

# Erronka 1. Aginte koadroa

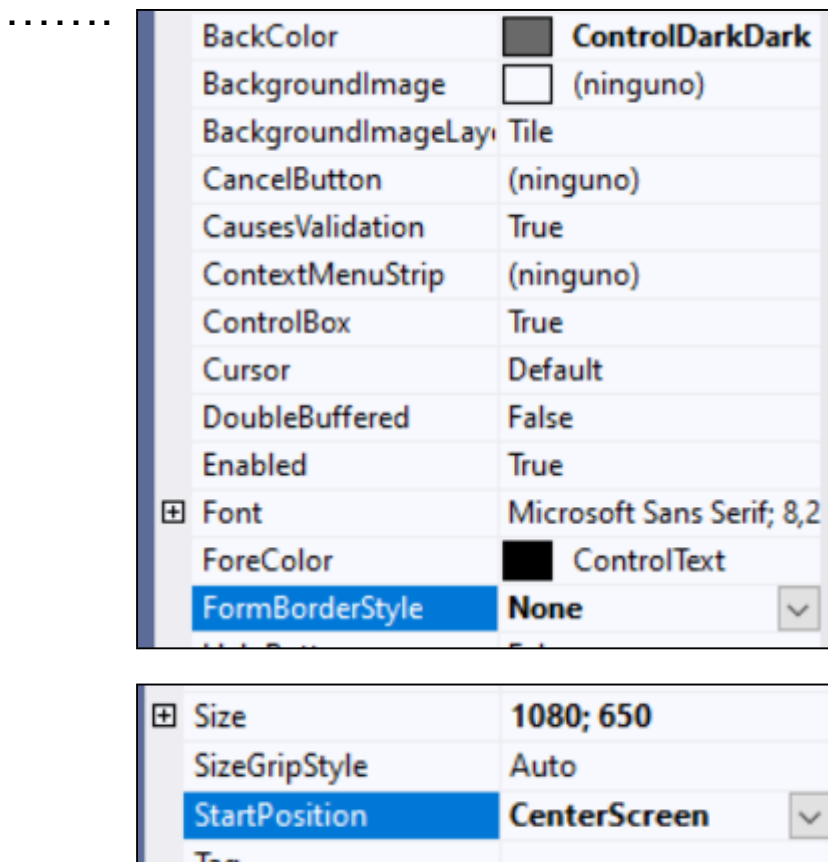
Visual Studio 2022 C# .NET

Idazmahairako interfaze garapena

PAAG 2023-2024 Ikasturtea

# Erronka 1. Aginte koadroa

- Lehenengo aginte koadro simple bat sortuko dugu
  - Aplikazio berria: **AginteKoadroa**
- **Aplicación de Windows Forms (.NET Framework)**
- Estilo berri batzuk jarri: BackColor, FormBorderStyle, Size, StartPosition,



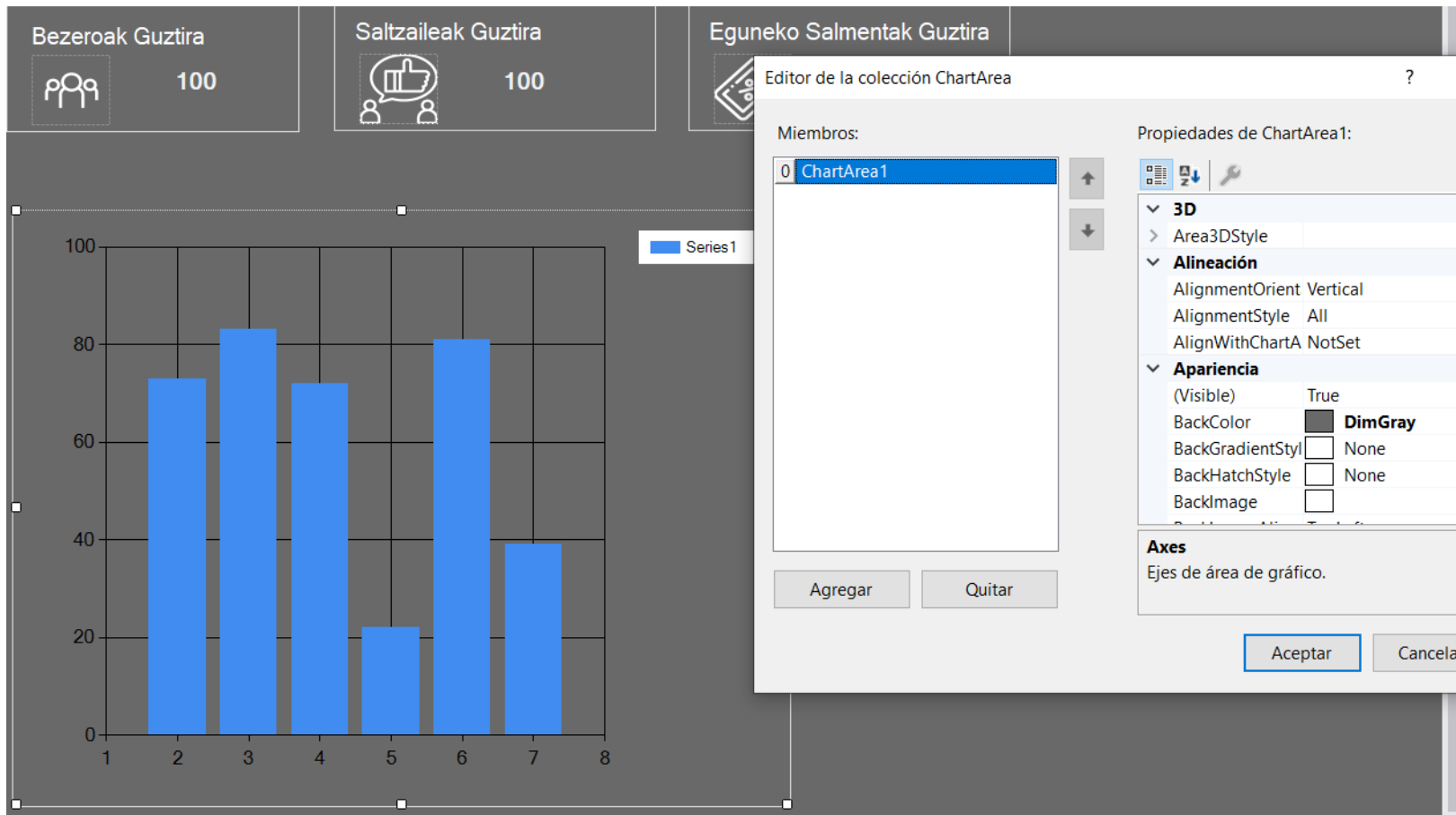
## Erronka 1. Aginte koadroa

- Osagai batzuk jarriko ditugu: etiketak, botoiak, groupbox
- Groupbox barruan: etiketa -> «Bezeroak Guztira»; picturebox -> irudia aukeratu + Modo de tamaño: «StretchImage»; etiketa -> 100
- Botoi bat aplikazio uzteko irudi batekin
- Lehenengo groupbox-a eginda beste bi kopiatuko dira: «Saltzaileak Guztira», «Eguneko Salmentak Guztira»



