

ASP .NET

Visual Studio 2022

ERRONKA2 - ASP .NET - Restful API
kontsumitzen

REST (Representational state transfer):

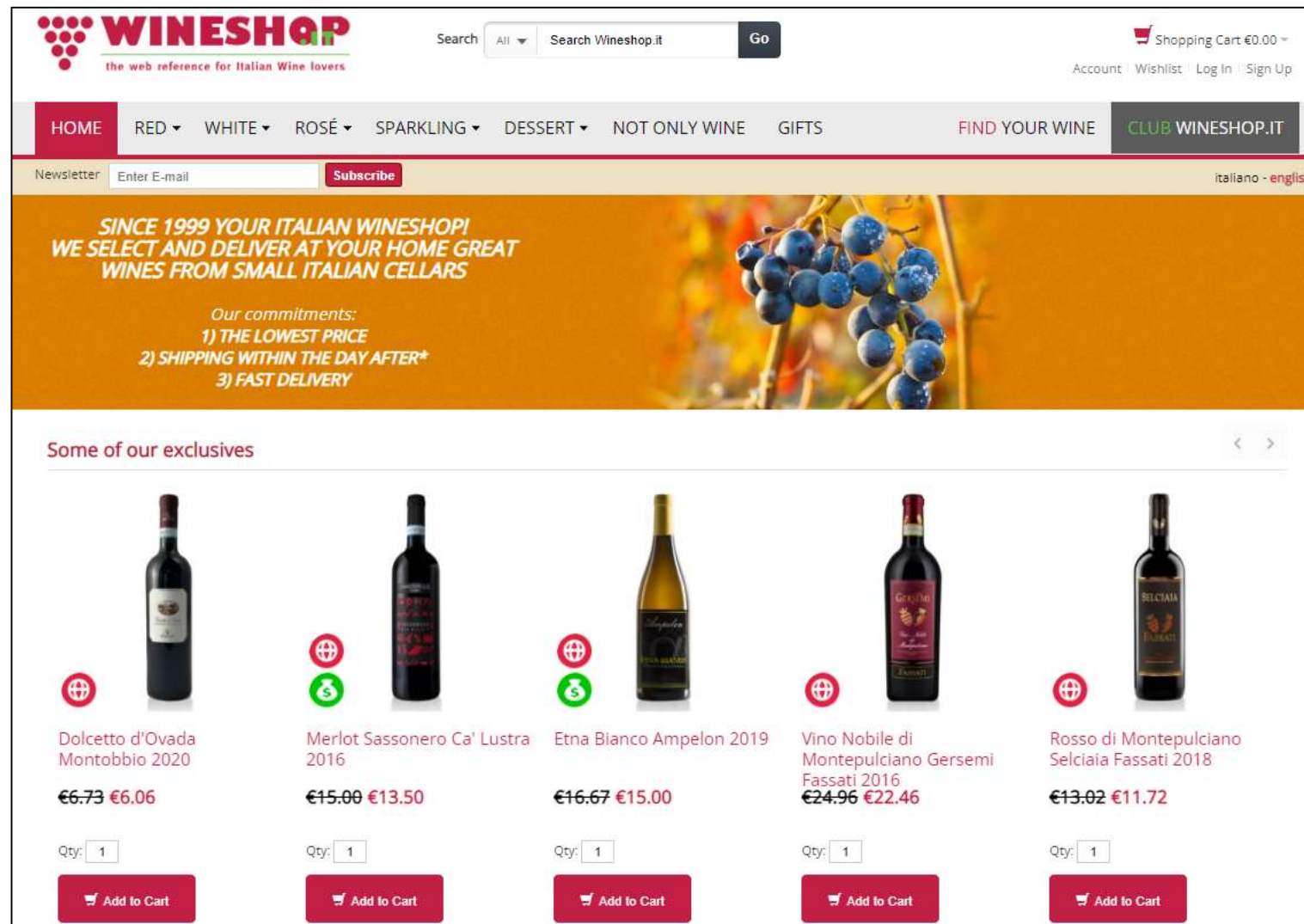
- HTTP protokoloa erabiltzen duen sistema arteko interfaze bat da.
- Formatu ezberdinetako (XML, JSON) datuak kudeatzeko prestatuta dago.
- Web aplikazioak, http erabiltzen badute, REST API- ak direla esaten da.
- Arkitektura estilo bat da, ez da framework edo protokolo berezi bat.
- REST aplikazio edo zerbitzu bat mota askotako framework eta lengoiaekin sortu daiteke.
- ASP.NET erabilia egiten den REST API-a, **Web API** deitzen da.

Gure helburua, REST API batek (Javarekin edo beste edozein lengoiaarekin) sortutako JSON datuak kudeatzea (bistaratu, berriak sortu, aldatu eta ezabatu) izango da.

