

8.A. Excepciones.

1. Excepciones.

1.2. El manejo de excepciones.

Como hemos comentado, **siempre debemos controlar las excepciones que se puedan producir o de lo contrario nuestro software quedará expuesto a fallos**. Las excepciones pueden tratarse de **dos formas**:

- **Interrupción.** En este caso se asume que el programa ha encontrado un error irreparable. La operación que dio lugar a la excepción se anula y se entiende que no hay manera de regresar al código que provocó la excepción. Es decir, la operación que originó el error, se anula.
- **Reanudación.** Se puede manejar el error y regresar de nuevo al código que provocó el error.

Java emplea la primera forma, pero puede simularse la segunda mediante la utilización de un bloque `try` en el interior de un `while`, que se repetirá hasta que el error deje de existir. En la siguiente imagen tienes un ejemplo de cómo llevar a cabo esta simulación.

```
7 public static void main(String[] args){
8     boolean fueradelimites=true;
9     int i; //Entero que tomará valores aleatorios de 0 a 9
10    String texto[] = {"uno","dos","tre","cuatro","cinco"}; //String que representa la moneda
11
12    while(fueradelimites){
13        try{
14            i= (int) Math.round(Math.random()*9); //Generamos un indice aleatorio
15            System.out.println(texto[i]);
16            fueradelimites=false;
17        }catch(ArrayIndexOutOfBoundsException exc){
18            System.out.println("Fallo en el indice");
19        }
20    }
21 }
22
23 }
```

En este ejemplo, a través de la función de generación de números aleatorios se obtiene el valor del índice `i`. Con dicho valor se accede a una posición del array que contiene cinco cadenas de caracteres. Este acceso, a veces puede generar un error del tipo `ArrayIndexOutOfBoundsException`, que debemos gestionar a través de un `catch`. Al estar el bloque `catch` dentro de un `while`, se seguirá intentando el acceso hasta que no haya error.