

TC1031: Programación de estructuras de datos y algoritmos fundamentales

Dr. Esteban Castillo Juarez

ITESM, Campus Santa Fe



esteban.castillojz@tec.mx



**Tecnológico
de Monterrey**

Evidencia 1: estructuras de datos en C

- Como parte de la competencia “SEG0702A” dentro del modelo TEC 21, **se plantea que los estudiantes realicen los siguientes programas en lenguaje C en equipos de tres.**
- Dichos programas, servirán como evidencia para evaluar sus habilidades en la creación de estructuras de datos dinámicas considerando la siguiente descripción:

Competencia:

SEG0702 Evalúa diversas tecnologías de la información, con apertura en la búsqueda e implementación de alternativas relevantes en la transformación de la práctica profesional.

Evalúa diversas tecnologías relevantes para la transformación de su práctica profesional.

Es consciente de los conceptos relacionados a las tecnologías emergentes.

Es usuario de diferentes sistemas computacionales ligados a su profesión. Toma conciencia de la importancia de la transformación digital.

Para esto, aplica el razonamiento algorítmico al modelar situaciones reales ligadas a su actividad profesional.

Programa 1

Creación y uso de una lista circular

Implementar en lenguaje C una **lista circular doblemente ligada** con las herramientas aprendidas en clase para manejo de memoria dinámica.

El programa deberá incluir lo siguiente:

1. Una función para **añadir un nodo** a la lista circular.
2. Una función para **añadir múltiples nodos consecutivos** a la lista circular.
3. Una función para **eliminar un nodo** de la lista circular.
4. Una función para **eliminar múltiples nodos** de la lista circular (todos los números repetidos de una lista circular).
5. Una función para **eliminar toda la lista circular**.

Programa 1

Creación y uso de una lista circular

Implementar en lenguaje C una **lista circular doblemente ligada** con las herramientas aprendidas en clase para manejo de memoria dinámica.

El programa deberá incluir lo siguiente:

6. Una función para **buscar un nodo** de la lista circular.
7. Una función para **saber si la lista circular esta vacía**.
8. Una función para **encontrar el numero de nodos** de la lista circular.
9. Otras funciones definidas por el equipo (se da libertad a los programadores de crear otras funciones que les sean útiles en la implementación de la lista circular).

Programa 1

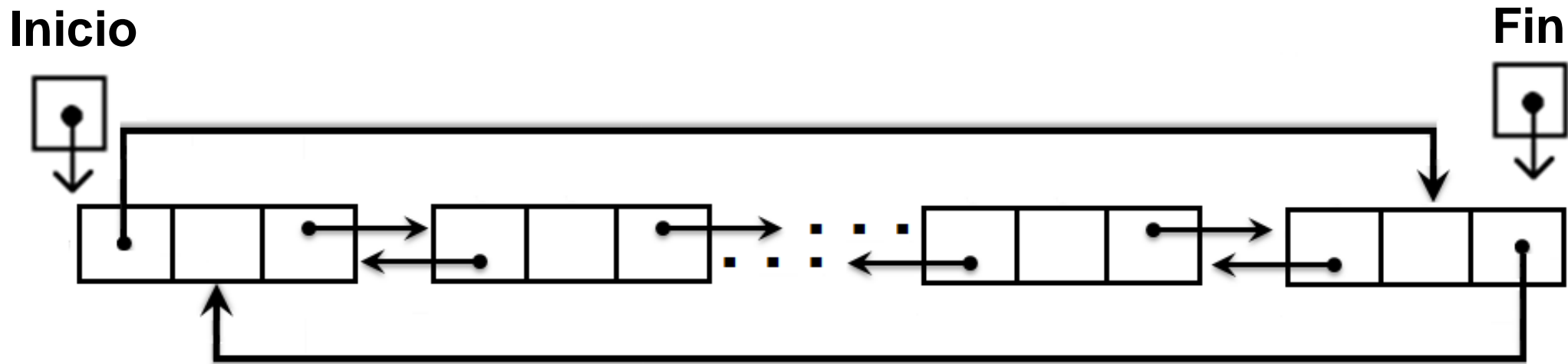
¿Que es una lista circular doblemente ligada?

