# PARADIGMAS DE LA PROGRAMACIÓN

## Práctica de Laboratorio (E1)

## Convocatoria Ordinaria – 2024/2025

*Publicado el 05 de noviembre de 2024*

## Simulación de una fábrica de galletas

### Programación Concurrente

Se desea modelar el funcionamiento de una fábrica de galletas. La fábrica comienza con cinco **reposteros**, tres **hornos** y tres **empaquetadores**. Los **reposteros** se encargan de producir tandas de galletas, mientras que los **hornos** hornean las galletas cuando alcanzan su capacidad máxima. Los **empaquetadores** son responsables de llevar las galletas horneadas al almacén.

Los **reposteros** tardan entre 2 y 4 segundos en producir una tanda de galletas, que puede ser entre 37 y 45 galletas, y luego las depositan en un horno. Una vez que completan de 3 a 5 tandas de producción, realizan una parada para el café. Durante este descanso, los reposteros se preparan café, pero solo hay una **cafetera** disponible, por lo que solo un repostero puede prepararse café a la vez. En preparar café tardan 2 segundos, y, una vez se han preparado el café, descansan entre 3 y 6 segundos. Los **hornos** tienen una capacidad máxima de 200 galletas y solo comienzan a hornear cuando están llenos, tardando 8 segundos en hornear todas las galletas. Los empaquetadores recogerán las galletas en lotes de 20, lo cual les toma entre 0,5 y 1 segundos por cada lote, y, cuando llegan a 100 galletas, las empaquetarán. No será hasta entonces cuando las lleven al almacén, lo que les toma entre 2 y 4 segundos. El **almacén** tiene una capacidad máxima de 1000 galletas y contará con un botón que el usuario podrá pulsar para "comer" galletas. Cada pulsación del ratón debe consumir 100 galletas. Mientras el almacén esté lleno, los empaquetadores no podrán depositar más galletas, por lo que el usuario deberá vaciar el almacén regularmente para permitir nuevas entregas.

Consideraciones adicionales:

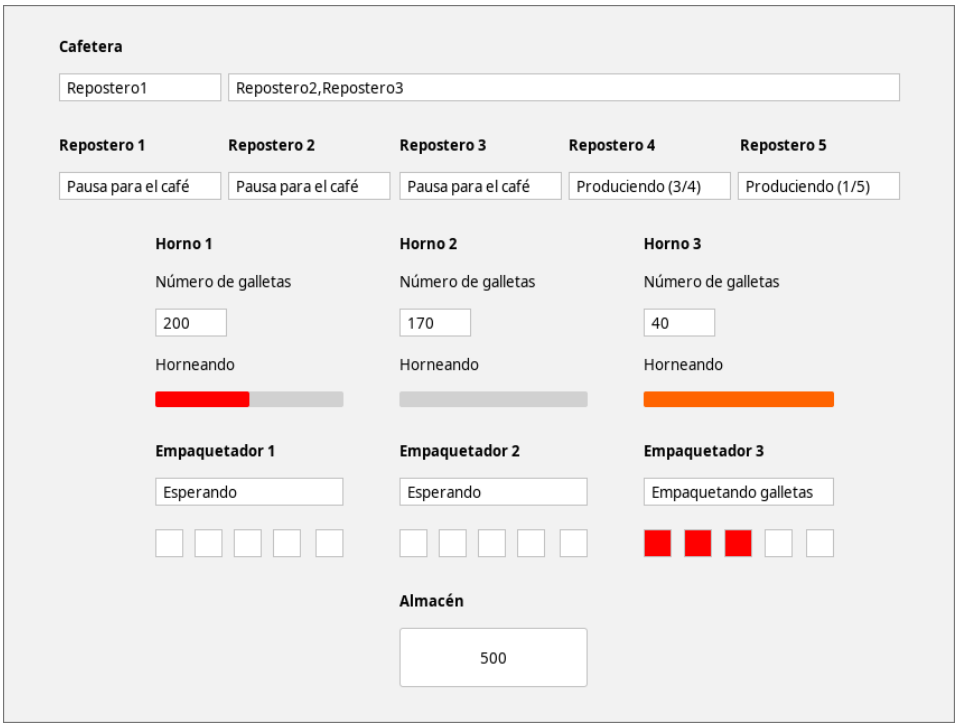
- Todos los roles (**reposteros**, **hornos** y **empaquetadores**) serán modelados obligatoriamente como hilos.
- La fábrica tendrá, en total, 5 **reposteros**, 3 **hornos** y 3 **empaquetadores**.
- Cada entidad tendrá un identificador único: "Repostero1", "Empaquetador1", "Horno1", etc.

- Los **reposteros** irán llenando los **hornos** en orden. Cuando en un horno no se puedan meter galletas, probarán con los siguientes hornos hasta que haya uno libre. Si no hay hornos disponibles, los reposteros esperarán hasta que uno quede libre.
- Los **hornos** hornean solo cuando están completamente llenos.
- El flujo de traslado de galletas del horno al almacén solo se detiene cuando el almacén está lleno.
- Cada **empaquetador** solo puede recoger galletas de un horno específico. El empaquetador 1 recoge galletas del horno 1, el empaquetador 2 del horno 2, y así sucesivamente.
- Antes de que los **reposteros** puedan volver a meter galletas después de un horneado, el horno se debe vaciar por completo.
- Si el **repostero** va a meter más galletas de las que faltan para completar la capacidad del horno, las galletas restantes se desperdician y se registran en el log.

