Matemáticas Computacionales

Práctica 1: Gráficas de Curvas

Alumno: María Fernanda Orozco Medellín

Semestre: Febrero-Junio 2021

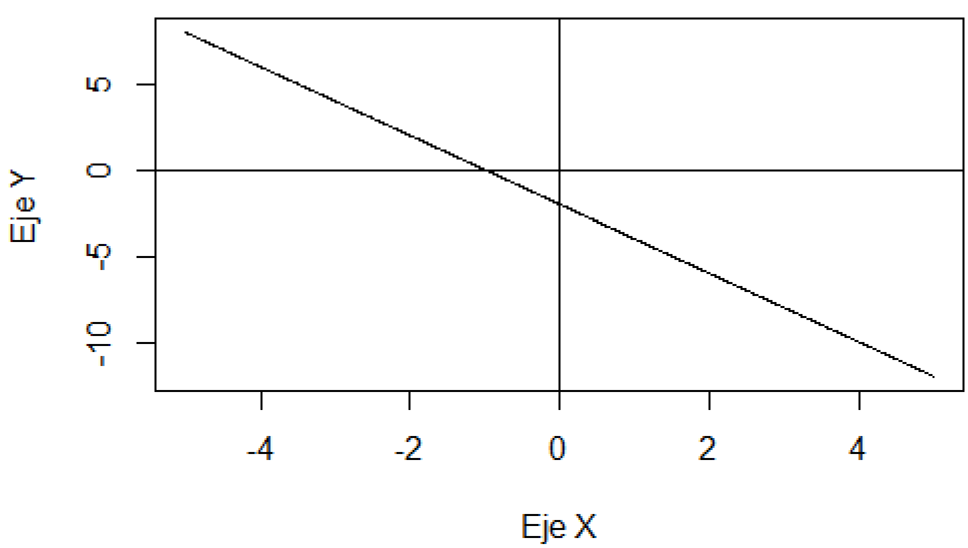
1. Introducción
2. Curvas en R^2

2.1 Línea recta

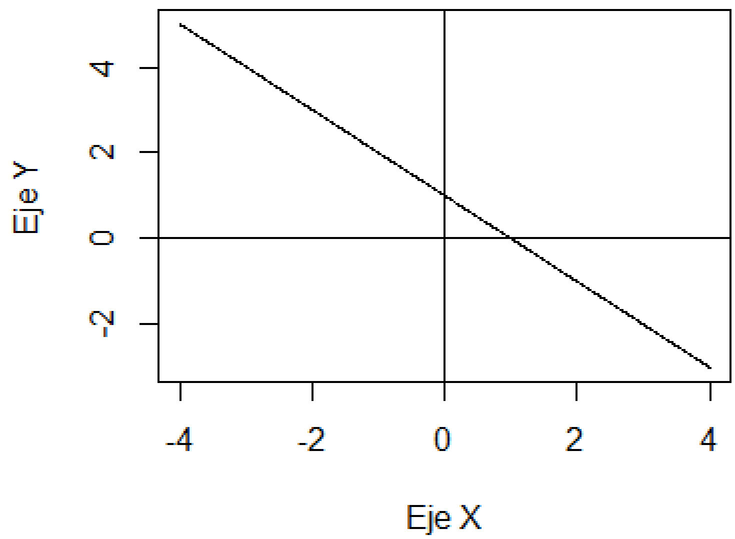
Llamamos línea recta al lugar geométrico de los puntos tales que tomados 2 puntos diferentes cualesquiera $P_1(x_1, y_1)$ y $P_2(x_1, y_2)$ del lugar, el valor de la pendiente m calculado por medio de la fórmula $m = \frac{y1-y2}{x1-x2}$, $x_1 \neq x_2$.

