

# Linguagem de Programação

Números Aleatórios

**ECT2303**

helton.maia@ufrn.br

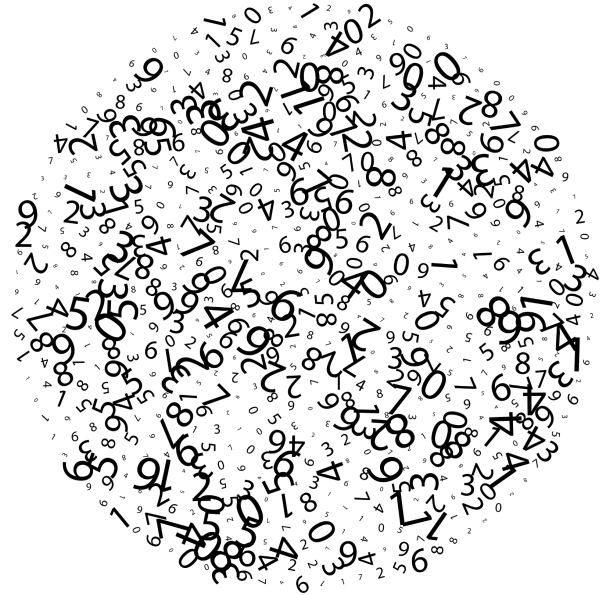
*What is Random ?*



# Números Aleatórios

Considerando a **aleatoriedade** como a ausência de padrões ou previsibilidade, temos a **geração aleatória de números** como um dispositivo que gera uma sequência de números ou símbolos, não podendo ser razoavelmente prevista melhor do que por uma chance aleatória.

Na verdade, a maioria dos programas de computadores são determinísticos, desta forma, quando necessitam, utilizam uma geração pseudo-aleatória de números.

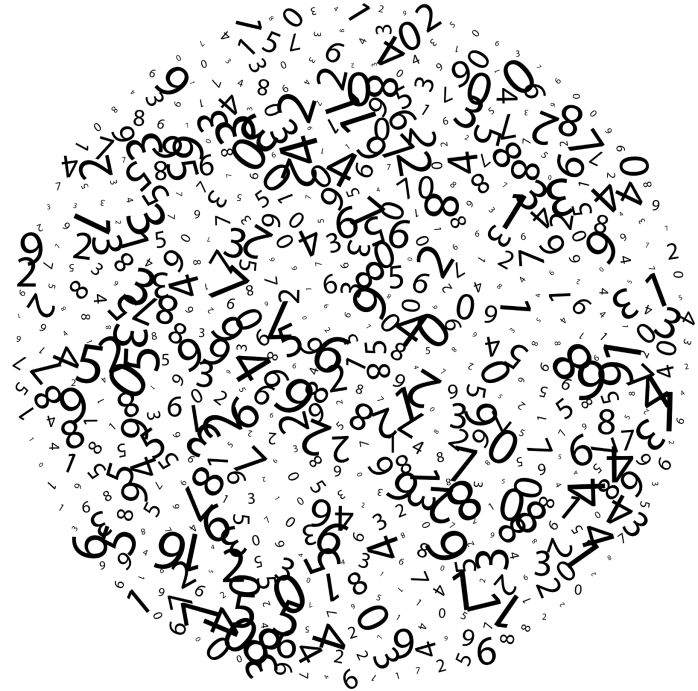


[Random.org](https://www.random.org)

# Números Aleatórios

## Exemplos de utilização:

Principalmente para testes de algoritmos, simuladores, segurança e criptografia, tomadas de decisão, jogos, loterias, inteligência artificial, etc.



# Geração de Números Aleatórios

Em C/C++: Utilize a biblioteca [cstdlib](#)

```
i = rand();
```

- A função **rand** retorna um número inteiro e pseudo-aleatório (um número que não é verdadeiramente aleatório, mas parece ser) entre 0 e RAND\_MAX;
- RAND\_MAX é uma macro da biblioteca. Um inteiro de 16 bits;

# Números Pseudo-Aleatórios

## Exemplo:

Criando uma sequência de cinco números pseudo-aleatórios. Execute mais de uma vez e compare os resultados.

```
1  #include <iostream>
2  #include <cstdlib>
3
4  using namespace std;
5
6  int main(){
7      //example 1
8      for(int i = 0; i<5; i++)
9          cout << rand() << " - ";
10
11      return 0;
12  }
13
```

# Escala e Deslocamento

Pode-se utilizar o operador de módulo “%” para auxiliar a função rand, por exemplo:

```
i = d + rand() % (dado)
```

deslocamento      escala

Exemplo:

```
dado = 1 + rand() % 6;
```

```
// v1 dentro da faixa 0 to 99
```

```
v1 = rand() % 100;
```

```
// v2 dentro da faixa 1 to 100
```

```
v2 = 1 + rand() % 100;
```

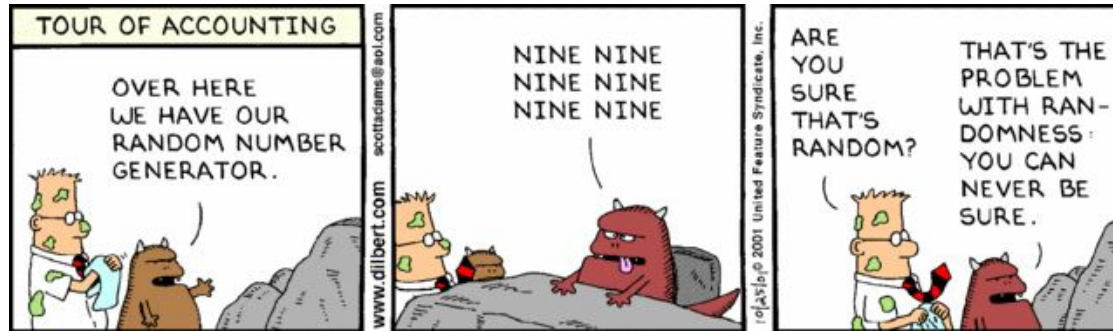
```
// v3 dentro da faixa 1985-2014
```

```
v3 = 1985 + rand() % 30;
```

Criando uma sequência pseudo-aleatória dentro de faixas específicas.

# Mudando de **rand** para **srand**

- Já foi dito que o rand, gera uma sequência pseudo-aleatória, sendo esta sequência calculada a partir de um parâmetro inicial, chamado de **semente**;
- Esta sequência vai ser repetida, a não ser que seja alterada a semente.





# Mudando de **rand** para **srand**

- Para que o gerador de números seja realmente aleatório (randômicos), deve-se utilizar a função `srand`, que semeia a função `rand` para produzir uma sequência diferente;
- Caso a semente não seja informada, o sistema adota o valor 0.

```
void srand(int seed)
```

# Produzindo uma sequência aleatória com o srand

A função `time` da biblioteca `ctime` utiliza o relógio do computador para obter o número de segundos que se passaram desde a meia-noite de primeiro de janeiro de 1970.

```
srand(time(NULL));
```

# Produzindo uma sequência aleatória com o srand

## Exemplo:

```
#include <iostream>
#include <cstdlib>
Using namespace std;

int main(){
//Usando tempo atual para gerar a semente
    srand(time(NULL));
    for(int i = 0; i<5; i++)
        cout << rand() << " ";
    return 0;
}
```

# Exercício de aprendizagem

Crie um programa que simula o lançamento de uma moeda, ( $n > 200$ ) vezes. Exiba na saída o número de vezes que cada um dos lados foi lançado.

?