

Explicación del funcionamiento del programa

Programa

```
cat("\014")
rm(list=ls())
N=as.numeric(readline("Inserta el número de valores de s"))
s=numeric(N)
f=numeric(N)
x=as.numeric(readline("Inserta el valor de x a evaluar"))
for (i in 1:N) {
  s[i]=as.numeric(readline(cat("Inserta s",i)))
}
for (i in 1:N) {
  f[i]=as.numeric(readline(cat("Inserta f(s",i,")")))
}
Polinomios_de_lagrange=matrix(0,nrow=N,ncol=N)
#Cada polinomio de lagrange estará almacenado en una fila de la matriz
for (j in 1:N){
  numerador=c()
  base=numeric(N)
  denominador=1
  y=0
  for (i in 1:N){
    if (i != j){
      numerador_x=c(0,numerador)
      numerador=c(numerador,0)
      numerador=(-s[i])*numerador+numerador_x
      while (y==0) {
        numerador[1]=-s[i]
        numerador[2]=1
        y=1
      }
      denominador=denominador*(s[j]-s[i])
    }
  }
  base=numerador/denominador
  for (i in 1:N) {
    Polinomios_de_lagrange[i,j]=base[i]
  }
}
Pol=numeric(N)
```

```

for (i in 1:N) {
  for (j in 1:N) {
    Pol[i]=Pol[i]+(Polinomios_de_lagrange[i,j]*f[j])
  }
}
#Ahora tenemos un polinomio interpolador calculado por Lagrange y una matriz con todas las
bases de Lagrange
#Vamos a sacar el resultado del valor de la funcion en un punto.
Valor_en_x=0
for (i in 1:N) {
  Valor_en_x=Valor_en_x+(Pol[i]*(x^(i-1)))
}
cat("El valor de la función en x es: ",Valor_en_x, "\n")
#Ahora mostramos el valor de la solicitada derivada y la función tanto normal como derivada
cat("La funcion polinómica es: ")
for (i in 1:N) {
  if (Pol[i] != 0) {
    cat(Pol[i],"x ^",i-1," ")
  }
}
Pol_copia_positiva=c(Pol)
for (i in 1:N) {
  if (Pol_copia_positiva[i] < 0){
    Pol_copia_positiva[i]=-Pol_copia_positiva[i]
  }
}
Sumatorioterminos=0
for (i in 1:N) {
  Sumatorioterminos=Sumatorioterminos+Pol_copia_positiva[i]
}
if (Sumatorioterminos == 0){
  cat("0")
}
#Ahora que tenemos todo empezamos con la derivacion
Grado_derivacion=as.numeric(readline("Inserta el grado de derivacion:"))
Pol_der=c(Pol,0)
for (j in 1:Grado_derivacion){
  for (i in 1:N) {
    Pol_der[i]=Pol_der[i+1]*i
  }
}
cat("\n")
Valor_derivada_en_x=0
for (i in 1:N) {

```

```

Valor_derivada_en_x=Valor_derivada_en_x+(Pol_der[i]*(x^(i-1)))
}
cat("El valor de la derivada solicitada en x es: ",Valor_derivada_en_x, "\n")
cat("La funcion de la derivada es: ")
for (i in 1:N) {
  if (Pol_der[i] != 0) {
    cat(Pol_der[i],"x ^",i-1," ")
  }
}
Pol_der_copia_positiva=c(Pol_der)
for (i in 1:N) {
  if (Pol_der_copia_positiva[i] < 0){
    Pol_der_copia_positiva[i]=-Pol_der_copia_positiva[i]
  }
}
Sumatorioterminos=0
for (i in 1:N) {
  Sumatorioterminos=Sumatorioterminos+Pol_der_copia_positiva[i]
}
if (Sumatorioterminos == 0){
  cat("0")
}
cat("\n")
#Ahora hacemos la derivacion con las bases de lagrange en vez de con el polinomio resultante
Polinomios_de_lagrange_der=matrix(0, nrow=N, ncol=N)
for (j in 1:N) {
  Base_derivada=numeric(N)
  for (k in 1:Grado_derivacion) {
    for (i in 1:(N-1)) {
      Base_derivada[i]=Polinomios_de_lagrange[i+1,j]*i
    }
  }
  Polinomios_de_lagrange_der[,j]=Base_derivada
}
Polinomios_de_lagrange_der_multiplicado_por_valor=matrix(0,nrow=N,ncol=N)
for (j in 1:N) {
  Polinomios_de_lagrange_der_multiplicado_por_valor[,j]=Polinomios_de_lagrange_der[,j]*f[j]
}
Sumatorios_vector=numeric(N)
for (i in 1:N) {
  Sumatorioterminosb=0
  for (j in 1:N) {

