

ASP .NET

Visual Studio 2022

RETO2 - ASP .NET – SASKIA

WineShop web aplikazioa

Karritoa kodifikatzen:

<https://shakeelosmani.wordpress.com/2015/12/27/asp-net-mvc-step-by-step-complete-e-commerce-tutorial/>

“Session” aldagaia erabiltzen dugu zein erabiltzaile konektatuta dagoen jakiteko eta horrela karrito ezberdinak ondo identifikatuta egongo dira. Hori egiteko program.cs fitxategian lerro batzuk gehitu behar ditugu

```
builder.Services.AddControllersWithViews();
```

```
builder.Services.AddDistributedMemoryCache();
```

```
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromSeconds(10);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});
```

```
//Gure zerbitzuak
```

```
.....
```

```
app.UseRouting();
```

```
app.UseAuthentication();
app.UseAuthorization();
```

```
app.UseSession();
```

```
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
app.MapRazorPages();
```

WineShop web aplikazioa

Erabiltzaile bakoitzak bere karrittoa du, eta zein karritotan gehitu behar diren ardoak jakiteko klase berezi bat egingo dugu: **Saskia.cs (zerbitzuen karpeta jarriko dugu)**

```
public class Saskia
{
    public string SaskiaId { get; set; }
    public const string CartSessionKey = "cartId";

    public static Saskia SaskiaLortu(HttpContext contestua)
    {
        var cart = new Saskia();
        cart.SaskiaId = cart.SaskiaLortuId(contestua);
        return cart;
    }
    public static Saskia SaskiaLortu(Controller controller)
    {
        return SaskiaLortu(controller.HttpContext);
    }
    public string SaskiaLortuId(HttpContext contestua)
    {
        if (contestua.Session.GetString(CartSessionKey) == null)
        {
            if (!string.IsNullOrEmpty(contestua.User.Identity.Name))
            {
                contestua.Session.SetString(CartSessionKey, contestua.User.Identity.Name);
            }
            else
            {
                Guid tempCartId = Guid.NewGuid();
                contestua.Session.SetString(CartSessionKey, tempCartId.ToString());
            }
        }
        return contestua.Session.GetString(CartSessionKey);
    }
}
```

WineShop web aplikazioa

Gero Kontroladorea sortu (zerbitzua erabiliko du: saskiaService)

SaskiaController

```
using Microsoft.AspNetCore.Mvc;
using WineShop.Services;
using WineShop.ViewModels;

namespace WineShop.Controllers
{
    public class SaskiaController : Controller
    {
        private readonly ISaskiaService _saskiaService;
        public SaskiaController(ISaskiaService saskiaService)
        {
            _saskiaService = saskiaService;
        }
        public async Task<IActionResult> Index()
        {
            return View(); // momentuz bista hutsa sortuko dugu (empty txantiloia)
        }
        public async Task<IActionResult> SaskiaGehitu(int id)
        {
            var cart = Saskia.SaskiaLortu(this.HttpContext); //aurretik sortu dugun Saskia klasea erabiliz
            await _saskiaService.SaskiaGehitu(id, cart.SaskiaId); //zerbitzu berrian karritoan gehitzeko
            return RedirectToAction("Index");
        }
    }
}
```

WineShop web aplikazioa

Zerbitzu berria: SaskiaService. Lehenengo Interfacea sortuko dugu

```
public interface ISaskiaService
{
    Task SaskiaGehitu(int ardoald, string saskiald);}
```

WineShop web aplikazioa**Zerbitzu berria: SaskiaService**

```
public class SaskiaService
{
    private Uri rutaTodos = new Uri("https://localhost:44367/api/SaskiaAlea/");

    private readonly IArdoaService _ardoaService;
    public SaskiaService(IArdoaService ardoaService)
    {
        _ardoaService = ardoaService;
    }
    public async Task SaskiaGehitu(int ardoald, string saskiald)
    {
        //Saskia existitzen den begiratu
        Uri rutasaskia = new Uri(rutaTodos, saskiald);
        SaskiaAlea cartitem = new SaskiaAlea();
        using (var httpClient = new HttpClient())
        {
            using (var response = await httpClient.GetAsync(rutasaskia))
            {
                string apiResponse = await response.Content.ReadAsStringAsync();
                cartitem = JsonConvert.DeserializeObject<SaskiaAlea>(apiResponse);
            }
        }
    }
}
```

Program.cs

```
services.AddScoped<ISaskiaService, SaskiaService>();
```

WineShop web aplikazioa**Zerbitzu berria: SaskiaService**

```

if (cartitem == null)
{
    // Karrito berria sortu
    cartitem.Ardoald = ardoald;
    cartitem.Saskiald = saskiald;
    cartitem.Kantitatea = 1;
    cartitem.Data = DateTime.Now;
    //_context.SaskiaAlea.Add(cartitem);
    using (var httpClient = new HttpClient())
    {
        StringContent content = new StringContent(JsonConvert.SerializeObject(cartitem),
Encoding.UTF8, "application/json");
        var response = await httpClient.PostAsync(rutaTodos, content);
        response.EnsureSuccessStatusCode();
    }
}
else
{
    cartitem.Kantitatea++;
    using (var httpClient = new HttpClient())
    {
        StringContent content = new StringContent(JsonConvert.SerializeObject(cartitem),
Encoding.UTF8, "application/json");
        var response = await httpClient.PutAsync(rutasaskia, content);
        response.EnsureSuccessStatusCode();
    }
}
}
}
}

```

WineShop web aplikazioa**Models berria: SaskiaService**

```
public class SaskiaAlea
{
    [Key]
    public int Id { get; set; }
    public string Saskiald { get; set; }
    public int Kantitatea { get; set; }
    public System.DateTime Data { get; set; }
    public int Ardoald { get; set; }
}
```


WineShop web aplikazioa

REST API-an aldaketak batzuk egin behar ditugu saskia irakurtzeko (Saskiald erabili behar da)

ISaskiaAleaService

```
public interface ISaskiaAleaService
{
    ....
    Task<SaskiaAlea> GetSaskiaAlea(string id);
}
```

SaskiaAleaService

```
public async Task<SaskiaAlea> GetSaskiaAlea(string id)
{
    return await _context.SaskiaAlea
        .SingleOrDefaultAsync(a => a.Saskiald == id);
}
```

SaskiaAleaController

```
public async Task<ActionResult<SaskiaAlea>> GetSaskiaAlea(string id)
{
    return await _saskiaAleaService.GetSaskiaAlea(id);
}
```

WineShop web aplikazioa

REST API-tik ardo baten informazioa lortuko dugu. Lehenengo pausua eredu berri bat egitea izango da, ardoaren motaren informazio ere irakurriko dugulako. Ardoa eta Mota arteko erlazioa jarriko dugu.