Los tiempos de los reposteros, hornos y empaquetadores se generarán aleatoriamente mediante las funciones **random** de Java, y todo el comportamiento del sistema se guardará en un log (un fichero de texto llamado **"evolucionGalletas.txt"**), además de mostrarse gráficamente por pantalla, de forma que sea sencillo analizar lo sucedido. El log guardará los eventos que van teniendo lugar, por ejemplo: "Repostero1-deja 40 galletas en el Horno1", "Repostero2 desecha 25 galletas", "Empaquetador1 recoge 20 galletas del Horno1", "Empaquetador1 esperando", etc. En cada línea de dicho log deberá constar la marca de tiempo (incluyendo la fecha, la hora, el minuto y el segundo determinado en el que tuvo lugar el evento) y el evento en sí.

Una posible interfaz del sistema sería la que se puede apreciar en la siguiente imagen:

InterfazServidor



## Programación Distribuida

Basándose en la aplicación concurrente descrita, se incluirá un nuevo módulo capaz de visualizar de forma remota el contenido de la fábrica (usando programación concurrente distribuida). Este módulo permitirá visualizar, mediante una interfaz gráfica, el estado del almacén, de los hornos y del personal.

Se desarrollarán dos programas:

- Un **servidor**, cuyo código base será el programa desarrollado en el apartado "Programación Concurrente", con interfaz gráfica y ampliado con la funcionalidad correspondiente para dar soporte al módulo de visualización de programación distribuida.
- Un **cliente** (módulo de visualización) que permita al usuario visualizar el estado del almacén, los hornos y los reposteros de forma remota a través de una interfaz gráfica. La interfaz mostrará el número de galletas generadas por los reposteros, el número de galletas desperdiciadas, las galletas horneadas y si cada horno está actualmente en proceso de horneado o no. Además, se visualizarán la cantidad de galletas en el almacén y las galletas consumidas por el usuario. La interfaz proporcionará también la capacidad de pausar o reanudar individualmente a cada uno de los reposteros.

La información mostrada será actualizada automáticamente en la interfaz del cliente con una periodicidad de 1 segundo, reflejando el estado real de la fábrica en todo momento.

Un posible ejemplo de interfaz gráfica para la parte del cliente se muestra en la siguiente figura.

InterfazCliente


Se podrán utilizar todos los mecanismos vistos en clase para resolver todos los problemas de comunicación y sincronización que se plantean en este enunciado. No obstante, se deben utilizar los mecanismos de sincronización y comunicación que resuelvan el problema de la forma más eficiente y óptima posible.

# Condiciones de entrega

1. *La práctica se realizará (**opcionalmente**) **por parejas** y deberá ser entregada **antes** de la fecha indicada en el Aula Virtual, a través de la tarea correspondiente, mediante la subida de dos archivos: la memoria de la práctica en formato PDF o DOC y el proyecto Netbeans completo, comprimido como **ZIP** (no utilizar extensión .rar). No se aceptarán trabajos enviados pasada la fecha límite de entrega.*
2. *La entrega fuera del plazo indicado en el Aula Virtual supondrá una reducción en la calificación final, siendo del 25% si se entrega el día siguiente a la fecha límite, o del 50% si se entrega dentro de los dos días siguientes. La entrega más allá de esos dos días no será admitida bajo ninguna circunstancia.*
3. *El proyecto entregado deberá ser un **proyecto de NetBeans**. No se admitirán proyectos realizados con otros entornos de desarrollo.*
4. *Para aprobar, es condición necesaria que todos los programas funcionen correctamente y de acuerdo a las especificaciones indicadas en los enunciados.*
5. *Se debe desarrollar la solución haciendo uso de buenas prácticas de programación. Por ejemplo, es necesario que todos los nombres de las clases comiencen por una letra mayúscula y todos los nombres de atributos y métodos comiencen por una letra minúscula; los atributos deberán ser privados, y sólo se podrá acceder a ellos mediante métodos getter y setter.*
6. ***Ambas partes** (concurrente y distribuida) de la práctica de laboratorio **se deberán entregar juntas** (es decir, en un único fichero ZIP y una única memoria), ya que la parte de programación distribuida se construye sobre la parte concurrente.*
7. *Si la práctica es realizada por una pareja, **sólo uno de los integrantes deberá subirla** al aula virtual, indicando el nombre de ambos alumnos.*
8. *En la portada de la memoria deberán figurar los datos siguientes:*
    a. ***Grado en Ingeniería en Sistemas de Información***
    b. ***Curso 2024/2025 – Convocatoria Ordinaria***
    c. ***DNI – Apellidos, Nombre***
9. ***La memoria deberá incluir, como anexo, el código fuente** del programa. Si esto no fuera así, la práctica no podrá ser aprobada.*
10. *La memoria explicativa de la práctica realizada deberá incluir, en el orden siguiente: 1) un análisis de alto nivel; 2) diseño general del sistema y de las herramientas de sincronización utilizados; 3) las clases principales que intervienen con su descripción (atributos y métodos); 4) un diagrama de clases que muestren cómo están relacionadas; y 5) el código fuente, como anexo.*
11. *Dicha documentación, exceptuando el código, no deberá extenderse más de 20 páginas. La calidad de la documentación – presentación, estructura, contenido, redacción – será un elemento básico en la evaluación de la práctica.*

12. *De cara a la defensa/examen de la práctica, el estudiante podrá llevar impresa una copia de la memoria entregada con la práctica.*

13. *La resolución de la práctica debe ser genuina y realizada desde cero, es decir, no se podrá utilizar ningún tipo de código fuente de prácticas presentadas a esta convocatoria o en convocatorias anteriores, bien sean propias o de otros compañeros. En caso de detectarse esta situación, la práctica no será evaluada y tendrá una calificación de 0 Suspenso.*