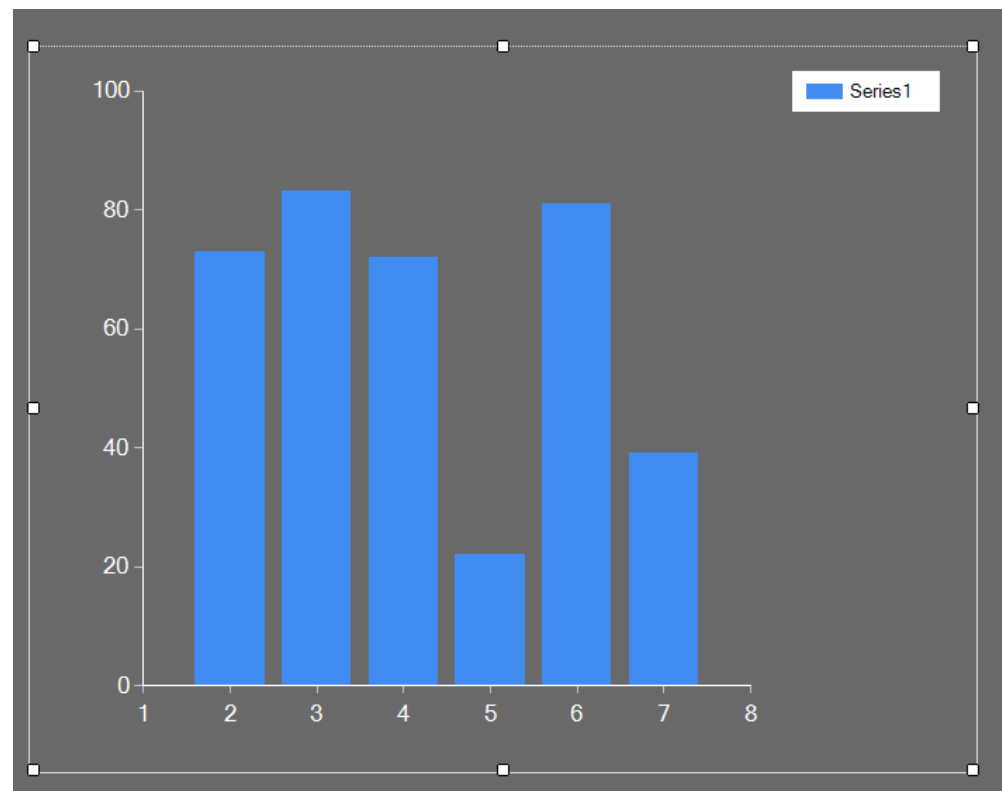
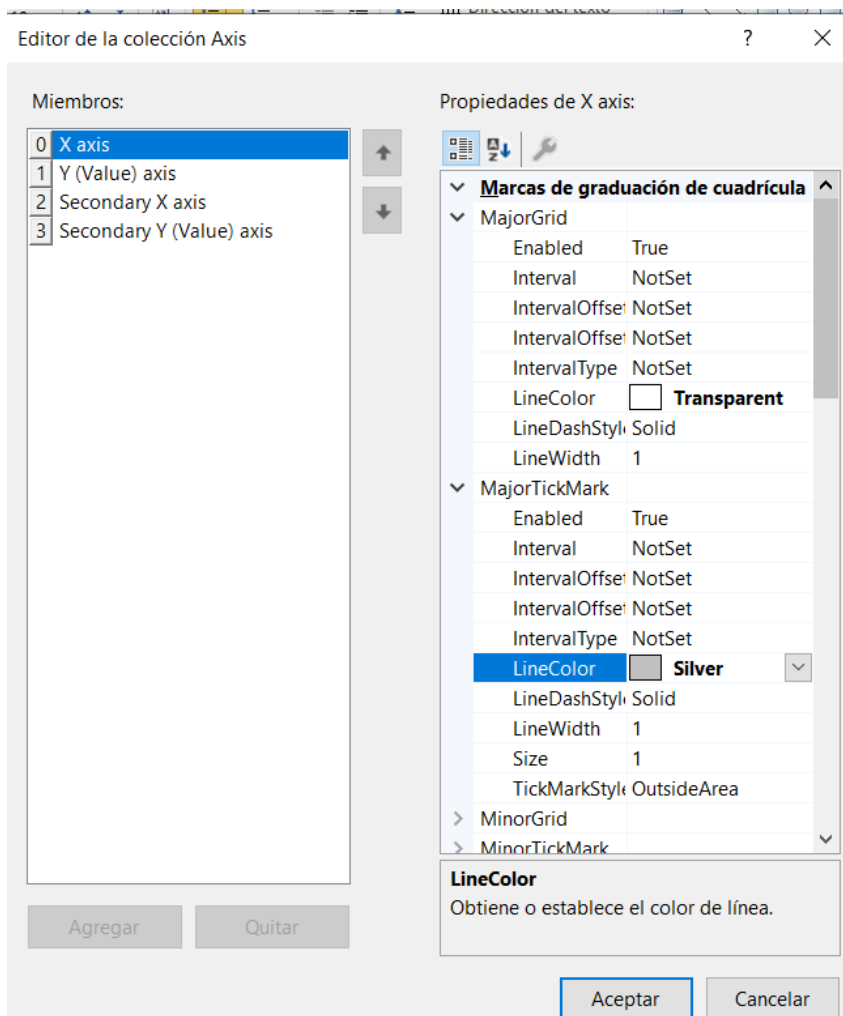
# Erronka 1. Aginte koadroa

- Behean beste **groupbox** sortuko dugu eta Datos multzoan dagoen **Chart** bat sartuko dugu:
  - Anchor erabiliko dugu behean ondo kokatuta uzteko osagai biak
  - Chart atzealdeko kolorea gris iluna
  - **ChartArea** aldatuko da



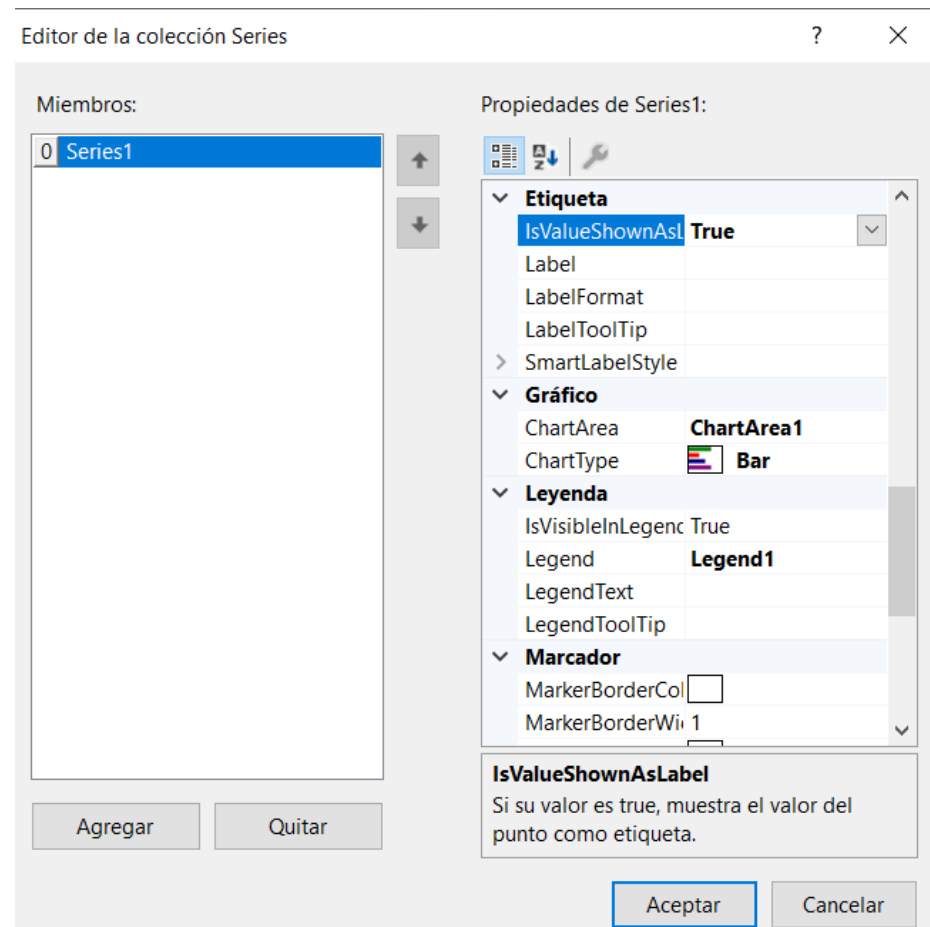
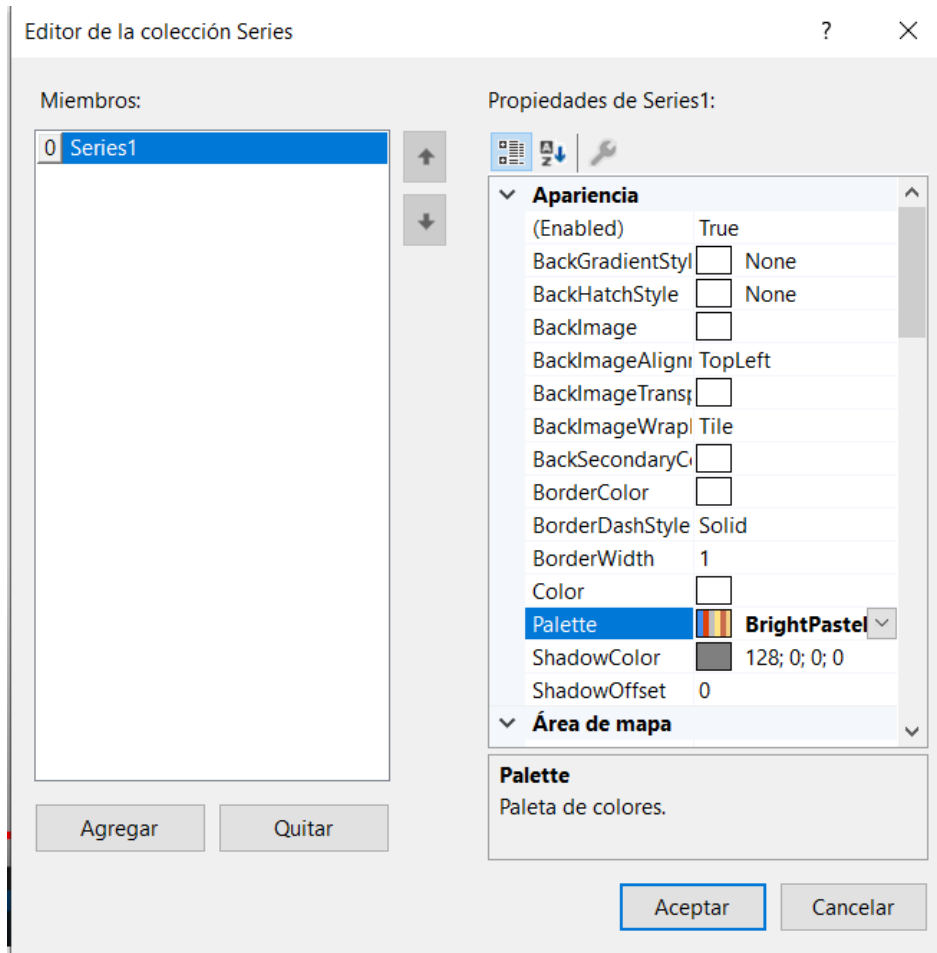
# Erronka 1. Aginte koadroa