Se utilizará la ecuación de la forma pendiente-intersección para graficar en R.

```
#Linea recta
m <- 4 #pendiente
b <- 4 #interseccion

#funcion de la linea recta
f <- function(m, b, x){
  return(m * x + b)
}

x <- seq(-6, 6, 0.01)#vector de -6 a 6
y <- f(m, b, x) #evaluamos

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y") #graficamos
abline(h = 0, v = 0) #una linea horizontal que pasa por el 0 en las x y una linea vertical que pasa por el 0 en las y

```




```
m <- -1 #pendiente
b <- 1 #interseccion

#funcion de la linea recta
f <- function(m, b, x){
  return(m * x + b)
}

x <- seq(-4, 4, 0.01)#vector de -4 a 4
y <- f(m, b, x) #evaluamos

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y") #graficamos
abline(h = 0, v = 0) #una linea horizontal que pasa por el 0 en las x y una linea vertical que pasa por el 0 en las y

```

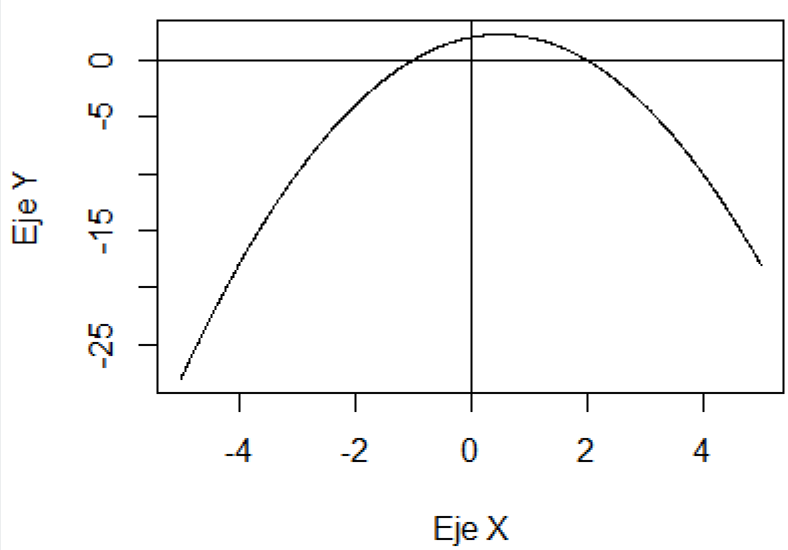


2.2 Parábola

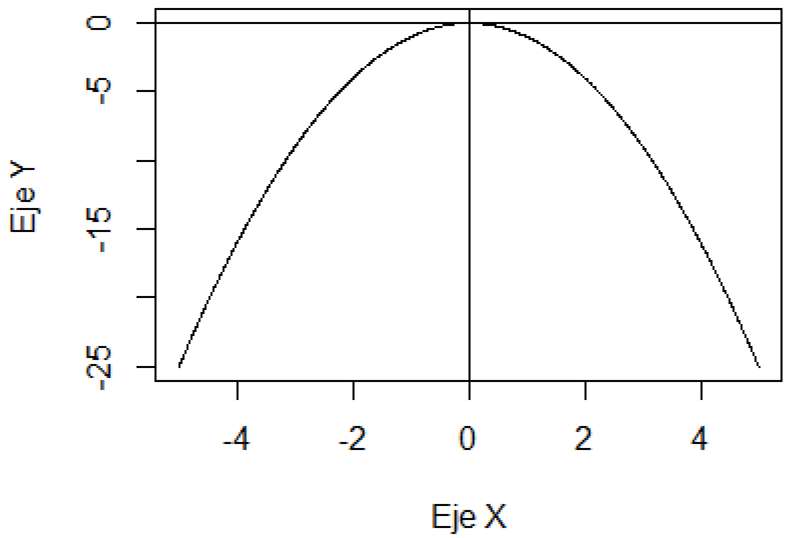
Una parábola es el lugar geométrico de un punto que se mueve en un plano de tal manera que su distancia es una recta fija, situada en el plano, es siempre igual a su distancia de un punto fijo del plano y que no pertenece a la recta. El punto fijo se llama foco y la recta fija directriz de la parábola.

