

Facultad de Ingeniería  
Técnicas de Programación Concurrente I [75.59]

Federico Farina <federicofarina22@gmail.com>  
Nicolás Vazquez <nickva1988@gmail.com>

22 de junio de 2014

# Índice

<b>1. Enunciado</b>	<b>1</b>
1.1. Objetivo . . . . .	1
1.2. Requerimientos Funcionales . . . . .	1
1.3. Requerimientos no Funcionales . . . . .	2
1.4. Tareas a Realizar . . . . .	2
1.5. Entrega . . . . .	2
1.6. Objetivo . . . . .	3
1.7. Requerimientos Funcionales . . . . .	3
1.8. Requerimientos no Funcionales . . . . .	4
1.9. Tareas a Realizar . . . . .	4
1.10. Entrega . . . . .	5
<b>2. Objetivo</b>	<b>6</b>
<b>3. Análisis del problema</b>	<b>7</b>
3.1. Actores . . . . .	7
3.2. Casos de uso . . . . .	7
3.3. Caso de uso: Recibir auto . . . . .	7
3.4. Caso de uso: Asignar empleado . . . . .	8
3.5. Caso de uso: Atender auto . . . . .	8
3.6. Caso de uso: Almacenar recaudación . . . . .	8
3.7. Caso de uso: Consultar recaudación . . . . .	9
<b>4. Tareas realizadas</b>	<b>10</b>
<b>5. Diagrama de casos de uso</b>	<b>12</b>
<b>6. Diagrama de clases</b>	<b>13</b>
<b>7. Diagrama de estado</b>	<b>14</b>
<b>8. Diagrama de procesos</b>	<b>15</b>

# 1 Enunciado

## 1.1. Objetivo

El objetivo de este proyecto consiste en implementar la simulación parcial del funcionamiento de una estación de servicio.

## 1.2. Requerimientos Funcionales

Esta simulación abarca a el funcionamiento de una estación de servicio. Los requerimientos funcionales son los siguientes:

1. La estación de servicio cuenta con un número determinado de surtidores, un jefe de estación y un conjunto de empleados que atienden de a un auto por vez.
2. Tanto el número de surtidores disponibles como el número de empleados deben ser configurables y se establecen al inicio de la simulación.
3. Cuando un auto llega a la estación de servicio es atendido en primer lugar por el jefe de estación quien asignará un empleado para atender al auto. Si no hay un empleado disponible, el auto se retira de la estación de servicio.
4. La estación de servicio cuenta con dos tipos de clientes: regulares y VIP.
5. El jefe de estación debe atender a los autos por orden de llegada de los mismos, salvo que llegue un cliente VIP, quien tendrá prioridad sobre los clientes regulares.
6. Cuando el empleado recibe un auto de parte del jefe de estación, debe localizar un surtidor libre para atender al auto en cuestión. Si no encuentra ningún surtidor libre, deberá esperar hasta tanto se libere alguno.
7. Una vez que el empleado obtiene un surtidor atiende al auto en cuestión. Mientras el empleado está atendiendo al auto no se puede utilizar el mismo surtidor para otro auto, ni el empleado es capaz de atender más de un auto a la vez.
8. Finalmente, antes de que el auto se retire el empleado deberá cobrar el cargo correspondiente, el cual será almacenado en la caja de la estación de servicio.
9. Todos los empleados guardan la recaudación en la misma caja y sólo un empleado puede utilizar la caja a la vez.
10. Durante cualquier momento de la simulación, el administrador de la estación de servicio podrá consultar la recaudación guardada en la caja. Mientras lo hace, si algún empleado tiene recaudación para guardar, deberá esperar hasta que el administrador finalice su consulta.
11. El administrador deberá tener prioridad sobre los empleados al momento de utilizar la caja.
12. Se deberá poder trackear en el log de la simulación qué empleado atendió a qué auto y qué surtidor utilizó al hacerlo.

### 1.3. 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.

Las facilidades de IPC que se podrán utilizar para la realización de este proyecto son las que abarcan la primera y la segunda parte de la materia.

### 1.4. Tareas a Realizar

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

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.

### 1.5. 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 debe contener los siguientes ítems:

1. Breve análisis del problema, incluyendo una especificación de los casos de uso de la aplicación.

2. Detalle de resolución de la lista de tareas anterior.
3. Diagrama que refleje los procesos, el flujo de comunicación entre ellos y los datos que intercambian.
4. Diagramas de clases realizados.
5. Diagrama de transición de estados de un empleado de la estación.

## 1.6. Objetivo

El objetivo de este proyecto consiste en implementar la simulación parcial del funcionamiento de una estación de servicio.

