# Definición del Problema

Se propone realizar un sistema de mensajería que permitirá al usuario acceder a un servidor remoto que gestionará las distintas conversaciones entre los clientes a través de la creación de salones virtuales tomando en cuenta los siguientes aspectos:

- Se deberán crear dos aplicaciones: una primera que actúe como la vista para el usuario y una segunda que aceptará conexiones a través de la red desde múltiples terminales.

- Debe considerar la elaboración de Ventanas de Windows tipo winforms como vista para la aplicación. Esta vista permitiría mostrar una lista de los usuarios actualmente en línea y así poder enviar uno o varios archivos a través de la red utilizando conexiones múltiples a través de hilos.

- La aplicación de servidor deberá brindar cualquier medio de almacenamiento persistente (archivos de datos o registro a través de base de datos) para registrar los distintos usuarios que se pueden conectar así como sus respectivas contraseñas.

- Se deberá crear un archivo de texto de configuración para ambas aplicaciones (vista y servidor) donde se almacenarán los valores de variables globales para que estas puedan ser especificadas por el usuario (como ruta del almacenamiento persistente, datos sobre conexiones de redes, número máximo de clientes simultáneos, entre otros).

Notas:

- ¿Podría usted incluir encriptación de datos a través de la red?

- Deberá permitirle al usuario tomarse una foto como imagen de perfil/avatar utilizando una cámara conectada a la PC, y guardarla en la base de datos aplicándole filtros de compresión para reducir su tamaño de almacenamiento.

Tabla de contenido

Definición del Problema 1

Análisis 1

Diseño 2

Código Fuente 7

Condiciones de Error 8

Ejemplos de Uso 9

Notas al Profesor 10

# Análisis

Desde un inicio el problema es bastante interesante y completo, como los anteriores. A diferencia de que en esta ocasión tenemos que desarrollar dos aplicaciones diferentes para el mismo fin. El primer reto que se nos presenta es pasar información no entre clases o entre soluciones, sino entre aplicaciones. Pero eso es sólo la punta del icebrg, ¿qué significa en realidad hacer un cliente de mensajería? Implica varias cosas:

* La creación de dos aplicaciones - ¿en que plataformas vamos a hacer estas dos aplicaciones? ¿Qué nos conviene en realidad? ¿Cómo diseñar una aplicación que sea sencilla de utilizar por el usuario, pero que al mismo tiempo cumpla con los requerimientos planteados?
* La utilización de hilos, ¿qué es lo que vamos a coser? Nada, sino que vamos a utilizar hilos – nuestra aplicación deberá tener la capacidad de hacer varias tareas al mismo tiempo de forma independiente cada una. Nunca hemos utilzado hilos – ninguno de los dos - ¿Será muy complicado? ¿Cómo vamos a manejar nuestra aplicación con hilos?
* Almacenamiento persistente es un requisito de este proyecto, pero con la experiencia adquirida a lo largo de este semestre no representará realmente un problema. Pero debemos un incluirlo en el análisis porque necesitamos plantearnos cual será la mejor solución.
* Debemos asegurarnos de crear un archivo de texto de configuración para las dos aplicaciones – tanto la vista del cliente, como la vista del servidor – para que variables globales puedan ser especificadas por los usuarios de esta aplicación.

El planteamiento del problema presenta dos requerimientos que podemos considerar opcionales debido a la complejidad de nuestro problema original y a que en realidad estos dos puntos no son implescindibles para el correcto funcionamiento de nuestras aplicaciones:

* Incluir encriptación de datos a través de la red. – No tenemos la menor idea de cómo vamos a encriptar los datos que transmitamos - aún tenemos que descubrir como vamos a transmitir.
* Permitirle al usuario tomarse una foto como imagen de perfil/avatar para guardarla en la base de datos – comprimiendola -. Sabemos como capturar con la cámara, lo que no sabemos es si nos va a dar tiempo implementar este conocimiento en nuestro proyecto, ya que tenemos una buena idea de lo que queremos hacer.

