

Sesión 02

Entity Framework Core

Instructor:

ERICK ARÓSTEGUI

earostegui@galaxy.edu.pe



8 NET

FULL-STACK DEVELOPER

ÍNDICE

01

Almacenamiento de información

02

Base de datos relacional

03

Entity Framework Core, Modelado de datos

04

Mapping y Querys en EF Core

05

Estrategias de uso y características avanzadas de Entity Framework Core

01



Almacenamiento de información

→ Almacenamiento de información

Se refiere a la **capacidad y métodos** que una **aplicación de software** utiliza para **guardar, mantener y recuperar información o datos**. Este almacenamiento puede ser **temporal** (como en la memoria caché) o **permanente** (como en una base de datos).



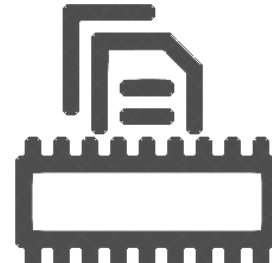
Base de Datos

Almacenamiento a largo plazo



Archivos locales

Almacenamiento en formato de archivos del sistema



Memoria Cache

Almacenamiento temporal de acceso rápido



Cloud Storage

Acceso desde cualquier dispositivo



Web Storage

Almacenamiento en el navegador web

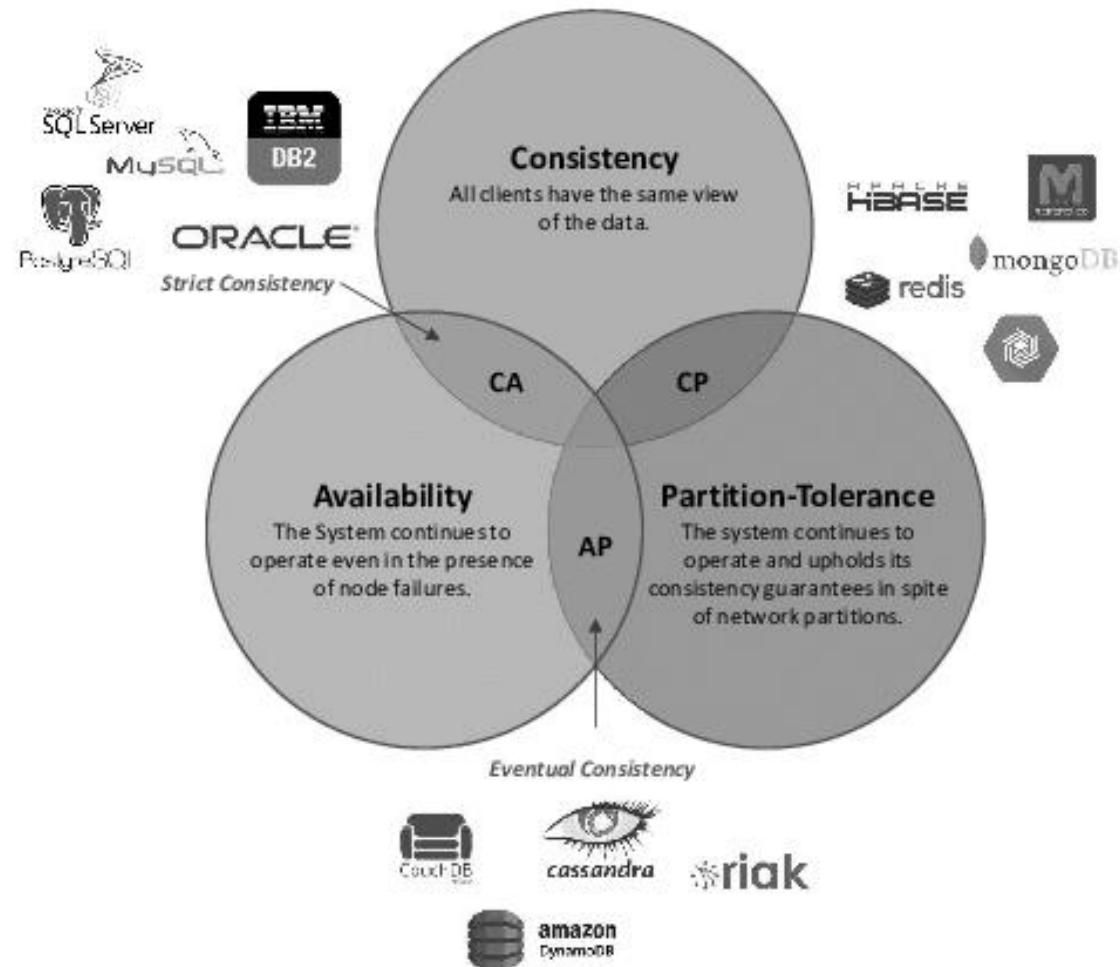
NoSQL

El término "**NoSQL**" puede ser un poco confuso porque no se refiere necesariamente a la falta de SQL (Structured Query Language). En cambio, NoSQL originalmente significaba "**No SQL**" pero ha evolucionado para significar "**Not Only SQL**". Esto significa que estas bases de datos no se limitan a la estructura tabular y al lenguaje de consulta SQL utilizados en las bases de datos relacionales, pero pueden usarlos si es necesario.

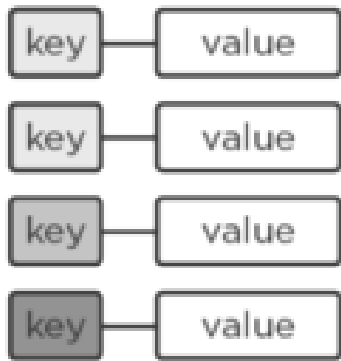
Las bases de datos NoSQL fueron creadas en respuesta a las limitaciones de las bases de datos relacionales, especialmente para manejar conjuntos de datos grandes, distribuidos y semi-estructurados.

Los sistemas NoSQL generalmente **no requieren un esquema fijo** como las bases de datos SQL, **son más fáciles de escalar en forma horizontal**, y fueron diseñados para operaciones simples en un gran número de registros.

Teorema de CAP

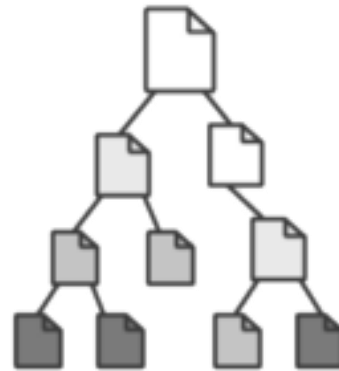


Tipos de Base de Datos NoSQL



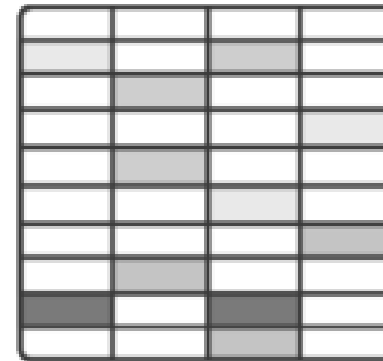
Clave-Valor

Vincula una clave dada a un registro de cualquier tipo



Documental

Se basan en el concepto detrás de clave-valor, extendiéndolo para admitir objetos complejos de varias capas denominados documentos



Columnar

Puede pensar en las filas como claves en un almacén clave-valor y las columnas como el valor



Grafos

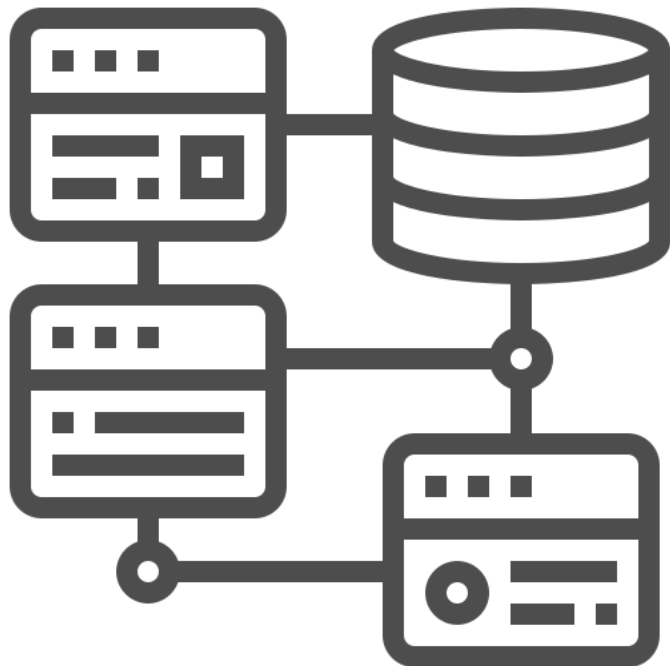
Enfoque hacia la relación entre entidades. Las entidades, como los usuarios, están representadas por nodos, mientras que las conexiones entre entidades dictan cómo se relacionan

02



Base de datos relacional

¿Qué es una base de datos?



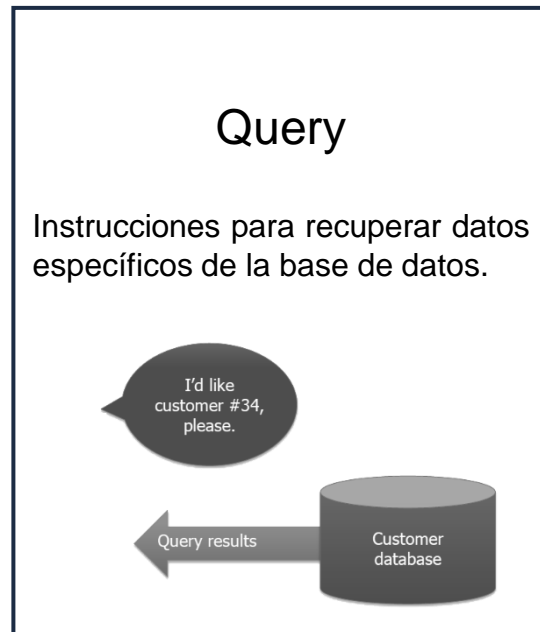
Una colección de datos almacenados en formato electrónico.

Usualmente diseñado para hacer que la **lectura y actualización de datos sea rápida y fácil**

Algo así como una guía telefónica, mucho más rápido de consultar, y mucho más fácil de corregir y agregar a

Términos Clave

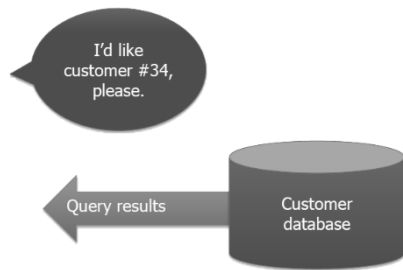
Términos Clave



Términos Clave

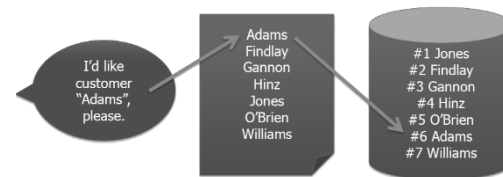
Query

Instrucciones para recuperar datos específicos de la base de datos.



Index

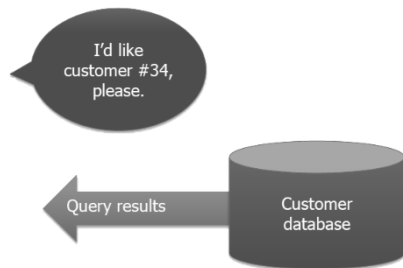
Una estructura de datos especial que ayuda a acelerar la selección. consultas, al igual que un índice en un libro



Términos Clave

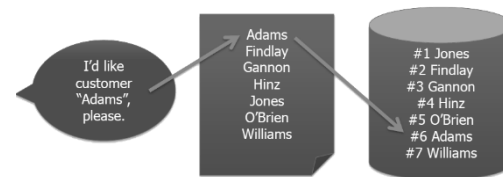
Query

Instrucciones para recuperar datos específicos de la base de datos.



Index

Una estructura de datos especial que ayuda a acelerar la selección. consultas, al igual que un índice en un libro



Almacenamiento físico

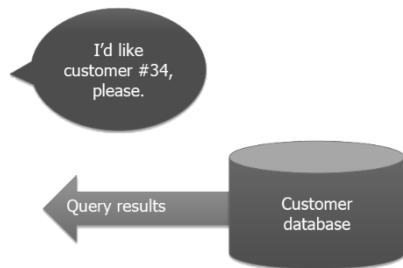
SQL Server almacena sus bases de datos en disco, en hasta tres diferentes tipos de archivos.



Términos Clave

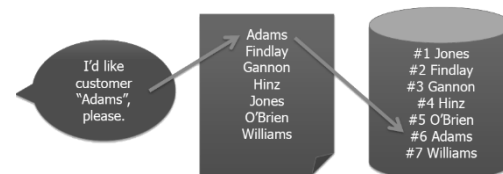
Query

Instrucciones para recuperar datos específicos de la base de datos.



