Semana 4

Curso: Desarrollo de aplicaciones web

Ciclo: 2023-1

Unidad II: Acceso a datos, Backend, Web API y Entity Framework

UNIVERSIDAD esan

MBA, Luis Chang - lchangu@esan.edu.pe

UNIDAD DE APRENDIZAJE II: ACCESO A DATOS, BACKEND, WEB API Y ENTITY FRAMEWORK

RESULTADOS DE APRENDIZAJE:

- Comprende el ecosistema .NET.
- Entiende la diferencia sobre .NET Framework y .NET Core.
- Comprende sobre la comunicación entre un proyecto web y la base de datos.
- Entiende sobre la importancia de utilizar un ORM.
- Comprende y desarrolla los enfoques de Entity Framework Core.
- Manipula objetos de base de datos mediante Entity Framework Core.
- Entiende la importancia de la programación orientada a objetos y el principio SOLID para la construcción de aplicaciones web empresariales.
- Construye aplicaciones backend con ASP.NET Core Web API.
- Comprende la importancia de los enfoques de Entity Framework Core (Code First y Database First).
- Construye aplicaciones sobre arquitectura limpia (Clean Arquitecture).
- Entiende la importancia de documentar las APIs.
- Entiende la importancia de un middleware en una aplicación web.
- Entiende el uso de procedimientos almacenados en una aplicación web.
- La capacidad de aplicar conocimientos de matemáticas, ciencias e ingeniería en la solución de problemas complejos de ingeniería.
- La capacidad de identificar, formular, buscar información y analizar problemas complejos de ingeniería para llegar a conclusiones fundamentadas usando principios básicos de matemáticas, ciencias naturales y ciencias de la ingeniería.
- La capacidad de comunicarse eficazmente, mediante la comprensión y redacción de informes y documentación de diseño, la realización de exposiciones, y la transmisión y recepción de instrucciones claras.
- La capacidad de comprender y evaluar el impacto de las soluciones a problemas complejos de ingeniería en un contexto global, económico, ambiental y social.
- La capacidad de aplicar el razonamiento informado mediante el conocimiento contextual para evaluar cuestiones sociales, de salud, de seguridad, legales y culturales y las consecuentes responsabilidades relevantes para la práctica profesional de la ingeniería.

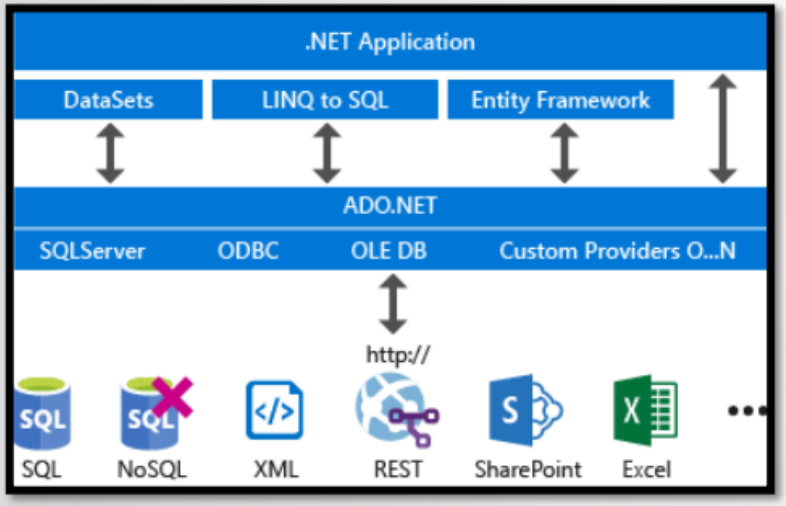
CONTENIDO:

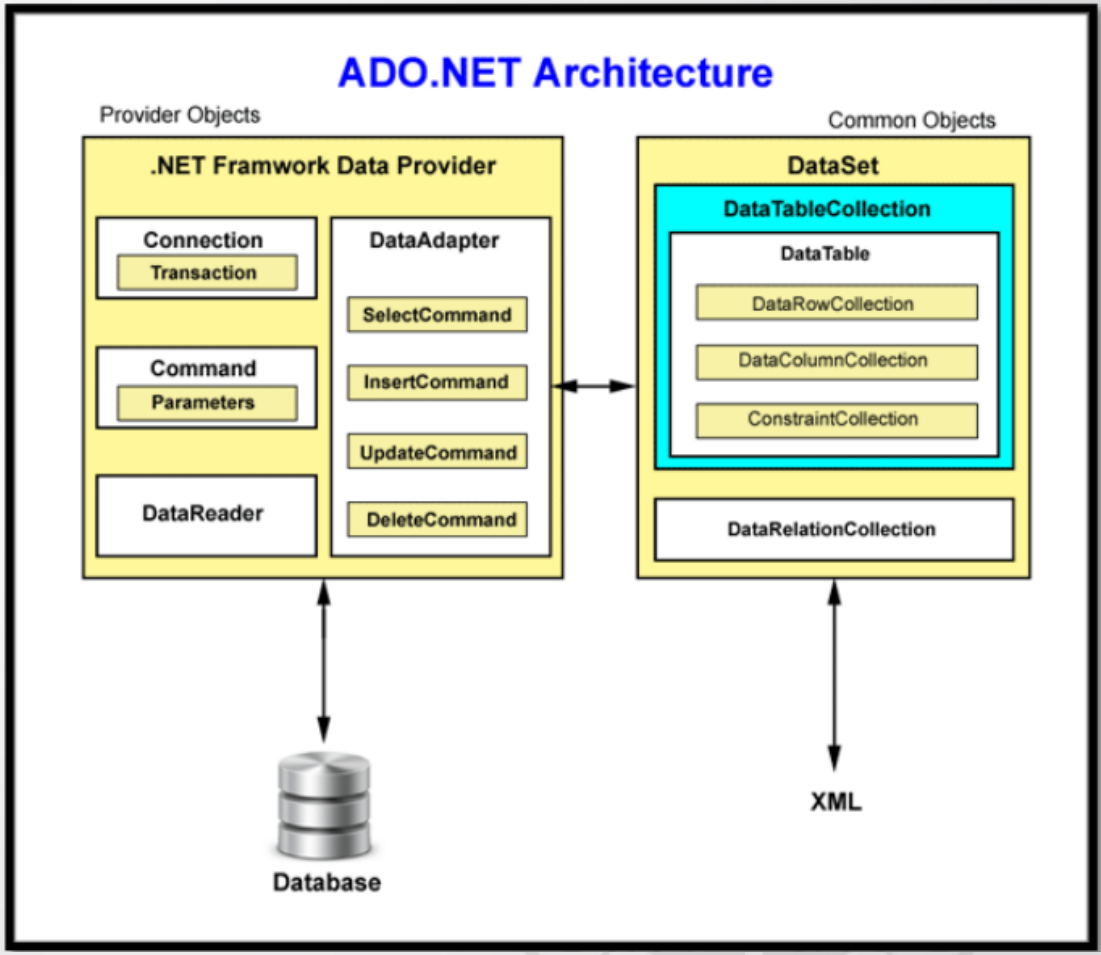
2.1. Comprende sobre .NET
2.2. ¿Qué es un ORM?
2.3. Principios SOLID
2.4. Patrones de diseño
2.5. Entity Framework Core – Enfoques.
2.6. Expresiones lambda y LINQ.