Mota:

```
public class Mota
{
    [Key]
    public int Id { get; set; }
    public string Izena { get; set; }
    public virtual IList<Ardoa> Ardoak { get; set; }
}
```

Ardoa:

```
Public class Ardoa
{
    .....
    public int Motald { get; set; }
    public virtual Mota Mota { get; set; }
```

WineShop web aplikazioa

Zerbitzua prestatuko dugu informazio hori jasotzeko

IArdoaService:

Task<Ardoa> GetArdoa(int id);

ArdoaService:

```
public async Task<Ardoa> GetArdoa(int id)
{
    Ardoa ardoa = new Ardoa();
    Uri rutaArdoBat = new Uri(rutaTodos, id.ToString());
    using (var httpClient = new HttpClient())
    {
        using (var response = await httpClient.GetAsync(rutaArdoBat))
        {
            string apiResponse = await response.Content.ReadAsStringAsync();
            ardoa = JsonConvert.DeserializeObject<Ardoa>(apiResponse);
        }
    }
    return ardoa;
}
```

WineShop web aplikazioa

Home/Index.cshtml orrian esteka gehituko dugu (karruseleko ardoei)

```
<div class="separator clear-left">
  <p class="btn-details">
    <i class="fa fa-list"></i><a href="" class="d-sm-none d-md-block">Detaile gehiago</a>
  </p>
  <p class="btn-add">
    <i class="fa fa-shopping-cart"></i>
    <a asp-controller="Saskia" asp-action="SaskiaGehitu" asp-route-id="@item.Id">Karritora gehitu</a>
  </p>
</div>
```


WineShop web aplikazioa

****Hau ezabatu, jada badago****

REST API-a aldatuko dugu (Beste soluzio batean daukaguna):

IArdoaService:

```
public interface IArdoaService
{
    Task<Ardoa> GetArdoa(int id);
}
```

ArdoaService:

```
public async Task<Ardoa> GetArdoa(int id)
{
    return await _context.Ardoa
        .Include(a => a.Mota)
        .SingleOrDefaultAsync(a => a.Id == id);
}
```

ArdoaController:

```
// GET: api/Ardoa/5
[HttpGet("{id}")]
public async Task<ActionResult<Ardoa>> GetArdoa(int id)
{
    return await _ardoaService.GetArdoa(id);
}
```

WineShop web aplikazioa

SaskiaViewModel sortuko dugu saskiaren datuak bistaratzeko (ViewModels karpeta berria sortuko dugu)

```
using WineShop.Models;

namespace WineShop.ViewModels
{
    public class SaskiaViewModel
    {
        public IList<SaskiaAlea> SaskiaAleak { get; set; }
        public decimal SaskiaGuztira { get; set; }
    }
}
```

WineShop web aplikazioa

Gero Kontroladorean Index metodoa aldatuko dugu, karritoan dauden erosketak bistaratzeko:
SaskiaController

```
using Microsoft.AspNetCore.Mvc;
using WineShop.Services;
using WineShop.ViewModels;

namespace WineShop.Controllers
{
    public class SaskiaController : Controller
    {
        .....
        public async Task<ActionResult> Index()
        {
            var cart = Saskia.SaskiaLortu(this.HttpContext);
            var saskiaViewModel = new SaskiaViewModel(); //ViewModel bat erabiliko dugu
            saskiaViewModel.SaskiaAleak = await _saskiaService.SaskiaLortuAleak(cart.Saskiald);
            saskiaViewModel.SaskiaGuztira = await _saskiaService.Guztira(cart.Saskiald);

            return View(saskiaViewModel);
        }
    }
}
```


WineShop web aplikazioa

SaskiaService, metodo berri bi. Lehenengo Interfacea aldatuko dugu

```
public interface ISaskiaService
{
    .....
    Task<IList<SaskiaAlea>> SaskiaLortuAleak(string saskiald);
    Task<decimal> Guztira(string saskiald);
}
```

WineShop web aplikazioa

Zerbitzua aldatu: SaskiaService

```
public class SaskiaService : ISaskiaService
```

```
{  
    .....  
  
    public async Task<List<SaskiaAlea>> SaskiaLortuAleak(string saskiald)  
    {  
        return await _context.SaskiaAlea  
            .Where(s => s.Saskiald == saskiald)  
            .ToListAsync();  
    }  
    public async Task<decimal> Guztira(string saskiald)  
    {  
        decimal? total = _context.SaskiaAlea  
            .Where(s => s.Saskiald == saskiald)  
            .Select(s => s.Kantitatea * s.Salneurria).Sum();  
        return total ?? 0;  
    }  
}
```

WineShop web aplikazioa

Views karpeta barruan, “Saskia” karpetan sortu genuen Index.cshtml aldatuko dugu :

```
@using System.Globalization;
@model WineShop.ViewModels.SaskiaViewModel
```

```
@{
    ViewData["Title"] = "Index";
}
<div class="container">
    <div id="update-message"></div>
    <table class="table table-striped">
        <thead>
            <tr class="borgona">
                <th>Izena</th>
                <th>Irudia</th>
                <th>Kantitatea</th>
                <th>Salneurria</th>
                <th>Guztira</th>
            </tr>
        </thead>
```

***hurrengo diapositiban jarraitzen du

Site.css

```
.altuera {
    max-height: 100px;
}
.btn-borgona {
    background-color: #981212;
}
.btn-borgona:hover {
    background-color: #f2c5c5;
    color: #FFF;
}
```

WineShop web aplikazioa

```
@foreach (var item in Model.SaskiaAleak)
{
    <tr id="row-@item.Ardoald">
        <td>
            @Html.DisplayFor(modelItem => item.Izena)
        </td>
        <td>
            
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Kantitatea)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Salneurria)
        </td>
        <td>
            @{ decimal guztira = item.Salneurria * item.Kantitatea;}
            @guztira
        </td>
    </tr>
}
<tr>
    <td><a class="btn btn-borgona text-light" asp-controller="Home" asp-action="Index">Bueltatu</a>
    <a class="btn btn-borgona text-light" asp-controller="Ordaindu" asp-action="Index">Ordaindu</a></td>
    <td></td>
    <td></td>
    <td class="float-right"><p>Guztira: <span id="cart-total">@Model.SaskiaGuztira.ToString("C2", CultureInfo.CurrentCulture)</span></p></td>
</tr>
</table>
</div>
```

Izena	Irudia	Kantitatea	Salneurria	Guztira
Remelluri zuria		2	45,00	90,00
Lindes de Remelluri		1	11,40	11,40
Veuve Clicquot Brut		1	32,55	32,55

Bueltatu

Ordaindu

Guztira: 133,95 €

WineShop web aplikazioa

Ordaintzeko prozesua egiteko, lehenengo ereduak sortuko ditugu. Datubasea eguneratu.

