

Mapeo Objeto Relacional

Sigfredo Aponte, Alonso Andia, Gabriela Atahuachi y Yhónn Condori

September 19, 2019

Abstract

En este artículo aprenderemos sobre el concepto y sus herramientas de mapeo objeto-relacional que nos permitirá crear una capa de acceso a los datos de una base de datos relacional, de tal forma que las tablas se transforman en clases y las filas de las tablas en objetos.

Palabras clave: ORM, base de datos, herramientas, clases, objetos.

Abstract

In this article we learn about concept and tools of Object-Relational mapping ORM that we will allow us to create an access layer to the data of a relational database, in such a way that the tables are transformed into classes and the rows of the table into objects.

Keywords: ORM, database, tools, classes, objects.

I. INTRODUCCION

EL mapeo relacional de objetos es una técnica que le permite consultar y manipular datos de una base de datos utilizando un paradigma orientado a objetos. Cuando se habla de ORM, la mayoría de personas se refieren a una biblioteca que implementa la técnica de mapeo relacional de objetos, de ahí la frase “un ORM”. Una biblioteca ORM es una biblioteca completamente ordinaria escrita en su idioma de elección que encapsula el código necesario para manipular los datos, por lo que ya no usa SQL; interactúa directamente con un objeto en el mismo idioma que está utilizando.

jetos sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (esencialmente la herencia y el polimorfismo)[1].

Las bases de datos relacionales solo permiten guardar tipos de datos primitivos (enteros, cadenas de texto, etc.) por lo que no se pueden guardar de forma directa los objetos de la aplicación en las tablas, sino que estos se deben de convertir antes en registros, que por lo general afectan a varias tablas. En el momento de volver a recuperar los datos, hay que hacer el proceso contrario, se deben convertir los registros en objetos. Es entonces cuando ORM cobra importancia, ya que se encarga de forma automática de convertir los objetos en registros y viceversa, simulando así tener una base de datos orientada a objetos [2].

II. MARCO TEÓRICO

i. Mapeo Objeto / Relacional

El mapeo objeto-relacional es una técnica de programación para convertir datos del sistema de tipos utilizado en un lenguaje de programación orientado a objetos al utilizado en una base de datos relacional. En la práctica esto crea una base de datos virtual orientada a ob-

ii. Estrategias de mapeo

ii.1 Mapeo de herencia

Las tres estrategias más comunes para representar un árbol de herencia de clases en un modelo relacional (en un RDBMS) son:

- Mapeo filtrado

- Mapeo horizontal
- Mapeo vertical
- Mapeo tabla genérica

ii.2 Mapeo de relaciones

- Relación Uno-Uno
- Relación Uno-Mucho
- Relación Mucho-Mucho

ii.3 Patrones de Diseño más conocidos

- Patrón Repository
- Patrón Active Record

iii. Ventajas del uso de ORM

El utilizar el ORM nos puede proporcionar un grupo importante de ventajas:

iii.1 Rapidez en el desarrollo

La mayoría de las herramientas actuales permiten la creación del modelo por medio del esquema de la base de datos, leyendo el esquema, nos crea el modelo adecuado. Una vez que se tiene el modelo creado, el programador irá mucho más rápido gracias a la metodología totalmente orientada a objetos, olvidándose así de las consultas a la base de datos[3].

iii.2 Abstracción de la base de datos

Al utilizar un sistema ORM, lo que conseguimos es separarnos totalmente del sistema de base de datos que utilizemos y así, si en un futuro debemos de cambiar de motor de bases de datos, tendremos la seguridad de que este cambio no nos afectará a nuestro sistema, siendo el cambio mas sencillo. Todo sistema ORM debe de generar de forma automática las consultas a la base de datos para convertir los registros en objetos (y viceversa) y éstas deben poder adaptarse a los distintos proveedores (MySQL, Oracle, PostgreSQL, etc...), esto permitirá a los desarrolladores tener una preocupación menos a la hora de comenzar un proyecto, ya que tienen la seguridad de que un cambio drástico en el proveedor de base de

datos, no impactará para nada en el tiempo de vida del proyecto[3].

iii.3 Reutilización

Nos permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas[3].

iii.4 Seguridad

Los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser los SQL injections. Inyección SQL es un método de infiltración de código intruso. Se dice que existe o se produjo una inyección SQL cuando, de alguna manera, se inserta o "inyecta" código SQL invasor dentro del código SQL programado, a fin de alterar el funcionamiento normal del programa y lograr así que se ejecute la porción de código "invasor", incrustado en la base de datos. Este tipo de intrusión normalmente es de carácter malicioso[3].

iii.5 Mantenimiento del código

Nos facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos[3].

