



IMPLEMENTACIONES

Iterador y Cadena de
Responsabilidad

ITERADOR

```
1      package model.IteratorPattern;
2
3  ✓    public interface Iterator {
4          public boolean hasNext();
5
6          public Object next();
7
8      }
```

```
1      package model.IteratorPattern;
2
3      public interface Container {
4          public Iterator getIterator();
5      }
```

```
1  package model.IteratorPattern;
2
3  ✓ public class NameRepository implements Container{
4      public String[] names={"Robert","John","Julie","Lora"};
5
6      @Override
7      public Iterator getIterator() {
8          return new NameIterator();
9      }
10  ✓ private class NameIterator implements Iterator {
11      int index;
12  ✓ public boolean hasNext(){
13      if(index < names.length){
14          return true;
15      }
16      return false;
17  }
18
19      @Override
20  ✓ public Object next() {
21      if(this.hasNext()){
22          return names[index++];
23      }
24      return null;
25  }
26
27  }
28  }
```

CADENA DE RESPONSABILIDAD

```
1  package model.ChainResponsabilityPattern;
2
3  ✓ public abstract class AbstractLogger {
4      public static int INFO = 1;
5      public static int DEBUG = 2;
6      public static int ERROR = 3;
7      protected int level;
8
9      //next element in chain or responsibility
10     protected AbstractLogger nextLogger;
11
12     public void setNextLogger(AbstractLogger nextLogger){
13         this.nextLogger = nextLogger;
14     }
15     ✓ public void logMessage(int level, String message){
16         if(this.level <= level){
17             write(message);
18         }
19         if(nextLogger !=null){
20             nextLogger.logMessage(level, message);
21         }
22     }
23     abstract protected void write(String message);
24 }
```

```
1    package model.ChainResponsabilityPattern;
2
3    ✓ public class ConsoleLogger extends AbstractLogger {
4        public ConsoleLogger(int level){
5            this.level = level;
6        }
7        @Override
8        protected void write(String message) {
9            System.out.println("Standard Console::Logger: " + message);
10       }
11    }
```



```
1    package model.ChainResponsabilityPattern;
2
3    ✓ public class ErrorLogger extends AbstractLogger {
4        public ErrorLogger(int level){
5            this.level = level;
6        }
7        @Override
8        protected void write(String message) {
9            System.out.println("Error Console::Logger: " + message);
10       }
11    }
```

```
1      package model.ChainResponsabilityPattern;
2
3  ✓   public class FileLogger extends AbstractLogger{
4          public FileLogger(int level){
5              this.level = level;
6          }
7          @Override
8          protected void write(String message) {
9              System.out.println("File::Logger: " + message);
10         }
11     }
```