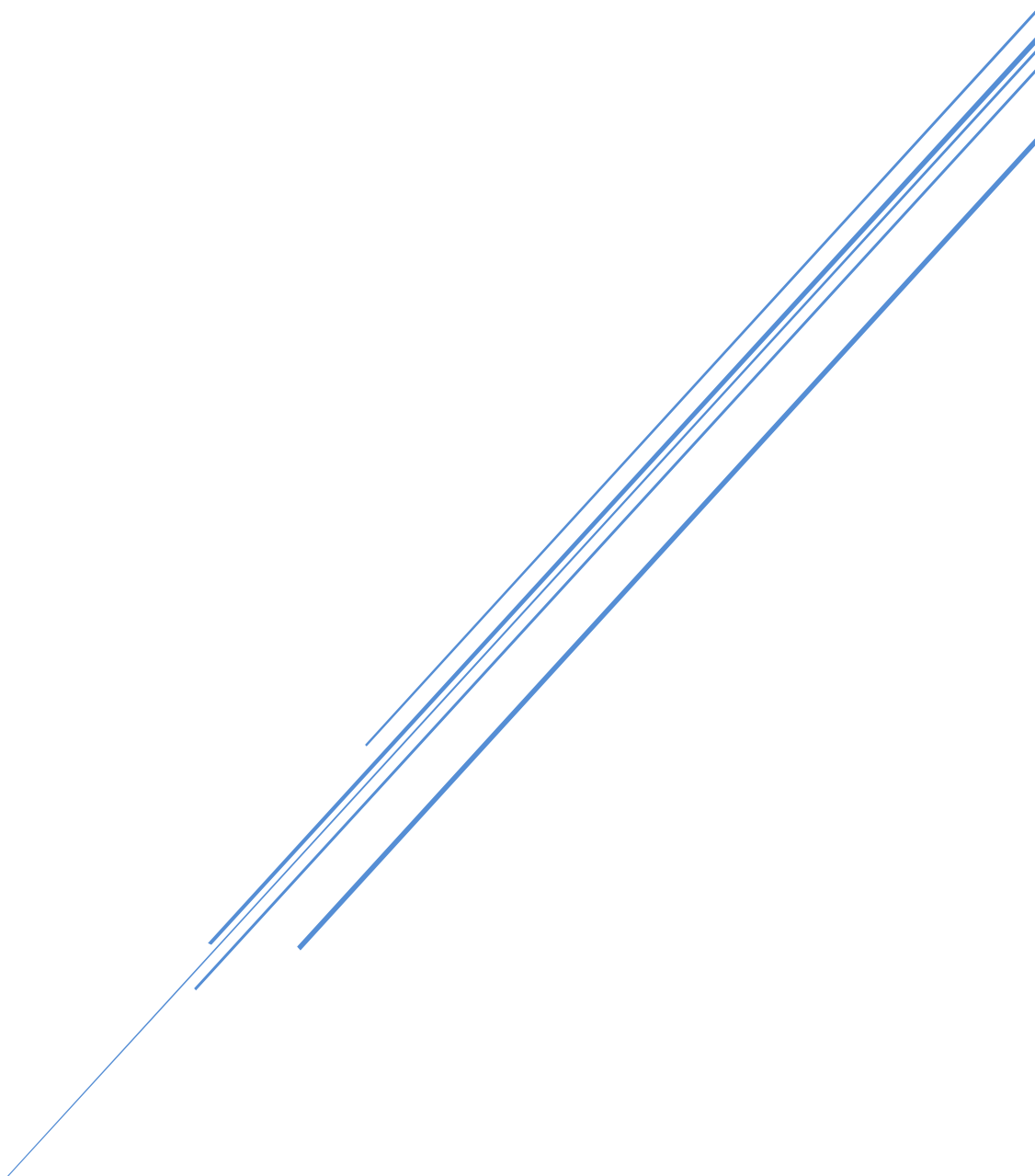


# TALLER DE DESEMPEÑO PROFESIONAL

Semana 11



UPC  
2016-01

## SignalR 2 and MVC 5

### Descripción general

Este tutorial presenta el desarrollo de aplicaciones web en tiempo real con ASP.NET SignalR 2 y ASP.NET MVC 5.

