



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software.

Nombre de la Práctica Colas.

Numero de Práctica 14

Nombre de la carrera Ingeniería de Software

Nombre de la materia Lab. Estructuras de Datos

Nombre del alumno Jesús Manuel Juárez Pasillas

Nombre del docente Aldonso Becerra Sánchez

Fecha: 23/09/2021

Práctica 14: Colas.

Introducción:

Las colas de prioridad nos permiten almacenar datos de acuerdo a la prioridad que tengan estos. Con esto podemos ir agregando los datos, todos con una prioridad definida la cual les define una posición en la cola. Este tipo de colas lo que hace es que dependiendo la prioridad que tengan los datos, estos salen con forme a su prioridad, definiendo el tipo de prioridad.

Desarrollo:

Para desarrollar esta practica se hizo uso de la clase Arreglo en la cual se agregaban colas dentro del arreglo. Con esto cada posición del arreglo definia una prioridad en la cual se agregarían los datos a la cola en la posición indicado por su prioridad.

Se hicieron varias clases con las cuales se definia manipular estos datos y obtener una cola de prioridad.

Primero se creo una clase a la cual se le llamo Monticulo, a esta clase se le agrego un atributo de ArregloDatos el cual almacenara las colas y definira la posición de cada prioridad, también se agrego el atributo de orden el cual es un TipoOrden que define si la prioridad es ascendente o descendente.

Los datos que se van a estar almacenando serán de tipo Proceso, el cual almacena información general de lo que sería un proceso. Al crear esta clase se tuvo que crear un nuevo paquete dentro de registros en el que se creo la clase Proceso y TipoProceso.

Proceso: Esta clase simula ser un proceso, el cual contiene todos los atributos de un proceso, además de dar al usuario la posibilidad de escoger el tipo de proceso que se quiere.

- Da la opción al usuario de escoger el tipo de proceso para agregar la prioridad.
`private TipoProceso determinarTipo()`
- Obtiene el tipo de proceso que es.
`public TipoProceso getTipo()`

TipoProceso: Esta clase contiene todos los tipos que puede tener un proceso, junto a su prioridad, esta dependerá del orden cambia de ser 1 la prioridad más alta a ser 7.

- `SISTEMA(1)`
- `DESCARGA(2)`

- *GUARDADO(3)*
- *ABRIR(4)*
- *CERRAR(5)*
- *IMPRESION(6)*
- *COPIADO(7)*
- *PEGADO(8)*

En el paquete estructuraslineales se agregó la clase Monticulo y en ella se agregaron los siguientes métodos:

Monticulo: Esta clase contiene las operaciones para poner, quitar e imprimir según el ordenamiento, además de hacerlo por prioridad.

- *Agrega colas a la cola para simular las prioridades.*
private void ingresarPrioridad(int tam)
- *Verifica si es que todas las colas dentro del arreglo están vacías.*
public boolean vacio()
- *Verifica si es que todas las colas dentro del arreglo están llenas.*
public boolean lleno()
- *Agrega un objeto a la posición que le corresponde según la prioridad que tiene y el orden.*
public boolean poner(Object info)
- *Quita el objeto que sigue según la prioridad que tiene y el orden.*
public Object quitar()
- *Imprime los datos conforme al orden.*
public void imprimir()
- *Regresa el ultimo dato agregado según el orden y la prioridad.*
public Object verTope()

Nota: Toda la documentación esta agregada en la carpeta “doc” dentro de la carpeta del proyecto (“edylab_2021_14/doc”).

Capturas del programa funcionando:

```

Ingresa el numero de procesos a ingresar: 10
Escoge la prioridad del Proceso p1
1.- SISTEMA, prioridad-> 1
2.- DESCARGA, prioridad-> 2
3.- GUARDADO, prioridad-> 3
4.- ABRIR, prioridad-> 4
5.- CERRAR, prioridad-> 5
6.- IMPRESION, prioridad-> 6
7.- COPIADO, prioridad-> 7
8.- PEGADO, prioridad-> 8
1
Escoge la prioridad del Proceso p2
1.- SISTEMA, prioridad-> 1
2.- DESCARGA, prioridad-> 2
3.- GUARDADO, prioridad-> 3
4.- ABRIR, prioridad-> 4
5.- CERRAR, prioridad-> 5
6.- IMPRESION, prioridad-> 6
7.- COPIADO, prioridad-> 7
8.- PEGADO, prioridad-> 8
3
Escoge la prioridad del Proceso p3
1.- SISTEMA, prioridad-> 1
2.- DESCARGA, prioridad-> 2
3.- GUARDADO, prioridad-> 3
4.- ABRIR, prioridad-> 4

