## ORT Chat

Preparado para:

Instituto de Tecnología ORT Argentina – Analista de Sistemas – Taller de Computación IV

Preparado por:

Brian Teper - Alumno

Junio de 2013

# Contenido

[1 – Descripción de la aplicación 3](#_Toc359508729)

[2 – Requisitos del sistema 3](#_Toc359508730)

[3 – Entorno de desarrollo 3](#_Toc359508731)

[4 – Tecnologías involucradas 4](#_Toc359508732)

[5 – Configuración 6](#_Toc359508733)

[6 – Contacto 9](#_Toc359508734)

# 1 – Descripción de la aplicación

Se desarrolló una aplicación para la materia Taller de Computación IV, del segundo año (cuarto cuatrimestre) de la carrera Analista de Sistemas. El objetivo de la aplicación es permitir la interacción en línea y en tiempo real de los distintos usuarios que se conecten a la misma. Los usuarios podrán acceder al chat mediante un sitio web, que funciona como cliente y como servidor al mismo tiempo, y mediante dispositivos móviles que soporten Windows Phone 8.0.

En esta primera versión, los usuarios que ingresan, se comunican con los demás presentes en una única sala de chat. Esto quiere decir que todos los usuarios que se conectan, ingresan a la misma sala o grupo e intercambian mensajes entre ellos. A futuro, se espera poder mejorar esta versión y agregarle nuevas funcionalidades, como por ejemplo: Login con Facebook, distintas salas o grupos de chat (estos podrían ser por curso o por materias), lista de contactos, envío de imágenes, y muchas cosas más que nos ofrecen algunos servicios de chat ya existentes.

# 2 – Requisitos del sistema

* Sistemas operativos compatibles
  + Windows 8 (x64)
* Arquitecturas compatibles
  + 64 bits (x64)
* Requisitos de hardware
  + 15 GB de espacio disponible en disco duro (VS2012 y WP8 SDK)
  + 4 GB de RAM
  + Procesador de 1,6 GHz o superior
* Para utilizar Windows Phone 8 Emulator
  + Edición Windows 8 Pro o superior
  + Procesador que admita la traducción de direcciones de segundo nivel (SLAT) 🡪 Core i3, i5 o i7.
  + Hyper-V habilitado (para indicaciones sobre cómo habilitarlo, ingrese al siguiente link <http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj863509(v=vs.105).aspx>)

# 3 – Entorno de desarrollo

* Visual Studio 2012 (<http://www.microsoft.com/visualstudio/eng/downloads>)
* Framework .NET 4.5 (<http://www.microsoft.com/en-us/download/details.aspx?id=30653>)
* Windows Phone 8 SDK (<http://www.microsoft.com/en-us/download/details.aspx?id=35471>)

# 4 – Tecnologías involucradas

* Aplicación Web (ORTChatMVC)
  + ASP .NET MVC 4 (<http://www.asp.net/mvc/mvc4>)

Es un framework para la creación de aplicaciones web escalables y basadas en estándares utilizando patrones de diseño bien establecidos y el poder de ASP .NET y el Framework .NET.

* + ASP .NET SignalR (<http://signalr.net/>) (NuGet Package\*)

Es una nueva librería para desarrolladores ASP .NET que hace que sea increíblemente fácil de añadir funcionalidad de la web en tiempo real a las aplicaciones. ¿Qué quiere decir “web en tiempo real”? Es la capacidad de tener código ‘server-side’ (del lado del servidor) que es enviado a todos los clientes conectados a medida que ocurre, es decir, en tiempo real.

SignalR hace uso de la nueva API (Interfaz de Programación de Aplicaciones, en español) de HTML5, los WebSockets, que permiten la comunicación bidireccional entre el navegador y el servidor. SignalR utiliza los WebSockets si están disponibles, y en caso de no estarlo, se basa en otras tecnologías y técnicas, mientras que nuestro código se mantiene de la misma manera.

SignalR también proporciona una API muy simple y de alto nivel para hacer RPC del servidor al cliente (llamadas a funciones JavaScript en los navegadores de sus clientes desde el lado del servidor .NET), así como también para la gestión de conexiones.