Index

Una estructura de datos especial que ayuda a acelerar la selección. consultas, al igual que un índice en un libro



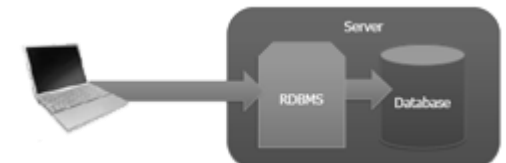
Almacenamiento físico

SQL Server almacena sus bases de datos en disco, en hasta tres diferentes tipos de archivos.



RDBMS

Sistema de gestión de base de datos relacional (por ejemplo, SQL Server) - una colección de software que gestiona el acceso a la base de datos.



Términos Clave

Términos Clave

Data Manipulation Language (DML)

El lenguaje de consulta utilizado para consultar y actualizar datos en la base de datos. SQL Server utiliza Transact-SQL (T-SQL), que es una variante del lenguaje general "SQL"

Términos Clave

Data Manipulation Language (DML)

El lenguaje de consulta utilizado para consultar y actualizar datos en la base de datos. SQL Server utiliza Transact-SQL (T-SQL), que es una variante del lenguaje general "SQL"

Data Definition Language (DDL)

Utilizado para manipular el esquema de una base de datos. Creando tablas, modificando índices, etc. Los estándares ANSI cubren muy poco DDL, aunque la mayoría de proveedores de bases de datos han convergido en convenciones similares. servidor SQL utiliza T-SQL

Términos Clave

Data Manipulation Language (DML)

El lenguaje de consulta utilizado para consultar y actualizar datos en la base de datos. SQL Server utiliza Transact-SQL (T-SQL), que es una variante del lenguaje general "SQL"

Data Definition Language (DDL)

Utilizado para manipular el esquema de una base de datos. Creando tablas, modificando índices, etc. Los estándares ANSI cubren muy poco DDL, aunque la mayoría de proveedores de bases de datos han convergido en convenciones similares. servidor SQL utiliza T-SQL

Stored Procedures

Son básicamente "archivos por lotes" o "scripts" que indican como ejecutar un conjunto dado de comandos en un orden específico.

Se ejecutan en el servidor, lo que puede reducir el procesamiento. Requisitos en equipos cliente y en aplicaciones cliente.

Términos Clave

Data Manipulation Language (DML)

El lenguaje de consulta utilizado para consultar y actualizar datos en la base de datos. SQL Server utiliza Transact-SQL (T-SQL), que es una variante del lenguaje general "SQL"

Data Definition Language (DDL)

Utilizado para manipular el esquema de una base de datos. Creando tablas, modificando índices, etc. Los estándares ANSI cubren muy poco DDL, aunque la mayoría de proveedores de bases de datos han convergido en convenciones similares. servidor SQL utiliza T-SQL

Stored Procedures

Son básicamente "archivos por lotes" o "scripts" que indican como ejecutar un conjunto dado de comandos en un orden específico.

Se ejecutan en el servidor, lo que puede reducir el procesamiento. Requisitos en equipos cliente y en aplicaciones cliente.

Normalización

El propósito de la normalización es reducir el almacenamiento de datos y para reducir la redundancia de datos asegurándose de que cualquier pieza dada de datos se almacena una sola vez

Son 5 técnicas involucradas en la normalización completa del diseño de una base de datos

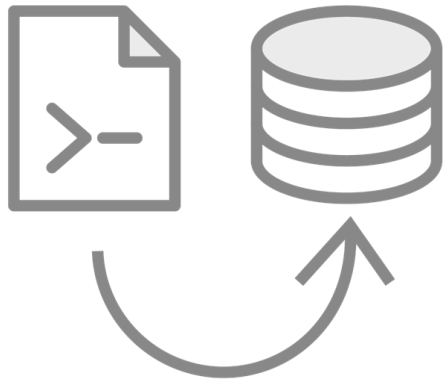
03



Entity Framework Core

Modelado de datos

Manejo de tareas de acceso a datos



Object Relational
Mapper (ORM)



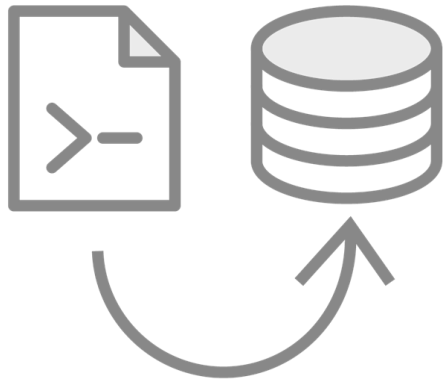
State/Change
tracker



Database schema
migrator

Entity Framework Core, Modelado de datos

Manejo de tareas de acceso a datos



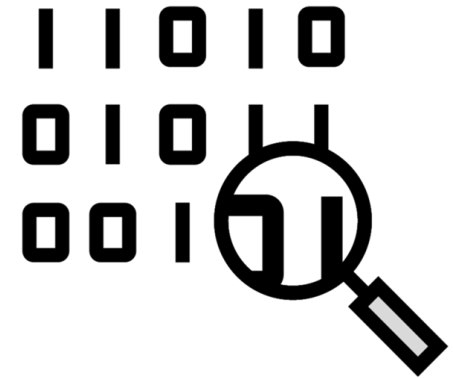
Object Relational
Mapper (ORM)



State/Change
tracker



Database schema
migrator



Transform LINQ
queries to SQL

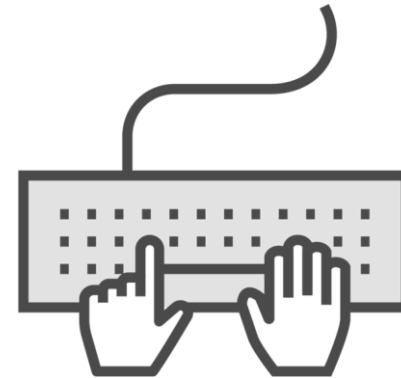
Consultas con EF Core



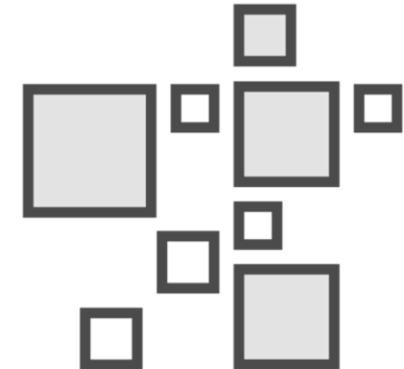
Porque usar el lenguaje de consulta LINQ integrado;