BezeroaEskaera.cs

```
namespace WineShop.Models
{
    public class BezeroaEskaera
    {
        [Key]
        [ScaffoldColumn(false)]
        public int Id { get; set; }
        [Required(ErrorMessage = "Izena derrigorrezkoa da")]
        public string Izena { get; set; }
        [Required(ErrorMessage = "Abizena derrigorrezkoa da")]
        public string Abizena { get; set; }
        [Required(ErrorMessage = "Helbidea derrigorrezkoa da")]
        public string Helbidea { get; set; }
        [Required(ErrorMessage = "Hiria derrigorrezkoa da")]
        public string Hiria { get; set; }
        [Required(ErrorMessage = "Herrialdea derrigorrezkoa da")]
        public string Herrialdea { get; set; }
        [Required(ErrorMessage = "Postakodea derrigorrezkoa da")]
        public string Postakodea { get; set; }
    }
}
```

```
[Required(ErrorMessage = "Telefoniaa derrigorrezkoa da")]
public string Telefonia { get; set; }
[ScaffoldColumn(false)]
public System.DateTime Data { get; set; }
[ScaffoldColumn(false)]
public string Erabiltzailea { get; set; }
public IList<Erosketa> Erosketak { get; set; }
}
}
```

Erosketa.cs

```
namespace WineShop.Models
{
    public class Erosketa
    {
        [Key]
        public int Id { get; set; }
        public int Kantitatea { get; set; }
        public int Ardoald { get; set; }
        public int BezeroaEskaeraId { get; set; }
        public virtual BezeroaEskaera BezeroaEskaera
        {get; set;}
    }
}
```

WineShop web aplikazioa

Kontroladore berria sortzen dugu. Authorize jartzen dugu kontroladorean erabiltzaile erregistratuak bakarrik sartzeko.

OrdainduController.cs

```
using WineShop.Data;
using WineShop.Models;
using Microsoft.AspNetCore.Authorization;
using WineShop.Services;

namespace WineShop.Controllers
{
    [Authorize]
    public class OrdainduController : Controller
    {
        private readonly WineShopDbContext _context;
        private readonly ISaskiaService _saskiaService;

        public OrdainduController(WineShopDbContext context, ISaskiaService saskiaService)
        {
            _context = context;
            _saskiaService = saskiaService;
        }
        public IActionResult Index()
        {
            return View();
        }
    }
}
```

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult>
Index([Bind("Abizena,Helbidea,Herrialdea,Hiria,Izena,Postakod
ea,Telefonoa")] BezeroaEskaera bezeroaEskaera)
{
    if (ModelState.IsValid)
    {
        /*Bezeroen datuak gorde*/
        bezeroaEskaera.Erabiltzailea =
HttpContext.User.Identity.Name;
        bezeroaEskaera.Data = DateTime.Now;
        await
_saskiaService.EskaeraBezeroaGehitu(bezeroaEskaera);

        /*Eskaera gorde*/
        var cart = Saskia.SaskiaLortu(this.HttpContext);
        await _saskiaService.EskaeraSortu(bezeroaEskaera,
cart.Saskiald);
        return RedirectToAction("Osatu", bezeroaEskaera);
    }
    return View(bezeroaEskaera);
}
```


WineShop web aplikazioa

Zerbitzuan gehitzen ditugun metodoak

ISaskiaService.cs

```
Task EskaeraBezeraoGehitu(BezeroaEskaera bezeroaEskaera);
Task EskaeraSortu(BezeroaEskaera bezeroaEskaera, string saskiald);
Task SaskiaEzabatu(string saskiald);
```

SaskiaService.cs

```
public async Task EskaeraBezeraoGehitu(BezeroaEskaera
bezeroaEskaera)
{
    _context.BezeroaEskaera.Add(bezeroaEskaera);
}
public async Task EskaeraSortu(BezeroaEskaera
bezeroaEskaera, string saskiald)
{
    var cartItems = await SaskiaLortuAleak(saskiald);

    foreach (var item in cartItems)
    {
        var erosketa = new Erosketa
        {
            Ardoald = item.Ardoald,
            BezeroaEskaerald = bezeroaEskaera.Id,
            Kantitatea = item.Kantitatea
        };
        _context.Erosketa.Add(erosketa);
    }
    _context.SaveChanges();
    await SaskiaEzabatu(saskiald);
}
```

```
public async Task SaskiaEzabatu(string
saskiald)
{
    var cartItems = await
_context.SaskiaAlea
.Where(c => c.Saskiald ==
saskiald)
.ToListAsync();

    foreach (var cartItem in cartItems)
    {
        _context.SaskiaAlea.Remove(cartItem);
    }
    _context.SaveChanges();
}
```


WineShop web aplikazioa

Create txantiloia aukeratu Index bista sortzeko (eredua BezeroaEskaera aukeratu)

```
@model WineShop.Models.BezeroaEskaera
```

```
@{
    ViewData["Title"] = "Index";
}
```

```
<h4>Ordaintzeko formularioa</h4>
```

```
<hr />
```

```
<div class="row">
```

```
    <div class="col-md-4">
```

```
        <form asp-action="Index">
```

```
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Izena" class="control-label"></label>
```

```
                <input asp-for="Izena" class="form-control" />
```

```
                <span asp-validation-for="Izena" class="text-danger"></span>
```

```
            </div>
```

```
.....
```

WineShop web aplikazioa

Bezeroaren datuak eskatzerakoan, aurretik erregistratu den ala ez konprobatuko dugu. Horrela bada, aurretik gorde ditugun datuak erakutsiko diogu eta aldatzeko aukera emango diogu
OrdainduController

```
namespace WineShop.Controllers
{
    [Authorize]
    public class OrdainduController : Controller
    {
        private readonly ISaskiaService _saskiaService;

        public OrdainduController(ISaskiaService saskiaService)
        {
            _saskiaService = saskiaService;
        }
        public async Task<ActionResult> Index()
        {
            var bezeroaEskaera = await _saskiaService.BezeroaDatuak(HttpContext.User.Identity.Name);
            return View(bezeroaEskaera);
        }
    }
}
```

WineShop web aplikazioa

ISaskiaService

Task<BezeroaEskaera> BezeroaDatuak(string erabiltzailea);

SaskiaService

