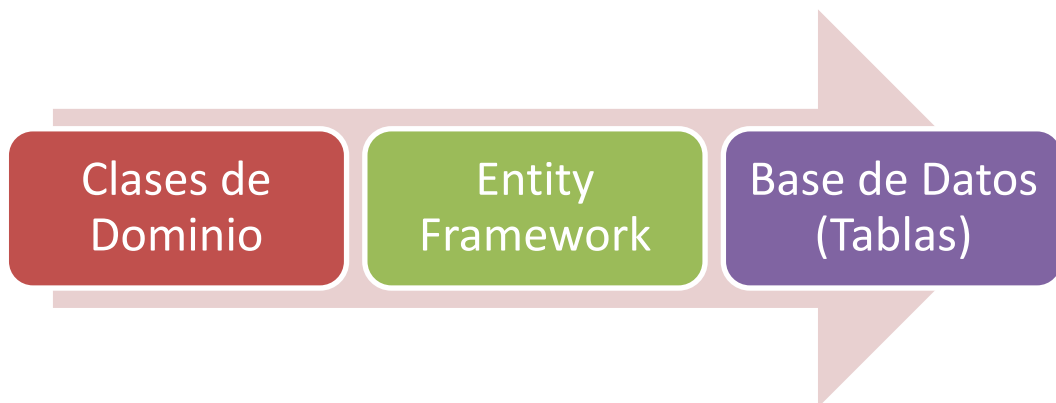


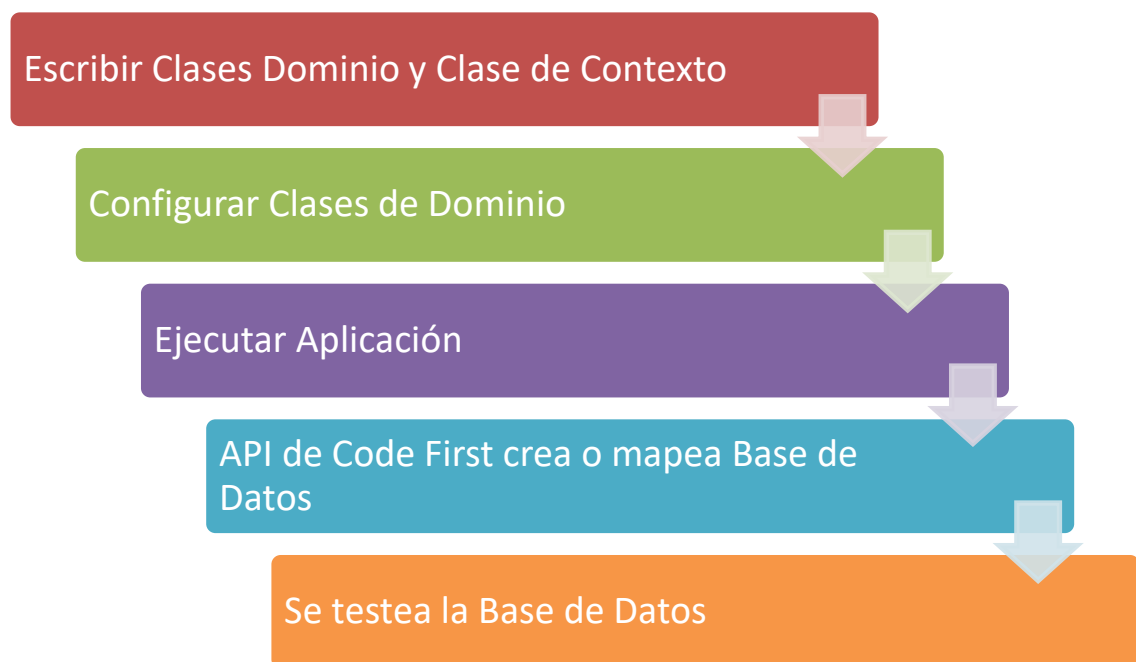
Introducción a Code First

Con Code First puede centrarse en el diseño del dominio y empezar a crear clases según sus requisitos de dominio en lugar de diseñar primero la base de datos y, a continuación, crear las clases que coincidan con el diseño de la base de datos. Las APIs de Code First crearán la base de datos sobre la marcha basándose en sus clases de entidad y en la configuración.