WineShop web aplikazioa

1. Web aplikazio bat sortuko dugu eta honen antzeko zerbait egingo dugu:

<https://www.wineshop.it/en/>

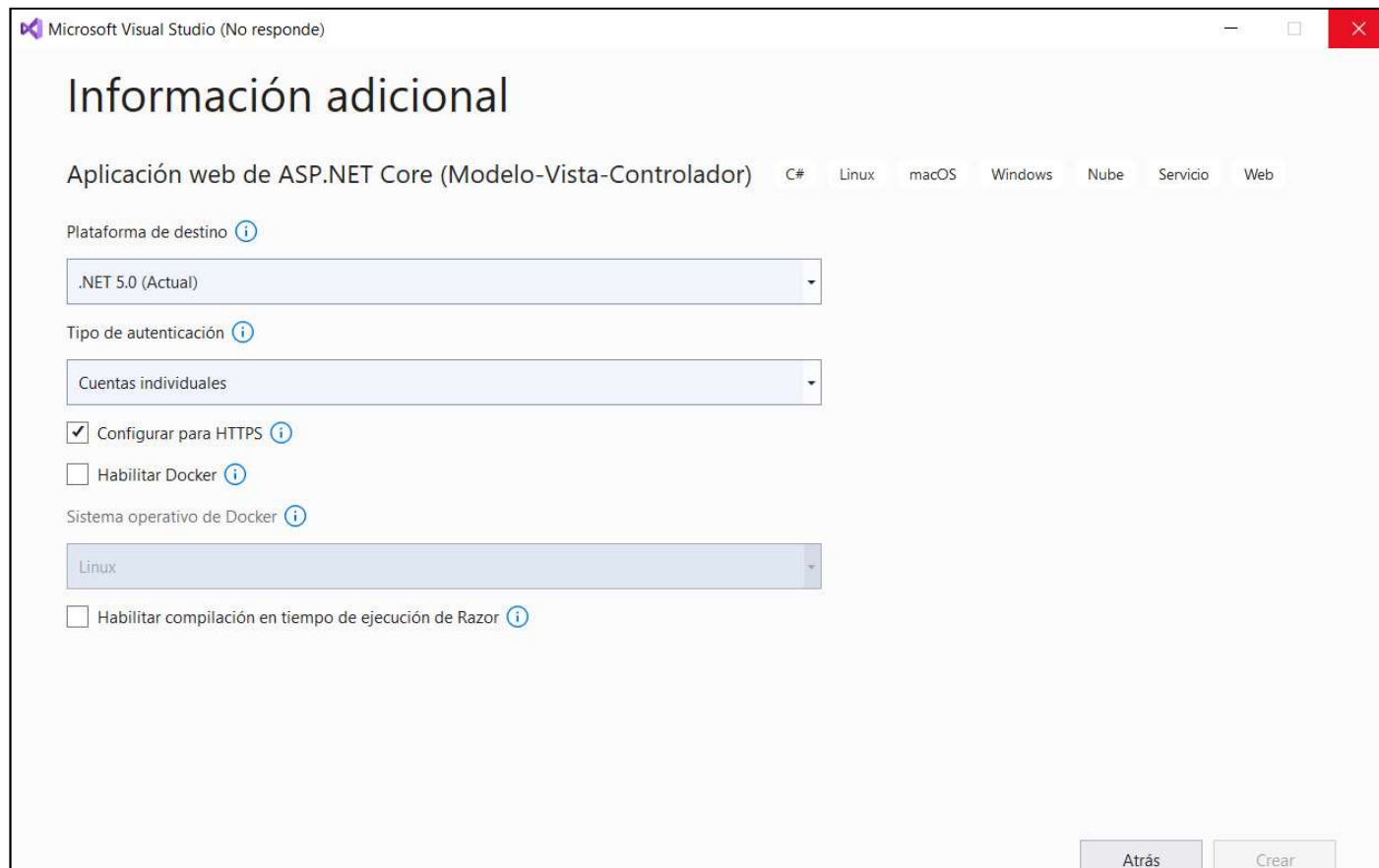


WineShop web aplikazioa

Visual Studio 2022 zabaldu eta proiektu berria sortu ezaugarri hauekin:

Aplicación web de ASP.net Core (Modelo-Vista-Controlador): WineShop .NET 6.0

Tipo de autenticación: Cuentas individuales



Microsoft Visual Studio (No responde)

Información adicional

Aplicación web de ASP.NET Core (Modelo-Vista-Controlador) C# Linux macOS Windows Nube Servicio Web

Plataforma de destino ⓘ

.NET 5.0 (Actual)