La definición excluye el caso en que el foco está sobre la directriz.

```
#Parabola
g <- function(x){
  return(-x^2 + x + 2)
}

x <- seq(-5, 5, 0.01)#vector de -5 a 5
y <- g(x)

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y") #graficamos
abline(h = 0, v = 0) #una linea horizontal que pasa por el 0 en las x y una linea vertical que pasa por el 0 en las y
```



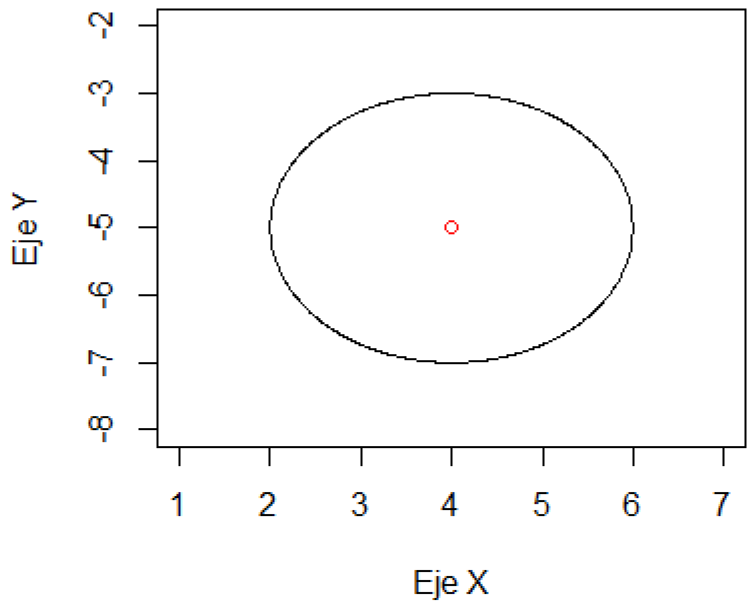

```
g <- function(x){
  return(-x^2)
}

x <- seq(-5, 5, 0.01)#vector de -5 a 5
y <- g(x)

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y") #graficamos
abline(h = 0, v = 0) #una linea horizontal que pasa por el 0 en las x y una linea vertical que pasa por el 0 en las y
```

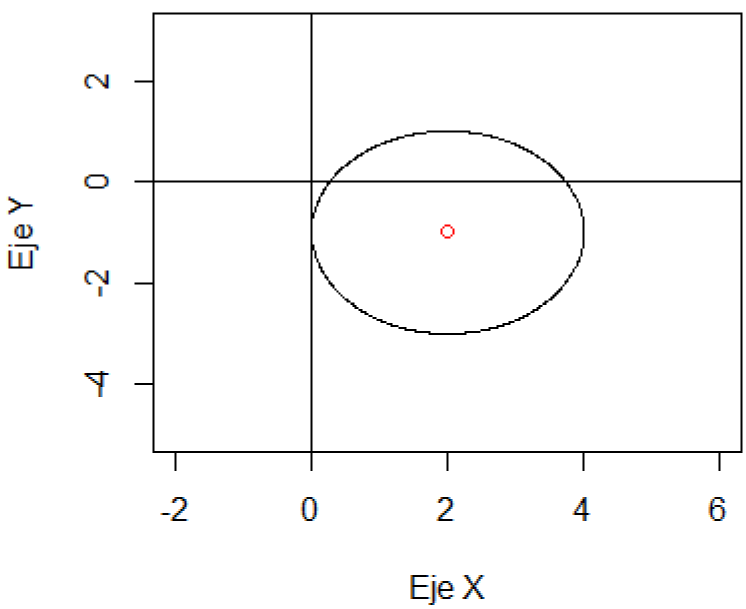


2.3 Circunferencia

La circunferencia es el lugar geométrico de un punto que se mueve en un plano de tal manera que se conserva siempre a una distancia constante de un punto fijo de ese plano. El punto fijo se llama centro de la circunferencia, y la distancia constante se llama radio.

```r
#Circunferencia
circunferencia <- function(h, k, r){
  if (r >= 0){ # r tiene que ser positivo
    if (r == 0){ # si es r = 0, entonces es un punto
      plot(x = h, y = k, xlab = "Eje X", ylab = "Eje Y") # grafica del punto
    } else{
      x <- seq(h - r, h + r, 0.01) # ya que no podemos graficar en todo R^2
      ypositiva <- k + sqrt(r^2 - ((x - h)^2)) # parte positiva de la circunferencia
      ynegativa <- k - sqrt(r^2 - ((x - h)^2)) # parte negativa de la circunferencia
      # graficamos primero la parte positiva
      plot(x, ypositiva, type = "l", xlim = c(h - (r + 2), h + (r + 2)), ylim = c(k - (r + 2), k + (r + 2)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l") # agregamos la parte negativa
      abline(h = 0, v = 0) # agregamos los ejes
      points(x = h, y = k, col = "blue") # dibujamos el centro
    }
  } else{
    return(print("El radio no es positivo."))
  }
}

# ejecutamos la funcion
circunferencia(2, -1, 2)

circunferencia(4, -5, 2)
```



