



Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico Matemáticas



Análisis Numérico

Evidencia 3:

“Polinomio de interpolación con diferencias divididas de Newton”



Alumno: Jesús Armando Espino Rodríguez
Matricula: 1844607

Profesora: María del Carmen Martínez Cejudo
Grupo: 031
Horario: 09:00 am a 10:00 am





Interpolación de polinomios con diferencias divididas de Newton.

- El método de interpolación de polinomios con diferencias divididas de Newton es una técnica para encontrar un polinomio que pase exactamente por un conjunto de puntos dados. Utiliza las diferencias divididas entre los puntos para construir un polinomio de grado igual al número de puntos menos uno. Este método es útil para aproximar funciones desconocidas o para interpolar valores entre puntos conocidos.

Veamos un ejemplo del método.

Tenemos los siguientes puntos:

x	1	2	9
y	7	8	5

Identificamos el polinomio: $c_0 + c_1(x - 1) + c_2(x - 1)(x - 2)$

Hacemos un sistema de ecuación para encontrar las constantes

$$c_0 + c_1(1 - 1) + c_2(1 - 1)(1 - 2) = 7$$

$$c_0 + c_1(2 - 1) + c_2(2 - 1)(2 - 2) = 8$$

$$c_0 + c_1(9 - 1) + c_2(9 - 1)(9 - 2) = 5$$

El sistema resuelto es el siguiente: $c_0 = 7$, $c_1 = 1$, $c_2 = 10/56$

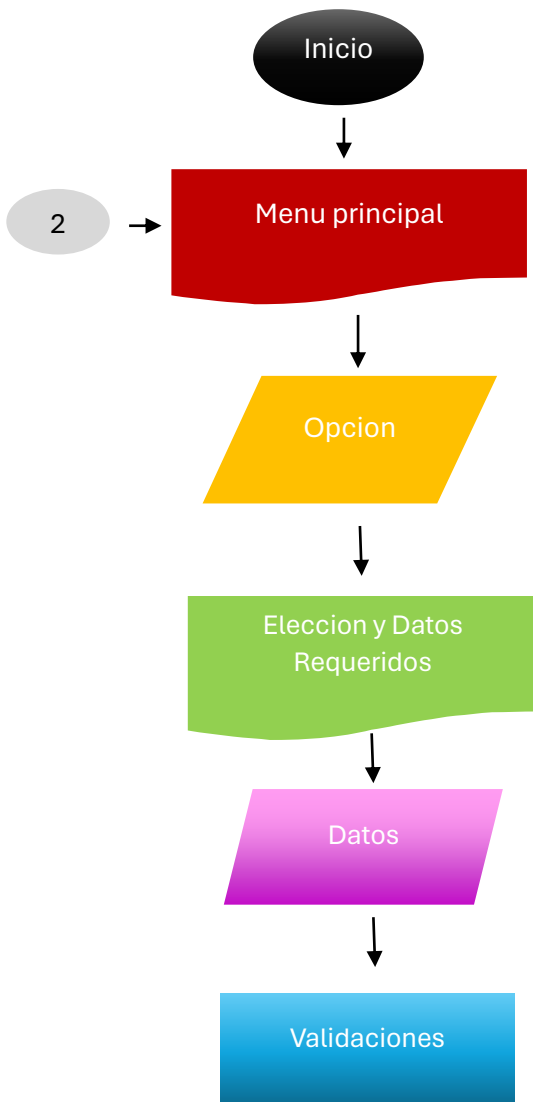
Procedemos a evaluar las constantes para encontrar el punto $y(3)$:

$$y(3) = c_0 + c_1(3 - 1) + c_2(3 - 1)(3 - 2)$$

$$y(3) = 7 + (1)(3 - 1) + (10/56)(3 - 1)(3 - 2)$$

$$y(3) = 8.642857142857142$$

Diagrama de Flujo



- El diagrama comenzaría con un nodo de inicio, indicando el inicio del programa.

- Se mostrará el menú principal del programa, el cual contará con 4 opciones actualmente.

