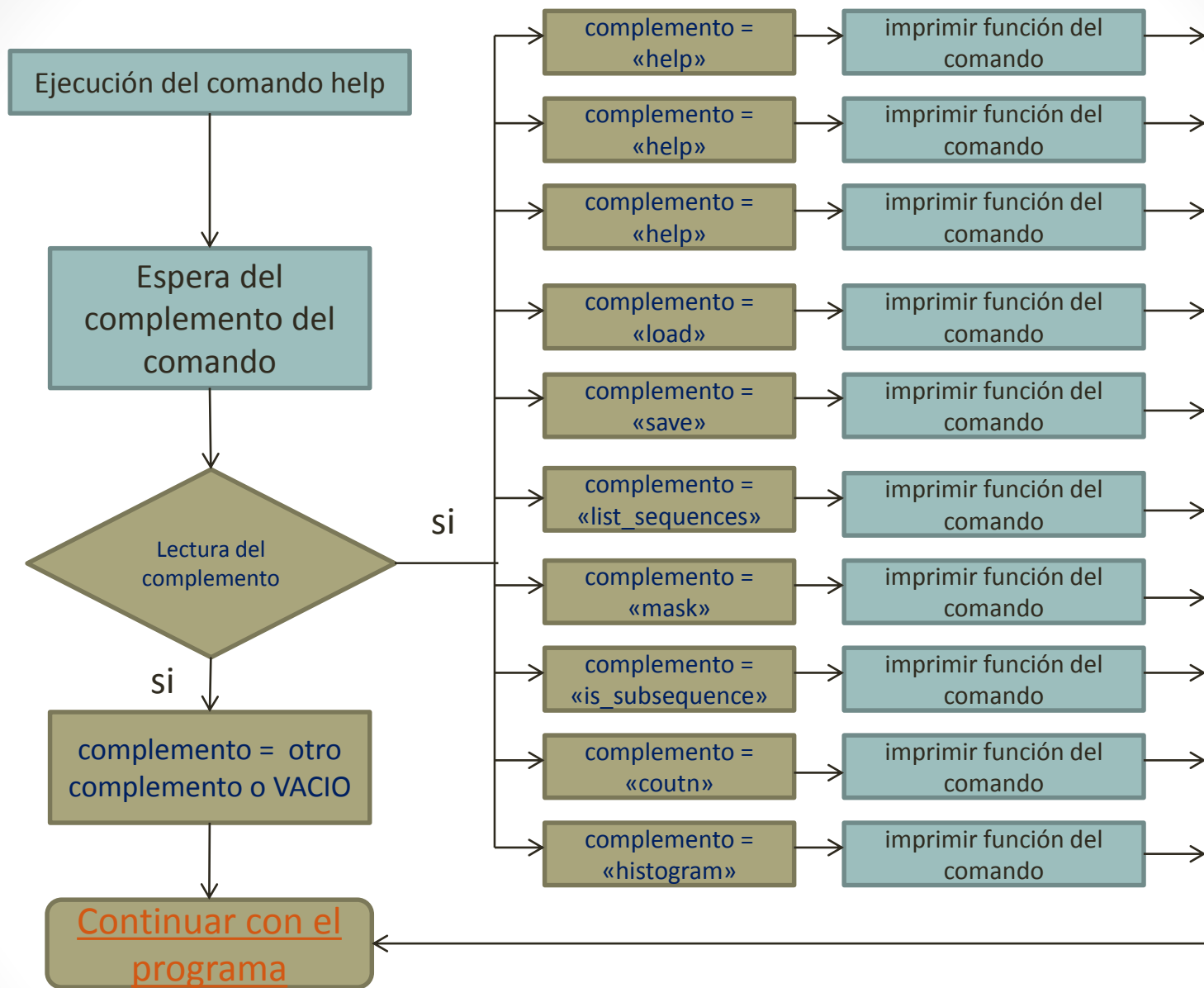


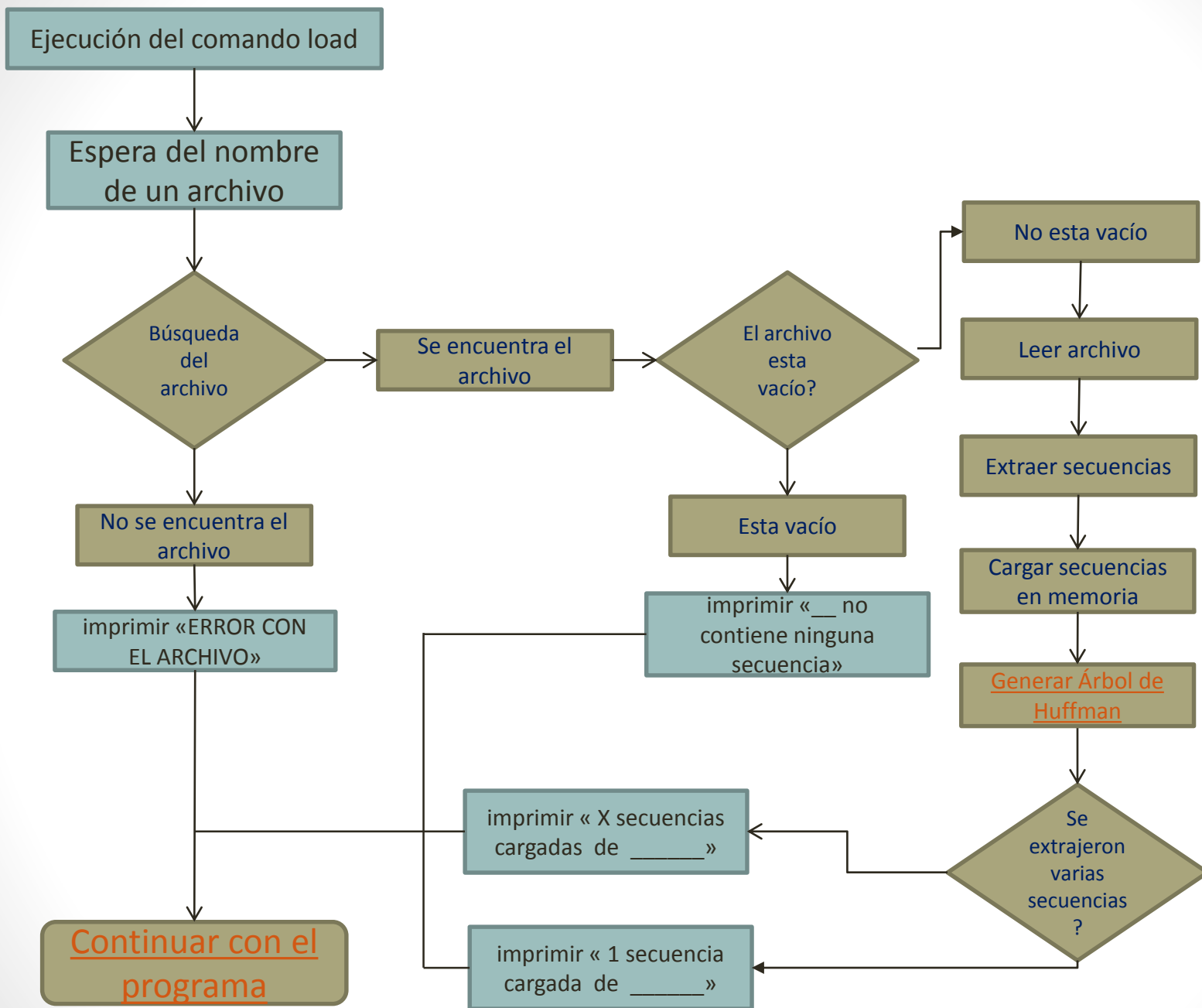