Escribir consultas en Clases .NET

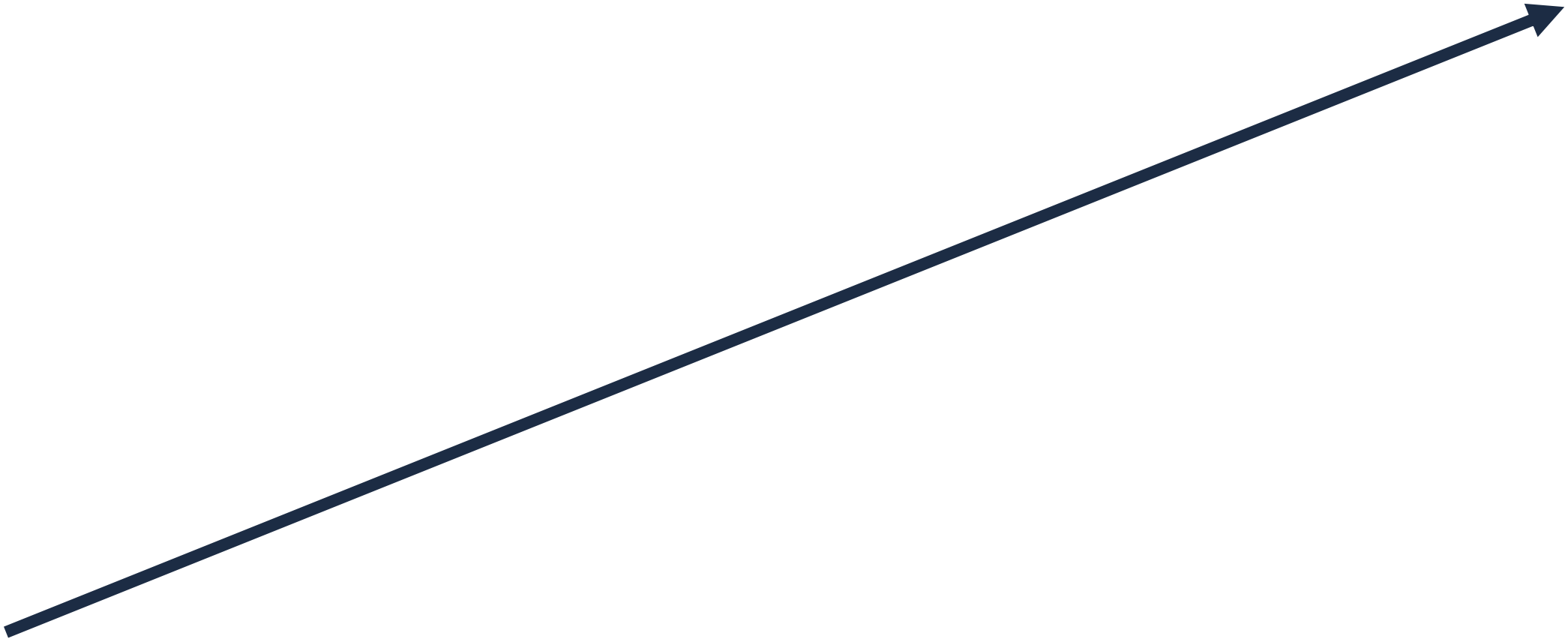


IntelliSense para simplificar la codificación

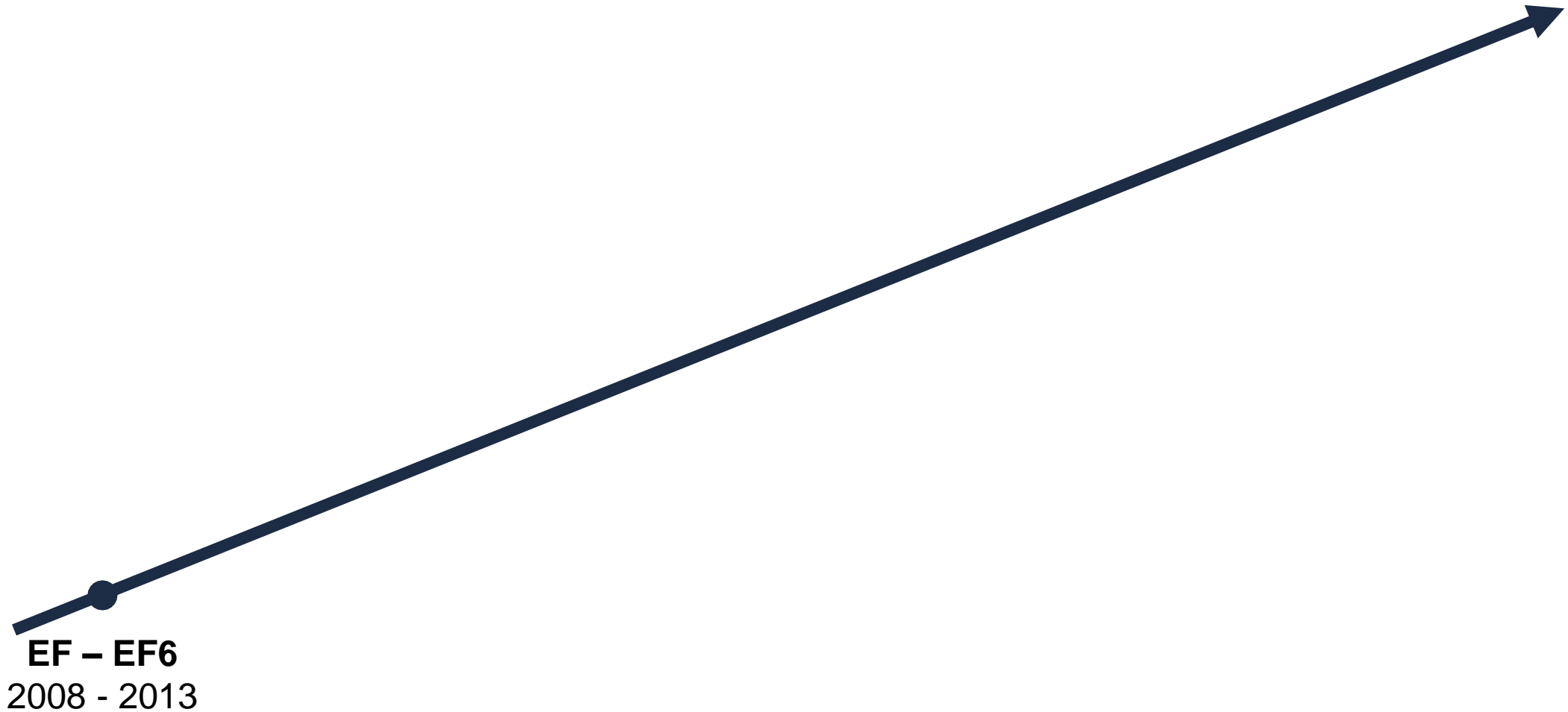


EF Core materializa objetos a partir de resultados de consulta

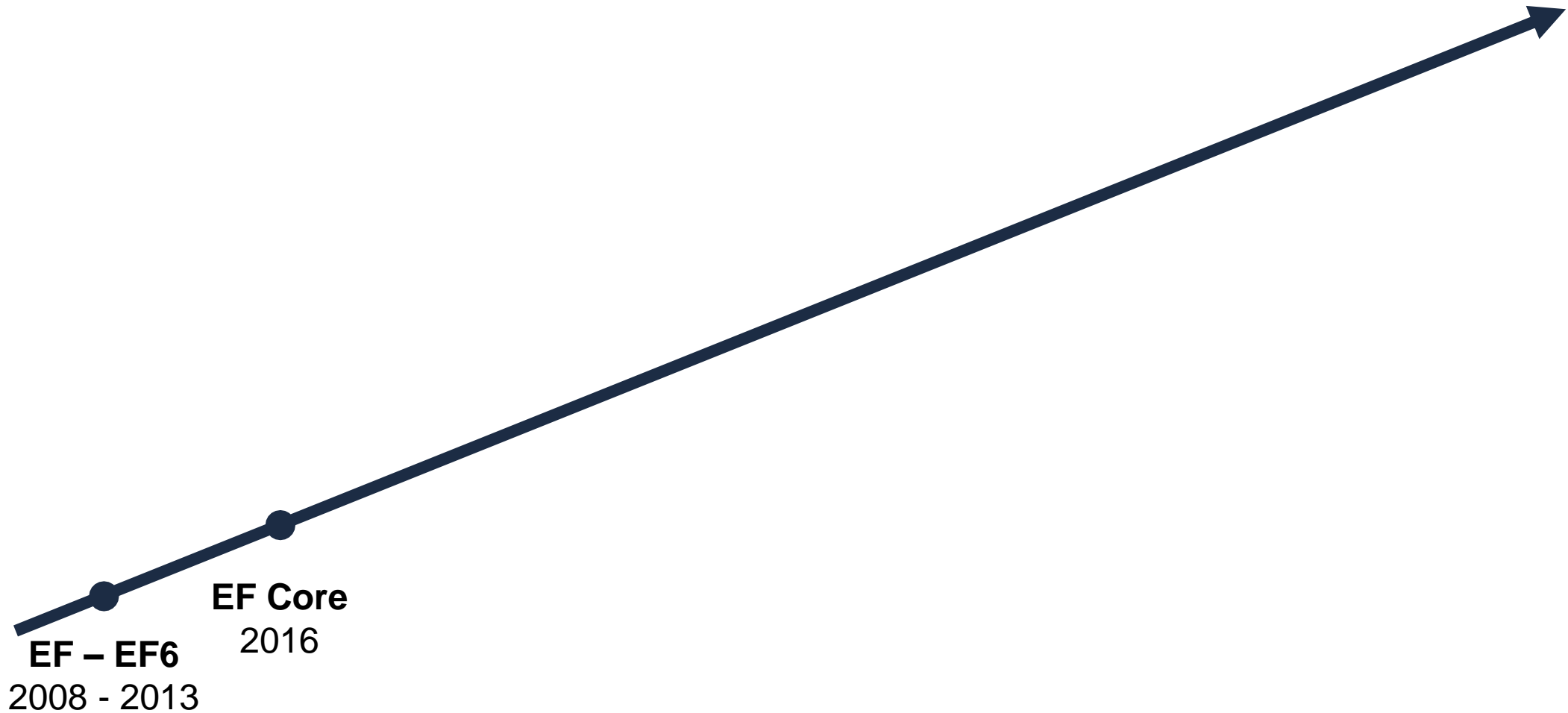
Evolución histórica



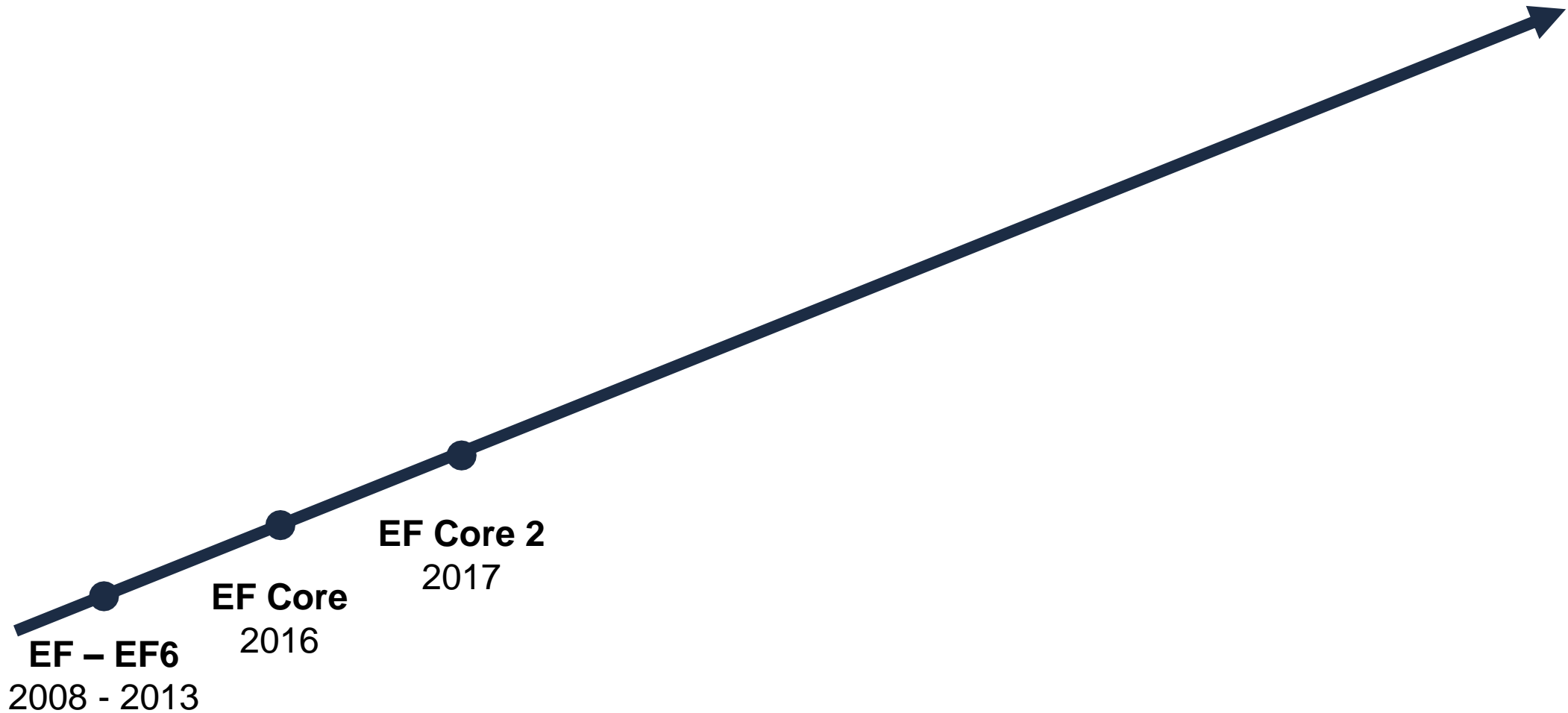
Evolución histórica



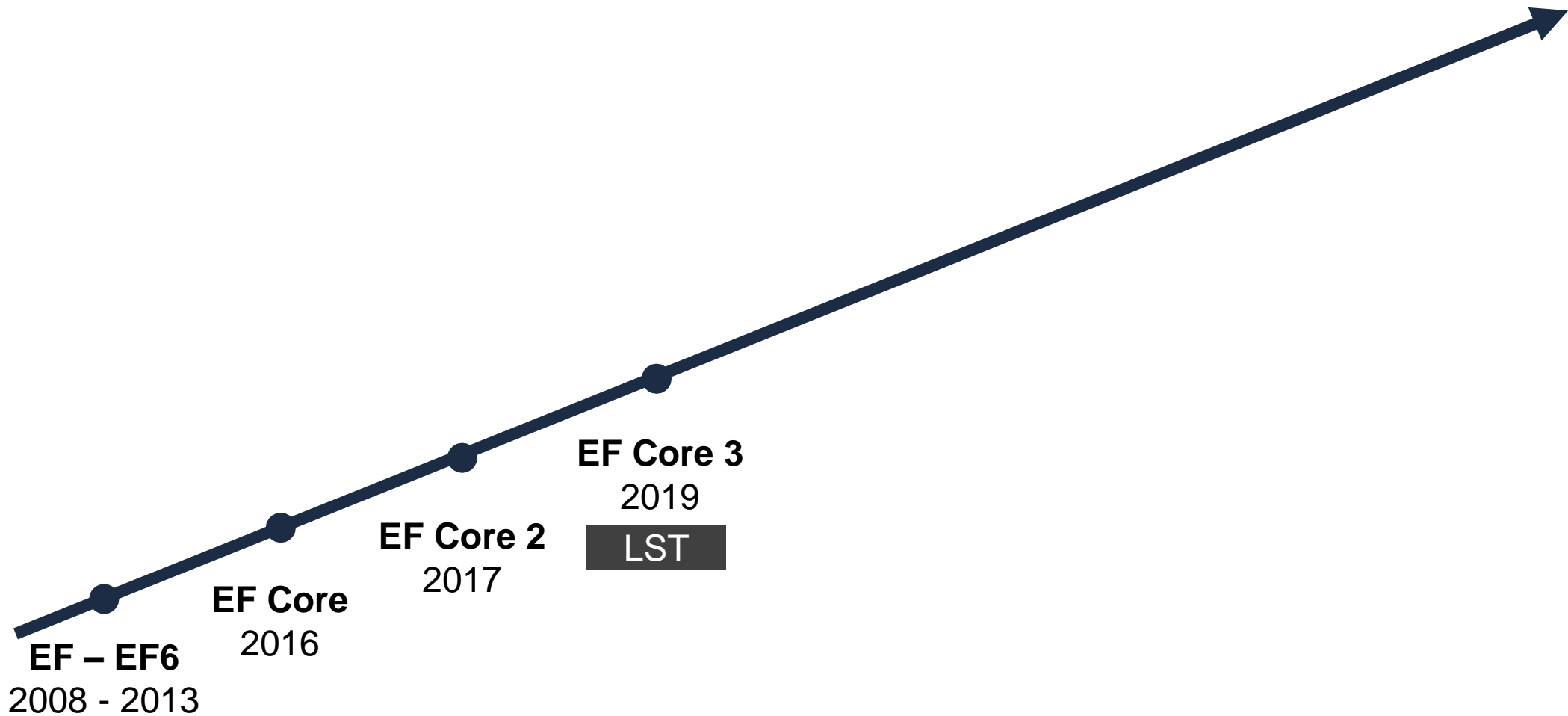
Evolución histórica



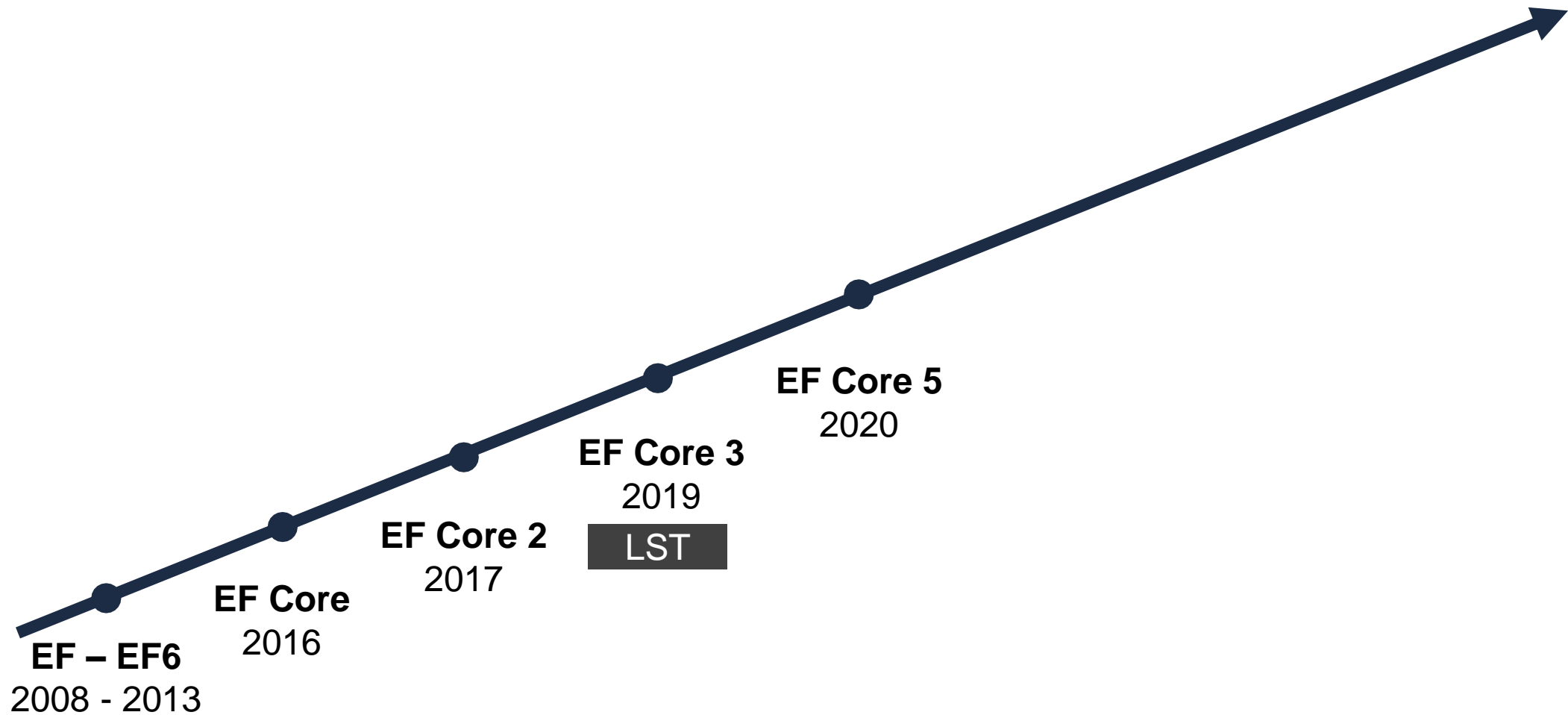
Evolución histórica



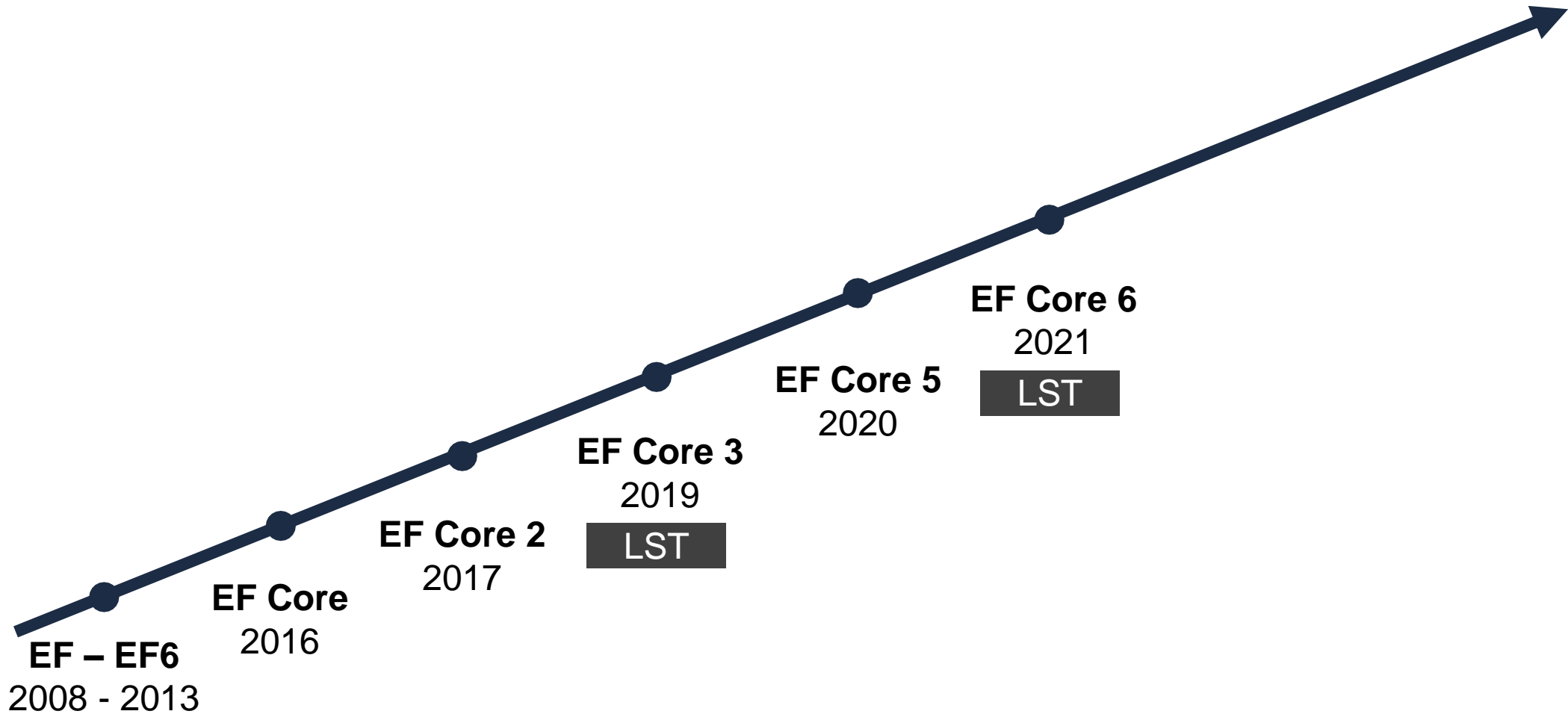
Evolución histórica



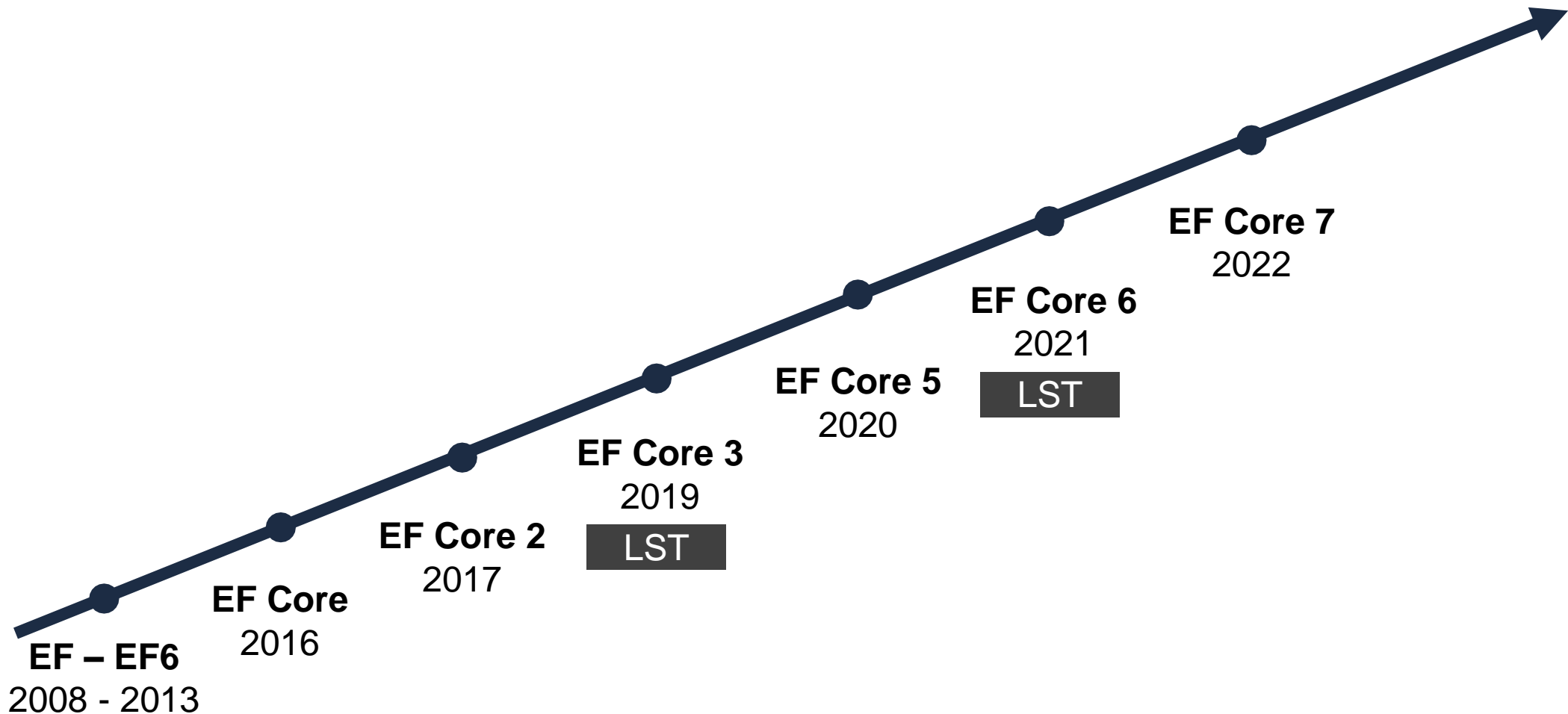
Evolución histórica



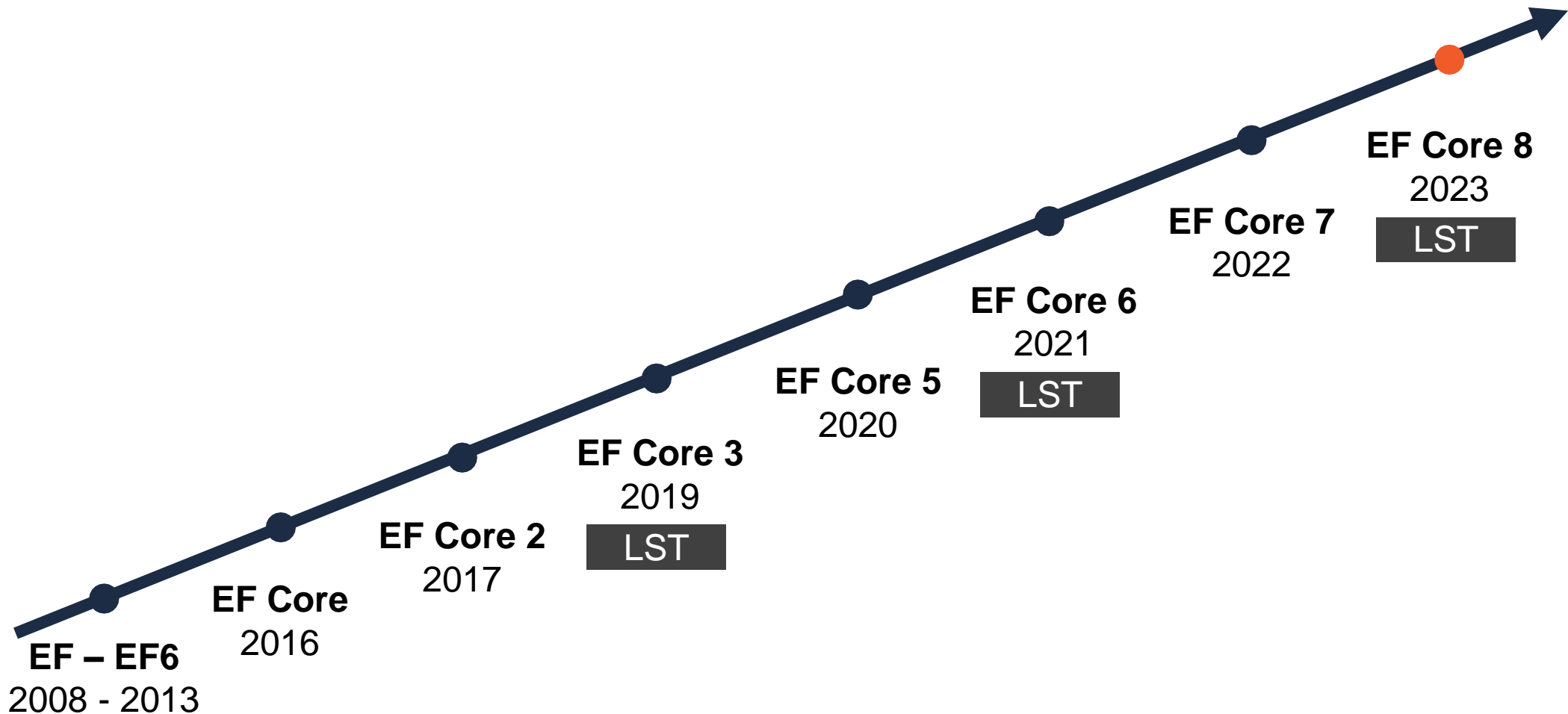
Evolución histórica



Evolución histórica



Evolución histórica



Proveedores de bases de datos creados por Microsoft






SQL Server

SQLite

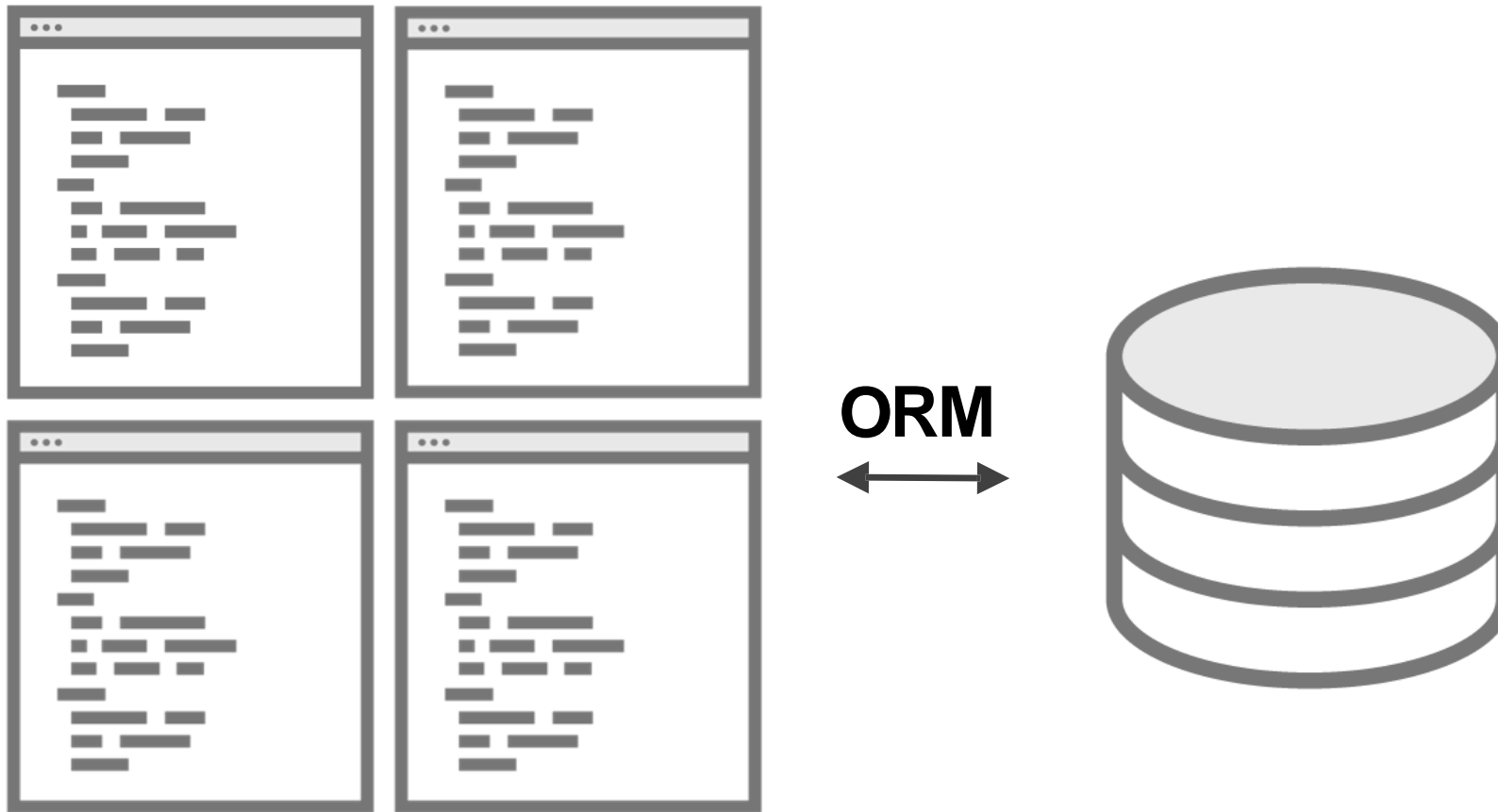
In-Memory

AzureCosmos DB