- «Ejes – Axes (ChartAreas Kolekzio barruan)» aukeratu beste propietate batzuk aldatzeko: MajorGrid, MajorTickMark (x eta y). Baita zenbakien eta lerroen kolorea (Apariencia, Etiquetas)

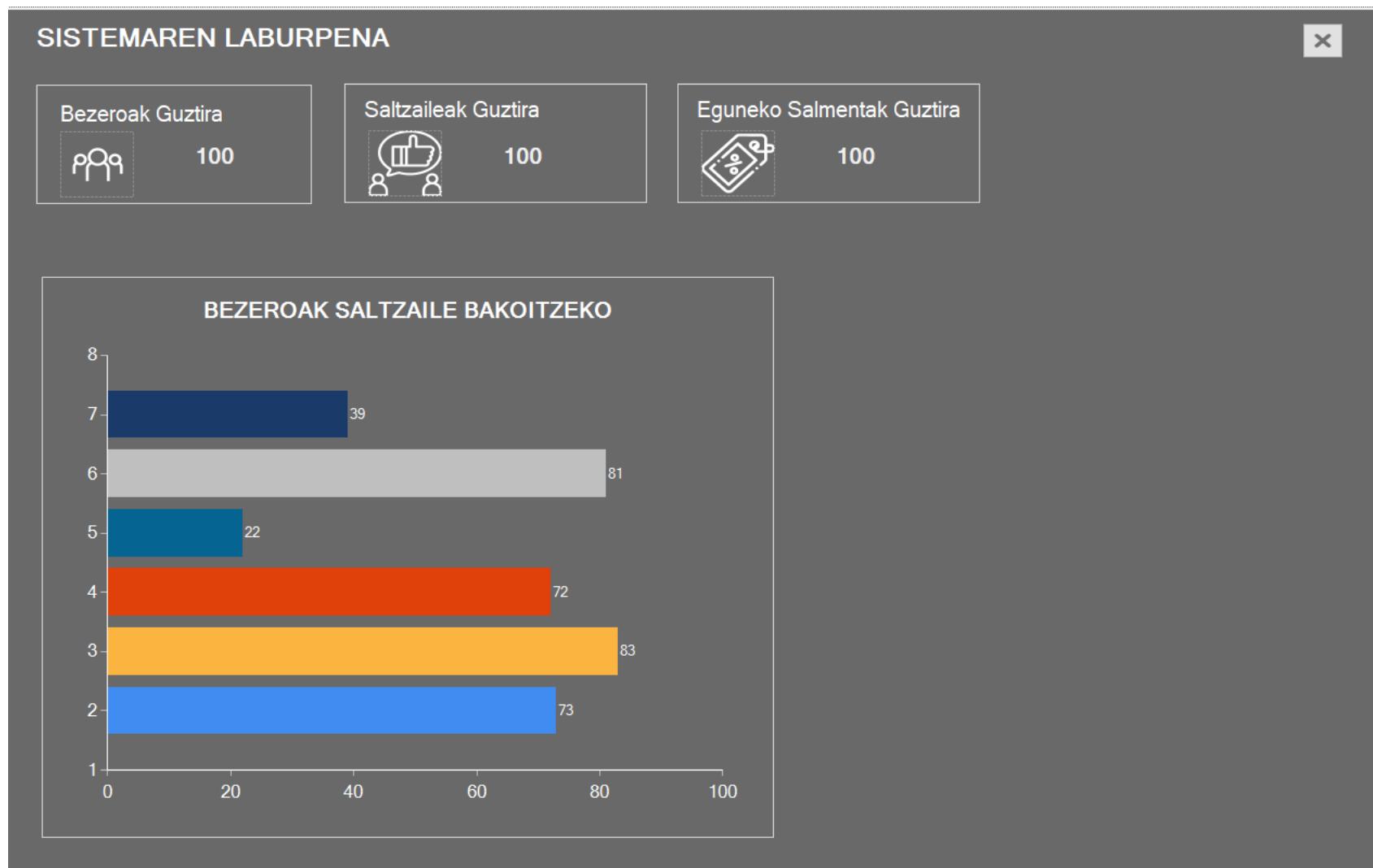


# Erronka 1. Aginte koadroa

- «Series» aldatu: Palette, ChartType, IsValueShownAsLabel, Aspecto de Etiqueta
- «Titles - agregar» titulu bat jartzeko
- «Legends» aldatu: (Enabled)=False



# Erronka 1. Aginte koadroa



# Erronka 1. Aginte koadroa

**ORM bat instalatuko dugu gure datu basea erabiltzeko:**

## Entity Framework



Entity Framework ORM (Object Relational Mapper) bat da, .NET erabiltzen duten programatzaileei datu baseen konplexutasunak ahaztu eragiten dituen eta zuzenean domeinuaren eredu objektuekin lan egitea errazten duena.

Horrela, programatzaileek aplikazioaren logikan fokalizatzen dira eta ez datuak non gordetzen diren, edo nola mapeatzen diren. Hori dena, EF-k egingo du.

Datu Base ezberdinak erabili daitezke, ORM-a arduratuko baita azpitik dagoen kudeaketa guztiarekin. (SQL Server, MySQL, Oracle, Postgresql,.....)

Gure ariketan, **Postgresql** erabiliko dugu.

NuGet erabiliz bi pakete instalatuko ditugu (Proyecto – Administrar paquetes NuGet):

	<b>EntityFramework</b> por Microsoft	6.4.0
	Entity Framework 6 (EF6) is a tried and tested object-relational mapper for .NET with many years of feature development and stabilization.	6.4.4
	<b>EntityFramework6.Npgsql</b> por Shay Rojansky, Emil Lenngren, Franc	6.4.3
	PostgreSQL provider for Entity Framework 6	



