Ejercicio 1:

2.

Gradient Descent de g1:

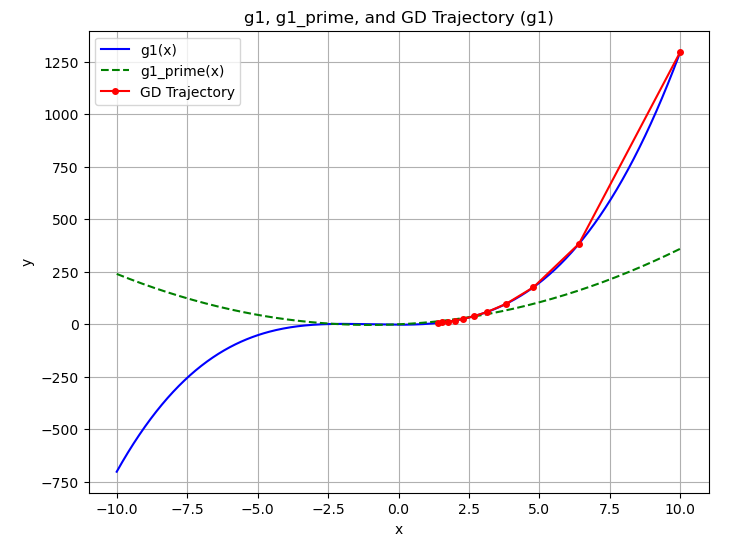
Initial x: 10

Learning Rate: 0.01

Iterations: 10

Final x\_min: 1.3916016857042504

Final g1(x\_min): 6.504579307627127



Gradient Descent de g2:

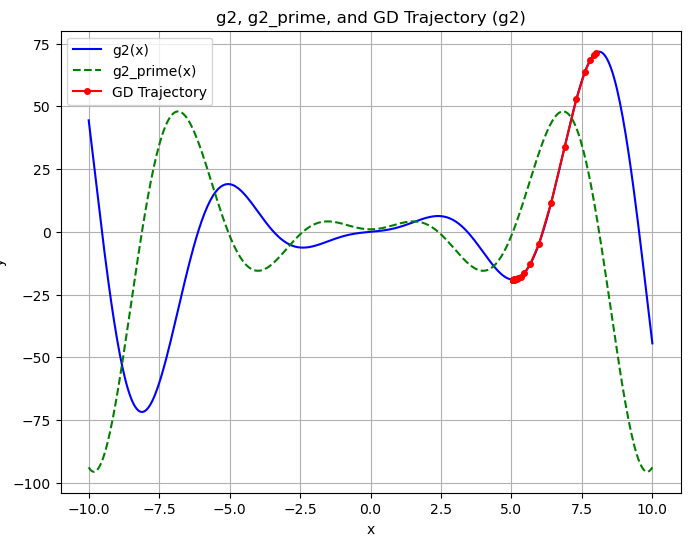
Initial x: 8

Learning Rate: 0.01

Iterations: 100

Final x\_min: 5.052856522624712

Final g2(x\_min): -19.0129678656309



Gradient Descent de g3:

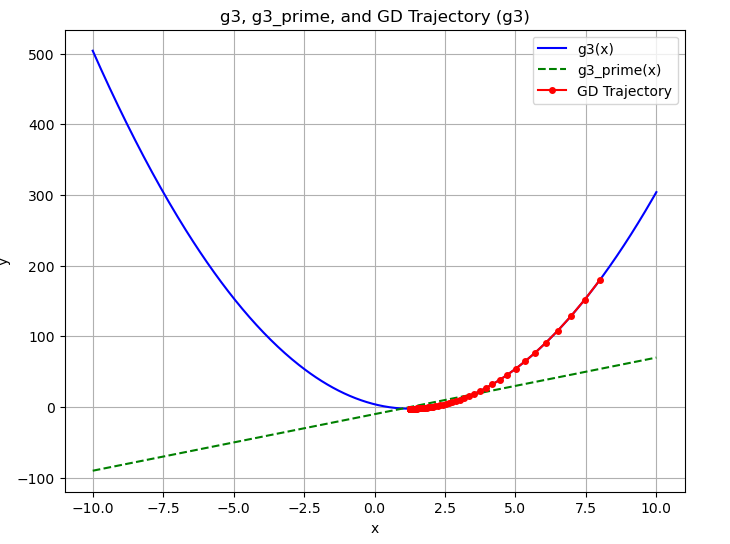
Initial x: 8

Learning Rate: 0.01

Iterations: 100

Final x\_min: 1.2516146801539347

Final g3(x\_min): -2.2499895712320024



Gradient Descent de g4:

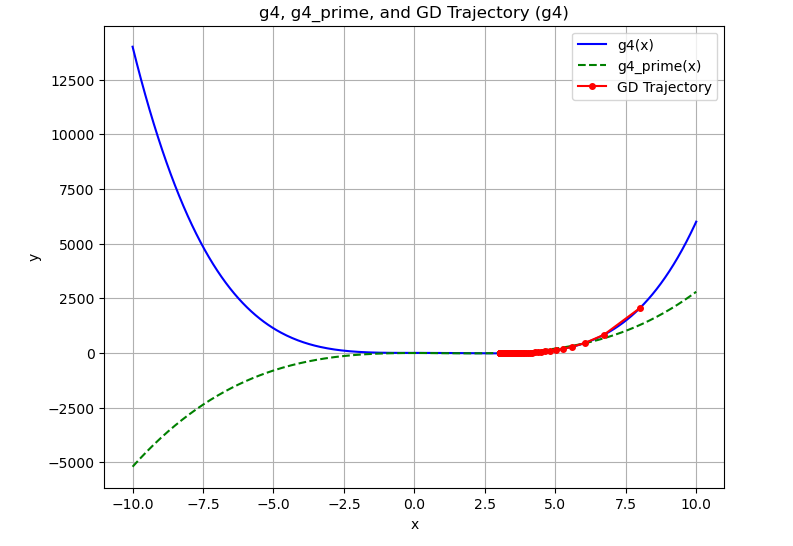
Initial x: 8

Learning Rate: 0.001

Iterations: 100

Final x\_min: 3.0243032113869823

Final g4(x\_min): -22.98925318485722



Preguntas de Reflexión:

• ¿Cómo afecta la tasa de aprendizaje a la convergencia del algoritmo?

Si es muy pequeña requiere muchas iteraciones para lograr precisión, ahora si es muy grande puede que no converja a la solución.

• ¿Qué impacto tiene el punto inicial en la localización de mínimos?

Si estamos cerca del mínimo, tenemos mas chance de encontrarlo. Si estamos lejos requeriremos una combinación de iteraciones y tasa de aprendizaje tal que nos permita alcanzarlo.

• ¿En qué funciones el algoritmo enfrenta más dificultades para encontrar el mínimo y por qué?

En las funciones monótonas.

Ejercicio 2

La derivada de h1(x) será: 2\*x - sin(x)

La derivada de h2(x) será: (2\*x + 5)\*exp(-x) - (x\*\*2 + 5\*x)\*exp(-x)

La derivada de h3(x) será: 2\*x + 1/(x + 1)

La derivada de h4(x) será: 5\*x\*\*4 - 15\*x\*\*2 + 4

Gradient Descent de h1:

Initial x: 10

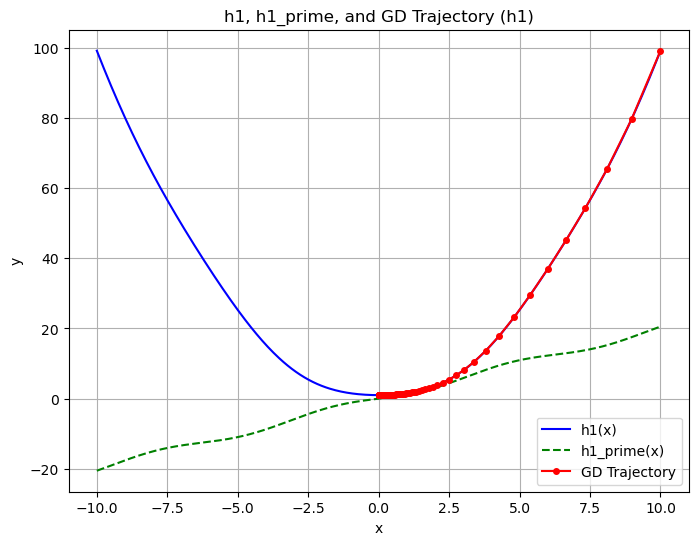
Learning Rate: 0.05

Iterations: 100

Final x\_min: 0.020248223269462372

Final h1(x\_min): 1.000205002276533

Plotting h1, h1\_prime, and gradient descent trajectory for h1...



