

# Algoritmo Método del gradiente

```
Declarar max=10
Definir la función
double F (double x, int n)
para n=2; x[0]=x, x[1]=y
regresar  $2*x[0]^2 + x[1] + 2*x[0] - x[0]*x[0] - 2*x[1]*x[1]$ 
```

```
void gradiente (double x, double grad, int n, double delta)
```

```
double temp [max]
```

```
For desde i=0 hasta i < n:
```

```
For desde j=0 hasta j < n ; temp[j] = x[j]
```

```
temp[i] = x[i] + delta
```

```
double F1 = F (temp, n)
```

```
temp[i] = x[i] - delta
```

```
double F2 = F (temp, n)
```

```
grad[i] = (F1 - F2) / (2 * delta).
```

```
Declarar variables
```

```
int n=2 Número de variables
```

```
double x[max]={-1.0, 1.0}
```

```
double grad [max]
```

```
double delta = 1e-5, tol = 1e-6
```

```
double h=0.1 Tamaño de paso fijo
```

```
int maxIter = 1000
```

```
For iter desde 0 hasta iter < maxIter
    gradiente (x, grad, n, delta)
```

```
double norm = 0
```

```
For i desde 0 hasta i < n
```

```
norm += grad[i] * grad[i]
```

```
norm = sqrt (norm)
```

```
If (norm < tol)
```

```
For i desde 0 hasta i < n
```

```
imprimir [x[i], F(x, n)]
```

```
Imprimir ("El punto aproximado es: ")
```

```
For i desde 0 hasta i < n
```

# Programa de Eujo (método del gradiente)

(INTICIO)

```
Declarar max = 10  
Definir la función  
double F (double x, int n)  
x[0] = x, x[1] = y  
regresar  $2^*x[0]^*x[1] + 2^*x[0]$   
 $-x[0]^*x[0] - 2^*x[1]^*x[1]$ 
```

```
void gradiente (double x, grad, delta, intr)  
double temp [max]  
For desde i=0 hasta i<n  
For desde j=0 hasta j<n  
temp[i] = x[i] + delta; F1 = F(temp, n)  
temp[i] = x[i] - delta; F2 = F(temp, n)  
grad[i] = (F1 - F2) / 2 * delta
```

```
Declarar variables  
int n, maxIter = 1000  
double x[max] = {-1.0, 1.0}  
double grad[max], delta = 1e-5,  
tol = 1e-6, h = 0.1
```

```
For iter desde 0 hasta iter < maxIter  
gradiente (x, grad, n, delta)
```

```
double norm = 0  
For i desde 0 hasta i < n  
norm += grad[i] * grad[i]; norm = sqrt(norm)  
if (norm < tol)  
For i desde 0 hasta i < n  
imprimir [x[i], F(x, n)]
```

Imprimir ("El punto  
aproximado es: ")  
For i desde 0 hasta  
i < n

(FIN)