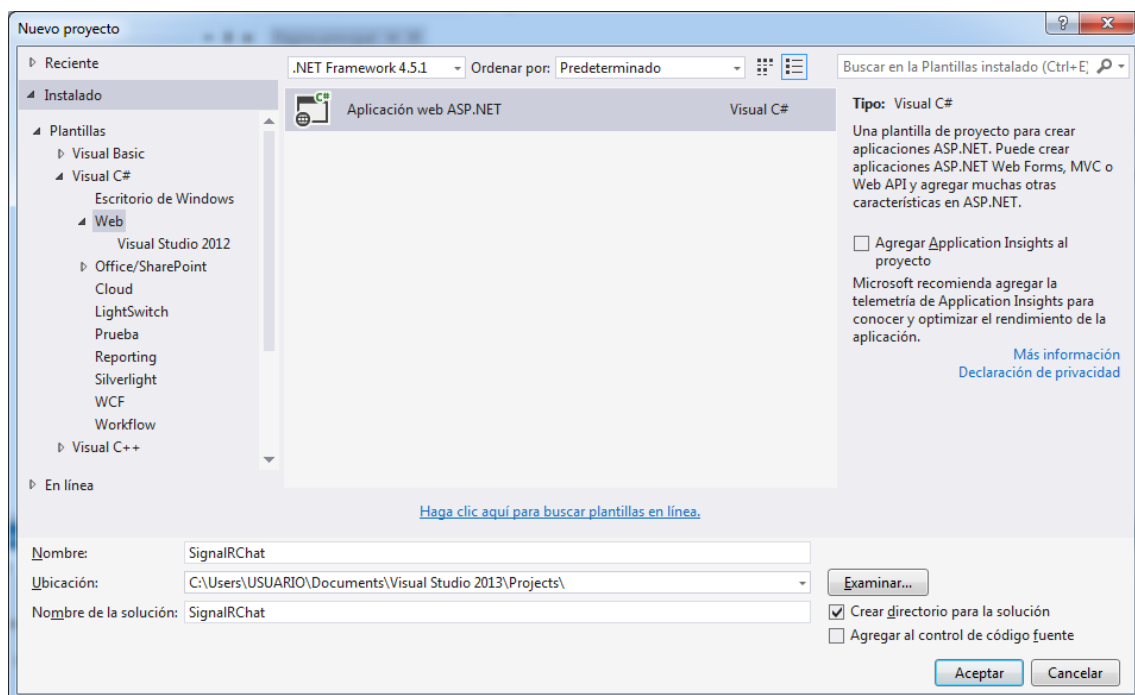
En este tema se explicarán los siguientes puntos:

- ✓ Agregar la librería SignalR a una aplicación MVC 5.
- ✓ Creación de “hub” y “OWIN” startup classes
- ✓ Uso de la librería SignalR jQuery en una página web para enviar mensajes y mostrar actualizaciones desde el “hub”.

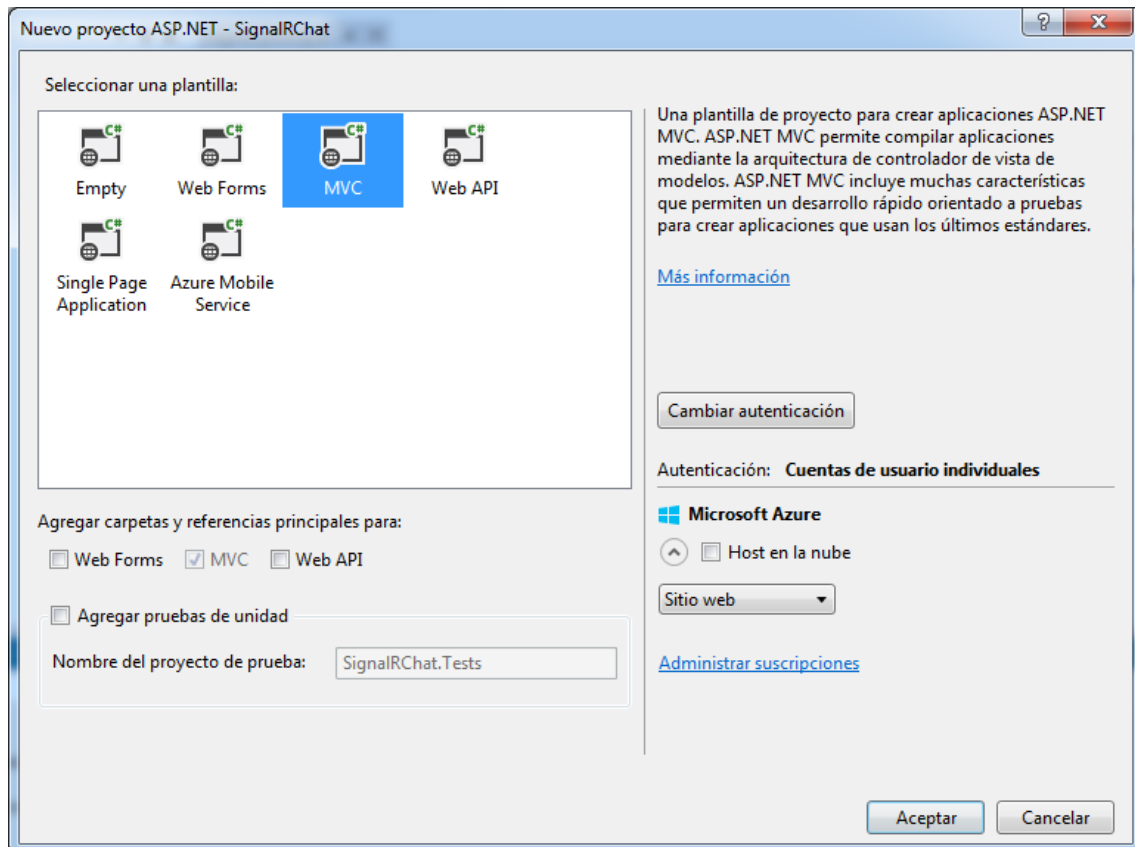
La siguiente captura de pantalla muestra la aplicación de chat completa que se ejecuta en un navegador.