Tipo de autenticación ⓘ

Cuentas individuales

☒ Configurar para HTTPS ⓘ

☐ Habilitar Docker ⓘ

Sistema operativo de Docker ⓘ

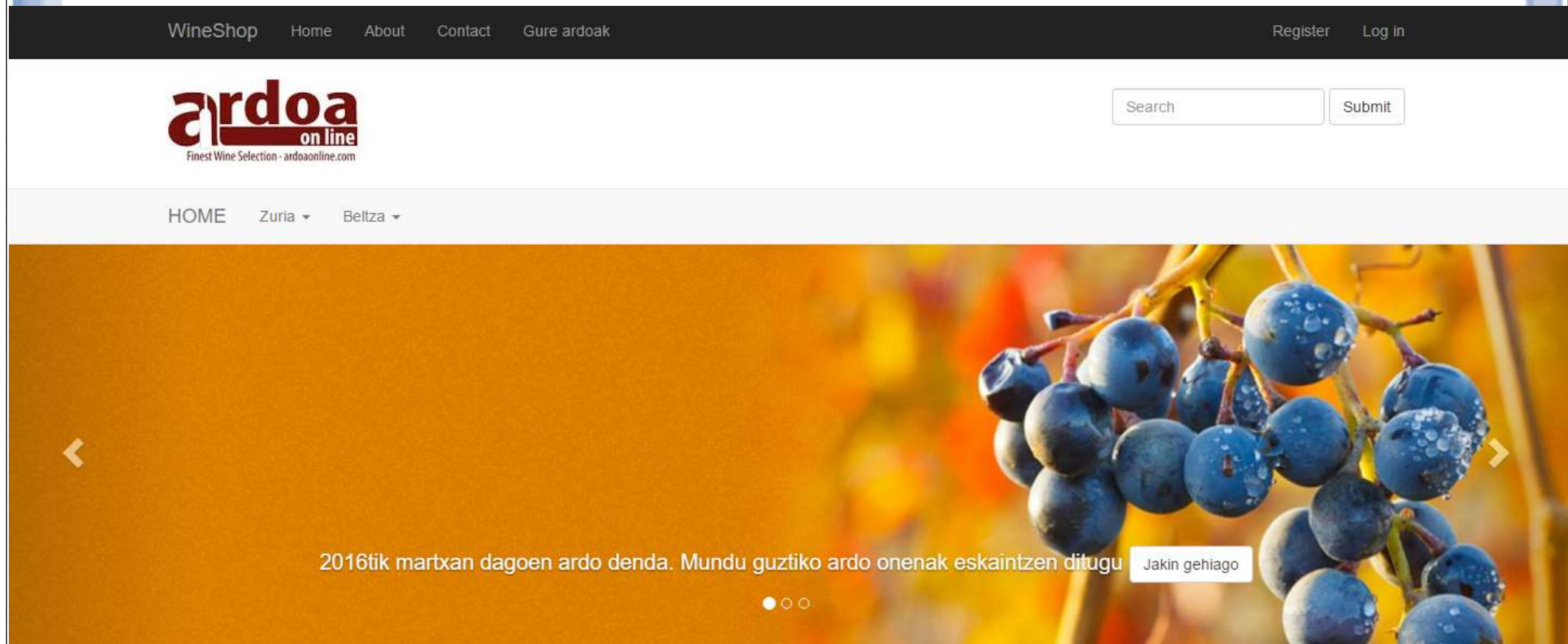
Linux

☐ Habilitar compilación en tiempo de ejecución de Razor ⓘ

Atrás Crear

WineShop web aplikazioa

- Home/Index aldatuko dugu, itxura hau edukitzeko:



WineShop web aplikazioa

- Carousela sortuko digu hiru irudi erabiliz: [banner1.jpg](#), [banner2.jpg](#) eta [banner3.jpg](#)
- Gaineko kapan beste bi kapa jarriko ditugu: bat logoa eta bestea menu bat izango da.
- Menua egiteko Bootstrap erabiliko dugu eta gure gustura aldatu:
Bootstrap instalatuta dator aplikazio barruan ([wwwroot-lib-bootstrap-dist...](#)) eta bertsioa ikusi dezakegu: [v5.1.0](#)

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

- [Navbar](#) erabiltzeko: components-navbar
- Bootstrapeko beste klase batzuk ere erabili behar dira, adibidez, [container](#) eta [container-fluid](#) markatzeko zein atal pantaila osoan agertu behar diran eta zein ez.
- Baita klase propioak sortu, adibidez marjinak kontrolatzeko (logoa, bilatu.....)

WineShop web aplikazioa_Layout.cshtml:

```

<link rel="stylesheet" href="~/lib/fontawesome/css/all.min.css" />
</head>
<body>
<header>
    .....
</header>
<div>
    <div class="container">
        <div class="row">
            <div class="col-md-6">
                
            </div>
            <div class="col-md-6">
                <form class="form-inline my-2 my-lg-0 float-right" id="searchForm">
                    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
                    <button class="btn btn-outline-danger my-2 my-sm-0" type="submit"><i class="fas fa-search"></i></button>
                </form>
            </div>
        </div>
    </div>
    <div>
        <main role="main" class="pb-3">
            @RenderBody()
        </main>
    </div>
</div>
<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>
<footer class="border-top footer text-muted">

```

Estiloak (wwwroot/css/site.css):

```

#searchForm {
    padding-top: 10%;
}

```

Images karpeta sortu (wwwroot barruan) eta behar den irudia sartu (ardoa.jpg)

Bootstrap jarrita dago baina font-awesome liburutegia ez (moodle-n)

WineShop web aplikazioa

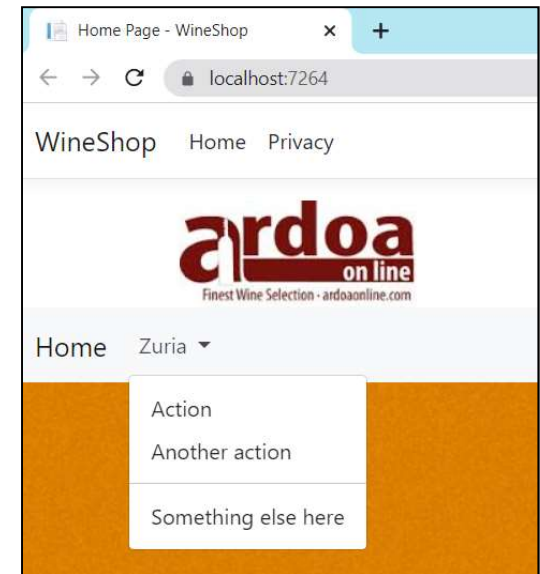
Home/Index.cshtml (irudiak moodle-n):

```
<div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2"></button>
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-label="Slide 3"></button>
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```


WineShop web aplikazioa

Home/Index.cshtml (bootstrap-eko navbar pertsonalizatuta. Karruselaren aurretik doa):

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Home</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown"
aria-expanded="false">
            Zuria
          </a>
          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>
```



