

# Ejercicio Práctico Entregable 5: Documentación

## ¿Que se puede ver en este documento?

En esta breve documentación se explica el objetivo del ejercicio entregable, sus características, y se describe la técnica implementada para obtener una solución.

## Objetivo del ejercicio:

Utilizar lo aprendido en las clases sobre algoritmos **BackTracking** para intentar conseguir una aproximación a la mejor solución posible de un problema complejo.

## Consigna:

La consigna es la misma que en el anterior ejercicio entregable, pero esta vez se nos proporcionan 2 datasets, el primero con 20 familias, y el segundo con 60, y son 10 días de exhibición.

Asimismo este vez SOLO se acepta como solución, la solución óptima, en la cual todas las familias están asignadas en uno de sus días de preferencia, y la suma de todos los bonos a pagar es la mínima posible.

## ¿Que algoritmo se utilizó para la solución?

Para resolver este problema se utilizó la técnica de BackTracking, un algoritmo de fuerza bruta que debido a su naturaleza nos garantiza hallar la solución óptima.

## ¿Cómo trabaja BackTracking?

BackTracking es un algoritmo de fuerza bruta que garantiza encontrar la solución simplemente porque la técnica consiste en probar todas las combinaciones posibles, por lo que eventualmente dará con una solución.

## Analizando el problema:

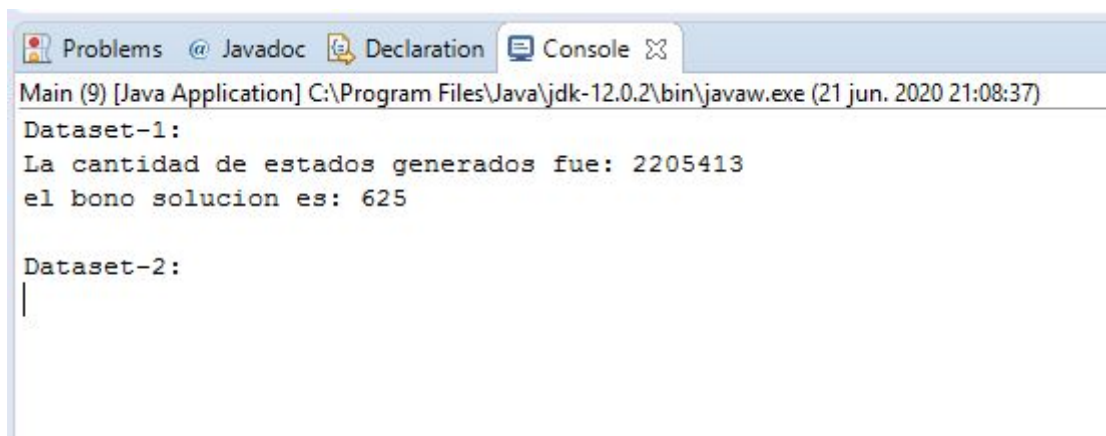
Se tiene un conjunto de familias ordenadas ascendentemente por su id numérica, cada familia establece cuales son los 3 días en los que puede asistir a la exhibición. El algoritmo implementado recorre la colección de familias, generando estados de recursión por cada día en los que la familia analizada puede ubicarse, cada vez que prueba ubicarla en un día, genera un nuevo estado en el que pasará a analizar la siguiente de las familias restantes (**en la carpeta del proyecto hay una imagen de ejemplo de cómo se va generando el árbol de exploración**).

- Árbol de Exploración:
  - 1- Se reciben las familias, el calendario de días, el calendario auxiliar, la suma de los bonos de la mejor solución, la suma de los bonos de la solución parcial, el cupo de los días, y el índice de la familia actual.
  - 2- Se obtienen los días de preferencia de la familia que se está analizando.

- 3- Se obtienen los estados posibles en base a los días de preferencia.
- 4-Se asigna el día a la familia y se llama a backTracking con la siguiente familia del arreglo, esto se hace para cada día posible, generando así los nuevos estados del árbol de exploración.
- 5-Este proceso se repite hasta que se hayan analizado todas las familias, entonces empieza a regresar a los estados anteriores, donde prueba elegir otro camino, generando a su vez nuevos estados posibles.

