



Universidad Nacional Autónoma de México  
Ingeniería en Computación  
Estructuras de datos

|                         |                                |
|-------------------------|--------------------------------|
| Asignación de Software. | Presentador fotográfico        |
| Responsable:            | M en C Ernesto Peñaloza Romero |

## Presentador Fotográfico

Existen un tipo especial de cola, llamado cola doble, en la cual los datos pueden entrar por ambos lados y salen por ambos lados.

Suponga que se dispone de una cámara fotográfica que al tomar una foto la añade automáticamente al visualizador de fotos en el dispositivo. Cuando el usuario pide ver las fotos inicia viendo primero desde el extremo por donde se insertan fotos y se mueve hacia las fotos más antiguas.

Al mismo tiempo que se toma la foto, estas se suben a la nube en donde los usuarios externos pueden ver las fotos iniciando por las primeras obtenidas y moviéndose hacia las fotos mas recientes.

Esta asignación consiste en la creacion de una cola doble que sea capaz de funcionar como se ha explicado anteriormente



El objetivo de esta asignación es la instrumentacion por medio de arreglos y/o una estructura enlazada e introducirlo en el uso de genericos e iteradores

## Especificaciones Técnicas:

La API de la clase ColaDoble es

```
public class ColaDoble <Item> implements Iterable<Item>
{
    public ColaDoble () //Construye una cola
    public boolean estaVacia() //¿Esta vacia la cola?
    public int tamanyo() //¿Cuántos elemento hay?
    public void insertaInicio(Item item) //Encolar un dato generico
```



**Universidad Nacional Autónoma de México**  
**Ingeniería en Computación**  
**Estructuras de datos**

---

```
public void insertaFinal(Item item) //Encolar un dato generico
public Item supprimeInicio() //Desencolar un dato generico
public Item supprimeFinal() //Desencolar un dato generico
public Iterator<Item> iterator() //retorna un iterador independiente
sobre los datos de la cola desde el frente hacia el final
public static void main(String[] args) //main de prueba
}
```

## Reglas de operación de Cola Doble

- Lance un NullPointerException si el cliente intenta añadir un item nulo
- Lance un java.util.NoSuchElementException si el cliente intenta ejecutar suprimir y no hay datos en la cola
- Lance un java.util.NoSuchElementException si el cliente llama el método next() en el iterador y no hay mas elementos que extraer.
- La implementación debe soportar cada operación de la cola doble (junto con la creación del iterador) en tiempo amortizado constante y usar espacio proporcional al número de items actualmente en la cola. Esto es, cualquier secuencia de M operaciones aleatorias (iniciando de una cola vacía) debe tomar como máximo M pasos en el peor de los casos, para alguna constante C.

## Reglas de operación del iterador

- La implementación del iterador debe soportar la construcción en tiempo lineal en el número de items y debe soportar las operaciones next() y hasNext() en tiempo constante en el peor de los casos
- Se debe usar una cantidad de memoria extra por iterador.
- El orden de dos o más iteradores a la misma cola debe ser mutuamente independiente. Cada iterador debe mantener su propio orden.

Su programa puede tener un main de prueba pero no será evaluado. En su lugar el evaluador automático proporcionará un main que generará una instancia de su programa y comenzará a someterlo a las diversas pruebas de inserción/supresión aleatoria, por ejemplo.

```
void EvaluadorPrueba1(Fotos [] arrFotos)
{
    ColaDoble cd<Fotos> = new ColaDoble();

    int i=0;
    while ( MiFoto: arrFotos.iterator())
    {
        switch (StdRandom.uniform(4))
        case: 0
            cd.insertaInicio(arrFotos[i++])
            break;
        case 1:
            cd.insertaFin(arrFotos[i++])
            break;
        case 2:
            cd.supprimeInicio()
            break;
```



Universidad Nacional Autónoma de México  
Ingeniería en Computación  
Estructuras de datos

```
        case 3:
            cd.suprimeFin()
            break;
    }
}
```

