

Laboratorio Calificado 09

2025-2

ÍNDICE

1	Instrucciones	3
2	Contexto.....	3
3	Objetivos.....	4
4	Requisitos.....	4
5	Entregables.....	5
6	Rúbrica de Evaluación.....	5

Laboratorio Calificado 09

Visualización de Gráficos en el Sistema de Gestión de Ensamblaje en una Línea de Producción Automatizada de Mecatrónica

1 Instrucciones

Se ha creado la tarea “Laboratorio Calificado 9” en la sección Laboratorio Calificado 9 de la página del curso-horario en PAIDEIA. Cada alumno deberá entregar un archivo comprimido (formato ZIP) con todos los archivos del desarrollo realizado en el laboratorio. El nombre del archivo comprimido deberá tener el formato:

L9_<Código del alumno de 8 dígitos>.<Extensión del archivo>

Como ejemplo, el nombre de archivo del alumno 20236969 sería “L9_20236969.zip”. Es COMPLETA RESPONSABILIDAD DE CADA ALUMNO el colocar nombres correctos a sus archivos y evitar así confusiones al momento de la calificación.

Está permitido usar cualquier material disponible en PAIDEIA y la documentación oficial de Microsoft. No está permitido el uso de materiales externos, IA o código de otros alumnos. De detectarse coincidencias, todos los implicados obtendrán la nota inapelable de cero.

2 Contexto

El Sistema de Gestión de Ensamblaje en una Línea de Producción Automatizada de Mecatrónica permite gestionar diversos elementos del entorno industrial, como operadores, máquinas y sus respectivos procesos.

Hasta el momento, el sistema cuenta con las capas de modelo, control, persistencia y presentación (GUI) completamente integradas.

En esta nueva etapa, se busca incorporar una funcionalidad gráfica que permita visualizar el rendimiento operativo de las máquinas de forma visual e intuitiva mediante una representación en coordenadas X e Y.

Para este fin, se empleará una función matemática de tipo sigmoide, la cual modela un comportamiento de crecimiento progresivo hasta un punto de saturación, típico en sistemas reales donde la eficiencia aumenta con la experiencia o el tiempo de operación:

$$y = \frac{100}{1 + e^{-k(x-x_0)}}$$

donde:

- x representa las horas o ciclos operativos de la máquina,
- y representa el rendimiento en porcentaje (%),
- k controla la tasa de crecimiento (pendiente de la curva), y
- x_0 indica el punto de inflexión (cuando se alcanza el 50% del rendimiento máximo).

3 Objetivos

Implementar una interfaz gráfica que permita:

- Representar un plano cartesiano **X–Y** dentro de la GUI.
- Graficar la función de rendimiento $y = 100/(1 + e^{-k(x-x_0)})$ para un rango de valores de x .
- Permitir la introducción de los parámetros k y x_0 desde la interfaz.
- Asociar la gráfica con una máquina registrada en el sistema.

4 Requisitos

a) Implementación del módulo de gráfico (4 puntos)

- Definir una clase que permita calcular los valores de la función sigmoide para un rango definido de x .
- Usar estructuras de datos adecuadas (listas o vectores) para almacenar los pares (x, y) .
- La clase debe ser modular, con métodos claros para el cálculo y recuperación de los valores generados.
- En C++/CLI, se puede calcular el valor de e elevado a una potencia utilizando la función:

```
double resultado = Math::Exp(-k * (x - x0));
```

donde Math::Exp(valor) retorna e^{valor} .

b) Diseño de la GUI para graficar (12 puntos)

- Crear una sección o formulario dentro de la GUI en C++/CLI (Windows Forms) que permita visualizar un plano cartesiano.
- Dibujar la curva de la función sigmoide con los valores generados.
- Incluir controles para la introducción de los parámetros k y x_0 .
- Incorporar botones como “Graficar” y “Limpiar”.
- Permitir seleccionar una máquina existente (por nombre o ID) para asociar la gráfica de rendimiento.

Orientación sobre la escala de dibujo:

- Para que el gráfico sea visible dentro del formulario, debe aplicarse una escala de conversión entre los valores reales y los píxeles del área de dibujo.
- Por ejemplo, si se usa un `PictureBox` de 400 px de ancho y 300 px de alto, puede aplicarse la siguiente escala:

- Escala horizontal (x): $px = x \times 20$

(para representar el rango de x de 0 a 20 en 400 píxeles)

- Escala vertical (y): $py = 300 - (y \times 3)$

(para invertir el eje Y y escalar el rendimiento entre 0 y 100 a 300 píxeles de altura)

c) Ejecución, validación y pruebas (4 puntos)

- Probar la funcionalidad con al menos una máquina real registrada en el sistema.
- Incluir capturas de pantalla del gráfico generado para distintos valores de k y x_0 .
- Verificar que el programa compile, ejecute correctamente y que el código esté bien comentado y estructurado.

5 Entregables

- Código fuente completo en C++/CLI, bien comentado.
- Capturas de pantalla del sistema funcionando.

6 Rúbrica de Evaluación

Puntaje Máximo: 20 puntos

Criterio	Excelente (100%)	Bueno (75%)	Regular (50%)	Deficiente (25%)	No Cumple (0%)
a) Implementación del módulo de gráfico (4 pts)	Calcula correctamente los pares (x, y) para un rango definido, usando estructuras de datos adecuadas y métodos modulares. (4 pts)	Cálculo correcto con estructura mejorable. (3 pts)	Cálculo parcial o con errores menores. (2 pts)	Cálculo incompleto o erróneo. (1 pt)	No se implementa el módulo. (0 pts)
b) Diseño de la GUI para graficar (12 pts)	GUI completa y funcional con plano cartesiano, controles para k y x_0 , botones operativos y asociación con máquina. (10-12 pts)	GUI funcional con errores menores en interacción o escala. (7-9 pts)	GUI con funcionalidades limitadas o visualización imperfecta. (4-6 pts)	GUI incompleta o con fallas en la representación gráfica. (1-3 pts)	No se implementa la GUI o no permite graficar. (0 pts)
c) Ejecución, validación y pruebas (4 pts)	Se prueba con una máquina real, incluye varias capturas con diferentes parámetros y código bien documentado. (4 pts)	Pruebas funcionales con documentación básica. (3 pts)	Pruebas parciales o poco documentadas. (2 pts)	Pruebas incompletas o sin evidencia suficiente. (1 pt)	No se realizan pruebas ni documentación. (0 pts)