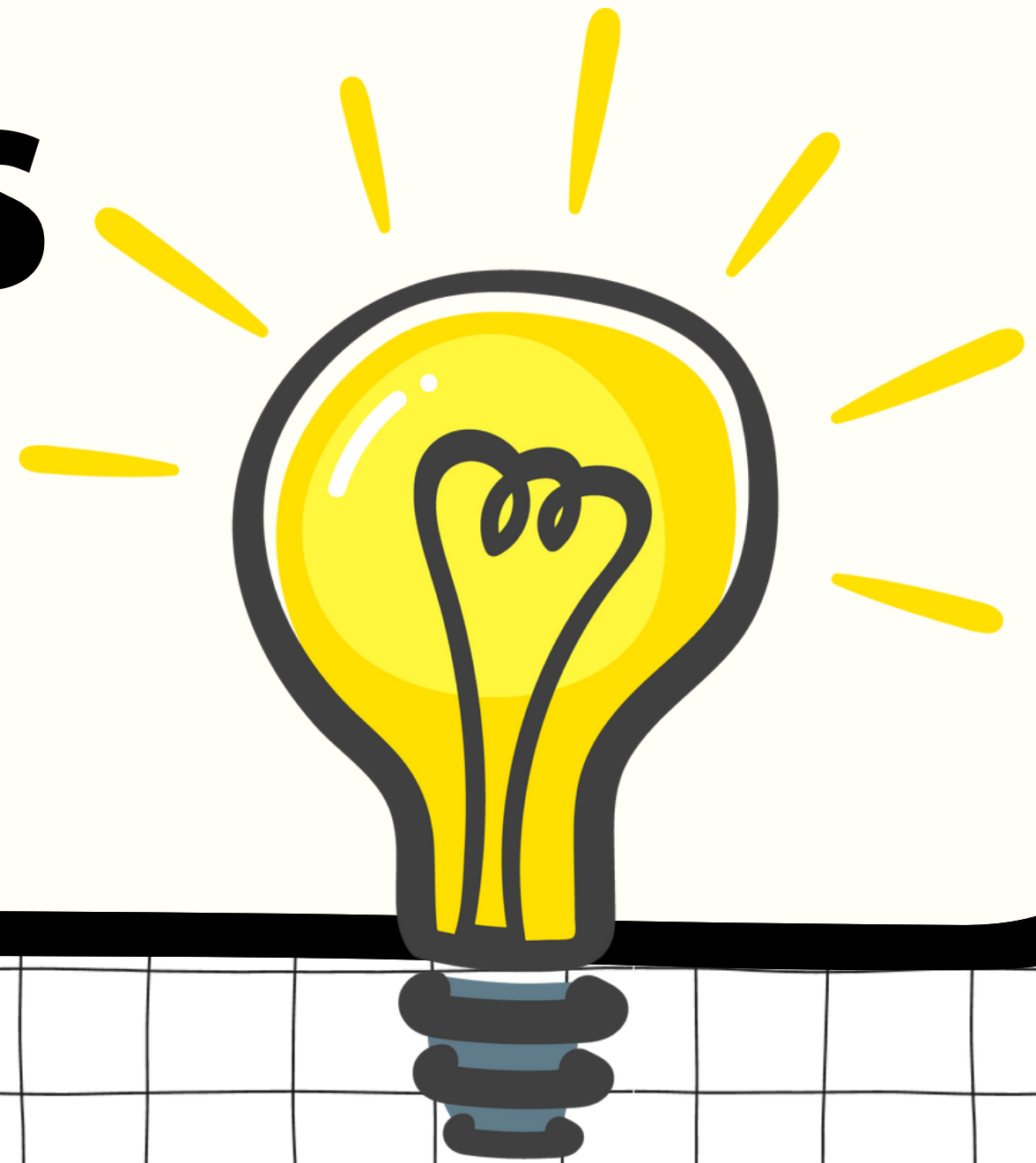




Estructuras de datos

Una aventura por StackOverflow



El equipo



**O. Isaías
Rodríguez**



R. Joaquín Pacab



Vacante
nos falta uno



Java

Lenguaje Seleccioando

Desde 1995, gran cantidad de frameworks y librerías, equipado con gran cantidad de estructuras de datos