Esquemático del funcionamiento del proyecto (entrega2)

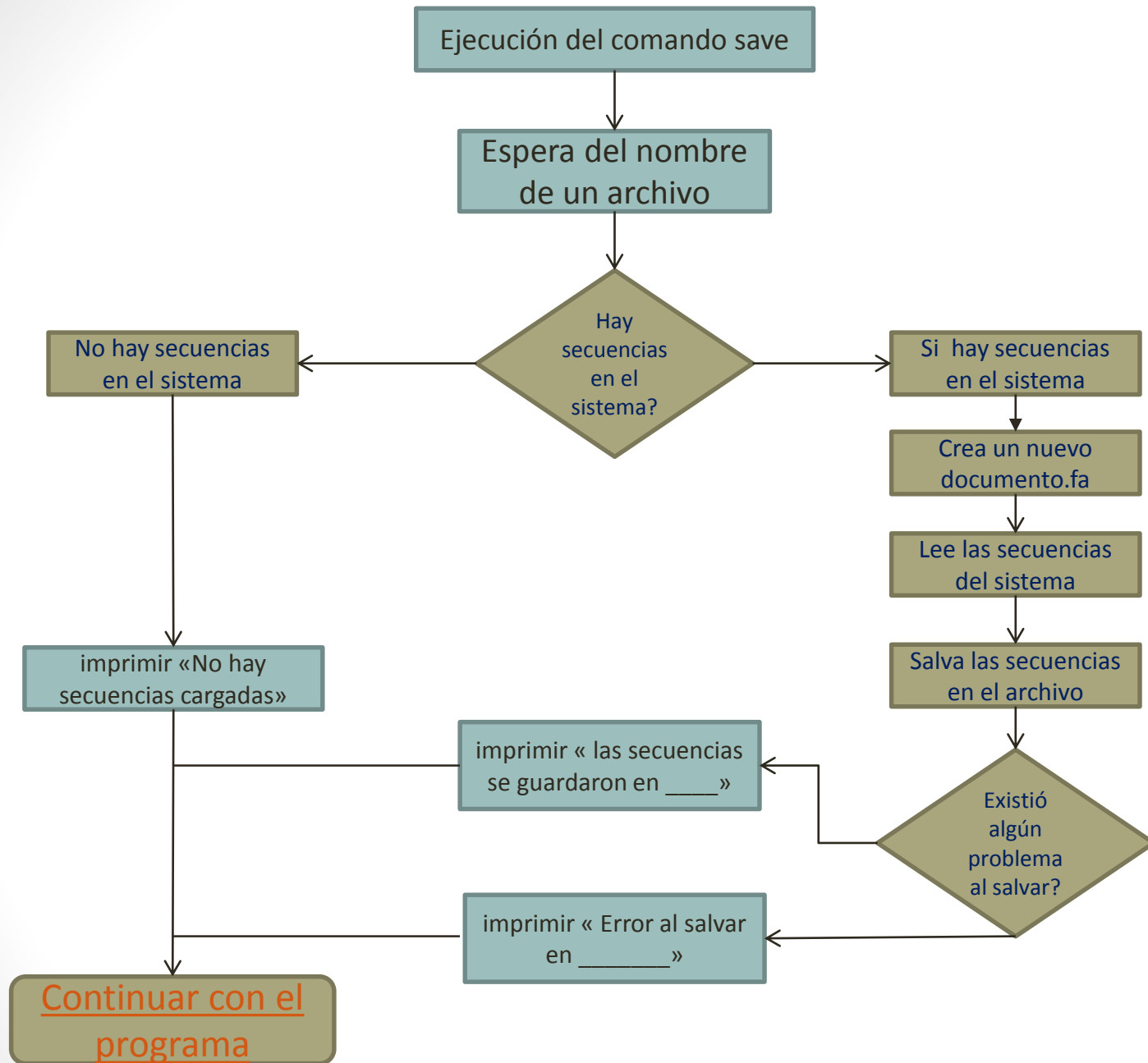