```

```

Escoge la prioridad del Proceso p10
1.- SISTEMA, prioridad-> 1
2.- DESCARGA, prioridad-> 2
3.- GUARDADO, prioridad-> 3
4.- ABRIR, prioridad-> 4
5.- CERRAR, prioridad-> 5
6.- IMPRESION, prioridad-> 6
7.- COPIADO, prioridad-> 7
8.- PEGADO, prioridad-> 8
8
Impresion del monticulo con todos los datos agregados
Prioridad 8:

Prioridad 7:

Prioridad 6:
Proceso:
    nombre = p6
    comando = c6
    archivo = a6
    ruta = r6
    propietario = p6
    tipo = prioridad: 6

```

```

Prioridad 5:
Proceso:
    nombre = p8
    comando = c8
    archivo = a8
    ruta = r8
    propietario = p8
    tipo = prioridad: 5
Proceso:
    nombre = p10
    comando = c10
    archivo = a10
    ruta = r10
    propietario = p10
    tipo = prioridad: 5
Prioridad 4:
Proceso:
    nombre = p4
    comando = c4
    archivo = a4
    ruta = r4
    propietario = p4
    tipo = prioridad: 4

```

```

Proceso:
    nombre = p7
    comando = c7
    archivo = a7
    ruta = r7
    propietario = p7
    tipo = prioridad: 4
Prioridad 3:
Proceso:
    nombre = p2
    comando = c2
    archivo = a2
    ruta = r2
    propietario = p2
    tipo = prioridad: 3
Proceso:
    nombre = p9
    comando = c9
    archivo = a9
    ruta = r9
    propietario = p9
    tipo = prioridad: 3

```

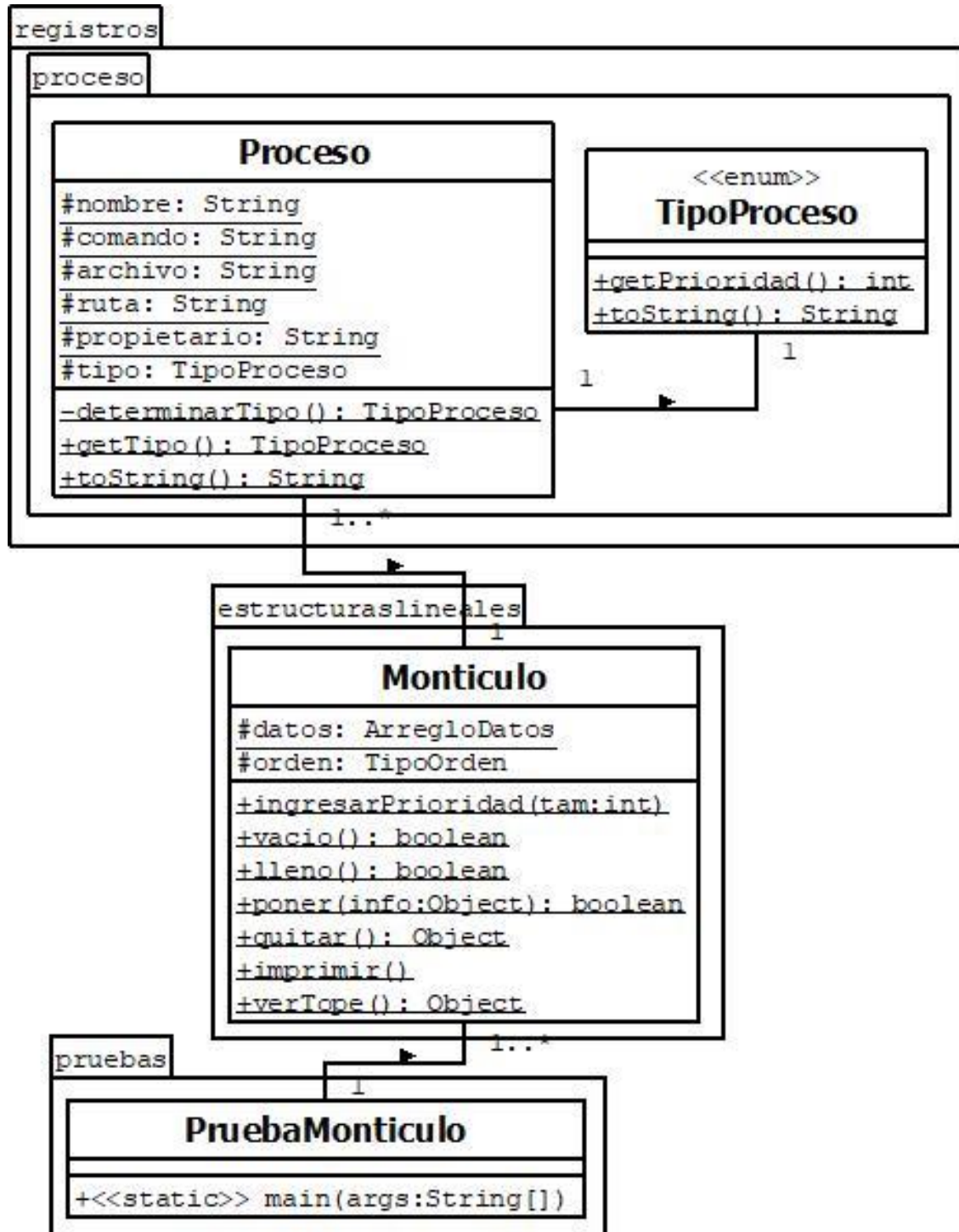
```

Prioridad 2:
Proceso:
    nombre = p3
    comando = c3
    archivo = a3
    ruta = r3
    propietario = p3
    tipo = prioridad: 2
Proceso:
    nombre = p5
    comando = c5
    archivo = a5
    ruta = r5
    propietario = p5
    tipo = prioridad: 2
Prioridad 1:
Proceso:
    nombre = p1
    comando = c1
    archivo = a1
    ruta = r1
    propietario = p1
    tipo = prioridad: 1