Generalidades

librería

No usamos librerías

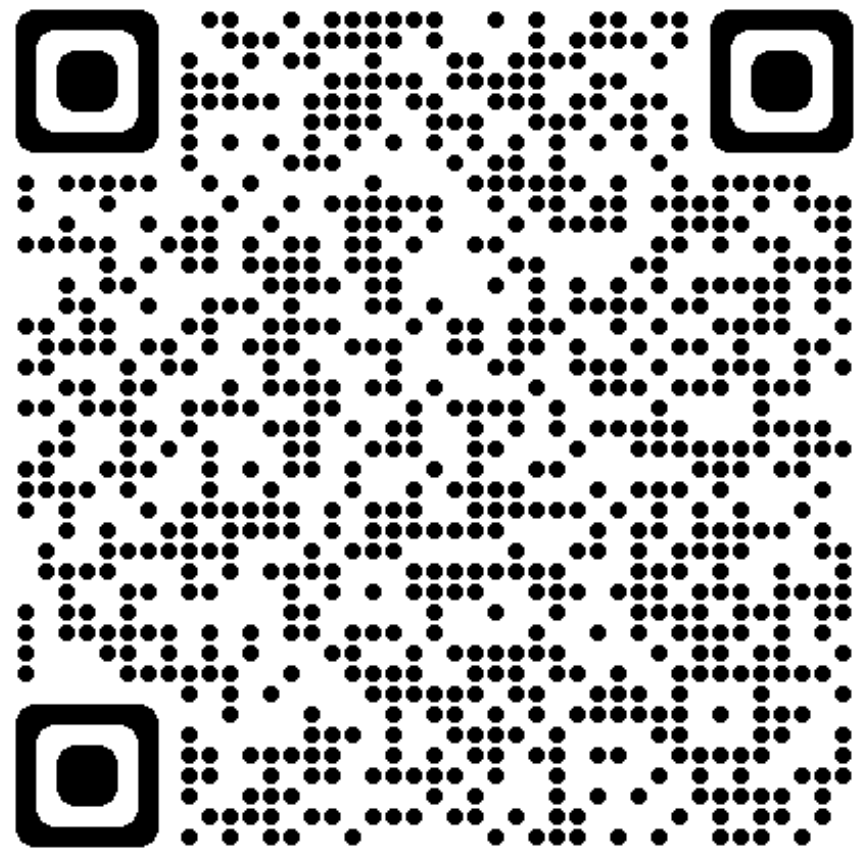
En la mayoría de los ejemplos no se usaron librerías, siquiera las propias de Java, por ejemplo, en `ArrayDeque`, es una clase propia creada que implementa las características de `Deque`.

Generalidades

librería

```
public interface DequeADT<T> {  
    // no la presenta en el código original  
    public void enqueueRear(T element);  
    // no la presenta en el código original  
    public void enqueueFront(T element);  
    // no la presenta en el código original  
    public T dequeueRear() throws  
        EmptyCollectionException;  
    // no la presenta en el código original  
    public T dequeueFront() throws  
        EmptyCollectionException;  
}
```


Method undefined for my type in a deque implementation in java



“

when I use some of the methods that were defined in the implementation of the deque, in my demonstration or driver program, I get an error saying the given method is undefined for the type that is my deque implementation class

**The method `enqueueRear(int)` is
undefined for the type
`ArrayDeque<Integer>Java(67108964)`**

**The method `dequeueRear()` in the type
`ArrayDeque<Integer>` is not applicable
for the arguments `(int)Java(67108979)`**



