

Universidad Tecnológica de Tula-Tepeji.

Tecnologías de la información y comunicación área de
sistemas informáticos.

Desarrollo de Aplicaciones III

Docente:
Erik Guerrero Bravo.

Alumno:
Fazur Alejandro Estrada Ponce.

Cuatrimestre y Grupo:
5TIC-G3

Contenido

Introducción.	3
¿Qué es un ORM?	3
Entity Framework.	3
Características.....	4
Database First	5
Ventajas o beneficios de Database First.....	5
Problemas o inconvenientes de Database First.....	5
Model First	6
Ventajas o beneficios de Model First	6
Problemas o inconvenientes de Model First.....	7
Code First	7
Ventajas o beneficios de Code First.....	7
Problemas o inconvenientes de Code First	8
Conclusiones.	8
Bibliografía	8

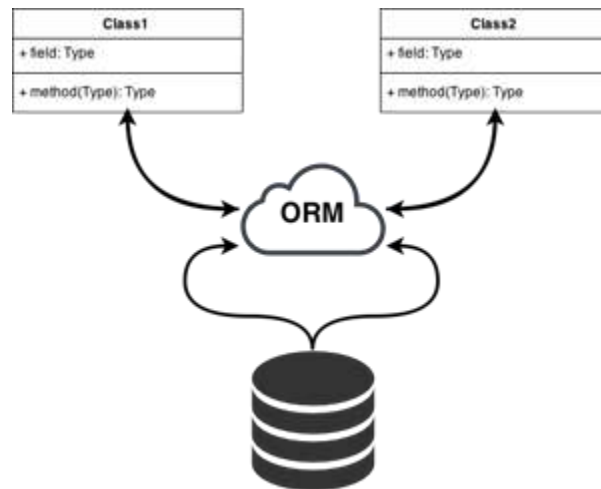
Introducción.

En el presente documento se explicara la tecnología de Entity Framework.

Entity Framework es un Asignador Relacional de Objetos (ORM), es una herramienta que simplifica el mapeo entre objetos en su software a las tablas y columnas de una base de datos relacional.

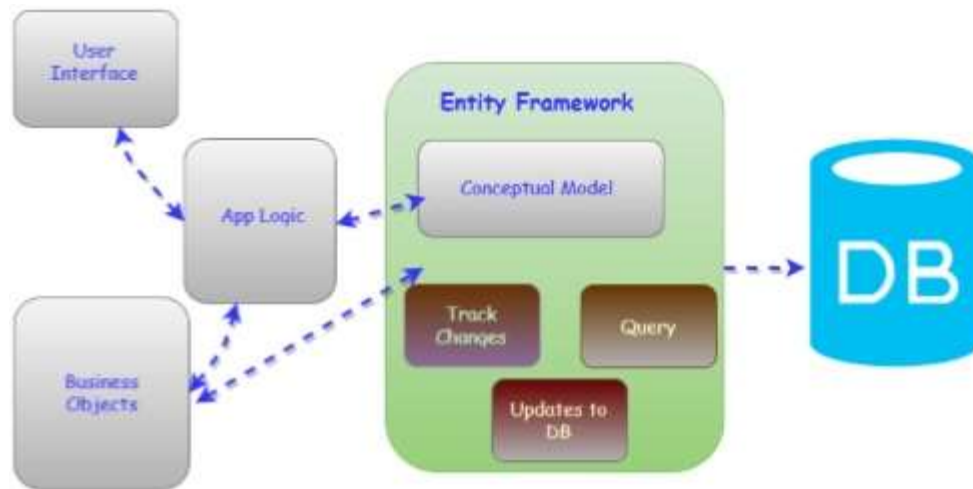
¿Qué es un ORM?

Es una técnica de programación para convertir los datos de los objetos en un formato correcto para poder guardar la información en una base de datos.



Entity Framework.

Entity Framework es un ORM y los ORM tienen como objetivo aumentar la productividad del desarrollador al reducir la tarea redundante de conservar los datos utilizados en las aplicaciones. (tutorialspoint, s.f.)



Características

- Entity Framework es una herramienta de Microsoft.
- Entity Framework se está desarrollando como un producto de código abierto.
- Entity Framework ya no está vinculado ni depende del ciclo de lanzamiento de .NET.
- Funciona con cualquier base de datos relacional con un proveedor de Entity Framework válido.
- Generación de comandos SQL desde LINQ a Entidades.
- Entity Framework creará consultas parametrizadas.
- Hace un seguimiento de los cambios en los objetos en memoria.
- Permite insertar, actualizar y eliminar la generación de comandos.
- Trabaja con un modelo visual o con tus propias clases.
- Entity Framework ha almacenado soporte de procedimientos.

Database First

En este modo se parte de una base de datos pre-existente con la que queremos trabajar. Se debe tener base de datos ya diseñada y probablemente con datos.

