



UNIVERSIDAD BANCARIA DE MÉXICO

"Constancia Unidad y trabajo"

INGENIERÍA EN SISTEMAS COMPUTACIONALES

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS DE LA SECRETARÍA DE
EDUCACIÓN PÚBLICA No. 2022241 DE FECHA 13 DE SEPTIEMBRE DE 2002.

NOMBRE DE LA MATERIA:

Métodos numéricos

NOMBRE DEL PROFESOR(A) :

Mauricio Gómez Gallegos

CUATRIMESTRE:

6to Cuatrimestre

TÍTULO DEL TRABAJO O INVESTIGACIÓN:

Documentación Programa Interpolación Lagrange

NOMBRE DE ALUMNO(S) :

Francisco de Jesus Pincle Puente



FECHA DE ENTREGA:

10 de junio 2025

Nombre del programa: Aplicación de Interpolación de Lagrange

INTRODUCCIÓN

Esta aplicación tiene como finalidad calcular e ilustrar paso a paso el polinomio de interpolación de Lagrange a partir de un conjunto de puntos ingresados por el usuario. Incluye:

- Entrada de datos en formato libre.
- Visualización detallada del procedimiento matemático.
- Representación gráfica del polinomio generado.
- Validación de errores comunes con retroalimentación visual.
- Interfaz con diseño temático estilo Gamer.

Requisitos del Sistema

- Ruby instalado (versión 2.5 o superior recomendada)
- Biblioteca GTK+3 (gtk3) instalada

Instala usando: **gem install gtk3**

Cómo Iniciar la Aplicación

1. Guarda el script con extensión **.rb**, por ejemplo: **Interpolación.rb**.
2. Ejecuta el archivo desde la terminal: **ruby Interpolación.rb**
3. La ventana de la aplicación se abrirá automáticamente.

Interfaz de Usuario

La ventana se divide en dos paneles principales:

Elemento	Descripción
Entrada de puntos	Campo de texto para introducir los puntos (x, y).
Botón Calcular y Graficar	Ejecuta el cálculo y genera la gráfica.
Etiqueta de error	Muestra errores en color rojo si hay problemas de formato.
Área de dibujo	Representa visualmente el polinomio resultante.

Panel Derecho (Procedimiento y Nota)

Elemento	Descripción
Texto scrollable	Muestra el procedimiento matemático paso a paso.
Etiqueta de nota	Guía visual con instrucciones, advertencias y formato de entrada correcto.

Interpolación LAGRANGE

Puntos de entrada (x,y):

Ej: (1,2);(2,3);(4,1)

Calcular y Graficar

Nota:

Escribe puntos como (x,y);(x2,y2);...

No repitas valores de x.

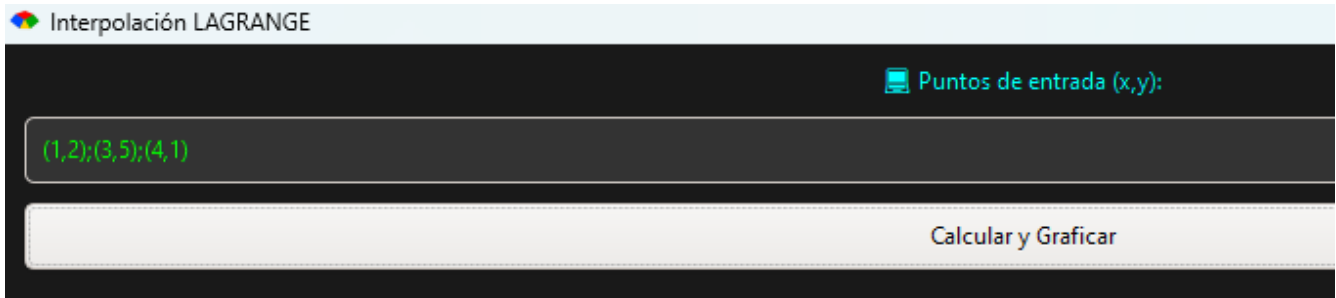
Ejemplo válidos: (1,2);(3,5);(4,1)

