

Caracas, 18 de Noviembre de 2025

## **PROYECTO # 1**

Se desea que usted elabore una aplicación en Microsoft® Visual C++ (o Javascript/Typescript, C# o Java) que implemente el despliegue y la manipulación de las primitivas 2D vistas en clase, así como otras funcionalidades para ofrecer una interfaz apropiada al usuario.

### **Despliegue general (7 puntos).**

Su aplicación debe implementar las siguientes primitivas 2D:

- línea (1 punto)
- elipse (1 punto)
- rectángulo (1 punto)
- triángulo (2 puntos)
- curva de Bézier (2 puntos)

### **Funcionalidades básicas (4 puntos)**

- Definir el color de borde actual. Esto significa que, al crear un objeto, aparecerá con ese color de borde. Si hay un objeto seleccionado, cambiará además su color de borde. (0.5 puntos)
- Definir el color de relleno actual. Esto significa que, al crear un objeto que soporta relleno, aparecerá con ese color de relleno. Si hay un objeto seleccionado, cambiará además su color de relleno. (0.5 puntos)
- Seleccionar objeto ya dibujado mediante clic del mouse. El usuario debe saber en todo momento cuál es la figura seleccionada, para poderle cambiar sus colores, y puntos de control. Para ello, debe hacer visibles y notorios los vértices o puntos de control del objeto seleccionado (objeto actual). En caso de curva de Bézier, mostrar además el polígono de control cuando la curva está seleccionada. (0.5 puntos)
- Seleccionar primitiva a crear con botones/íconos. (0.5 puntos)
- Modificar puntos de control. Responder al clic y dragging. Actualizar despliegue en tiempo real mientras se mueve el mouse. (1 punto).
- Borrar primitiva seleccionada vía interfaz y/o presionando tecla “del”. (0.5 puntos)
- Definir color de los puntos de control (no tiene que ver con el color del objeto, y es una propiedad global). Los puntos de control deben ser desplegados como un pequeño círculo relleno de alrededor de 4 píxeles de radio (0.5 puntos)

### **Curva de Bézier (2 puntos)**

- Seleccionar el color del polígono de control. Este polígono se muestra junto los vértices únicamente cuando la curva está seleccionada (0.5 puntos)
- Aumentar grado de la curva. (0.5 puntos)
- Subdividir la curva para un valor de  $t$  en  $[0,1]$ . Ese valor de  $t$  debe ser introducido con un “slider”, mostrando en todo momento, el valor real de  $t$ . Debe sustituir el objeto por 2 nuevos objetos. (1 punto)

### **Funcionalidades especiales (5 puntos)**

- Abrir/Guardar un dibujo. Puede utilizar JSON (texto) o archivo binario (1 punto)
- Enviar primitiva seleccionada un nivel atrás, un nivel adelante, al fondo o al tope. (1 punto)
- Cambiar el tipo de mouse cuando esté sobre el objeto gráfico desplegado y cuando esté en un punto de control. (0.75 puntos)
- Borrar todas las primitivas. (0.25 puntos)
- Al utilizar SHIFT mientras se draggea con el ratón, la elipse se desplegará como círculo, y el rectángulo como cuadrado. (0.5 puntos)
- Permitir la selección de un nivel de transparencia en cada color (borde, relleno, color de polígono de control, vértices, etc). Al hacer setPixel, considerar el nivel de transparencia del pixel para mezclarse el color que ya está dibujado en el búfer ( $\text{color\_de\_buffer}[x][y] = \text{color\_de\_buffer}[x][y] * (1-\text{alpha}) + \text{color} * \text{alpha}$ ). Implementar este blending con cuidado, debido a las conversiones de tipo (1 punto).
- Cambiar color de fondo. Use std::fill para llenar el búfer con un RBBA. (0.5 puntos)

### **Calidad y consistencia (2 puntos )**

- Se evaluará el código, revisando las buenas prácticas de programación: comentarios, claridad, uso correcto de clases abstractas, reducción de código duplicado al mínimo, legibilidad, asignación de nombres a: clases, objetos, métodos, constantes, atributos, etc. Uso correcto de constructores, parámetros, eficacia, eficiencia, robustez, escalabilidad, ... (1 punto)
- Interfaz, precisión al seleccionar los objetos ya dibujados. (1 punto)

### **Funcionalidades opcionales (20 puntos)**

- Una opción para indicar que se desplieguen las figuras con las primitivas implementadas por ustedes o con las primitivas de un API gráfico 2D: MFC, WIN32, etc. (4 puntos)
- Hacer y deshacer en cascada todas las operaciones (Undo/Redo) (4 puntos)
- Implementar Antialiasing (2 puntos)
- Cambiar el ancho de las líneas y bordes de primitivas (2 puntos)
- Cambiar el tipo de línea/bordes (continua, puntos, segmentos, segmento+punto) (2 puntos)
- Cortar, copiar y pegar primitivas (2 puntos)

- Salvar render (buffer) como PNG o JPG (2 puntos)
- Implementación de rotación de rectángulo (2 puntos)

### **Notas y aclaratorias importantes**

- La aplicación debe implementarse utilizando programación orientada a objetos, por lo que se sugiere la utilización de herencia, clases abstractas y concretas, sobrecarga de operaciones en la creación de sus métodos y clases.
- Los algoritmos para el despliegue de las primitivas deben implementarse con aritmética de enteros e incremental. En el caso del triángulo, puede elegir utilizar aritmética entera o real.
- El usuario podrá modificar cualquier figura creada. Basta seleccionarla con clic, y modificar sus puntos de control.
- Procure utilizar estructuras dinámicas. Evitar memory leaks.
- No desplegar el mismo píxel más de una vez al dibujar cada primitiva.

### **Entrega**

El proyecto puede ser realizado de forma individual o en grupos de dos (2) personas. En el caso de que se realice de manera grupal, deben implementar todas las funcionalidades opcionales, y la calificación estará sobre 40 puntos. Para llevarla al rango de 20 puntos, simplemente se divide entre 2. En caso que se realice individualmente, no hay que implementar requerimientos opcionales.

Entregar el código fuente de la aplicación (y todo lo necesario para compilar/correr) mediante un link de dropbox a recs34@gmail.com. El proyecto debe ser entregado a más tardar el día viernes 19 de diciembre del 2025. Escribir como subject: proyecto 1 – ICG 2025-II <nombres>

**R.C.**