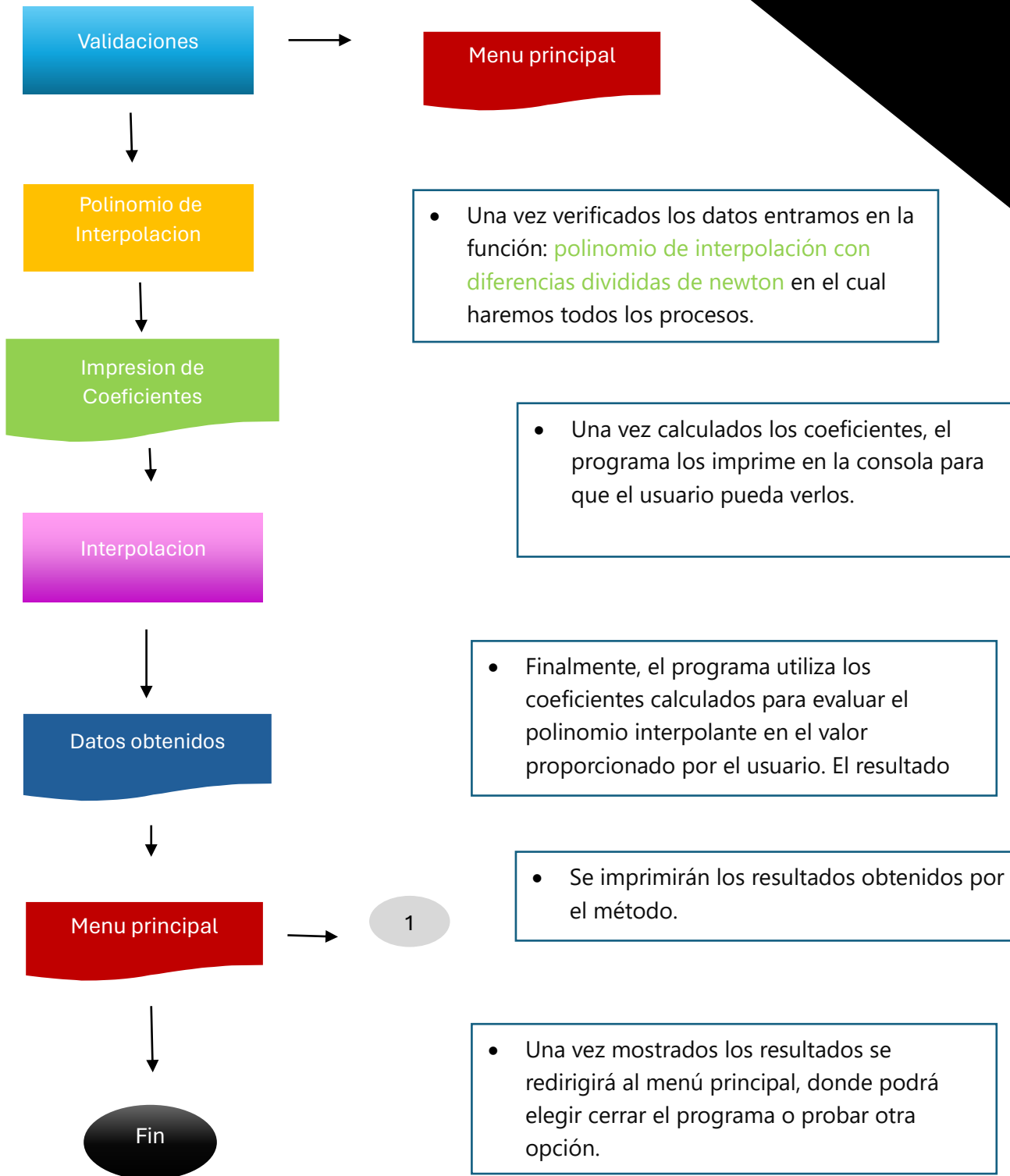
- Se debe ingresar una opción de las mostradas en el menú principal.

- Se imprimirá la elección tomada, para posteriormente pedirnos los datos requeridos por el método.

