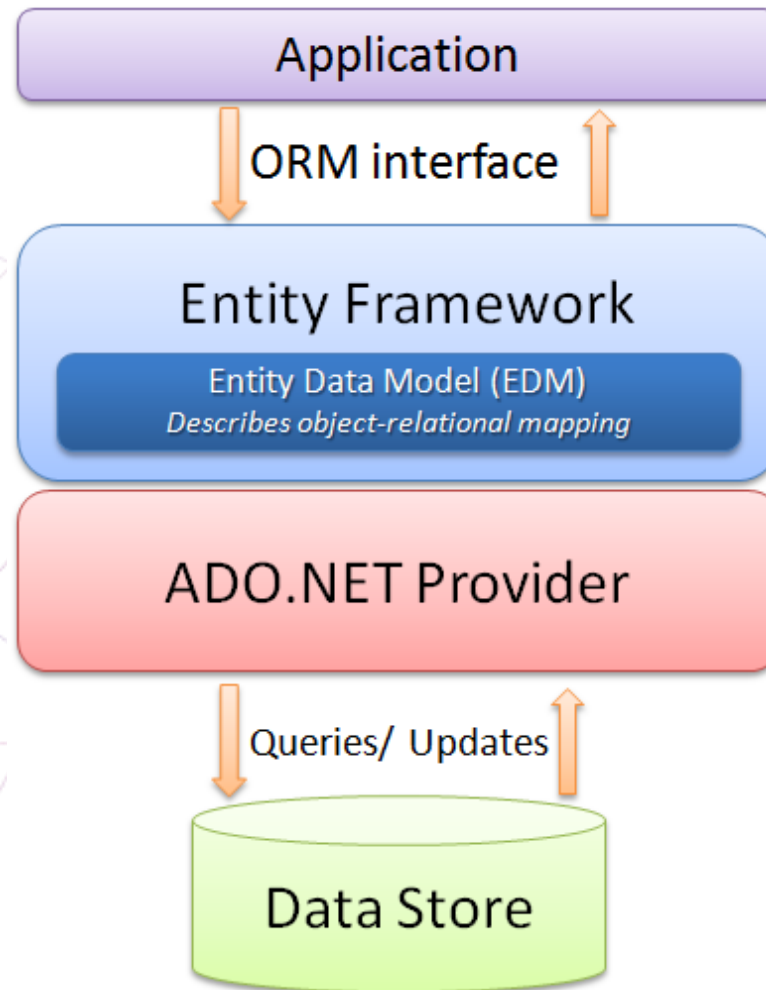
# Entity Framework

Alguns recursos do Entity Framework são:

- Lazy Loading
- Transações atômicas
- Estrutura de classes e interfaces facilita a testabilidade
- Mapeamento do banco para o código
- Codificação independente de base (Code First)
- Configuração simples através de convenções



# Entity Framework





# Vantagens do uso de ORM

1. Vantagens da implementação do Domain Model Pattern
2. Extrema redução de código
3. Alterações no modelo são realizadas em um único local
4. Queries geradas e otimizadas
5. Navegação entre objetos
6. Configuração de consulta e leitura configuráveis
7. Suporte a concorrência de dados
8. Gestão de cache de dados
9. Transação e isolamento
10. Gestão de índices

# Code First Conventions

São um conjunto de regras que o E.F. vai utilizar no mapeamento. Facilita a produtividade e padronização do código. Podem ser desativadas e alteradas.

- ***Typed Discovery***

- O contexto irá *buscar* todas as entidades mapeadas a uma classe já incluída e esta também fará parte do mapeamento. Para ignorar uma Classe será necessário utilizar um atributo (*NotMapped*).

- ***Primary Key Convention***

- Será automaticamente definido como PK a propriedade com o nome “ID” ou com o nome da Entidade seguida por “ID”. Exemplo “ClienteID”.

# Code First Conventions

- ***Relationship Convention***

- A construção do relacionamento no Diagrama de Classes também permitirá ao contexto entender os relacionamentos entre objetos no banco de dados. Sejam esses relacionamento *1 pra muitos* ou *muitos pra muitos*.

- ***Complex Types Convention***

- Ao criar um relacionamento de entidades o EF irá criar o relacionamento mesmo que a propriedade não tenha sido explicitamente identificada

- ***Pluralization Convention***

- Sem especificar o objeto da base de dados o EF irá adotar a nomenclatura no plural das entidades, sempre em inglês. Ex: *Car(classe) -> Cars(tabela)*

# Data Annotations

- Para o mecanismo de trabalho via *Code First* os data annotations são atributos que, ao especificados no modelo, configuram o mapeamento
- Descrevem a estrutura do modelo e seus relacionamentos
- Fácil leitura e manutenção
- Alguns atributos são:
  - [Key]
  - [Required]
  - [Table]
  - [NotMapped]
  - Etc...

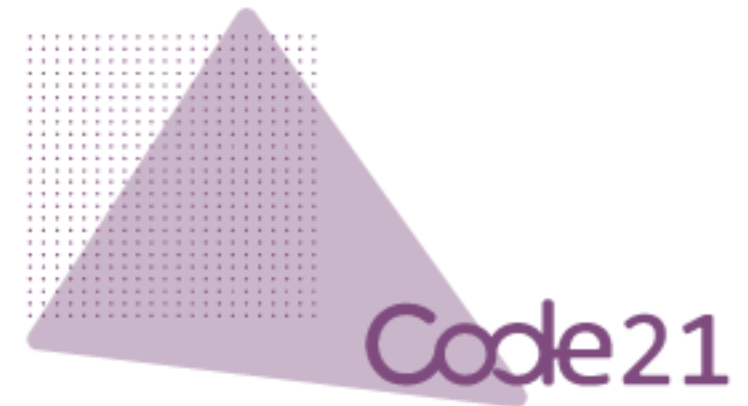


# Data Annotations

```
[Table("newTable")]  
public class Artist  
{  
    [Key]  
    public int ArtistID { get; set; }  
  
    [Required()]  
    [StringLength(100,MinimumLength=2)]  
    public string ArtistName { get; set; }  
}
```

