

# Programación Orientada a Objetos

## Práctica 2

**Inicio: 17 de Marzo**

**Entrega: 20 de Abril a las 23:55 p.m.**

Esta práctica tiene como objetivo familiarizarse con el uso de clases, herencia, polimorfismo, control de acceso, colecciones, excepciones y entrada/salida en Java. Se valorará no sólo el correcto funcionamiento del código entregado, sino también que se haya realizado un buen diseño orientado a objetos.

Se pide desarrollar una aplicación para definir y simular cadenas de montaje que realizan el ensamblaje de productos a partir de piezas sencillas. Para ello, el ejercicio se ha organizado en varios pasos sucesivos.

### Paso 1: Cadenas de montaje (4.5 puntos)

Se desea poder definir **cadenas de montaje** para el ensamblaje de productos. Una cadena de montaje puede incluir máquinas de los siguientes tipos: generadores de piezas, ensambladores, y empaquetadoras. El transporte de piezas y productos entre las máquinas se realiza a través de cintas transportadoras. Cintas y máquinas tienen un identificador único que se debe asignar automáticamente en el constructor, y no es modificable.

Las **cintas transportadoras** tienen una capacidad configurable que indica el número máximo de piezas y productos que es capaz de transportar simultáneamente. Cada cinta transportadora tiene una o varias cintas o máquinas como entrada, y una única cinta o máquina de salida.

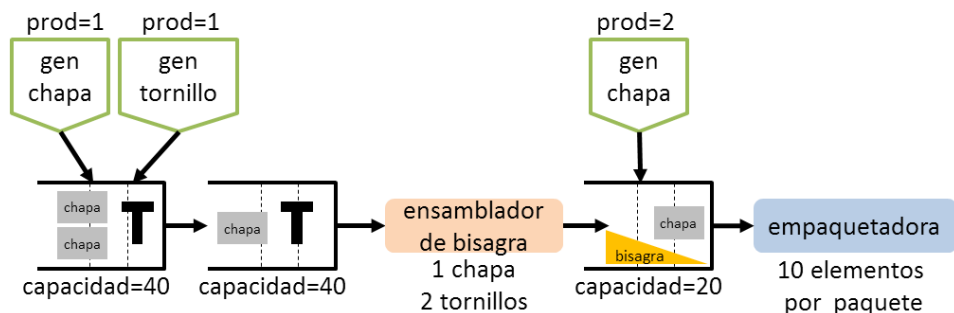
Los **generadores** se encargan de producir piezas de un único tipo (chapas o tornillos) y tienen una productividad que establece cuántas piezas generan de una vez, teniendo que ser 1 como mínimo. Los generadores siempre deben estar conectados a una cinta transportadora de salida, que será donde depositen las piezas que generan. En cambio, los generadores no tienen cintas transportadoras de entrada.

Los **ensambladores** ensamblan chapas y tornillos para construir productos complejos. Un ensamblador se configura con el tipo de producto que elabora (ej. "bisagra") y los elementos que necesita para elaborarlo (ej. 1 chapa y 2 tornillos). El número mínimo de elementos a ensamblar debe ser 2, ya sean del mismo o de distinto tipo. Un ensamblador puede tener una o más cintas transportadoras de entrada, pero sólo una cinta de salida.

Las **empaquetadoras** generan paquetes de productos. El número de productos por paquete es configurable, y debe ser mayor que 0. Las empaquetadoras tienen una cinta transportadora de entrada, y ninguna de salida.

Construye y entrega la clase `CadenaMontaje`, que deberá tener métodos para configurar una cadena de montaje, así como un método `validar` que compruebe que la cadena cumple las restricciones anteriores, o en caso contrario, lance una excepción propia (es decir, que herede de la clase `Exception` de java) con la lista de errores detectados.

Construye un `main` que cree la siguiente cadena de montaje y la valide. Además, elabora el diagrama de clases con el diseño de tu código, que deberás entregar junto con el código resultante en este apartado.



---

## Paso 2: Simulador (4.5 puntos)

Extiende la clase `CadenaMontaje` con un método `simular` que permita simular una cadena de montaje válida (si es inválida, el método lanzará una excepción). En cada paso de simulación, se deberá identificar qué acciones pueden realizarse, y se realizará una de ellas de manera aleatoria. Las acciones posibles en cada paso de simulación son:

- generación de una pieza por un generador. Para poder realizar esta acción, la cinta conectada al generador debe tener espacio disponible para el número de piezas a generar (productividad del generador).
- mover una pieza o producto de una cinta a otra. La segunda cinta deberá tener espacio disponible para ello.
- ensamblaje de un producto. A la hora de ensamblar un producto, el ensamblador comprobará si las cintas transportadoras de entrada contienen los elementos necesarios para crear el producto, y si la cinta de salida tiene espacio disponible. De ser así, el ensamblador eliminará los elementos que necesite de la cinta (o cintas) de entrada, y depositará el producto ensamblado en la cinta de salida.
- empaquetado de productos. Para realizar esta acción, la cinta de entrada de la empaquetadora debe tener suficientes elementos para formar un paquete. De ser así, la empaquetadora eliminará los elementos necesarios de la cinta de entrada y creará un paquete.

El método `simular` realizará 100 pasos de simulación (podrán ser menos si la cadena llega a un estado de bloqueo, por ejemplo, si todas las cintas transportadoras están llenas y no hay ensambladores ni empaquetadoras que puedan procesar las piezas/productos que contienen). El resultado de la simulación será un fichero que contenga las acciones realizadas (ej. generador 1 genera 1 tornillo, 1 tornillo se mueve de la cinta 1 a la cinta 2, etc.), así como el número total de paquetes generados y su contenido. Si la simulación llegara a un estado de bloqueo, el fichero también deberá incluir una descripción del contenido de las cintas al llegar al estado de bloqueo.

Actualiza el diagrama de clases realizado en el apartado 1, y entrégalo junto con el código.

---

## Paso 3: Menú interactivo (1 punto)

Extiende tu clase `CadenaMontaje` con un método `simular_interactivo` que permita la simulación interactiva de la cadena de montaje. En cada paso de simulación se mostrarán por pantalla las acciones posibles, y el usuario elegirá la que realizar. En cualquier momento el usuario podrá finalizar la simulación.

### Normas de Entrega

Comprime las clases realizadas en la práctica, junto con el diagrama de clases de la aplicación, en un único fichero zip/rar con nombre *p2-nia1-nia2.zip*, donde **nia1** y **nia2** serán los números de identificación de cada alumno de la pareja. La finalidad de las clases y métodos deberá estar documentada adecuadamente con comentarios javadoc en el código. Uno de los miembros de la pareja entregará el fichero a través de moodle antes de la fecha de entrega indicada en la cabecera del enunciado.