### Creación del proyecto

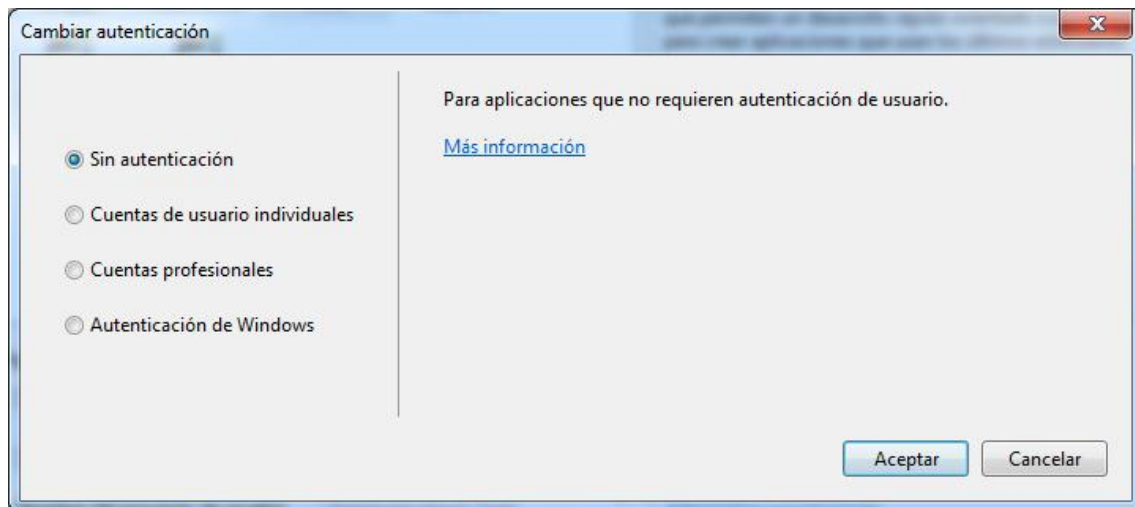
Creamos una aplicación C# ASP.NET utilizando .NET Framework 4.5.