Juan Pablo Rodríguez
Luis Felipe Urdaneta
2015

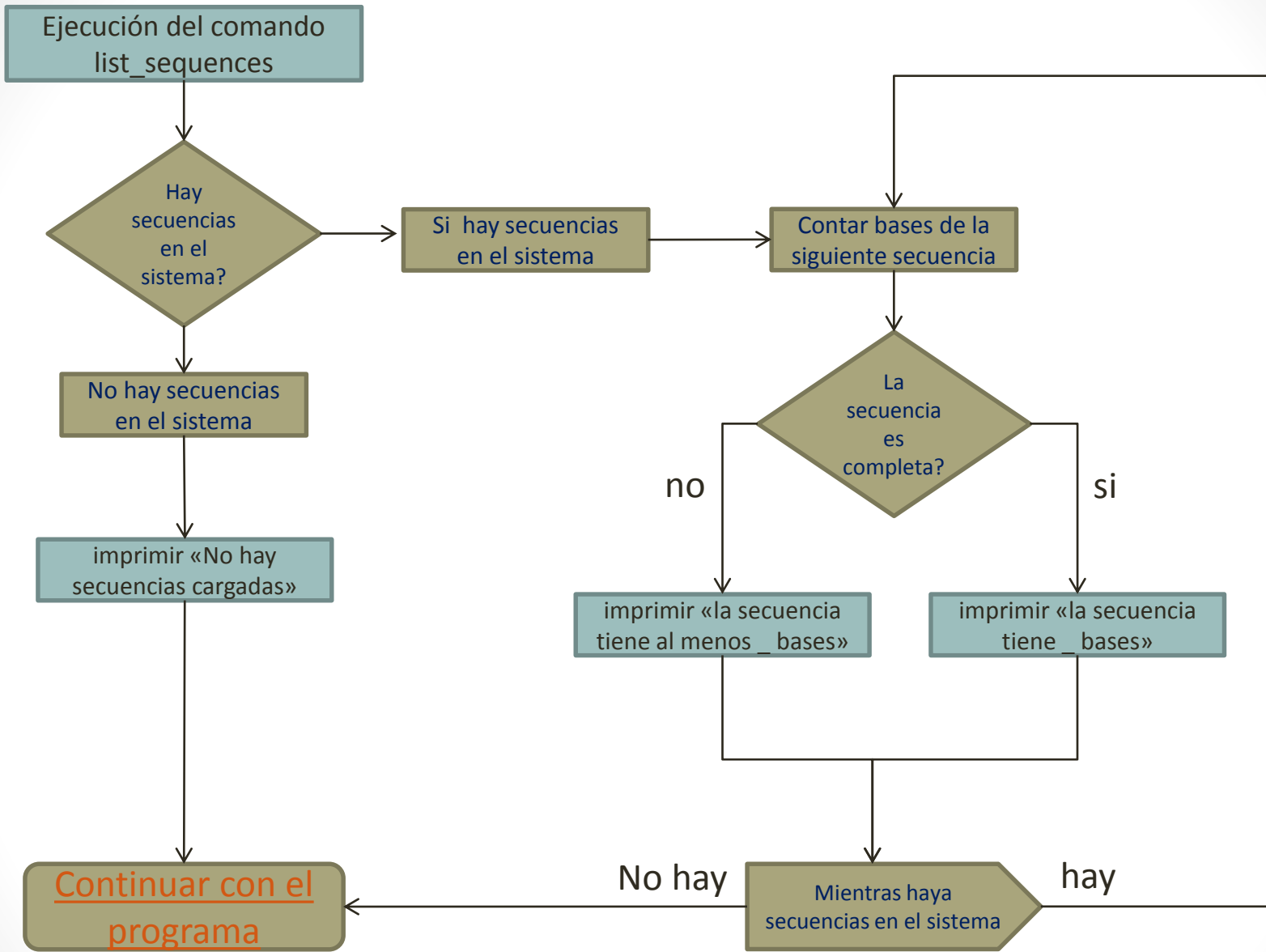
Nota: para navegar utilice los hipervínculos dentro de cada segmento.