Desarrollo de aplicaciones web – Semana 4

COMPRENDE SOBRE .NET Y ADO NET

ADO.NET


.NET Application
DataSets
LINQ to SQL
Entity Framework
ADO.NET
SQLServer
ODBC
OLE DB
Custom Providers O...N
http://
SQL
NoSQL
XML
REST
SharePoint
Excel

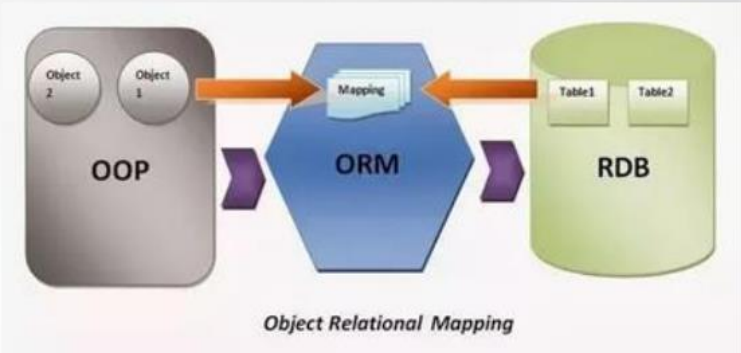

ADO.NET Architecture
Provider Objects
Common Objects
.NET Framwork Data Provider
Connection
Transaction
Command
Parameters
DataReader
DataAdapter
SelectCommand
InsertCommand
UpdateCommand
DeleteCommand
DataSet
DataTableCollection
DataTable
DataRowCollection
DataColumnCollection
ConstraintCollection
DataRelationCollection
Database
XML

Desarrollo de aplicaciones web – Semana 4

¿QUÉ ES UN ORM?

Object Relational Mapping

- Es un mecanismo que permite abordar, acceder y manipular objetos (Mapeo) sin tener que considerar cómo los objetos se relacionan con sus fuentes de datos.
- Permite crear una base de datos orientada a objetos “virtual” (Persistencia), sobre una base de datos relacional.
- Se considera una capa intermedia entre la base de datos y los objetos de la base de datos.
- ¿Por qué trabajar con un ORM?: Abstracción de la base de datos, POO, seguridad.



ENTITY FRAMEWORK

LINQ

Microsoft .NET

HIBERNATE

Consultemos a la base de datos..

Consultemos a la base de datos sin un ORM...

Los primeros 5 registros de la tabla Clientes

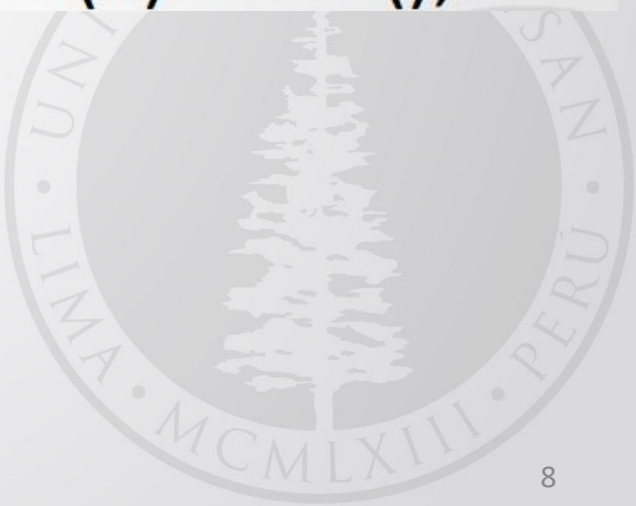
ORACLE®	SELECT * FROM Clientes WHERE rownum<=5;
Microsoft SQL Server	SELECT TOP 5 * FROM Clientes
MySQL	SELECT * FROM Clientes LIMIT 5
IBM DB2	SELECT * FROM Clientes fetch first 5 rows only

Object Relational Mapping

Consultemos a la base de datos con un ORM...

