

EF proporciona las siguientes **dos técnicas para la realización de consultas** en el modelo. Ambas técnicas se aplican igualmente a los modelos creados con Code First y EF Designer (DataBase First).

1. Mediante **LINQ** en la base de datos (más información en [Consultas en LINQ to Entities](#))

→ esta técnica crea la consulta en la base de datos, lo que **siempre** conlleva un viaje de ida y vuelta a la **base de datos** aun cuando las entidades devueltas ya existan en el contexto

→ admite dos tipos de sintaxis:

◆ [sintaxis de expresiones de consulta](#)

ejemplo:

```
using (AdventureWorksEntities context = new AdventureWorksEntities())
{
    IQueryable<string> productNames =
        from p in context.Products
        select p.Name;

    Console.WriteLine("Product Names:");
    foreach (String productName in productNames)
    {
        Console.WriteLine(productName);
    }
}
```

◆ [sintaxis de consultas basadas en métodos \(o expresiones lambda\)](#)

ejemplo:

```
using (AdventureWorksEntities context = new AdventureWorksEntities())
{
    IQueryable<string> productNames = context.Products
        .Select(p => p.Name);

    Console.WriteLine("Product Names:");
    foreach (String productName in productNames)
    {
        Console.WriteLine(productName);
    }
}
```

2. Haciendo uso del método **Find** de DbSet (más información en [Búsqueda de una entidad por clave principal](#))

→ esta técnica emplea el valor de **clave principal** para intentar buscar una entidad cuyo seguimiento realiza el contexto

→ diferencias importantes respecto al uso de LINQ:

- ◆ Solo se realiza el viaje de ida y vuelta a la base de datos si la entidad con la clave especificada no se encuentra en el contexto.
- ◆ Find devuelve entidades que están en estado *Added*. Es decir, Find devuelve entidades que se han agregado al contexto pero que aún no se han guardado en la base de datos.

ejemplo:

```
using (var context = new BloggingContext())
{
    // Will hit the database
    var blog = context.Blogs.Find(3);

    // Will return the same instance without hitting the database
    var blogAgain = context.Blogs.Find(3);

    context.Blogs.Add(new Blog { Id = -1 });

    // Encontrará el nuevo blog aunque no exista en la base de datos
    var newBlog = context.Blogs.Find(-1);

    // Will find a User which has a string primary key
    var user = context.Users.Find("johndoe1987");
}
```