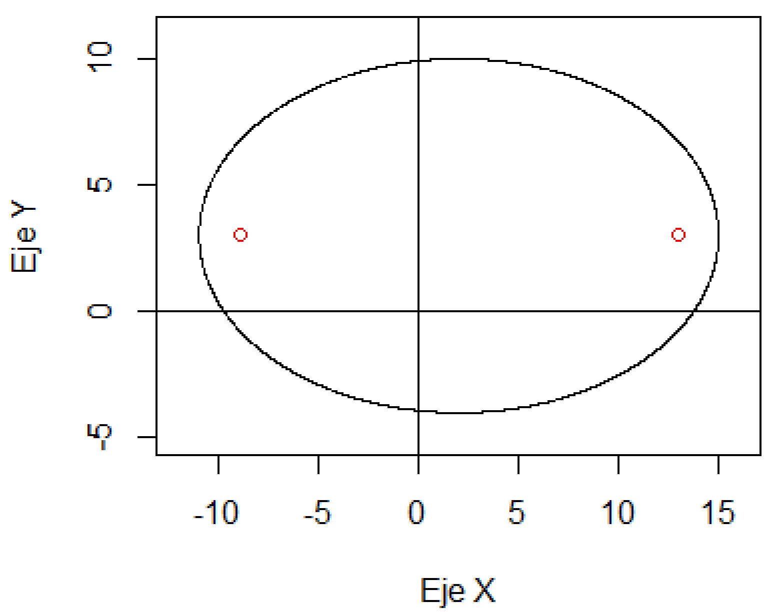


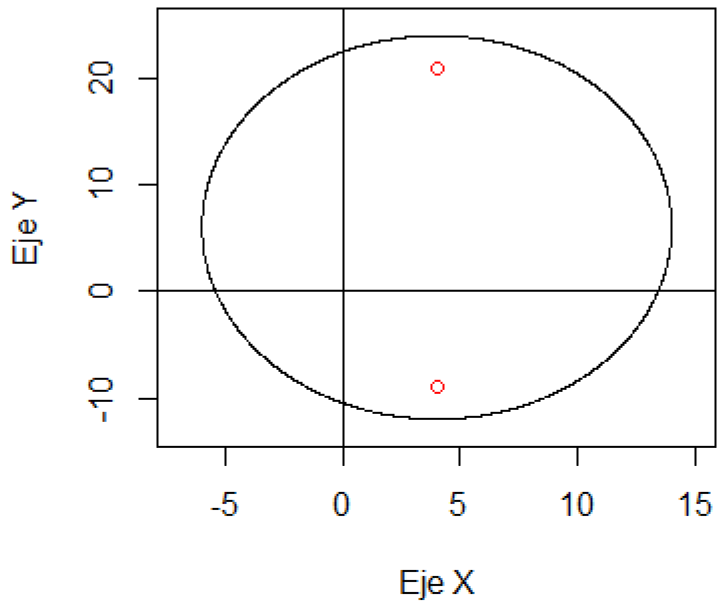
2.4 Elipse

Una elipse es el lugar geométrico de un punto que se mueve en un plano de tal manera que la suma de sus distancias a dos puntos fijos de ese plano es siempre igual a una constante, mayor que la distancia entre los dos puntos.

