Proyecto Interpool

Documentación Técnica

Versión 14.1

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 06/11/10 | 13.0 | Documentación Técnica del proyecto Interpool | Marcos Sander |
| 07/11/10 | 13.1 | Revisión de SQA | Javier Madeiro |
| 14/11/10 | 14.0 | Agregado del deploy y subir base de datos a Azure | Marcos Sander |
| 14/11/10 | 14.1 | Revisión de SQA | Javier Madeiro |
|  |  |  |  |
|  |  |  |  |

Contenido

[1. Arquitectura del sistema 3](#_Toc277508952)

[2. Proyectos 4](#_Toc277508953)

[3. Base de datos 4](#_Toc277508954)

[3.1. Crear y cargar la base local 4](#_Toc277508955)

[3.2.1 Creación de la base y configuración de la misma en Visual Studio 4](#_Toc277508956)

[3.2.2 Generar el modelo de la base de datos 4](#_Toc277508957)

[3.2.3 Cargar la Base de Datos 5](#_Toc277508958)

[Ejecutar el script All.sql y esperar a que se carguen los datos en la base. Verificar que no hubo error en la carga de los datos. 5](#_Toc277508959)

[3.2. Crear y cargar la base en Azure 5](#_Toc277508960)

[3.2.4 Creación de la base 5](#_Toc277508961)

[3.2.5 Generar el Transact-SQL Script 5](#_Toc277508962)

[3.2.6 Cargar la base de datos 5](#_Toc277508963)

[4. Instalación del sistema 6](#_Toc277508964)

[4.1. Levantando el juego en alguna pc o notebook 6](#_Toc277508965)

[4.2. Instalar el juego en un WP7 6](#_Toc277508966)

[4.3. Levantando el servidor InterpoolCloud en forma local 6](#_Toc277508967)

[4.4. Levantando el servidor InterpoolCloud en la nube 6](#_Toc277508968)

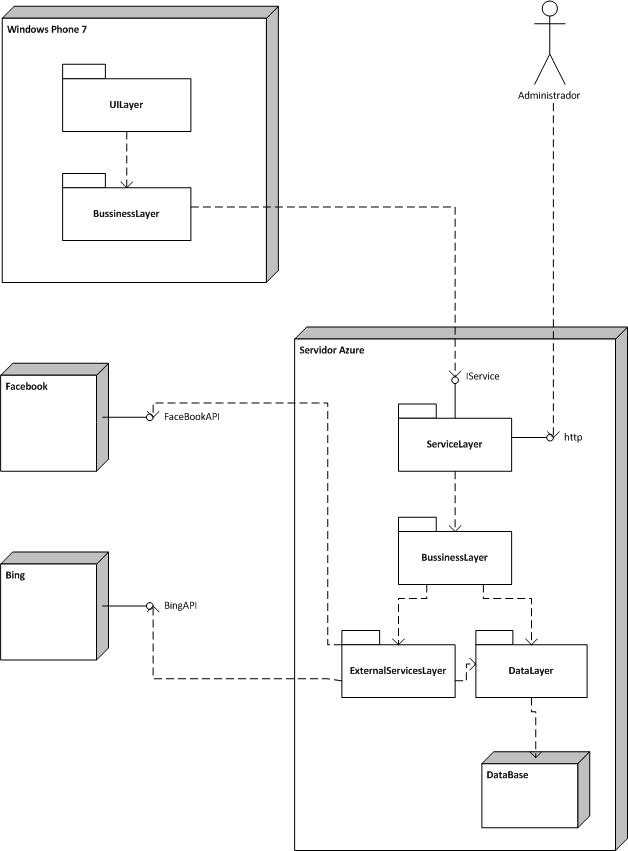
1. Arquitectura del sistema

Para el desarrollo del sistema Interpool, elegimos una arquitectura Cliente-Servidor, en que los clientes, con el celular con sistema operativo Windows Phone 7, consumen servicios del servidor que está en la nube, implementado sobre la tecnología Azure.

El sistema en el nodo Windows Phone 7, fue desarrollado en dos capas, la capa de Presentación, desarrollada con Silverlight y la capa Lógica que recibe las acciones a ser tomadas en el juego y se comunica con el servidor, para llevar a cabo tales acciones.