El siguiente segmento de código pertenece al evaluador. ColaDobleRef es la clase que contiene la solución y que se usa como referencia para evaluar sus resultados. ColaDoble se declara como **ColaDoble<Integer> cd**; Para probar, puede eliminar la declaración y use de ColaDobleRef y validar que no se disparan excepciones de referencia nula. Se muestra en azul lo que puede eliminar para probar su clase

```
System.out.println("Prueba 4: Llamadas random a insertaInicio y
suprimeInicio ");
//Verificar que esta vacia al inicio de todo
System.out.println("* 5 llamadas random (p1=0.9 p2=0.0 p3=0.1 p4=0.0)");
cd = new ColaDoble<>();
cdr = new ColaDobleRef<>();

contaPrueba=0;
fracaso = false;
for (k = 0; k < 5 && !fracaso; k++)
{
    try
    {
        contaPrueba++;
        if (StdRandom.uniforme(9) == 0)
        {
            int dataoa, datob;
            dataoa=cd.suprimeInicio();
            datob=cdr.suprimeInicio();
            if (dataoa != datob)
            {
                System.out.println("Fallo al remover con suprimeInicio
operacion "+contaPrueba);
                System.out.println("Estudiante valor: "+dataoa);
                System.out.println("Referencia valor: "+datob);
                fracaso=true;
            }
        }
        else
        {
            cd.insertaInicio(contaPrueba);
            cdr.insertaInicio(contaPrueba);
        }
    }
    catch(NoSuchElementException e)
    {}
    catch(Exception e)
    {
        fracaso = true;
    }
}
```



Universidad Nacional Autónoma de México  
Ingeniería en Computación  
Estructuras de datos

---

```
System.out.println (e.toString());  
}  
}  
  
if (fracaso)  
    System.out.println("==> Prueba fallada ");  
else  
    iPruebasPasadas++;
```

## Especificaciones de entrega

---

- El archivo zip deberá llamarse: EDD04-NumeroCuenta-NombreAlumnoSinEspacios.zip
- No se aceptan archivos rar. El evaluador solo admite archivos zip
- El zip se enviará únicamente al correo [ernestop@unam.mx](mailto:ernestop@unam.mx).
- Solo debe enviar el archivo ColaDoble.java. Este se empaquetará en el archivo zip y se enviará a la cuenta de correo indicada para ello. Este archivo deben venir en la raíz del zip y se deberán poder desempacar sin crear ningún tipo de subdirectorio, sino que se quedarán en el mismo directorio en donde se encuentra el zip.
- En el cuerpo del mensaje se deberá indicar claramente el siguiente sujeto: FESA
- En un archivo texto con el nombre CONCLUSION.TXT deberá venir su nombre en la primera línea y sin espacios en blanco, seguido de un salto de línea. Después pondrá las conclusiones de su proyecto. Este archivo se empaquetará en la raíz de su archivo zip
- La clase deberá venir incluida en un package llamado colaDoble. Debe respetar la minúscula inicial o el robot indicará que su clase no fue encontrada
- Si no se cubren las condiciones de entrega del zip o del correo, el evaluador no procesará su tarea y no será evaluada.
- Su asignación será evaluada automáticamente. En la sección de calificaciones de la plataforma se mostrará el resultado de su evaluación. Su retroalimentación se enviará a vuelta de correo. La calificación más alta deberá depositarse en classroom como evidencia. Sin ella, la evaluación no quedará asentada en firme
- Dependiendo de la carga en el servidor, la evaluación puede tardar varios minutos en ser calculada.
- El asunto en el correo debe ser FESA. Sin ningún otro texto adicional o el robot no podrá tomarlo
- 

## FAQ

---

- ¿Puedo entregar una API diferente? No, solo se evaluará la API que se ajuste estrictamente a la API mostrada
- ¿Puedo invocar otras librerías java además de las estándar? No, solo puede utilizar las librerías estándar porque se utilizará por el evaluador automático. No se garantiza la compilación si se usa cualquier otra librería.
- Requiero más métodos en las clases propuestas, ¿puedo renombrar las existentes o mandar una clase extra? No, si lo requiere, haga una clase privada anidada y privada y vuelva privados sus métodos. No cambie el nombre de los métodos de la API o sus argumentos o el orden de los argumentos.
- ¿Por qué el siguiente código para la creación de un arreglo genérico produce un error en tiempo de compilación cuando Item es un parámetro de tipo genérico?

```
Item[] a = new Item[1];
```

Java prohíbe la creación de arreglos de tipo genérico. En su lugar utilice

```
Item[] a = (Item[]) new Object[1];
```



**Universidad Nacional Autónoma de México**  
**Ingeniería en Computación**  
**Estructuras de datos**

---

Esto provocará un warning en Java que será tomado como algo normal.