Estilo visual: Gamer Mode Activado

Ingresa de Datos

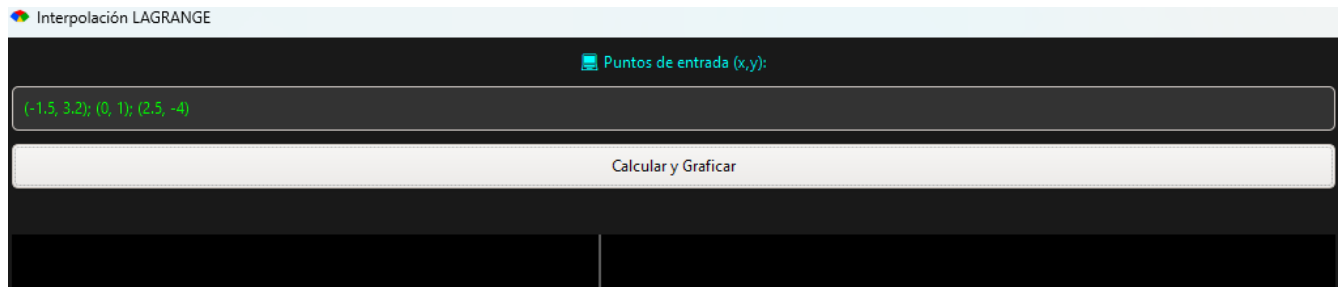
Formato correcto:

- Escribe los puntos separados por punto y coma ;, por ejemplo:



The screenshot shows the 'Interpolación LAGRANGE' application interface. At the top, there is a title bar with a small icon and the text 'Interpolación LAGRANGE'. Below this, there is a dark grey area with a label 'Puntos de entrada (x,y):' and a small icon of a document. A text input field below this label contains the string '(1,2);(3,5);(4,1)' in green text. At the bottom of the dark grey area, there is a light grey button with the text 'Calcular y Graficar'.

- Acepta números decimales y negativos:



The screenshot shows the 'Interpolación LAGRANGE' application interface. At the top, there is a title bar with a small icon and the text 'Interpolación LAGRANGE'. Below this, there is a dark grey area with a label 'Puntos de entrada (x,y):' and a small icon of a document. A text input field below this label contains the string '(-1.5, 3.2); (0, 1); (2.5, -4)' in green text. At the bottom of the dark grey area, there is a light grey button with the text 'Calcular y Graficar'.

Errores comunes detectados automáticamente:

- Valores de x repetidos (no permitidos).
- Formato incorrecto de punto, por ejemplo:



The screenshot shows the 'Interpolación LAGRANGE' application interface. At the top, there is a title bar with a small icon and the text 'Interpolación LAGRANGE'. Below this, there is a dark grey area with a label 'Puntos de entrada (x,y):' and a small icon of a document. A text input field below this label contains the string '(1;2)' in green text. At the bottom of the dark grey area, there is a light grey button with the text 'Calcular y Graficar'. Below the button, there is a red error message: 'Error: Formato inválido: '(1''.

¿Qué Hace el Programa?

1. Procesamiento de Datos:

- Analiza cada punto ingresado.
- Valida la unicidad de x.
- Convierte los puntos a valores numéricos reales.

2. Cálculo del Polinomio:

- Usa la fórmula de interpolación de Lagrange:

$$P(x) = \sum_{i=0}^n y_i \cdot \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

3. Procedimiento Paso a Paso:

- Muestra cómo se construye cada término $L_i(x)$.
- Detalla los numeradores, denominadores y el producto final con y_i .
- Presenta el polinomio completo ensamblado.

4. Gráfica:

- Dibuja el polinomio sobre ejes cartesianos (fondo negro).
- Representa ejes X e Y en gris.
- Traza la curva en verde brillante.
- Escala fija: X de -200 a 200, Y centrado.