Los primeros 5 registros de la tabla Clientes.

```
var listado = data.Clientes.Take(5).ToList();
```



Desarrollo de aplicaciones web – Semana 4

PRINCIPIOS SOLID

Programación Orientada a Objetos: Pilares

```
public class Customer
{
  // Fields, properties, methods and events go here…
}

Customer object1 = new Customer();

public class Manager : Employee

{
    // Employee fields, properties, methods and events
are inherited
    // New Manager fields, properties, methods and
events go here…
}
```

Pilares de la POO

- Abstracción
- Encapsulamiento.
- Herencia.
- Polimorfismo.

Principios SOLID

- **S**: Single responsibility principle o Principio de responsabilidad única. Keyword: "**Decoupled**"
- **O**: Open/closed principle o Principio de abierto/cerrado. Keyword: "**Abstraction**"
- **L**: Liskov substitution principle o Principio de sustitución de Liskov. Keyword: "**Replaceable**"
- **I:** Interface segregation principle o Principio de segregación de la interfaz. Keyword: "**Segregate Interfaces**".
- **D:** Dependency inversion principle o Principio de inversión de dependencia. Keyword: "**Dependency**".

Fuente: Enmilocalfunciona.io

Desarrollo de aplicaciones web – Semana 4

PATRONES DE DISEÑO

Patrones de diseño

- Los patrones de diseño son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. Son como planos prefabricados que se pueden personalizar para resolver un problema de diseño recurrente en tu código.

¿En qué consiste el patrón?

•El **propósito** del patrón explica brevemente el problema y la solución.
•La **motivación** explica en más detalle el problema y la solución que brinda el patrón.
•La **estructura** de las clases muestra cada una de las partes del patrón y el modo en que se relacionan.
•El **ejemplo de código** en uno de los lenguajes de programación populares facilita la asimilación de la idea que se esconde tras el patrón.

Desarrollo de aplicaciones web – Semana 4

ENTITY FRAMEWORK CORE - ENFOQUES

Entity Framework Core

- Entity Framework (EF) Core es una versión ligera, extensible, de código abierto y multiplataforma de la popular tecnología de acceso a datos de Entity Framework.
- EF Core puede servir como un mapeador relacional de objetos (ORM), que:
  - Permite a los desarrolladores de .NET trabajar con una base de datos utilizando objetos .NET.
  - Elimina la necesidad de la mayor parte del código de acceso a datos que normalmente es necesario escribir.
- EF Core admite muchos motores de base de datos.
- EF Core Admite
  - Acceder mediante "**Modelos**".
  - Consultas utilizando **LINQ**.
- **CRUD** a las tablas de base de datos.

Fuente: Página Web Oficial - Microsoft Entity Framework Core

Desarrollo de aplicaciones web – Semana 4

EXPRESIONES LAMBDA Y LINQ

Query Expressions & Lambda Expressions

```
// Lista de Estudiantes
IList<Student> studentList = new List<Student>() {
new Student() { StudentID = 1, StudentName = "Jose", Age = 13} ,
new Student() { StudentID = 2, StudentName = "Carlos", Age = 21 } ,
new Student() { StudentID = 3, StudentName = "Bryan", Age = 18 } ,
new Student() { StudentID = 4, StudentName = "Daniel" , Age = 20} ,
new Student() { StudentID = 5, StudentName = "Eduardo" , Age = 15 }
};
// Método 1: LINQ Query Expressions para consultar a los estudiantes
adolescentes
var teenAgerStudent = from s in studentList
where s.Age > 12 && s.Age < 18
select s;
// Método 2: Lambda expression para consultar a los estudiantes
adolescentes
var teenAgerStudent1 = studentList.Where(x=>x.Age > 12 && x.Age <
18).Select(x=>x).ToList();
```

Referencias

- Price, M. J. (2022). *C# 11 and .NET 7 – Modern Cross-Platform Development Fundamentals: Start building websites and services with ASP.NET Core 7, Blazor, and EF Core 7, 7th Edition.* Packt Publishing. Chapter 1 Pages 10-24 Chapter 10 425-438 Chapter 10 433-486 Chapter 11 Pages 489-517