**GRAFICAS CON Highcharts**

* CREAMOS UN CONTROLADOR EN BLANCO

Dashboard

* EN VISTAS

Se crea una carpeta llamada Dashboard

Creamos una vista para el index.

* Creamos una carpeta para los scripts de las graficas

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **Función de la gráfica de pastel Pastel.js**

**Adaptamos la grafica para que consuma un api get y reciba los datos por medio de una variable data:**

//Vamos a consumir datos de una api por medio del metodo Get

//y archivos json

$(document).ready(function () {

//Peticion a una api

//iniciamos un metodo ajax

$.ajax({

type: "GET",

contentType: "aplication/json; charset=utf-8",

dataType: "json",

url: urlBase + '/DataPastel',

//si hay un error enviamos un alert por medio de una funcion.

error: function () {

alert("Ocurrio un error al obtener los Datos");

},

//si todo salio con exito

success: function (data) {

console.log("DATA RECIBIDA:")

console.log(data);

//le enviamos la data a la grafica pastel

GraficaPastel(data);

}

})

});

function GraficaPastel(data) {

// div id="" donde se construye la grafica

Highcharts.chart('pastel', {

chart: {

plotBackgroundColor: null,

plotBorderWidth: null,

plotShadow: false,

type: 'pie'

},

title: {

text: 'Fremeworks más Usados en el 2022'

},

tooltip: {

pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'

},

accessibility: {

point: {

valueSuffix: '%'

}

},

plotOptions: {

pie: {

allowPointSelect: true,

cursor: 'pointer',

dataLabels: {

enabled: false

},

showInLegend: true

}

},

series: [{

name: 'Framework',

colorByPoint: true,

//es la data que viene como parametro en un json

data: data

}]

});

}

* **HTML VISTA INDEX DE LA GraficaPastel**

**Creamos una urlBase para enviarla a la graficaPastel**

@{

ViewBag.Title = "Index DashBoard";

}

<h2>@ViewBag.Title</h2>

<!--div que va a renderizar la grafica-->

<div id="pastel"></div>

<!--Scriptes que utiliza la grafica-->

<!--agregamos tambien jquery-->

@section Scripts{

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script src="https://code.highcharts.com/highcharts.js"></script>

<script src="https://code.highcharts.com/modules/exporting.js"></script>

<script src="https://code.highcharts.com/modules/export-data.js"></script>

<script src="https://code.highcharts.com/modules/accessibility.js"></script>

<!--el script de configuración de la grafica-->

<script src="~/Scripts/Graficas/Pastel.js"></script>

<!--script para obtener la url base-->

<script>

//Creamos la urlBase que enviaremos a la grafica Pastel https://localhost:44360/Dashboard

var urlBase = 'https://' + '@Context.Request.Url.Host' + ':' + '@Context.Request.Url.Port' + '@Context.Request.Path';

console.log("URL:");

console.log(urlBase);

</script>

}

* **MODELO SeriePastel**

**Inicializa las propiedades en el constructor y tiene un método que crea datos estáticos de tipo SeriePstel**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Graficas.Models

{

public class SeriePastel

{

public string name { get; set; }

public double y { get; set; }

public bool sliced { get; set; }

public bool selected { get; set; }

//creamos un constructor vacio

public SeriePastel()

{

}

//creamos un constructor para inicializar el objeto

//como no van a llevar informacion ponemos a sliced y selectef = false

public SeriePastel(string name, double y, bool sliced=false, bool selected=false)

{

this.name = name;

this.y = y;

this.sliced = sliced;

this.selected = selected;

}

//Creamos un metodo que devuelva un listado de data, aqui la información podria venir de una DB

//en este caso es data statica

public List<SeriePastel> GetDataDummy()

{

List<SeriePastel> lista = new List<SeriePastel>();

//agregamos a la lista objetos de tipo SeriePastel

lista.Add(new SeriePastel("Angular", 45));

lista.Add(new SeriePastel("VueJS", 50));

lista.Add(new SeriePastel("ReactJS", 60));

lista.Add(new SeriePastel("CSS3", 34));

lista.Add(new SeriePastel("HTML5", 28));

return lista;

}

}

}

* **CONTROLLER**

**Crea un objeto de tipo SeriePastel y obtiene los datos estáticos pastel.GetDataDummy y los envia a la vista index del controller Dashboard, en un formato json.**

using Graficas.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Graficas.Controllers

{

public class DashboardController : Controller

{

// GET: Dashboard

public ActionResult Index()

{

return View();

}

//metodo que crea un objeto serie de tipo SeriePastel y le envia

//la data del metodo GetDataDummy en formato json

public JsonResult DataPastel()

{

SeriePastel pastel = new SeriePastel();

return Json(pastel.GetDataDummy(), JsonRequestBehavior.AllowGet);

}

}

}

URL DataPastel json

<https://localhost:44360/Dashboard/DataPastel>

Texto

Descripción generada automáticamente

* **VISTA INDEX DEL CONTROLLER DASHBOARD**

LA vista renderiza la grafica en el div id=pastel

Por medio del <script src="~/Scripts/Graficas/Pastel.js"></script>

Que envia la urlBase al archivo pastel.js

@{

ViewBag.Title = "Index DashBoard";

}

<h2>@ViewBag.Title</h2>

<!--div que va a renderizar la grafica-->

<div id="pastel"></div>

<!--Scriptes que utiliza la grafica-->

<!--agregamos tambien jquery-->

@section Scripts{

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script src="https://code.highcharts.com/highcharts.js"></script>

<script src="https://code.highcharts.com/modules/exporting.js"></script>

<script src="https://code.highcharts.com/modules/export-data.js"></script>

<script src="https://code.highcharts.com/modules/accessibility.js"></script>

<!--el script de configuración de la grafica-->

<script src="~/Scripts/Graficas/Pastel.js"></script>

<!--script para obtener la url base-->

<script>

//Creamos la urlBase que enviaremos a la grafica Pastel https://localhost:44360/Dashboard

var urlBase = 'https://' + '@Context.Request.Url.Host' + ':' + '@Context.Request.Url.Port' + '@Context.Request.Path';

console.log("URL:");

console.log(urlBase);

</script>

}

* **Pastel.js**

**Recibe la url base y le agrega /DataPastel**

**Que es donde viene el json con la data**

//Vamos a consumir datos de una api por medio del metodo Get

//y archivos json

$(document).ready(function () {

//Peticion a una api

//iniciamos un metodo ajax

$.ajax({

type: "GET",

contentType: "aplication/json; charset=utf-8",

dataType: "json",

url: urlBase + '/DataPastel',

//si hay un error enviamos un alert por medio de una funcion.

error: function () {

alert("Ocurrio un error al obtener los Datos");

},

//si todo salio con exito

success: function (data) {

console.log("DATA RECIBIDA:")

console.log(data);

//le enviamos la data a la grafica pastel

GraficaPastel(data);

}

})

});

function GraficaPastel(data) {

// div id="" donde se construye la grafica

Highcharts.chart('pastel', {

chart: {

plotBackgroundColor: null,

plotBorderWidth: null,

plotShadow: false,

type: 'pie'

},

title: {

text: 'Fremeworks más Usados en el 2022'

},

tooltip: {

pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'

},

accessibility: {

point: {

valueSuffix: '%'

}

},

plotOptions: {

pie: {

allowPointSelect: true,

cursor: 'pointer',

dataLabels: {

enabled: false

},

showInLegend: true

}

},

series: [{

name: 'Framework',

colorByPoint: true,

//es la data que viene como parametro en un json

data: data

}]

});

}