```

```

Sumatorioterminosb=Polinomios_de_lagrange_der_multiplicado_por_valor[i,j]+Sumatorioterminos
b
}
Sumatorios_vector[i]=Sumatorioterminosb
}
#Ahora en Sumatorios_vector tengo un vector con los sumatorios de todos los terminos de cada
columna (Coincide el grado de x)
Sumatoriofinal=0
for (i in 1:N){
  Sumatoriofinal=Sumatoriofinal+(Sumatorios_vector[i]*(x^(i-1)))
}
cat("El valor de la derivada solicitada en x mediante la derivación de cada base de lagrange
individualmente es: ",Sumatoriofinal, "\n")

```

Explicación

```

cat("\014")
rm(list=ls())

```

- **cat("\014")**: Limpia la consola (similar a un "clear console").
- **rm(list=ls())**: Elimina todas las variables del entorno de trabajo, asegurándose de que no haya valores residuales.

```

N=as.numeric(readline("Inserta el número de valores de s"))
s=numeric(N)
f=numeric(N)
x=as.numeric(readline("Inserta el valor de x a evaluar"))

```

- **N**: Se pide al usuario cuántos puntos s se van a introducir.
- **s**: Vector para almacenar los valores de s .
- **f**: Vector para almacenar los valores de $f(s)$, que representan los valores de la función en los puntos s .
- **x**: Se pide un valor x para evaluar la función resultante en ese punto.

```

for (i in 1:N) {
  s[i]=as.numeric(readline(cat("Inserta s",i)))
}
for (i in 1:N) {
  f[i]=as.numeric(readline(cat("Inserta f(s",i,"")))

```

```
}
```

- El usuario ingresa los puntos s y los valores de la función $f(s)$ correspondientes.
- Se utiliza un bucle `for` para iterar desde 1 hasta N , solicitando cada uno de los puntos s y sus respectivos valores de la función $f(s)$.

```
Polinomios_de_lagrange=matrix(0,nrow=N,ncol=N)
```

- Se crea una matriz para almacenar los coeficientes de los polinomios de Lagrange. Cada fila representa un polinomio de base de Lagrange.

```
for (j in 1:N){
  numerador=c()
  base=numeric(N)
  denominador=1
  y=0
  for (i in 1:N){
    if (i != j){
      numerador_x=c(0,numerador)
      numerador=c(numerador,0)
      numerador=(-s[i]*numerador)+numerador_x
      while (y==0) {
        numerador[1]=-s[i]
        numerador[2]=1
        y=1
      }
      denominador=denominador*(s[j]-s[i])
    }
  }
  base=numerador/denominador
  for (i in 1:N) {
    Polinomios_de_lagrange[i,j]=base[i]
  }
}
```

Para cada j (de 1 a N), se construye el polinomio base de Lagrange $L_j(x)$ para el punto $s[j]$.

- **Numerador:** Se construye un polinomio

$$(x - s_1)(x - s_2) \cdots (x - s_{j-1})(x - s_{j+1}) \cdots (x - s_N)$$

- **Denominador:** Calcula la constante de normalización

$$(s_j - s_1)(s_j - s_2) \cdots (s_j - s_{j-1})(s_j - s_{j+1}) \cdots (s_j - s_N)$$

- Cada fila de la matriz **Polinomios_de_lagrange** almacena los coeficientes del polinomio base $L_j(x)$ para cada j .

```
Pol=numeric(N)
for (i in 1:N) {
  for (j in 1:N) {
    Pol[i]=Pol[i]+(Polinomios_de_lagrange[i,j]*f[j])
  }
}
```

- Se construye el polinomio interpolador de Lagrange $P(x)$ combinando los polinomios base $L_j(x)$ y los valores $f(s_j)$.
- El polinomio final **Pol** es la combinación lineal:

$$P(x) = f(s_1) \cdot L_1(x) + f(s_2) \cdot L_2(x) + \cdots + f(s_N) \cdot L_N(x)$$

```
Valor_en_x=0
for (i in 1:N) {
  Valor_en_x=Valor_en_x+(Pol[i]*(x^(i-1)))
}
cat("El valor de la función en x es: ",Valor_en_x, "\n")
```

- Evalúa el polinomio $P(x)$ en el valor de x ingresado por el usuario.
- Se utiliza la forma de coeficientes para calcular el valor de $P(x)$ mediante la fórmula:

$$P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{N-1}x^{N-1}$$

```
cat("La funcion polinómica es: ")
for (i in 1:N) {
  if (Pol[i] != 0) {
```

```

cat(Pol[i],"x ^",i-1," ")
}
}

```

- Se imprime la forma explícita del polinomio interpolador, mostrando cada coeficiente a_j con su respectivo grado i .

```

Pol_copia_positiva=c(Pol)
for (i in 1:N) {
  if (Pol_copia_positiva[i] < 0){
    Pol_copia_positiva[i]=-Pol_copia_positiva[i]
  }
}
Sumatorioterminos=0
for (i in 1:N) {
  Sumatorioterminos=Sumatorioterminos+Pol_copia_positiva[i]
}
if (Sumatorioterminos == 0){
  cat("0")
}