```
public async Task<BezeroaEskaera> BezeroaDatuak(string erabiltzailea)
{
    var bezeroaEskaera = await _context.BezeroaEskaera
        .SingleOrDefaultAsync(b => b.Erabiltzailea == erabiltzailea);
    return bezeroaEskaera;
}
```

WineShop web aplikazioa

Bezeroaren datuak berriak edo aldatzekoak diren kontroladorean bereiztuko ditugu

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Index([Bind("Abizena,Helbidea,Herrialdea,Hiria,Izena,Postakodea,Telefonoa")]
BezeroaEskaera bezeroaEskaera)
{
    if (ModelState.IsValid)
    {
        bezeroaEskaera.Erabiltailea = HttpContext.User.Identity.Name;
        bezeroaEskaera.Data = DateTime.Now;
        /*Bezeroen datuak gorde edo aldatu egin behar den begiratu*/
        if ((await _saskiaService.BezeroaDatuak(bezeroaEskaera.Erabiltailea))
            != null)
        {
            await _saskiaService.EskaeraBezeroaAldatu(bezeroaEskaera);
        }
        else
        {
            await _saskiaService.EskaeraBezeroaGehitu(bezeroaEskaera);
        }
        /*Eskaera gorde*/
        var cart = Saskia.SaskiaLortu(this.HttpContext);
        await _saskiaService.EskaeraSortu(bezeroaEskaera, cart.Saskiald);

        return RedirectToAction("Osatu", new { bezeroa =
            bezeroaEskaera.Izena + " " + bezeroaEskaera.Abizena, saskiald = cart.Saskiald
        });
    }
    return View(bezeroaEskaera);
}
```

```
public async Task<ActionResult> Osatu(string
bezeroa, string saskiald)
{
    var osatuViewModel = new
OsatuViewModel(); //ViewModel bat erabiliko dugu
    osatuViewModel.SaskiaAleak = await
_saskiaService.SaskiaLortuAleak(saskiald);
    osatuViewModel.bezeroa = bezeroa;
    return View(osatuViewModel);
}
```

Momentuz ez dugu ezabatuko. Beraz kontroladore honetatik linea hau kendu edo komentatuta utzi

```
await _saskiaService.SaskiaEzabatu(saskiald);
```

WineShop web aplikazioa

ISaskiaService

Task EskaeraBezeroaAldatu(BezeroaEskaera bezeroaEskaera);

SaskiaService

```
public async Task EskaeraBezeroaAldatu(BezeroaEskaera bezeroaEskaera)
{
    var zaharra = await _context.BezeroaEskaera.SingleOrDefaultAsync(b => b.Erabiltzailea == bezeroaEskaera.Erabiltzailea);

    zaharra.Izena = bezeroaEskaera.Izena;
    zaharra.Abizena = bezeroaEskaera.Abizena;
    zaharra.Data = bezeroaEskaera.Data;
    zaharra.Helbidea = bezeroaEskaera.Helbidea;
    zaharra.Hiria = bezeroaEskaera.Hiria;
    zaharra.Herrialdea = bezeroaEskaera.Herrialdea;
    zaharra.Postakodea = bezeroaEskaera.Postakodea;
    zaharra.Telefonia = bezeroaEskaera.Telefonia;
    _context.SaveChanges();
}
```

WineShop web aplikazioa

OsatuViewModel

```
using WineShop.Models;

namespace WineShop.ViewModels
{
    public class OsatuViewModel
    {
        public string Bezeroa { get; set; }
        public string Saskiald { get; set; }
        public IList<SaskiaAlea> SaskiaAleak { get; set; }
    }
}
```

WineShop web aplikazioaOsatu metodoaren bista sortuko dugu

@model WineShop.ViewModels.OsatuViewModel

```

@{
    ViewData["Title"] = "Osatu";
}

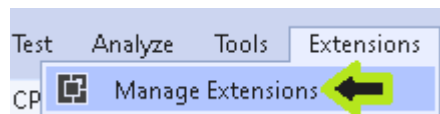
<div class="container">
    <p>@Html.DisplayFor(modelItem => Model.Bezeroa), zure erosketa ondo osatu da</p>
    <table class="table">
        <thead>
            <tr>
                <th>
                    Ardoa
                </th>
                <th>
                    Kantitatea
                </th>
            </tr>
        </thead>
        <tbody>
            @foreach (var item in Model.SaskiaAleak)
            {
                <tr>
                    <td>
                        @Html.DisplayFor(modelItem => item.Izena)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.Kantitatea)
                    </td>
                </tr>
            }
            <tr>
                <td>
                    <a class="btn btn-borgona text-light" asp-controller="Home" asp-action="Index">Bueltatu</a>
                    <a class="btn btn-borgona text-light" asp-controller="Report" asp-action="Inprimatu">Tiketa</a>
                </td>
                <td></td>
            </tr>
        </tbody>
    </table>
</div>

```

WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

Luzapen bi instalatu:



Manage Extensions

▸ Installed

Sort by: Relevance

▾ Online

▾ Visual Studio Marketplace

Search Results

▸ Controls

▸ Templates

▸ Tools

▸ Updates



Microsoft Reporting Services Projects

The Microsoft RDL report designer, projects and wizards for creating professional reports. This package provides support for...

Microsoft



Microsoft RDLC Report Designer

RDLC Designer and projects for ASP.Net and Windows Forms for Reporting Services Report Viewer Control

Microsoft

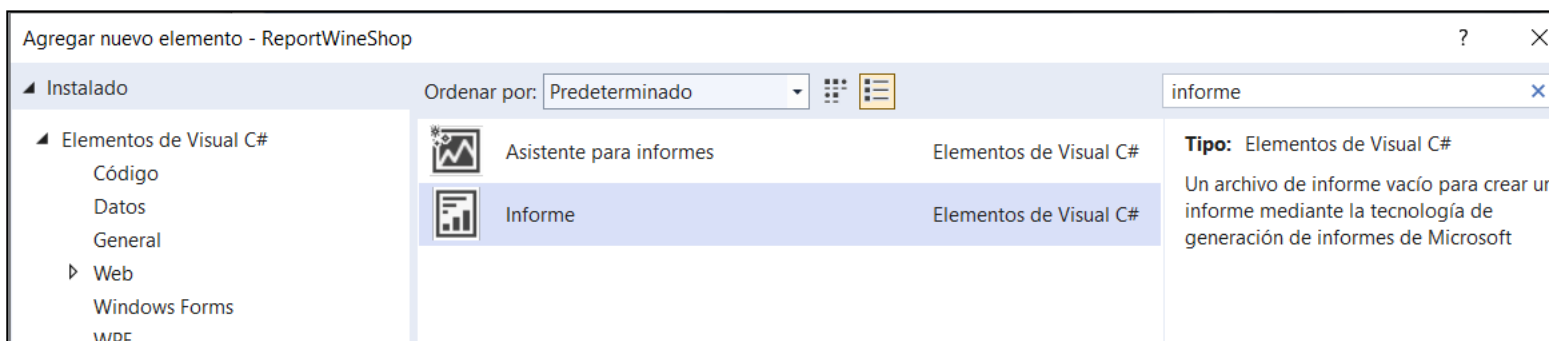


WineShop web aplikazioa

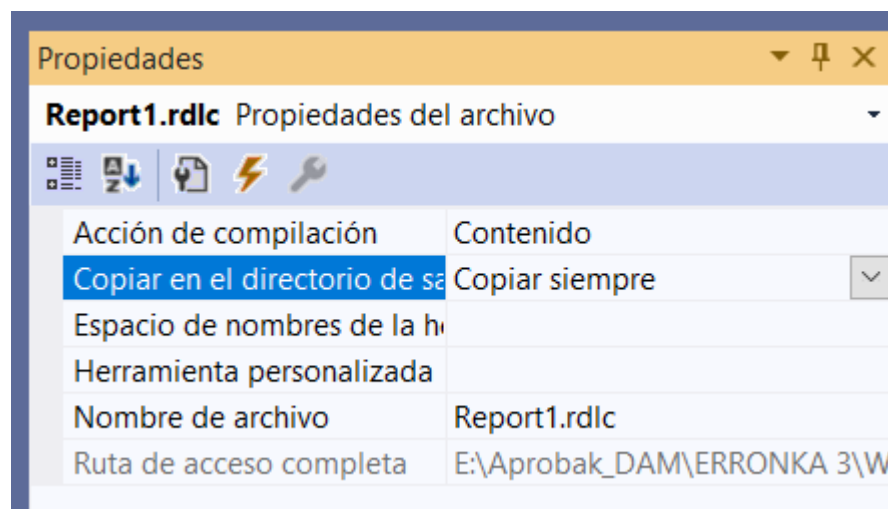
Tiketak sortzeko prestatuko dugu:

Microsoft RDLC – Report Viewer erabiliko dugu eta horretarako proiektu berri bat sortuko dugu gure soluzio barruan: ReportWineShop (Windows forms proiektua)

Osagai berria gehitu: «informe» motakoa



Propiedades zabaldu eta aukera hauek jarri:

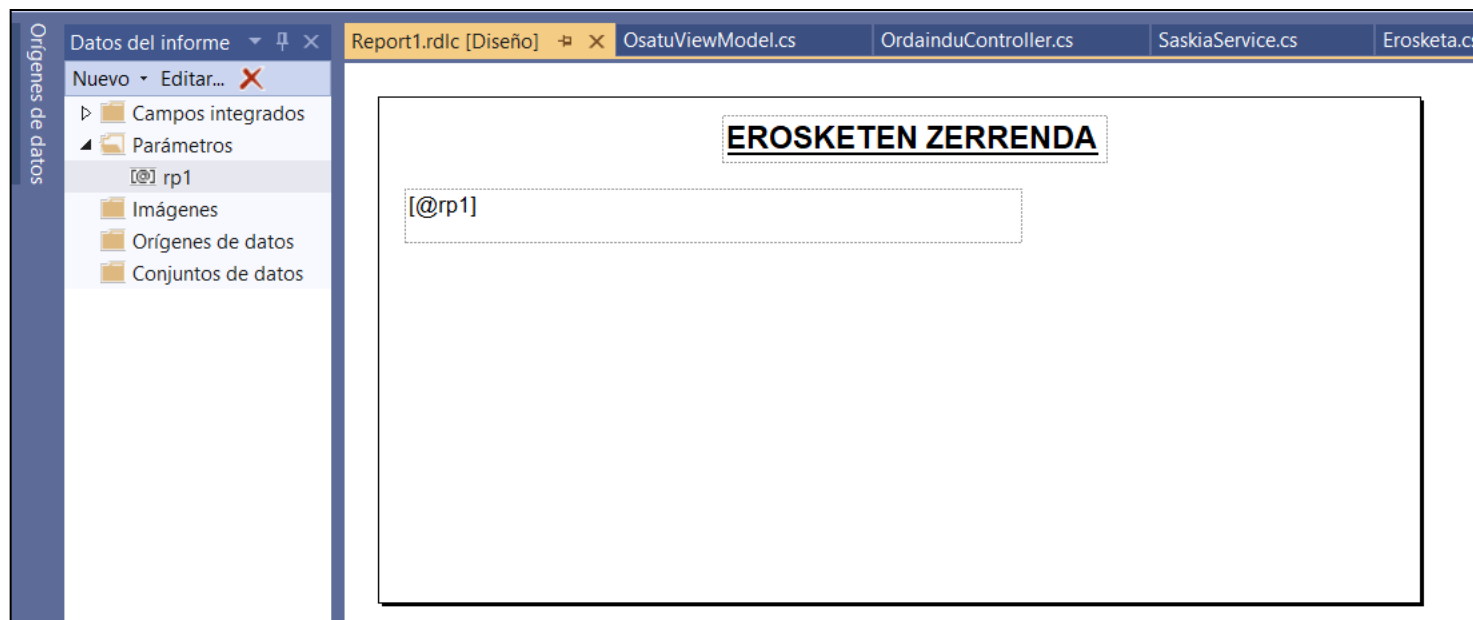


WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

Diseinatu nahi den bezala (titulua, letra tamaina, negrita.....)

Parametro bat sortu eta textbox baten barruan jarri



WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

WineShop proiektuan, «Administrador de paquetes NuGet» - **AspNetCore.Reporting**



Karpeta berria sortu **wwwroot** barruan: **Reports** eta barruan **Report1.rdlc** kopiatu

WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

Kontroladore berria sortuko dugu(Hutsik dagoena): **ReportController**

```
public class ReportController : Controller
{
    private readonly IWebHostEnvironment _webHostEnvironment;
    public ReportController(IWebHostEnvironment webHostEnvironment)
    {
        _webHostEnvironment = webHostEnvironment;
    }
    public IActionResult Inprimatu()
    {
        string mimetype = "";
        int extension = 1;
        var path = $"{_webHostEnvironment.WebRootPath}\\Reports\\Report1.rdlc";
        Dictionary<string, string> parameters = new Dictionary<string, string>();
        parameters.Add("rp1", "WineShop Denda online");
        LocalReport localReport = new LocalReport(path);
        var result = localReport.Execute(RenderType.Pdf, extension, parameters, mimetype);
        return File(result.MainStream, "application/pdf");
    }
}
```

Program.cs

```
public static IHostBuilder CreateHostBuilder(string[] args) =>
    Host.CreateDefaultBuilder(args)
        .ConfigureWebHostDefaults(webBuilder =>
        {
            webBuilder.UseContentRoot(Directory.GetCurrentDirectory());
            webBuilder.UseWebRoot("wwwroot");
            webBuilder.UseStartup<Startup>();
        });
```

WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

Exekutatu eta txostena bistaratuko du:

Batzuetan errore bat irtetzen du:

An unhandled exception occurred while processing the request.

