

Diagrama de Flujo Serie de Taylor

(Inicio)

Definir función factorial con for
 $\text{int Facto}=1$. For($\text{int } i=1$; $i \leq n$; $i++$)

Declarar Float $x, x_i, \text{valor actual}, \text{valor anterior}, E_a, E_s$
 $\text{int } n, \text{opcion}, k$
 $\text{char } \text{iniciar}$.

def

Ingresar y guardar x, x_i, E_s, n
 $\text{Float } h = x - x_i, \text{epsilon} = x \cdot x_i / 2$

Arreglos \rightarrow Float Funcion[7] FuncionLe
 $= f(x_i) \rightarrow f(\text{epsilon})$
 $= f'(x_i) \rightarrow f'(\text{epsilon})$
 $= f''(x_i) \rightarrow f''(\text{epsilon})$
 $= f'''(x_i) \rightarrow f'''(\text{epsilon})$
 $= f^{(4)}(x_i) \rightarrow f^{(4)}(\text{epsilon})$
 $= f^{(5)}(x_i) \rightarrow f^{(5)}(\text{epsilon})$

Puntero a función seleccionada
 Imprimir "opciones 1-5"
 Imprimir opcion, scanf("%d", &opcion)

Switch (opcion)
 Case 1 "valor verdadero" $= 1/(x-2)$
 Case 2 $= \sqrt{x}$
 Case 3 $= 2^x$
 Case 4 $= \log_{10}(x)$
 Case 5 $= \cos(x)$

Sino For

if
 $n \leq 0$
 else

Verdadero

Switch (opcion)
 Case 1 $\text{valor actual} = 1/(x_i-2)$
 Case 2 $= \sqrt{x_i}$
 Case 3 $= 2^{x_i}$
 Case 4 $= \log_{10}(x_i)$
 Case 5 $= \cos(x_i)$

float residuo = 0

for($\text{int } k=0$; $k \leq n$; $k++$)

valor anterior = valor actual

$\text{valor actual} += (\text{funcionseleccionada}[k] * \text{pow}(h, k)) / \text{factorial}(k)$

$E_a = (\text{valor verdadero} - \text{valor actual})$

Residuo

No Si

Imprimir "no se pueda obtener
 Mn para un n > 5
 No hay suficientes
 derivadas"

Float derivada suma
 Switch (opcion)
 Case 1 $= \text{funcionLe}[k+1]$
 Case 2 $= \text{funcionRe}[k+1]$
 Case 3
 Case 4
 Case 5
 $\text{residuo} = (\text{derivada suma} * \text{pow}(h, k+1)) / \text{factorial}(k+1)$

Imprimir n, valor actual, E_a, Residuo

Si No

¿cálculo
 correcto?

Imprimir:
 Error calculado en k

if
 $E_a \leq E_s$
 Si

Pseudocódigo de Serie de Taylor

Bibliotecas

```
int factorial(int n)
int facto = 1
for (int i = 1; i <= n; i++)
    facto = facto * i;
```

```
int n, opcion, k; char inicial; float x, xi, valor actual, valor anterior, Ea, Es;
do {
```

Ingresar y guardar x, xi, Es, n

float h = xi - x; epsilon = (x + xi) / 2

float Funcion1[7] Guardar derivadas en arreglo para cada funcion

```
Funcion1[0] = 1 / (xi - 2);
Funcion1[1] = -1 / pow((xi - 2), 2);
Funcion1[2] = 2 / pow((xi - 2), 3);
Funcion1[3] = -6 / pow((xi - 2), 4);
// hasta sexta derivada
```

Hacer otro arreglo igual pero las funciones y sus derivadas evaluadas en epsilon

float * funcionseleccionada

float valor verdadero

printf("selecciona la funcion")

1: F(x) = 1 / (x - 2) 2: F(x) = sqrt(x) 3: F(x) = pow(2, x) 4: F(x) = log(x) 5: F(x) = cos(x)

switch (opcion)

Case 1: funcionseleccionada = Funcion1[0] valor verdadero = 1 / (xi - 2); break

Para cada funcion Case 1 a Case 5

if (n == 0)

switch (opcion)

Case 1: valor actual = 1 / (xi - 2); break;

else

valor actual = 0; float res = 0;

for (int k = 0; k <= n; k++)

valor anterior = valor actual

valor actual += (funcionseleccionada[k] * pow(h, k)) / factorial(k)

Ea = fabs(valor verdadero - valor actual)

if (k == 0 && k <= 5)

float derivada suma

Switch (opcion)

Case 1: derivadaSuma = funcion 1 e $[k+1]$, break

hasta Case 5

residuo = (derivadaSuma * pow(h, k+1)) / factorial((k+1))

else

PrintK(no hay suficientes derivadas definidas)

if (Ea < Es)

Printf("El error deseado se alcanza en la iteración n)

Printf(valor verdadero de la función)

(escoger otra función)

while

Fin

Cambios

El primer cambio realizado fue que después de que el usuario meta sus datos se guardo en arreglos cada una de las 5 funciones con sus derivadas hasta la sexta evaluadas en x_i y otro para epsilon que se usa para el Δn . Para que el usuario escogiera su función se agrego un menú con un puntero. Seguido de esto se uso un switch de control para conocer el valor verdadero con el que se comparara E_a con E_s . En el ciclo if-else se dejo casi igual, cuando $n=0$ el valor actual en ese momento es la función evaluada en x_i y else un ciclo for para el calculo de Taylor junto con E_a . Dentro del ciclo for se puso otro if-else para el residuo acompañado de un switch para la función que se selecciona. Como límite se puso que n no puede ser mayor a 5 ya que para valores más altos Δn no será posible por falta de derivadas, y por ultimo un if-break en si $E_a < E_s$ imprima en cual iteración se alcanza el error aun imprimiendo hasta el n registrado.