3rd Party y proveedores de bases de datos de código abierto

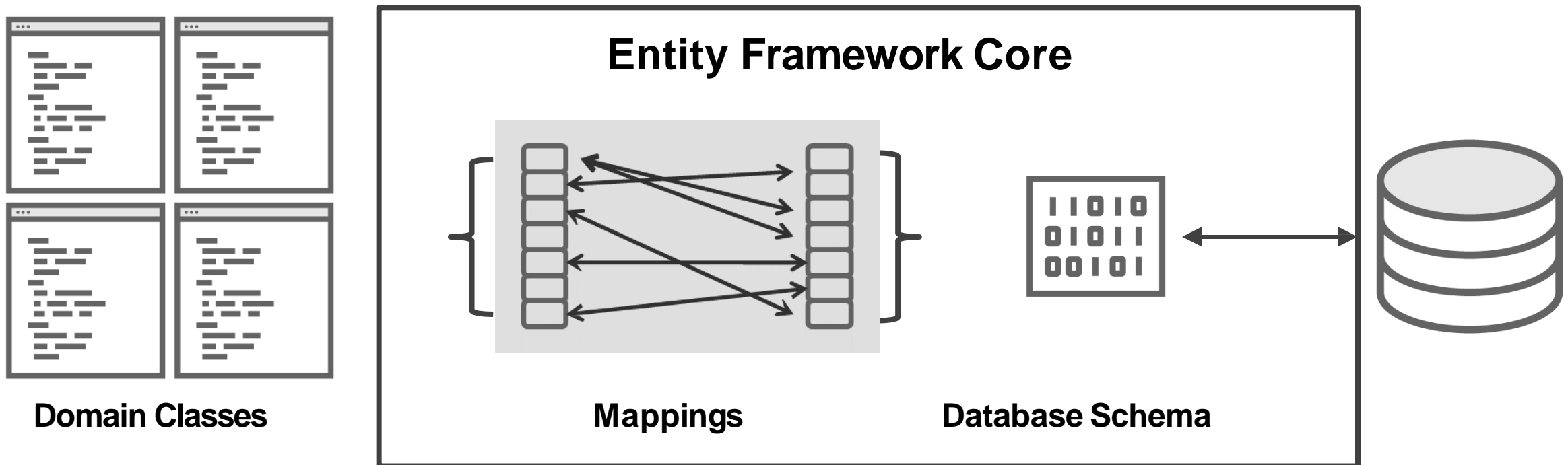
- | | |
|---|---|
|  MySQL |  Db2 & Informix |
|  Oracle DB |  MS Access |
|  PostgreSQL |  Google Cloud Spanner |
|  SQL Server |  SQL Server Compact |
|  SQLite |  Progress OpenEdge |
|  Firebird | |

ORM típico



Entity Framework Core, Modelado de datos

EF Core, mapeo inteligente y flexible



04




Mapping y Querys en EF Core




Traducción de tipo .NET a tipo de datos

Author type described in C#

```
public class Author
{
    public int AuthorId { get; set; }
    public string? FirstName { get; set; }
    public string? LastName { get; set; }
}
```

SQL Server table

 dbo.Author

 AuthorId (PK, int, not null)
 FirstName (nvarchar(max), null)
 LastName (nvarchar(max), null)

DbContext de EF Core es fundamental para toda persistencia