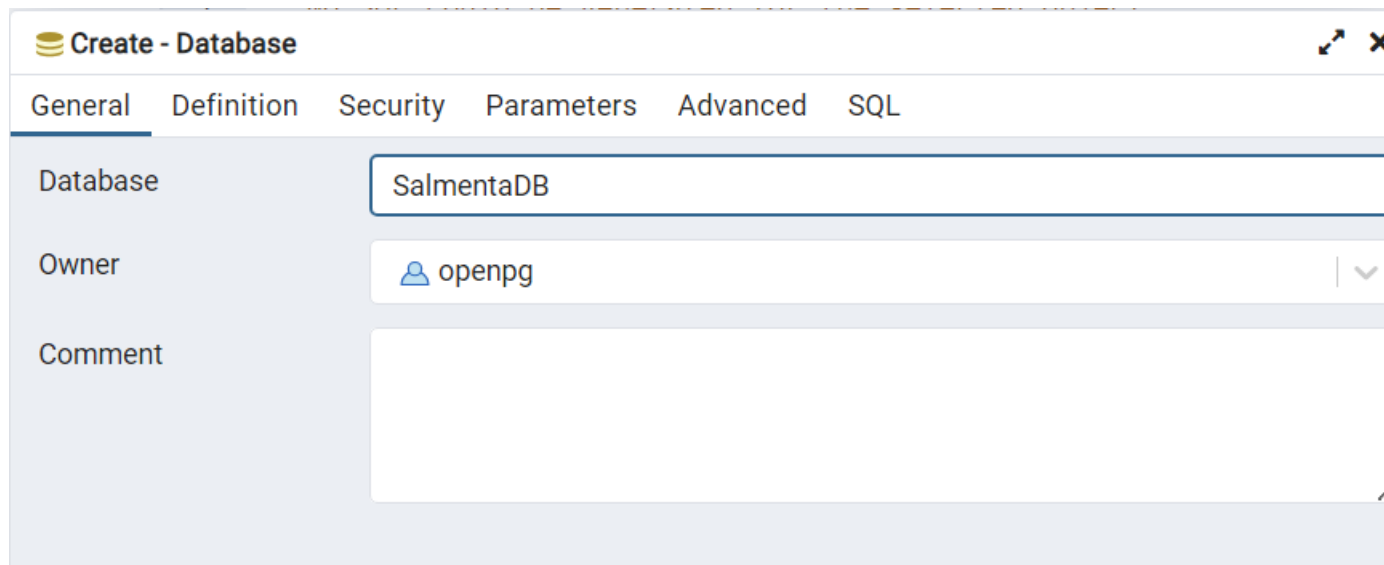
# Erronka 1. Aginte koadroa

- **PgAdmin** (postgresql administratzeko interfaze grafikoa) bezalako tresna bat erabilita Datu Base bat sortuko dugu **Postgresql-an: SalmentaDB**, hiru taulekin,

## **Bezeroa, Saltzailea, Salmenta.**

- Saltzaile 1 -> n bezero ditu
- Bezero 1 -> n salmenta ditu.

Lehenengo **SalmentaDB** sortu eskuz:

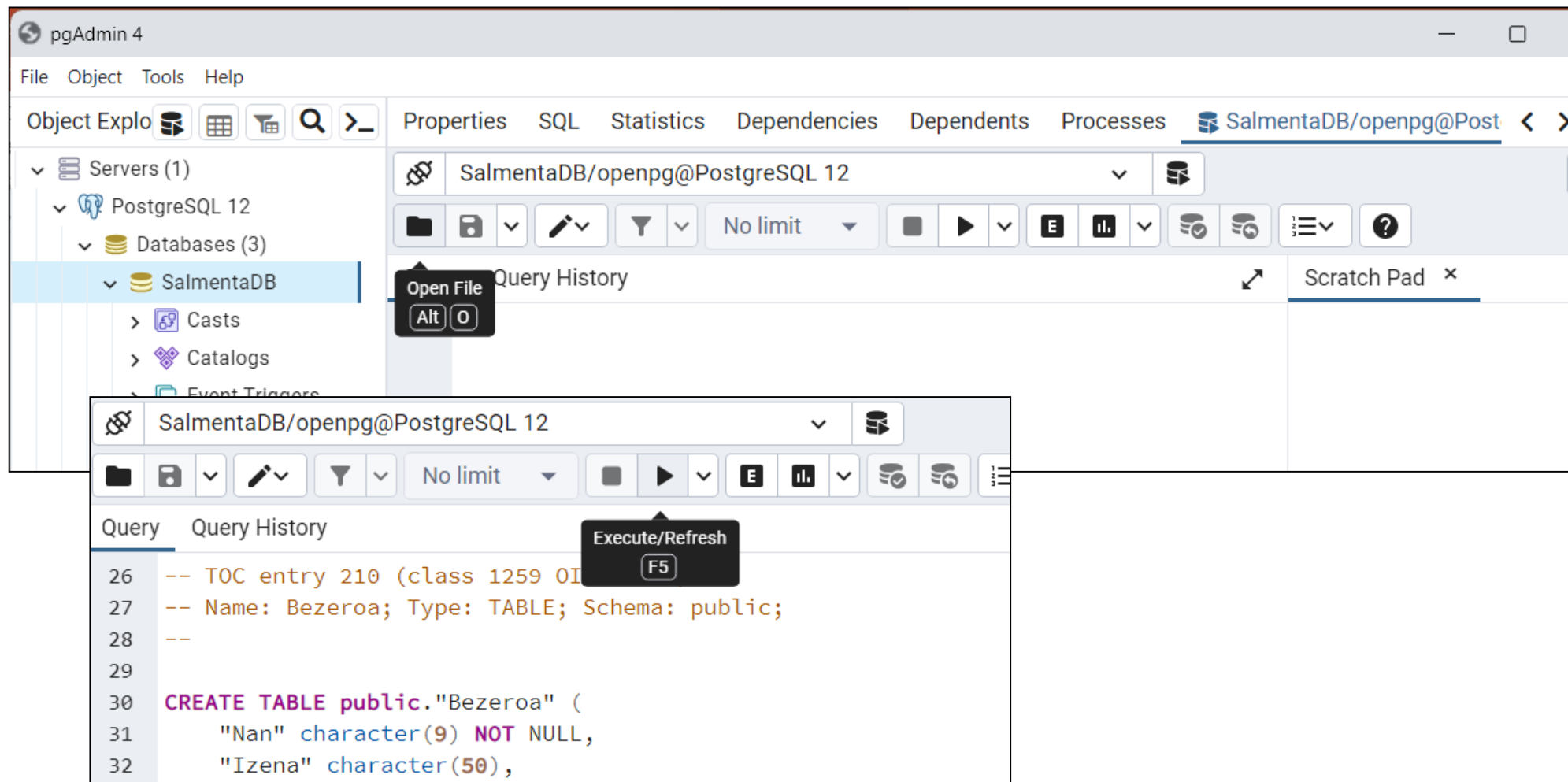


The image shows a screenshot of the 'Create - Database' dialog box in PgAdmin. The dialog has a title bar with a PostgreSQL icon and the text 'Create - Database'. Below the title bar are tabs for 'General', 'Definition', 'Security', 'Parameters', 'Advanced', and 'SQL'. The 'General' tab is selected. It contains three fields: 'Database' with the value 'SalmentaDB', 'Owner' with a dropdown menu showing 'openpg', and 'Comment' with a large empty text area. There are also icons for window control (maximize, close) in the top right and a refresh icon in the bottom right.

# Erronka 1. Aginte koadroa

Hurrengo diapositiban dagoen egitura sortu behar da eta horretarako Moodle-n dagoen **SalmentaDb.sql** fitxategia inportatu behar da:

## Tools – Query Tool – Open File (fitxategia aukeratu) - Execute



# Erronka 1. Aginte koadroa

**Saltzailea**

General **Columns** Advanced Constraints Parameters Security SQL

Inherited from table(s)

Columns +

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	Nan	character	9		<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
	Izena	character	50		<input type="checkbox"/> No	<input type="checkbox"/> No
	Taldea	character	10		<input type="checkbox"/> No	<input type="checkbox"/> No

Datu basea **freskatu** taula berriak ikusteko:  
taulan egitura konprobatu eta bakoitzaren  
gainean eskuineko botoiarekin **View/ Edit Data**  
–All Rows

**Bezeroa**

General **Columns** Advanced Constraints Parameters Security SQL

Inherited from table(s)

Columns +

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	Nan	character	9		<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
	Izena	character	50		<input type="checkbox"/> No	<input type="checkbox"/> No
	Helbidea	character	50		<input type="checkbox"/> No	<input type="checkbox"/> No
	Telf	character	9		<input type="checkbox"/> No	<input type="checkbox"/> No
	Emaila	character	50		<input type="checkbox"/> No	<input type="checkbox"/> No
	NanSaltzailea	character	9		<input type="checkbox"/> No	<input type="checkbox"/> No

**Salmenta**

General **Columns** Advanced Constraints Parameters Security SQL

Inherited from table(s)

Columns +

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	Id	bigint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
	Data	date			<input type="checkbox"/> No	<input type="checkbox"/> No
	Zenbatekoa	numeric	9	2	<input type="checkbox"/> No	<input type="checkbox"/> No
	NanBezeroa	character	9		<input type="checkbox"/> No	<input type="checkbox"/> No

# Erronka 1. Aginte koadroa

Datuak erabiltzeko prestatuko dugu:

- **App.config** aldatu konexioaren informazioa gehitzeko:

```
<connectionStrings>
<add name="SalmentaDbContext"
connectionString="Server=localhost;port=5433;Database=SalmentaDB;Username=openpg;Password=openpgpwd;" providerName="Npgsql" />
</connectionStrings>
<entityFramework>
  <providers>
    <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    <provider invariantName="Npgsql" type="Npgsql.NpgsqlServices, EntityFramework6.Npgsql" />
  </providers>
  <defaultConnectionFactory type="Npgsql.NpgsqlConnectionFactory, EntityFramework6.Npgsql" />
</entityFramework>
```

Username:openpg -> odoo-k sortu duen erabiltzaile nagusia da.