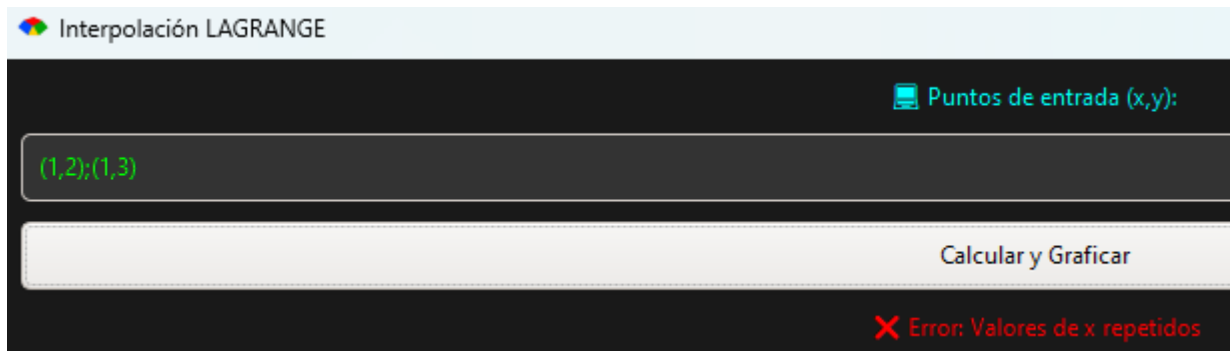
Gráfica – Detalles Técnicos

- Resolución del área de dibujo: **900x600 píxeles**.
- Centro del gráfico: punto (450, 300) en el canvas.
- Cada unidad de x representa 1 píxel horizontal.
- Cada unidad de y representa 1 píxel vertical (invertido para representar el eje positivo hacia arriba).
- La curva se dibuja en tiempo real.

Validación y Manejo de Errores

Cuando se detecta un error:

- Se muestra una alerta en rojo debajo del botón.
- Se evita el cálculo y el área gráfica no se actualiza.
- Ejemplos de mensajes:
 - Error: Formato inválido: '(1,2'
 - Error: Valores de x repetidos



The screenshot shows a software window titled "Interpolación LAGRANGE". Inside, there is a label "Puntos de entrada (x,y):" with a corresponding input field containing the text "(1,2);(1,3)". Below the input field is a button labeled "Calcular y Graficar". At the bottom of the window, a red error message is displayed: "Error: Valores de x repetidos".

Notas Importantes

- No se permite ingresar dos puntos con el mismo valor de x.
- El sistema no resuelve automáticamente raíces ni factores, pero muestra toda la construcción simbólica.
- La interpolación es precisa, pero la gráfica puede estar limitada por el rango visual.

