# 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

Como se nota desde la entrada este problema es bastante complejo no sólo por todos los aspectos que abarca sino también por las restricciones e interrelaciones que presenta internamente. Deberemos mantener claros todos los conceptos aprendidos hasta ahora para lograr una buena aplicación. Analicemos nuestro problema:

* En lo primero que nos fijamos es que esta apicación es una aplicación familiar, lo que quiere decir que nuestra aplicación debe ser tan intuitiva que si interfaz pueda ser utilizada por cualquier miembro de la familia sin importar la edad. Hay muchas tecnologías para escoger, ¿cuál nos conviene?
* Otro problema con el que nos encontramos inmediatamente son las múltiples restricciones que nos solicita el problema en base a cada producto. ¿Cómo vamos a verificar cada restricción cuando un usuario quiera acceder a un producto al cual no tiene permiso de consumir?
* El problema solicita que se construya un módulo para subir un mapa de la casa y especificar las distintas áreas de la misma. ¿Existe alguna opción para “mappear” una imagen a voluntad del usuario?
* Ya tenemos experiencia en el área de realizar reportes pero, el primer requisito es que debe ser bastante intuitivo. ¿De qué forma vamos a cumplir ambos requisitos con lo que conocemos ahora? ¿Existirá otro método?
* Un aspecto interesante de la aplicación es que le informará al usuario cuáles productos están próximos a vencer, ¿Cuál parámetro vamos a tomar en cuenta para genenerar la lista de productos proximos a caducar? ¿La fecha de caducidad? ¿La semana de caducidad?
* El módulo de cocina: este módulo es complicado porque hace exactamente lo mismo que el módulo de consumir un producto cualquiera, la diferencia en este caso debemos facilitarle al usuario el acceso especificamente a los alimentos para que pueda consumirlos. ¿Cómo vamos a implementar esto?
* Los productos se pueden gastar, descomponer, robar, etc. Lo que quiere decir que su existencia no sólo se controla por el uso de cada usuario, sino que debe exisitr un módulo que permita al usuario tener un control total sobre la existencia del producto como tal.

Estos son los problemas a los que nos enfrentamos cuando analizamos este proyecto. ¡Trabajemos!

# Diseño

Inmediatamente nos encontramos con la interrogante sobre cual tecnología utilizar para este programa, porque el un requerimiento del mismo que tenga una interfaz agradabe y que pueda ser utilizado por toda la familia. Pusimos nuestros cerebros a trabajar y decidimos que nuestra aplicación debe ser capaz de correr en una tableta - que cualquier miembro de la familia puede utilizar -, y patiendo de ahí entonces buscaremos cuales lenguajes, interfaces y demás son las que mejores se adaptan a nuestras necesidades.

Para el almacenamiento peristente “we got it covered”, hasta que más tarde nos encontramos con la noticia de que nuestra interfaz no soporta SQLServer, MySQL y PosgreSQL.

Como ya es costumbre entre nosotros vamos a diseñar nuestro programa respondiendo a cada problema específico planteado en nuestro análisis:

* Habíamos oido y visto en Internet que existe la posibilidad de diseñar aplicaciones en C#, implementando interfaz Android – así que decidimos probar esto hasta que nos dimos cuenta de que diseñar una interfaz android es un dolor de cabeza y esta sólo soporta SQLite. Así que para “facilitarnos la existencia” nos decidimos por crear una aplicación tipo “Windows Store”, cuyo producto final dependiendo del diseñador, puede ser bastante intuitivo. El problema con esto es que esta interfaz tampoco soporta el SQL tradicional, así que otra vez la solución para el almacenamiento persistente es SQLite. Así que por suerte o destino, tenemos que aprender este nueva forma de interactuar con nuestra base de datos.
* Para las restricciones de los productos, podemos asignarle a cada producto un nivel de accesiblidad, donde el número más bajo indica que puede ser consumido/accedido por cualquier miembro de la familia, mientras que los números más grandes cuales restricciones tiene. Así que comparando el nivel de accesibilidad con el grado de acceso que tenga el usuario seremos capaces de controlar cuales usuarios consumen un producto en específico.
* El interfaz “Windows Store” es bastante limitada porque se ejecuta en dispositivos limitados, así que el Crystal Reports que hemos utilizado a lo largo del semestre no nos es útil. Pero al mismo tiempo, la interfaz nos da la faciidad de recolectar y presentar la infomación de una manera agradable.
* Vamos a tomar como referencia la semana siguiente a la actual y presentar los productos que expiren en esa semana, para así presentar los productos próximos a vencer. Así siempre estará actualizada y si no se ha consumido un producto antes de su fecha de caducidad, se marca como expirado.
* El módulo de cocina será igual al módulo de consumir un producto cualquiera, sólo que en este caso el usuario va a introducir de forma sencilla todos los artículos que va a consumir para preparar esa comida.

Veamos algunas clases, diagramas y gráficos.









# Código Fuente

Adjunto en el archivo comprimido.

# Condiciones de Error

Por lo general, hay motivos que pueden causar que el usuario de un programa obtenga una anomalía o que el algoritmo usado tenga ciertas restricciones de uso. Ejemplo de esto es cuando a un programa que opera con números positivos se le digitan negativos. Ocurre también con funciones que no son capaces de operar con un número muy grande (por ejemplo, el factorial).

Estas anomalías deben enumerarse en esta sección del trabajo. Deberá incluir ejemplos de corrida con errores en esta sección donde se contemple el mismo claramente y se pueda visualizar cualquier mensaje de error desplegado. Tenga cuidado al decir que su programa no tiene condiciones de error. Quedará luego a consideración del profesor valorar las condiciones de error.

# 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.

# Notas al Profesor

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