- Una lista circular es una especie de lista doblemente ligada que posee una característica adicional asociada al desplazamiento: esta no tiene un fin natural.
- Para que la lista circular no tenga fin aparente, el puntero siguiente del último elemento apuntará hacia el primer elemento de la lista en lugar de apuntar al valor NULL así como el puntero anterior del primer elemento de la lista apuntara al ultimo elemento, lo cual genera una lista totalmente ligada.
- **Una lista circular doblemente ligada es una lista en la que el último nodo a punta al primero (y viceversa).**

Programa 1

¿Que es una lista circular doblemente ligada?

En las listas circulares doblemente ligadas, nunca se llega a una posición en la que ya no sea posible desplazarse. Cuando se llegue al último elemento, el desplazamiento volverá a comenzar desde el primer elemento.



Programa 1

En las listas circulares dociblemente ligadas

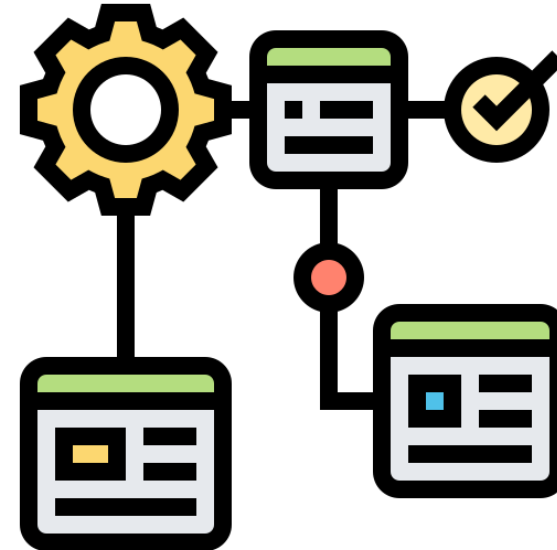
- No existe ningún nodo que apunte a NULL.
- El nodo inicio y fin pueden ser ajustados de acuerdo a las necesidades del programador, lo cual se traduce en escoger desde donde se “empieza” y desde donde se “termina” un **recorrido** completo de todos los elementos de la estructura de datos.
- La lista no tiene fin natural ya que al llegar al último nodo empieza de nuevo la lista.
- Se accede a la lista mediante el primer nodo o el ultimo.
- Si no se tiene cuidado al manejar la lista circular se pueden crear bucles infinitos.

Programa 1

Las listas circulares doblemente ligadas

Son como las listas revisadas hasta ahora en cuanto a las operaciones que se pueden realizar sobre ellas:

- Inicializar la lista.
- Añadir o insertar nodos en la lista.
- Borrar nodos de la lista.
- Moverse a través de la lista.
- Obtener la longitud de la lista.
- Verificar si la lista esta vacía.
- Buscar elementos dentro de la lista.



Programa 2

Creación y uso de una cola/queue

Implementar en lenguaje C una **cola/queue con prioridades** utilizando las herramientas aprendidas en clase para manejo de memoria dinámica.

El programa deberá incluir lo siguiente:

1. Una función para **añadir un nodo en la** cola/queue con prioridades.
2. Una función para **añadir múltiples nodos consecutivos** a la cola/queue con prioridades.
3. Una función para **eliminar un nodo** de la cola/queue con prioridades.
4. Una función para **eliminar toda la cola/queue con prioridades**.

Programa 2

Creación y uso de una cola/queue

Implementar en lenguaje C una **cola/queue con prioridades** utilizando las herramientas aprendidas en clase para manejo de memoria dinámica.

El programa deberá incluir lo siguiente:

5. Una función para **buscar un nodo** en la cola/queue con prioridades.
6. Una función para **saber si la cola/queue con prioridades esta vacía**.
7. Una función para **encontrar el numero de nodos** en la cola/queue con prioridades.
8. Otras funciones definidas por el equipo (se da libertad a los programadores de crear otras funciones que les sean útiles en la implementación de la cola/queue con prioridades).

