Introducción a C# y ASP.NET MVC

Capítulo 2: Acceso a datos .NET

Sesión 2: Entity Framework

Al término de esta sesion, Ud. podrá:

- Conocer Entity Framework
- Usar Entity Framework para el acceso a datos
- Insertar datos usando Entity Framework

Sesión 2: Entity Framework

- Introducción a ADO.NET EF
- EDM: Modelo de datos
- Entity Designer
- EDM: Estructura
- ObjectContext y ObjectStateManager
- Leyendo y Editando datos con EF

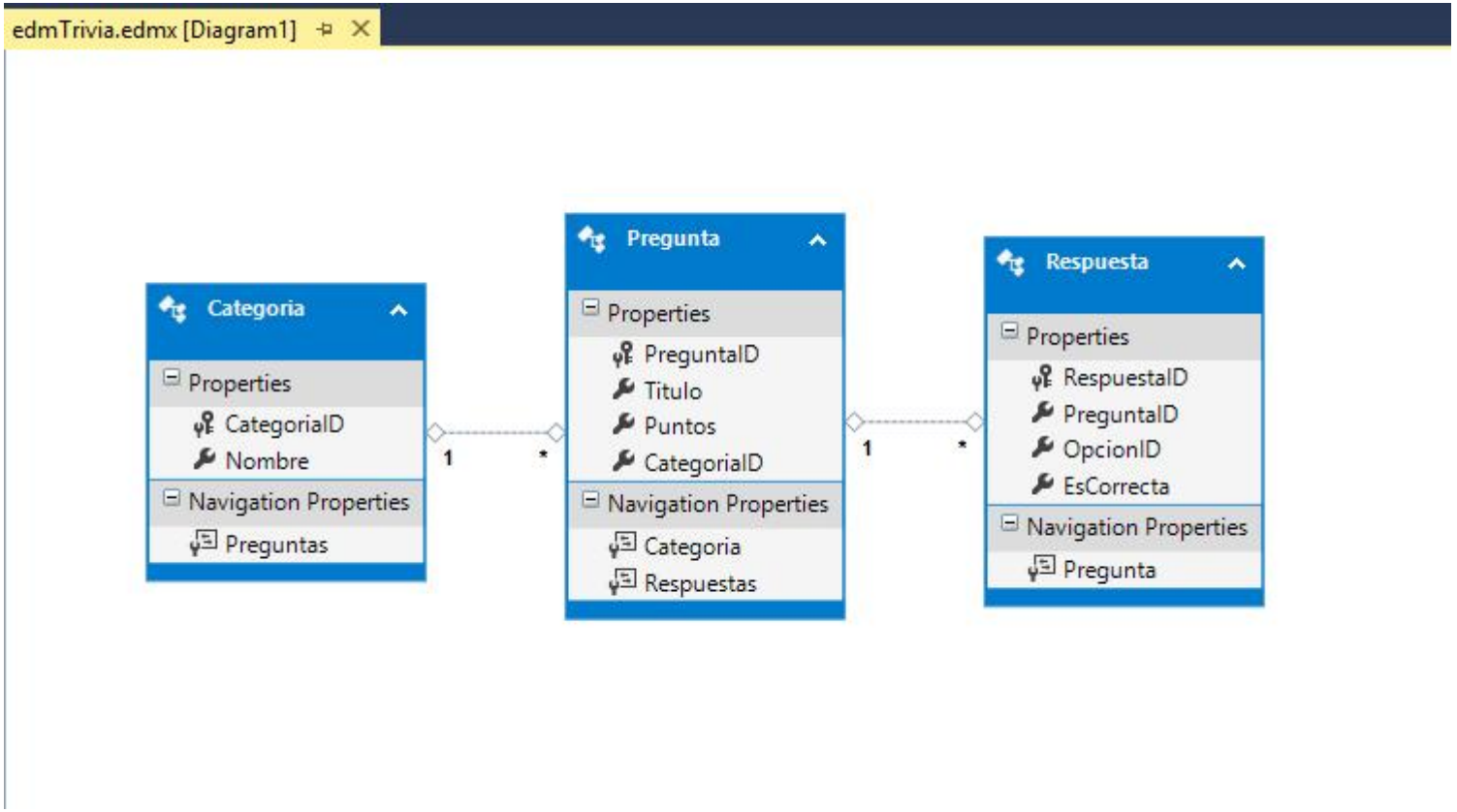
Introducción a ADO.NET EF

- ADO.NET Entity Framework:
  - Entity Data Model (EDM)
  - Entity SQL
- Características
  - Escribir código contra un modelo conceptual
  - Escribir código independiente de la base de datos
  - Actualizaciones contra una fuente de datos
  - El código soporta validaciones de tipos en tiempo de ejecución

EDM: Modelo de datos

- Existen 3 modelos de datos:
  - Modelo Físico:
    - Como la información es guardada en base de datos
  - Modelo Lógico:
    - Describe como los datos son guardados en las tablas y como se relacionan usando claves
  - Modelo Conceptual
    - Vista de negocio de los datos, mapeado al modelo lógico
- Entity Data Model, es el modelo conceptual definido en Entity Framework

Entity Designer



EDM: Estructura

- EDM es almacenado en un archivo XML *.edmx, que contiene:
  - Sección SSDL
  - Sección CSDL
  - Sección de mapeo C-S
- Se puede editar manualmente el archivo para:
  - Agregar nuevas columnas
  - Editar las claves de entidad
  - Definir funciones personalizadas

ObjectContext y ObjectStateManager

- La clase ObjectContext permite interactuar contra un modelo conceptual
- ObjectContext:
  - Maneja la conexión hacia la base de datos
  - Mantiene la metadata descriptiva del modelo
  - Maneja la información de las entidades usando la clase ObjectStateManager
- ObjectStateManager
  - Hace el seguimiento de todas las modificaciones hechas a los objetos del actual contexto

Leyendo y Editando datos con EF

- Leyendo data:

```
TriviaEntities DBContext = new TriviaEntities();
// Print a list of categories.
foreach (Categoria cat in DBContext.Categorias)
{
    Console.WriteLine("{0} {1}",cat.CategoriaID, cat.Nombre);
}
```

- Modificando data:

```
// Update the category
var cat2 = DBContext.Categorias.First(c => c.Nombre == "Mundial 2016");
if (cat2 != null)
{
    cat2.Nombre = "Mundial 2010";
}
DBContext.SaveChanges();
```

Demostración:
Usar Entity Framework

- En esta demostración vamos a:
  - Crear un modelo EDM
  - Usar EF para recuperar información de una base de datos
  - Usar EF para insertar registros a una base de datos

Resumen de Sesión

- EntityFramework simplica el acceso a datos usando un modelo de objetos.
- EntityDesigner permite actualizar y editar las tablas a usar.
- Se puede acceder o insertar registros usando objetos