Password: openpgpwd

# Erronka 1. Aginte koadroa

Datuak erabiltzeko prestatuko dugu:

- **Models** karpeta sortu eta barruan EF-k erabiltzeko kontestua eta ereduak sortuko ditugu :

## SalmentaDbContext.cs

```
using System.Data.Entity;
using System.Data.Entity.ModelConfiguration.Conventions;

namespace AginteKoadroa_PG.Models
{
    public class SalmentaDbContext : DbContext
    {
        public SalmentaDbContext() : base(nameOrConnectionString: "SalmentaDbContext")
        {}

        public DbSet<Bezera> Bezera { get; set; }
        public DbSet<Salmenta> Salmenta { get; set; }
        public DbSet<Saltzailea> Saltzailea { get; set; }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            modelBuilder.HasDefaultSchema("public");
            modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
        }
    }
}
```

# Erronka 1. Aginte koadroa

Datuak erabiltzeko prestatuko dugu:

## **Bezeroa.cs**

```
using System.ComponentModel.DataAnnotations;
namespace AginteKoadroa_PG.Models
{
    public partial class Bezeroa
    {
        [Key]
        public string Id { get; set; }
        public string Izena { get; set; }
        public string Helbidea { get; set; }
        public string Telf { get; set; }
        public string Emaila { get; set; }
        public string Saltzaileald { get; set; }
        public virtual Saltzailea Saltzailea { get; set; }
        public virtual List<Salmenta> Salmenta { get; set; }
    }
}
```

## **Saltzailea.cs**

```
using System.ComponentModel.DataAnnotations;
namespace AginteKoadroa_PG.Models
{
    public partial class Saltzailea
    {
        [Key]
        public string Id { get; set; }
        public string Izena { get; set; }
        public string Taldea { get; set; }
        public virtual List<Bezeroa> Bezeroa { get; set; }
    }
}
```

## **Salmenta.cs**

```
using System.ComponentModel.DataAnnotations;
namespace AginteKoadroa_PG.Models
{
    public partial class Salmenta
    {
        [Key]
        public int Id { get; set; }
        public DateTime Data { get; set; }
        public decimal Zenbatekoa { get; set; }
        public string Bezeroald { get; set; }
        public virtual Bezeroa Bezeroa { get; set; }
    }
}
```

# Erronka 1. Aginte koadroa

Datuak erabiltzeko prestatuko dugu:

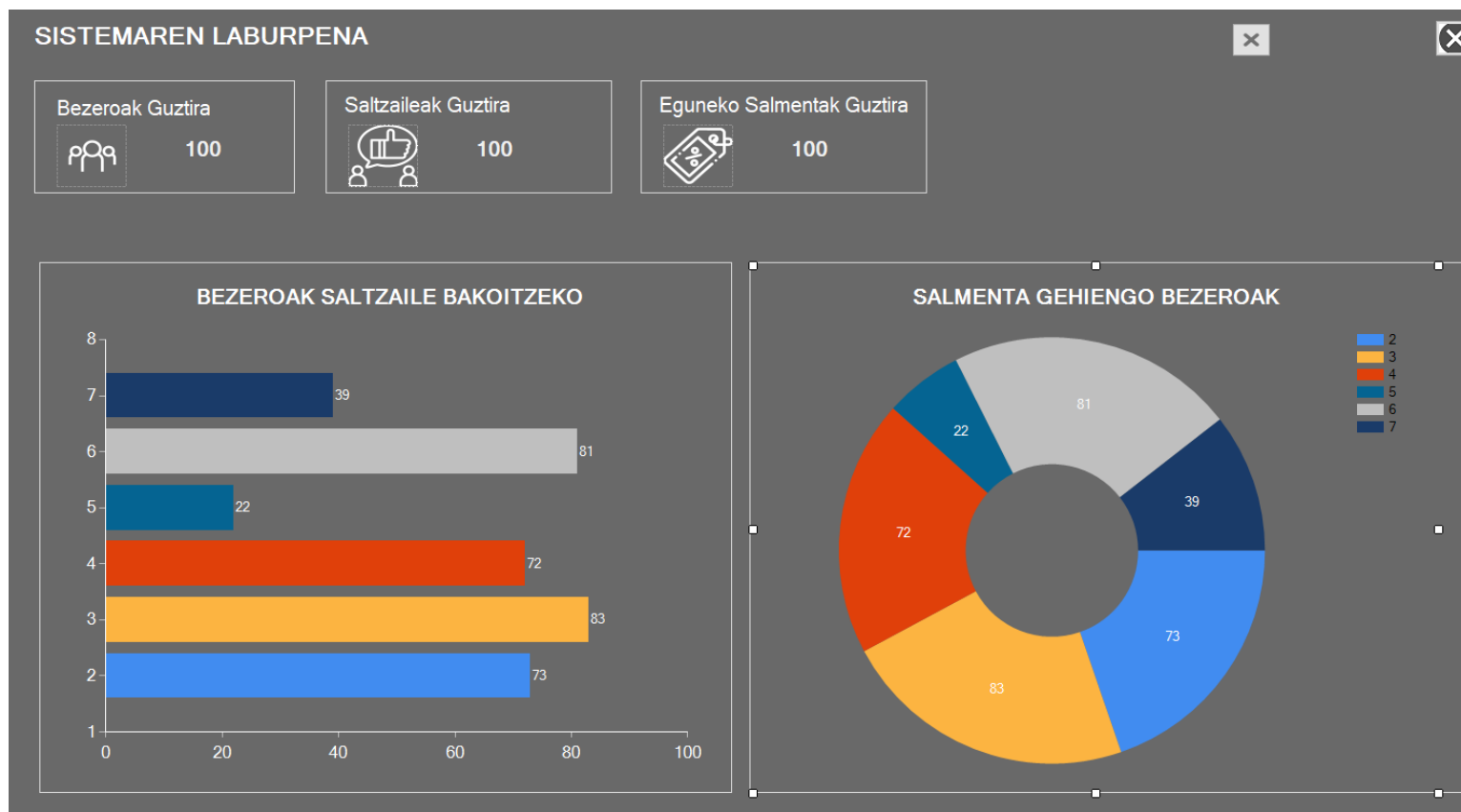
- **Form1** formularioaren «**load**» ebentoa prestatu datuak irakurtzeko:

```
private void Form1_Load(object sender, EventArgs e)
{
    using (var db = new SalmentaDbContext())
    {
        var bezeroaData = db.Bezeroa
            .Include("Saltzailea")
            .GroupBy(b => b.Saltzailea.Izena)
            .ToDictionary(g => g.Key, g => g.Count());

        if (bezeroaData != null)
        {
            if (bezeroaData.Count > 0)
            {
                chart1.DataSource = bezeroaData;
                chart1.Series[0].YValueMembers = "Value";
                chart1.Series[0].XValueMember = "Key";
                chart1.DataBind();
            }
        }
    }
}
```

# Erronka 1. Aginte koadroa

- Beste grafiko bat gehitu. Bi modutara egin daiteke:
  - lehenengotik kopiatuta eta aldaketak egin itxura ezberdina edukitzeko: Titles, Series(Etiqueta: IsValueShownAsLabel->True, Leyenda: IsVisibleInLegend->True), Legends.
  - Aurretik sortuta daukagun kontrol pertsonalizatu bat erabili





