

## Resumen LINQ

### Filtrar

```
contexto.Producto.Where(p => p.Categoriald == 1).ToList();
```

### Ordenar

```
contexto.Producto.OrderBy(p => p.Clave).ToList();
```

### Seleccionar solo unos campos

```
contexto.Producto.Select(p => new  
    {  
        p.Clave,  
        p.Nombre  
    }).ToList();
```

### Agrupar

```
from productos in contexto.Producto  
    group productos by productos.Categoriald into p  
    select p).ToList()
```

### In

```
contexto.Producto.Where  
    (p => categoriasId.Contains(p.Categoriald)).ToList()
```

## Inner Join

```
from categorias in contexto.Categoria
    join productos in contexto.Producto
    on categorias.Id equals productos.Categoriald
select new
{ categoria.Id, categoria.Nombre, IdProducto = productos.Id,
  NombreProducto = productos.Nombre
}).ToList()
```

## Left Join

```
(from p in contexto.Producto
  join c in contexto.Categoria
  on p.Categoriald equals c.Id
  into cat
from categoria in cat.DefaultIfEmpty()
  select new
{ Categoria = categoria.Nombre, p.Clave, p.Nombre
}).ToList()
```

## Group Join

```
(from c in contexto.Categoria
  join p in contexto.Producto
    on c.Id equals p.Categoriald into grupo
select new
{
  Categoria = c.Nombre,
  Productos = grupo.Select(g => new { g.Clave, g.Nombre })
}).ToList()
```

## Group Join y Count

```
from categorias in contexto.Categoria
  join productos in contexto.Producto
    on categorias.Id equals productos.Categoriald
    into grupo
select new { Categoria = categorias.Nombre,
  TotalProductos = grupo.Count()}.ToList()
```

Group Join separados por coma

```
(from c in contexto.Categoria
    join p in contexto.Producto
        on c.Id equals p.Categoriald
    into grupo
select new
{
    Categoria = c.Nombre,
    Productos = string.Join(",", grupo.Select(g => g.Nombre))
}).ToList();
```