Controla qué clases se asignan a la base de datos

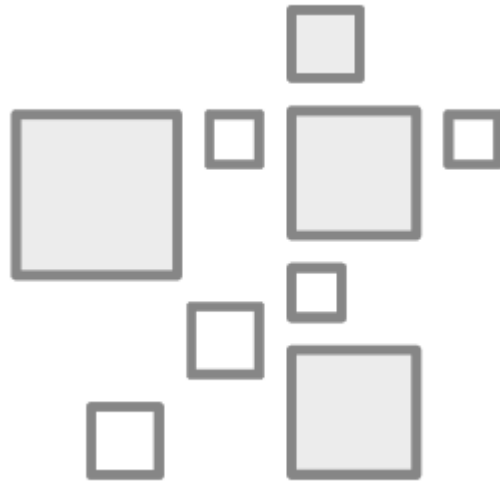
Siempre es necesario crear una clase que herede de DbContext

Controla todas las interacciones con la base de datos

Controla cómo se asignan esas clases a la base de datos.

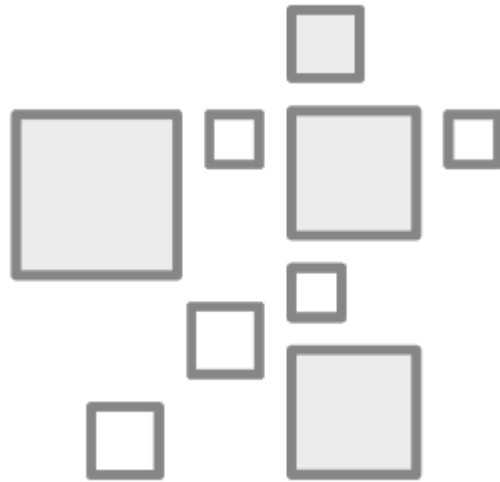
Entidades y el DbContext

Entidades y el DbContext

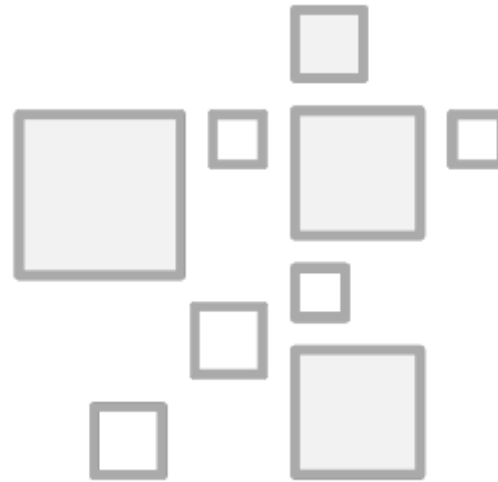


Objetos en memoria (In-Memory)

Entidades y el DbContext

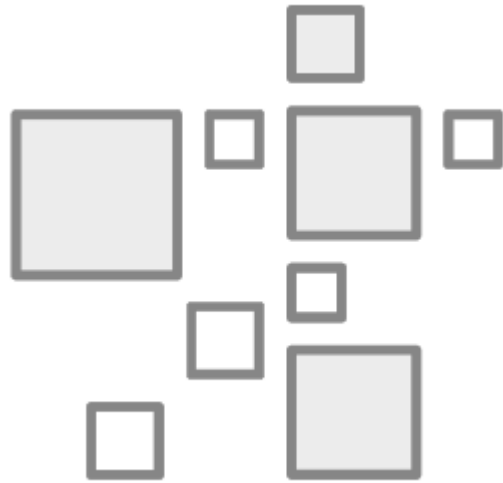


Objetos en memoria (In-Memory)

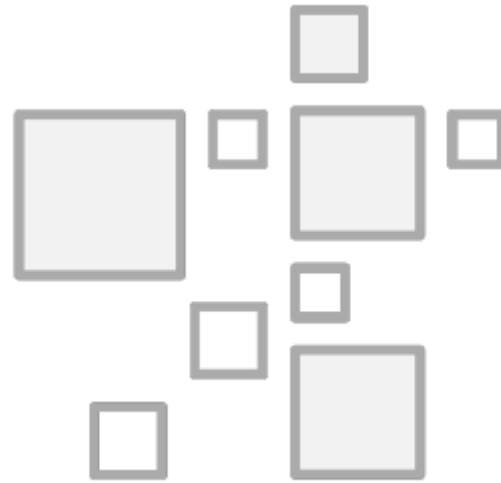


Entidades:
Objetos en memoria con
propiedades clave (identidad)
de los cuales el DbContext es
consciente.

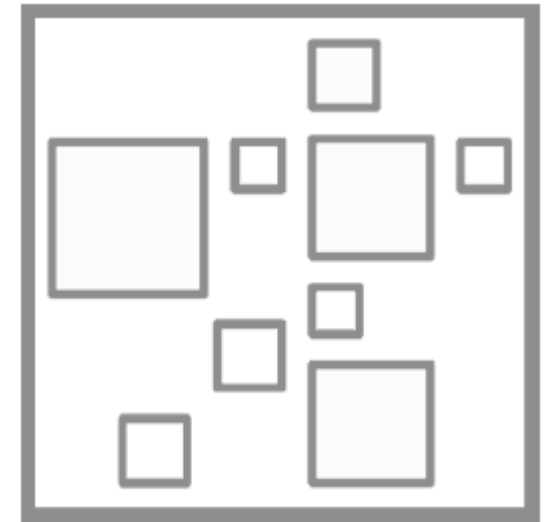
Entidades y el DbContext



Objetos en memoria (In-Memory)



Entidades:
Objetos en memoria con
propiedades clave (identidad)
de los cuales el DbContext es
consciente.