Código del Programa

Interpolación.rb

```
C: > Users > puent > OneDrive > Documentos > FRANK > Mau > Pincle_Puente_Francisco_de_Jesus_Programa_7 > Interpolación.rb > ...

1  require 'gtk3' # Importa la biblioteca GTK+3 para crear interfaces gráficas en Ruby
2
3  # === FUNCIÓN PARA CALCULAR LA INTERPOLACIÓN DE LAGRANGE EN UN PUNTO X ===
4  def lagrange(x, points)
5    # Suma cada término del polinomio de Lagrange
6    points.each_with_index.reduce(0.0) do |total, ((xi, yi), i)|
7      # Calcula el término  $L_i(x)$ 
8      term = points.each_with_index.reduce(yi) do |prod, ((xj, _), j)|
9        i == j ? prod : prod * (x - xj) / (xi - xj)
10     end
11     total + term
12   end
13 end
14
15 # === FUNCIÓN PARA MOSTRAR EL PROCEDIMIENTO PASO A PASO DE LAGRANGE ===
16 def lagrange_steps(points)
17   steps = "=== DESARROLLO PASO A PASO ===\n\n"
18
19   points.each_with_index do |(xi, yi), i|
20     numerators = []
21     denominators = []
22     explanation = "Paso #{i+1}: Calculamos el término  $L_{\#{i}}(x)$ :\n"
23
24     # Construye numeradores y denominadores del término  $L_i(x)$ 
25     points.each_with_index do |(xj, _), j|
26       next if i == j
27       numerators << "(x - #{xj})"
28       denominators << "(#{xi} - #{xj})"
29     end
30
31     # Muestra el término completo de  $L_i(x)$ 
32     li_expr = numerators.join(" * ") + " / " + denominators.join(" * ")
33     explanation += " $L_{\#{i}}(x) = \#{li\_expr}$ \n"
34     explanation += "Multiplicamos por  $y_{\#{i}} = \#{yi}$ :\n"
35     explanation += "Término completo:  $\#{yi} * (\#{li\_expr})$ \n\n"
36     steps += explanation
37   end
38 end
```

```

39 # Construye y muestra el polinomio completo
40 steps += "=== POLINOMIO FINAL ===\nP(x) = " + points.each_with_index.map do |(xi, yi), i|
41     numerators = []
42     denominators = []
43     points.each_with_index do |(xj, _), j|
44         next if i == j
45         numerators << "(x - #{xj})"
46         denominators << "({xi} - #{xj})"
47     end
48     "#{yi} * (#{numerators.join(" * ")}) / #{denominators.join(" * ")}"
49 end.join(" + ") + "\n\n"
50
51 steps
52 end
53
54 # === CREACIÓN DE LA APLICACIÓN GTK ===
55 app = Gtk::Application.new("com.gamer.lagrange", :flags_none)
56
57 # Cuando se inicia la aplicación...
58 app.signal_connect("activate") do |application|
59     # Crea la ventana principal
60     window = Gtk::ApplicationWindow.new(application)
61     window.set_title("Interpolación LAGRANGE") # Título de la ventana
62     window.set_default_size(1400, 750) # Tamaño inicial de la ventana
63     window.set_border_width(10) # Borde interior
64
65     grid = Gtk::Box.new(:horizontal, 10) # Contenedor principal horizontal
66     window.override_background_color(:normal, Gdk::RGBA::new(0.1, 0.1, 0.1, 1)) # Fondo gamer oscuro
67
68     # === PANEL IZQUIERDO: Entrada, botón y gráfico ===
69     left_panel = Gtk::Box.new(:vertical, 10)
70
71     entry = Gtk::Entry.new
72     entry.placeholder_text = "Ej: (1,2);(2,3);(4,1)" # Texto de ayuda
73     entry.override_background_color(:normal, Gdk::RGBA::new(0.2, 0.2, 0.2, 1)) # Color de fondo
74     entry.override_color(:normal, Gdk::RGBA::new(0, 1, 0, 1)) # Color del texto (verde)
75

```



```

76 button = Gtk::Button.new(label: "Calcular y Graficar") # Botón principal
77 button.override_background_color(:normal, Gdk::RGBA::new(0.3, 0.3, 0.3, 1)) # Fondo gris oscuro
78 button.override_color(:normal, Gdk::RGBA::new(0, 0, 0, 1)) # Texto negro
79
80 drawing_area = Gtk::DrawingArea.new
81 drawing_area.set_size_request(900, 600) # Área de dibujo para graficar
82
83 error_label = Gtk::Label.new # Etiqueta para mostrar errores
84 error_label.override_color(:normal, Gdk::RGBA::new(1, 0, 0, 1)) # Color rojo
85
86 # Agrega widgets al panel izquierdo
87 left_panel.pack_start(Gtk::Label.new("■ Puntos de entrada (x,y):").tap { |l| l.override_color(:normal, Gdk::RGBA::new(0, 1, 1, 1)) }, expand: false)
88 left_panel.pack_start(entry, expand: false, fill: false, padding: 0)
89 left_panel.pack_start(button, expand: false, fill: false, padding: 0)
90 left_panel.pack_start(error_label, expand: false, fill: false, padding: 0)
91 left_panel.pack_start(drawing_area, expand: true, fill: true, padding: 0)
92
93 # === PANEL DERECHO: Procedimiento y nota ===
94 right_panel = Gtk::Box.new(:vertical, 10)
95 scrolled_window = Gtk::ScrolledWindow.new
96 scrolled_window.set_policy(:automatic, :automatic)
97
98 text_view = Gtk::TextView.new # Área de texto para mostrar el procedimiento paso a paso
99 text_view.editable = false
100 text_view.override_background_color(:normal, Gdk::RGBA::new(0.15, 0.15, 0.15, 1))
101 text_view.override_color(:normal, Gdk::RGBA::new(0, 1, 0, 1))
102 text_view.set_wrap_mode(:word)
103
104 scrolled_window.add(text_view)
105
106 note_label = Gtk::Label.new("""
107 ✨ Nota:
108 • Escribe puntos como (x,y);(x2,y2);...
109 • No repitas valores de x.
110 • Ejemplo válido: (1,2);(3,5);(4,1)
111 • Estilo visual: Gamer Mode Activado
112 """)