Gradient Descent de h2:

Initial x: 1

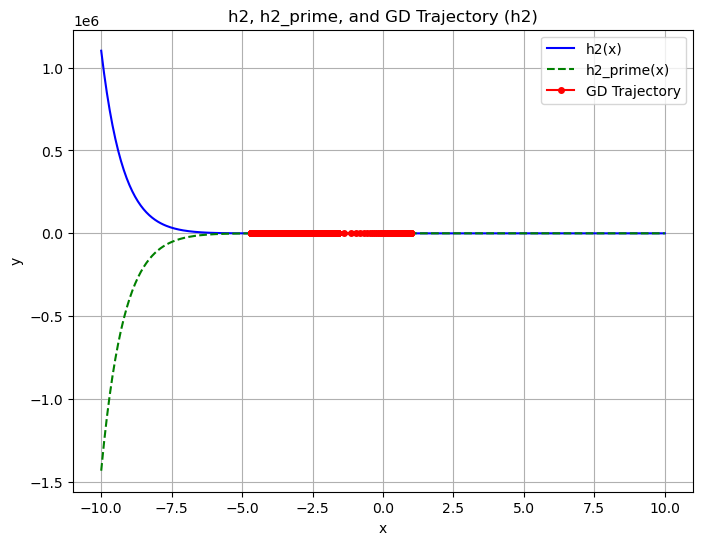
Learning Rate: 0.01

Iterations: 1000

Final x\_min: -4.520566140621624

Final h2(x\_min): -199.1491513292823

Plotting h2, h2\_prime, and gradient descent trajectory for h2...



No hay mínimo local en esta función….

Gradient Descent de h3:

Initial x: 1

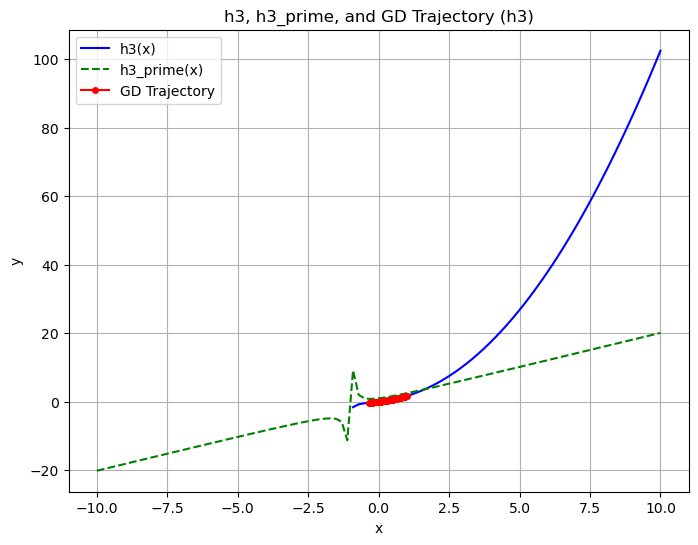
Learning Rate: 0.01

Iterations: 100

Final x\_min: -0.32715271794633516

Final h3(x\_min): -0.2892079953966418

Plotting h3, h3\_prime, and gradient descent trajectory for h3...



No hay mínimo local en esta función….

Gradient Descent de h4:

Initial x: 1

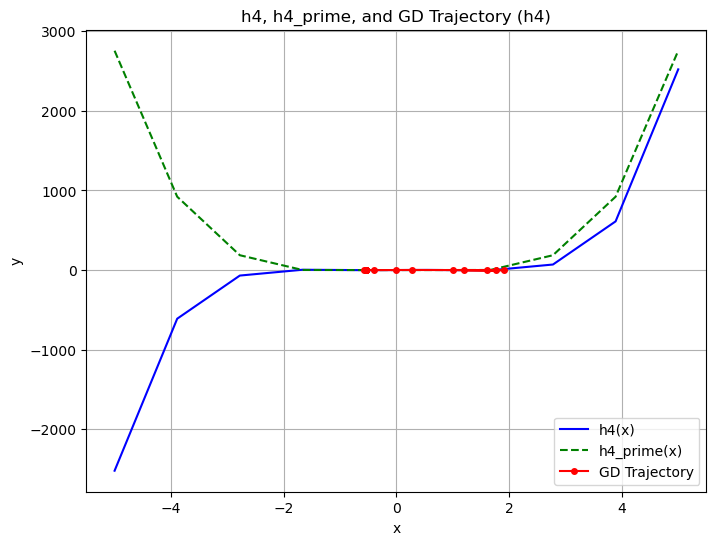
Learning Rate: 0.1

Iterations: 15000

Final x\_min: -0.543912255902338

Final h4(x\_min): -1.4186966255829239

Plotting h4, h4\_prime, and gradient descent trajectory for h4...



* ¿Qué diferencias observa entre funciones polinomiales y no polinomiales en cuanto a la efectividad del algoritmo?

En el caso de las polinomiales, el algoritmo es efectivo porque existe el mínimo local, pero en las no polinomiales, el algoritmo no avisa que no hay mínimo local.

* ¿Cómo influyen las características de las funciones en la dificultad para encontrar mínimos?

El algoritmo no converge y no lo notamos hasta no ejecutarlo repetidas veces.