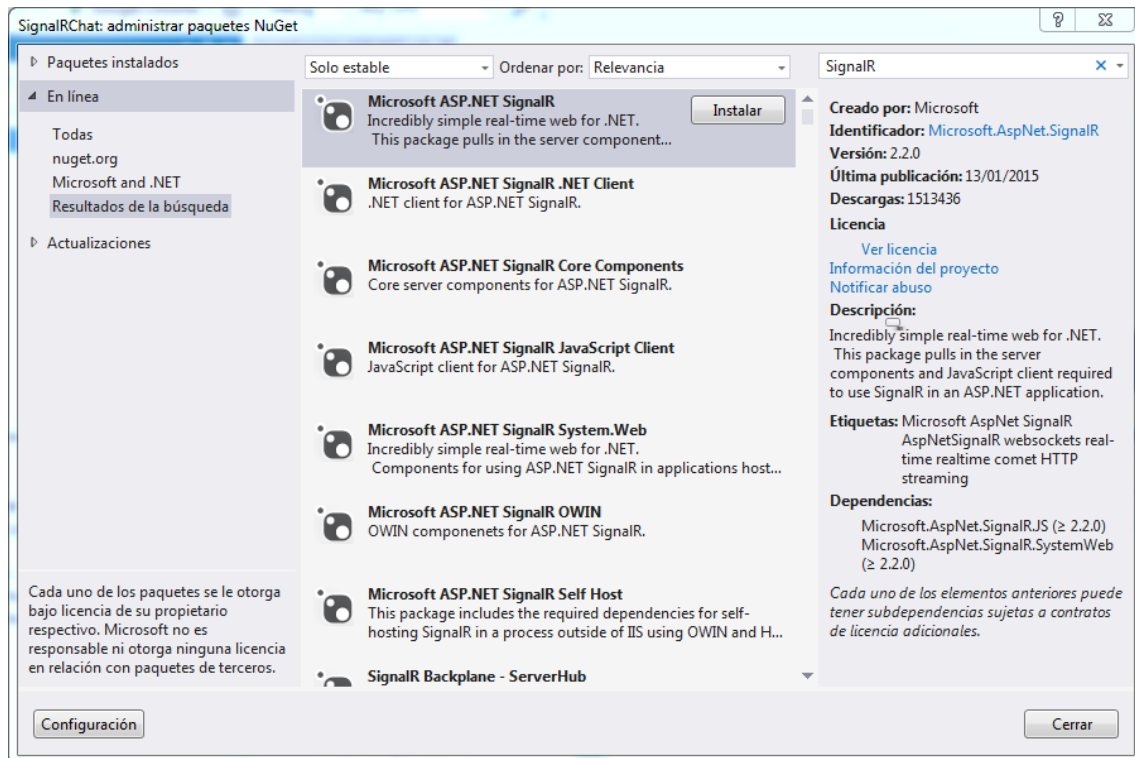
Seleccionamos la plantilla MVC



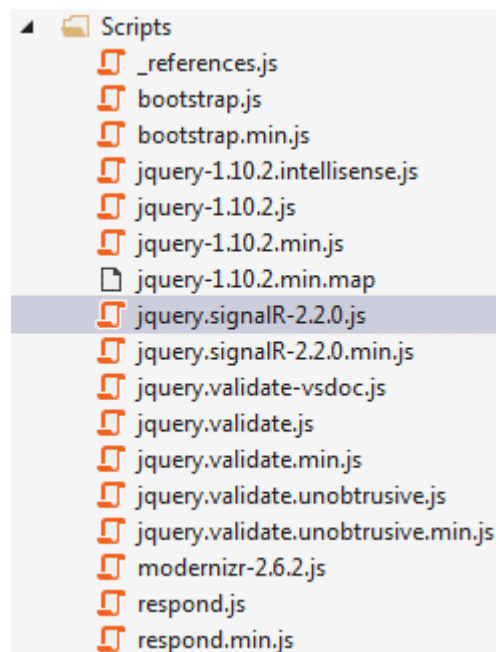
Cambiamos la autenticación y seleccionamos “Sin autenticación”



Seleccionamos “Administrar paquetes NuGet” en el proyecto e instalamos SignalR



Esto agrega al proyecto un conjunto de Scripts y referencias de ensamblaje que habilitan la funcionalidad SignalR



En la carpeta Models creamos la crease User para almacenar los usuarios que se encuentren en línea

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

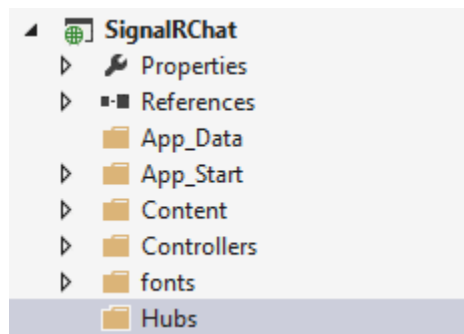
namespace SignalRChat.Models
{
```

```

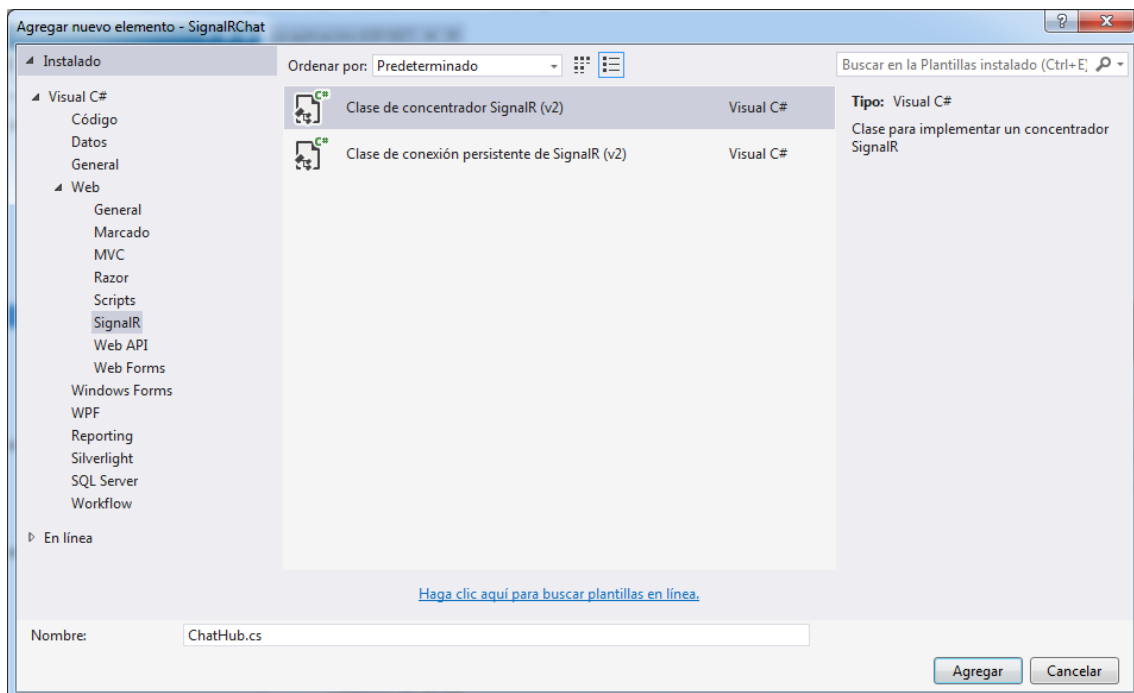
public class User
{
    public string id;
    public string name;
    public string url;
}

```

Creamos una carpeta denominada Hubs en el proyecto



En la carpeta Hubs, creamos una clase de concentrador SignalR (v2) y la denominamos ChatHub



Reemplazamos el código de la clase generada con el siguiente

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Microsoft.AspNet.SignalR;
//Importaciones
using System.Web.Mvc;
using System.Web.Script.Serialization;
using SignalRChat.Models;