```

- Convierte todos los coeficientes de **Pol** en valores absolutos.
- Calcula la suma de los coeficientes.
- Si la suma es 0, imprime "**0**", lo que sugiere que el polinomio resultante es 0.

```

Grado_derivacion=as.numeric(readline("Inserta el grado de derivacion:"))

```

- Se solicita al usuario que introduzca el grado de la derivada que desea calcular.
- Este valor se almacena en la variable **Grado_derivacion**, que se utilizará en las secciones posteriores para calcular las derivadas del polinomio y de las bases de Lagrange.

```

Pol_der=c(Pol,0)
for (j in 1:Grado_derivacion){
  for (i in 1:N) {
    Pol_der[i]=Pol_der[i+1]*i
  }
}

```

1. **Extensión del polinomio:** Se añade un 0 adicional al final de **Pol** para permitir el corrimiento de coeficientes durante la derivación.
2. **Cálculo de la derivada:**

- Se aplica la regla de la potencia: la derivada de $(a_n x^n)$ es $(n \cdot a_n x^{n-1})$.
- Para cada coeficiente del polinomio, se calcula su derivada y se guarda en la posición correspondiente de **Pol_der**.
- Este proceso se repite para cada grado de derivación solicitado por el usuario.

```

Valor_derivada_en_x=0
for (i in 1:N) {
  Valor_derivada_en_x=Valor_derivada_en_x+(Pol_der[i]*(x^(i-1)))
}
cat("El valor de la derivada solicitada en x es: ",Valor_derivada_en_x, "\n")

```

- Se evalúa la derivada del polinomio en el punto x usando la misma fórmula utilizada previamente para la evaluación del polinomio.
- Se muestra el resultado en pantalla.

```

cat("La funcion de la derivada es: ")
for (i in 1:N) {
  if (Pol_der[i] != 0) {
    cat(Pol_der[i],"x ^",i-1," ")
  }
}

```

- Muestra la forma explícita de la función derivada, presentando cada coeficiente con su respectivo exponente de x.

```

Pol_der_copia_positiva=c(Pol_der)
for (i in 1:N) {
  if (Pol_der_copia_positiva[i] < 0){
    Pol_der_copia_positiva[i]=-Pol_der_copia_positiva[i]
  }
}
Sumatorioterminos=0
for (i in 1:N) {
  Sumatorioterminos=Sumatorioterminos+Pol_der_copia_positiva[i]
}
if (Sumatorioterminos == 0){
  cat("0")
}

```

- Convierte a positivo cada coeficiente de la derivada.
- Calcula la suma de los coeficientes absolutos.
- Si la suma es 0, entonces la derivada es nula, lo que se indica mostrando **0** en la consola.

```

Polinomios_de_lagrange_der=matrix(0, nrow=N, ncol=N)
for (j in 1:N) {
  Base_derivada=numeric(N)
  for (k in 1:Grado_derivacion) {
    for (i in 1:(N-1)) {
      Base_derivada[i]=Polinomios_de_lagrange[i+1,j]*i
    }
  }
  Polinomios_de_lagrange_der[,j]=Base_derivada
}

```

- Se calcula la derivada de cada base de Lagrange ($L_j(x)$) de forma individual.
- Para cada grado de derivación solicitado, se aplica la regla de la potencia, calculando la derivada de cada coeficiente de la base de Lagrange.
- La derivada de cada base se almacena en la matriz **Polinomios_de_lagrange_der**.

```

Polinomios_de_lagrange_der_multiplicado_por_valor=matrix(0,nrow=N,ncol=N)
for (j in 1:N) {
  Polinomios_de_lagrange_der_multiplicado_por_valor[,j]=Polinomios_de_lagrange_der[,j]*f[j]
}

```

- Multiplica cada base derivada $L'_j(x)$ por su respectivo valor ($f(s_j)$) para construir la derivada del polinomio interpolador.

```

Sumatorios_vector=numeric(N)
for (i in 1:N) {
  Sumatorioterminosb=0
  for (j in 1:N) {
    Sumatorioterminosb=Polinomios_de_lagrange_der_multiplicado_por_valor[i,j]+Sumatorioterminosb
  }
  Sumatorios_vector[i]=Sumatorioterminosb
}

```

- Suma las contribuciones de todas las bases derivadas multiplicadas por ($f(s_j)$).
- El resultado se almacena en **Sumatorios_vector**, que representa los coeficientes del polinomio derivado.

```
Sumatoriofinal=0
for (i in 1:N){
  Sumatoriofinal=Sumatoriofinal+(Sumatorios_vector[i]*(x^(i-1)))
}
cat("El valor de la derivada solicitada en x mediante la derivación de cada base de lagrange
individualmente es: ",Sumatoriofinal, "\n")
```

- Evalúa el polinomio derivado (obtenido usando las bases de Lagrange) en el valor de x .
- Imprime el resultado, que debe coincidir con la evaluación obtenida previamente al derivar el polinomio completo.