Con todo lo mencionado, se puede observar lo simple que nos resulta que cuando un usuario cliente envía un nuevo mensaje a la sala de conversación, desde nuestro servidor .NET llamemos a una función JavaScript en cada uno de nuestros clientes (o los seleccionados) para que los mismos agreguen el nuevo mensaje a la conversación.

* + HTML5 (<http://www.w3schools.com/html/html5_intro.asp>)

Es la quinta revisión importante del lenguaje básico de la World Wide Web. El desarrollo del lenguaje de marcado es regulado por el Consorcio W3C. Esta versión ofrece numerosas ventajas y avances con respecto a las anteriores, que únicamente los browsers más actualizados pueden disfrutar.

* + JavaScript

Es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma ‘client-side’ (del lado del cliente), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, entre muchas otras cosas. Además existe una forma de JavaScript ‘server-side’ (del lado del servidor).

* + jQuery (<http://jquery.com/>)

jQuery es una biblioteca JavaScript rápida, pequeña y rica en funciones. Se encarga de tareas como manipulación y recorrido de HTML, control de eventos, animación y permite la utilización de Ajax mucho más simple con una API fácil de usar que funciona a través de una gran variedad de browsers (navegadores). Con una combinación de flexibilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript.

Esta biblioteca ya viene incluida en el template (plantilla) de proyecto MVC 4.

* + JSON.NET (<http://james.newtonking.com/projects/json-net.aspx>) (NuGet Package\*)

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero. Es fácil de comprender y leer para los seres humanos y es fácil de ‘parsear’ (analizar) y generar para las computadoras. JSON es un formato de texto que es completamente independiente del lenguaje que se utilice. Estas propiedades hacen que sea el lenguaje ideal de intercambio de datos.

JSON.NET es un framework JSON, hecho en .NET y de alto rendimiento. Es considerado el mejor serializador JSON de .NET.

* Aplicación Windows Phone 8 (ORTChatWP8)
  + XAML

Es un lenguaje de marcado basado en XML y desarrollado por Microsoft. Es el lenguaje detrás de la presentación visual de una aplicación que se desarrolla en Microsoft Expression Blend, como así HTML es el lenguaje detrás de la presentación visual de una página web.

* + SignalR Client (NuGet Package\*)

Este ítem ya ha sido descripto en la sección anterior “Aplicación Web (ORTChatMVC)”. Vale destacar que en esta ocasión se utilizó únicamente la librería que permite conectarse como un cliente en un entorno SignalR. Esta librería es “Microsoft.AspNet.SignalR.Client”.

* + JSON.NET (<http://james.newtonking.com/projects/json-net.aspx>) (NuGet Package\*)

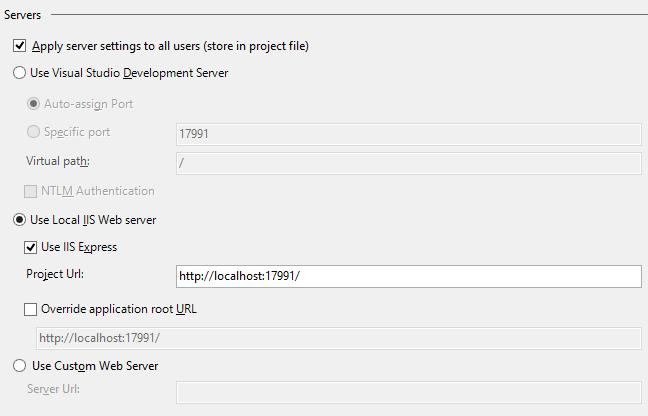
Este ítem ya ha sido descripto en la sección anterior “Aplicación Web (ORTChatMVC)”.

\* NuGet: es una extensión de Visual Studio que facilita la instalación de librerías (.dll) de terceras partes en Visual Studio (<http://nuget.org/>).

# 5 – Configuración

* *Aplicación Web (ORTChatMVC)*

- Para testear la aplicación en un ambiente de desarrollo, hacemos click derecho en el proyecto ORTChatMVC (dentro de Visual Studio) y nos dirigimos a las propiedades del mismo. Allí mismo, seleccionamos la opción Web en el menú lateral izquierdo y, una vez dentro, seleccionamos la opción “Use Local IIS Web server” y tildamos la opción “Use IIS Express”.