WineShop web aplikazioa

«Zuria» zerrenda leku finko baten ikusteko:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light bg-menua">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Home</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse erlatiboa" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item dropdown full-width">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown"
aria-expanded="false">
            Zuria
          </a>
          .....
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Estiloak (wwwroot/css/site.css):

```
.bg-menua {
  background-color: #dddada !important;
}
.erlatiboa {
  position: relative;
}
.full-width.dropdown {
  position: static;
}
.full-width.dropdown > .dropdown-menu {
  left: 0;
}
.full-width.dropdown > .dropdown-menu > li > a {
  white-space: normal;
}
```

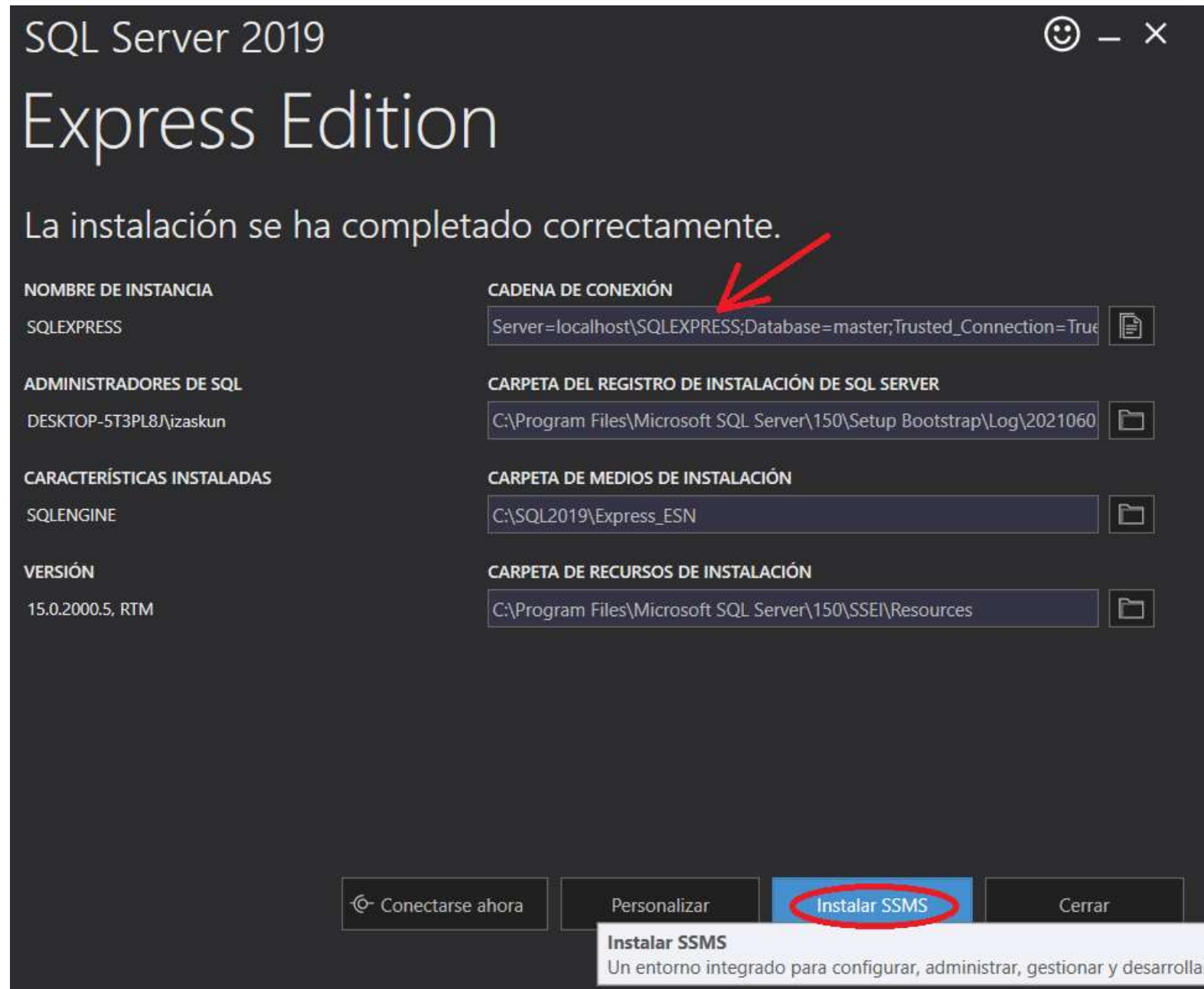
WineShop web aplikazioa

Bakoitzak bere REST API-a edukiko du (moodle-n dago)

Datu Basea instalatu

SQL Server:

- Lehenengo Datu Base zerbitzaria instalatu (**SQL SERVER EXPRESS, bertsio berriena topatu eta «Básica» aukeratu**). Bukatzeko dagoenean **SSMS** ere instalatu:
- SQL Server Konfigurazio tresnan, zerbitzaria martxan dagoela konprobatu.
 - Zerbitzariaren izena: **ekipoaren_izena**
- SSMS: “Sql Server Management Studio” ireki eta konfigurazioa aldatuko dugu: Herramientas – Opciones – Diseñadores : desmarkatu “Impedir guardar”
- Visual Studio-tik ere, datu baseak kukeatzeko aukera dugu. “Ver – explorador de servidores” eta datu basearentzako konexioa gehitu
- Datu base berriak egiterakoan: “autonumérico” motako eremua sortu nahi badugu, “int” motako eremua sortu eta eskuineko botoiarekin “propiedades” zabaldu eta “columna de identidad” lerroan eremua aukeratu. Adibidez, **gakoa**
int identity (1,1) not null



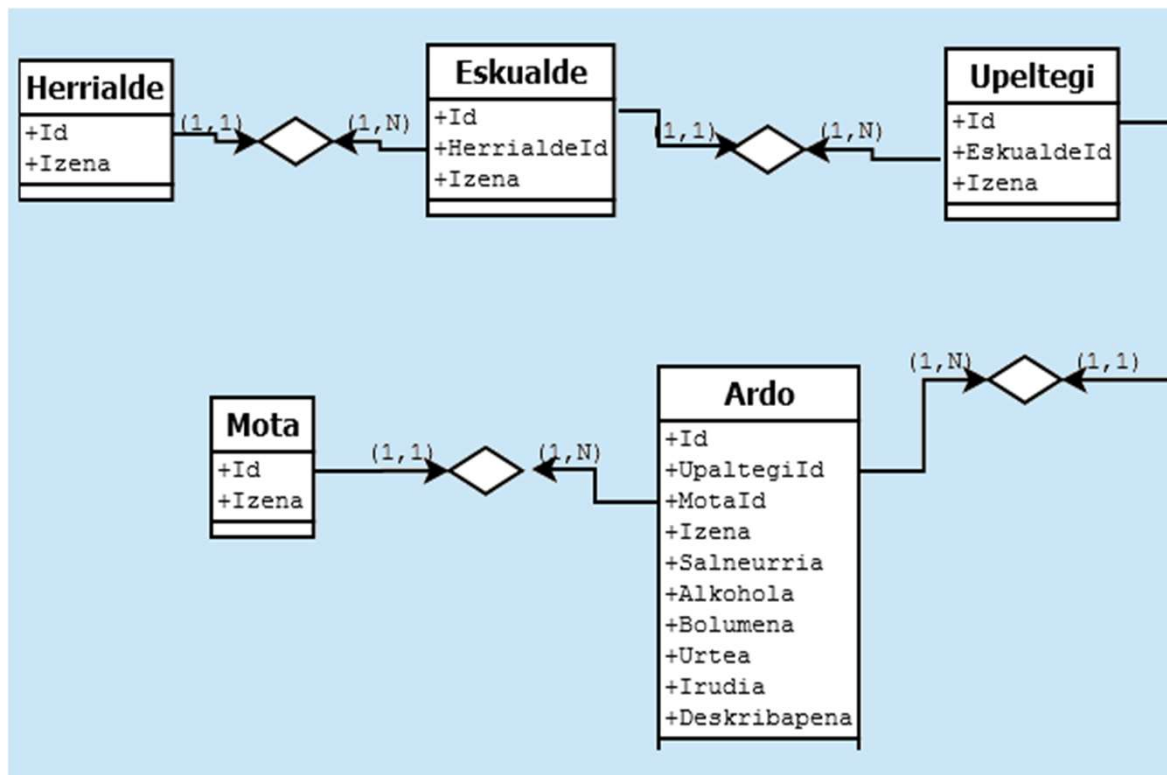
WineShop web aplikazioa

REST API-a erabiltzeko emango ditugun pausuak:

1. Erabiltzen duen datu basea sortu behar da (moodle-n **wineshop.sql** fitxategia)
2. Konfigurazioa aldatu bakoitza bere Datu Base zerbitzaria erabiltzeko
3. Martxan jarri eta nabigatzailetik konprobatu datuak ikusten direla:

<http://localhost:7049/api/Ardo> (portu zenbakia ezberdina izan daiteke)

Gure REST API-aren datu basea honelakoa da:



WineShop web aplikazioa

Menuan, ardo motak jarriko ditugu (zuria, gorria,), eta saguarekin gainetik pasatzerakoan upeltegi izenak agertuko dira



WineShop web aplikazioa

- **Services** karpeta sortuko dugu. Karpeta honetan gure zerbitzuak kokatuko ditugu, **rest api-arekin komunikazio guztiak kudeatzeko**. (Rest api-a erabili barik, datu base batetik irakurriko bagenu ere, zerbitzu hauek berdin sortuko genituzke. Ez da garrantzitsua informazioa non dagoen kokatuta).
- Zerbitzu bakoitzak bi osagai ditu: **interfazea** eta **implementazioa(klasea)**: **IArdoaService.cs** eta **ArdoaService.cs**
- Interfazea kontrato bezalakoa da. Metodoak nolakoak izan behar diren definitzen ditu, bakarrik, besterik ez. Implementazioa klase batean idatzi behar da.
- Models karpetan eredu berria sortuko dugu (rest api-ak bidaltzen digun informazioarekin bat datorrena): **ArdoaUpeltegi**

```
public class ArdoaUpeltegi
{
    public int Id { get; set; }
    public string Izena { get; set; }
    public string Upeltegilzena { get; set; }
}
```

WineShop web aplikazioa

- **Index.cshtml** dagoen menuko aukerak automatikoki sortzeko aukera ezberdinak ditugu:

<http://asp.net-hacker.rocks/2016/02/18/extending-razor-views.html>

- **ViewComponent** erabiliko dugu.