# Diseño

Esta es la recta final, nuestro último proyecto – si Moronta no nos quema – de Programación 1. Es hora de poner en pråctica todo lo aprendido y o que aún no para lograr un puen proyecto ya que este tiene más peso que los tres portafolios anteriores.

Así que tomando eso en cuenta decidimos ir más allá y más o menos “superarnos a nosotros mismos” : Creamos nuestras primera aplicación multi-plataforma. ¿Qué quiere decir esto? Que nuestra aplicación funciona de igual forma en Windows, Linux y MacOS. De hecho el desarrollo de esta aplicación fue hecha desde dos sistemas operativos distintos pero ambas en lenguaje C#, con un poco de GTK y otro poco de Android. ¿Qué otras cositas utilizamos?

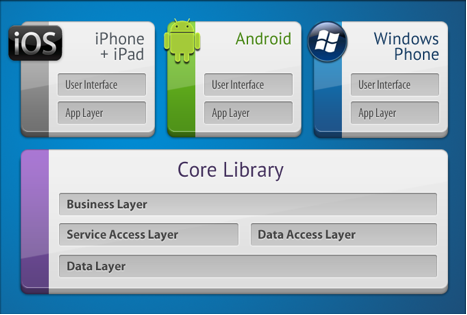
* La creación de dos aplicaciones – Ya probamos y aprendimos WPF, vimos que WindowsForms ya forma parte del pasado, en nuestro último proyecto dimos nuestros primeros pasos en Metro. ¿Y ahora qué? Pues vamos a mudarnos totalmente al entorno móvil e iniciamos nuestro viaje con Android. Así que nuestro servidor va a estar en la computadora (GTK - Mono, C#, Multiplataforma) y nuestro cliente será Android – que fácilmente podemos portar a iOS e incluso a Windows Phone gracias a Xamarin, más del 80% del código escrito en C#. Con esto cumplimos con el primer requerimiento.
* Trabajar con temas desconocidos es el pan diario de Programación1, e hilos no es la excepción – el concepto es conocido, pero en realidad no hemos trabajado con ellos en lo absoluto. Luego de leer un poco nos dimos cuenta de que realmente no es más complejo que decir: “voy a crear un hilo nuevo y ahí dentro quiero que ejcutes tal o cual método”. Y literalmente nos olvidamos de ese hilo que terminará sólo. Esto es imprescindible para nuestra aplicación ya que la naturaleza de la misma requiere que se ejecuten varias acciones independientes una de la otra al mismo tiempo.
* El almacenamiento persistente no fue problema – César está considerando seriamente volarse la materia de Base de Datos 1 – Y la ventaja que tenemos en esta ocasión es que como nuestro servidor es quien se encarga de negociar la información y nuestra vista – el cliente – sólo la ve, podemos utilizar SQL – MySQL, SQLServer o Postresql. A César le gusta MySQL así que vamos a utilizar esta. Repetimos, podemos trabajar con cualquiera de las mencionadas anteriormente, hemos aprendido bastante ese semestre.
* Para darle un poco de flexibilidad a nuestra aplicación decidimos que aquellos datos variables – como la dirección IP del servidor y el número máximo de los clientes – sean especificados por el mismo usuario y la aplicación utilice esos datos para un correcto funcionamiento de la misma.

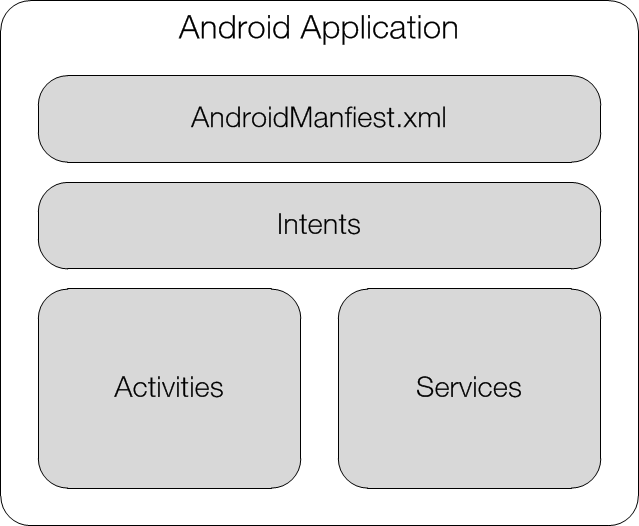
Como es costumbre, veamos algunas tablas y diagramas.











