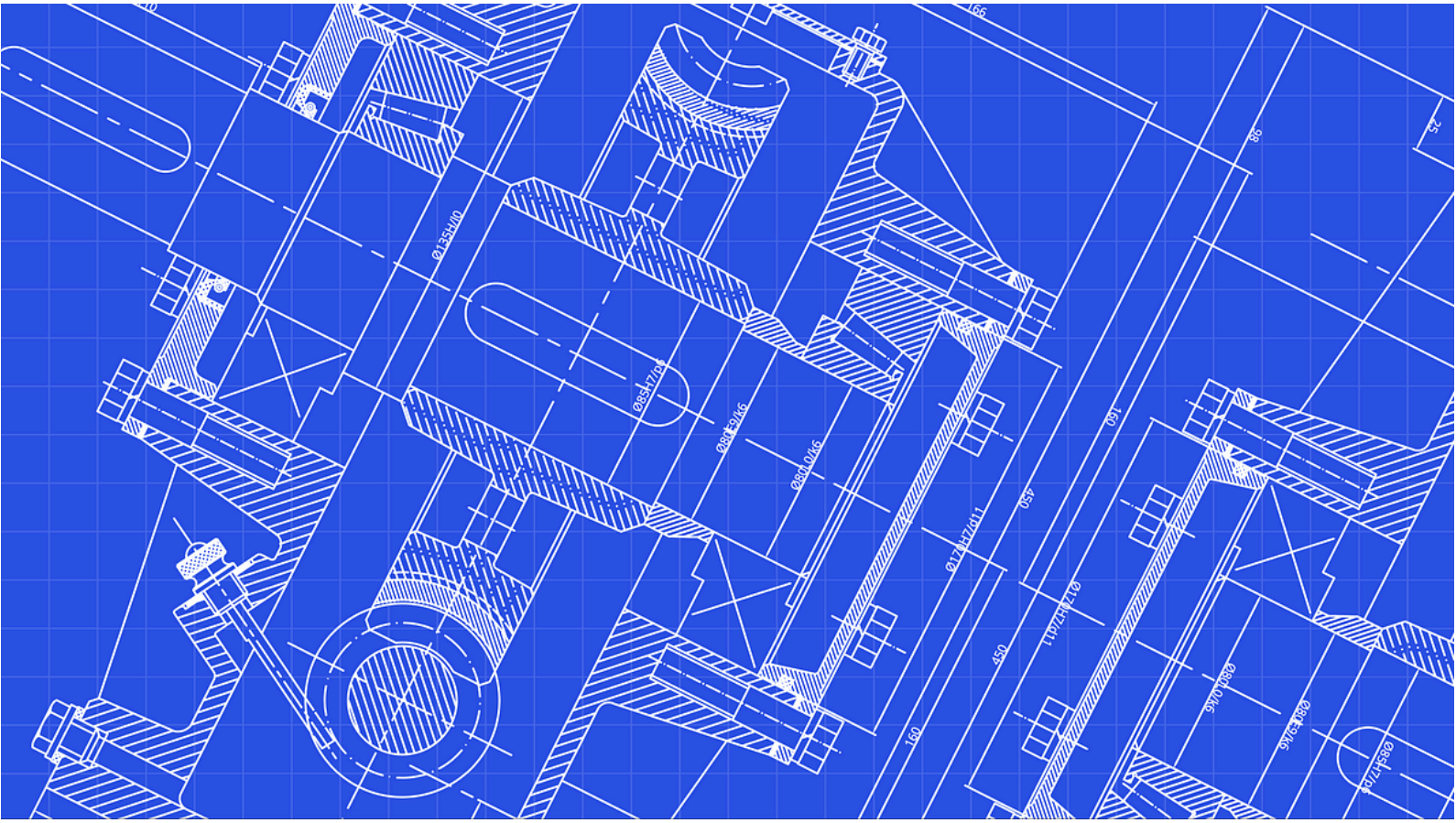

Tiempo estimado	Dificultad	Tipo de actividad
3 horas	**Media**	**Evaluación**

Unión Europea
Fondo Europeo de Desarrollo Regional
Una manera de hacer Europa

GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

IES Mare Nostrum

Normas

- **El código entregado tiene que compilar sin problemas**
- **Se tiene que proporcionar un main que pruebe todas las funciones implementadas, si no, no se corregirá el examen**
- **Existen algunos errores (comentados en clase) que hará que la nota máxima de este examen sea de 4**
- **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**
- **Se ha de implementar siguiendo la filosofía de la programación orientada a objetos.**
- **Has de mantener el código modular y bien nombrado, si no se entiende o no está lo suficientemente caro, esa parte no se corregirá**

Leyenda que puedes encontrar en el examen

- **(A)** → Si no implementa alguno de estos puntos, la nota del exámen será de 0.
- **(B)** → Si no se implementa alguno de estos puntos, se bajará 4 puntos sobre 10 al examen
- **(C)** → Si no se implementa alguno de estos puntos, se bajará 3 puntos sobre 10 al examen
- **(D)** → Si no se implementa alguno de estos puntos, se bajará 2 puntos sobre 10 al examen
- Cada fallo penalizará en función de la gravedad del fallo

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

(0.5 puntos) EJERCICIO 1

- Implementa una clase llamada Point2D: representa un punto con x y y.
- Implementa una clase llamada Vector2D: como Point2D, pero se usa para desplazamientos.
- Implementa una clase llamada Rect2D
- Implementa una clase llamada Color (RGB): 3 doubles

(0.5 puntos) EJERCICIO 2

- Implementa una interfaz llamada ICanvas con los siguientes métodos
  - SetCurrentColor(Color color)
  - DrawPolygon(Point2D[] points)
  - DrawCircle(Rect2D rect)
- Implementa una clase que implemente la interfaz ICanvas: Para hacer esto, lo que tienen que hacer las funciones es simplemente escribir por pantalla lo que se le pasa como parámetro, por ejemplo:

```
public void DrawPolygon(Point2D[] points) {
    Console.WriteLine("Dibujando polígono con puntos: " + points.Length);
    for (i = 0…)
        Console.WriteLine(el punto i tiene las coordenadas …);
}
```

(2 punto) EJERCICIO 3

- Crea una interfaz llamada IShape.
  - Tiene las siguientes propiedades
    - string Name (mutable)
    - Color Color (inmutable)
    - bool HasArea (inmutable)
    - double Area (inmutable)
    - double Perimeter (inmutable)
    - Point2D Center (inmutable)
    - Rect2D Rect → Devuelve un rectángulo que envuelve a la figura (inmutable)
    - **(C)** IBlueprint Owner → Devuelve el plano que contiene a esta figura. Para hacer esto hay que tener en cuenta muchas cosas
  - Tiene los siguientes métodos
    - void Draw(ICanvas canvas);
    - void Displace(Vector2D direction);
- Crea una clase llamada Shape que implemente la interfaz IShape
- Es obligatorio que esta clase sea abstracta porque ha de tener al menos un método o una property abstracta

(2 puntos) EJERCICIO 4

- Implementa la clase 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:

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

  - (A) 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
  - Point
  - Segment → viene definido por dos Point2D

(3 puntos) EJERCICIO 6

- Crea la interfaz llamada IBlueprint y una implementación de la misma.
- Tiene los siguientes métodos:
  - (A) int GetShapeCount()
  - (A) IShape GetShapeAt(index);
  - (A) void AddShape(IShape shape);
  - void RemoveShape(delegado);
  - IShape GetShape(delegado);
  - (D) List<IShape> FilterShapes(delegado);
  - void Draw(ICanvas canvas);