# Exercício 15

Configurações Iniciais do Entity Framework

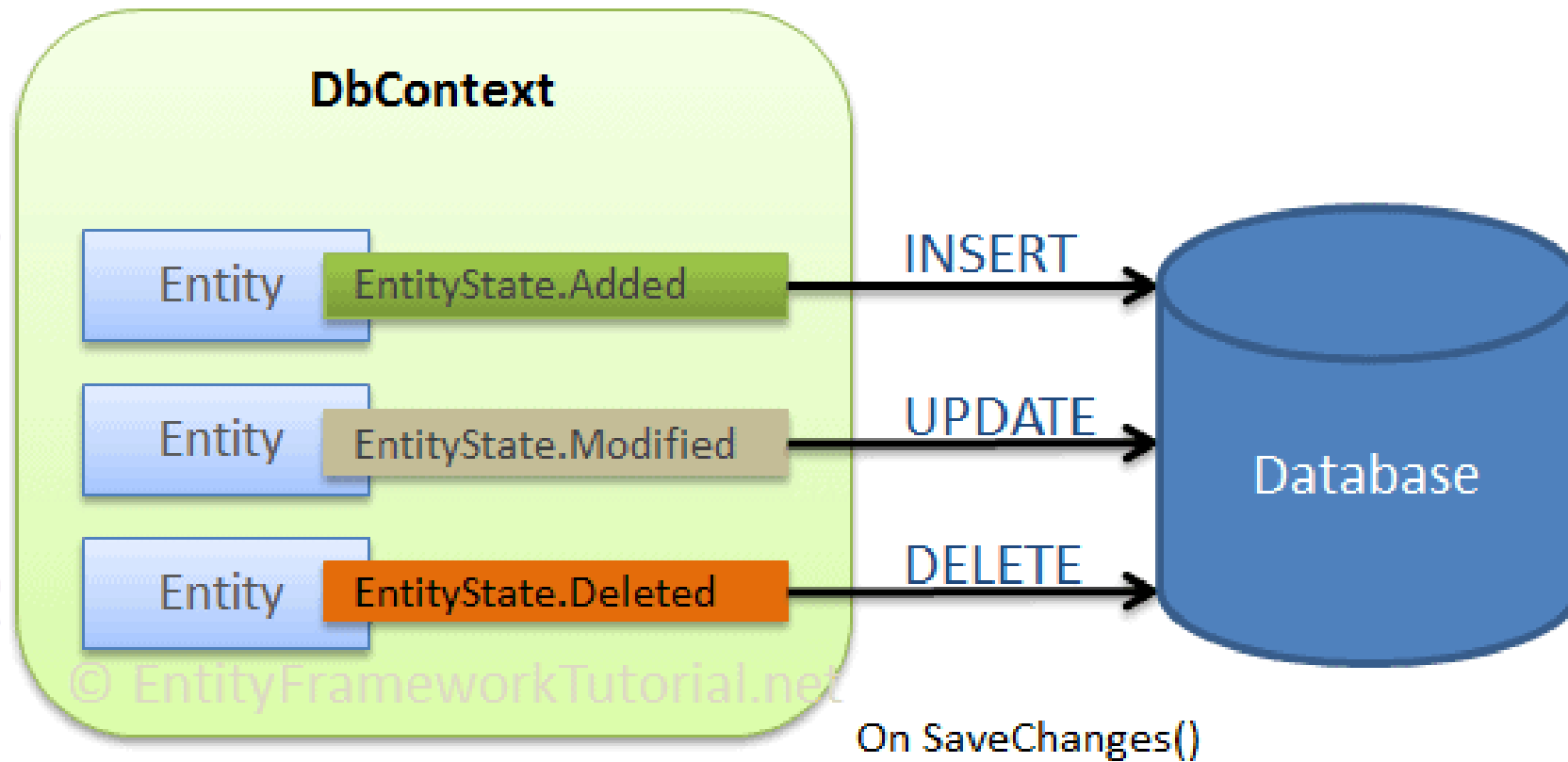


# Alterando Dados usando o E.F.

Além da leitura dos dados o Entity Framework também gera os comandos para as alterações, inclusões e exclusões

- As entidades alteradas ou vinculadas ao contexto não são enviadas diretamente para o banco de dados
- Todo o processo de alteração do E.F. trabalha de forma atômica
- Em caso de erro automaticamente ele retornar as alterações
- Explicitamente precisamos acionar a sincronização com o banco
- É possível modificar o “Estado” de um objeto vinculado ao contexto

# Alterando Dados usando o E.F.





# Alterando Dados usando o E.F.

```
public IActionResult Insert()
{
    var filme = new Filme { Titulo = "A Outra Face",
        Sinopse = "Teste",
        EmCartaz = false };
    catalogoContext.Filmes.Add(filme);
    catalogoContext.SaveChanges();

    return View();
}
```

```
public IActionResult Update()
{
    var filme = catalogoContext.Filmes.First();
    filme.Titulo = "Avatar";
    filme.EmCartaz = false;
    catalogoContext.SaveChanges();

    return View();
}
```

```
public IActionResult Delete()
{
    var filme = catalogoContext.Filmes.First();
    catalogoContext.Filmes.Remove(filme);
    catalogoContext.SaveChanges();

    return View();
}
```

# Exercício 16

Incluindo, Alterando e Excluindo Registros