DbContext:
Contiene objetos EntityEntry
con punteros de referencia a
objetos en memoria.

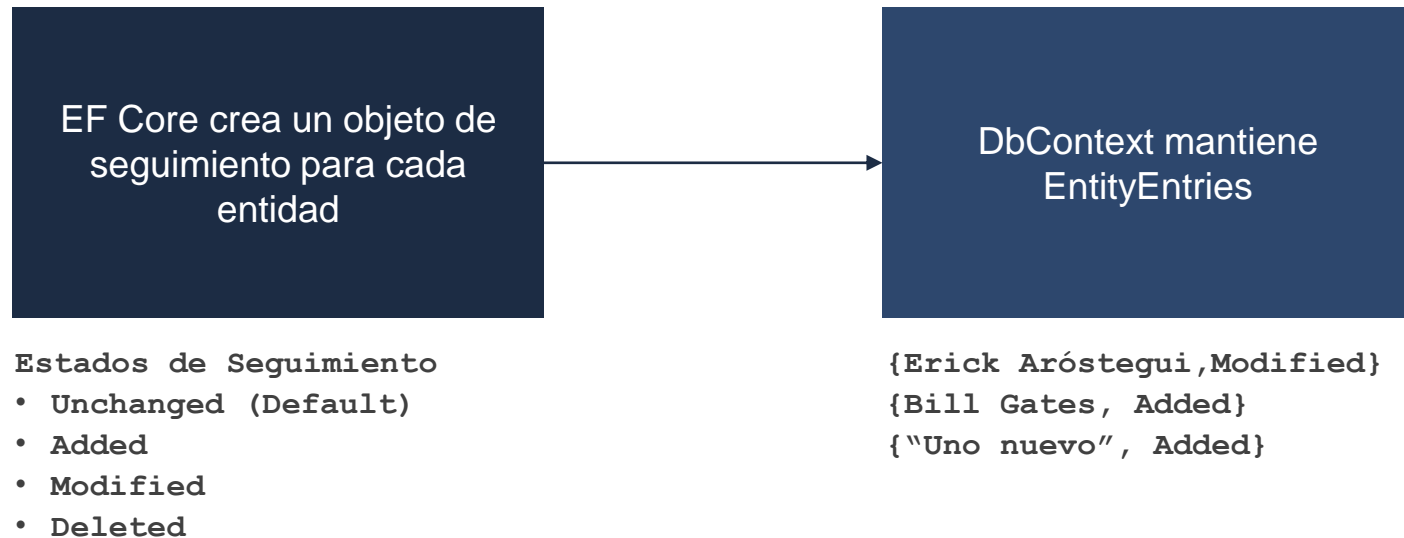
Flujo de trabajo de seguimiento y guardado

EF Core crea un objeto de seguimiento para cada entidad

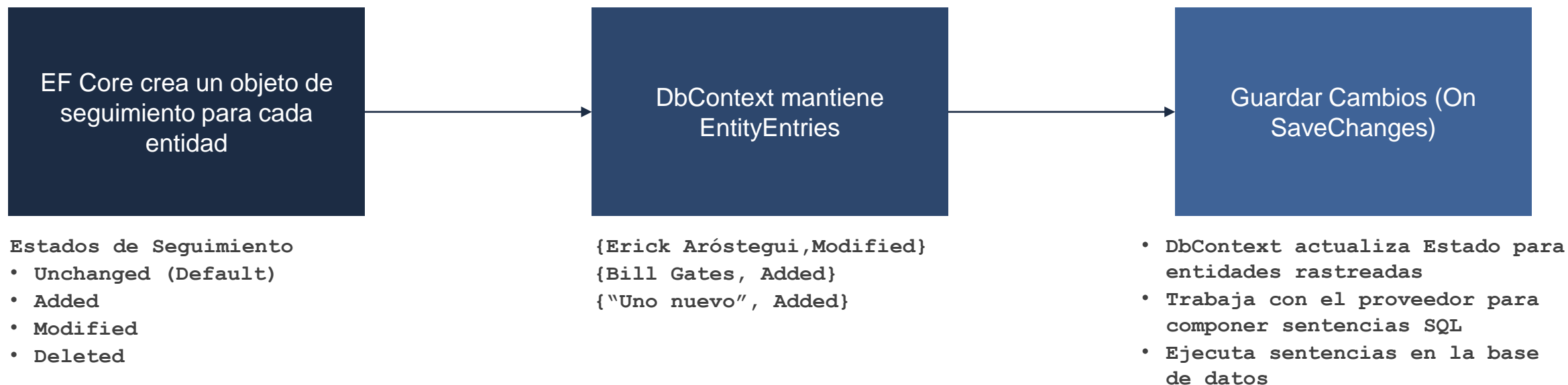
Estados de Seguimiento

- `Unchanged` (Default)
- `Added`
- `Modified`
- `Deleted`

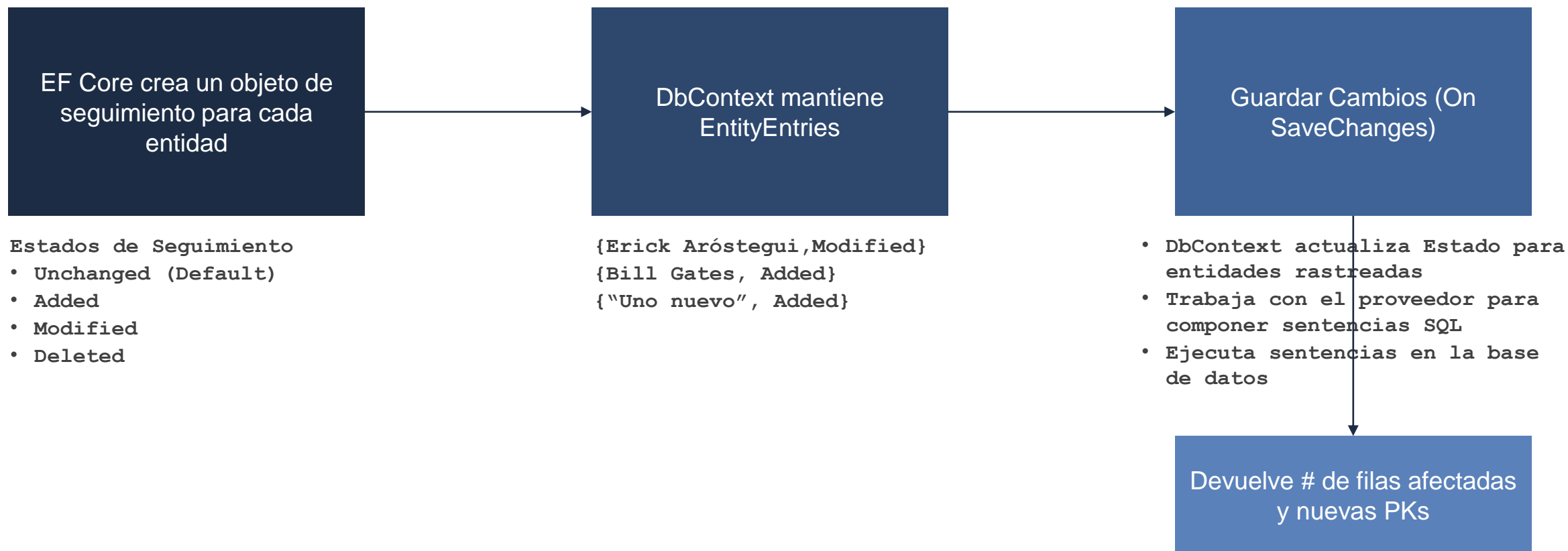
Flujo de trabajo de seguimiento y guardado



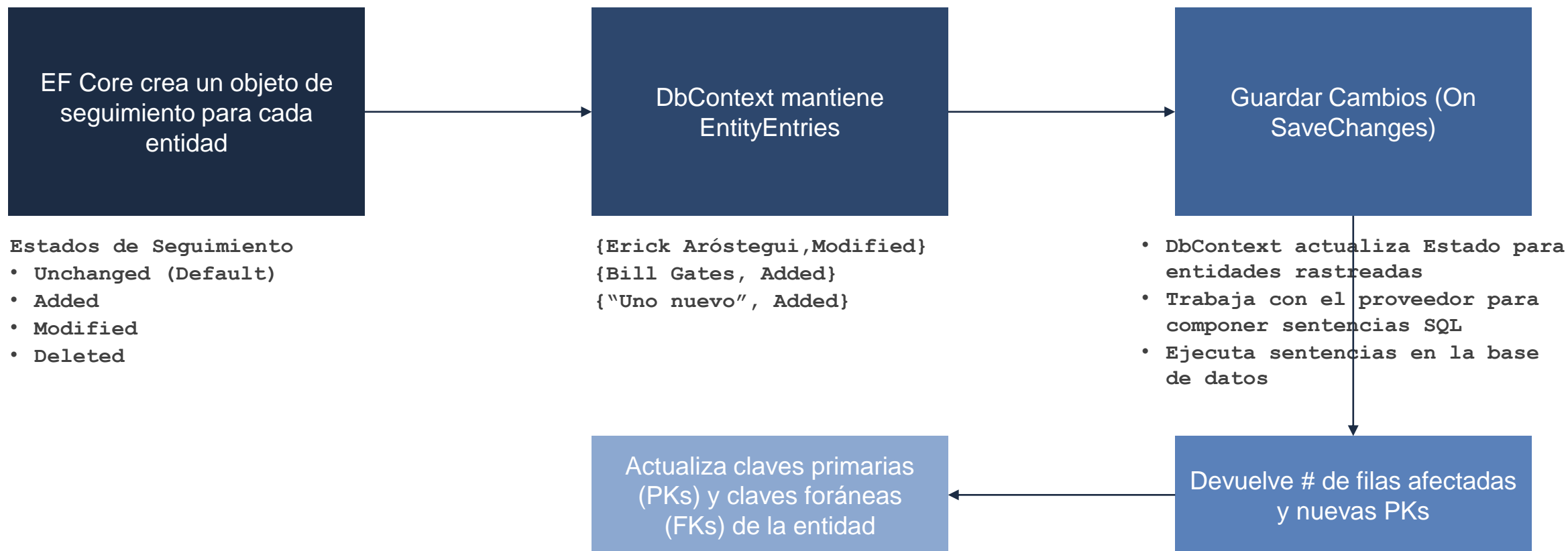
Flujo de trabajo de seguimiento y guardado



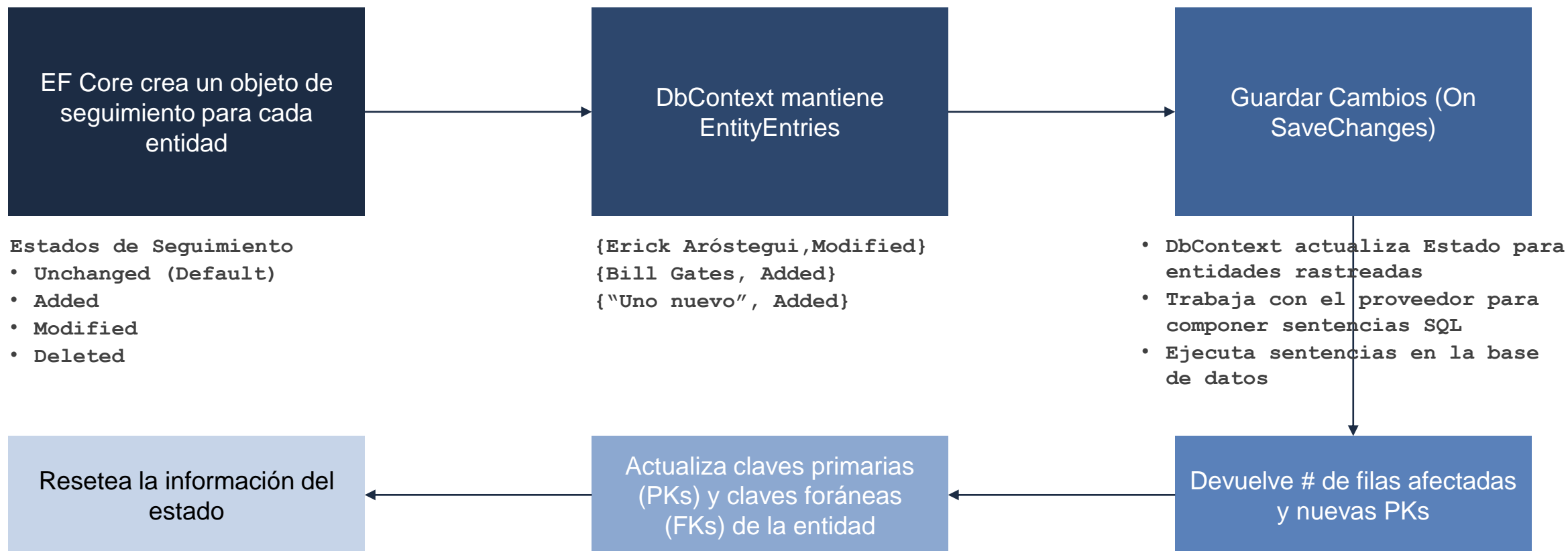
Flujo de trabajo de seguimiento y guardado



Flujo de trabajo de seguimiento y guardado



Flujo de trabajo de seguimiento y guardado



EF Core Flujo de Migraciones



Database Migration

Evolución del esquema de la base de datos para que coincida con los cambios en el modelo descrito por el código

EF Core Migrations API

EF Core Migrations API

Inspeccionar el modelo de
datos

EF Core Migrations API

Inspeccionar el modelo de
datos

Comparar con el estado del
modelo de datos después de la
migración anterior

EF Core Migrations API

Inspeccionar el modelo de
datos

Comparar con el estado del
modelo de datos después de la
migración anterior

¡Hay un nuevo BookCount int en
la clase de Libro!

