

Algoritmo (pseudocódigo corregido)

Ejemplos de una serie de Taylor

5 funciones

$$f_1(x) = e^x$$

$$f_2(x) = \sin(x)$$

$$f_3(x) = \cos(x)$$

$$f_4(x) = \ln(1+x)$$

$$f_5(x) = 1/(1-x)$$

Variables

enteras opcion, n=0

dobles x, epsilon, suma=0, termino=0, fact, ewcr

Inicio

escribir ("Elige una función")

escribir ("1. e^x ")

escribir ("2. $\sin(x)$ ")

escribir ("3. $\cos(x)$ ")

escribir ("4. $\ln(1+x)$ ")

escribir ("5. $1/(1-x)$ ")

leer (opcion)

escribir ("Ingresa el valor de x")

leer (x)

escribir ("Ingresa el valor de epsilon")

leer (epsilon)

Segun sea el caso (opcion):

Caso 1:

Hacer que


```
fact = 1
para que la (variable entera i = 1, i <= n, i++)
    fact *= i
termino = raíz de (x^n) / fact
suma += termino
n++
mientras que (termino > epsilon)
    escribir ("Aproximación de e^x =")
    escribir ("Valor real: ")
    error = exponente de (x) - suma
    escribir ("Error: ")
    finalizar el caso 1
```

Caso 2:

```
hacer que
    fact = 1
    para que la (variable entera i = 1; i <= 2 * n + 1; i++)
        fact *= i
    termino = raíz de (-1, n) * raíz de (x, 2 * n + 1) / fact
    suma += termino
    n++
mientras que (termino > epsilon)
    escribir ("Aproximación de sen(x) =")
    escribir ("Valor real: ")
    error = sen(x) - suma
    escribir ("Error: ")
    finalizar el caso 2
```

Caso 3:

```
hacer que
    fact = 1
    para que la (variable entera i = 1; i <= 2 * n; i++)
```


Gabriele Lossen-Casanova

31

08

2020

Scribe

```
fact = 1
termino = raíz de (-1, n) * raíz de (x, 2+n) / fact
suma += termino
n++
mientras que (termino > epsilon)
  escribir ("Aproximación de cos(x): ")
  escribir ("valor real: ")
  error = cos(x) - suma
  escribir ("Error: ")
  finalizar el caso 3
```

Caso 4:

Si $(x \leq -1)$

escribir ("Error ln(1+x) no se puede calcular")
 regresar a 1

n=1

hacer que

fact = 1

para que la variable entera i=1; i<=n; i++

fact *= i

termino = raíz de (-1, n) * raíz de (x, n) / fact

suma += termino

n++

mientras que (termino > epsilon)

escribir ("Aproximación de ln(1+x): ")

escribir ("valor real: ")

error = (logaritmo de (1+x)) - suma

escribir ("Error: ")

finalizar el caso 5

Caso 5:

Si $(x \geq 1)$


```
escribir ("Error: no se puede calcular")
regresar a 1
hacer que
  fact = 1
  para que la variable entera i = 1; i <= n; i++
    fact *= i
  termino = valor de (x^n) / fact
  suma = suma + termino
  n++
  mientras que (termino > epsilon)
    escribir ("Aproximacion 1/(1-x) = ")
    escribir ("valor negl: ")
    Error = (1/(1-x)) - suma
    escribir ("Error = ")
  Fin hacer el caso S
```

Cuando no:

```
escribir ("No se puede calcular")
regresar a 1
```

```
escribir ("Valor de n: ")
```

Fin

Diferencias del algoritmo anterior al algoritmo actual.

- Se cambiaron las variables.
- Se uso otra condición diferente a los que se tenía solo se dejó el switch.

Gabriel Lasso Casanova

31 08 2020

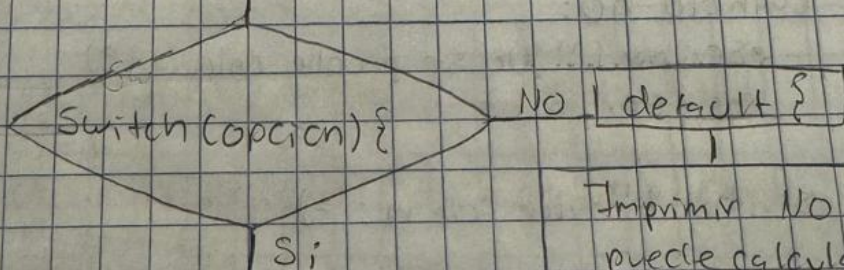
Scribe

Diagrama de Flujo corregido

Inicio

Declarar variables enteras opcion, n
Declarar variables dobles x, epsilon, suma = 0, termino = 0,
fact, error

Imprimir "Elige una función:
Imprimir 1. e^x
Imprimir 2. $\sin(x)$
Imprimir 3. $\cos(x)$
Imprimir 4. $\ln(1+x)$
Imprimir 5. $1/(1-x)$
Imprimir Ingresa el valor de x
Imprimir Ingresa el valor de epsilon



```
Case (opcion) 1:  
do {  
    fact = 1;  
    for (int i = 1; i <= n; i++)  
        fact *= i;  
    termino = pow(x, n) / fact;  
    suma += termino;  
    n++;  
}
```

```
while (termino > epsilon)  
Imprimir Aproximacion
```



```
Imprimir Valor real
error =  $\sin(x) - \text{Suma}$ 
Imprimir error
break;
```

case 2:

do {

fact = 1;

for (int i = 1; i <= $2 * n + 1$; i++)

fact *= i;

termino = $\text{pow}(-1, n) * \text{pow}(x, 2 * n + 1) / \text{fact}$

Suma += termino

n++

while (termino > epsilon)

Imprimir Aproximacion

Imprimir Valor real

error = $\sin(x) - \text{Suma}$;

Imprimir error

break;

case 3:

do {

fact = 1;

for (int i = 1; i <= $2 * n$; i++)

fact *= i;

termino = $\text{pow}(-1, n) * \text{pow}(x, 2 * n) / \text{fact}$

Suma += termino

n++

while (termino > epsilon)

Imprimir Aproximacion

Imprimir Valor real

error = $\cos(x) - \text{Suma}$

Imprimir error

break


```
case 4:  
if (x <= -1) {  
    Imprimir "no se puede calcular"  
    return 1; }  
n = 1  
do {  
    fact = 1  
    for (int i = 1; i <= n; i++)  
        fact *= i  
    termino = pow(-1, n) * pow(x, n) / fact  
    suma += termino  
    n++  
} while (termino > epsilon)  
Imprimir Aproximacion  
Imprimir valor real  
error = (log(1-x)) - suma  
Imprimir error  
break;
```

```
case 5:  
if (x >= 1) {  
    Imprimir "no se puede calcular"  
    return 1; }  
do {  
    fact = 1;  
    for (int i = 1; i <= n; i++)  
        fact *= i  
    termino = pow(x, n) / fact;  
    suma += termino;  
    n++;  
} while (termino > epsilon)  
Imprimir Aproximacion
```


Gabriela Cossan Casanova

D 31 M 08 A 2025

Scribe®

```
Imprimir Valor de l  
error = (1/(1-x)) - suma  
Imprimir error  
break; }
```

```
Imprimir Valor de n
```

Fin