

Ejercicio 2:

1.- Principios de diseño usados: PRINCIPIO DE INVERSIÓN DE LA DEPENDENCIA.

Hemos utilizado este principio debido a que la mejor manera de resolver el ejercicio es con una interfaz o clase abstracta, en nuestro caso usamos la clase abstracta. Y así no depende de clases y funciones concretas. Utilizamos ArrayList para hacer una implementación concreta de una lista.

2.- Patrón de diseño: PATRÓN COMPOSICIÓN.

Hemos utilizado este patrón ya que queremos representar colecciones de objetos en forma de árbol, y es el más adecuado. Este patrón está compuesto por un componente, un componente concreto y una composición, en este caso, el componente se corresponde con EquipoComponent (), el componente concreto con Trabajador (elemento concreto, objeto único) y por último, Equipo (compuesto por trabajadores y subequipos) y Proyecto (compuesto por equipos que pueden tener subequipos y trabajadores) con composición.

3.- Para el diagrama dinámico hemos escogido el de secuencia porque es más específico.