- ¿Puede una clase anidada tener un constructor? Si
- ¿Qué debo suponer sobre las entradas de la cola? Que se trata de cualquier tipo de dato. El número de datos a introducir puede ser un número entre cero y el número de fotos que quepan en la memoria del dispositivo.
- ¿Cuál es el tiempo que se considera correcto para la clase

| Cola doble                | Requisito de tiempo          |
|---------------------------|------------------------------|
| operaciones no iterador   | Tiempo constante amortizado  |
| Construcción del iterador | Lineal en el número de items |

- ¿Cómo será evaluado mi programa? Será pasado por un evaluador automático, el cual realizará múltiples pruebas. El resultado de este proceso de evaluación se enviará a vuelta de correo.
- ¿En qué consistirán las pruebas? Se verificará entre otras cosas, que cada método se comporte adecuadamente, se solicitará la generación de varias colas, se verificará que se comporte correctamente el iterador. Se crearán varios iteradores sobre la misma cola y se verificará que cada uno sea independiente de los otros. Se crearán varias colas en paralelo y se verificará que los datos en las colas sean independientes unas de otras.
- ¿Puedo mandar mi programa varias veces? Si, las que considere necesarias
- ¿En qué consiste la evaluación de estilo? Conforme lo señalado, se pasará el programa por el evaluador de estilo y deberá venir limpio. En caso de que no sea así se eliminará un punto de la calificación obtenida. Vea documento de estilos depositado en el DV
- Si en alguna de las ocasiones mando un programa y en ocasión posterior mando el programa modificado y obtengo peor calificación, ¿cuál queda? La más alta.
- ¿Cuál es la escala: estilo 20%, corrección 50%, memoria: 10%, conclusiones 10%, velocidad: 10%
- ¿Qué significa corrección? Que el programa se ejecute adecuadamente, conforme lo que se ha especificado.
- ¿Qué puede hacer que falle en mi programa? Que no se elijan las estructuras de datos correctas (arreglos, listas, etc) para elaborar el programa. Sin una adecuada comprensión de los tiempos de acceso y utilizando erróneamente la estructura, el programa tendrá que ser desechado y ser reescrito desde el inicio.
- ¿Cuánto tiempo tengo para que se ejecute mi programa? el evaluador le dará 10 segundos como máximo por cada ejecución del programa
- ¿Qué sucede si el programa envía una excepción por referencia nula? El evaluador abortará la ejecución de su prueba y mandará el mensaje de excepción. El evaluador hace el mejor esfuerzo por continuar la ejecución de la prueba, pero si el fallo es muy grave e inesperado podría tirar el evaluador
- ¿Qué sucede si el programa aborta mi programa? Pueden suceder dos cosas, o bien no alcanza a generar un archivo con la evaluación, o bien logra generar una salida. En el primero caso se le regresará un correo con una leyenda indicando de manera genérica el posible problema, en el segundo caso se indicará posiblemente un "Tiempo máximo de ejecución alcanzado", en cuyo caso podría significar que tardó más de 10 segundos y el evaluador abortó la ejecución de manera controlada
- ¿Qué debo hacer si mi correo no tiene anexo un resultado? Significa que tu entrega tuvo algún problema grave y debe ser validado manualmente, razón por la cual deberás enviar tu trabajo a la dirección indicada en el correo de respuesta indicando el caso para ser revisado
- ¿Puedo mandar mi programa varias veces? Si.
- ¿Cuál es la escala?: estilo 20%, corrección 70%, conclusiones 10%
- ¿Qué sucede si el evaluador tarda más de 30 minutos en regresarme mi evaluación? Es posible que el evaluador encuentre situaciones que lo hagan caerse y dejar de funcionar. En caso de un tiempo largo sin



**Universidad Nacional Autónoma de México**  
**Ingeniería en Computación**  
**Estructuras de datos**

---

respuesta, favor de reportarlo para que se valide que esta arriba