<https://docs.microsoft.com/es-es/aspnet/core/mvc/views/view-components?view=aspnetcore-6.0>

WineShop web aplikazioa

- **ViewComponents** karpeta sortuko dugu, eta barruan klase berri bat: **ArdoakViewComponent.cs**, Controller bezala funtzionatzen du eta zerbitzua erabiliko du. Zerbitzuak injektatu egiten dira. (<https://docs.microsoft.com/es-es/aspnet/core/mvc/controllers/dependency-injection?view=aspnetcore-6.0>)
- .

```
using Microsoft.AspNetCore.Mvc;
using WineShop.Services;
public class ArdoakViewComponent : ViewComponent
{
    private readonly IArdoaService _ardoaService;

    public ArdoakViewComponent(IArdoaService ardoaService)
    {
        _ardoaService = ardoaService;
    }
    public async Task<IViewComponentResult> InvokeAsync(int id)
    {
        return View(await _ardoaService.GetMota(id));
    }
}
```

WineShop web aplikazioa

IArdoaService.cs: (Interfazea)

```
public interface IArdoaService
{
    Task<IList<ArdoaUpeltegi>> GetMota(int id);
}
```

ArdoaService.cs: (implementazioa)

```
public class ArdoaService : IArdoaService
{
    //Rest api-aren Url-a jarri, portua ezberdina izan daiteke

    private Uri rutaTodos = new Uri("https://localhost:7049/api/Ardoa/");

    public async Task<IList<ArdoaUpeltegi>> GetMota(int id)
    {
        List<ArdoaUpeltegi> ardoaUpeltegiList = new List<ArdoaUpeltegi>();
        Uri rutaUpeltegiak = new Uri(rutaTodos, "Mota/" + id.ToString());
        using (var httpClient = new HttpClient())
        {
            using (var response = await httpClient.GetAsync(rutaUpeltegiak))
            {
                string apiResponse = await response.Content.ReadAsStringAsync();
                ardoaUpeltegiList = JsonConvert.DeserializeObject<List<ArdoaUpeltegi>>(apiResponse);
            }
        }
        return ardoaUpeltegiList;
    }
}
```

WineShop web aplikazioa

Zerbitzuak (interfazea eta klasea) hasierako fitxategian definitu behar ditugu, gero Controller eta ViewComponent ezberdinetan injektatzeko:

Program.cs:

```
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromSeconds(900);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});

//Gure zerbitzuak
builder.Services.AddScoped<IArdoaService, ArdoaService>();

var app = builder.Build();
```

WineShop web aplikazioa

- **Components** karpeta sortu behar da **Views/Shared** barruan
- Eta horren barruan beste karpeta bat: **Ardoak**
- Eta horren barruan bista bat: **Default.cshtml** («Agregar» - «nuevo elemento con scaffolding» - «vista de razor»)

×

Agregar Vista de Razor

Nombre de vista

default

Plantilla

List

Clase de modelo

ArdoaUpeltegi (WineShop.Models)

Clase de contexto de datos

<Obligatoria>

Opciones

☐ Crear como vista parcial

☒ Hacer referencia a bibliotecas de scripts

☒ Usar página de diseño

...

(Dejar en blanco si se define en un archivo _viewstart de Razor)