- Se ingresan los datos requeridos, en este caso se pedirá un numero de puntos y un valor a interpolar.

- Se realizarán validaciones sobre los datos ingresados en caso de no ser correctos se mostrará el menú principal del programa.

While



Codificación del programa

Menú de bienvenida

```
def menu():
    while True:
        print("\nSeleccione una de las siguientes opciones:")
        print("1. Eliminación gaussiana")
        print("2. Método de Bisección")
        print("3. Polinomio de Interpolación con diferencias divididas de Newton")
        print("4. ...")
        print("5. Cerrar el programa")

        opcion = input("Ingrese el número de la opción deseada: ")

        if opcion == "1":
            resolver_eliminacion_gaussiana()
        elif opcion == "2":
            resolver_metodo_biseccion()
        elif opcion == "3":
            interpolacion_diferencias_divididas()
        elif opcion == "5":
            print("Cerrando el programa...")
            break
        else:
            print("Opción inválida. Por favor, seleccione una opción válida.")
```

- Para empezar con este programa, tenemos un menú principal, en el cual, actualmente tenemos 4 opciones:
 - Eliminación gaussiana.
 - Método bisección.
 - Polinomio de interpolación con diferencias divididas de Newton
 - Cerrando el programa
- Comenzamos seleccionando una opción en la terminal, para después entrar en un sub-menú donde se pedirán los datos necesarios para resolver el problema planteado
- En caso de ingresar una opción que este fuera de las antes mencionadas, se mostrara un mensaje de error, para después mostrar nuevamente el menú.

Función: interpolación_diferencias_divididas

```
def interpolacion_diferencias_divididas():
    print("Has seleccionado Polinomio de Interpolacion con diferencias divididas de Newton.")
    # Pedir al usuario la información relacionada con los puntos
    while True:
        try:
            n = int(input("Ingrese el número de puntos: "))
            if n <= 0:
                raise ValueError("El número de puntos debe ser un entero positivo.")
            break
        except ValueError as e:
            print("Error:", e)

    x = []
    y = []
    for i in range(n):
        while True:
            try:
                punto_x = float(input("Ingrese el valor de x{}: ".format(i)))
                punto_y = float(input("Ingrese el valor de y{}: ".format(i)))
                x.append(punto_x)
                y.append(punto_y)
                break
            except ValueError:
                print("Error: Por favor, ingrese un número real válido.")

    # Pedir al usuario el valor a interpolar
    while True:
        try:
            valor_interpol = float(input("Ingrese el valor a interpolar: "))
            break
        except ValueError:
            print("Error: Por favor, ingrese un número real válido.")

    # Calcular coeficientes
    coeficientes = diferencias_divididas(x, y)
```

- Entrando a esta función, encontraremos un mensaje confirmando la elección que hemos tomado. Después tendremos un mensaje que nos pedirá ingresar una función por pantalla, verificando que esta cumpla con los criterios de sintaxis.
- Entramos en un bucle, en el cual se pedirá ingresar el número de puntos que se ingresaran, dentro de este bucle se hacen validaciones para verificar que los intervalos sean enteros.
- Después procedemos a ingresar los valores de dichos puntos, y por último el valor a interpolar, corroborando que los valores sean correctos, al final de este método hacemos llamado de dos funciones.

Función: diferencias_divididas y evaluar_polinomio

```
def diferencias_divididas(x, y):  
    n = len(x)  
    coeficientes = [0] * n  
    for i in range(n):  
        coeficientes[i] = y[i]  
  
    for j in range(1, n):  
        for i in range(n - 1, j - 1, -1):  
            coeficientes[i] = (coeficientes[i] - coeficientes[i - 1]) / (x[i] - x[i - j])  
  
    return coeficientes
```

- El método `diferencias_divididas` nos permite saber cuales son los valores de las constantes, primero verifica cuantos puntos fueron ingresados, para después entrar en un ciclo que se encarga de evaluar los valores de las constantes.

```
def evaluar_polinomio(x, coeficientes, valor):  
    n = len(x)  
    resultado = coeficientes[n - 1]  
    for i in range(n - 2, -1, -1):  
        resultado = resultado * (valor - x[i]) + coeficientes[i]  
    return resultado
```

- El método `evaluar_polinomio` se encarga de usar el valor de las constantes, y el valor a interpolar para evaluar la función del problema. Esto nos lleva a devolver los resultados obtenido para después imprimirlos al usuario.