```

```

113 note_label.override_color(:normal, Gdk::RGBA::new(1, 1, 1, 1)) # Blanco
114
115 right_panel.pack_start(scrolled_window, expand: true, fill: true, padding: 0)
116 right_panel.pack_start(note_label, expand: false, fill: false, padding: 10)
117
118 # === LÓGICA DEL BOTÓN ===
119 button.signal_connect("clicked") do
120   begin
121     raw_input = entry.text.strip # Obtiene el texto ingresado
122     points = raw_input.split(';').map do |pair|
123       match = pair.strip.match[/\((?!\s*(-?\d+(?:\.\d+)?)\s*,\s*(-?\d+(?:\.\d+)?)\s*\)\)/]
124       raise "Formato inválido: '#{pair}'" unless match
125       [match[1].to_f, match[3].to_f] # Convierte a float
126     end
127
128     xs = points.map(&:first)
129     raise "Valores de x repetidos" if xs.uniq.size != xs.size # Evita duplicados
130
131     error_label.text = "" # Limpia errores anteriores
132     text_view.buffer.text = lagrange_steps(points) # Muestra procedimiento
133
134     # === DIBUJAR LA GRÁFICA ===
135     drawing_area.signal_connect("draw") do |_, cr|
136       cr.set_source_rgb(0, 0, 0) # Fondo negro
137       cr.paint
138
139       cr.set_source_rgb(0.7, 0.7, 0.7) # Color gris para ejes
140       cr.set_line_width(1)
141       cr.move_to(450, 0)
142       cr.line_to(450, 600)
143       cr.move_to(0, 300)
144       cr.line_to(900, 300)
145       cr.stroke
146

```

```

147 cr.set_source_rgb(0, 1, 0) # Color verde para la curva
148 cr.set_line_width(2)
149 prev = nil
150
151 # Recorre valores de x entre -200 y 200
152 (-200..200).step(1) do |x|
153   y = lagrange(x, points)
154   next if y.nan? || y.infinite?
155
156   xp = 450 + x # Ajuste horizontal
157   yp = 300 - y # Ajuste vertical
158   if prev
159     cr.move_to(*prev)
160     cr.line_to(xp, yp)
161   end
162   prev = [xp, yp]
163 end
164 cr.stroke
165 end
166 drawing_area.queue_draw # Refresca el área de dibujo
167
168 rescue => e
169   error_label.text = "✖ Error: #{e.message}" # Muestra errores
170 end
171 end
172
173 # Agrega ambos paneles al contenedor principal
174 grid.pack_start(left_panel, expand: true, fill: true, padding: 0)
175 grid.pack_start(right_panel, expand: true, fill: true, padding: 0)
176
177 window.add(grid) # Agrega la grilla a la ventana
178 window.show_all # Muestra todos los elementos
179 end
180
181 app.run # Inicia la aplicación GTK

```