

Para saber mais: Um pouco de java.util.Random

Muitas aplicações, principalmente jogos, usam números aleatórios. Esses números também podem ser úteis para criar cenários para testes. No nosso exemplo, os números aleatórios foram úteis para preencher as vendas com uma quantidade!

No mundo Java temos uma classe que possui a responsabilidade de criar números pseudoaleatórios, o `java.util.Random`:

<https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>
(<https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>)

Nesse exercício vamos ver rapidamente as principais funcionalidades do `Random`. Para testar crie uma classe simples com um método `main`:

```
package br.com.caelum.livraria.teste;

import java.util.Random;

public class TesteRandom {

    public static void main(String[] args) {

        Random gerador = new Random();

        int resultado = gerador.nextInt(10);
        System.out.println(resultado);

        int resultado2 = gerador.nextInt(10);
        System.out.println(resultado2);
    }
}
```

Execute algumas vezes essa classe e veja que aparecem os números entre 0 e 9 aleatoriamente.

Com essa classe não podemos apenas gerar inteiros como também valores do tipo double, long ou boolean, por exemplo:

```
Random geradorBoolean = new Random();

boolean valor = geradorBoolean.nextBoolean();
System.out.println(valor);

boolean valor2 = geradorBoolean.nextBoolean();
System.out.println(valor2);
```

Agora para que usamos esse *seed*? O *seed* ou *semente* alimenta o gerador com um valor inicial, e a partir desse valor serão gerados os próximos valores. Se a gente passa sempre o mesmo *seed* serão gerados os mesmos valores!! Teste

isso, por exemplo:

```
Random geradorBoolean = new Random(10); //seed inicial é 10  
  
boolean valor = geradorBoolean.nextBoolean(); //sempre true  
System.out.println(valor);  
  
boolean valor2 = geradorBoolean.nextBoolean(); //sempre false  
System.out.println(valor2);
```

Agora pensando contrário, se a gente passa sempre um *seed* diferente, serão gerados sempre valores diferentes, correto? Normalmente, você usa como *seed* os millisegundos do sistema, por exemplo:

```
long millis = System.currentTimeMillis();  
Random geradorBoolean = new Random(millis);  
  
boolean valor = geradorBoolean.nextBoolean();  
System.out.println(valor);  
  
boolean valor2 = geradorBoolean.nextBoolean();  
System.out.println(valor2);
```

Entendendo o *seed* podemos então gerar sequências de valores iguais ou diferentes!! Tudo bem?