## 1.7. Requerimientos Funcionales

Esta simulación abarca a el funcionamiento de una estación de servicio. Los requerimientos funcionales son los siguientes:

1. La estación de servicio cuenta con un número determinado de surtidores, un jefe de estación y un conjunto de empleados que atienden de a un auto por vez.
2. Tanto el número de surtidores disponibles como el número de empleados deben ser configurables y se establecen al inicio de la simulación.
3. Cuando un auto llega a la estación de servicio es atendido en primer lugar por el jefe de estación quien asignará un empleado para atender al auto. Si no hay un empleado disponible, el auto se retira de la estación de servicio.
4. El jefe de estación debe atender a los autos por orden de llegada de los mismos.
5. Cuando el empleado recibe un auto de parte del jefe de estación, debe localizar un surtidor libre para atender al auto en cuestión. Si no encuentra ningún surtidor libre, deberá esperar hasta tanto se libere alguno.
6. Una vez que el empleado obtiene un surtidor atiende al auto en cuestión. Mientras el empleado está atendiendo al auto no se puede utilizar el mismo surtidor para otro auto, ni el empleado es capaz de atender más de un auto a la vez.
7. Finalmente, antes de que el auto se retire el empleado deberá cobrar el cargo correspondiente, el cual será almacenado en la caja de la estación de servicio.
8. Todos los empleados guardan la recaudación en la misma caja y sólo un empleado puede utilizar la caja a la vez.
9. Durante cualquier momento de la simulación, el administrador de la estación de servicio podrá consultar la recaudación guardada en la caja. Mientras lo hace, si algún empleado tiene recaudación para guardar, deberá esperar hasta que el administrador finalice su consulta.

## 1.8. 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.

Las facilidades de IPC que se podrán utilizar para la realización de este proyecto son las que abarcan la primera parte de la materia, es decir, hasta el primer parcial. Dichas facilidades son:

- Memoria compartida
- Señales
- Pipes y fifos
- Locks
- Semaforos

Cualquier otra facilidad queda expresamente excluida para este proyecto.

## 1.9. Tareas a Realizar

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

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.

### 1.10. 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 debe contener los siguientes ítems:

1. Breve análisis del problema, incluyendo una especificación de los casos de uso de la aplicación.
2. Detalle de resolución de la lista de tareas anterior.
3. Diagrama que refleje los procesos, el flujo de comunicación entre ellos y los datos que intercambian.
4. Diagramas de clases realizados.
5. Diagrama de transición de estados del jefe de estación.

## 2 Objetivo

El objetivo de este proyecto es el desarrollo de una aplicación conocida como ConcuStation. Esta aplicación permitirá simular el funcionamiento de una estación de servicio. Al iniciar la aplicación, el usuario podrá decidir el número de empleados y surtidores que desee. Los empleados atienden un auto por vez y almacenan la recaudación en una caja común. En todo momento se podrá consultar el saldo disponible en la caja.



### 3 Análisis del problema

Para el análisis del problema, se identificaron los actores que intervienen en el sistema. A continuación se presenta una breve descripción de cada uno.

#### 3.1. Actores

- Jefe de estación: Encargado de recibir los autos que ingresan en la estación de servicio y asignar un empleado para su atención.
- Empleado: Encargado de atender los autos y almacenar la recaudación en la caja.
- Administrador: Consulta la recaudación disponible en la caja.

#### 3.2. Casos de uso

Se identificaron los siguientes casos de uso:

- Recibir auto: El jefe de estación recibe un nuevo auto que ingresa a la estación de servicio.
- Asignar empleado: El jefe de estación asigna un empleado disponible para atender el auto que ha ingresado a la estación de servicio.
- Atender auto: El empleado atiende un auto asignado previamente por el jefe de estación.
- Almacenar recaudación: El empleado almacena la recaudación obtenida al atender un auto en la caja de la estación de servicio.
- Consultar recaudación: El administrador de la estación de servicio consulta la recaudación contenida en la caja.

#### 3.3. Caso de uso: Recibir auto

1. **Descripción:**
2. **Actores participantes:** Jefe de Estación
3. **Pre-condiciones:** -
4. **Flujo principal**
  - a)
  - b)
  - c)
  - d)
  - e) Fin del caso de uso

**Post-condiciones:**

### 3.4. Caso de uso: Asignar empleado

1. **Descripción:**
2. **Actores participantes:**
3. **Pre-condiciones:**
4. **Flujo principal**
  - a)
  - b)
  - c)
  - d)
  - e) Fin del caso de uso

