

# Entity Framework

## Tecnología ADO.NET

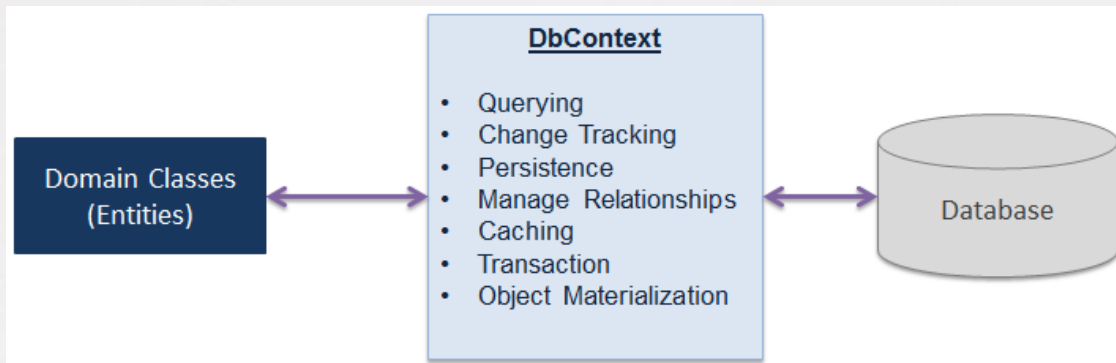
Admite el desarrollo de aplicaciones de software orientadas a datos.

## Defición:



ORM que interactúa con datos almacenados en base de datos relacionales con objetos .NET fuertemente tipados que representan el dominio de la aplicación.

# ¿Como funciona?



Se deberá **crear o importar** un modelo que asigna las entidades y relaciones que se definen en el modelo de la base de datos.

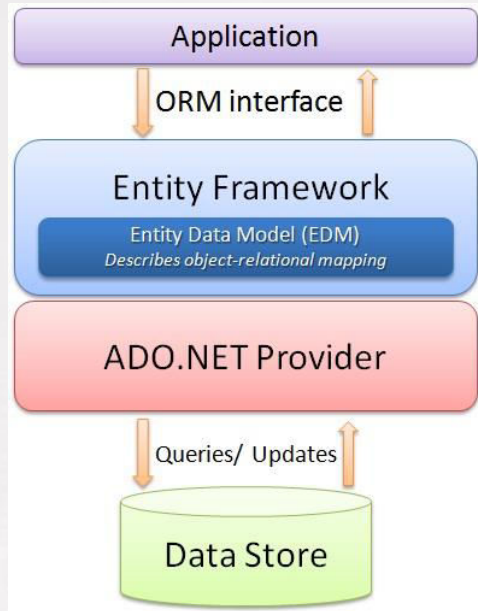
La clase principal interactúa con **System.Data.Entity.DbContext** denominado clase de contexto.

-Ejecutar consultas

-Almacenar los cambios del objeto en la base de datos

-Enlazar objetos en memoria a los controles de interfaz de usuario

## Compuesto por:



Los modelos pueden generarse a partir de bases de datos existentes y luego editarse manualmente, o bien se pueden crear desde cero y luego usarse para generar nuevas bases de datos.

# Implementa muchas características de O/RM populares:

- Seguimiento de cambios automáticos.
- Traducción de consultas con LINQ.
- Carga diligente, diferida y explícita.
- Diseñador visual para crear modelos de entidad.
- Capacidades de asignación enriquecidas que incluyen compatibilidad con:
  - Relaciones de uno a uno, de uno a varios y entre varios
  - Herencia (tabla por jerarquía, tabla por tipo y tabla por clase concreta)
  - Tipos complejos
  - Procedimientos almacenados

# Operaciones sobre la base de datos

Abrir el contexto

```
public void UseProducts()
{
    using (var context = new ProductContext())
    {
        // Perform data access using the context
    }
}
```

# Operaciones sobre la base de datos

## Consulta de datos con LINQ

```
using (var context = new BloggingContext())
{
    // Query for all blogs with names starting with B
    var blogs = from b in context.Blogs
                where b.Name.StartsWith("B")
                select b;

    // Query for the Blog named ADO.NET Blog
    var blog = context.Blogs
                  .Where(b => b.Name == "ADO.NET Blog")
                  .FirstOrDefault();
}
```

# Operaciones sobre la base de datos

## Consulta de datos SIN Formato

```
using (var context = new BloggingContext())
{
    var blogs = context.Blogs.SqlQuery("SELECT * FROM dbo.Blogs").ToList();
}
```



# Operaciones sobre la base de datos

Agregar entidad al contexto

```
using (var context = new BloggingContext())
{
    var blog = new Blog { Name = "ADO.NET Blog" };
    context.Blogs.Add(blog);
    context.SaveChanges();
}
```

# Operaciones sobre la base de datos

Modificar entidad del contexto

```
var existingBlog = new Blog { BlogId = 1, Name = "ADO.NET Blog" };  
  
using (var context = new BloggingContext())  
{  
    context.Entry(existingBlog).State = EntityState.Modified;  
  
    // Do some more work...  
  
    context.SaveChanges();  
}
```