Ejercicio Java

1. Diagrama de clases: traducimos el ejercicio a un diagrama de clases. Identificamos las clases y la relación entre ellas. Las nombramos, escribimos los atributos y los métodos con los tipos y los modificadores de acceso.
2. Estudiamos diferentes patrones: builder, publisher-subscriber, factory, facade, abstract factory, singletone y composite. Después de hacer varias pruebas con ellos e intentar varias implementaciones, optamos por trabajar con el patrón Builder porque nos parece que es la solución que mejor se adapta a nuestro problema.
3. Patrón Builder: comentamos los pasos principales

* Interfaz Builder: define todos los caminos posibles para configurar un producto. Construimos la interfaz con los métodos para elegir la figura, el color de fondo y el color del borde.
* FiguraBuilder: clase que implementa la interfaz builder y por tanto sus métodos cuyas funcionalidades se adaptan ahora. Consta además de variables para guardar cada característica en modo privado. También consta de un método getResult que devuelve una nueva figura y sus características.
* Figura: es la product class. Compuesta de constantes para los atributos, un constructor que le da el estado con características que recibe por parámetros. Métodos getters y setters que devuelven el valor de las constantes.
* FiguraTipo: clase para obtener el tipo de figura que se pide.
* ColorFondo: clase para para devolver el color de fondo de la figura.
* ColorBorde: clase para devolver el color del borde de la figura.
* Director: controla los constructores y define el orden de los pasos de construcción. Aquí construimos las figuras Corazón, Rayo y Estrella.
* Lienzo: código cliente. Todo confluye en este punto. Se instancian las figuras y se muestran por pantalla.

Hemos intentado implementar el patrón Builder, sin embargo hemos encontrado algunas dificultades y no hemos sido capaces de resolver los problemas que se plantean en algunas partes del patrón.