

Primer Proyecto: Primavera Concurrente

75.59 - Técnicas de Programación Concurrente I

Objetivo

Se debe implementar un sistema de software que simule el movimiento del mercado de floricultura.

Requerimientos Funcionales

La proximidad de la llegada de la primavera pone nuevamente en acción a los productores de flores.

La cadena productiva del mercado de flores está conformada por: productores del campo (son los que recolectan las flores), centros de distribución y puntos de venta (como el Mercado de flores de Barracas).

Los productores trabajan en el campo y cosechan *rosas* o *tulipanes*, las colocan en cajones de 10 ramos y las transportan al centro de distribución. Cada ramo está identificado con el nombre del productor en una etiqueta que cuelga del tallo. Los productores llevan un cajón al centro de distribución y continúan luego con el trabajo de campo, repitiendo cada uno el proceso sucesivamente.

En los Centros de Distribución se clasifican, separándolas entre rosas y tulipanes y se empaquetan de a 100 ramos para ahorrar costos de Logística, y se transportan a los puntos de venta.

En los puntos de venta, los comerciantes:

1. reciben pedidos por Internet con cantidades de rosas y de tulipanes. Los comerciantes arman los pedidos y emiten el remito que contiene la lista de las flores que componen el pedido (con su correspondiente nombre del productor), y lo entregan al sistema de repartos con bicicletas.
2. atienden al público en general: arman el ramo en el momento, lo decoran y se lo entregan al cliente en el momento. Los clientes ingresan de forma aleatoria al punto de venta y son atendidos de a uno.

Se deberá implementar lo siguiente:

1. Se deberá poder consultar en todo momento: quién es el productor que más flores vendió y cuál es el tipo de flores más comprado.
2. Se deberá poder configurar los pedidos que se reciben por Internet.

Como condiciones adicionales a las planteadas por el ejercicio, se deberán cumplir las siguientes:

1. La simulación debe poder pausarse para ser reanudada su ejecución más adelante. De forma que el stock de los centros de distribución y de venta sea persistido entre ejecuciones.
2. Se deberá poder configurar la cantidad de Centros de Distribución y de puntos de venta.

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. Todas las configuraciones deberán poder realizarse sin recompilar la aplicación.
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. Se deberá poder seguir el recorrido de cada una de las flores.
6. La aplicación deberá funcionar en una única computadora.
7. Las facilidades de IPC que se podrán utilizar para la comunicación entre procesos son:
 - a) Pipes, FIFOs
 - b) Memoria compartida

Tareas a Realizar

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

1. Dividir el proyecto en procesos.
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. Diseñar y documentar el protocolo de comunicación.
4. Realizar la codificación de la aplicación. El código fuente debe estar documentado.

Entrega

La entrega del proyecto comprende lo siguiente:

1. Informe, se deberá presentar impreso en una carpeta o folio y en forma digital (PDF) a través del campus
2. El código fuente de la aplicación, que se entregará únicamente mediante el campus

La entrega en el campus estará habilitada hasta las 19 hs de la fecha indicada oportunamente.

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

1. Detalle de resolución de la lista de tareas anterior.
2. Diagramas de clases.
3. Listado y detalle de las *user-stories*.