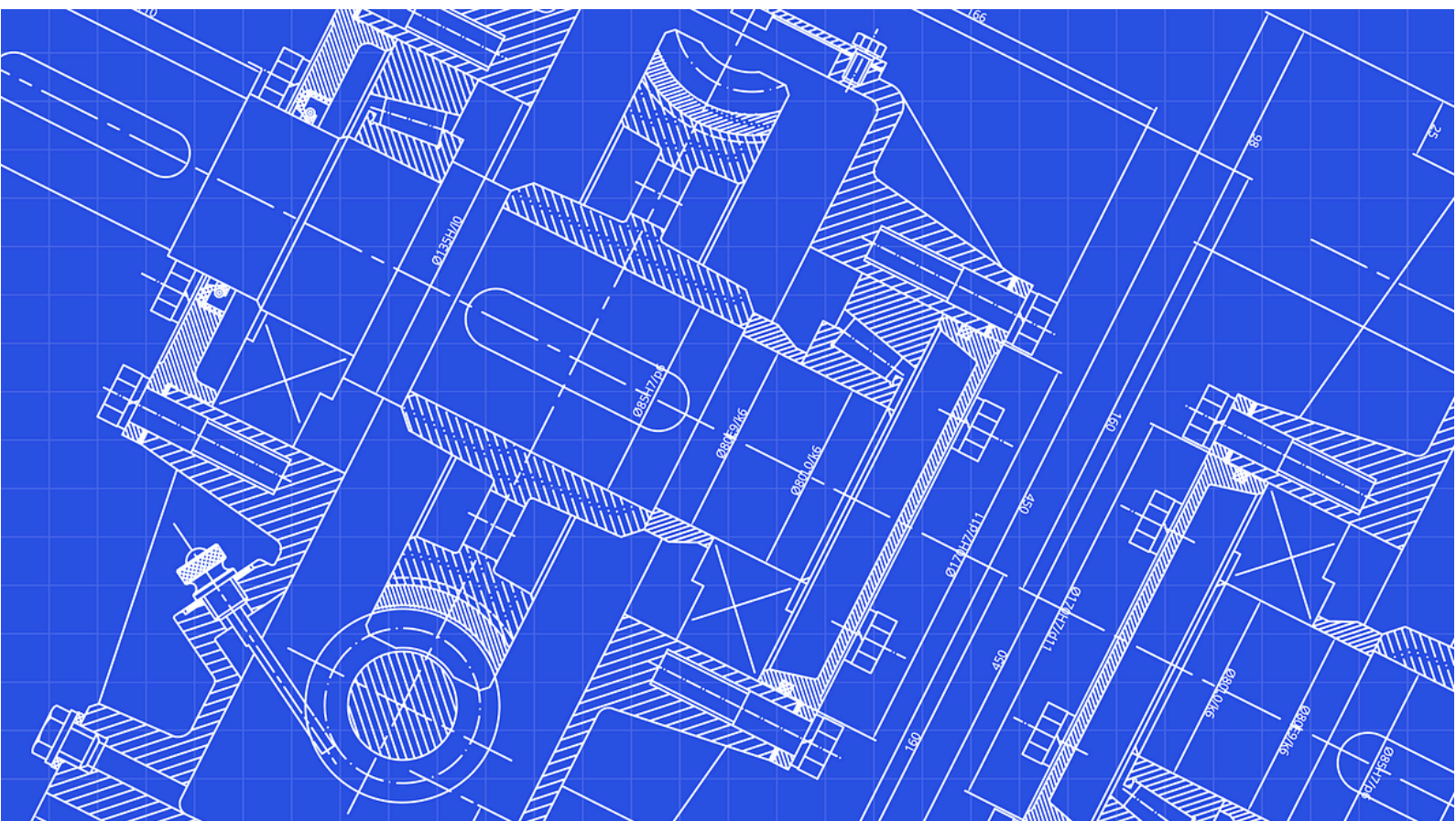


Nombre:	<b><u>Nota</u></b>
Curso:	