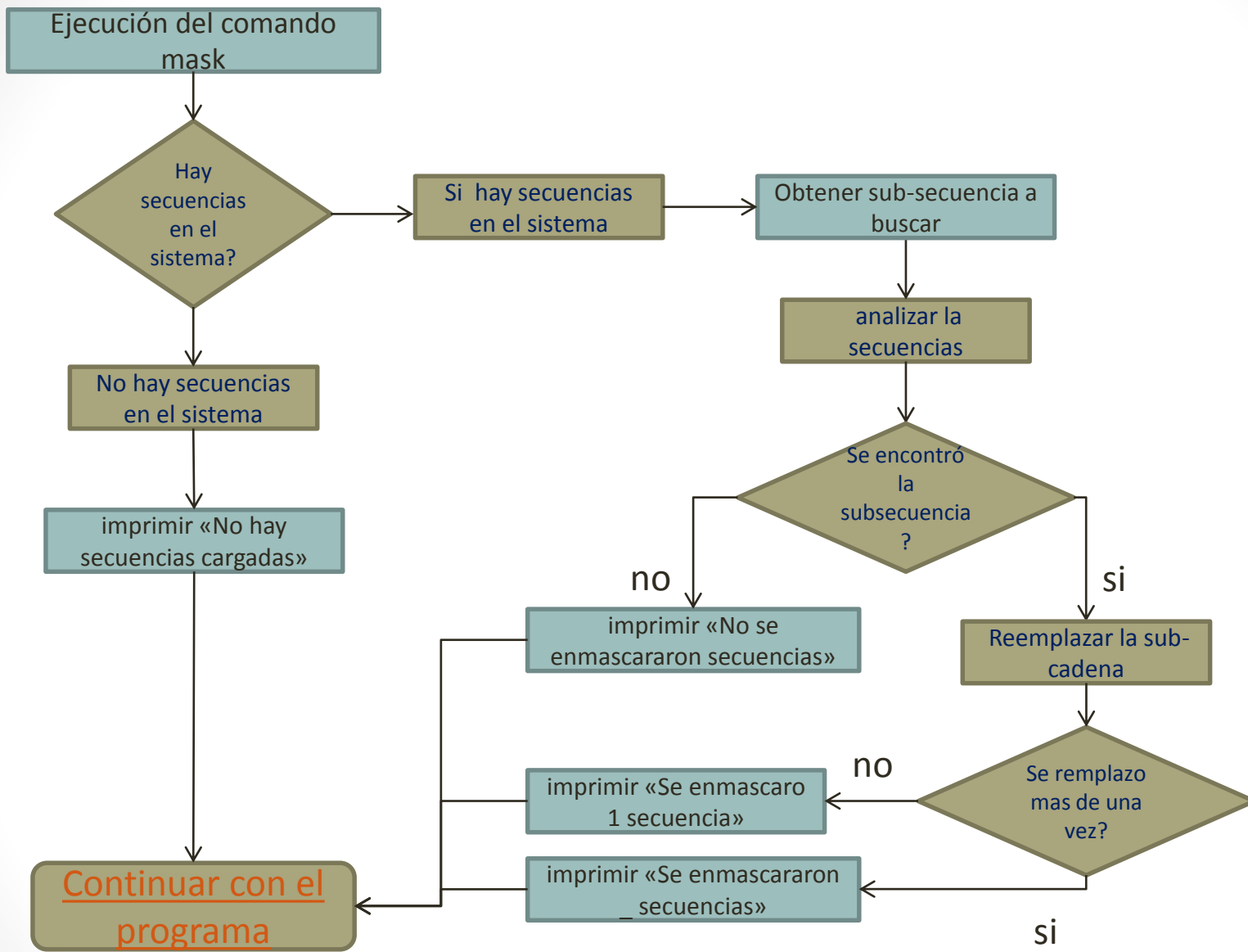
[Como se forma el Árbol de Huffman.](#)
