

## Introducción

El problema propuesto a resolver se trata sobre optimizar la cantidad de vacunados diarios entre un rango de tiempo (desde las 9:00 hasta las 13:00 horas) de un centro de vacunación llamado "Clotario Blest" de la comuna de Maipú. Este centro cuenta con 4 secciones para la aplicación de vacunas, las cuales llamaremos "box", que cumplen con los requisitos impuestos por el ministerio referente a la ventilación. Cada box cuenta con distintas características relacionadas al aforo máximo de personas y a la cantidad de vacunados que se pueden vacunar antes de empezar la sanitización, así como la duración de esta misma. Aquella información se verá resumida en la siguiente tabla.

Box	Aforo Máximo	Tiempo de sanitización	Tiempo de sanitización
Box1	4 personas	Cada 12 vacunados/as	4 minutos
Box2	4 personas	Cada 12 vacunados/as	4 minutos
Box3	7 personas	Cada 21 vacunados/as	6 minutos
Box4	9 personas	Cada 27 vacunados/as	8 minutos

Además, se sabe por diversos estudios realizados previamente que un solo profesional se demora 4 minutos en atender a una persona, entre 2 profesionales se demoran 2 minutos por persona y entre 3 o más personas se demoran 1 minuto por persona (mayor a 3 no hay ganancia de tiempo). Además, los profesionales se pueden mover de su box, pero el hacer esto les demora 1 minuto.

Para resolver el problema anterior, se utilizará el método de Búsqueda en Espacios de Soluciones. El cual consiste en probar todas las combinaciones posibles que se puedan generar dado un numero n, así generar de manera segura cual será la forma más efectiva de ordenar a los médicos.

## Solución Propuesta

Como bien se ha dicho en la introducción, para resolver este problema se ha utilizado el algoritmo llamado "Búsqueda en Espacio de Soluciones" (BES), el cual consiste en a partir de un n dado y definiendo con anterioridad como es una solución, genera todas las posibles soluciones que se le pueden dar al problema, y una vez terminado busca la solución la cual, según nuestro criterio, busca maximizar la cantidad de vacunados dentro de un rango de horas y respetando las características dadas para poder funcionar. Antes de implementar el algoritmo, debemos buscar la forma de poder calcular los vacunados que se puedan inocular. Para ello, dejare expuesto un pseudo código de una de las funciones para calcular el máximo posible que se puede encontrar en un box, para este ejemplo se ocupara el box1, pero la idea de la función es aplicable para el resto de box.

```
maxBox1(num n): num
    num tTotal<-0 ...tiempo Total
    num vTotal<-1 ... vacunados Total
    num tRestante ... almacena los vacunados x
    minuto
    num tRestanteC ... copia tRestante
    if n = 1 then
        tRestante<-4
        tRestanteC<-tRestante
    if n = 2 then
        tRestante<-2
        tRestanteC<-tRestante
    if n >=3 then
        tRestante<-1
        tRestanteC<-tRestante
    else ... caso box vacio
        return 0
    while tTotal < 240 do
        if tRestante = 0 then
            vTotal<- vTotal+1
            tRestante<-RestanteC
            tTotal<- total+1
            if MOD(vTotal) = 12 then
                tTotal <- tTotal + 3
                ...Sanitizacion
            else
                tRestante<-tRestante-1
                tTotal<-tTotal + 1
    return vTotal
```

El pseudo código anterior es el esquema el cual permite a partir de un número de médicos dado calcular el total de vacunados con  $T(n) = 23c + n$  y  $O(n) = n$ , la implementación para los demás boxes es bastante similar, solo que hay que tener cuidado de cambiar los valores de los vacunados necesarios para entrar a sanitización y el conteo de minutos que se demoran en desinfectarse un box. Asumiendo entonces que tenemos en nuestro poder las funciones maxBox1, maxBox2, maxBox3, maxBox4; procederé a explicar cómo se realizó la implementación de BES:

```
BES(num tMedicos): array
  num vMayor<-0
  inicializar(box)
  for q<-0 to 4
    for w<-0 to 4
      for e<-0 to 7
        for r<-0 to 9
          num res<- maxBox1(q)+maxBox2(w)+maxBox3(e)+maxBox4(r)
          if res>vMayor then
            vMayor<-res
            box[0]<-q
            box[1]<-w
            box[2]<-e
            box[3]<-r
  return box
```