Impresión de resultados

```
# Calcular coeficientes
coeficientes = diferencias_divididas(x, y)

# Imprimir coeficientes
print("Los coeficientes del polinomio interpolante son:")
for i, coeficiente in enumerate(coeficientes):
    print("c{} = {:.6f}".format(i, coeficiente))

# Evaluar el polinomio en el valor a interpolar
resultado = evaluar_polinomio(x, coeficientes, valor_interpoliar)

print("El resultado de la interpolación en x =", valor_interpoliar, "es y =", resultado)
```

```
print("El resultado de la interpolación en x =", valor_interpoliar, "es y =", resultado)
```

- La impresión de los resultados se muestra en esta sección, la impresión consta de los valores de las constantes, el punto interpolado y el valor resultado de evaluar estos datos en la función del problema.
- Una vez mostrados los resultados se regresa al menú principal.

Manual del usuario

Seleccione una de las siguientes opciones:

1. Eliminación gaussiana
2. Método de Bisección
3. Polinomio de Interpolacion con diferencias divididas de Newton
4. ...
5. Cerrar el programa

Ingrese el número de la opción deseada: 3

- Aquí vemos el menú principal del programa, donde tendremos que seleccionar una opción.
- En este caso iremos por la opción 3 la cual nos dirigirá al método de polinio de interpolación con diferencias divididas de Newton

Has seleccionado Polinomio de Interpolacion con diferencias divididas de Newton.

Ingrese el número de puntos: 3

Ingrese el valor de x_0 : 1

Ingrese el valor de y_0 : 7

Ingrese el valor de x_1 : 2

Ingrese el valor de y_1 : 8

Ingrese el valor de x_2 : 9

Ingrese el valor de y_2 : 5

Ingrese el valor a interpolar: 3

- Al ingresar al método nos imprime un mensaje reiterando la opción elegida, para después preguntarnos el numero de puntos, los valores que le daremos y el valor por el cual evaluaremos la función del problema

```
Los coeficientes del polinomio interpolante son:  
c0 = 7.000000  
c1 = 1.000000  
c2 = -0.178571  
El resultado de la interpolación en x = 3.0 es y = 8.642857142857142
```

- El método usará los valores ingresados anteriormente para resolver el problema y posteriormente imprimir los resultados obtenidos, entre ellos tenemos el valor de las constantes, el valor del punto evaluado y el valor obtenido de dicha evaluación.

```
El resultado de la interpolación en x = 3.0 es y = 8.642857142857142
```

```
Seleccione una de las siguientes opciones:
```

1. Eliminación gaussiana
2. Método de Bisección
3. Polinomio de Interpolación con diferencias divididas de Newton
4. ...
5. Cerrar el programa

```
Ingrese el número de la opción deseada: 5
```

```
Cerrando el programa...
```

- Para finalizar tenemos devuelta el menú, al terminar el método este nos devolverá y se nos mostrará el menú inicial, en esta ocasión elegimos la opción 5, y cierra el programa.

Conclusiones

- Durante la implementación del programa se vieron posibles alternativas para llevar a cabo este trabajo, se añadió el método de polinomio de interpolación con diferencias divididas al menú del programa anterior.
- Fue entretenido elaborar este documento, se hizo con la intención de ser llamativo y de fácil comprensión, siendo un guía en cada paso de la elaboración e introducción del método.
- Espero que sea de su agrado y éxito a todos.

Referencias Bibliográficas

- Kolman, Bernard, y Hill, David R. (2019). "Elementary Linear Algebra with Applications". Pearson.
- Python Documentation: <https://docs.python.org/3/library/array.html>
- "Análisis Numérico" de Richard L. Burden y J. Douglas Faires.