```

```

namespace SignalRChat.Hubs
{
    public class ChatHub : Hub
    {
        static List<User> users = new List<User>();
        static long counter = 0; //Contar usuarios conectados

        public void Send(string name, string picture, string message)
        {
            // Call the addNewMessageToPage method to update clients.
            Clients.All.addNewMessageToPage(name, picture, message);
        }

        public void Update(string name, string url)
        {
            var context =
GlobalHost.ConnectionManager.GetHubContext<ChatHub>();

            User aux = new User();
            aux.id = Context.ConnectionId;
            aux.name = name;
            aux.url = url;
            users.Add(aux);

            UpdateHub(); //Actualiza listado
        }

        public override System.Threading.Tasks.Task OnConnected()
        {
            counter = counter + 1; //Nuevo usuario conectado
            var context =
GlobalHost.ConnectionManager.GetHubContext<ChatHub>();
            context.Clients.All.online(counter);
            return base.OnConnected();
        }

        public override System.Threading.Tasks.Task OnReconnected()
        {
            counter = counter + 1; //Usuario re-conectado
            var context =
GlobalHost.ConnectionManager.GetHubContext<ChatHub>();
            context.Clients.All.online(counter);
            return base.OnReconnected();
        }

        public override System.Threading.Tasks.Task OnDisconnected(bool
stopCalled)
        {
            counter = counter - 1; //Usuario desconectado
            var context =
GlobalHost.ConnectionManager.GetHubContext<ChatHub>();
            context.Clients.All.online(counter);

            string current_id = Context.ConnectionId; //Id de la conexion
            users.RemoveAll(x => x.id == current_id);

            UpdateHub(); //Actualiza listado
            return base.OnDisconnected(stopCalled);
        }

        public void UpdateHub()
        {

```

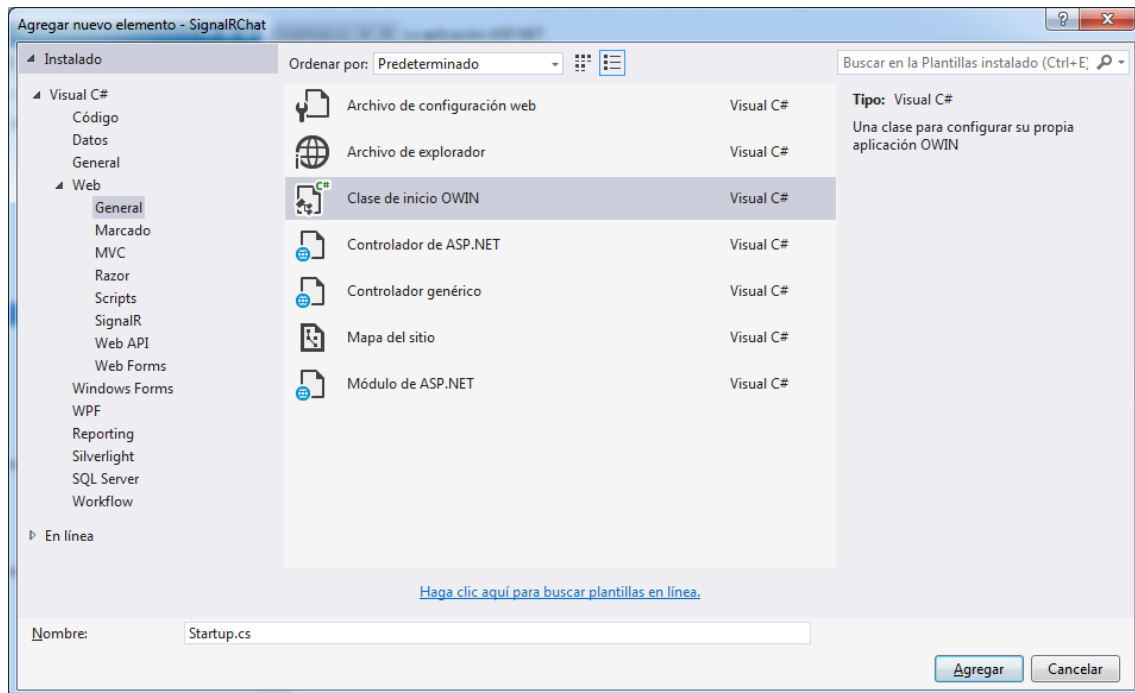
```

        var context =
GlobalHost.ConnectionManager.GetHubContext<ChatHub>();
        //Actualizar usuarios
        JavaScriptSerializer jss = new JavaScriptSerializer();
        string output = jss.Serialize(users);

        context.Clients.All.getUsers(output);
    }
}
}

