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")
}
```

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 Lj(x) y los valores f(s_j).
- El polinomio final **Pol** es la combinación lineal:

$$P(x) = f(s1) \cdot L1(x) + f(s2) \cdot L2(x) + \cdots + f(sN) \cdot LN(x)$$

```
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.