Un dibujo en blanco y negro

El contenido generado por IA puede ser incorrecto.

# Introducción

Una cola se asemeja a la pila en varios aspectos; sin embargo, a diferencia de esta, se rige por una premisa tipo FIFO (Primero en entrar, primero en salir). De esta forma, los datos se introducen uno por uno como una transportadora. En la fila, también se emplea las operaciones push (para introducir un elemento) y pop (concluir ese elemento).

# Código

## Nodo

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Como se podrá notar, el dato almacena el contenido real que se debe guardar en el nodo, asignado como Objeto para mayor flexibilidad. Mientras que el Nodo siguiente, se refiere al siguiente objeto de la lista, en caso de que sea nulo, indica el final de la lista.

El constructor inicia el campo dato con el valor proporcionado. Así un nodo recién creado está aislado hasta que se asigne un enlace hacia otro nodo.

## Lista Simple Enlazada

Texto

El contenido generado por IA puede ser incorrecto.

A recomendación de un algoritmo, una lista simple enlazada parece ser la mas conveniente para poder implementar la pila y la cola de una forma menos compleja. La clase almacena tres campos privados que controlan el estado de la lista: cabeza (primer elemento), cola (último elemento) y tamaño (longitud de la lista).

### Insertar Final

Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Este método recibe un objeto, crea un nuevo elemento y lo añade a la extrema derecha de la lista:

* En caso de que la lista este vacía, tanto el primero como el último se refieren al nuevo elemento.
* Por el contrario, el elemento actual enlaza su campo con el siguiente, haciendo que cola sea el elemento más reciente.
* Por consecuencia, aumenta la longitud de la lista.

### Insertar Inicio

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

El método añade la cabeza de la lista y ajusta los punteros siguientes:

* En caso de que la lista este vacía, tanto el primero como el último se refieren al nuevo elemento.
* Si ya hay elementos, el elemento siguiente se enlaza a la cabeza, mientras que la variable se actualiza al nuevo elemento.
* Aumenta la longitud de la lista.

### Eliminar Inicio

Texto

El contenido generado por IA puede ser incorrecto.

Esto es para liberar el primer elemento de la lista:

* Si la lista está vacía, se captura la excepción RuntimeException indicando lo que pasa.
* Guarda el valor de cabeza.dato para retornarlo al final.
* Actualiza la cabeza para apuntar al siguiente nodo.
* Si la nueva cabeza es nula, significa que la lista ya está vacía.
* Decrementa la longitud de la lista y devuelve el valor almacenado.

### Obtener Inicio

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Consulta el valor almacenado en la cabeza sin modificar la lista:

* Se captura la excepción como “VACIO” si no hay nodos.
* Por otro lado, se retorna directamente a cabeza.dato.

### Estado

public boolean vacio() {return tamaño == 0; }  
public int tamaño() {return tamaño; }

EL programa indicará que está vacío si la longitud es cero, el otro método retorna la longitud sin recorrer la lista.

### Mostrar

Texto

El contenido generado por IA puede ser incorrecto.

* Usa el puntero actual que comienza en la cabeza.
* Siempre y cuando haya elementos, imprimirá el elemento con una flecha y avanza al siguiente.
* Al romper el ciclo, el programa indicará que la lista está vacía.

## Control de la Pila

Texto, Carta

El contenido generado por IA puede ser incorrecto.

Este usa la orden LIFO reutilizando la Lista Simple para almacenar elementos usando operaciones como push, pop y peek.

* Establece una nueva lista de la clase ListaEnlazada.
* Inserta un nuevo elemento en la pila.
* Expulsa y retorna el elemento de la pila.
* Consulta la cabecera sin alterar la pila.
* Indica si la pila está vacía.
* Mide la longitud de la pila.
* Muestra por consola todos los elementos de la pila de arriba hacia abajo.

## Control de la Cola

Texto

El contenido generado por IA puede ser incorrecto.

En esta parte, la lista deja de ser una lista y pasa de la orden LIFO a FIFO, también usando la lista simple, de modo que los elementos se encolan al final y se desencolan a la cabeza, como una cinta transportadora.

* Enfila el nuevo elemento.
* Desenfila y retorna el elemento que está al frente de la cola.
* Consulta el elemento al frente sin alterar la cola.
* Indica que la cola está vacía.
* Mide la longitud de la cola.
* Imprime a la salida todos los elementos de frente a final.