## Erronka 1. Aginte koadroa

- Bigarren grafikoa jartzeko **kontrol pertsonalizatu bat** sortuko dugu eta gero, proiektu honetan txertatu

```
private void Form1_Load(object sender, EventArgs e)

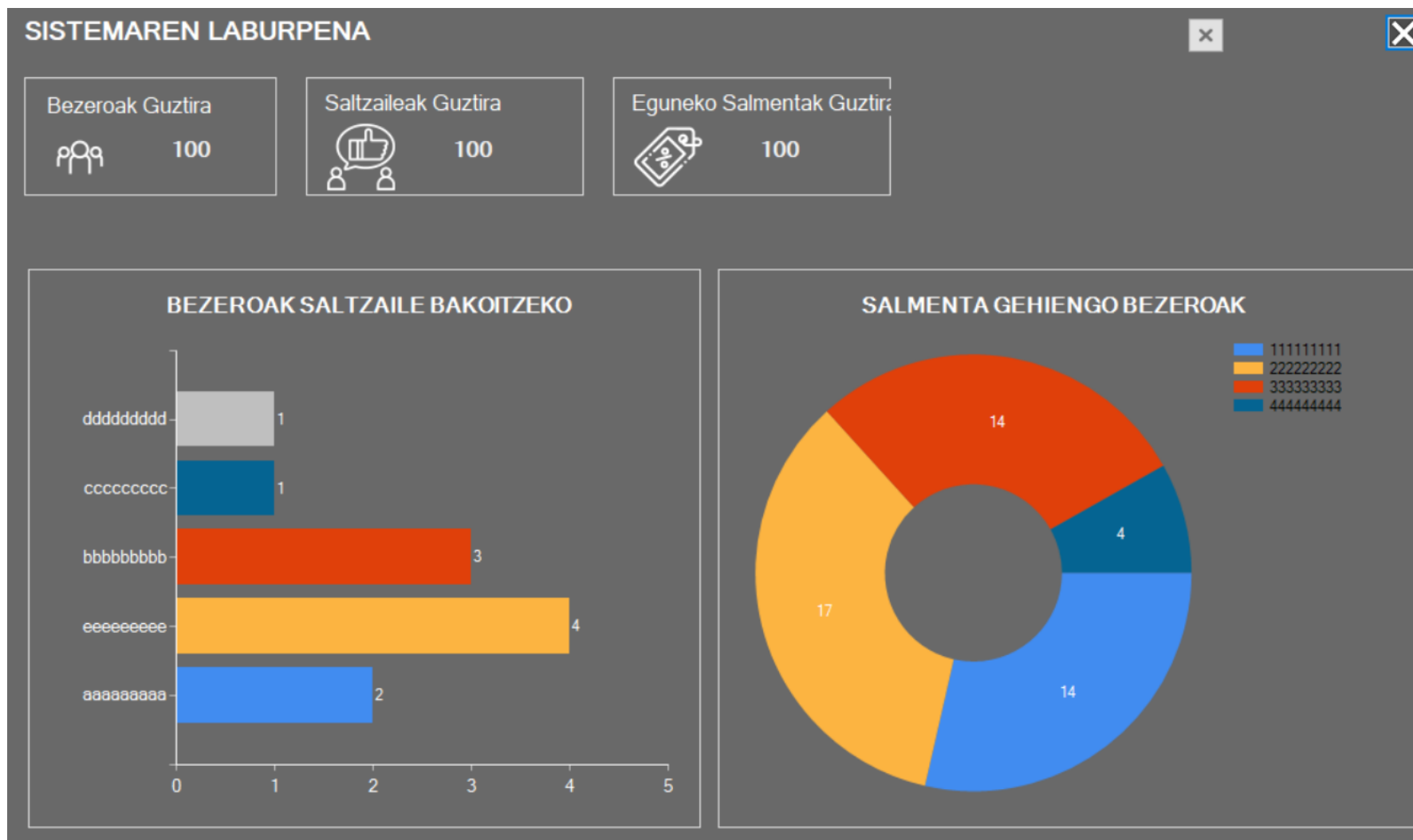
    using (var db = new SalmentaDbContext())

    {
        .....

        //bigarren grafikoa
        var salmentaData = db.Salmenta
            .Include("Bezeroa")
            .GroupBy(b => b.Bezeroa.Izena)
            .ToDictionary(g => g.Key, g => g.Sum(b => b.Zenbatekoa));

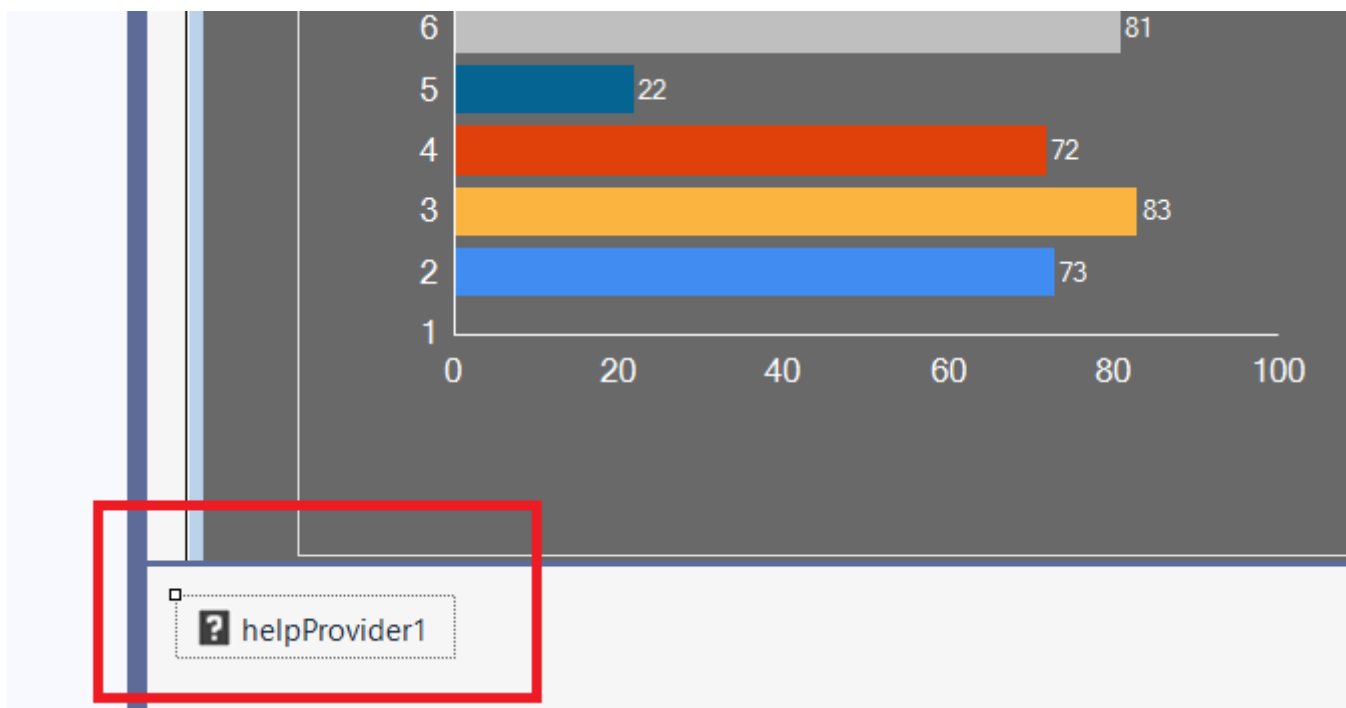
        if (salmentaData != null)
        {
            if (salmentaData.Count > 0)
            {
                var kontrolak = grafikoa1.Controls.OfType<System.Windows.Forms.DataVisualization.Charting.Chart>();
                foreach (var kontrola in kontrolak)
                {
                    kontrola.Titles[0].Text = "SALMENTA GEHIENGO BEZEROAK";
                    kontrola.DataSource = salmentaData;
                    kontrola.Series[0].YValueMembers = "Value";
                    kontrola.Series[0].XValueMember = "Key";
                    kontrola.DataBind();
                }
            }
        }
    }
```

# Erronka 1. Aginte koadroa



## Erronka 1. Aginte koadroa

- Laguntza mezuak prestatuko ditugu. Horretarako **HelpProvider** kontrola erabiliko dugu.
- Formularioa aukeratu eta «propiedades» - «HelpButton = True, MaximizeBox = False, MinimizeBox = False
- HelpProvider kontrola formulario barrura gehitu (behean agertuko da)