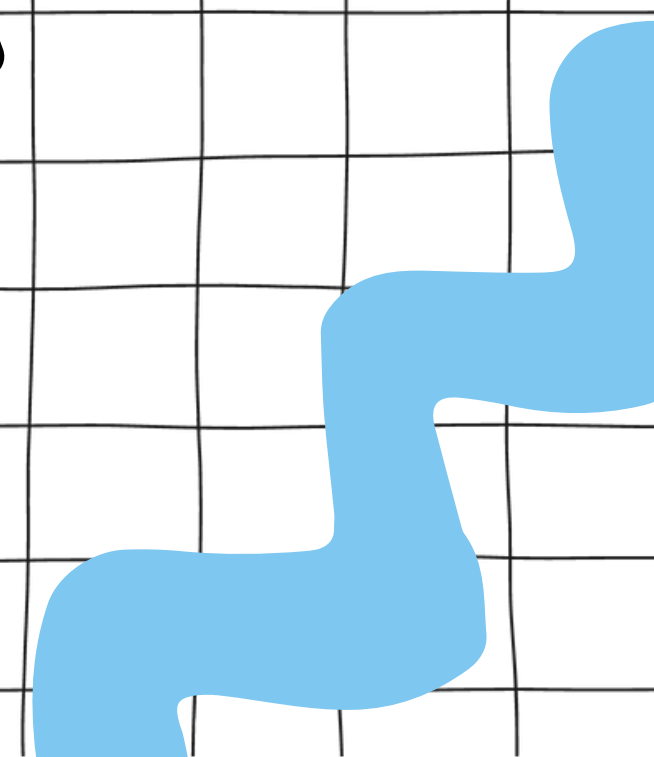
can't believe it

1 Remove `import java.util.ArrayDeque;` - you aren't testing your class `ArrayDeque`, you're invoking the one from `java.util` – [Elliott Frisch](#) May 8, 2017 at 23:06

on top of what @ElliottFrisch has mentioned, within your main method, when you invoke `dequeueRear` you're attempting to pass an argument to it whereas the signature of the method is --> `dequeueRear()` . – [Ousmane D.](#) May 8, 2017 at 23:10 ✎

@ElliottFrisch That worked! Thank you SO much :) – [Johnny Sack](#) May 8, 2017 at 23:15

@Aominè You're right, can't believe I didn't notice that. Thank you so much!! – [Johnny Sack](#) May 8, 2017 at 23:15



¿Qué estás quitando de la deque?

You are calling a method that exists but there is no parameter in the method declaration.
That method is

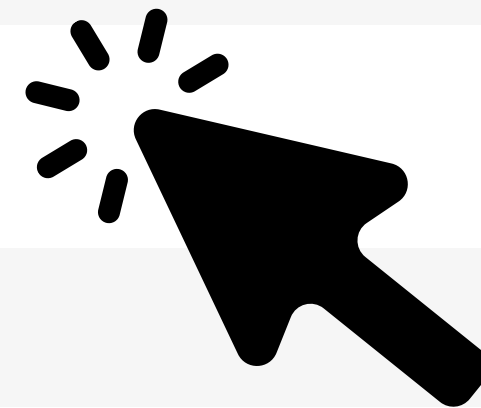
```
public T dequeueRear()
```

and you try to do the following

```
cad.dequeueRear(22);
```

You must do the following instead

```
cad.dequeueRear();
```



```
1 The method dequeueRear() in the type ArrayDeque<Integer> is not applicable  
2 for the arguments (int) Java(67108979)
```

```
Arra
```

```
Integer com.expo1.DequeueError.ArrayDeque.dequeueRear() throws  
/* A EmptyCollectionException
```

```
cad. View Problem (Alt+F8) Quick Fix... (Ctrl+.)
```

```
cad.dequeueRear(20);
```

You, 28 seconds ago • Uncommitted changes

Problema aplicativo

...

Problema: Un restaurante con acceso VIP

Descripción del problema: Supongamos que en un restaurante se tiene una cola de clientes, sin embargo algunos tienen acceso privilegiado mediante unas tarjetas VIP. En ese caso necesitamos una estructura que se comporte como una cola, pero que permita la inserción tanto al principio como al final. Para ello necesitamos una Deque.

