

Informe de Trabajo Práctico Final

Programación Concurrente

Joaquín Medel
Legajo: FAI-3386

[Repositorio GitHub](#)

1 de diciembre de 2025

Facultad de Informática (FAI) - Universidad Nacional del Comahue

Descripción del diseño e implementación de la solución concurrente
para el Complejo Invernal.

Índice

| | |
|--|----------|
| 1. Enunciado | 3 |
| 2. Conceptos Generales | 3 |
| 3. Recursos Compartidos | 4 |
| 3.1. Complejo Invernal | 4 |
| 3.1.1. Centro de Clases Grupales | 5 |
| 3.1.2. Confitería | 5 |
| 3.1.3. Entrada | 5 |
| 3.1.4. Medio de Elevación | 6 |
| 4. Hilos | 6 |
| 4.1. Instructor | 6 |
| 4.2. Persona | 6 |
| 4.3. Reloj | 6 |

1. Enunciado

Se desea simular la actividad diaria del Complejo Invernal “Caída Rápida”. Dicho complejo ha implementado un nuevo sistema de pases para el ingreso de los esquiadores. El complejo cuenta con 4 medios de elevación y cada medio de elevación cuenta con un grupo de n molinetes (n puede variar entre 1 y 4) que dan el acceso al medio al esquiador. Por ejemplo: el medio de elevación Silla Triple del Complejo posee 3 (tres) molinetes que le dan acceso a 3(tres) esquiadores por vez. En cada molinete hay un lector de tarjetas cuyo fin es dejar pasar sólo aquellas personas que tienen pase, pero también sirve para contar el uso de cada molinete. La suma de todos los molinetes de un medio determina la cantidad de veces que fue utilizado el medio. Al final del día se desea saber cuantos esquiadores utilizaron cada medio de elevación. El complejo habilita sus medios de elevación, diariamente, a las 10:00hs; y permanecen habilitados hasta las 17:00hs. Durante ese periodo, los esquiadores tendrán acceso a los medios de elevación, ilimitadamente. Cada esquiador experimentado esquía por cierto tiempo utilizando los distintos medios de elevación para acceder a las pistas, descansa, toma algo en la confitería y continua esquiando. El Complejo ofrece a los visitantes la posibilidad de contratar clases grupales diarias de sky o de snowboard, dictadas por un staff de 5(cinco) instructores. Para que una clase pueda comenzar, es necesario que se tenga un grupo de 4 (cuatro) alumnos y 1(un) instructor disponible, mientras tanto los instructores permanecen esperando en la Cabina de Instructores. En el caso de que pasado un tiempo prudencial no se logre armar el grupo, los esquiadores desisten de tomar la clase, y se les devuelve el dinero. Por otro lado, el Complejo dispone de una confitería con capacidad para 100 personas, que cuenta con dos mostradores para servirse comidas rápidas y un mostrador para postre. El circuito es el siguiente: la persona ingresa a la confitería y paga en una única caja el menú que desea consumir, dicho menú puede o no incluir postre. A continuación, pasa por uno de los mostradores a retirar la comida pagada, y luego retira el postre. Es importante destacar, que si la persona llega hasta la caja es porque existe disponibilidad de mesas. Realizar la implementación completa de la situación presentada en Java. Teniendo en cuenta:

- Que el código debe estar correctamente documentado.
- Que se cuenta con un set de prueba que permita probar el funcionamiento de la simulación, es decir, considerar un número apropiado de esquiadores. Decidir aleatoriamente el comportamiento de cada esquiador.

El objetivo de este trabajo es modelar y resolver problemas de concurrencia, sincronización y exclusión mutua dentro de un sistema que simula un complejo de actividades invernales.

2. Conceptos Generales

Para la resolución de este trabajo práctico se han aplicado diversos mecanismos de sincronización propios de la programación concurrente. Además se tomaron ciertas decisiones de diseño para lograr un trabajo más prolijo y manejable, dentro de las cosas "extras" que se implementaron esta la interfaz, primero todo se imprimía por consola utilizando una clase y métodos estáticos se pasó todos los mensajes de

debug a esa interfaz, luego de los mensajes de debug tambien se implemento una representacion visual utilizada durante el desarrollo del trabajo, para poder hacer funcion el trabajo simplemente se necesita ejecutar el main de App.java encontrado en /(default package). El set de pruebas esta conformado por 40 personas , 1 reloj y 5 instructores.Utilizando numeros Random se consigue una aleatoriedad en las acciones que toman los esquiadores/personas

3. Recursos Compartidos

A continuación se presenta un resumen de los recursos compartidos identificados en el sistema y la herramienta de concurrencia seleccionada para su gestión:

| Recurso Compartido | Mecanismo de Sincronización |
|---------------------------------------|---|
| Entrada al Complejo | Monitores (<code>synchronized</code> , <code>wait</code> , <code>notify</code>) |
| Centro de Clases Grupales | Locks Explícitos (<code>ReentrantLock</code>) y Variables de Condición (<code>Condition</code>) |
| Confitería (Mesas, Caja, Mostradores) | Semáforos (<code>Semaphore</code>) |
| Medios de Elevación | Barreras Cíclicas (<code>CyclicBarrier</code>) y Contadores Atómicos (<code>AtomicInteger</code>) |

Cuadro 1: Mapeo de recursos y herramientas de sincronización.