## Erronka 1. Aginte koadroa

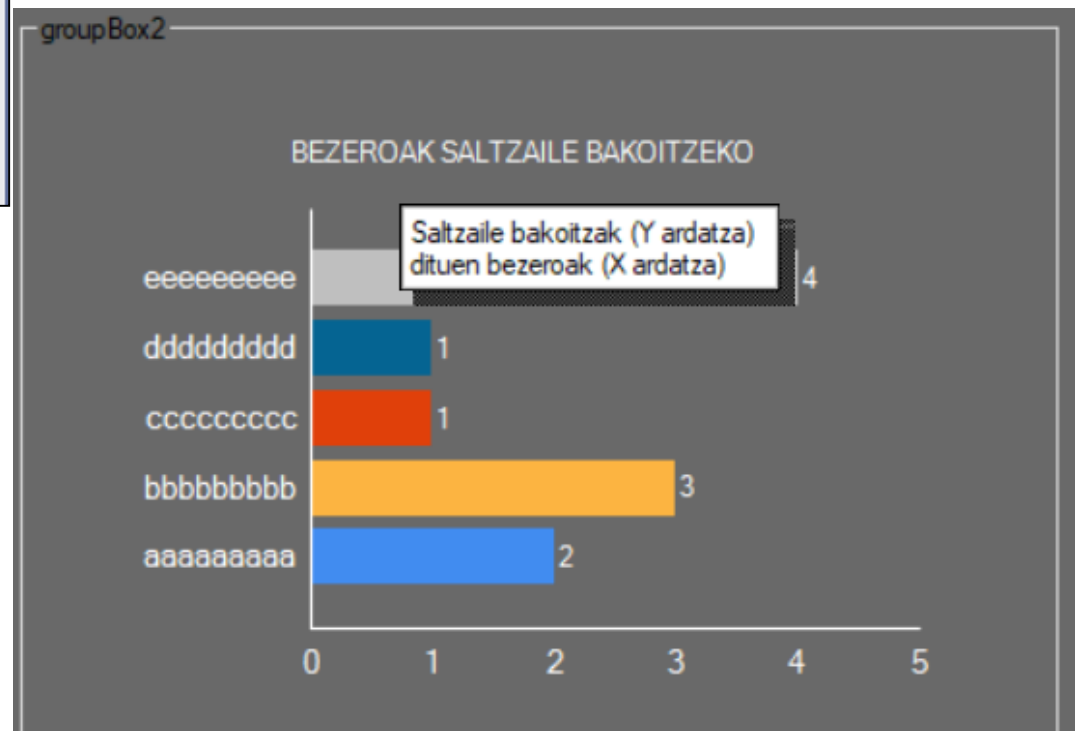
- Orain laguntza mezua gehituko dugu nahi dugun osagaietan, adibidez, grafiko bietan:
  - Aukeratu lehenengo grafikoa eta propietatea aldatu: «Saltzaile bakoitzak (Y ardatza) dituen bezeroak (X ardatza)»

Enabled	True
GenerateMember	True
HelpKeyword en helpProvide	
HelpNavigator en helpProvic	AssociateIndex
HelpString en helpProvider1	Saltzaile bakoitzak (Y ardatza)
ImeMode	NoControl

Martxan ikusteko leihoaren galdera ikurrean klik egin eta gero grafikoaren gainean

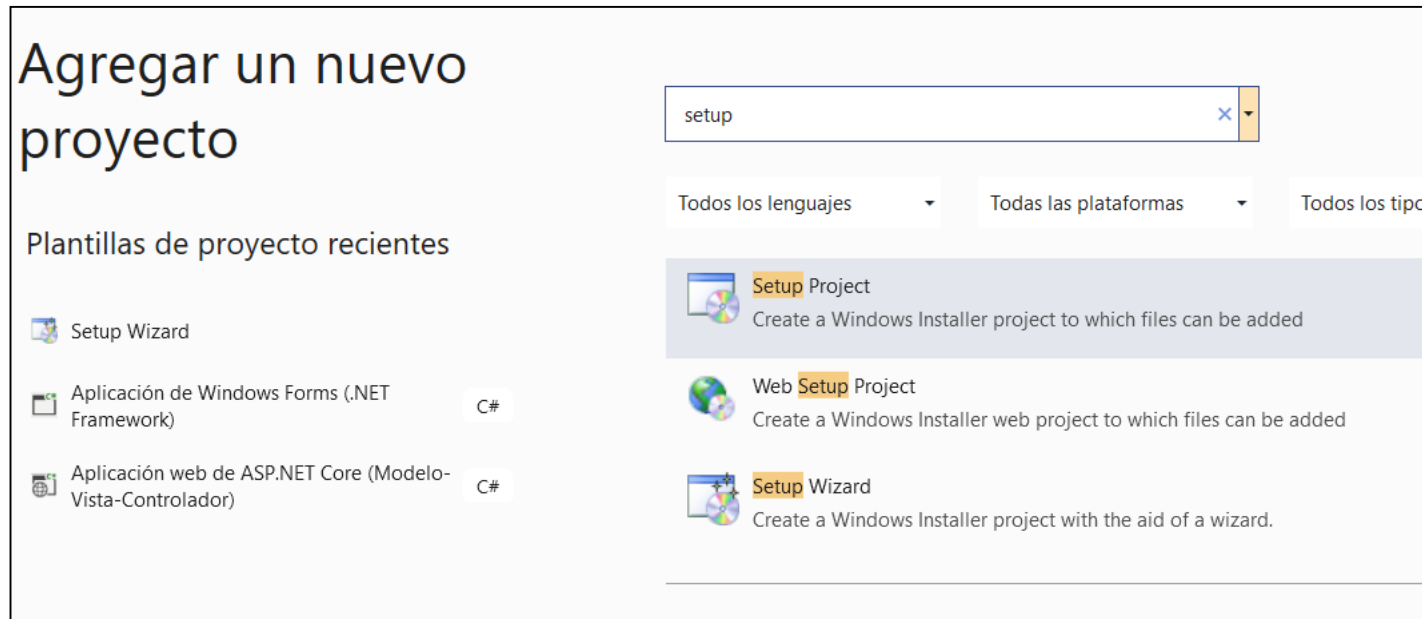
HelpProvider kontrolak **.chm** luzapena duen fitxategi bat erabili dezake laguntza emateko

Antzerako beste kontrol bat dago : **ToolTip (baina sagua gaineretik pasaten denean)**



# Erronka 1. Aginte koadroa

- Aplikazioa prestatzeko banaketarako instaladorea jarriko dugu: «Microsoft Installer» topatu <https://marketplace.visualstudio.com> web orrian edo zuzenean aplikazioko menuan – Extensiones – Administrar extensiones (En línea – Visual Studio Marketplace)
- Instalatuta eta gero: proiektu berria gehitu gure soluzio barruan eta «setup» motako proiektuak topatu (asistentea daukadana erabil daiteke)



# Erronka 1. Aginte koadroa

- Asistentearen pausuak jarraitu eta bukatu. Gero «Application Folder» eskuineko botoiarekin gehitu «Resultados del proyecto»

The screenshot shows the Visual Studio IDE with the 'File System (Setup1)' window on the left. The 'Application Folder' is selected. A context menu is open over the 'Application Folder', showing options like 'Add', 'Crear nuevo acceso directo', 'View', 'Cortar', and 'Pegar'. The 'Agregar grupo de resultados del proyecto' dialog box is open on the right. It shows the project 'AginteKoadroa' and a list of result groups including 'Resultado principal', 'Recursos adaptados', 'Símbolos de depuración', 'Archivos de contenido', 'Archivos de código fuente', 'Archivos de documentación', and 'Ensamblados de serialización XML'. The 'Configuración' is set to '(Activo)' and the 'Descripción' is 'Contiene los archivos DLL o EXE generados por el proyecto.'.

- Sarbide zuzena ere prestatzeko esango diogu eta «User's Desktop» barrura arrastatuko dugu