```

En la carpeta Hubs, creamos una clase de inicio OWIN y la denominamos Startup



Reemplazamos el código de la clase generada con el siguiente

```

using System;
using System.Threading.Tasks;
using Microsoft.Owin;
using Owin;

[assembly: OwinStartup(typeof(SignalRChat.Hubs.Startup))]

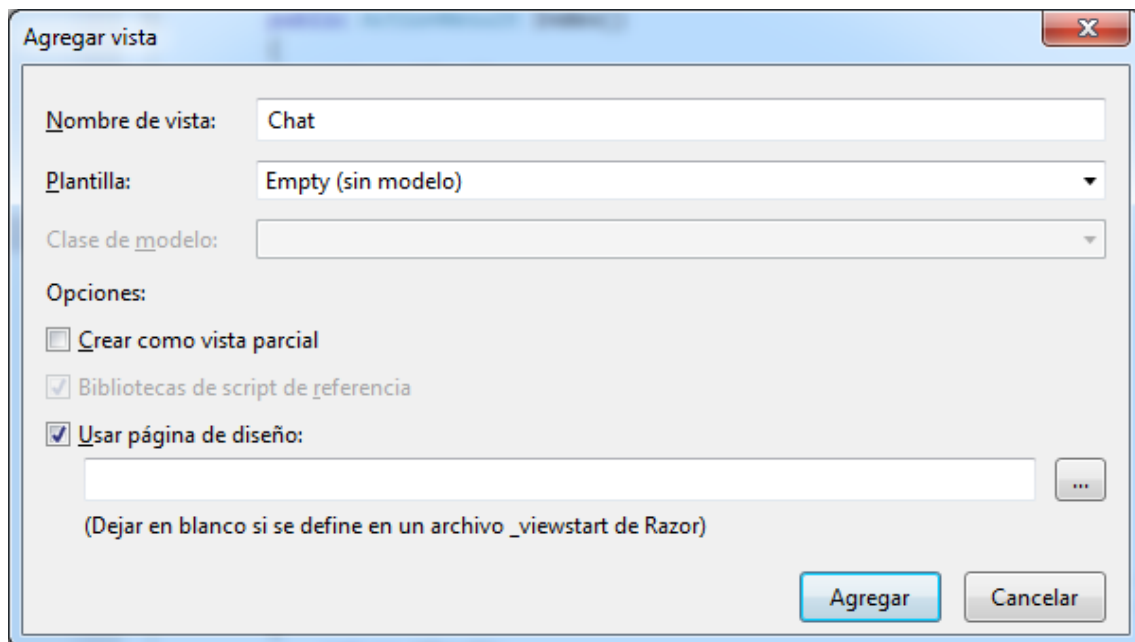
namespace SignalRChat.Hubs
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            // Para obtener más información acerca de cómo configurar su
            // aplicación, visite http://go.microsoft.com/fwlink/?LinkID=316888
            app.MapSignalR();
        }
    }
}

```

Editamos la clase HomeController ubicada en la carpeta Controllers/HomeController.cs y agregamos lo siguiente

```
public ActionResult Chat()
{
    return View();
}
```

Agregamos la respectiva vista



Reemplazamos el código de la vista con lo siguiente

```
@{
    ViewBag.Title = "Chat";
    Layout = "~/Views/Shared/_Nav.cshtml";
}
<h2>Chat</h2>
<link rel="stylesheet" type="text/css" href="~/Scripts/Chat.css">
<div class="row">
</div>
<div id="list"></div>
<section>
    <div class="container_user">
        <!--Datos del usuario-->
        <input id="displayname" />
        <br /><br />
        <input type="hidden" id="displaypicture" />
        <br /><br />
        
        <br /><br />
        <input type="button" id="confirm" value="Confirm" />
        <!--/-->
    </div>
    <br />
    <div class="container_chat" hidden>
        <input type="text" id="message" />
        <input type="button" id="sendmessage" value="Send" />
    </div>
</section>
```



```

        <ul id="discussion"></ul>
    </div>
</section>
<!--http://codepen.io/retyui/pen/zxGqPJ?editors=0100-->
<!--https://www.simple-talk.com/dotnet/asp.net/tracking-online-users-with-signalr/-->
@section scripts {
    <!--Script references. -->
    <!--The jQuery library is required and is referenced by default in
    _Layout.cshtml. -->
    <!--Reference the SignalR library. -->
    <script src="~/Scripts/jquery.signalR-2.2.0.min.js"></script>
    <!--Reference the autogenerated SignalR hub script. -->
    <script src="~/signalr/hubs"></script>
    <!--SignalR script to update the chat page and send messages.-->
    <script src="~/Scripts/ChatHub.js"></script>
}

```

Debemos asegurarnos de coincida el nombre del Script importados, dado que estos se actualizan cada cierto tiempo.