EF Core Migrations API

Inspeccionar el modelo de
datos

Comparar con el estado del
modelo de datos después de la
migración anterior

Crear script SQL (y ejecutar, si
lo desea) para actualizar el
esquema de la base de datos

¡Hay un nuevo BookCount int en
la clase de Libro!

EF Core Migrations API

Inspeccionar el modelo de
datos

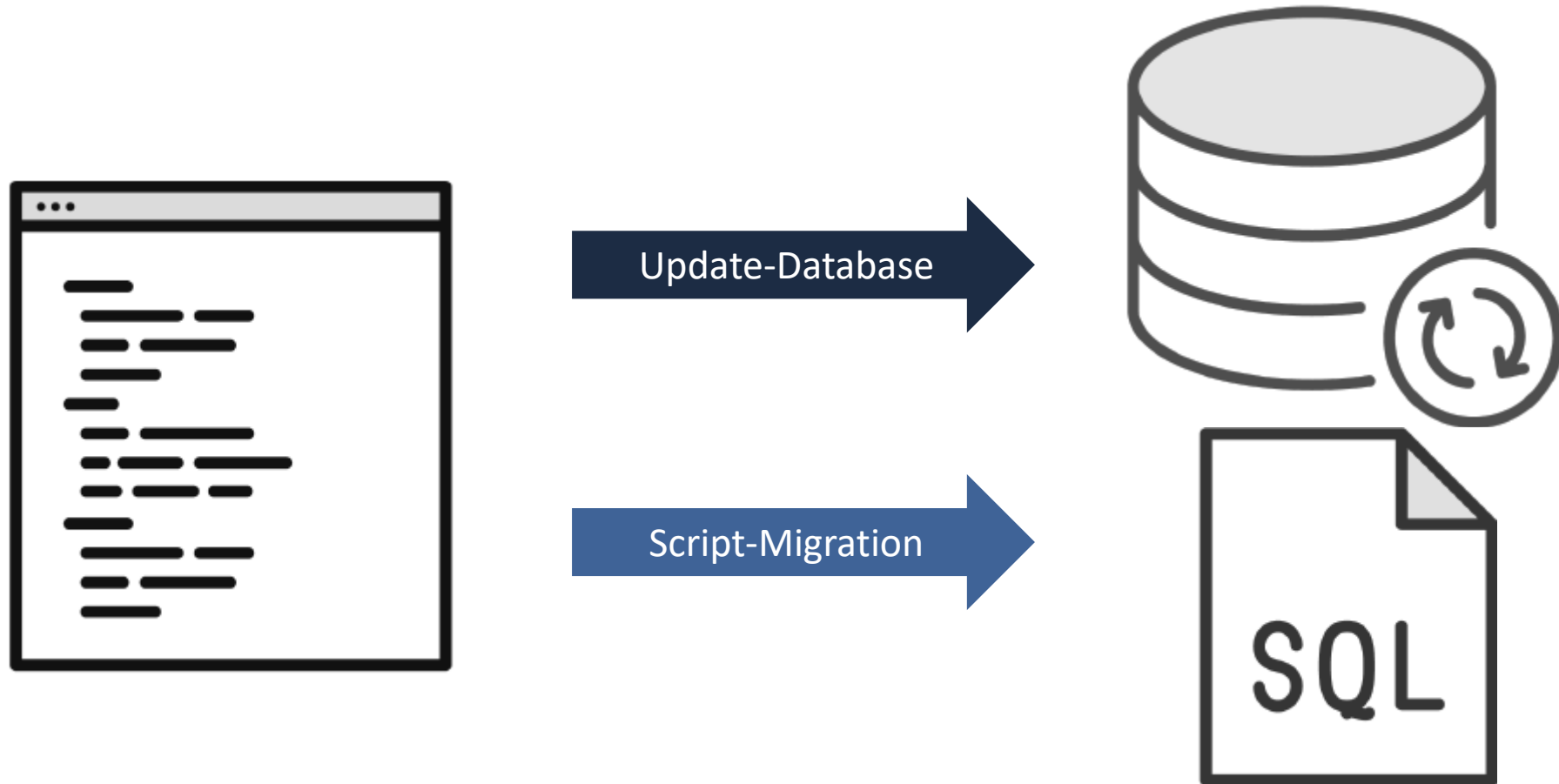
Comparar con el estado del
modelo de datos después de la
migración anterior

Crear script SQL (y ejecutar, si
lo desea) para actualizar el
esquema de la base de datos

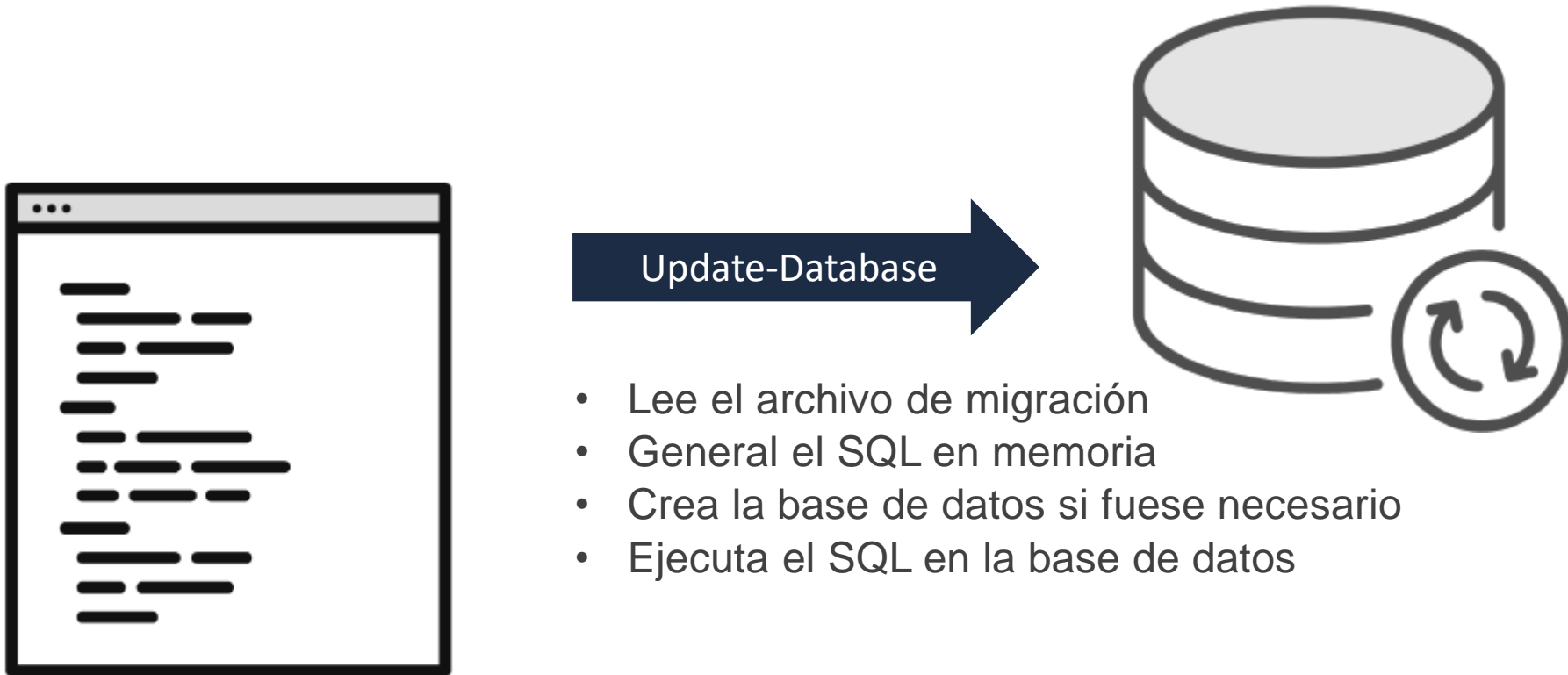
¡Hay un nuevo BookCount int en
la clase de Libro!

Agregar la columna int
BookCount a la tabla Books

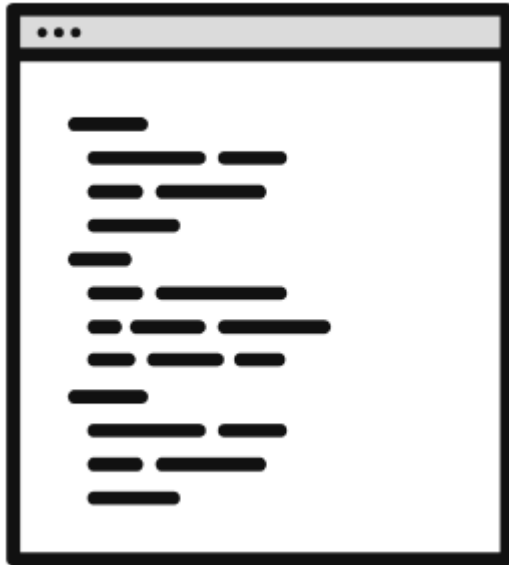
Utilizando migraciones



Utilizando migraciones



Utilizando migraciones



- Lee el archivo de migración
- Genera el SQL
- Por defecto lo muestra en el editor SQL
- Usa parámetros para definir la salida del archivo



Recomendación de migraciones

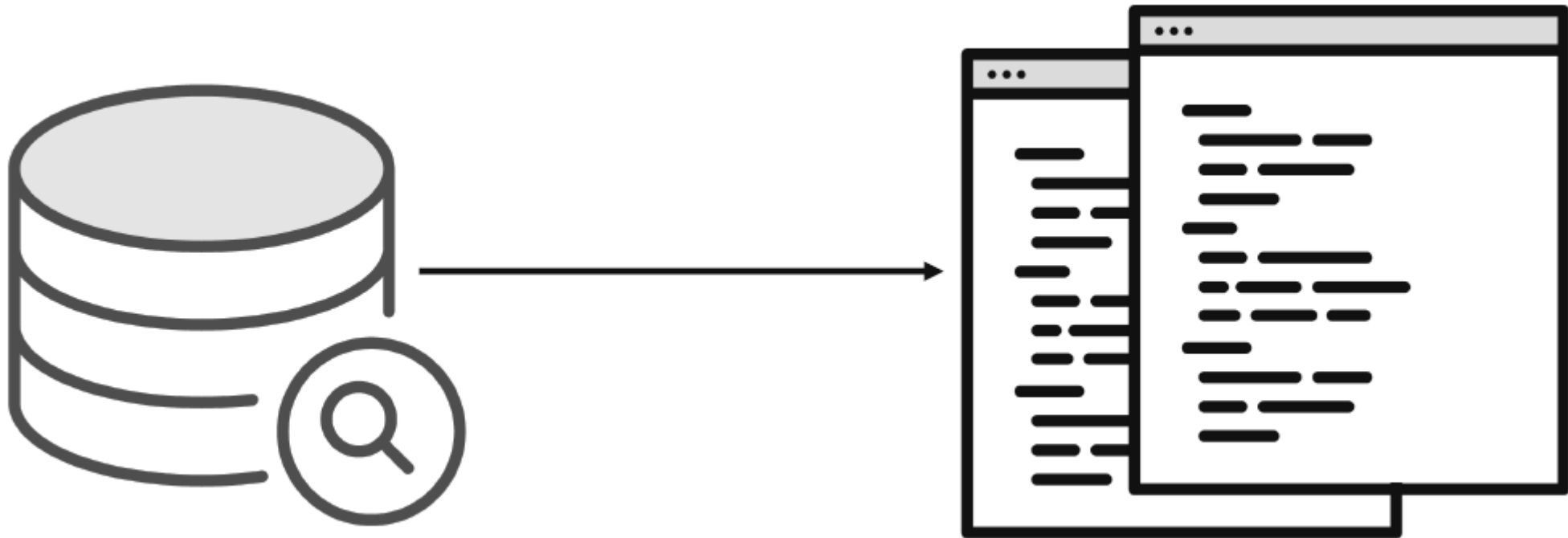


Base de datos de Desarrollo
update-database



Base de datos de Producción
script-migration

Ingeniería Inversa



Scaffold Ingeniería Inversa



Operación única para crear clases de "stake in the ground" y DbContext



A través de la herramienta de línea de comandos con una variedad de opciones



EF Core Power Tools impulsado por la comunidad* proporciona una interfaz de usuario visual

*creado/mantenido por el MVP de confianza de Microsoft, Erik Ejlskov Jensen

Beneficios de EF Core Migrations



Determina el esquema con la misma lógica utilizada para describir el modelo de datos y las asignaciones.