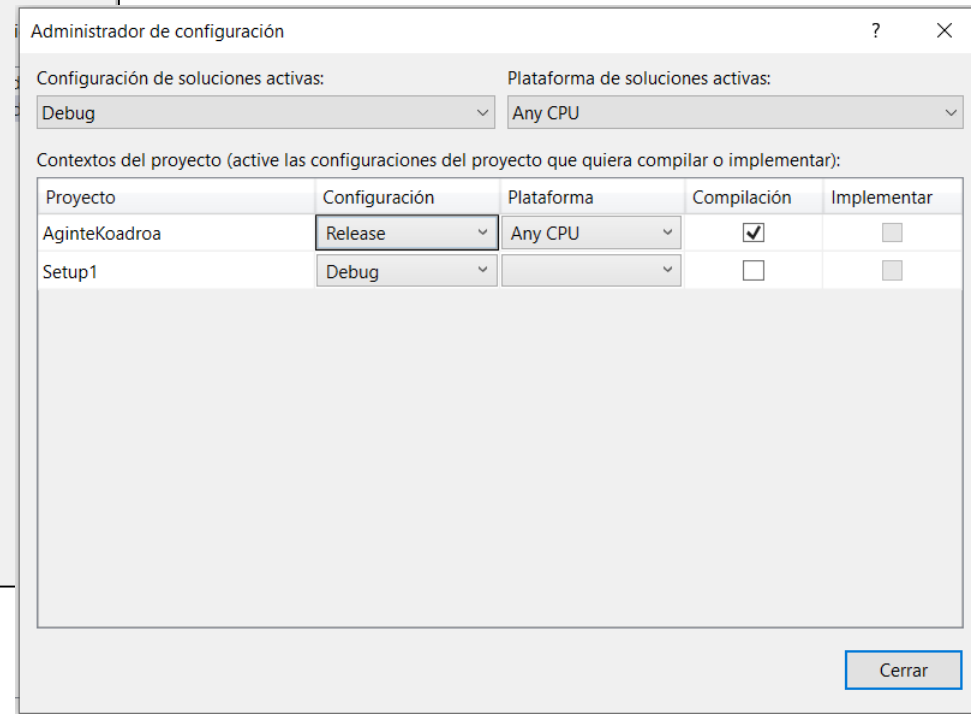
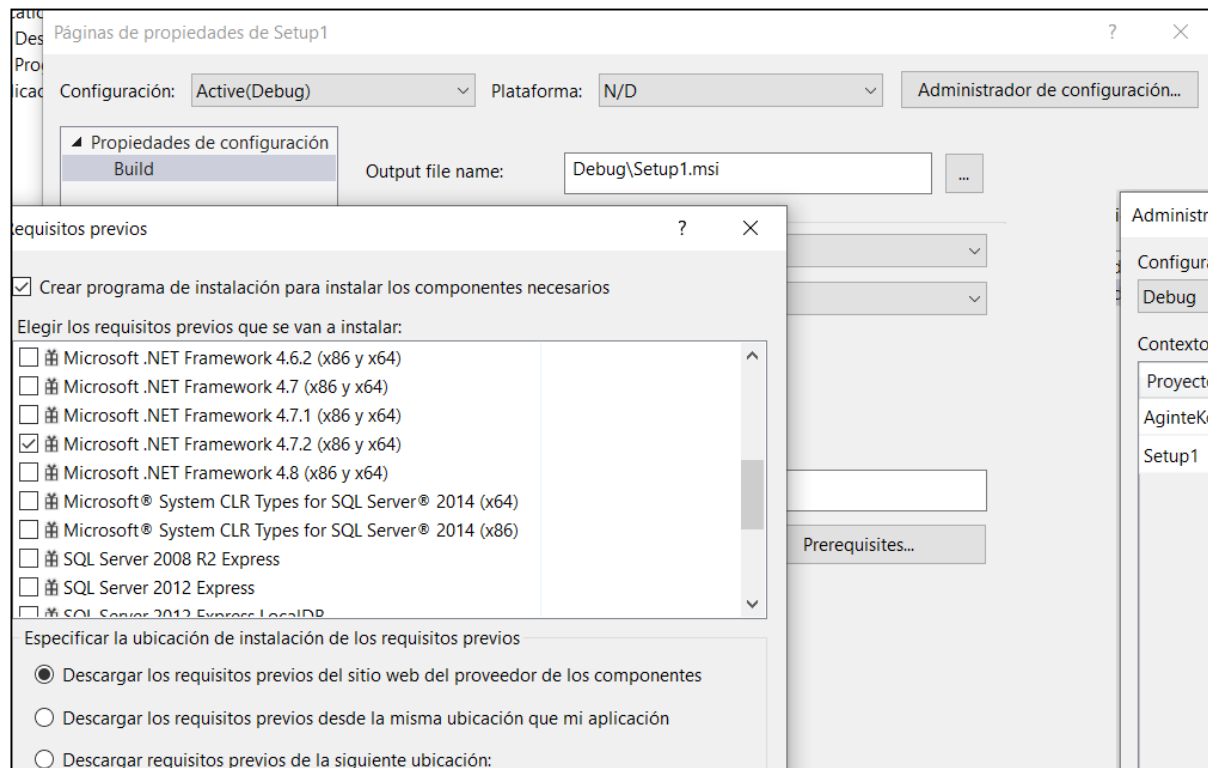
The screenshot shows the 'File System' window with the 'Resultado principal from AginteKoadroa (Active)' folder selected. A context menu is open over the folder, showing options like 'Create Shortcut to Resultado principal from AginteKoadroa (Active)' and 'Dependencias'.

## Erronka 1. Aginte koadroa

- Bigarren sarbide zuzena «User's Programs Menu» barruan sortu dugun «Aplicaciones» karpetara
- Ikono bat gehitzen dugu karpeta nagusira eta gero sarbide zuzen bietako propietateetan gehitzen ditugu
- Setup1 proiektuaren propietateetan
  - «TargetPlatform» aldatu eta x64 jarri.
  - «AddRemoveProgramsIcon» aldatu

# Erronka 1. Aginte koadroa

- «Setup1» proiektuaren gainean eskuineko botoiarekin «Páginas de propiedades de Setup1»
  - «Prerequisitos» konprobatu
  - «Administrador de configuración» prestatu «Release» sortzeko







# Erronka 1. Aginte koadroa

- Rekonpilatu eta instaladorea karpetan egongo da

```
Salida
Mostrar salida de: Compilación
Operación Recompilar iniciada...
1>----- Operación Recompilar todo iniciada: proyecto: AginteKoadroa, configuración: Release Any CPU -----
1> AginteKoadroa -> E:\Aprobak_DAM\AginteKoadroa - setup\AginteKoadroa\bin\Release\AginteKoadroa.exe
----- Starting pre-build validation for project 'Setup1' -----
----- Pre-build validation for project 'Setup1' completed -----
2>----- Operación Recompilar todo iniciada: proyecto: Setup1, configuración: Debug -----
Building file 'E:\Aprobak_DAM\AginteKoadroa - setup\Setup1\Debug\Setup1.msi'...
Packaging file 'cuadro.ico'...
Packaging file 'AginteKoadroa.exe.config'...
Packaging file 'System.Net.Http.dll'...
Packaging file 'AginteKoadroa.exe'...
===== Recompilar todo: 2 correctos, 0 incorrectos, 0 omitidos =====
```

Nuevo vol (E:) > Aprobak\_DAM > AginteKoadroa - setup > Setup1 > Debug

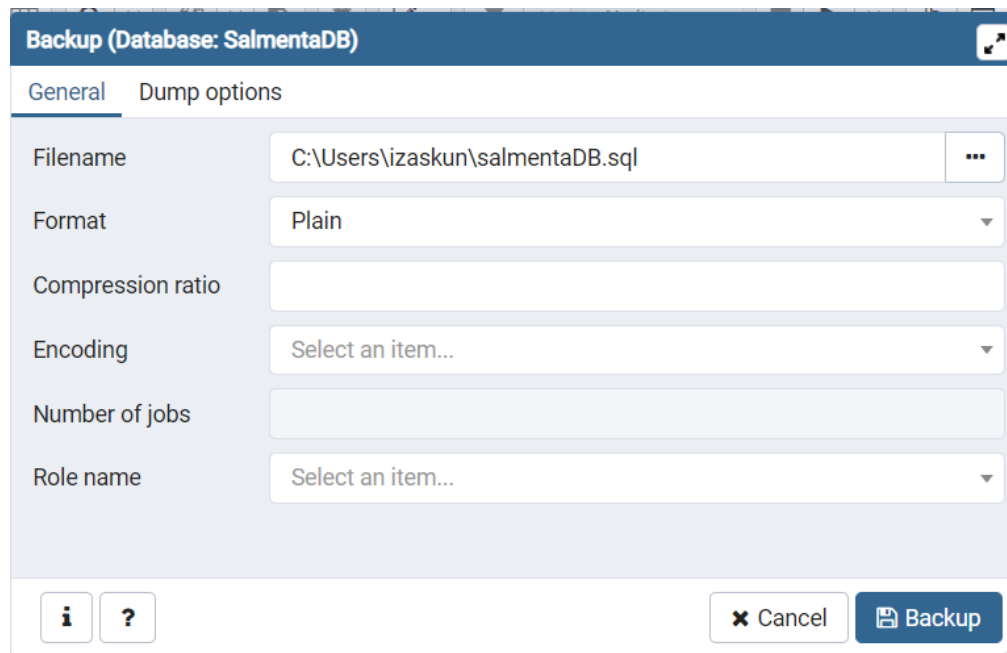
Nombre	Fecha de modificación	Tipo
 setup.exe	11/10/2021 19:02	Aplicación
 Setup1.msi	11/10/2021 19:02	Paquete de Windo...

# Erronka 1. Aginte koadroa

Eskuineko botoia SalmentaDB gainean eta «Backup» aukeratu:

Format: «**Plain**»

Dump options: «**Insert command..**»



Backup (Database: SalmentaDB)

General Dump options

Filename C:\Users\izaskun\salmantaDB.sql

Format Plain

Compression ratio

Encoding Select an item...

Number of jobs

Role name Select an item...

Cancel Backup

# Erronka 1. Aginte koadroa

**Estekak:**

**Linq:**

<https://www.tutorialsteacher.com/linq/linq-standard-query-operators>

<https://www.learnentityframeworkcore.com/dbset/querying-data>

**Laguntza sortzeko informazioa:**

<https://learn.microsoft.com/es-es/dotnet/desktop/winforms/advanced/help-systems-in-windows-forms-applications?view=netframeworkdesktop-4.8>