Agregamos un archivo Script denominado ChatHub en la carpeta Scripts y ponemos lo siguiente

```

$(function () {

    // Reference the auto-generated proxy for the hub.
    var chat = $.connection.chatHub;

    // Create a function that the hub can call back to display messages.
    chat.client.addNewMessageToPage = function (name,picture, message) {
        // Add the message to the page.
        $('#discussion').append('<li class="you">'
            + '<a class="user" href="#"></a>'
            + '<div class="date">' + htmlEncode(name) + '</div>'
            + '<div class="message">'
            + '<p>' + htmlEncode(message) + '</p></div></li>');
    };

    // Get the user name and store it to prepend to messages.
    var name = prompt('Enter your name:', '');
    if (name != null && name!='') {
        $('#displayname').val(name);
    }
    else
    {
        name = "Default";
        $('#displayname').val(name);
    }

    //Obtiene el avatar del usuario
    var picture = prompt('Enter the url of an image:', '');
    if (picture != null && picture != '') {
        $('#displaypicture').val(picture);
    }
}

```

```

        $('#preview').attr('src', picture) ;
    }
    else {
        picture =
"https://media.licdn.com/mpr/mpr/shrinknp_200_200/p/4/005/06e/3b0/05b5373.jpg"
        $('#displaypicture').val(picture);
        $('#preview').attr('src', picture);
    }

    // Set initial focus to message input box.
    $('#message').focus();

    // Start the connection.
    $.connection.hub.start().done(function () {

        $('#sendmessage').click(function () {
            // Call the Send method on the hub.
            chat.server.send($('#displayname').val(),
$('#displaypicture').val(), $('#message').val());
            // Clear text box and reset focus for next comment.
            $('#message').val('').focus();
        });

        $('#confirm').click(function () {
            $('#.container_chat').show();
            $('#.container_user').hide();
            //Update nav
            $('#final_displayname').text(name);
            $('#final_displaypicture').attr('src', picture);

            // Call the New_User method on the hub.
            chat.server.update(name,picture);
        });
    });

    //Invoca la expresion dinamica que indica la cantidad de usuarios
conectados
    chat.client.online = function (count) {
        $('#counter').text(count);
    }

    //Invoca la expresion dinamica que indica el listado de usuarios
    chat.client.getUsers = function (list) {
        $("#list_of_users").empty();
        var myObject = eval('(' + list + ')');
        for (i in myObject) {
            //Itero sobre solo el atributo name de la clase recibida en JSON
            $("#list_of_users").append("<li>" +
                "<img src='" + myObject[i]["url"] + "' alt='Avatar' height='30'
width='30'> " +
                myObject[i]["name"] +
                "</li>");
        }
    }
});

    // This optional function html-encodes messages for display in the page.
    function htmlEncode(value) {
        var encodedValue = $('<div />').text(value).html();
    }

```

```
    return encodedValue;
}
```

Otros estilos y Scripts usados se encuentran en la demo y son los siguientes:

- ✓ ~/Scripts/Chat.css
- ✓ ~/Scripts/External/...
- ✓ ~/Content/External/...

En la vista Views/Shared/\_Layout.cshtml agregamos el enlace a la vista de chat

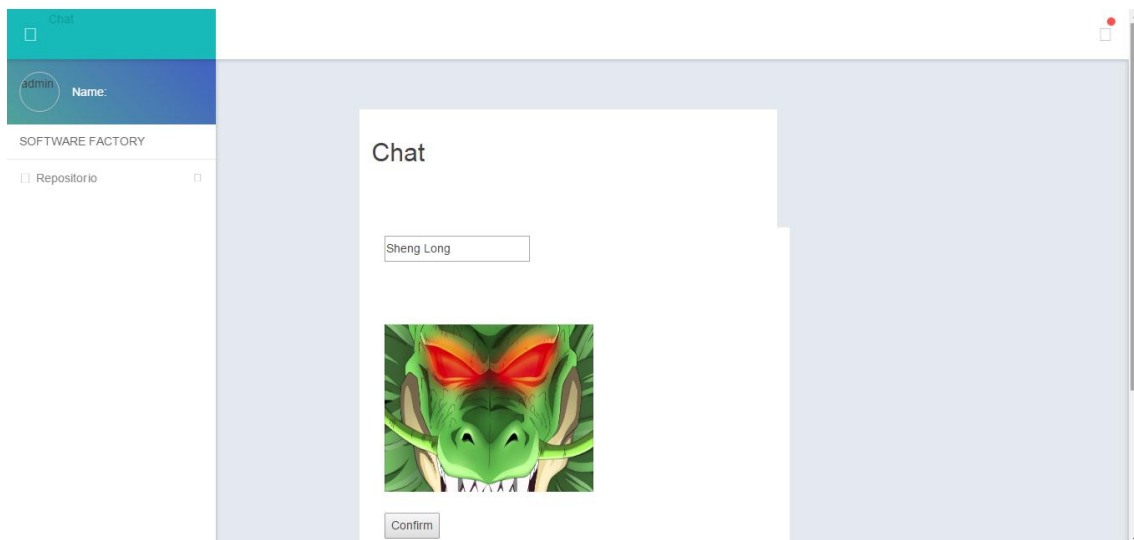
```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li>@Html.ActionLink("Inicio", "Index", "Home")</li>
    <li>@Html.ActionLink("Acerca de", "About", "Home")</li>
    <li>@Html.ActionLink("Contacto", "Contact", "Home")</li>
    <li>@Html.ActionLink("Chat", "Chat", "Home")</li>
  </ul>
  @Html.Partial("_LoginPartial")
</div>
```

