**EJERCICIO 2 – PLANIFICADOR DE TAREAS**

**Principios de Diseño**

Para este ejercicio, los principales principios de diseño utilizados han sido el Principio de Responsabilidad Única y el de Inversión de la Dependencia.

El Principio de Responsabilidad Única puede verse aplicado en la clase Graphic, que tiene la responsabilidad de gestionar el grafo; en la clase Document, la cual gestiona las dependencias de un proyecto de la empresa. La clase Dependence se encarga de asegurarse de que una dependencia tenga un nodo (tarea) padre y un hijo, y la clase Node se responsabiliza de que cada Nodo tenga un nombre tipo char y un nivel.

El Principio de Inversión de la Dependencia lo utilizamos en el atributo 'map' de la clase Graphic. Lo declaramos tipo Map pero utilizamos su implementación HashMap. También lo utilizamos en el método 'showLevel' de la misma clase para declarar una lista tipo List<Character> pero utilizar su implementación ArrayList, y lo mismo hacemos en el contstructor de la clase Document y su método getDocument(). En el caso de que nos interesara, sería muy sencillo cambiar a otra implementación de Map o List en estos casos.

**Patrones de Diseño**

**Patrón Inmutable**

Uno de los patrones de diseño utilizados es el Patrón Inmutable. Lo aplicamos para la clase Document, la cual tiene una serie de dependencias de tareas que no queremos que cambien, de la misma forma que no es muy cómodo eliminar nodos en un HashMap. De esta forma, para cada grafo que queramos hacer, necesitaremos un documento, y en el caso de querer modificar el grafo, habrá que crear un nuevo documento con las modificaciones oportunas y crear el grafo de nuevo.

Diagrama

Descripción generada automáticamente

**Patrón Estrategia**

El otro patrón de diseño utilizado es el Patrón Estrategia, para definir una familia de algoritmos para recorrer la lista de tareas de un proyecto con la interfaz TaskOrder, compuesta por tres clases para ordenar las tareas por dependencia fuerte, dependencia débil u orden jerárquico.

Diagrama

Descripción generada automáticamente