Como desarrollador, primero empezarás escribiendo clases en C# y las clases de contexto. Al ejecutar la aplicación, las API de Code First crearán la nueva base de datos, si todavía no existen, y asignarán las clases a la base de datos mediante las convenciones predeterminadas de Code First. También puedes configurar las clases de dominio para sustituir las convenciones predeterminadas para poder asignarlas a las tablas de la base de datos utilizando los atributos de DataAnnotation o la fluent API.

El flujo de trabajo básico de Code First:



Las Convenciones de Code First

Las convenciones de Code First es un conjunto de reglas predeterminadas para configurar automáticamente un modelo conceptual basado en las definiciones de las clases de dominio. Las convenciones de Code First se definen en el espacio de nombres `System.Data.Entity.ModelConfiguration.Conventions`.

A continuación, vamos a ver una visión general de varias de las convenciones de Code First.

Descubrimiento de Tipos

Al establecer una clase de contexto con las propiedades DbSet para las clases que queremos que sean parte del modelo, Code First creará las tablas para las clases incluidas como propiedades DbSet. Code First también incluye todos los tipos referenciados incluidos en estas clases, incluso si los tipos referenciados se definen en un ensamblado diferente.

Las convenciones para el descubrimiento de tipos son:

1. Code First incluye tipos definidos como una propiedad DbSet en la clase de contexto.
2. Code First incluye los tipos de referencia incluidos en los tipos de entidad incluso si se definen en un ensamblado diferente.
3. Code First incluye las clases derivadas incluso si sólo la clase base se define como propiedad DbSet.

Clave Primaria

Code First crea automáticamente una clave primaria en cada tabla. La convención predeterminada para la clave principal es que Code First creará una clave primaria para una propiedad si el nombre de la propiedad es Id o el `<nombre_clase> Id`. El tipo de datos de una propiedad que sea clave principal puede ser cualquier cosa, pero si el tipo de la propiedad de clave principal es numérico o GUID, se configurará como una columna de identidad.

Si se ha definido una propiedad de clave distinta de Id o `<Nombre_Clase> Id`, Entity Framework lanzará `ModelValidationException`.

Convención de Relaciones

Code First infiere en la relación entre las dos entidades que utilizan la propiedad de navegación. Esta propiedad de navegación puede ser un simple tipo de referencia o un tipo de colección. Code First creará automáticamente la relación indicada en el modelo insertando la columna de clave externa en la tabla indicada.

Convención sobre claves externas

Code First inserta automáticamente una clave externa cuando encuentra una propiedad de navegación. Se recomienda incluir una propiedad de clave externa en el extremo dependiente de una relación.

Code First infiere la multiplicidad de la relación basada en la nulidad de la clave externa. Si la propiedad es nullable entonces la relación se registra como null. De lo contrario, la relación se registra como NOT NULL.

Convención sobre tipos complejos

Code First crea un tipo complejo para la clase que no incluye la propiedad clave y también si la clave primaria no está registrada usando DataAnnotation o Fluent API.

Convenciones Predeterminadas de Code First

Convención por Defecto	Descripción
Table Name	<Nombre Clase Entidad> + 's' Entity Framework creará la tabla de Base de Datos con el nombre de la clase de la entidad más el sufijo 's'
Primary key Name	1) Id 2) <Nombre Clase Entidad> + "Id" (case insensitive) Entity Framework creará una columna de clave primaria para la propiedad llamada Id o <Nombre Clase Entidad> + "Id" (sin distinción entre mayúsculas y minúsculas)
Foreign key property Name	De forma predeterminada, Entity Framework buscará la propiedad de clave externa con el mismo nombre que el nombre de la clave principal de la entidad principal. Si la propiedad de clave externa no existe, entonces Entity Framework creará una columna FK en la tabla Db con <Nombre de la propiedad de navegación dependiente> + "_" + <Entidad principal Nombre de la propiedad de la clave principal>
Null column	Entity Framework crea una columna null para todas las propiedades de tipo de referencia y propiedades primitivas anulables.
Not Null Column	Entity Framework crea columnas no nulas para propiedades de clave principal y propiedades de tipo de valor no anulables.
DB Columns order	Entity Framework creará columnas de Bases de Datos iguales que el orden de las propiedades en una clase de entidad. Sin embargo, las columnas de clave principal se moverán primero.
Properties mapping to DB	De forma predeterminada, todas las propiedades se asignarán a la base de datos. Utilice el atributo [NotMapped] para excluir la propiedad o la clase de la asignación de la Base de Datos.
Cascade delete	Enabled Por defecto para todos los tipos de relaciones.

