

# Algoritmos e Estruturas de Dados I

## IEC012

## Linguagem C

### - Funções Matemáticas -

**Prof. Eduardo Souto**

esouto@dcc.ufam.edu.br

Slides preparados pelo Prof. Leandro Galvão  
galvao@dcc.ufam.edu.br

# Valor absoluto

- ❖ **abs**: Valor absoluto de um número **inteiro**.
- ❖ Definida na biblioteca **stdlib.h**

```
abs (<número> ) ;
```

```
abs (-4) ;
```

```
abs (4) ;
```

```
abs (num) ;
```

```
abs (num/2) ;
```

```
abs (-num) ;
```

# Valor absoluto

- ❖ **fabs**: Valor absoluto de um número **real**.
- ❖ Definida na biblioteca **math.h**

```
fabs (<número real>);
```

```
fabs (-4.123);  
fabs (4./9.);  
fabs (num);  
fabs (num/.2);  
fabs (-num);
```

# Potenciação

- ❖ **pow**: operação de potenciação.
- ❖ **exp**: número neperiano elevado ao argumento.
- ❖ Resultado é sempre do tipo **float**, independente do tipo do argumento.
- ❖ Definidas na biblioteca **math.h**

```
pow(base, expoente); // base^(expoente)  
exp(num); // e^(num)
```

# Logaritmo

- ❖ Definidas na biblioteca `math.h`
- ❖ `log`: logaritmo natural (base neperiana).
- ❖ `log10`: logaritmo na base 10.
- ❖ Resultado é sempre do tipo `float`, independente do tipo do argumento.

```
log(valor);  
log10(valor);  
log2_10 = log(10)/log(2);
```

# Raiz quadrada

- ❖ `sqrt`: raiz quadrada (square root).
- ❖ Resultado é sempre do tipo `float`, independente do tipo do argumento.
- ❖ Definida na biblioteca `math.h`

```
sqrt (número) ;
```

# Funções Trigonométricas

- ❖ Definidas na biblioteca `math.h`
- ❖ `sin`: seno de um ângulo.
- ❖ `cos`: cosseno de um ângulo.
- ❖ `tan`: tangente de um ângulo.

```
sin (angulo) ;  
cos (angulo) ;  
tan (angulo) ;
```

# Funções Trigonométricas

- ❖ Definidas na biblioteca `math.h`
- ❖ `asin`: arco-seno de um valor.
- ❖ `acos`: arco-cosseno de um valor.
- ❖ `atan`: arco-tangente de um valor.

```
asin(valor);  
acos(valor);  
atan(valor);
```



# Funções Trigonométricas Hiperbólicas

- ❖ Definidas na biblioteca `math.h`
- ❖ `sinh`: seno hiperbólico de um valor.
- ❖ `cosh`: cosseno hiperbólico de um valor.
- ❖ `tanh`: tangente hiperbólica de um valor.

```
sinh(valor);  
cosh(valor);  
tanh(valor);
```

# Arredondamento

- ❖ **round**: arredonda para o inteiro mais próximo.
- ❖ **ceil**: arredonda para o menor inteiro maior que o número.
- ❖ **floor**: arredonda para o maior inteiro menor que o número.
- ❖ Resultado é sempre do tipo **float**.

# Gerando números aleatórios

- ❖ Para gerar números aleatórios, pode-se utilizar as funções `srand()` e `rand()`, definidas na biblioteca `stdlib.h`.
- ❖ A função `rand` gera um número aleatório entre zero e `RAND_MAX`, definido no próprio arquivo `stdlib.h`.
- ❖ A função `srand` especifica o valor inicial (`semente`) do gerador de números aleatórios.

```
srand(valor inteiro);  
rand();
```

