

# Universidad de las Fuerzas Armadas - ESPE



Departamento de Ciencias de la Computación (DCCO)

Carrera de Ingeniería de Software

Computación Gráfica

**Tema:** Dibujo de líneas consecutivas usando .Net

**Estudiante:**

Sagnay Pilamunga Luis Eduardo

**Docente:**

Ing. Morales Caixa Dario Javier

**Fecha:**

2025-05-07

## **1. Introducción**

Este informe documenta el desarrollo de una aplicación interactiva en C# utilizando Windows Forms, cuyo objetivo es permitir al usuario dibujar líneas de manera dinámica dentro de un área gráfica. El proyecto responde a la necesidad de comprender la manipulación de eventos, dibujo gráfico y estructura modular del código orientado a objetos en un entorno visual.

El programa fue diseñado con un enfoque en la interacción simple: el usuario hace clic en la superficie, y cada clic genera una línea conectada al anterior, comenzando siempre desde el punto de origen (0,0). Además, se incluye un botón de limpieza para reiniciar el dibujo.

## **2. Objetivo**

Desarrollar una aplicación en C# que permita al usuario dibujar líneas conectadas mediante clics en un PictureBox, iniciando desde el punto (0,0) y extendiéndose con cada clic, incluyendo funcionalidad para borrar el contenido.

## **3. Desarrollo**

El proyecto está compuesto por dos clases principales:

### **3.1 Clase Line (Modelo de lógica de dibujo)**

Encapsula la lógica de manejo de puntos y dibujo. Administra una lista de puntos (`List<PointF>`) que representan las coordenadas por donde se deben trazar líneas. Al inicializarse, contiene el punto (0,0) como origen. Proporciona métodos para:

- Agregar puntos (`AddPoint`)
- Dibujar los puntos y líneas (`DrawAll`)
- Limpiar la lista de puntos y reiniciar el dibujo (`Clear`)

Esta clase asegura que el dibujo siempre se realice en función de los puntos acumulados, respetando el orden y la conexión entre ellos.

### **3.2 Clase FrmLines (Interfaz gráfica)**

Es el formulario principal que contiene:

- Un PictureBox como superficie de dibujo.
- Un botón Clear para reiniciar el dibujo.

Esta clase administra los eventos:

- **MouseClicked:** Agrega un nuevo punto a la lista y solicita redibujar el canvas.
- **Paint:** Redibuja todo el conjunto de líneas y puntos usando los datos almacenados.

## 4. Ejecución del Programa

### 4.1 Al iniciar:

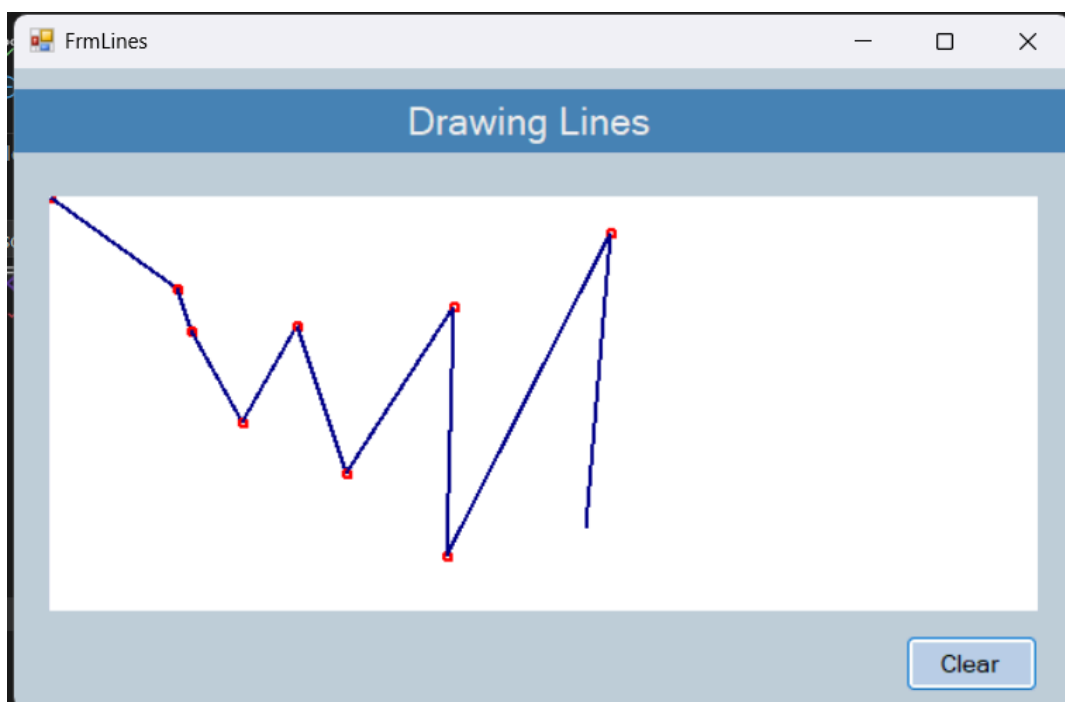
- Se crea una instancia de Line, que añade automáticamente el punto (0,0).

### 4.2 Al hacer clic en el PictureBox:

- Se captura la posición del clic.
- Se agrega este nuevo punto al final de la lista.
- Se ejecuta un `Invalidate()` para solicitar un repintado.
- El método `DrawAll` se ejecuta en el evento `Paint`, dibujando líneas entre cada par de puntos consecutivos.

### 4.3 Al presionar "Clear":

- Se elimina el contenido de la lista de puntos.
- Se agrega nuevamente el punto de origen (0,0).
- Se limpia el dibujo al redibujar el canvas vacío.



## **5. Conclusiones**

El programa cumple eficazmente su función principal: dibujar líneas consecutivas a partir del punto (0,0), extendiéndose con cada clic del usuario en el área gráfica. La funcionalidad de dibujo se desarrolla de forma encadenada y continua, donde cada nuevo clic genera una línea desde el último punto registrado hasta la nueva posición, lo cual proporciona una experiencia visual coherente. La separación entre la lógica de dibujo (Line) y la interfaz gráfica (FrmLines) permite una gestión clara y precisa del comportamiento del programa, asegurando que su funcionalidad sea modular, entendible y fácil de mantener.