En el nodo Servidor Azure, se implementó una arquitectura en 4 capas bien definidas. La capa de Servicios, que maneja la comunicación con Windows Phone, la capa de Negocios, que implementa la lógica de cada servicio, la capa de Servicios Externos, que implementa la comunicación con Facebook y Bing y la capa de Datos, que sirve para la comunicación con la base de datos.

Una decisión importante de la arquitectura y del diseño de juego, es qué, todos los servicios del juego y operaciones crean, modifican o eliminan datos de la base de datos. Esto nos permite que el estado del juego este determinado según lo que está en la base de datos y si por ejemplo, se cierra el juego o cualquier problema de comunicación que haya con Azure, el juego sigue en el mismo estado. Esto incluso permite mantener consistencia si un usuario utiliza múltiples teléfonos asociados a la misma cuenta de Facebook.



1. Proyectos

El desarrollo está divido básicamente en 2 proyectos, un proyecto que implementa la funcionalidad del juego en Windows Phone 7, (WP7.sln) y el otro que implementa la funcionalidad del servidor en Azure (InterpoolCloud.sln).

En un ambiente de prueba, se puede levantar el proyecto del juego que corre en Windows Phone, con Visual Studio, teniendo instalado el emulador de Windows Phone 7 y también se puede levantar el proyecto que corre en el servidor Azure, de forma local con otra instancia de Visual Studio, también teniendo instalado el emulador de Azure Fabric.

En funcionamiento del software, se hace el deploy del proyecto InterpoolCloud.sln para que se ejecute en el servidor de Azure. Por otra parte se carga el software del Windows Phone en los teléfonos que funcionan con el sistema operativo Windows Phone 7.

1. Base de datos

La base de datos es una base de datos relacional sobre SQL Server Express 2008.

* **En forma local**, podemos levantar la base de datos.
* **En Azure,** usamos cómo administrador de base de datos, el proyecto Houston, en modo Web provisto gratuitamente (este manejador de base de datos de Azure se encuentra en: <https://www.sqlazurelabs.com/houston.aspx>). Para acceder a la base de datos desde ahí, se proporciona la dirección IP del servidor, el nombre de la base de datos, nombre de usuario de Azure y contraseña.

Para cargar los datos a la base de datos en Azure, ejecuto lo que está en el AllNube.sql, utilizando el administrador Houston.

* 1. Crear y cargar la base local
     1. Creación de la base y configuración de la misma en Visual Studio

1. Con algún administrador de base de datos, el SQLServer Management Studio por ejemplo, accedemos al servidor de la base de datos y creamos la base, que la llamamos InterpoolDB (sin importar en principio la estructura de la misma, ya que las tablas y relaciones de la base serán creadas más adelante).
2. En Visual Studio, teniendo el proyecto InterpoolCloud abierto, configuramos la conexión con la base de datos anteriormente creada. Para esto vamos a la pestaña Server Explorer, cliqueamos sobre el ícono para crear nueva conexión con la base de datos, y elegimos el servidor de base de datos que vamos a usar (el mismo sobre el cual creamos la base de datos), la base de datos que habíamos creado y la autenticación para acceder a la misma (puede ser autenticación Windows o usuario creado para SQLServer).
   * 1. Generar el modelo de la base de datos
3. En la pestaña Solution Explorer, voy al archivo donde está el modelo de las tablas (InterpoolModel.edmx), y las relaciones.
4. Botón derecho del mouse y click en Generate DB from model (darle aceptar a todo).
5. Abrir el script SQL que se genera a partir de lo que hicieron en el paso 2. (es un archivo InterpoolModel.edmx.sql).
6. Luego botón derecho sobre el archivo (InterpoolModel.edmx.sql) y luego clickeo en la opción "Execute SQL".
   * 1. Cargar la Base de Datos
7. Abrir el SQL Management Studio y luego abrir el archivo q se llama ALL.sql que está con los archivos del proyecto en la carpeta /Recursos/DataBase/All.sql

Ejecutar el script All.sql y esperar a que se carguen los datos en la base. Verificar que no hubo error en la carga de los datos.