En esta sección se detalla la implementación de los recursos pasivos. El sistema se estructura de forma jerárquica.

3.1. Complejo Invernal

El **Complejo Invernal** actúa como el recurso agrupador principal. Es la clase de alto nivel que contiene y gestiona el acceso a las distintas instalaciones (Recursos).

- **Encapsulamiento:** Los hilos (Personas, Instructores) solo conocen e interactúan con este recurso principal. El resto de los recursos (Confitería, Clases, etc.) son internos o "privados", encapsulando la lógica específica para mantener la cohesión del sistema.
- **Inicialización:** Al instanciar el complejo, se inicializan todos los sub-recursos y los hilos internos. Además, se crean los medios de elevación configurables con una cantidad de molinetes en un rango determinado.

A continuación se describen los componentes internos:

3.1.1. Centro de Clases Grupales

Recurso utilizado para sincronizar **Personas** con **Instructores** mediante el uso de **Locks** y variables de condición (**Condition**).

- **Lógica del Instructor:** Al intentar instruir, el hilo toma el Lock. Si no hay suficientes alumnos, espera en la condition **hayAlumnos**. Cuando es despertado (se alcanzó el cupo), decrementa la cantidad de alumnos en espera (ya que pasan a estar en clase) y notifica a cada uno de ellos. Finalmente, libera el Lock y simula el tiempo de la clase.
- **Lógica de la Persona:** Toma el Lock e incrementa el contador de alumnos esperando. Si su llegada completa el cupo necesario, notifica al Instructor. Luego, espera a que la clase comience con un tiempo máximo de tolerancia (*timeout*).
 - *Si entra a clase:* Libera el Lock y simula la actividad.
 - *Si expira el tiempo:* Se retira de la cola, decrementando el contador de espera y liberando el recurso.

3.1.2. Confitería

Utilizado para sincronizar el flujo de personas en el área de comida. La implementación se basa en **Semáforos** para controlar capacidades y flujos. Consta de 4 semáforos principales: Mesas, Caja, Mostrador de Comida Rápida y Mostrador de Postres.

1. **Ingreso (Mesas):** Para ingresar, la persona debe adquirir un permiso del semáforo de mesas. Si no hay disponibilidad, se bloquea hasta que una mesa se libere.
2. **Pago (Caja):** Simula una cola utilizando un semáforo. La persona "paga" (simulación de tiempo) y libera el permiso.
3. **Comida Rápida:** Similar a la caja, se hace cola en el semáforo correspondiente para retirar el pedido.
4. **Postre (Opcional):** Dependiendo de un valor aleatorio, la persona puede decidir ir al sector de postres, repitiendo la lógica de cola y espera mediante semáforos.
5. **Salida:** Tras consumir (simulación de tiempo), la persona libera la mesa (permiso del semáforo inicial) y se retira.

3.1.3. Entrada

Controla el acceso general al predio utilizando **Monitores**.

- **Hilo Reloj:** Es el encargado de modificar el estado del monitor. Periódicamente actualiza la hora y, al llegar al horario de cierre o apertura, cambia la bandera de estado (abierto/cerrado) y notifica a los hilos esperando.

- **Persona:** Al intentar ingresar, verifica el estado. Si está abierto, entra. Si está cerrado, queda bloqueada esperando la notificación de apertura.

3.1.4. Medio de Elevación

Gestiona el transporte de personas hacia las pistas utilizando **CyclicBarrier**.

- **Selección:** La persona selecciona aleatoriamente un medio de elevación disponible.
- **Barrera:** La persona se une a la barrera (`await()`) esperando que se llene el cupo (la capacidad de la silla/cabina).
- **Ejecución:** Cuando la barrera se completa (llega la última persona necesaria), se dispara la acción de subir. Se actualizan los contadores de uso y se simula el tiempo de ascenso. Durante este tiempo, los hilos permanecen sincronizados.
- **Timeout:** Se contempla el manejo de excepciones (`BrokenBarrierException`). Si una persona espera demasiado tiempo sin que se llene la cabina, la barrera se rompe y las personas desisten de subir para no quedar bloqueadas indefinidamente.

4. Hilos

Utilizando la clase `Thread` se implementaron:

4.1. Instructor

Representa a los profesionales encargados de las clases.

- **Rol:** Esperar a que se forme un grupo, dar la clase y liberar recursos.
- **Interacción:** Interactúa principalmente con el *Centro de Clases Grupales*.

4.2. Persona

Representa a los usuarios o clientes del complejo.

- **Rol:** Entrar al complejo, utilizar el medio de elevación, asistir a clases o ir a la confitería.
- **Ciclo de vida:** Compite por los recursos compartidos con otras instancias de *Persona*.

4.3. Reloj

Hilo encargado del control temporal.

- **Rol:** Simula el paso del tiempo y determina la apertura/cierre.
- **Función:** Notifica a los demás hilos cuando se cumple el horario de cierre.