## Lista de Procesos

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Define un objecto inmutable que encapsula un identificador de proceso (PID) y un nombre, y ofrece una representación legible a través del método toString().

* PRIVATE FINAL INT PID almacena el identificador único del proceso (elemento que es inmutable).
* PRIVATE FINAL STRING NOMBRE contiene el descriptivo del proceso (también rígido).
* El constructor garantiza que ambos atributos tengan un valor inmutable, esto facilita clases seguras para hilos y evita estados inconsistentes.
* El método toString() traduce cualquier objeto a una cadena de texto, devolviendo un formato separado de la lógica de concatenación.

## Interruptor

Texto

El contenido generado por IA puede ser incorrecto.

* El atributo TIPO es una cadena que describe la naturaleza de la interrupción.
* El constructor recibe un parámetro TIPO y lo asigna al campo interno.

## Simulador

Texto

El contenido generado por IA puede ser incorrecto.

Este integra las estructuras de datos en un módulo de gestión de un OS simulado (nombrándolo como Super Disk Operating System, piensa más que DOS, piensa SuperDOS™). Cada instancia de SuperDOS mantiene:

* Una cola FIFO que almacena los procesos a ejecutarse.
* Una pila LIFO que guarda interrupciones en orden de llegada.
* Un contador para asignar un identificador único a cada nuevo proceso.

### Crear Proceso

Texto

El contenido generado por IA puede ser incorrecto.

Construye un nuevo Proceso, asignando un identificador y luego suma uno al siguiente PID. Inserta ese proceso al final de la cola de listos mediante el enfilado. Este simula cómo el OS recibe y encola nuevas tareas.

### Disparar Interrupción

Texto

El contenido generado por IA puede ser incorrecto.

Crea un nuevo objeto Interrupcion con el nombre o tipo recibido, empuja esa interrupción de la pila de atención inmediata. Este simula eventos en relación al hardware que el OS debe atender con prioridad.

### Ejecución del Ciclo

Texto

El contenido generado por IA puede ser incorrecto.

Este método emula el bucle principal de una agenda que alterna entre el manejo del hardware y del software:

* En caso de que haya interrupciones pendientes, extrae la última y atiende el primero.
* Si no hay interrupciones, pero sí procesos listos, desencola el siguiente proceso con dq() y simula su ejecución.
* Si no hay interrupciones ni procesos, el CPU se suspende.

### Estado Actual

Texto

El contenido generado por IA puede ser incorrecto.

Este observa el estado interno en cualquier momento y depurar la simulación paso a paso.

## Clase Principal

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

A continuación, este es un desglose de capa parte de la clase principal para ver el ejemplo que puede llegar a hacer SuperDOS:

* Se crea una nueva instancia DOS, en su constructor se inician la cola de procesos, la pila de interrupciones y el contador nextPid.
* Cada llamada genera un proceso con su nombre y PID únicos, los procesos se añaden al final de la cosa de listos en la siguiente orden: EDITOR 🡪 COMPIALDOR 🡪 NAVEGADOR.
* El método del estado actual muestra en consola el contenido de la cola y pila, esto ayuda a confirmar que los tres procesos están listos y que no hay interrupciones pendientes.
* Los disparos de interrupciones simulan dos eventos externos, en este caso el teclado y el temporizador. Las interrupciones se apilan y se ejecutan en la orden LIFO.
* El bucle se repite 6 veces:
  + Llama al método de ejecución del ciclo donde la computadora:
    - Atiende primero la última interrupción apilada.
    - Si no hay interrupciones, ejecuta el siguiente proceso de la cola con la orden FIFO.
    - Si no hay nada, la instancia se desactiva.

# Salida

Tabla

El contenido generado por IA puede ser incorrecto.

* Se asignan los procesos en la cola, en este caso Editor, Compilador y Navegador.
* Estos se organizan en la cola y se enlistan bajo la orden el primero en entrar, el primero en salir.
* Como la pila de interrupciones está vacía, se asignan los eventos del hardware para apilarlos bajo la orden del último en entrar, el primero en salir. Como el temporizador fue el último evento, será el primero en ser atendido.
* Una vez ejecutado el temporizador, la pila atiende el teclado, eventualmente estará vacía la pila otra vez.
* Regresando con la cola, como el editor fue el primero en entrar, será el primero en ser ejecutado, así será el orden hasta que quede vacío.
* Una vez vacías la cola y la pila, el sistema quedará inactivo, y así concluye el programa.