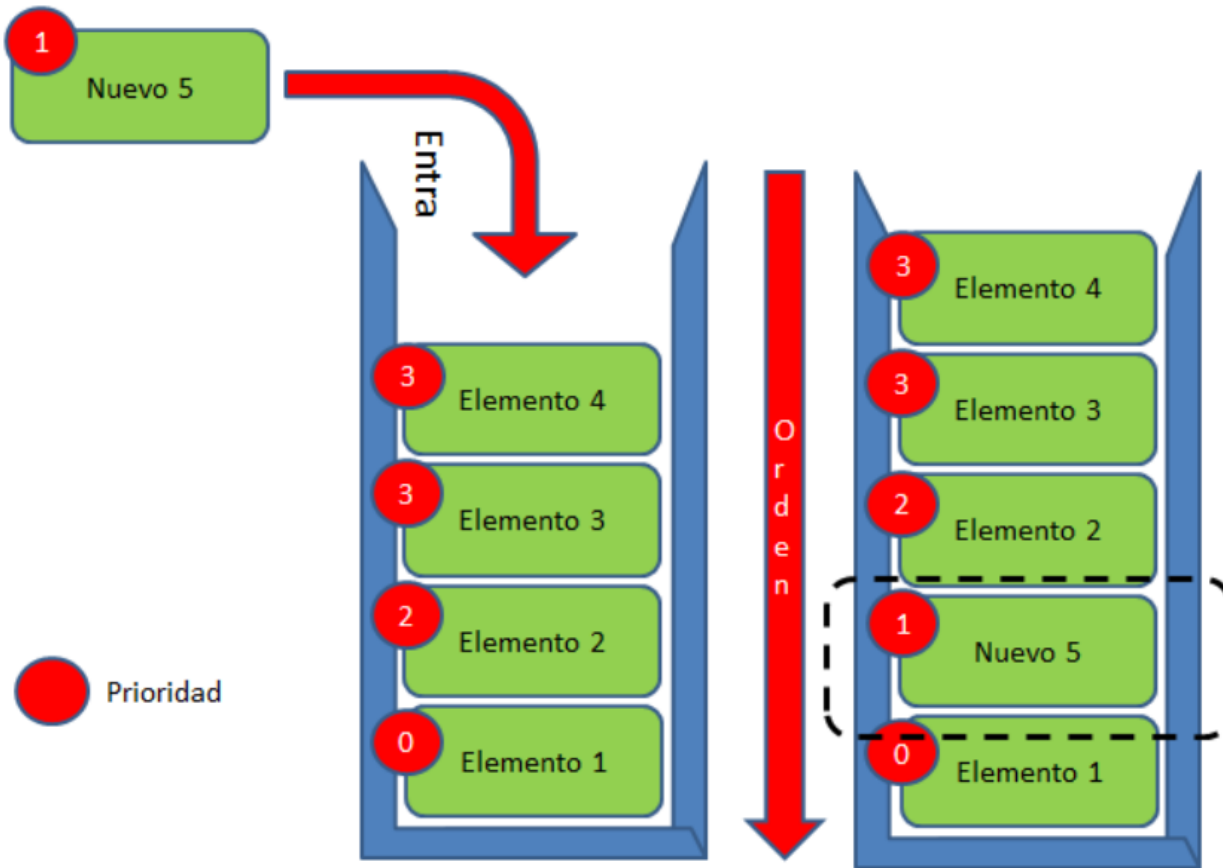
Programa 2

¿Que es una cola/queue con prioridades?

- Esta es una variante de la cola/queue convencional y trabaja muy parecido a la anterior sin embargo esta tiene un característica que la hace única y es que cuando un nuevo elemento entra en la cola no siempre se coloca al final.
- las colas/queues con prioridad tienen un mecanismo (puede ser aleatorio o de acuerdo a la lógica de un negocio, por ejemplo un banco con clientes regulares y platino) para determinar la prioridad de cada elemento de la cola, cuando un elemento entra en la cola lo primero que hace es ver que prioridad tiene y luego lo inserta en la posición que le corresponde.

Programa 2

¿Cómo funciona una cola/queue con prioridades?