**Post-condiciones:**

### 3.5. Caso de uso: Atender auto

1. **Descripción:** El empleado atiende el auto asignado por el jefe de estación y almacena la recaudación en la caja.
2. **Actores participantes:** Empleado
3. **Pre-condiciones:** El auto a atender debe ser previamente asignado por el jefe de estación. El empleado no puede estar ocupado atendiendo otro auto. El tiempo de carga debe ser menor al tiempo de simulación restante.
4. **Flujo principal**
  - a) El empleado busca un surtidor libre para atender el auto
  - b) El empleado determina el tiempo de carga según la capacidad del tanque de nafta del auto
  - c) El empleado carga nafta en el auto
  - d) El empleado almacena la recaudación resultante en la caja de la estación de servicio
  - e) Fin del caso de uso

**Post-condiciones:** La recaudación es almacenada en la caja y el empleado se encuentra disponible para atender un nuevo auto

### 3.6. Caso de uso: Almacenar recaudación

1. **Descripción:**
2. **Actores participantes:** Empleado
3. **Pre-condiciones:**
4. **Flujo principal**
  - a)

- b)*
- c)*
- d)*
- e)* Fin del caso de uso

**Post-condiciones:** La recaudación es almacenada en la caja y el empleado se encuentra disponible para atender un nuevo auto

### 3.7. Caso de uso: Consultar recaudación

1. Descripción:
2. Actores participantes: Administrador
3. Pre-condiciones:
4. Flujo principal
  - a)*
  - b)*
  - c)*
  - d)*
  - e)* Fin del caso de uso

**Post-condiciones:**

## 4 Tareas realizadas

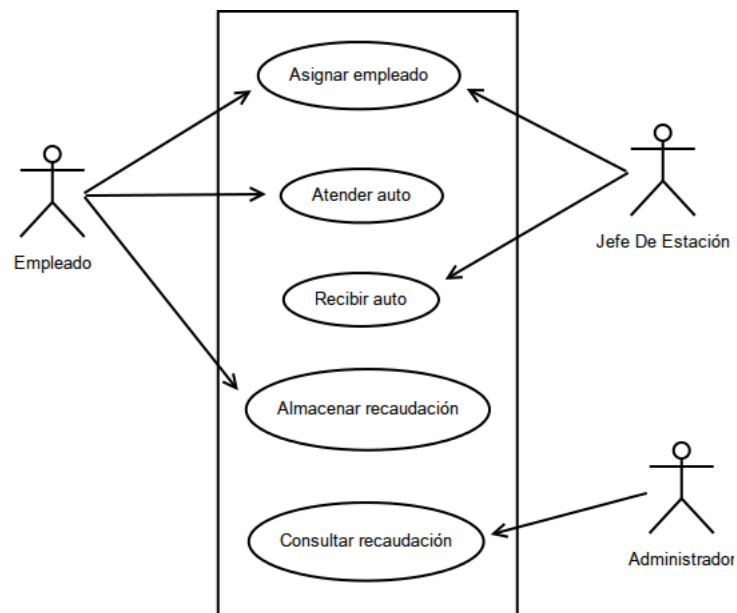
1. Dividir el programa en procesos. El objetivo es lograr que cada programa participante este conformado por un conjunto de procesos que sean lo más sencillos posible.
  - Se crean dos procesos principales: Un proceso asociado a la vista (interfaz gráfica de usuario) y un proceso que controla el flujo de ejecución de la simulación.
  - Se crea un proceso cada vez que entra un auto a la estación de servicio: Al ingresar es recibido por el jefe de estación, el cual busca algún empleado que se encuentre disponible para ser asignado a atender el auto. Si hay algún empleado disponible, se crea un nuevo proceso para atender el auto. La duración de la atención está determinada por la capacidad en litros del tanque de nafta del auto.

NOTA: En el caso de no haber empleados disponibles al ingresar un nuevo auto, no es posible atenderlo y el mismo se retira de la estación de servicio.
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?
  - El proceso principal se comunica con la vista por medio de un canal de comunicación. Al ingresar un nuevo auto, se puede determinar su capacidad mediante la interfaz gráfica de usuario. La vista escribe en el canal dicha capacidad y el proceso principal la lee para determinar el tiempo que necesita el auto en ser atendido.
  - El proceso principal y los procesos responsables de atender los autos que ingresan en la estación de servicio dejan el registro de su actividad en un único archivo de texto si el programa es ejecutado en modo debug. Este archivo debe ser compartido por todos los procesos.
  - Cada vez que se lanza un proceso para atender un nuevo auto, el proceso escribe la recaudación resultante en una caja común. Esta caja es compartida por todos los procesos y por el administrador de la estación de servicio, el cual en todo momento puede consultar la recaudación contenida en la caja.
  - La vista debe comunicar al proceso principal la finalización del tiempo de simulación. Este tiempo es controlado por la vista mediante un timer.
3. Tratar de mapear la comunicación entre los procesos a los problemas conocidos de concurrencia.