Los dos puntos fijos de llaman focos de la elipse. La definición de una elipse excluye el caso en que el punto móvil esté sobre el segmento que une los focos.

```r

#Elipse
elipse <- function(h, k, a, b, horizontal){
  if (a > b){ # a tiene que ser mayor que b
    c <- sqrt(a^2 - b^2) # calculamos c
    if (horizontal){ # si es una elipse horizontal
      x <- seq(h - a, h + a, 0.01) #definimos el dominio
      ypositiva <- k + sqrt((b^2 - (b^2/a^2) * ((x - h)^2))) # parte positiva
      ynegativa <- k - sqrt((b^2 - (b^2/a^2) * ((x - h)^2))) # parte negativa
      # graficamos primero la parte positiva
      plot(x, ypositiva, type = "l", xlim = c(h - (a + 1), h + (a + 1)), ylim = c(k - (b + 1), k + (b + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l") # agregamos la parte negativa
      abline(h = 0, v = 0) # ejes coordenados
      points(x = c(h - c, h + c), y = c(k, k), col = "red") # focos
    } else{
      x <- seq(h - b, h + b, 0.01)
      ypositiva <- k + sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
      ynegativa <- k - sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
      plot(x, ypositiva, type = "l", xlim = c(h - (b + 1), h + (b + 1)), ylim = c(k - (a + 1), k + (a + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l")
      abline(h = 0, v = 0)
      points(x = c(h, h), y = c(k - c, k + c), col = "red")
    }
  } else {
    return(print("No cumple las condiciones para ser una elipse. (a no es mayor que b)"))
  }
}

elipse(2, 3, 13, 7, TRUE)

elipse(4, 6, 18, 10, FALSE)
```



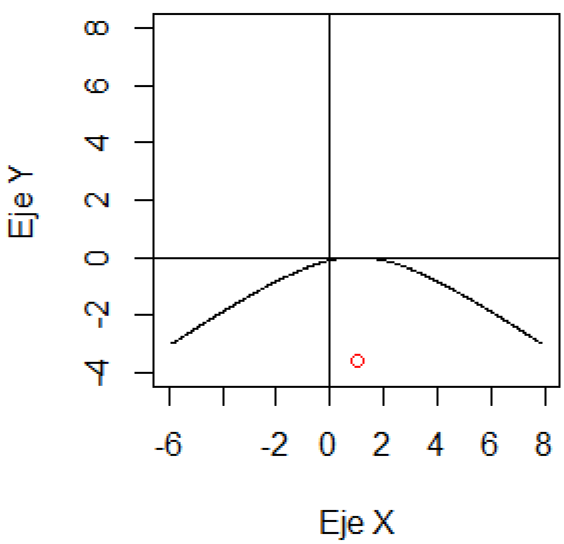


2.5 Hipérbola

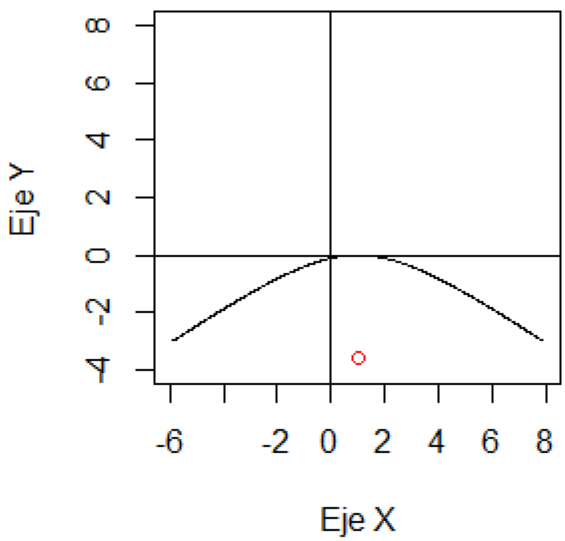
Una hipérbola es el lugar geométrico de un punto que se mueve en un plano de tal manera que el valor absoluto de la circunferencia de sus distancias a dos puntos fijos del plano, llamados focos, es siempre igual a una cantidad constante, positiva y menor que la distancia entre los focos.