Ventajas o beneficios de Database First

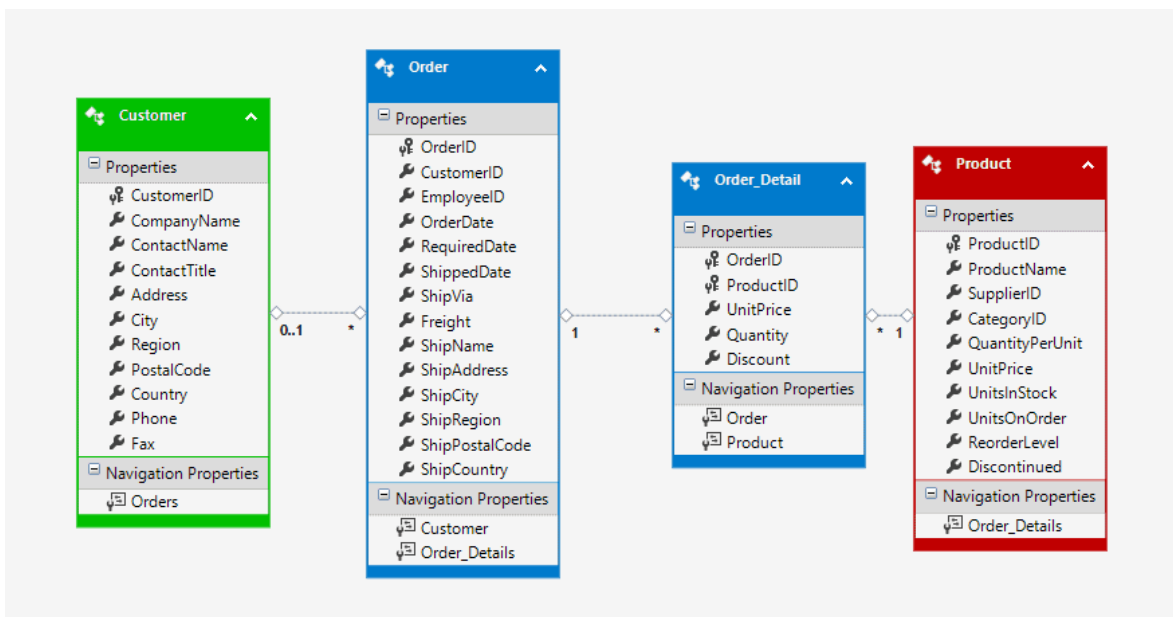
- Son muy sencillas de implementar, pues casi todo va en automático y parten de la base de datos.
- Si tienes costumbre de trabajar con bases de datos, llevando a cabo primero el diseño E-R, y te sientes menos cómodo con código C#, entonces estarás como en casa.
- Se lleva bien con proyectos de datos grandes, con muchas tablas, convenientemente repartidas en varios modelos.
- Es muy complicado que pierdas datos al hacer modificaciones en el modelo, ya que las harás en la base de datos, no en el código de la aplicación.

Problemas o inconvenientes de Database First

- El código resultante para los modelos tiene infinidad de código auto-generado sobre el que tenemos poco control y que puede acabar pesando mucho.
- Si necesitas personalizar las clases o su comportamiento debes extenderlas con clases parciales o bien tocando las plantillas T4.
- Cada vez que haces cambios en la base de datos se debe regenerar el archivo EDMX que la representa, lo cual puede llevar bastante tiempo dependiendo de lo grande que sea.

Model First

En el model first Describes visualmente el modelo de objetos que quieres crear, con las entidades, sus propiedades, relaciones.



Ventajas o beneficios de Model First

- El diseñador visual facilita enormemente el diseño del modelo de datos.
- Te evita escribir código como definir la base de datos (aunque puedes generar el modelo inicial de una base de datos preexistente si quieres).
- Es muy productivo en proyectos pequeños, pues te olvidas de definir clases y de tocar bases de datos.

Problemas o inconvenientes de Model First

- Pierdes el control tanto de la base de datos como de las clases generadas.
- Si quieres extender las clases, como antes, debes tocar las plantillas T4 o definir clases parciales.
- Por defecto los cambios que hagas en el modelo no generan scripts SQL incrementales, sino completos.

Code First

Definimos nuestras clases mediante código.

El programador define su modelo usando clases normales. Relaciona unas con otras simplemente haciendo referencia entre ellas en propiedades.

Ventajas o beneficios de Code First

- Máximo control sobre el código de tu modelo de datos en C#, ya que son clases que construyes desde cero.
- Te ofrece control sobre las colecciones de objetos.
- La estructura de la base de datos es muy fácil de mantener bajo control.

Problemas o inconvenientes de Code First

- Debes dominar bastante más EF que con los enfoques anteriores.
- Cualquier cosa que necesites persistir u obtener de tu base de datos la tienes que implementar en código C# y LINQ. La base de datos que se genera por detrás no la puedes tocar.
- Si tu base de datos es muy grande y con muchas tablas, la gestión de la base de datos se puede hacer difícil.

Conclusiones.

Entity Framework nos ofrece variadas técnicas de trabajo para la conexión entre la base de datos y nuestra aplicación dándole así la comodidad al desarrollador de escoger la que más se adapte a sus necesidades y trabajo, facilitándole su desarrollo del mismo.

Bibliografía

(s.f.). Obtenido de tutorialspoint:

https://www.tutorialspoint.com/entity_framework/entity_framework_overview.htm