```
# INTERPOLACIÓN CON SISTEMA DE ECUACIONES
p=function(a,x,f){
	n=length(x);d=0
	A=matrix(c(0),nrow=n,ncol=n)
	for(i in 1:n){
		for(j in 1:n){
			A[i,j]=x[i]**(j-1)
		}
	}
	B=matrix(c(f),nrow=n,ncol=1)
	c=solve(A,B)
	for(i in 1:length(c)){
		d=d+c[i]*a**(i-1)
	}
	return(d)
}
#-----------------------------------------------------------------------------------------------------------
# BASES LAGRANGE
Bases_Lagrange=function(a,x){
	n=length(x)
	prod=1
	for(i in 1:n){
		prod[i]=1
		for(j in 1:n){
			if(i!=j){
				prod[i]=prod[i]*(a-x[j])/(x[i]-x[j])
			}
		}
	}
	return(prod)
}
```

```
w=function(Bases_Lagrange,f){
	suma=0; n=length(f)
	L=Bases_Lagrange(a,x)
	for(i in 1:n){
		suma=suma+f[i]*L[i]
	}
	return(suma)
}
#---------------------------------------------------------------------------
# DIFERENCIAS DIVIDIDAS/ FÓRMULA DE NEWTON
m=function(a,x,f){
	suma=f[1];n=length(x)
	A=matrix(c(0),nrow=n,ncol=n)
	A[,1]=f
	for(j in 2:n){
		for(i in 1:(n-j+1)){
			A[i,j]=(A[i+1,j-1]-A[i,j-1])/(x[i+j-1]-x[i])
		}
	}
	d=A[1,]
	for(i in 2:n){
		prod=1
		for(j in 1:(i-1)){
			prod=prod*(a-x[j])
		}
		suma=suma+d[i]*prod
	}
	return(suma)
}
```

```
#-----------------------------------------------------------------------------------------------
# *POR PARTES: BASES LAGRANGE GRADO1
#Por bases de Lagrange de grado 1
A=function(a,x){
	n=length(x);L=0
	if(a<=x[2]&a>=x[1]){
		L[1]=(a-x[2])/(x[1]-x[2])
		}else{
			L[1]=0
			}
		if(a<=x[n]&a>=x[n-1]){
			L[n]=(a-x[n-1])/(x[n]-x[n-1])
			}else{
				L[n]=0
			}
	for(i in 2:(n-1)){
		if(a<=x[i]&a>=x[i-1]){
			L[i]=(a-x[i-1])/(x[i]-x[i-1])
			}else if(a<=x[i+1]&a>=x[i]){
				L[i]=(a-x[i+1])/(x[i]-x[i+1])
					}else{
						L[i]=0
				}
		}
	return(L)
}
u=function(A,a,f){
	suma=0;n=length(f)
	L=A(a,x)
	for(i in 1:n){
		suma=suma+f[i]*L[i]
```

```
        }
        return(suma)
}
#-------------------------------------------------------------------------------------------
#POR PARTES* BASES LAGRANGE GRADO 2
#Por tramos con polinomios de grado 2. Necesario número de puntos impar
B=function(a,x){
        L=0;n=length(x)
        if(a<=x[3]&a>=x[1]){
                L[1]=(a-x[2])*(a-x[3])/((x[1]-x[2])*(x[1]-x[3]))
                }else{
                        L[1]=0
                }
        if(a<=x[n]&a>=x[n-2]){
                L[n]=(a-x[n-2])*(a-x[n-1])/((x[n]-x[n-1])*(x[n]-x[n-2]))
                }else{
                        L[n]=0
                }
        for(i in seq(2,(n-1),2)){
                if(a<=x[i+1]&a>=x[i-1]){
                        L[i]=(a-x[i-1])*(a-x[i+1])/((x[i]-x[i-1])*(x[i]-x[i+1]))
                        }else{
                                L[i]=0
                }
        }
        if(length(x)>3){
                for(i in seq(3,(n-2),2)){
                        if(a<=x[i]&a>=x[i-2]){
                                L[i]=(a-x[i-2])*(a-x[i-1])/((x[i]-x[i-2])*(x[i]-x[i-1]))
                                }else if(a<=x[i+2]&a>=x[i]){
                                        L[i]=(a-x[i+1])*(a-x[i+2])/((x[i]-x[i+1])*(x[i]-x[i+2]))
```

```
                                                }else{
                                                        L[i]=0
                                                }
                                        }
                        }
        return(L)
}
uu=function(B,a,f){
        suma=0;n=length(f)
        V=B(a,x)
        for(i in 1:n){
                suma=suma+f[i]*V[i]
                }
        return(suma)
}
#-------------------------------------------------------------------------------------------------
#*POR PARTES FÓRMULA DE NEWTON GRADO 1
AA=function(a,x,f){
        n=length(x)
        for(i in 1:(n-1)){
                if(a<=x[i+1]&a>=x[i]){
                        u=f[i]+(f[i+1]-f[i])/(x[i+1]-x[i])
                        }
                }
        return(u)
}
#-------------------------------------------------------------------------------------------------
#DATOS USADOS EN LOS PRIMEROS EJEMPLOS
a=10
x=c(1,4,5,6,9,18)
f=c(2,12,31,42,-12,-16)
```