Los archivos de migración pasan a formar parte del control de código fuente



Crear paquetes de migración "compatibles con DevOps" para la implementación/ejecución



Genere scripts SQL e incorpórelos a su propio proceso

LINQ

Consulta integrada de lenguaje (Language Integrated Query). Una característica de .NET desde hace mucho tiempo con una extensión EF Core que, con la ayuda del proveedor de base de datos correspondiente, **transforma esas consultas linq en consultas SQL.**

EF Core's LINQ Query Workflow

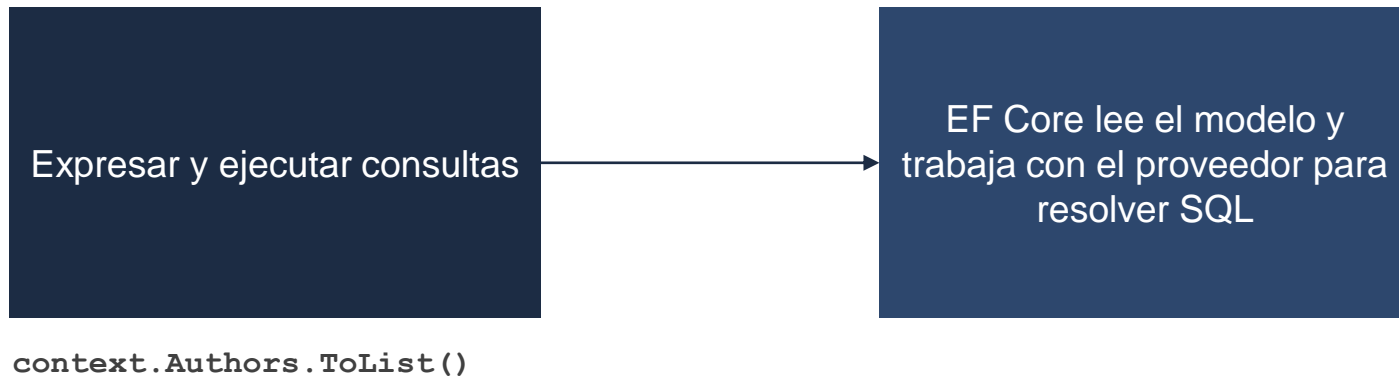
Expresar y ejecutar consultas

EF Core's LINQ Query Workflow

Expresar y ejecutar consultas

```
context.Authors.ToList()
```

EF Core's LINQ Query Workflow



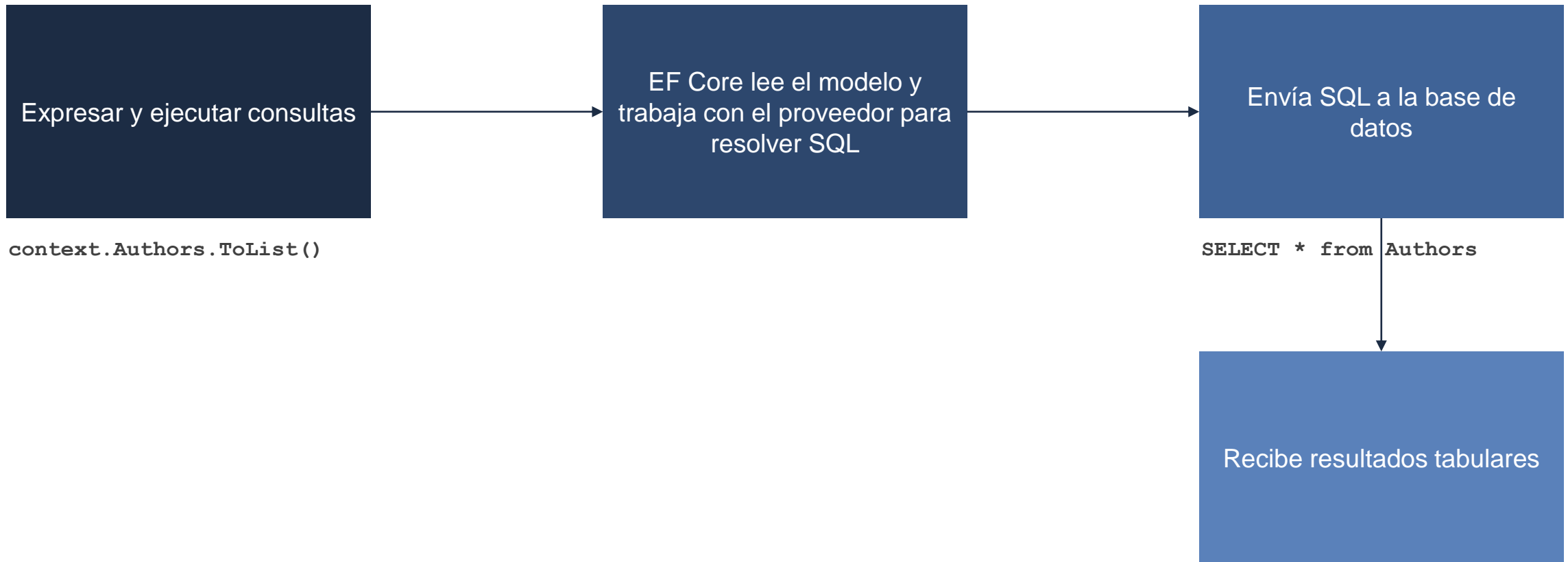
EF Core's LINQ Query Workflow



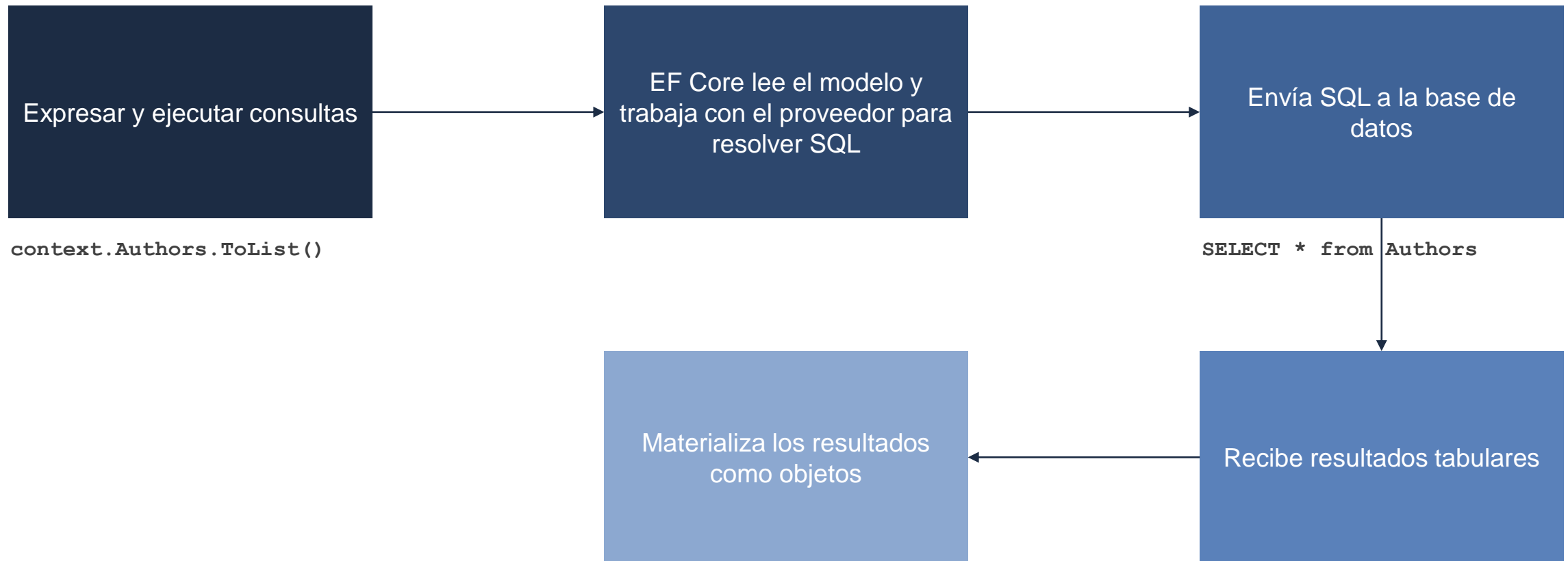
EF Core's LINQ Query Workflow



EF Core's LINQ Query Workflow



EF Core's LINQ Query Workflow



EF Core's LINQ Query Workflow



EF Core's LINQ Query Workflow

The Simplest Query

```
var context=  
    new BigPictureContext();
```

```
List<Author> authors=  
    context.Authors.ToList();
```

```
SELECT [a].[AuthorId],  
       [a].[FirstName], [a].[LastName]  
FROM [Authors] AS [a]
```

- ◀ Instantiate the DbContext class
- ◀ Query the Authors DbSet
- ◀ Execute with LINQ execution method: ToList()
- ◀ The generated SQL projects all columns known to EF Core

LINQ to Entities, algunos métodos

Query Methods

`Where()`
`OrderBy()`
`Skip()`
`Take()` (Skip+Take for paging)
`Join()`
`GroupJoin()`
`GroupBy()`
`Select()` (for projection)
`SelectMany()`

Execution Methods

All with Async counterparts

`ToList()`
`First()`
`FirstOrDefault()`
`Single()`
`SingleOrDefault()`
`Last()`
`LastOrDefault()`
`Count()`
`LongCount()`
`Min()`, `Max()`
`Average()`, `Sum()`
`AsEnumerable()`

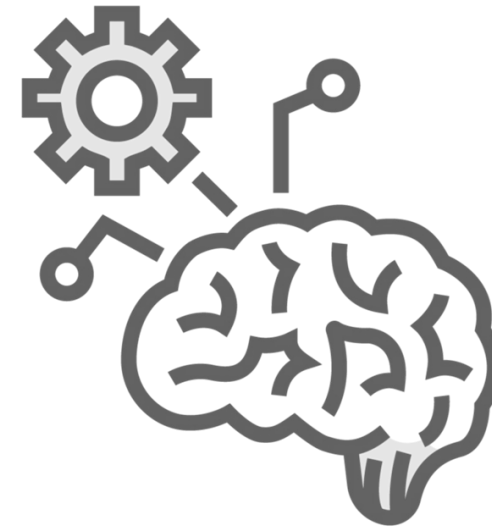
05

Estrategias de uso y características avanzadas de Entity Framework Core

Por convención o por sus propias reglas



Use convenciones
cuando sus necesidades
sean justas y
comunes



Personaliza para
satisfacer tus
necesidades

A veces son las pequeñas cosas

Table Name

```
modelBuilder.Entity<Author>().ToTable("OurBookAuthors");
```

Column Name

```
└─ dbo.OurBookAuthors
  └─ Columns
    └─ AuthorId (PK, int, not null)
    └─ fname (nvarchar(max), not null)
    └─ lname_or_surname (nvarchar(max), not null)
```

```
modelBuilder.Entity<Author>()
    .Property(a => a.LastName).HasColumnName("lname_or_surname");
```

Aplicando Mappings

```
modelBuilder.Entity<Author>()  
    .Property(a => a.LastName)  
    .HasColumnName("lname_or_surname");
```

Fluent API (definir en DbContext)

Todas las asignaciones posibles Solo
son conocido por EF Core

```
[Column("lname_or_surname")]  
public string LastName { get; set; }
```

Data Annotations

(Definir en clase empresarial)

Asignaciones limitadas
El conocimiento de la base de datos se
filtra a la lógica empresarial

Aprovechar procedimientos almacenados

Uso para consultas y
comandos

Ejecutar con
parámetros



GRACIAS
POR SU PREFERENCIA