  
Si iniciamos la aplicación, veremos que se inicializa el IIS Express y nuestro sitio web arrancará en nuestro navegador seleccionado.

- Para poder acceder a nuestro servidor de chat web desde otras máquinas conectadas a la misma red wi-fi, debemos modificar el archivo de configuración llamado “applicationhost.config” que se encuentra ubicado en la siguiente ruta “C:\Users\NOMBRE\_DE\_USUARIO\Documents\IISExpress\config”. Lo abrimos con cualquier editor de texto y buscamos la entrada que pertenece a nuestro sitio:

<site name="ORTChatMVC" id="ID\_GENERADO">

Dentro de este nodo, encontraremos el nodo de Bindings:

<bindings>

<binding protocol="http" bindingInformation="\*:17991:localhost" />

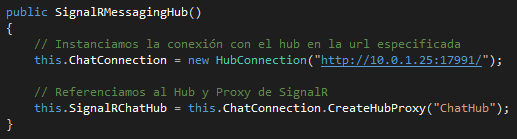
</bindings>

Es aquí donde debemos modificar el bindingInformation "\*:17991:localhost" por "\*:17991:\*", permitiendo de esta manera, acceder al servidor por su IP.

Por ejemplo, si la IP de nuestro servidor es 10.0.1.25, accederemos al sitio con la siguiente URL: <http://10.0.1.25:17991/Home/Chat>

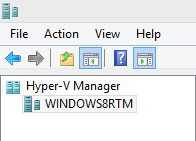
* *Aplicación Windows Phone 8 (ORTChatWP8)*

- El primer paso y más importante es modificar la IP y el puerto a la que apunta el cliente de chat en la aplicación móvil. Para realizar esto, una vez obtenida la IP o url del servidor, nos dirigimos al Visual Studio, y en el proyecto ORTChatWP8, dentro de la carpeta SignalR, modificamos la dirección IP con el puerto que aparece de ejemplo:

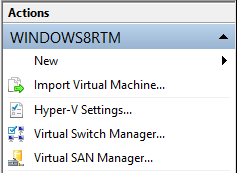


- Si vamos a testear la aplicación con un dispositivo móvil físico, sólo debemos asegurarnos que el mismo esté conectado dentro de la misma red wi-fi para poder acceder al servidor de chat.

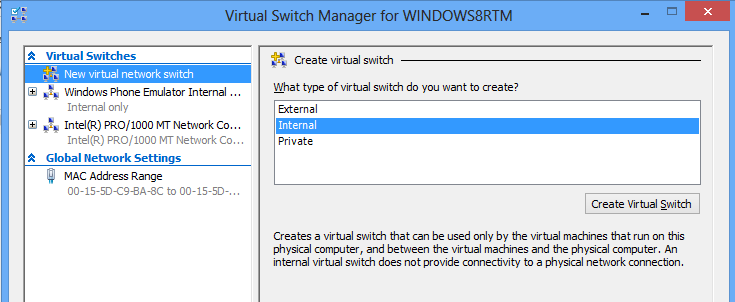
- Si vamos a testear la aplicación desde el emulador de Windows Phone, antes de inicializar el mismo, debemos dirigirnos a la aplicación Hyper-V Manager (podremos encontrar la misma si la buscamos en la pantalla de Home de Windows 8). Allí dentro, debemos seleccionar el nombre de nuestra PC en el menú lateral izquierdo:



Luego, en el menú lateral derecho seleccionamos la opción Virtual Switch Manager.



En esta pantalla, creamos una nueva Virtual Network Switch de tipo Internal, y luego, modificamos su nombre a “Windows Phone Emulator Internal Switch”.



# 6 – Contacto

Por cualquier duda o aclaración, por favor contactarse con:

Alejandro Jait – Coordinador Carrera Analista de Sistemas: ajait@ort.edu.ar

Brian Teper – Alumno de la Carrera Analista de Sistemas: brian.teper@outlook.com