### Resultados:

La cátedra de programación III nos proporcionó el valor óptimo de la suma de los bonos a pagar, para ambos Dataset, Siendo 625 dólares para el primer dataset, y 260 para el segundo. Para chequear si el algoritmo funciona se deben obtener exactamente los mismos valores. Adicionalmente se incluyó el número de estados generados durante la ejecución del algoritmo.



```
Main (9) [Java Application] C:\Program Files\Java\jdk-12.0.2\bin\javaw.exe (21 jun. 2020 21:08:37)
Dataset-1:
La cantidad de estados generados fue: 2205413
el bono solucion es: 625

Dataset-2:
|
```

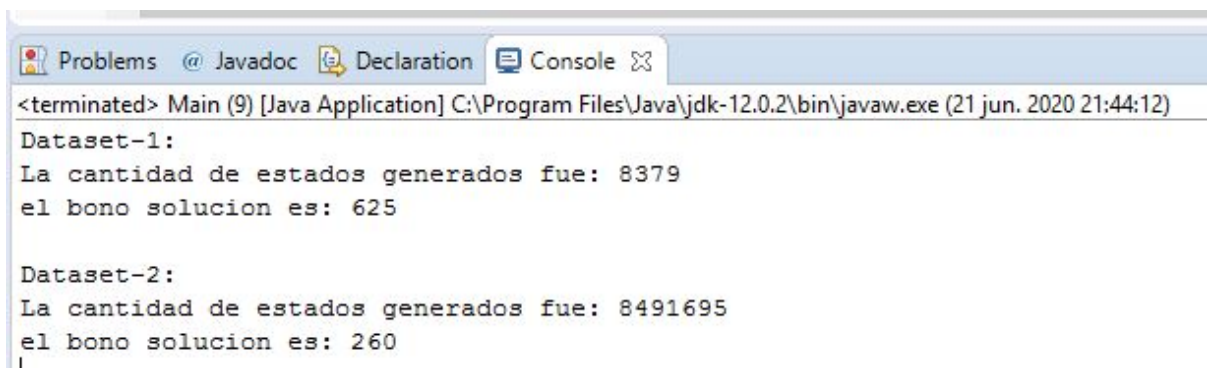
Para el Dataset-1 el resultado fue el esperado. sin embargo la carga de procesamiento es demasiada, podemos observar que generó **más de 2 millones de estados** para encontrar la solución. De igual forma incluso luego de varios minutos, el costo del algoritmo es tan grande que no logra encontrar la solución óptima para el Dataset-2 que contiene el triple de familias que el Dataset-1.

### ¿Existe alguna manera de disminuir la carga de ejecución del algoritmo?

Si bien el algoritmo encuentra la solución óptima (en algún futuro), es demasiado costoso, esto se debe a que cada decisión que tome tiene un impacto directo en los estados que generará a continuación. Por si solo BackTracking no conoce cómo detenerse hasta que no haya generado todas las posibilidades. Utilizando una técnica de **Poda** se puede evitar la generación de estados “inservibles”.

Para este caso se considera inservible a cualquier rama del árbol en la que ya se haya superado la suma de los bonos de la **mejor solución hasta el momento**.

## Resultados:



```
<terminated> Main (9) [Java Application] C:\Program Files\Java\jdk-12.0.2\bin\javaw.exe (21 jun. 2020 21:44:12)
Dataset-1:
La cantidad de estados generados fue: 8379
el bono solucion es: 625

Dataset-2:
La cantidad de estados generados fue: 8491695
el bono solucion es: 260
|
```

Esta vez el costo se redujo notablemente para ambos Dataset, sobre todo para el segundo que en pocos segundos encontro la solucion optima luego de generar **más de 8 millones de estados**.

## Conclusión:

BackTracking es efectivo a la hora de resolver problemas, pero según la complejidad de los mismos, incluso aplicando técnicas de poda podría volverse inviable debido a la excesiva cantidad de posibilidades.