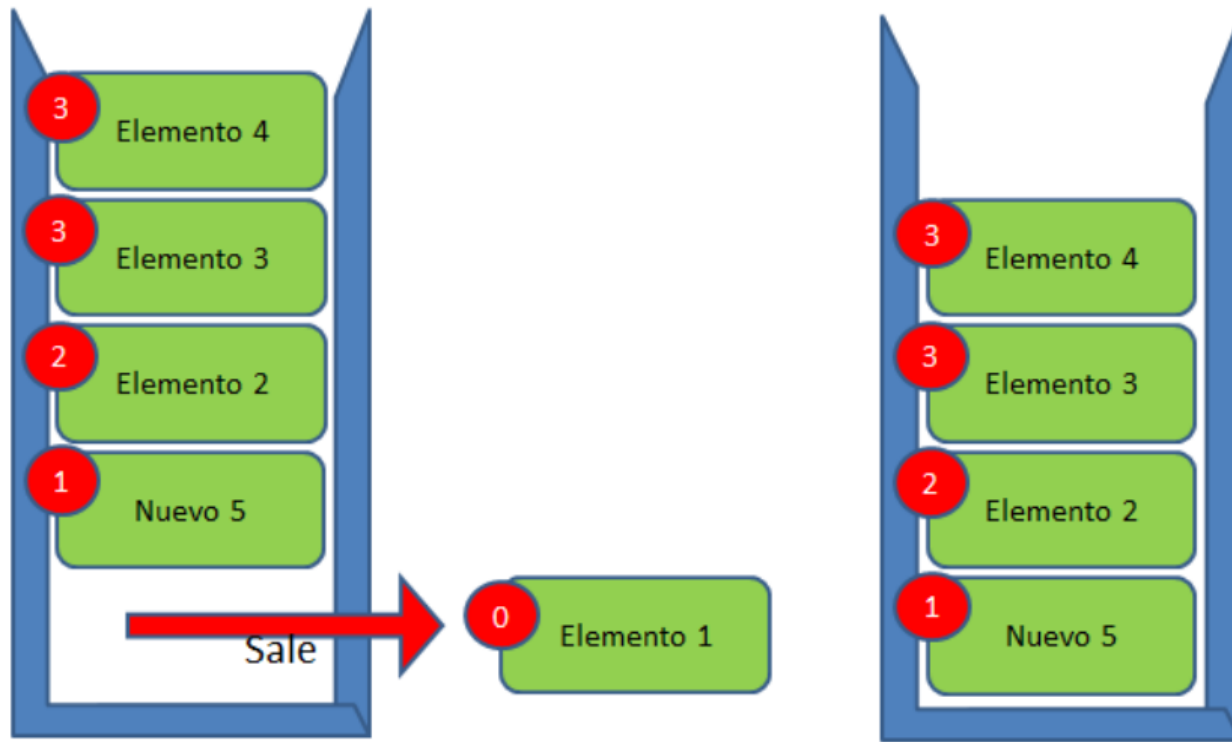


Insertar un nodo

- Como podemos apreciar en la imagen, un elemento con prioridad 1 es colocado en la cola y es puesto casi hasta a delante, sin embargo al tener prioridad 1 no pudo llegar hasta el principio ya que existía un elemento con mas prioridad (prioridad 0) en la Cola.
- Si nos damos cuenta de esta manera podemos priorizar la forma en que los elementos son colocados en la cola/queue.

Programa 2

¿Cómo funciona una cola/queue con prioridades?



extraer un nodo

- A partir de la imagen podemos apreciar como nodos con prioridad mas alta son sacados de la cola/queue primero (Izquierda) y como queda la cola/queue una vez que el nodo es retirado.
- Si prestamos atención la operación de retirar nodos de la cola/queue es exactamente igual que las colas/queues clásicas.