El pseudo código anterior, teniendo una cantidad de instrucciones de  $T(n) = 10c + n^4$  y una complejidad de  $O(n) = n^4$ , puede parecer un poco grotesco y quizás parezca demasiado para el sistema, pero gracias a que C es un lenguaje de programación imperativo apegado a la máquina, no presenta ningún problema para él.

## Resultados y Análisis

Explicado a grandes rasgos las funciones más importantes, nos podemos fijar que el programa no implementa una de las opciones que nos entrega el problema, que es mover los médicos de box en box, esto pudo haberse visto implementado cada vez que entraba un box a sanitización, pero su no aprovechamiento viene del motivo de la poca ganancia de vacunados por sobre el esfuerzo físico y mental que deban hacer el personal. Si bien es cierto se lograría una mejor eficiencia, esto se puede llegar a ver contrarrestado con la poca consistencia que tiene un personal de permanecer en un mismo box. Dicho de otra forma, teniendo al personal en un box constante, evitas el posible error humano que pueda ocurrir cuando hay demasiado movimiento de personal, en donde pueden ocurrir cosas como equivocarse del box de destino, y si esto lo vamos reflejando en varios días, la inconsistencia y error irían en aumento.

Dicho esto, se presentaran los resultados obtenidos por el programa en una tabla y un gráfico:

N	Total Vacunados
1	47
2	94
3	141
4	188
5	217
6	249
7	278
8	310
9	339
10	371
11	400
12	432
13	432
14	432
15	432
16	432
17	432
18	432
19	432
20	432

Figura 1.

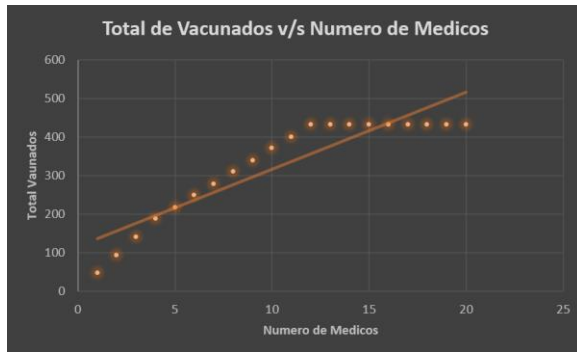


Figura 2. Representación de la tabla

Dado los resultados anteriores, podemos apreciar claramente que no hay una ganancia al momento de que el total de médicos sea mayor igual a 12.

Con esta implementación, además, se calculó el tiempo de ejecución que tuvo el programa con cada uno de los  $n$  entre 0 y 20, pero el tiempo de ejecución en todos los casos es despreciable ( $<0.00$ ).

## Conclusión

El trabajo anterior permitió un primer acercamiento con el cálculo de complejidad de algoritmos creados por nosotros mismos, por lo que fue una gran oportunidad.

La solución propuesta tiene bastante más desventajas que ventajas, siendo las desventajas tales como la posible notoriedad de pérdida de eficacia al no mover a los médicos al momento de desinfectar los boxes, lo cual se ve directamente afectado en una menor cantidad de vacunados por día. Por otra parte, esta respuesta ofrece una respuesta sencilla al problema la que además no ofrece tiempos de ejecución notorios.

Dicho lo anterior, siento que el desafío de poder implementar el movimiento de boxes fue perdido, pero no con esta derrota se perderán las ganas y motivaciones de seguir mejorando futuras implementaciones y formas de plantearse los problemas.