NotSupportedException: No data is available for encoding 1252 For information on defining a custom encoding, see
System.Text.Encoding.GetEncoding(int codepage)

LocalProcessingException: An error occurred during local report processing.;No data is available for encoding 1252.
AspNetCore.Reporting.InternalLocalReport.InternalRender(string format, bool allowInternalRenderers, string deviceInfo, PageCountMode pageCountMode, C

Hau konpontzeko:

Administrazio paquetes NuGet eta instalatu: **System.Text.Encoding.CodePages**

Eta metodoan:

using System.Text;

```
public IActionResult Inprimatu()
{
    Encoding.RegisterProvider(CodePagesEncodingProvider.Instance);

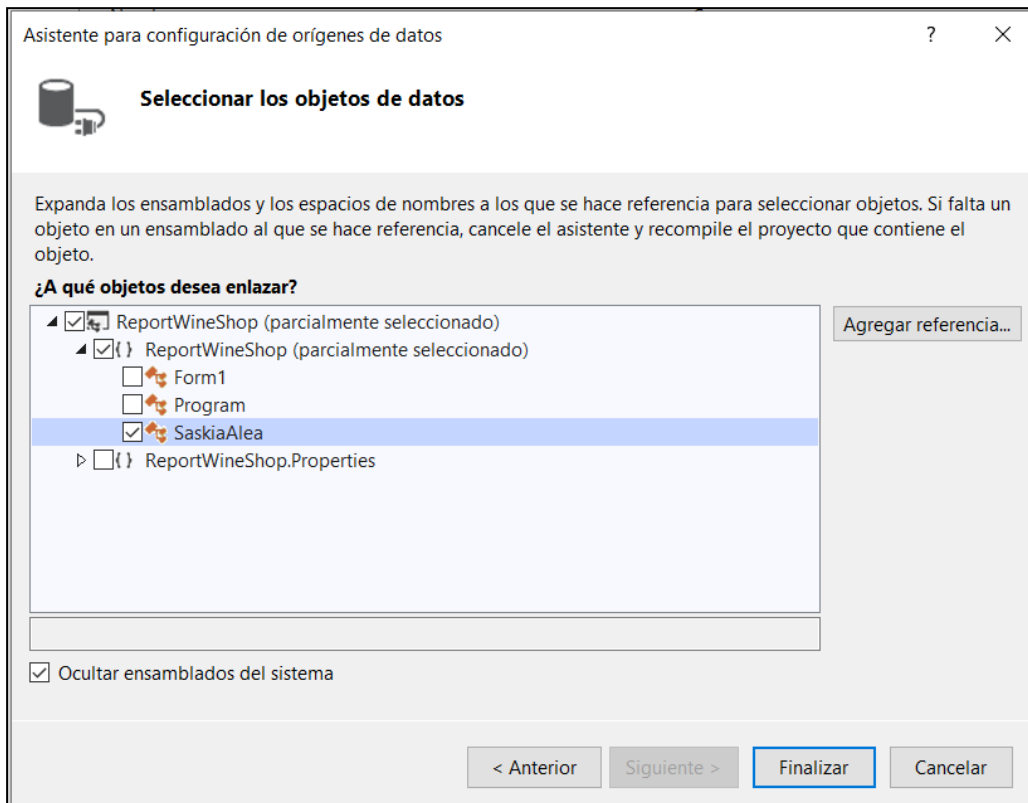
    string mimetype = "";
    ....
    ....
}
```

WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

Orain diseinua aldatuko dugu erosketen informazioa bistaratzeko (**ReportWineShop proiektua**):

- Beste parametro bi jarri, bezeroaren izena eta guztira bistaratzeko.
- Dataset bat sortuko dugu karrtoa bistaratzeko:
 - SaskiaAlea klasea kopia WineShop proiektutik eta namespace aldatu (Konpilatu)
 - «Conjunto de datos – Agregar conjunto de datos – Objeto – Saskia Alea aukeratu

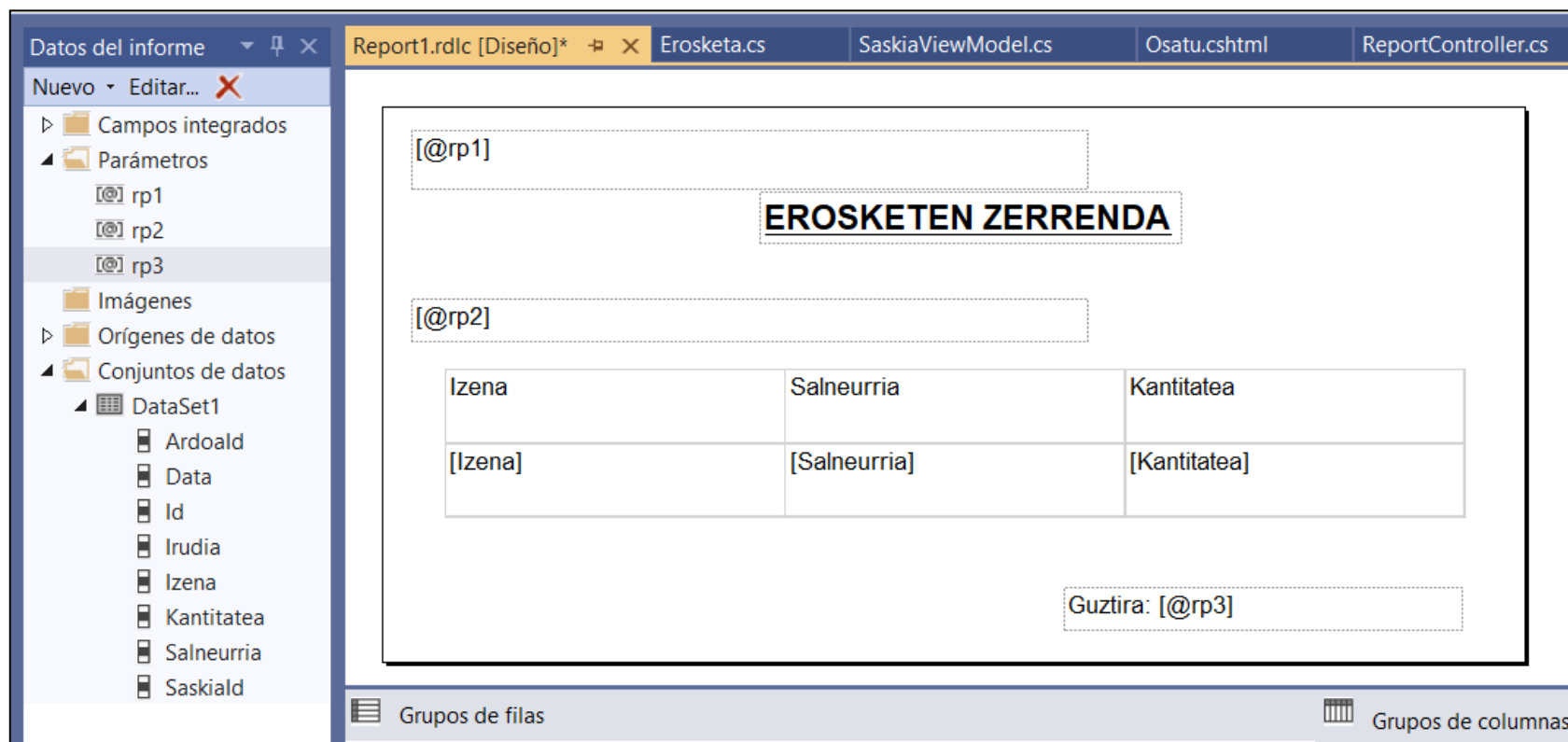


WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

Exekutatu eta txostena bistaratuko du:

«Cuadro de Herramientas – Table » aukeratu eta arrastatu diseinura. Dataset barruan bistaratu nahi ditugun osagaiak arrastatu ere taula barrura.