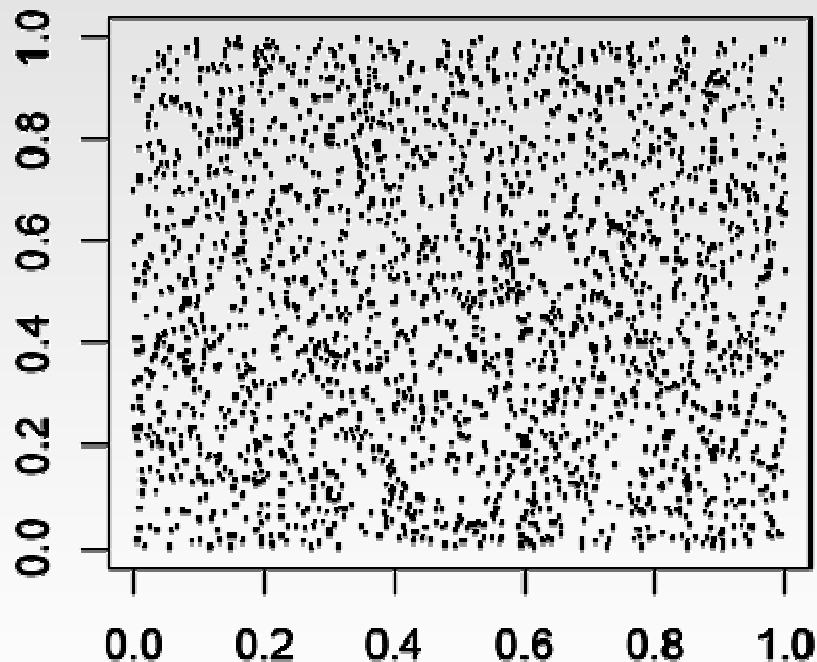
# Gerando números aleatórios

- ❖ Para gerar números aleatórios entre 0 e  $N-1$ , basta obter o resto da divisão do resultado de `rand()` por  $N$ .
- ❖ Para gerar diferentes seqüências de números a cada execução do programa, pode-se utilizar a hora do sistema como semente.
- ❖ A hora do sistema é obtida pela função `time()` (definida na biblioteca `time.h`).

```
srand(time(NULL));  
num_aleat = rand() % 101;
```

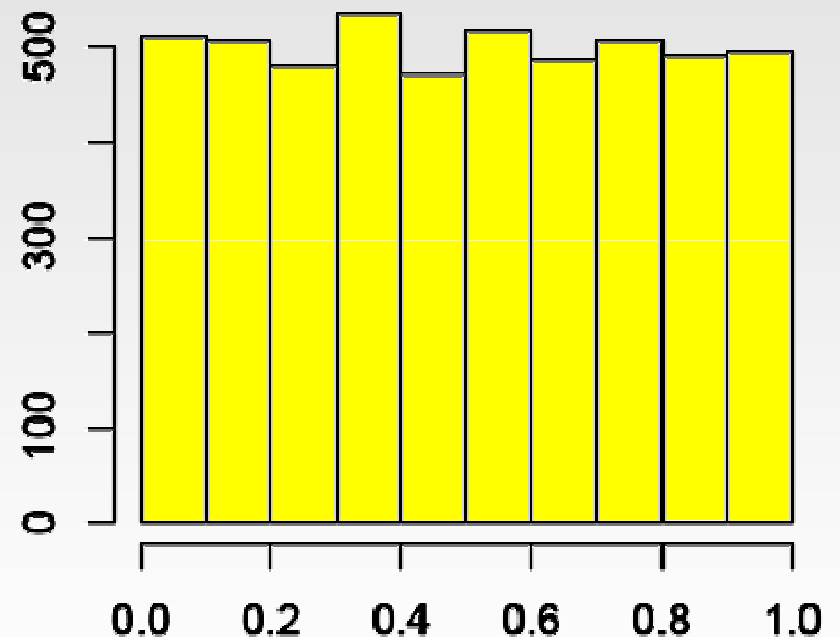
# Gerando números aleatórios :: Analisando aleatoriedade

Espalhamento  
(scatter plot)



Auxilia na detecção de **tendências**. Cada par de números da seqüência aleatória forma uma coordenada do gráfico.

Histograma



Auxilia na verificação do balanceamento, da uniformidade da geração de números.

# Questões

