**COMPONENTES DEL EQUIPO –> B1\_06**

* Dídimo Javier Negro Castellanos
* Óscar Jiménez Jiménez
* Jorge Alberto Gómez León

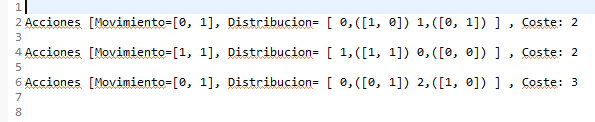
**NOMBRE DEL REPOSITORIO -> B1\_06**

<https://github.com/didimoj/B1_06>

**JUSTIFICACIÓN DE LA TAREA REALIZADA**

**Hito 3**

Primero de todo, hemos creado un nuevo método getCosto() en la clase Estado, al cual le pasamos como argumento una Acción y devuelve el coste de dicha acción más uno. Además, el coste de cada acción es calculado e imprimido junto a la acción a la que corresponde. Ver ejemplo:

 Por otra parte, hemos definido el algoritmo básico de búsqueda e implementado las estrategias de anchura, profundidad (simple, acotada e iterativa) y costo uniforme. En nuestra clase Operaciones implementamos las estrategias de en profundidad (BusquedaAcotada y BusquedaIterativa) y con un switch asociamos una valoración del nodo diferente para cada estrategia de búsqueda.

En cuanto al método BusquedaAcotada, le pasamos como argumentos el Problema, la Estrategia a utilizar y la Profundidad máxima (prof\_max) a la que puede llegar el algoritmo. La única diferencia con la BusquedaIterativa es que a esta le añadimos además un incremento de profundidad (inc\_prof). Ambos métodos devuelven una cola de nodos.

Por último, al ejecutar nuestro programa escribimos en un fichero de texto la secuencia de acciones con la que alcanzamos la solución. Ver imagen anterior, escritura en fichero de texto.