Amaitutakoan, beste proiektuan kopiatu: Reports karpetan

WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

Orain datuak hartu eta txostenera pasatuko ditugu (**WineShop proiektua**).

Osatu.cshtml:

```
<td>
  <a class="btn btn-borgona text-light" asp-controller="Home" asp-action="Index">Bueltatu</a>
  <a class="btn btn-borgona text-light" asp-controller="Report" asp-action="Inprimatu" asp-route-bezeroa="@Model.Bezeroa" asp-
route-saskiald="@Model.Saskiald">Tiketa</a>
</td>
```

ReportController:

```
public class ReportController : Controller
{
    private readonly IWebHostEnvironment _webHostEnvironment;
    private readonly ISaskiaService _saskiaService;
    public ReportController(IWebHostEnvironment webHostEnvironment, ISaskiaService saskiaService)
    {
        _webHostEnvironment = webHostEnvironment;
        _saskiaService = saskiaService;
    }
}
```


WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

ReportController:

```
public async Task<IActionResult> Inprimatu(string bezeroa, string saskiald)
{
    string mimetype = "";
    int extension = 1;
    var path = $"({_webHostEnvironment.WebRootPath}\\Reports\\Report1.rdlc";
    Dictionary<string, string> parameters = new Dictionary<string, string>();
    parameters.Add("rp1", "WineShop Denda online");
    parameters.Add("rp2", bezeroa);
    decimal guztira = await _saskiaService.Guztira(saskiald);
    parameters.Add("rp3", guztira.ToString());
    var saskiaAleak = await _saskiaService.SaskiaLortuAleak(saskiald);

    LocalReport localReport = new LocalReport(path);
    localReport.AddDataSource("DataSet1", saskiaAleak);
    var result = localReport.Execute(RenderType.Pdf, extension, parameters, mimetype);
    return File(result.MainStream, "application/pdf");
}
```

WineShop web aplikazioa

Tiketak sortzeko prestatuko dugu:

WineShop Denda online

EROSKETEN
ZERRENDA

izaskun gg

Izena	Salneurria	Kantitatea
Remelluri zuria	45,00	1
Veuve Clicquot Brut	32,55	1
Meissner	23,80	1

Guztira: 101,3500

WineShop web aplikazioa

Jarraipena aukera edukitzeko Erosketa ereduan, bi zutabe gehituko ditugu: data eta jarraipena (taulan aldatu ere)

```
public DateTime Data { get; set; }
public int Jarraipena { get; set; }
```

ISaskiaService

int ErosketaGehitu(BezeroaEskaera bezeroaEskaera, string saskiald);

SaskiaService

```
public int ErosketaGehitu(BezeroaEskaera bezeroaEskaera, string saskiald)
{
    var jarraipena = JarraipenaLortu();
    jarraipena++;

    var cartItems = SaskiaLortuAleak(saskiald);

    foreach (var item in cartItems)
    {
        var erosketa = new Erosketa
        {
            BezeroaEskaeraId = bezeroaEskaera.Id,
            ArdoId = item.ArdoId,
            Kantitatea = item.Kantitatea,
            Data = DateTime.Now,
            Jarraipena = jarraipena
        };
        _context.Add(erosketa);
    }
    _context.SaveChanges();
    return jarraipena;
}
```

```
public int JarraipenaLortu()
{
    return _context.Erosketa
        .Max(e => e.Jarraipena);
}
```

WineShop web aplikazioa

OrdainduController

```
var jarraipena=_saskiaService.ErosketaGehitu(bezeroaEskaera, cart.Saskiald);
```

```
return RedirectToAction("Osatu", new { id = jarraipena });}
```

```
public IActionResult Osatu(int id)  
{  
    ViewData["jarraipena"] = id;  
    return View();  
}
```

Osatu.cshtml

```
<div class="container">  
    <p>Zure erosketa ondo osatu da</p>  
    <p>Erosketaren identifikatzailea: @ViewData["jarraipena"]</p>  
</div>
```

WineShop web aplikazioa

Saskitik kentzeko kodigoa egingo dugu.

Index.html

```
<td class="padina">
    <a href="" class="RemoveLink" data-id="@item.Ardoald">
        Saskitik kendu
    </a>
</td>
```

site.js

```
/*Saskitik ezabatzeko*/
$(document).ready(function () {
    $(".RemoveLink").click(function (e) {
        e.preventDefault();
        // Get the id from the link
        var recordToDelete = $(this).attr("data-id");
        if (recordToDelete != "") {
            // Perform the ajax post
            $.post("/Saskia/SaskiaKendu", { "id": recordToDelete },
                function (data) {
                    // Successful requests get here
                    // Update the page elements
                    if (data.itemKopurua === 0) {
                        $('#row-' + data.ezabatutakold).fadeOut('slow');
                    } else {
                        $('#item-count-' + data.ezabatutakold).text(data.itemKopurua);
                    }
                    $('#cart-total').text(data.saskiaGuztira);
                    $('#update-message').text(data.mezua);

                    //Aurrerago gehituko dugun lerroa
                    //$('#cart-status').text(data.itemGuztiak);
                });
        }
    });
});
```

WineShop web aplikazioa

Saskitik kentzeko metodoa egingo dugu.

SaskiaController

```
[HttpPost]
public ActionResult SaskiaKendu(int id)
{
    var cart = Saskia.SaskiaLortu(this.HttpContext);

    var saskiaItem = _saskiaService.SaskiaAleaDatuak(id, cart.SaskiaId);

    int itemCount = _saskiaService.SaskiaKendu(id, cart.SaskiaId);

    var results = new SaskiaKenduViewModel
    {
        Mezua= HtmlEncoder.Default.Encode(saskiaItem.Ardoa.Izena) + " zure saskitik kendu da",
        SaskiaGuztira = _saskiaService.Guztira(cart.SaskiaId),
        //ItemGuztiak = _saskiaService.ItemGuztiak(cart.SaskiaId),
        ItemKopurua = itemCount,
        Ezabatutakold = id
    };

    return Json(results);    }
```

WineShop web aplikazioa

Saskia zerbitzuak eguneratuko ditugu:

ISaskiaService

SaskiaAlea SaskiaAleaDatuak(int ardoald, string saskiald);

int SaskiaKendu(int ardoald, string saskiald);

int ItemGuztiak(string saskiald);

