

Resolución de Sistemas de Ecuaciones Lineales Python, R con Shiny

NOMBRE: Beatriz Umiña Machaca

Código: 230035

Docente: Fred Torres Cruz

05 de mayo de 2025

1. Introducción

El propósito de este proyecto es resolver sistemas de ecuaciones lineales con dos variables (x y y) utilizando métodos algebraicos: **sustitución**, **igualación** y **reducción**. El sistema fue inicialmente implementado en **Python** para su uso en consola, y posteriormente migrado y mejorado en **R** utilizando el paquete *Shiny*, para ofrecer una interfaz gráfica interactiva.

2. Primera Versión: Python (Interfaz de Consola)

Características

- Implementación en consola usando `sympy`, `re` y entrada de texto con `input()`.
- Análisis de ecuaciones con expresiones regulares (`re.match`).
- Resolución de sistemas mediante tres métodos:

```
Ingresa:  
1: 2*x-y=7  
2: x-3*y=-7  
  
Elige el método para resolver:  
1. Sustitución  
2. Igualación  
3. Reducción  
4. Salir
```

- Sustitución

```
SISTEMA DE ECUACIONES

Ingresa:
1: x+2*y=10
2: 2*x-y=5

Elige el método para resolver:
1. Sustitución
2. Igualación
3. Reducción
4. Salir
Opción: 1

Método de Sustitución:
Resultado:
x = 4
y = 3
```

- Igualación

```
Ingresa:
1: x+2*y=10
2: 2*x-y=5

Elige el método para resolver:
1. Sustitución
2. Igualación
3. Reducción
4. Salir
Opción: 2

Método de Igualación:
Resultado:
x = 4
y = 3
```

- Reducción (resolución directa con solve)

```
Ingresa:
1: x+2*y=10
2: 2*x-y=5

Elige el método para resolver:
1. Sustitución
2. Igualación
3. Reducción
4. Salir
Opción: 3

Método de Reducción:
Resultado:
x = 4
y = 3
```

- Manejo de errores en entrada y resolución.

Lógica de resolución

- Se parsean ecuaciones tipo $ax + by = c$ desde texto.
- Se despeja una variable y se sustituye en la otra ecuación.
- Se imprime el resultado paso a paso para el usuario.

Ventajas

- Rápido de desarrollar.
- Ideal para pruebas de lógica algebraica.

Limitaciones

- Requiere conocimiento básico de sintaxis matemática para ingresar las ecuaciones.
- No amigable para usuarios no técnicos.
- Poco visual e interacción limitada.

3. Segunda Versión: R con Shiny (Interfaz Gráfica Web)

Características

- Interfaz gráfica desarrollada con `shiny`, accesible vía navegador.

<http://127.0.0.1:6597>

Resolución de Sistemas de Ecuaciones Lineales

a₁ (coef. de x en Ecuación 1):

b₁ (coef. de y en Ecuación 1):

c₁ (resultado de Ecuación 1):

a₂ (coef. de x en Ecuación 2):

b₂ (coef. de y en Ecuación 2):

c₂ (resultado de Ecuación 2):

Método de resolución:
Sustitución

☒ Resolver

Resultado:
Este programa resuelve sistemas de ecuaciones lineales con dos variables usando métodos algebraicos.

center

Entradas numéricas para coeficientes a , b , c de cada ecuación.

a₁ (coef. de x en Ecuación 1):

b₁ (coef. de y en Ecuación 1):

c₁ (resultado de Ecuación 1):

a₂ (coef. de x en Ecuación 2):

b₂ (coef. de y en Ecuación 2):

c₂ (resultado de Ecuación 2):

Método de resolución:
Sustitución

Menú desplegable para elegir el método de resolución.

Método de resolución:

Sustitución

Sustitución

Igualación

Reducción

Botón para ejecutar el cálculo y mostrar resultados.