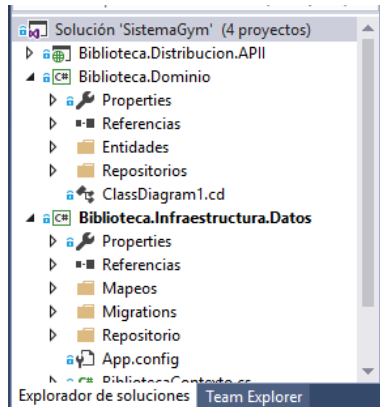
iii.6 Lenguaje propio para realizar las consultas

Estos sistemas de mapeo traen su propio lenguaje para hacer las consultas, lo que hace que los usuarios dejen de utilizar las sentencias SQL para que pasen a utilizar el lenguaje propio de cada herramienta: Otra característica importante (de hecho seguramente la que más) es la exposición de clases y métodos que nos dan las herramientas ORM para poder extraer los datos de la forma que necesitemos (filtros, ordenaciones, agrupaciones)[3].

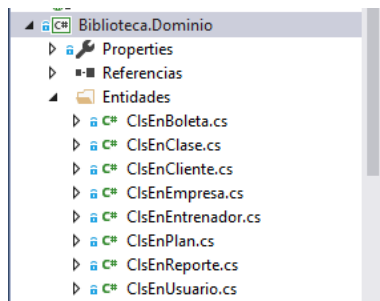
III. ANALISIS

En nuestro ejemplo usaremos la herramienta de Entity Framework.

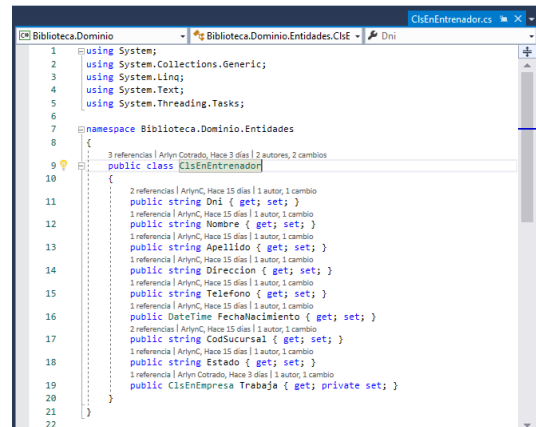
- En nuestra solución SistemaGYM, tenemos dos proyectos que necesitamos para el mapeo.



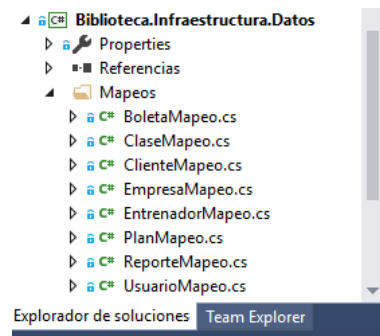
- En el proyecto Biblioteca.Dominio, donde se encuentra la capa de entidades.

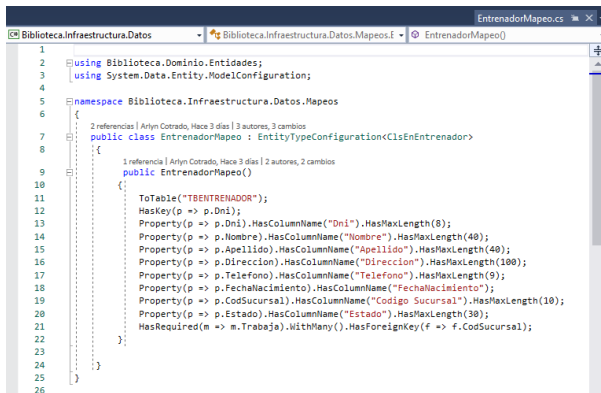


- En cada clase se especifican los atributos y sus relaciones.

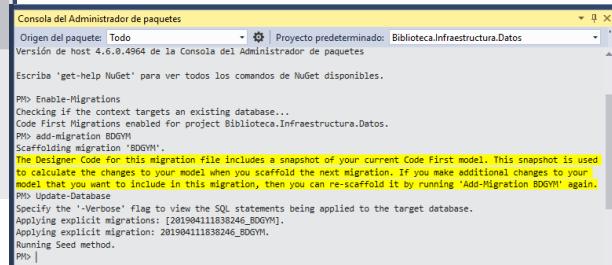


- En el proyecto Biblioteca.Infraestructura.Datos, donde se encuentran las tablas, relaciones y llaves.