### Examinando el código

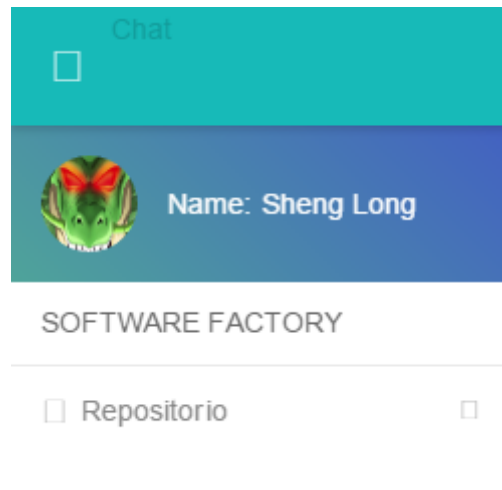
Se dará una mejor explicación del código durante la clase y se subirá una PPT a Github. Esto con el propósito de facilitar el entendimiento del código.

### Usando la Demo

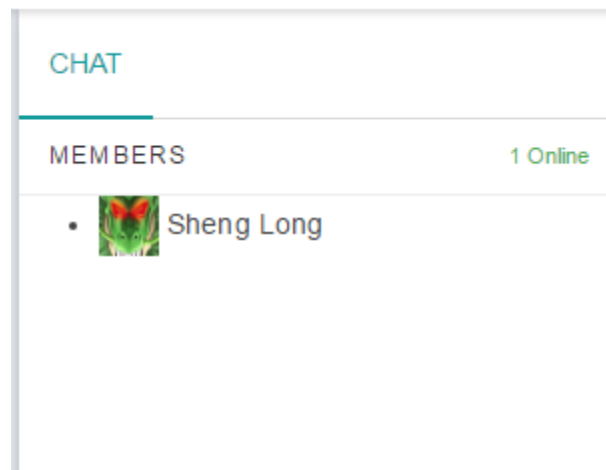
Al ingresar los datos requeridos debemos confirmar estos para proseguir.



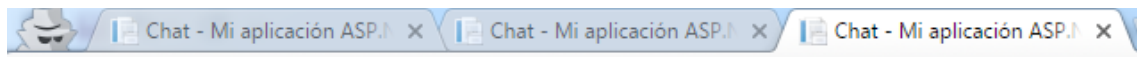
De ser así, los datos se muestran en la sección de la izquierda.



Además se actualizan la cantidad de usuarios conectados, cuyos datos se encuentran a la derecha



Una conversación grupal sería de la siguiente manera





Goku

Sheng Long tenemos que revivir a alguien...



Sheng Long

No me digas que se trata de Krilin... "he had one fucking job"






Krilin


Insensibles :v


## CHAT

### MEMBERS



3 Online

-  Sheng Long
-  Goku
-  Krilin

 Chat

 Name: Krilin

SOFTWARE FACTORY

 Repositorio 

Send



Goku

Sheng Long tenemos que revivir a alguien...



Sheng Long

No me digas que se trata de Krilin... "he had one fucking job"



Krilin

Insensibles :v




© 2015 westilian

Crafted with by westilian

## CHAT

### MEMBERS

3 Online

-  Sheng Long
-  Goku
-  Krilin

## Referencias

- ✓ Window prompt() Method: Usado para capturar el nombre del usuario
  - URL: [http://www.w3schools.com/jsref/met\\_win\\_prompt.asp](http://www.w3schools.com/jsref/met_win_prompt.asp)
- ✓ Tutorial: Getting Started with SignalR 2 and MVC 5: Guía para desarrollar la demo
  - URL: <http://www.asp.net/signalr/overview/getting-started/tutorial-getting-started-with-signalr-and-mvc>
- ✓ CSS para chat
  - URL: <http://www.bypeople.com/css-chat/>
  - URL del estilo usado: <http://www.bypeople.com/css-chat-bubble-blur/>
- ✓ Tracking Online Users with SignalR
  - <https://www.simple-talk.com/dotnet/asp.net/tracking-online-users-with-signalr/>
- ✓ Convert JSON string to array of JSON objects in Javascript
  - <http://stackoverflow.com/questions/4375537/convert-json-string-to-array-of-json-objects-in-javascript>