Resolución de Sistemas de Ecuaciones Lineales

a. (coef. de x en Ecuación 1):
1

b. (coef. de y en Ecuación 1):
-1

c. (resultado de Ecuación 1):
3

a. (coef. de x en Ecuación 2):
2

b. (coef. de y en Ecuación 2):
1

c. (resultado de Ecuación 2):
12

Método de resolución:
Sustitución

☒ Resolver

Resultado:

☒ Solución encontrada:
x = 5
y = 2

Este programa resuelve sistemas de ecuaciones lineales con dos variables usando métodos algebraicos.

Lógica de resolución

- Los métodos implementan la lógica matemática directamente en R:
 - En sustitución e igualación, se usan funciones y `uniroot` para encontrar raíces.
 - En reducción, se resuelve el sistema con álgebra matricial (`solve()`).

Mejoras respecto a Python

- Interfaz intuitiva y visual, sin necesidad de escribir ecuaciones.
- Mayor control de errores de entrada (valores numéricos esperados).
- Mejor presentación del resultado.
- Ideal para educación, demostraciones y uso público.

4. Conclusión

La evolución del proyecto refleja una clara mejora en accesibilidad, usabilidad y presentación. Mientras que la versión en Python demostró ser funcional y útil para propósitos técnicos, la migración a R con Shiny permitió ofrecer una solución más visual, intuitiva y apta para usuarios de todos los niveles. Este desarrollo progresivo evidencia cómo una herramienta técnica puede transformarse en una aplicación amigable y accesible, sin perder su rigor matemático.

5. Anexos

Anexo A: Código en Python

```

import re
from sympy import symbols, Eq, solve, sympify

x, y = symbols('x y')

def parsear_ecuacion(ecuacion_str):
    ecuacion_str = ecuacion_str.replace("_", "")
    match = re.match(r'(.+)=([\+|-]?[d+])', ecuacion_str)
    if match:
        izquierda = sympify(match.group(1))
        derecha = sympify(match.group(2))
        return Eq(izquierda, derecha)
    else:
        raise ValueError("La ecuación no tiene el formato esperado")

def ingreso_ecuaciones():
    print("Ingresa:")
    try:
        eq1_input = input("1: ")
        eq2_input = input("2: ")
        eq1 = parsear_ecuacion(eq1_input)
        eq2 = parsear_ecuacion(eq2_input)
        return eq1, eq2
    except Exception as e:
        print("Error al interpretar las ecuaciones:", e)
        return None, None

def metodo_sustitucion(eq1, eq2):
    print("\nMétodo de Sustitución:")
    try:
        despeje = solve(eq1, y)
        if not despeje:
            despeje = solve(eq1, x)
            sustituida = eq2.subs(x, despeje[0])
            solucion = solve(sustituida, y)
            x_valor = despeje[0].subs(y, solucion[0])
        else:
            sustituida = eq2.subs(y, despeje[0])
            solucion = solve(sustituida, x)
            y_valor = despeje[0].subs(x, solucion[0])
            x_valor = solucion[0]
        print("Resultado:")
        print("x=", x_valor)
        print("y=", y_valor if 'y_valor' in locals() else solucion[0])
    except Exception as e:
        print("No se pudo resolver por sustitución:", e)

def metodo_igualacion(eq1, eq2):
    print("\nMétodo de Igualación:")
    try:
        despeje1 = solve(eq1, y)
        despeje2 = solve(eq2, y)
        igualado = Eq(despeje1[0], despeje2[0])
        x_valor = solve(igualado, x)[0]
        y_valor = despeje1[0].subs(x, x_valor)
        print("Resultado:")
        print("x=", x_valor)
        print("y=", y_valor)
    except Exception as e:

```

```

        print("\nNo se pudo resolver por igualaci n:", e)

def metodo_reduccion(eq1, eq2):
    print("\nM todo de Reduc ci n:")
    try:
        sol = solve((eq1, eq2), (x, y))
        print("\nResultado:")
        print("x=", sol[x])
        print("y=", sol[y])
    except Exception as e:
        print("\nNo se pudo resolver por reducci n:", e)

def main():
    print("\nSISTEMA DE ECUACIONES\n")
    eq1, eq2 = ingreso_ecuaciones()
    if eq1 is None or eq2 is None:
        return

    while True:
        print("\nElige el m todo para resolver:")
        print("1. Sustituci n")
        print("2. Igualaci n")
        print("3. Reduc ci n")
        print("4. Salir")
        opcion = input("Opci n: ")

        if opcion == '1':
            metodo_sustitucion(eq1, eq2)
        elif opcion == '2':
            metodo_igualacion(eq1, eq2)
        elif opcion == '3':
            metodo_reduccion(eq1, eq2)
        elif opcion == '4':
            print("\nHasta luego. Vuelve pronto:")
            break
        else:
            print("\nopcion invalida.")

if __name__ == "__main__":
    main()

```

Anexo B: Código en R (Shiny)

```

library(shiny)
resolver_sustitucion <- function(a1, b1, c1, a2, b2, c2) {

  if (b1 == 0) stop("No se puede aplicar sustituci n si b1=0")
  despeje_y <- function(x) (c1 - a1 * x) / b1

  f <- function(x) a2 * x + b2 * despeje_y(x) - c2
  x_val <- uniroot(f, c(-1e3, 1e3))$root
  y_val <- despeje_y(x_val)
  return(c(x = round(x_val, 4), y = round(y_val, 4)))
}

resolver_igualacion <- function(a1, b1, c1, a2, b2, c2) {

  if (b1 == 0 || b2 == 0) stop("No se puede aplicar igualaci n si b1 o b2=0")
  y1 <- function(x) (c1 - a1 * x) / b1
  y2 <- function(x) (c2 - a2 * x) / b2

```

```

f <- function(x) y1(x) - y2(x)
x_val <- uniroot(f, c(-1e3, 1e3))$root
y_val <- y1(x_val)
return(c(x = round(x_val, 4), y = round(y_val, 4)))
}

resolver_reduccion <- function(a1, b1, c1, a2, b2, c2) {
  mat <- matrix(c(a1, b1, a2, b2), ncol = 2, byrow = TRUE)
  vec <- c(c1, c2)
  sol <- solve(mat, vec)
  return(c(x = round(sol[1], 4), y = round(sol[2], 4)))
}

ui <- fluidPage(
  titlePanel("Resolución de Sistemas de Ecuaciones Lineales"),

  sidebarLayout(
    sidebarPanel(
      numericInput("a1", "a (coef. de x en Ecuación 1):", value = 1),
      numericInput("b1", "b (coef. de y en Ecuación 1):", value = 1),
      numericInput("c1", "c (resultado de Ecuación 1):", value = 1),
      numericInput("a2", "a (coef. de x en Ecuación 2):", value = 1),
      numericInput("b2", "b (coef. de y en Ecuación 2):", value = 1),
      numericInput("c2", "c (resultado de Ecuación 2):", value = 1),
      selectInput("metodo", "Método de resolución:",
        choices = c("Sustitución", "Igualación", "Reducción")),
      actionButton("resolver", "Resolver")
    ),

    mainPanel(
      h4("Resultado:"),
      verbatimTextOutput("resultado"),
      tags$br(),
      tags$p("Este programa resuelve sistemas de ecuaciones lineales con dos variables usando todos los métodos algebraicos.")
    )
  )
)

server <- function(input, output) {
  observeEvent(input$resolver, {
    a1 <- input$a1; b1 <- input$b1; c1 <- input$c1
    a2 <- input$a2; b2 <- input$b2; c2 <- input$c2
    metodo <- input$metodo

    resultado <- tryCatch({
      if (metodo == "Sustitución") {
        res <- resolver_sustitucion(a1, b1, c1, a2, b2, c2)
      } else if (metodo == "Igualación") {
        res <- resolver_igualacion(a1, b1, c1, a2, b2, c2)
      } else {
        res <- resolver_reduccion(a1, b1, c1, a2, b2, c2)
      }
      paste0("Solución encontrada: x=", res["x"], " y=", res["y"])
    }, error = function(e) {
      paste("Error:", e$message)
    })

    output$resultado <- renderText({ resultado })
  })
}

```



```
  })  
}  
  
shinyApp(ui = ui, server = server)
```