```

Código agregado:

Todas las clases, atributos y métodos que se muestran son nuevos, además del paquete proceso que se encuentra dentro del paquete registros.



Pre-evaluación:

Pre-Evaluación para prácticas de Laboratorio de Estructuras de Datos	PRE-EVALUACIÓN DEL ALUMNO
CUMPLE CON LA FUNCIONALIDAD SOLICITADA.	No
DISPONE DE CÓDIGO AUTO-DOCUMENTADO.	Sí
DISPONE DE CÓDIGO DOCUMENTADO A NIVEL DE CLASE Y MÉTODO.	Sí
DISPONE DE INDENTACIÓN CORRECTA.	Sí
CUMPLE LA POO.	Sí
DISPONE DE UNA FORMA FÁCIL DE UTILIZAR EL PROGRAMA PARA EL USUARIO.	Sí
DISPONE DE UN REPORTE CON FORMATO IDC.	Sí
LA INFORMACIÓN DEL REPORTE ESTÁ LIBRE DE ERRORES DE ORTOGRAFÍA.	Sí
SE ENTREGÓ EN TIEMPO Y FORMA LA PRÁCTICA.	No
INCLUYE LA DOCUMENTACIÓN GENERADA CON JAVADOC.	Sí
INCLUYE EL CÓDIGO AGREGADO EN FORMATO UML.	Sí
INCLUYE LAS CAPTURAS DE PANTALLA DEL PROGRAMA FUNCIONANDO.	Sí
LA PRÁCTICA ESTÁ TOTALMENTE REALIZADA (ESPECIFIQUE EL PORCENTAJE COMPLETADO).	80%
Observaciones:	

Conclusión:

Utilizar colas para ordenar los datos para que entren y salgan con un orden permite usarlas para una gran variedad de escenarios.

Las colas nos permiten guardar datos para posteriormente usarlos con un orden. Implementar este orden junto a una prioridad permite guardar datos con cierto tipo de prioridad, para posteriormente usarlos en forma mucho más ordenada, ya que un dato con más prioridad se usará primero que un dato con menos prioridad.