**DOCUMENTO DE JUSTIFICACION DE USO/NO USO DE PATRONES**

ADAPTER **(No)**

No hay obstáculos entre la comunicación de las clases, adicionalmente no se usó dado que actualmente la estructura de las clases abstractas contienen la suficiente abstracción para las comunicaciones de los objetos así como lo es su propia fabrica

COMPOSITE **(No)**

No se usó porque ninguno de los elementos está compuesto por otro, es decir una pregunta no está compuesta de otra pregunta. De igual manera no se usó dado que no existe una agrupación de objetos del mismo tipo y por ende se utilizó una agrupación por stack

FACADE **(Si)**

Porque vamos a implementar alta cohesión de datos y bajo acoplamiento con el fin que el mantenimiento sea más fácil ya que habrán menos relaciones entre las clases. Es síntesis la fachada la usamos para la creación de la pregunta usando la factoría, es decir que al momento de crear una pregunta utilizamos la fachada para controlar ó redirigir la selección el tipo de pregunta

DECORATOR **(No)**

En realidad sabemos que el decorator es uno de los mejores patrones para las preguntas pero no supimos aplicarlo correctamente, por este concepto no nos arriesgamos a realizar algo que no iba quedar bien, en cambio hicimos una fábrica de preguntas permitiendo agregar responsabilidades por cada una sin usar el decorator

FLYWEIGHT **(Si)**

La idea es descargar o liberar la BD de tantas transacciones por ende lo que hicimos es minimizar la cantidad de consultas directamente a la DB, por consiguiente cada vez que se carga la lista de materias dicha lista no la trae esta información de la base de datos directamente si no de una cache precargada y cuando se agrega un elemento se guarda en la BD y se almacena en la cache y con este proceso se descarga significativamente la transaccionalidad en la BD