# Código Fuente

Adjunto en el archivo comprimido.

# Condiciones de Error

Cuando se cierra la conexión del servidor a veces no la abre de nuevo correctamente.

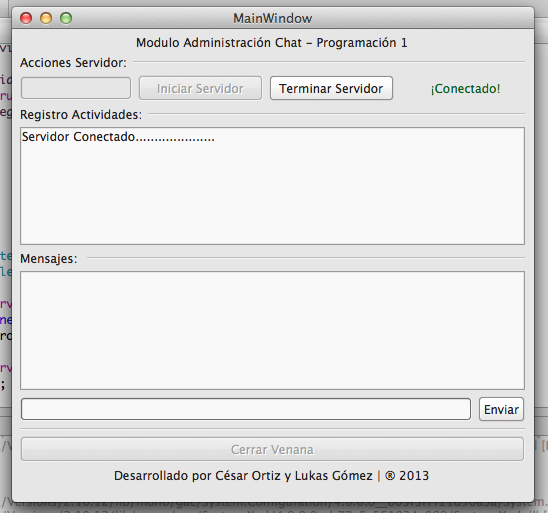
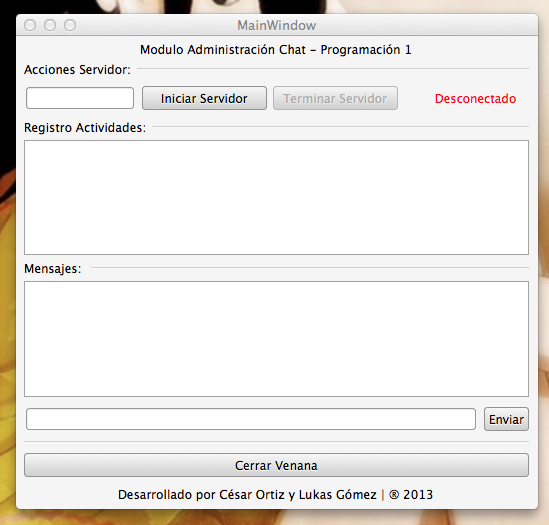
# Ejemplos de Uso

Una vez terminado el programa, se realiza una ejecución con datos de prueba y la salida de la corrida. Se incluyen en la documentación bajo este título.

Puede anexar capturas de pantalla, o puede generar una corrida en texto. Por ejemplo, si el ejecutable se llama “ejemplo.exe”, en la línea de comandos de Windows podemos obtener la salida de la siguiente manera:

**C:\ejemplo.exe > salida.txt**

Esto es lo que llamamos un “redireccionamiento”. Al ejecutarse “ejemplo.exe”, en vez de enviar las salidas a la pantalla, las direccionará al archivo “salida.txt”. El mismo es editable y puede incluirlo en su documentación. Si existe un archivo llamado “salida.txt”, será borrado y se creará uno nuevo.



# D'Negro:Users:lukas:Desktop:Screen Shot 2013-04-23 at 5.15.20 AM.pngD'Negro:Users:lukas:Desktop:Screen Shot 2013-04-23 at 11.13.35 AM.png

# D'Negro:Users:lukas:Desktop:Screen Shot 2013-04-23 at 11.13.50 AM.pngD'Negro:Users:lukas:Desktop:Screen Shot 2013-04-23 at 11.14.01 AM.png

# Notas al Profesor

En caso de que desee hacer un señalamiento al profesor acerca de su trabajo, incluirlo bajo este título al final.