* 1. Crear y cargar la base en Azure
     1. Creación de la base

1. Ir a <https://sql.azure.com/> e ingresar nombre de usuario y contraseña para acceder a la cuenta de Azure.
2. Elegir la opción Create Database y elegir un nombre para la base de datos a crear.
   * 1. Generar el Transact-SQL Script

En este paso, vamos a crear un Transact-SQL Script, que puede compararse con el que creamos en el paso 3.1.2 para la base local, pero este tiene sólo comandos que son aceptados en Azure.

1. En Visual Studio, en Object Explorer, hacer botón derecho en la base de datos InterpoolDB, seleccionar Tasks y seleccionar Generate Scripts.
2. En el diálogo abierto, clickear Next, seleccionar Script entire database and all database objets, y clickear Next.
3. En **Set Scripting Options**, seleccionar las siguientes opciones:
4. Elegir **Output Type** como **Save script to a specific location**. Seleccionar **Save to file**. Clickear **Single file**. Elegir el nombre del archivo y donde se va a guardar en **File name**. Clickear en **Advanced**.
5. En **Advanced Scripting Options** elegir la opción “Script for the database engine type” cómo “SQL Azure Database”, seleccionar la opción “Convert UDDTs to Base Types” cómo “True” y la opción “Types of data to script” cómo “Schema and data”. Clickear **OK**.
6. Clickear **Next**, clickear **Next**, y al final clickear **Finish**.
   * 1. Cargar la base de datos

Usar el proyecto Houston (<https://www.sqlazurelabs.com/houston.aspx>) ó el SQL Server Management Studio para acceder a la base de datos y ejecutar el script generado en el paso anterior.

Luego abrir el archivo de nombre ALLNube.sql, que se encuentra con los archivos del proyecto en la carpeta /Recursos/DataBase/AllNube.sql

Ejecutar el script AllNube.sql y esperar a que se carguen los datos en la base. Verificar que no hubo error en la carga de los datos.

1. Instalación del sistema

Para Windows Phone 7, se instala un ejecutable que estará disponible en el Market Place de Windows.

Está la opción de prueba en desarrollo, que es levantar un proyecto Visual Studio con el emulador de Windows Phone 7 instalado, utilizando algún pc, notebook, etc.

* 1. Levantando el juego en alguna pc o notebook

Entrar a la carpeta WP7 en el trunk y abrir la solución que hay en esa carpeta.

Luego de abrir la solución que estaba en WP7 hacer CTRL + F5 para ejecutar el proyecto.

NOTA: Es necesario tener el InterpoolCloud abierto y corriendo para que pueda brindar los servicios requeridos por el juego corriendo en el WP7.

* 1. Instalar el juego en un WP7

Acceder al Market Place de Windows y seleccionar el juego para bajarlo. Ejecutar el software descargado.

Al igual para Azure, hacemos el deploy del proyecto a la nube, y ahí tenemos los servicios que van a consumir celulares con el sistema Windows Phone 7, que ejecuten el juego.

Tenemos la opción de ejecutarlo de forma local, utilizando el emulador de Azure Fabric.

* 1. Levantando el servidor InterpoolCloud en forma local

Abrir el proyecto InterpoolCloud y luego CTRL + F5.

NOTA: Es necesario que la aplicación corra siempre en el puerto 81. En caso de ya haberlo abierto es necesario bajar el servicio Azure yendo a la barra de tareas, apretamos botón derecho sobre el ícono y luego "exit" para bajar el servicio. Luego de haber bajado el servicio, podemos realizar nuevamente el paso 1).

* 1. Levantando el servidor InterpoolCloud en la nube

1. Estando en el Visual Studio, con en el proyecto InterpoolCloud abierto:

* Editar el web.config para que apunte a la base de la nube
* Comentar todas las constantes locales y dejar las de la nube. Las constantes son: RedirectUrlAfterLoginFacebook, ConsumerKey, ConsumerSecret, FacebookCallbackUrl.
* Rebuild a toda la solución.
* Publish al Proyecto InterpoolCloud. Se crean 2 archivos que los vamos a subir a la nube.

1. Ingresar a la cuenta de Azure que tenemos. Para esto vamos a (<https://windows.azure.com/Cloud/>). Ingresamos el usuario y contraseña de nuestra cuenta en Azure.
2. Seleccionar Upgrade del servicio.
3. Cargar los dos archivos que generó el Visual Studio en el paso 1 y poner una etiqueta a la "subida"
4. Darle OK y esperar a que el servicio quede subido y corriendo en la nube.