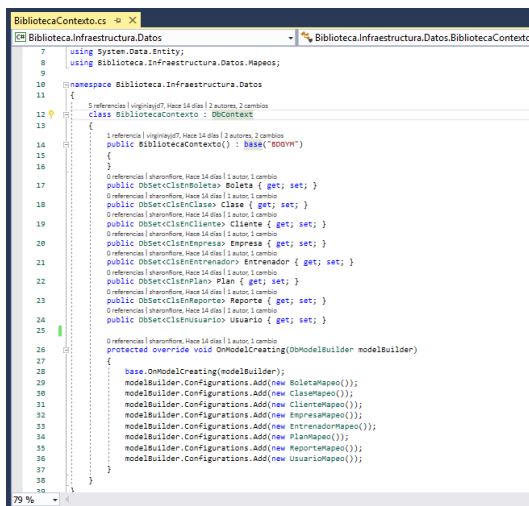




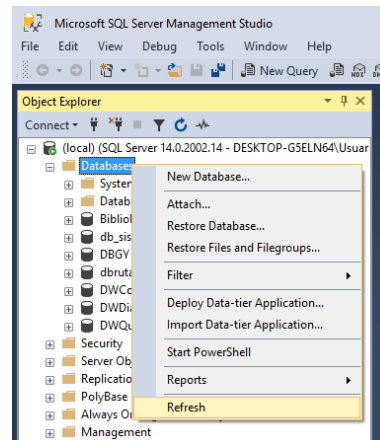
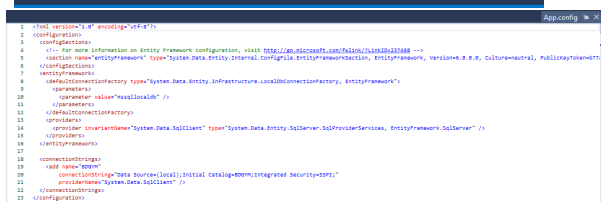
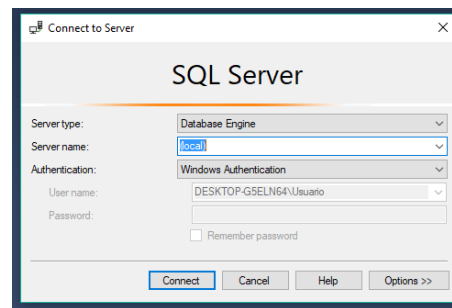
- Abrimos la Consola del Administrador de paquetes. Ingresamos los siguientes comandos: Enable-Migrations, add-migration, Update-Database.

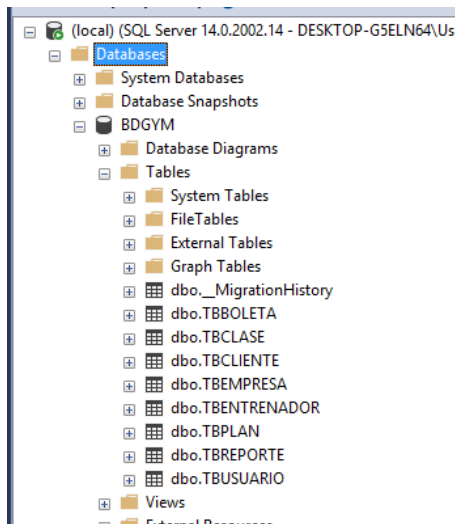


- Creamos el contexto de la base de datos.



- Por ultimo, vemos la base de datos, para ello entramos a SQL Server.





IV. CONCLUSIONES

ORM resuelve el desajuste del código objeto y la base de datos relacional con tres enfoques: de abajo hacia arriba, de arriba hacia abajo y se encuentran en el medio. Cada enfoque tiene su parte de beneficios y desventajas. Al seleccionar la mejor solución de software, los desarrolladores deben comprender completamente el entorno y los requisitos de diseño.

REFERENCES

- [1] Osmel Yanes, Hansel Gracia (2011). Mapeo Objeto / Relacional (ORM)
- [2] Iván Guardado (2010). Introducción a Object-Relational Mapping (ORM)
- [3] Fabiola Becerra, Elizabeth Hidalgo-Gato, Jesús Pérez (2009). OBJECT/RELATIONAL MAPPING
- [4] Ambler, SW (2010). Mapping objects to relational databases: O/R mapping in detail

