

# Segundo Proyecto: ConcuConversation

75.59 - Técnicas de Programación Concurrente I

---

## Objetivo

El objetivo de este proyecto es el desarrollo de una aplicación conocida como "ConcuConversation". Esta aplicación permitirá a los usuarios enviar y recibir mensajes de texto al mismo tiempo. Cada participante de una conversación ejecutará un programa que le permitirá intercambiar los mensajes de texto con los demás participantes de esa conversación.

## Componentes del Sistema

El sistema estará conformado por los siguientes módulos:

1. Servidor de conversaciones
2. Clientes

### Servidor de conversaciones

Este módulo se encarga de llevar el registro de las conversaciones que se encuentran activas, así como de los usuarios que forman parte de ellas.

### Clientes

Son los programas que utilizan los usuarios para conversar entre ellos. Cuando un cliente se inicia, debe anunciarse en el servidor y debe indicar si desea iniciar una conversación nueva o unirse a una conversación existente.

## Requerimientos Funcionales

Los requerimientos funcionales generales se enumeran a continuación:

1. El sistema deberá soportar múltiples conversaciones simultáneas.
2. Los clientes deberán poder enviar y recibir mensajes de texto al mismo tiempo.

### Servidor de conversaciones

A continuación se listan los requerimientos funcionales del servidor de conversaciones:

1. Es el primer módulo del sistema que se ejecuta.
2. Lleva un registro de las conversaciones que se encuentran activas y de qué usuarios están en cada conversación. Las conversaciones se identifican con un nombre.
3. Si un cliente solicita iniciar una conversación nueva, el servidor crea el registro y agrega al usuario como único participante de esa conversación.

4. Cuando un cliente abandona una conversación y dicha conversación queda vacía, entonces el servidor debe eliminar el registro.

## **Cientes**

Los requerimientos funcionales para un programa cliente son los siguientes:

1. Cuando un cliente se inicia debe informarlo al servidor de conversaciones.
2. Un cliente estará conectado a una única conversación a la vez. Si el usuario quiere participar en más de una conversación, debe ejecutar tantos clientes como conversaciones en las que quiera participar.
3. El usuario deberá poder indicar si quiere iniciar una conversación nueva (en ese caso el cliente se encontrará solo en la conversación) o si desea unirse a una conversación existente.
4. Cuando el usuario desea abandonar la conversación, deberá informarlo al servidor.
5. El usuario deberá poder consultar las conversaciones disponibles.
6. El usuario deberá poder consultar qué usuarios se encuentran actualmente en la conversación en la cual se encuentra.

## **Requerimientos no Funcionales**

Los siguientes son los requerimientos no funcionales de la aplicación:

1. El proyecto deberá ser desarrollado en lenguaje C o C++, siendo este último el lenguaje de preferencia.
2. La simulación puede no tener interfaz gráfica y ejecutarse en una o varias consolas de línea de comandos.
3. El proyecto deberá funcionar en ambiente Unix / Linux.
4. La aplicación deberá funcionar en una única computadora.
5. El programa deberá poder ejecutarse en "modo debug", lo cual dejará registro de la actividad que realiza en un único archivo de texto para su revisión posterior.
6. Se podrá utilizar cualquier mecanismo de concurrencia visto en la materia para la implementación de este proyecto

## **Tareas a Realizar**

A continuación se listan las tareas a realizar para completar el desarrollo del trabajo práctico:

1. Dividir el proyecto en procesos. El objetivo es lograr que la simulación esté conformada por un conjunto de procesos que sean lo más sencillos posible.
2. Una vez obtenida la división en procesos, establecer un esquema de comunicación entre ellos teniendo en cuenta los requerimientos de la aplicación. ¿Qué procesos se comunican entre sí? ¿Qué datos necesitan compartir para poder trabajar?
3. Tratar de mapear la comunicación entre los procesos a los problemas conocidos de concurrencia.
4. Determinar los mecanismos de concurrencia a utilizar para cada una de las comunicaciones entre procesos que fueron detectadas en el ítem 2. No se requiere la utilización de algún mecanismo específico, la elección en cada caso queda a cargo del grupo y debe estar debidamente justificada.
5. Realizar la codificación de la aplicación. El código fuente debe estar documentado.

## Informe

El informe a entregar junto con el trabajo práctico debe contener los siguientes ítems:

1. Breve análisis de problema, incluyendo una especificación de los casos de uso de la aplicación (punto de vista del usuario que utiliza el cliente).
2. Detalle de resolución de la lista de tareas anterior. Prestar atención especial al ítem 4 de la lista de tareas, ya que se deberá justificar cada uno de los mecanismos de concurrencia elegidos.
3. Diagramas de clases de un cliente y del servidor de conversaciones.
4. Diagramas de transición de estados de un cliente.