Dominio	Problema de Concurrencia
Entrada y atención de autos	Lector-Escritor
Caja	Exclusión mutua
Empleados disponibles	Productor-Consumidor
Nuevos autos	Productor-consumidor

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.
  - Si el programa es ejecutado en modo debug, se utiliza un Lock File para sincronizar correctamente la escritura de los procesos en el archivo de texto.
  - Para determinar que surtidor se encuentra libre, se utiliza una memoria compartida con tantas posiciones como surtidores existan en la estación de servicio. Esta memoria es compartida por todos los empleados.
  - La caja es una memoria compartida por todos los procesos que se lanzan cada vez que ingresa un nuevo auto. Dado que estos procesos pueden ejecutarse concurrentemente, se utiliza un semáforo de exclusión mutua para la sincronización de la lectura y escritura en la memoria compartida.

## 5 Diagrama de casos de uso



**Figura 1:** Diagrama de casos de uso

## 6 Diagrama de clases

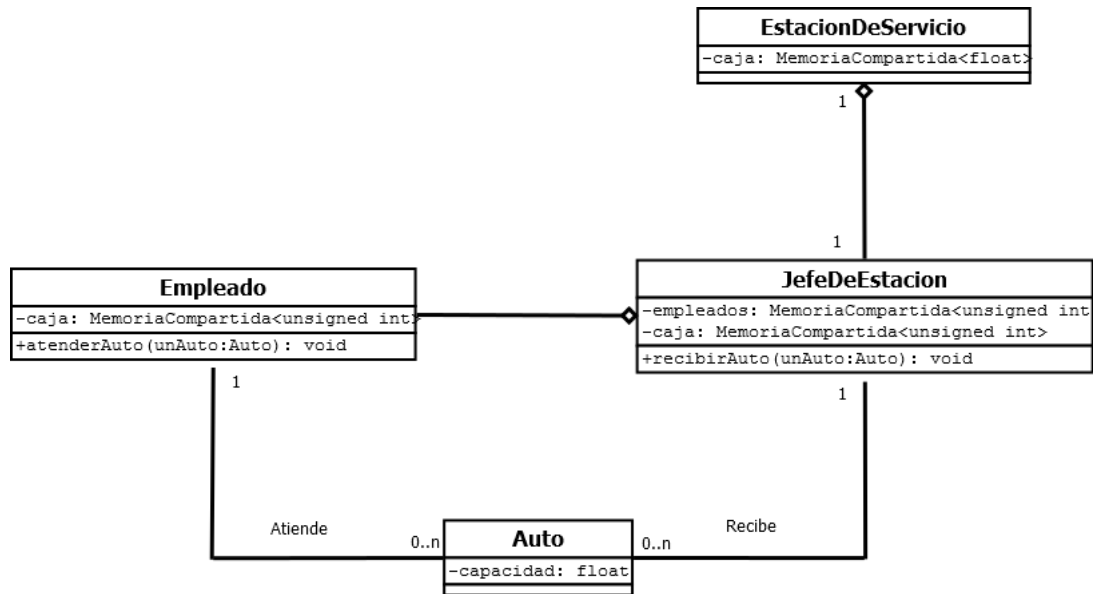
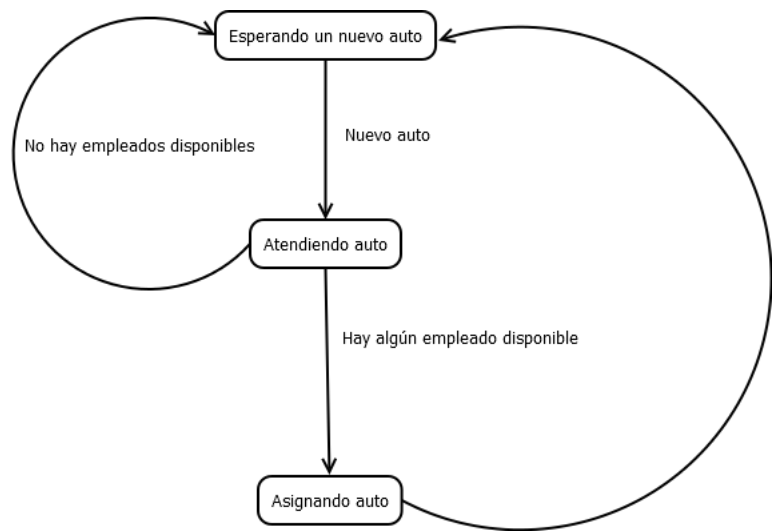


Figura 2: Diagrama de clases

## 7 Diagrama de estado



**Figura 3:** Diagrama de estado del jefe de estación



## 8 Diagrama de procesos