## Normas

- El código entregado tiene que compilar sin problemas
- Existen algunos errores (comentados en clase) que hará que la nota máxima de este examen sea de 2
- Cada ejercicio se evaluará por separado, se ha de obtener un mínimo de un 4 en cada ejercicio, de lo contrario, la nota máxima de esta prueba será de 4.
- La nota final será la suma ponderada de cada ejercicio.
- **LA NOTA MÍNIMA PARA PODER SUPERAR ESTA PRUEBA es de 4, si no se obtiene un 4, no se podrá superar la asignatura, siendo un 4 la nota máxima de la misma**



Tiempo estimado

Dificultad

Tipo de actividad

**3 horas**

**Media**

**Evaluación**





Se desea realizar una aplicación para manejar el diseño de planos, realiza las siguientes implementaciones:

**(1 punto) EJERCICIO 1**

- A esta actividad le acompaña un archivo con la especificación de un canvas, has de usarlo. Implementa las propiedades del Rect2D: Area y Center

**(1 punto) EJERCICIO 2**

- Crea una interfaz llamada IShape.
  - Tiene las siguientes propiedades get/set
    - string Name
    - Color Color
    - bool HasArea
    - double Area
    - double Perimeter
    - Point2D Center
    - Rect2D Rect → Devuelve un rectángulo que envuelve a la figura
  - Tiene los siguientes métodos
    - void Draw(ICanvas canvas);

**(2 puntos) EJERCICIO 3**

- Crea una clase llamada Shape que implemente la interfaz IShape
  - Tiene dos atributos, que son el nombre (protected) y el color (private). Estos atributos se le pasarán como parámetro por el constructor.
  - La función Draw(ICanvas canvas) lo único que hará sera poner en el canvas el color de la figura. Las clases que hereden de Shape podrán sobrescribir esta función
  - Cada property que no sea Name ni Color se implementará mediante la llamada a una función que se se implementará por las clases hijas

**(3 puntos) EJERCICIO 4**

- Implementa la Utils que tiene los siguientes métodos de clase
  - GetDistance(Point2D a, Point2D b) → Devuelve la distancia entre dos puntos
  - GetBoundingBox(Point2D[] points) → Devuelve un Rect2D que es el rectángulo mínimo que envuelve los puntos que se le pasan como parámetro.
  - GetArea(Point2D[] points) → Devuelve el área formada por un polígono con los puntos que se le pasan. La fórmula es la siguiente:

$$A = \frac{1}{2} \sum_{i=1}^n (y_i + y_{i+1})(x_i - x_{i+1})$$
$$= \frac{1}{2} \left( (y_1 + y_2)(x_1 - x_2) + \dots + (y_n + y_1)(x_n - x_1) \right)$$

- GetPerimeter(Point2D[] points) → Devuelve el perímetro de un polígono que tiene los vértices que se le pasan como parámetro a esta función



### (2 puntos) EJERCICIO 5

- Implementa las clases que heredan de Shape:
  - Circle → tiene una property get/set que es el radio
  - Rectangle → Tiene una función que se la pasa un index y te devuelve una de sus esquinas
  - Point
  - Segment → viene definido por dos Point2D

### (2 puntos) EJERCICIO 6

- Implementa la clase Polygon que hereda de Shape
  - Consta de un array de Point2D
  - Tiene una property int PointCount
  - Tiene las siguientes funciones
    - Clear() → Borra todos los puntos
    - Close() → Marca al polígono como un polígono cerrado
    - AddPoint(Point2D point) → Añade un punto al polígono
    - GetPointAt(int index) → Devuelve el punto que está en una posición determinada

### (3 puntos) EJERCICIO 7

- Crea la interfaz llamada IBlueprint y una implementación de la misma.
- Tiene los siguientes métodos:
  - void AddShape(IShape shape);
  - void RemoveShapeWithName(string name);
  - IShape GetShapeWithName(string name);
  - List<IShape> GetShapes(FilterDelegate del); → El delegado acepta un IShape y devuelve un booleano
  - void Draw(ICanvas canvas);

#### Especificaciones:

- Se ha de implementar siguiendo las filosofías de la programación orientada a objetos.
- A este examen le acompaña una especificación de un canvas, deberás usarlos para realizar la actividad