





```

    derivada_x = dfx(x_actual, y_actual)
    derivada_y = dfy(x_actual, y_actual)

    h_optimo = encontrar_h_optimo(x_actual, y_actual)

    x_siguiente = x_actual + derivada_x * h_optimo
    y_siguiente = y_actual + derivada_y * h_optimo

    IMPRIMIR "Iteración:", iteracion
    IMPRIMIR "Punto actual:", x_actual, y_actual
    IMPRIMIR "Gradiente:", derivada_x, derivada_y
    IMPRIMIR "h óptimo:", h_optimo
    IMPRIMIR "Nuevo punto:", x_siguiente, y_siguiente

    x_actual = x_siguiente
    y_actual = y_siguiente

```

IMPRIMIR "Resultado Final", x_actual, y_actual

Int main()

```

double x_inicial = -1, y_inicial = 1
int max_iter = 20, tolerancia = 1e-6

metodo_gradiente(x_inicial, y_inicial, max_iter,
                  tolerancia)

Imprimir resultados finales

```

FIN