



# Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

---

## Práctica 14

### Datos generales:

Nombre de la Práctica	Colas
Nombre de la carrera	Ingeniería de Software
Nombre de la materia	Estructuras de Datos
Número y nombre de Unidad(es) temática(s)	II. Estructuras lineales.
Docente que imparte la materia	Aldonso Becerra Sánchez
Fecha de entrega para los alumnos	22-septiembre-2021
Fecha de entrega con extensión y penalización	23-septiembre-2021
Fecha de elaboración	22-septiembre-2021

Objetivo de la tarea	Familiarizarse con el uso de las colas y su funcionamiento
Tiempo aproximado de realización	2.5 horas
Introducción	En muchos ambientes es requerido realizar procesos donde se requiere ir almacenando elementos e irlos agregando a una cola de espera, la cual será procesada conforme llegaron.

### Referencias que debe consultar el alumno (si se requieren):

#### Referencia 1:

1.Cairo, Osvaldo; Guardati, Silvia. Estructura de Datos, Tercera Edición. McGraw-Hill, México, Tercera Edición, 2006.

#### Referencia 2:

2.Mark Allen Weiss. Estructura de datos en Java. Ed. Addison Wesley.



# Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

---

## Referencia 3:

3. Joyanes Aguilar, Luis. Fundamentos de Programación. Algoritmos y Estructuras de Datos. Tercera Edición, 2003. McGraw – Hill.

## Actividades que debe realizar el alumno:

### Actividad inicial:

Generar el reporte en formato IDC.

### Actividad 1:

Primero genere la **Introducción**.

### Actividad 2:

Defina un programa propietario que implemente la funcionalidad de una cola de prioridad usando montículos para representar las prioridades. Para esto se requiere analizar si se ocupa realizar herencia o no, además del tipo de implementación que seleccionaría para los montículos en arreglos.

Consideraciones:

- Usar un TDA montículo (arreglo para representar las prioridades únicas). Defina el programa de tal manera que permita considerar ascendente o descendente en el uso de prioridades. Defina o reutilice enumerados.
- Use un TDA separado que permita guardar los procesos (elementos a guardar en la cola).
- Debe definir un catálogo de tipos de procesos (Por ejemplo, procesos del sistema operativo, procesos de impresión, procesos de descarga, procesos de copiado, ...), donde a cada tipo le corresponderá una prioridad igual a su número de tipo. Cabe aclarar que de los procesos se ocupa información adicional como, nombre, comando o archivo ejecutado, ruta, propietario, etc. De esta manera cuando se van a agregando nuevos proceso (indicando su tipo, a través de un menú), se les asigna de manera automática su número de prioridad.
- Haga la cola de prioridad lo más genérica y reutilizable como sea posible.

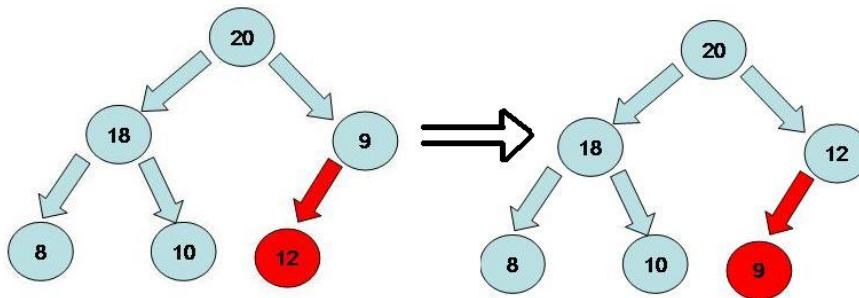


#### Resumen de montículos.

Para todo nodo del árbol se debe cumplir que su valor, sea mayor o igual que el valor de cualquiera de sus hijos (en el caso ascendente). Para representar un montículo en un arreglo lineal debe tenerse en cuenta para todo nodo K, lo siguiente:

- 1.- El nodo K se almacena en la posición K correspondiente del arreglo.
- 2.- El hijo izquierdo del nodo K se almacena en la posición  $2*K$  del arreglo.
- 3.- El hijo derecho del nodo K se almacena en la posición  $2*K + 1$  del arreglo.

**NOTA:** No olvide adecuar el caso en que la posición de arreglos en Java inicia en 0.



Arreglo

20	18	9	8	10	
1	2	3	4	5	6

Posiciones de índices

20	18	12	8	10	9
1	2	3	4	5	6

#### INSERCIÓN DE UN ELEMENTO EN UN MONTÍCULO

La inserción de un elemento en un montículo se lleva a cabo de los siguientes pasos:

- 1.- Se inserta el elemento en la primera posición disponible.
- 2.- Se verifica si su valor es mayor que el de su padre, si se cumple esta condición entonces se efectúa el intercambio, si no se cumple, entonces el algoritmo se detiene y el elemento queda ubicado en su posición correcta en el montículo.

**NOTA:** Cabe aclarar que el paso 2 se aplica en forma repetitiva y de abajo hacia arriba.



# Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

---

## ELIMINACIÓN DE UN MONTÍCULO

El proceso para obtener los elementos ordenados se efectúa eliminando la raíz del montículo en forma repetida. Pasos:

- 1.- Se reemplaza la raíz con el elemento que ocupa la última posición del montículo.
- 2.- Se verifica si el valor de la raíz es menor que el valor más grande de sus hijos. Si se cumple la condición se efectúa el intercambio, si no se cumple, entonces el algoritmo se detiene y el elemento queda ubicado en su posición correcta en el montículo.

### Actividad 3:

Realice una simulación de una cola de procesos enviados a descarga, a ejecución en un sistema operativo y a impresión para su ejecución con el fin de verificar la implementación de su cola de prioridad.

Realice el **Desarrollo** para las actividades 2 y 3.

### Actividad 4:

Pruebe el funcionamiento del programa de la actividad 2 y 3 con todo y sus capturas de pantalla.

### Actividad 5:

Realice la sección de **Código agregado** (diagrama de clases UML).

### Actividad 6:

Realice la sección de **Pre-evaluación** (use los lineamientos establecidos).

### Actividad 7:

Finalmente haga las **Conclusiones**.

### Actividad 8:

Enviar en <http://ingsoftware.reduaz.mx/moodle>



# Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

---

**Archivo anexo que se requiere para esta tarea (opcional):**

**Dudas o comentarios:** [a7donso@gmail.com](mailto:a7donso@gmail.com)