SaskiaService

```
public SaskiaAlea SaskiaAleaDatuak(int ardoald, string
saskiald)
{
    var cartItem = _context.SaskiaAlea
        .Include(s => s.Ardoa)
        .SingleOrDefault(
            c => c.Saskiald == saskiald
            && c.Ardoald == ardoald);

    return cartItem;
}
```

```
public int SaskiaKendu(int ardoald, string saskiald)
{
    var cartItem = _context.SaskiaAlea.SingleOrDefault(
        c => c.Saskiald == saskiald
        && c.Ardoald == ardoald);

    int itemCount = 0;

    if (cartItem != null)
    {
        if (cartItem.Kantitatea > 1)
        {
            cartItem.Kantitatea--;
            itemCount = cartItem.Kantitatea;
        }
        else
        {
            _context.SaskiaAlea.Remove(cartItem);
        }

        _context.SaveChanges();
    }
    return itemCount;
}
```

WineShop web aplikazioa**SaskiaService**

```
public int ItemGuztiak(string saskiald)
{
    int? count = _context.SaskiaAlea
        .Where(s => s.Saskiald == saskiald)
        .Select(s => s.Kantitatea).Sum();

    return count ?? 0;
}
```

SaskiaKenduViewModel

```
namespace WineShop.ViewModels
{
    public class SaskiaKenduViewModel
    {
        public string Mezua { get; set; }
        public decimal SaskiaGuztira { get; set; }
        //public int ItemGuztiak { get; set; }
        public int ItemKopurua { get; set; }
        public int Ezabatutakold { get; set; }
    }
}
```


WineShop web aplikazioa

Saskia goiko menuan ikusteko:

ViewComponents karpeta barruan:

SaskiaViewComponent

namespace WineShop.ViewComponents

```
{  
    public class SaskiaViewComponent : ViewComponent  
    {  
        private readonly ISaskiaService _saskiaService;  
  
        public SaskiaViewComponent(ISaskiaService saskiaService)  
        {  
            _saskiaService = saskiaService;  
        }  
        public IViewComponentResult Invoke()  
        {  
            var cart = Saskia.SaskiaLortu(this.HttpContext);  
  
            ViewData["ItemGuztiak"] = _saskiaService.ItemGuztiak(cart.Saskiald);  
  
            return View();  
        }  
    }  
}
```

WineShop web aplikazioa

Shared/Components/Saskia:

Default.cshtml

```
<ul class="nav navbar-nav navbar-right">
  <li><a asp-area="" asp-controller="Saskia" asp-action="Index">
    <span class="fas fa-shopping-cart"></span>
    (<span id="cart-status">@ ViewData["ItemGuztiak"] </span>)
  </a>
</li>
</ul>
```

Site.js, SaskiaController eta SaskiaKenduViewModel fitxategietan komentatuta utzi genituen erroak prestatu

Eta _Layout.cshtml barruan:

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li><a asp-area="" asp-controller="Home" asp-action="Index">Home</a></li>
    <li><a asp-area="" asp-controller="Home" asp-action="About">About</a></li>
    <li><a asp-area="" asp-controller="Home" asp-action="Contact">Contact</a></li>
    <li><a asp-area="" asp-controller="Ardoa" asp-action="Index">Gure ardoak</a></li>
  </ul>
  @await Component.InvokeAsync("Saskia")
  @await Html.PartialAsync("_LoginPartial")
</div>
```

WineShop web aplikazioa

Saskia/Index.cshtml:

```
<div class="col-md-2 padina col-height col-middle">
  <p class="row">
    <a href="" class="RemoveLink" data-id="@item.Ardoald">
      <span class="glyphicon glyphicon-minus"></span>
    </a>
    span id="item-count-@item.Ardoald">@item.Kantitatea</span>
    <a href="" class="AddLink" data-id="@item.Ardoald">
      <span class="glyphicon glyphicon-plus"></span>
    </a>
  </p>
</div>
```

Site.js:

```
/*Saskira gehitzeko*/
$(document).ready(function () {
  $(".AddLink").click(function (e) {
    e.preventDefault();
    // Get the id from the link
    var recordToAdd = $(this).attr("data-id");
    if (recordToAdd !== "") {
      // Perform the ajax post
      $.post("/Saskia/SaskiaGehituJson", { "id": recordToAdd },
        function (data) {
          // Successful requests get here
          // Update the page elements
          $('#item-count-' + data.ezabatutakold).text(data.itemKopurua);
          $('#cart-total').text(data.saskiaGuztira);
          $('#update-message').text(data.mezua);
          $('#cart-status').text('Cart (' + data.itemGuztiak + ')');
        });
    }
  });
});
```

WineShop web aplikazioa

Saskian beste aldaketa batzuk egingo ditugu

Zure saskia:

Chateau La Gordonne zure saskitik kendu da



Chateau La
Gordonne

14,00

- 2 +

14,00€



White Label

22,00

- 1 +

22,00€

Guztira

50

Jarraitu erosten>>

Ordaindu>>

WineShop web aplikazioa**SaskiaController:**

```
public IActionResult SaskiaGehituJson(int id)
{
    var cart = Saskia.SaskiaLortu(this.HttpContext);

    var saskialtem = _saskiaService.SaskiaAleaDatuak(id, cart.Saskiald);

    _saskiaService.SaskiaGehitu(id, cart.Saskiald);

    var results = new SaskiaKenduViewModel
    {
        Mezua = HtmlEncoder.Default.Encode(saskialtem.Ardoa.Izena) + " zure saskira gehitu da",
        SaskiaGuztira = _saskiaService.Guztira(cart.Saskiald),
        ItemGuztiak = _saskiaService.ItemGuztiak(cart.Saskiald),
        ItemKopurua = saskialtem.Kantitatea,
        Ezabatutakold = id
    };

    return Json(results);
}
```

Estekak:

***** ASP.net Core *****

<https://docs.microsoft.com/en-us/aspnet/core/>

LINQ:

<http://linsamples.com/>

***** Razor *****

<http://www.asp.net/web-pages/overview/getting-started/introducing-razor-syntax-%28c%29>

***** Orriak erabiltzen *****

<https://visualstudiomagazine.com/articles/2019/03/01/extending-razor-pages.aspx>

***** Bootstrap *****

<http://getbootstrap.com/css/>

http://librosweb.es/bootstrap_3/

<http://www.w3schools.com/bootstrap/default.asp>

***** CSS *****

<http://border-radius.com/>

*** Proiektua publikatu ***

<https://msdn.microsoft.com/en-us/library/31kztyey.aspx>

*** Produktuen slider***

<http://bootsnipp.com/snippets/featured/carousel-product-cart-slider>

*** Shopping Cart ***

<https://www.asp.net/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/shopping-cart>

<https://shakeelosmani.wordpress.com/2015/12/27/asp-net-mvc-step-by-step-complete-e-commerce-tutorial/>

<http://benjii.me/2016/07/using-sessions-and-httpcontext-in-aspnetcore-and-mvc-core/>

*** Erabiltzaileen identifikazioa ***

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity>

*** SSL eta beste ezaugarri batzuk***

<http://blogs.msdn.com/b/webdev/archive/2015/04/29/new-asp-net-features-and-fixes-in-visual-studio-2015-rc.aspx>