Agregar

Cancelar

WineShop web aplikazioa

Default.cshtml

```
@model IEnumerable<WineShop.Models.ArdoaUpeltegi>

<ul class="dropdown-menu">

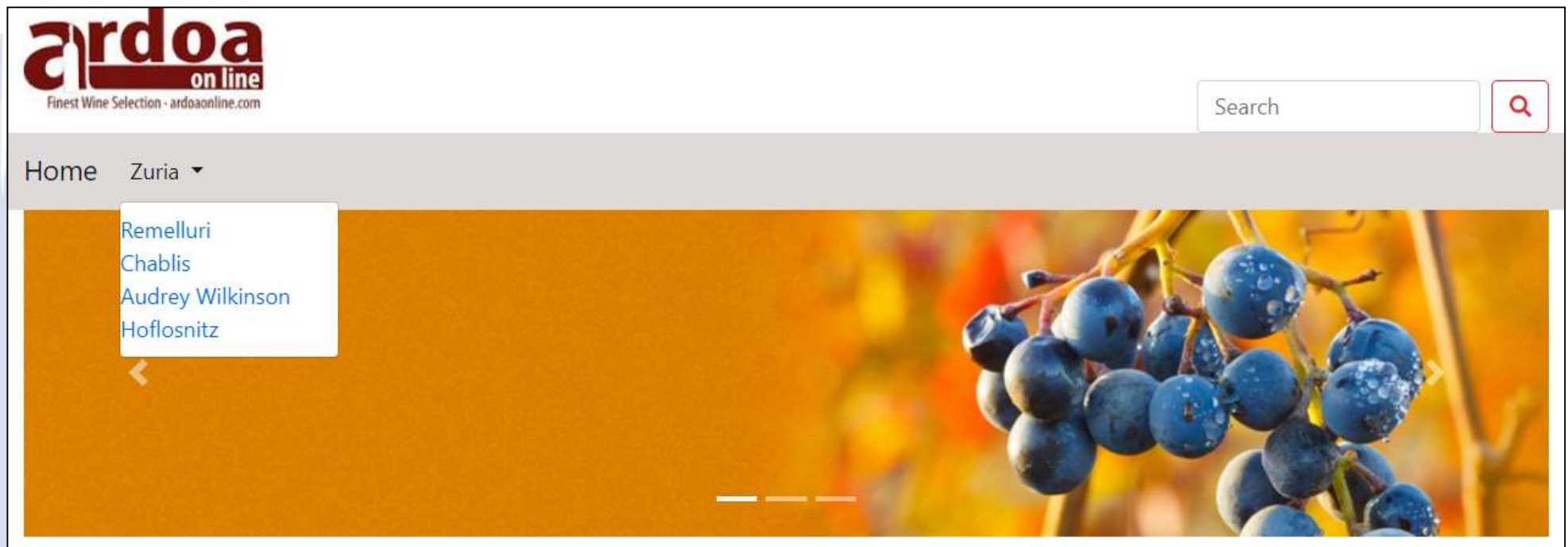
    @foreach (var item in Model)
    {
        <li><a href="#">@Html.DisplayFor(modelItem => item.Upeltegilzena)</a></li>
    }
</ul>
```

Eta Views/Home/Index.html barruan

```
<li class="nav-item dropdown full-width">
    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown"
    aria-haspopup="true" aria-expanded="false">
        Zuria
    </a>
    @await Component.InvokeAsync("Ardoak", new { id = 1 })
</li>
```

WineShop web aplikazioa

Eta exekutatzen badugu:



WineShop web aplikazioa

Beste aldaketa batzuk, estiloak aldatzeko adibidean bezala: Default.cshtml

@model IEnumerable<WineShop.Models.ArdoaUpeltegi>

```
<ul class="dropdown-menu">
<li>
  <div class="container padinaMenu">
    <div class="col-md-4 nireMenu">
      <h4 class="borgona">Aukeratu upeltegia:</h4>
      @{var izena = "";
        foreach (var item in Model)
        {
          if (item.Upeltegizena != izena)
          {
            <a href="#">@Html.DisplayFor(modelItem => item.Upeltegizena)</a><br />
            izena = item.Upeltegizena;
          }
        }
      }
    </div>
    <div class="col-md-8">
      
    </div>
  </div>
</li>
</ul>
```

WineShop web aplikazioa

Gure menua itxura hau edukitzeko:

Site.css

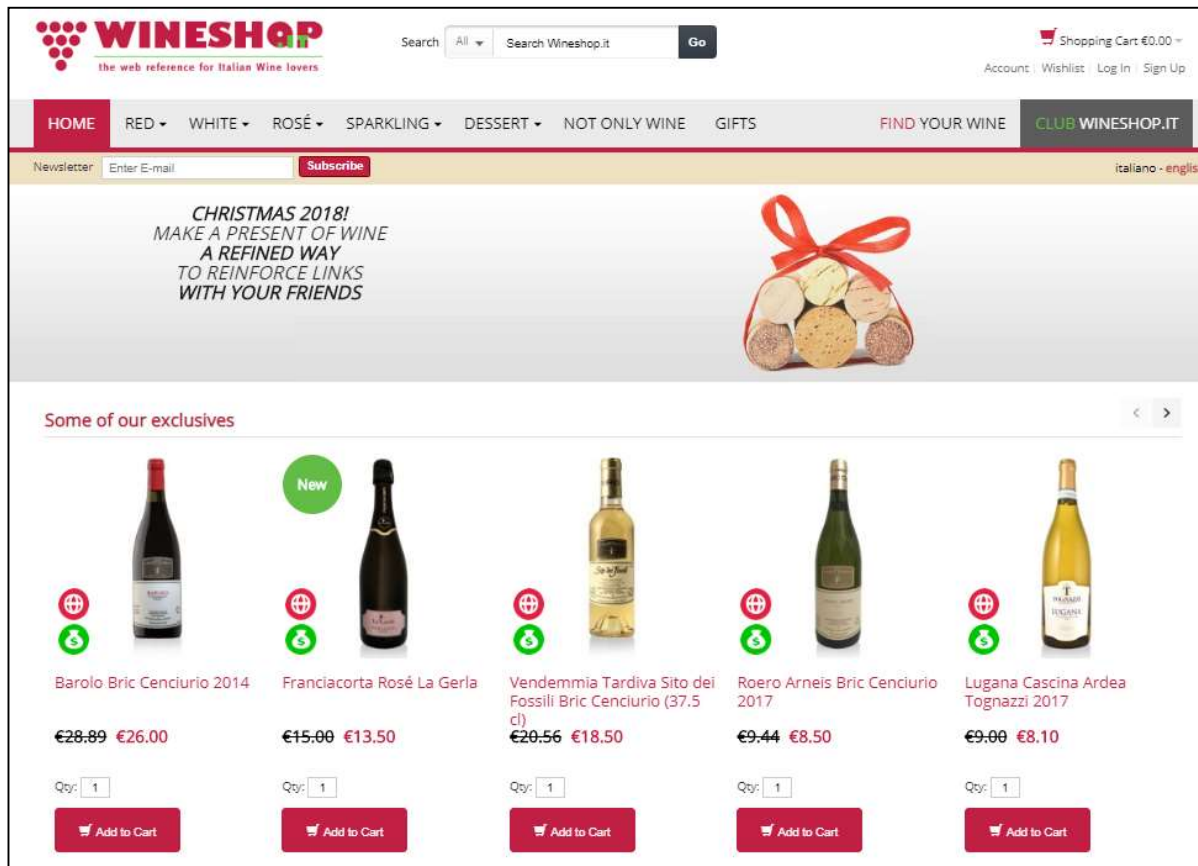
```
@media screen and (min-width:480px) {  
  .dropdown:hover .dropdown-menu {  
    display: block;  
    margin-top: 0;  
  }  
}  
.padinaMenu {  
  padding: 15px;  
}  
.borgona {  
  color: #981212;  
}  
.nireMenua .active a,  
.nireMenua .active a:hover {  
  background-color: #981212;  
}  
.nireMenua > a {  
  color: #777;  
  text-decoration:none;  
}  
.nireMenua > a:hover {  
  color: #981212;  
}
```