La siguiente tabla lista la asignación de tipo de datos C# con el tipo de datos SQL y el tipo de datos y la longitud de la columna de clave principal.

C# DataType	Related DB Column DataType	PK Column DataType & Length
int	int	int, Identity column increment by 1
string	nvarchar(Max)	nvarchar(128)
decimal	decimal(18,2)	decimal(18,2)
float	real	real
byte[]	varbinary(Max)	varbinary(128)
datetime	datetime	datetime
bool	bit	bit
byte	tinyint	tinyint
short	smallint	smallint
long	bigint	bigint
double	float	float
char	No mapping	No mapping
sbyte	No mapping (throws exception)	No mapping
object	No mapping	No mapping

Estas convenciones se pueden invalidar utilizando DataAnnotation o Fluent API. Además, puedes definir convenciones personalizadas para tu aplicación utilizando Entity Framework 6.0, en adelante.

Lista de DataAnnotations

Class	Description
AssociatedMetadataType	Extiende la información de metadatos de una clase mediante la adición de atributos e información de propiedades que se define en una clase asociada.
AssociationAttribute	Especifica que un miembro de entidad representa una relación de datos, como una relación de clave externa.
ConcurrencyCheckAttribute	Especifica que una propiedad participa en verificaciones de concurrencia optimistas.
CustomValidationAttribute	Especifica un método de validación personalizado para validar una instancia de propiedad o clase.
DataTypeAttribute	Especifica el nombre de un tipo adicional para asociar con un campo de datos.
DisplayAttribute	Proporciona un atributo de propósito general que le permite especificar cadenas localizables para tipos y miembros de clases parciales de entidad.
DisplayColumnAttribute	Especifica la columna que se muestra en la tabla referida como una columna de clave externa.
DisplayFormatAttribute	Especifica cómo los campos de datos se visualizan y formatean mediante ASP.NET Dynamic Data.
EditableAttribute	Indica si un campo de datos es editable.
EnumDataTypeAttribute	Habilita una enumeración de .NET Framework para asignarse a una columna de datos.
FilterUIHintAttribute	Representa un atributo que se utiliza para especificar el comportamiento de filtrado de una columna.
KeyAttribute	Indica una o más propiedades que identifican de manera única a una entidad.
MetadataTypeAttribute	Especifica la clase de metadatos a asociar con una clase de modelo de datos.
RangeAttribute	Especifica las restricciones de rango numérico para el valor de

	un campo de datos.
RegularExpressionAttribute	Especifica que un valor de campo de datos en ASP.NET Dynamic Data debe coincidir con la expresión regular especificada.
RequiredAttribute	Especifica que se requiere un valor de campo de datos.
ScaffoldColumnAttribute	Especifica si una columna de clase o de datos utiliza scaffolding.
ScaffoldTableAttribute	Especifica si una tabla de clase o de datos utiliza scaffolding.
StringLengthAttribute	Especifica la longitud mínima y máxima de caracteres permitidos en un campo de datos.
TimestampAttribute	Especifica el tipo de datos de la columna como una versión de fila.
UIHintAttribute	Especifica la plantilla o el control de usuario que utiliza Dynamic Data para mostrar un campo de datos.
ValidationAttribute	Sirve como clase base para todos los atributos de validación.
ValidationContext	Describe el contexto en el que se realiza una comprobación de validación.
ValidationException	Representa la excepción que se produce durante la validación de un campo de datos cuando se utiliza la clase ValidationAttribute.
ValidationResult	Representa un contenedor para los resultados de una solicitud de validación.
Validator	Define una clase auxiliar que se puede utilizar para validar objetos, propiedades y métodos cuando se incluye en sus atributos ValidationAttribute asociados.