Iterators and throwing exception

Code

```
import java.util.Iterator;
import java.util.NoSuchElementException;

public class Deque<Item> implements Iterable<Item>{

    private int size;

    private Node<Item> first;
    private Node<Item> last;

    private class Node<Item>{
        Item item;
        Node<Item> next;
        Node<Item> prev;

        Node(Item item) {
            this.item = item;
            next = null;
            prev = null;
        }
    }

    // construct an empty deque
    public Deque(){
        first = null;
        last = null;
        size = 0;
    }

    public boolean isEmpty(){return size == 0;} // is the deque empty?
    public int size(){return size;}           // return the number of items on

    // add the item to the front
    public void addFirst(Item item){
```

I am trying to make a double ended deque but I keep running into errors I frankly have no idea how to solve. The first one is regarding deque. In my iterator function, I keep getting the following error and I have no idea why:

```
Deque.java:105: error: incompatible types: Deque.DequeIterator cannot be converted to
    return new DequeIterator();
```

Additionally, I have been trying to throw exceptions but haven't been able to for some reason. I keep getting errors such as the following:

```
Deque.java:72: error: cannot find symbol
    throw java.util.NoSuchElementException();
              ^
symbol:   class util
location: package java
```

¿Qué está ocurriendo aquí?

**Deque.java:105: error: incompatible types: Deque.Dequelterator cannot be converted to Iterator<Item>
return new Dequelterator();**

Hay un problema de tipo incompatible en el código en la línea 105 de la clase Deque.java. El error indica que se está intentando devolver un objeto de tipo Dequelterator donde se espera un Iterator<Item>.

¿Qué está ocurriendo aquí?

Deque.java:72: error: cannot find symbol
throw java.util.NoSuchElementException;
 ^
symbol: class util
location: package java

Este error indica que el compilador no puede encontrar el símbolo util en el paquete java. Cuando se intenta lanzar una excepción de tipo NoSuchElementException, el compilador espera que esta clase se encuentre en el paquete java.util, pero parece que no está siendo importada o reconocida correctamente.

Código erroneo

```
public Iterator<Item> iterator() {  
    return new DequeIterator();    Type mismatch: cannot convert from Deque.DequeIterator to Iterator<Item>  
}  
  
private class DequeIterator<Item> implements Iterable<Item> {    The type Deque<Item>.DequeIterator<Item> must implement the inherited abstract method Iterable<Item>.iterator()  
    private Node current;    Deque.Node is a raw type. References to generic type Deque<Item>.Node<Item> should be parameterized  
  
    public DequeIterator() {  
        this.current = first;  
    }  
  
    public boolean hasNext() {
```

Su primer problema es que su clase `DequeIterator` implementa `Iterable`, cuando debería implementar `Iterator`. `Iterable` generalmente se usa para cosas como colecciones, que luego pueden proporcionar una instancia de `Iterator`. Parece que ya implementaste los métodos para `Iterator`, por lo que esto debería ser solo una cuestión de cambiar la línea para que sea:

```
private class DequeIterator<Item> implements Iterator<Item> {
```


Código erróneo

```
// remove and return the item from the front
public Item removeFirst() {
    if (this.isEmpty()) {
        // linea erronea
        throw java.util.NoSuchElementException(); java.util cannot be resolved to a type
        // linea correcta
        // throw new java.util.NoSuchElementException();
    } else {
        Item item = first.item;
        first = first.next;
        first.prev = null;
        if (size == 1) {
            first = last;
        }
        size--;
        return item;
    }
}

// remove and return the item from the end
public Item removeLast() {
    if (this.isEmpty()) {
        // linea erronea
        throw java.util.NoSuchElementException(); java.util cannot be resolved to a type
        // linea correcta
        // throw new java.util.NoSuchElementException();
    }
}
```

Para su segundo problema, le falta la nueva palabra clave para construir la excepción. Debería ser el siguiente:

```
throw new java.util.NoSuchElementException();
```


Problema aplicativo

...

Problema: Implementar un programa que simule el proceso de compra en una tienda en línea, utilizando un deque para administrar los productos en el carrito de compras.

Descripción del problema: Supongamos que estás construyendo una aplicación para una tienda en línea. Quieres permitir a los usuarios agregar productos a su carrito de compras y luego proceder con el pago. Sin embargo, quieres implementar la funcionalidad de manera que los usuarios puedan agregar productos tanto al principio como al final del carrito, y también puedan eliminar el primer o último producto agregado, siguiendo el principio de una cola doblemente terminada (deque).



Veamos el código para solucionarlo