Menua agertzeko saguarekin gainetik pasatzerakoan

Berdina egingo dugu esteka guztietan. Mota guztien estekak egingo ditugu

WineShop web aplikazioa

Eskaintzen karrusela egingo dugu, eta horretarako rest api-ak ardoen informazio guztia bueltatzen duen url-a daukagu: <https://localhost:7049/api/Ardoa/>



WineShop web aplikazioa

IArdoaService.cs

```
namespace WineShop.Services
{
    public interface IArdoaService
    {
        .....

        Task<List<Ardoa>> GetArdoak();
    }
}
```

ArdoaService.cs

```
namespace WineShop.Services
{
    .....

    public async Task<List<Ardoa>> GetArdoak()
    {
        List<Ardoa> ardoaList = new List<Ardoa>();
        using (var httpClient = new HttpClient())
        {
            using (var response = await httpClient.GetAsync(rutaTodos))
            {
                string apiResponse = await response.Content.ReadAsStringAsync();
                ardoaList = JsonConvert.DeserializeObject<List<Ardoa>>(apiResponse);
            }
        }
        return ardoaList;
    }
}
```


WineShop web aplikazioa

Eredu berria sortu behar dugu: Ardoa.cs (Models karpetan barruan)

```
public class Ardoa
{
    public int Id { get; set; }
    public string Izena { get; set; }
    public Decimal Salneurria { get; set; }
    public int Alkohola { get; set; }
    public int Edukia { get; set; }
    public int Urtea { get; set; }
    public string Irudia { get; set; }
    public string Deskribapena { get; set; }
    public bool Eskaintza { get; set; }
    public int Deskontua { get; set; }
    public int Upeltegiald { get; set; }
    public int Motald { get; set; }
}
```

WineShop web aplikazioa

Zerbitzua erabili behar dugun kontroladorean injektatzen da:

HomeController.cs

```
public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;
    private readonly IArdoaService _ardoaService;

    public HomeController(ILogger<HomeController> logger, IArdoaService ardoaService)
    {
        _logger = logger;
        _ardoaService = ardoaService;
    }

    public async Task<ActionResult> Index()
    {
        List<Ardoa> ardoaList = new List<Ardoa>();
        ardoaList = await _ardoaService.GetArdoak();

        return View(ardoaList.Where(a => a.Eskaintza == true));
    }
}
```

WineShop web aplikazioa

Eskaintzen karrusela egiteko, eginda dagoen kodigo hau erabiliko dugu oinarritzat (Home-Index):

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

**** Esteka honetatik css-a kopiatu eta Site.css fitxategian itsatsi. Behean dauden moldaketak egin ere bai. ****

Adaptazioak eginda, gure datu basetik irakurtzen ditugun eskaintzak erakusteko (moodle-en kodigo osoa)

```
<div class="carousel-inner">
    @{var itemLerroko = 0;
      var fila1 = false;}
    @{
      foreach (var item in Model)
      {
        if (itemLerroko == 0)
        { if (fila1 == false)
          {
            @:<div class="carousel-item
active">
              @: <div class="row">
                }
              else
              {
                @:<div class="carousel-item">
                @: <div class="row">
                  }
              }
            }
          }
        }
      }
    <div class="col-sm-3">
```

.....

```
.....
</div>
    itemLerroko++;
    fila1 = true;
    if (itemLerroko > 3)
    {
      @:</div>
      @:</div>
      itemLerroko = 0;
    }
  }
```

Site.css

```
.tachado {
  text-decoration: line-through;
}

.col-item .photo img {
  margin: 0 auto;
  /*width: 100%;*/
}

.col-item .irudia {
  height: 150px;
  width: auto;
}
```