## Respuestas

- ¿El objetivo del patrón se distingue en el diseño? Si, el objetivo se distingue, lo que se quiere lograr es poder agregar responsabilidades a un objeto de forma dinámica, esto **PNGFilterLauncher** se puede lograr mediante el uso del patrón Decorator ya que permite que se puedan agregar más de un Filtro de manera «static»-inputFile: File dinámica a la imagen que se quiera. «static»-outputFile: File - ¿La estructura del proyecto coincide con la estructura y los «static»+main(args: String[\*]) participantes del patrón? «static»+processArgs(args: String[\*]) No, en ningún momento un Decorador contiene algún componente ya sea otro decorador o un componente «static»+initializeFilters() concreto, además el componente concreto no es ninguna clase sino que es una imagen externa la cual recibe los filtros. - En el caso que el patrón coincida, puede distinguir un "smell" o activeFilters 0..\* algo que se aleja del patrón presentado en el libro? filters Lo raro que puedo observar es que en el patrón presentado en el libro el Decorador contiene al componente concreto que luego los Decoradores concretos van a ir decorando, mientras que en «abstract» << Decorator>> este sistema, el Componente concreto es externo y no es Filter contenido por el Decorador de la misma forma que se presenta el patrón en el libro. «abstract»+filter(image: BufferedImage): BufferedImage RGBShiftRepeater Artifacter «final»-X\_REPEAT: Integer -tempFile: File «final»-Y REPEAT: Integer -compressionQuality: Real +filter(image: BufferedImage): BufferedImage +filter(image: BufferedImage): BufferedImage <<ConcreteDecorator>> <<ConcreteDecorator>> **RGBShifter** Dull +filter(image: BufferedImage): BufferedImage +filter(image: BufferedImage): BufferedImage +shiftValue(original: Integer, shiftMagnitude: Integer): Integer <<ConcreteDecorator>> <<ConcreteDecorator>> Repeater «final»-X\_REPEAT: Integer Rainbow «final»-Y REPEAT: Integer +filter(image: BufferedImage): BufferedImage +filter(image: BufferedImage): BufferedImage <<ConcreteDecorator>> <<ConcreteDecorator>> Monochrome +filter(image: BufferedImage): BufferedImage

<<ConcreteDecorator>>