Formato de entrega

- En ambos programas muestre como se comportan todas las funciones solicitadas.
- Se deja a su discreción como mostrar las operaciones de inserción, borrado y elementos misceláneos (vacía, cantidad de elementos, etc.).
- También se deja a su discreción el tipo de datos que manejen las estructuras (int, float, char, etc.) pero todos los nodos deben ser del mismo tipo que elijan.
- Es importante destacar que después de la llamada de cada función (inserción, eliminación, etc.) deberá imprimir la estructura resultante o el valor regresado de dicha función (como en el caso de la longitud de las estructuras).

Formato de entrega

Poster argumentativo

Como parte de la evidencia, deberán adjuntar como documentación un poster argumentativo que resuelva las siguientes preguntas:

1. ¿Que son las estructuras de datos?
2. ¿Por qué son importantes las estructuras de datos?
3. ¿Qué tan relevantes son las lista circulares doblemente ligadas y colas/queues con prioridad?
4. ¿Cómo implementar en C (o cualquier lenguaje de programación) una lista circular doblemente ligada y cola/queue con prioridad? (mas explicación/imágenes que mostrar código)
5. ¿Cuál es su propuesta de solución? (mas explicación/imágenes que mostrar código)
6. ¿Cuál ha sido su experiencia hasta ahora con respecto a las estructuras de datos?

Formato de entrega

Poster argumentativo

El poster debe contener las siguientes secciones de forma mínima:

1. Introducción
2. Metodología
3. Implementación
4. Resultados
5. Conclusiones
6. Referencias
7. Otras secciones/subsecciones asociadas a las preguntas de la diapositiva anterior pueden estar contenidas en los elementos anteriores o pueden agregar alguna nueva sección, se deja a su discreción...

Formato de entrega

Poster argumentativo

El poster debe tener las siguientes características distintivas:

1. Debe tener un titulo (se deja a su discreción)
2. Nombre de los autores así como afiliación (ITESM...)
3. **El tamaño del poster debe ser A0 (841 X 1189 mm / 33,1 X 46,8 pulgadas).**
4. El formato del poster es electrónico (NO DEBEN IMPRIMIRLO). Su propósito es mas como documentación del código así como reflexión de la clase hasta ahora.
5. En la entrega final deberán adjuntar el poster en formato “.pdf”.

Formato de entrega

Poster argumentativo

Algunas recomendaciones para crear su poster;

1. El título tiene que ser descriptivo, pero al mismo tiempo ser capaz de captar el interés del lector.
2. El total del texto empleado en el póster no debería superar las 300 palabras.
3. Tiene que ser gráficamente atractivo.
4. La organización de los contenidos es fundamental, para que el contenido general del póster se entienda desde el primer golpe.
5. Deben diferenciar claramente los distintos apartados y ordena las ideas.

Formato de entrega

Poster argumentativo

Algunas recomendaciones para crear su poster;

6. Toda la información que pueda sustituirse con un recurso gráfico es mejor, plasmarlo con una imagen ayuda mas que un texto.
7. Todos los datos e información incluidos en el póster deben ser relevantes. Si no es relevante, no lo incluyas.
8. Muy atento a las faltas de ortografía, su presencia puede restarle credibilidad a tu investigación.
9. Asegúrate de que todos los datos que ofreces en el póster son consistentes.
- 10. Dale tu estilo propio y hazlo original y único.**

Formato de entrega

- La evidencia es en equipos de tres personas.
- **Se entregara la fecha indicada en la plataforma de CANVAS.**
- Deberán entregar/enviar un **único** archivo comprimido (.zip) con lo siguiente:
 - **Programas escritos en el lenguaje C** (no se admitirán otros formatos aparte de la extensión “.c”).
 - **Poster argumentativo (.pdf)** con explicación de los problemas a resolver así como una descripción de los pasos tomados (mas explicación que código) en la construcción de las soluciones.



Formato de entrega

- La evidencia contiene varios ejercicios-programas. Cada programa debe tener un nombre mnemónico alusivo al problema resuelto.
- NO olvidar colocar en el nombre del archivo comprimido lo siguiente: Evidencia1-"suMatriculaITESM".
- **Cada integrante del equipo deberá subir la evidencia a su portafolio individual en CANVAS.**
- **No habrá prorrogas ni cambios en la fecha de entrega o el formato.**



Suerte!