La definición de la hipérbola excluye el caso en que el punto móvil se mueve sobre la recta que pasa por los focos a excepción del segmento comprendido entre ellos. Los focos y el punto medio de este segmento no pueden pertenecer al lugar geométrico.

```r
#Hiperbola
hiperbola <- function(h, k, a, b, horizontal){
  c <- sqrt(a^2 + b^2) # calculamos c
  if (horizontal){ # hiperbola sobre el eje x
    xizq <- seq(h - (a + 3), h - a, 0.01) # dominio izquierdo
    xder <- seq(h + a, h + (a + 3), 0.01) # dominio derecho
    yizqpositiva <- k + sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte positiva del dominio izquierdo
    yizqnegativa <- k - sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte negativa del dominio izquierdo
    yderpositiva <- k + sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte positiva del dominio derecho
    ydernegativa <- k - sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte negativa del dominio derecho
    # graficamos la parte positiva del dominio izquierdo
    plot(xizq, yizqpositiva, type = "l", xlim = c(h - (a + 4), h + (a + 4)), ylim = c(k - (b + 4), k + (b + 4)),
         xlab = "Eje X", ylab = "Eje Y")
    lines(xizq, yizqnegativa, type = "l") # agregamos parte negativa del dominio izquierdo
    lines(xder, yizqpositiva, type = "l")
    lines(xder, yizqnegativa, type = "l")
    abline(h = 0, v = 0) # ejes coordenados
    points(x = c(h - (a + c)), y = c(k), col = "green") # focos
  } else{ # hiperbola sobre el eje y
    yizq <- seq(k - (a + 3), k - a, 0.01) # rango inferior
    yder <- seq(k + a, k + (a + 3), 0.01) # rango superior
    xizq <- seq(k - (a + 3), k - a, 0.01) # dominio inferior
    xder <- seq(k + a, k + (a + 3), 0.01) # dominio superior
    xizqpositiva <- h + sqrt((b^2/a^2)*((yizq - k)^2) - b^2) # parte positiva del rango inferior
    xizqnegativa <- h - sqrt((b^2/a^2)*((yizq - k)^2) - b^2) # parte negativa del rango superior
    yizqpositiva <- h + sqrt((b^2/a^2)*((yizq - k)^2) - b^2) # parte positiva del dominio inferior
    yizqnegativa <- h - sqrt((b^2/a^2)*((yizq - k)^2) - b^2) # parte negativa del dominio superior
    # graficamos
    plot(xizqpositiva, yizq, type = "l", xlim = c(h - (b + 4), h + (b + 4)), ylim = c(k - (a + 4), k + (a + 4)),
         xlab = "Eje X", ylab = "Eje Y")
    lines(xizqnegativa, yizq, type = "l")
    lines(yizqpositiva, yizq, type = "l")
    lines(yizqnegativa, yizq, type = "l")
    abline(h = 0, v = 0)
    points(x = c(h), y = c(k - (a + c)), col = "green") # focos
  }
}

hiperbola(1, 2, 2, 3, FALSE)
```



```r
#Hiperbola
hiperbola <- function(h, k, a, b, horizontal){
  c <- sqrt(a^2 + b^2) # calculamos c
  if (horizontal){ # hiperbola sobre el eje x
    xizq <- seq(h - (a + 3), h - a, 0.01) # dominio izquierdo
    xder <- seq(h + a, h + (a + 3), 0.01) # dominio derecho
    yizqpositiva <- k + sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte positiva del dominio izquierdo
    yizqnegativa <- k - sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte negativa del dominio izquierdo
    # graficamos la parte positiva del dominio izquierdo
    plot(xizq, yizqpositiva, type = "l", xlim = c(h - (a + 4), h + (a + 4)), ylim = c(k - (b + 4), k + (b + 4)),
         xlab = "Eje X", ylab = "Eje Y")
    lines(xizq, yizqnegativa, type = "l") # agregamos parte negativa del dominio izquierdo
    abline(h = 0, v = 0) # ejes coordenados
    points(x = c(h - (a + c)), y = c(k), col = "red") # focos
  } else{ # hiperbola sobre el eje y
    yizq <- seq(k - (a + 3), k - a, 0.01) # rango inferior
    yder <- seq(k + a, k + (a + 3), 0.01) # rango superior
    xizqpositiva <- h + sqrt((b^2/a^2)*((yizq - k)^2) - b^2) # parte positiva del rango inferior
    xizqnegativa <- h - sqrt((b^2/a^2)*((yizq - k)^2) - b^2) # parte negativa del rango superior
    # graficamos
    plot(xizqpositiva, yizq, type = "l", xlim = c(h - (b + 4), h + (b + 4)), ylim = c(k - (a + 4), k + (a + 4)),
         xlab = "Eje X", ylab = "Eje Y")
    lines(xizqnegativa, yizq, type = "l")
    abline(h = 0, v = 0)
    points(x = c(h), y = c(k - (a + c)), col = "red") # focos
  }
}

hiperbola(1, 2, 2, 3, FALSE)
```



3. Referencias

1] Charles H Lehmann. Geometría analítica. LIMUSA, 1965.

2] https://github.com/Orozco96/Matematicas_Computacionales/tree/main/Pr%C3%A1ctica_01