[Como esta codificado el archivo binario.](#)

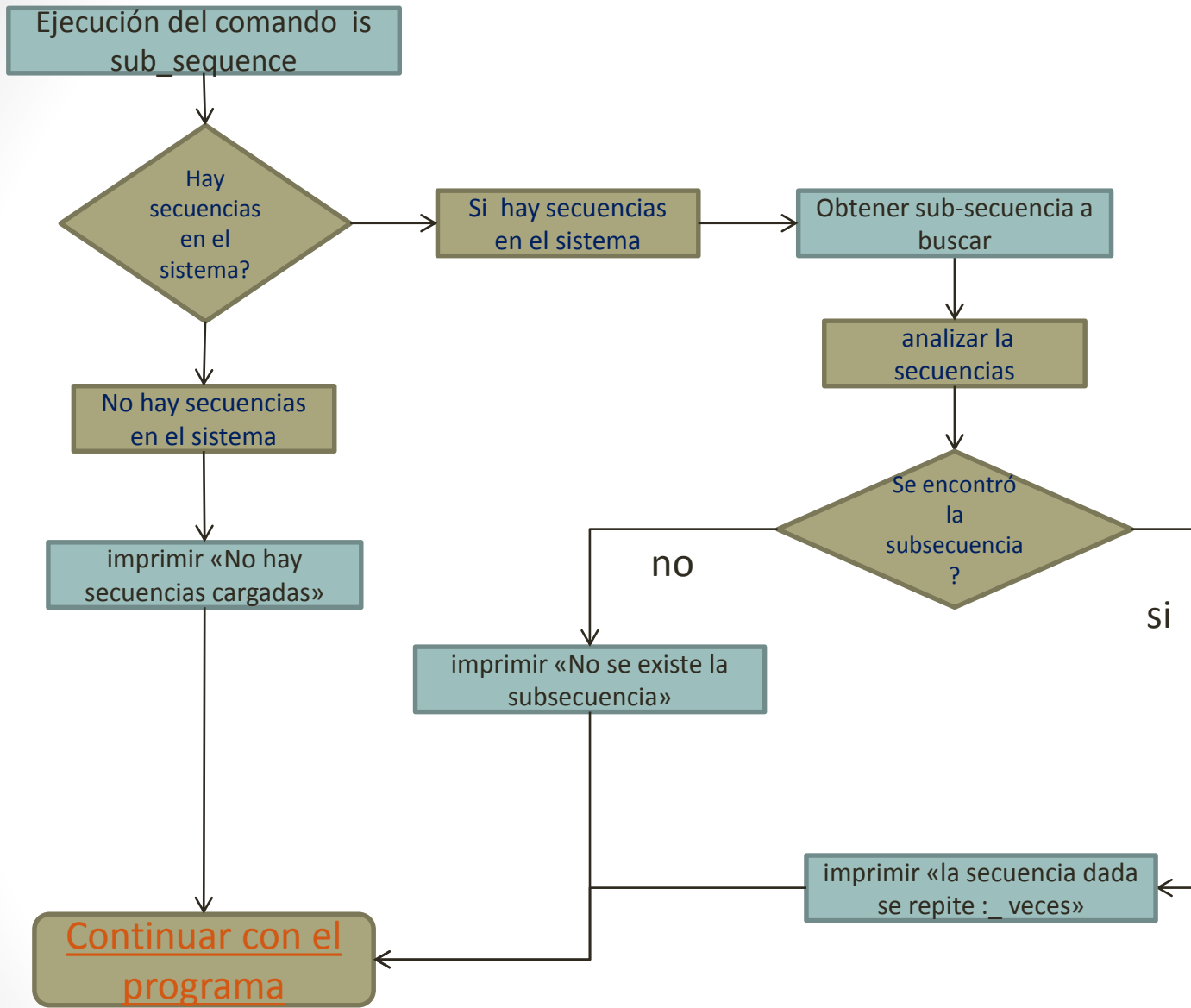


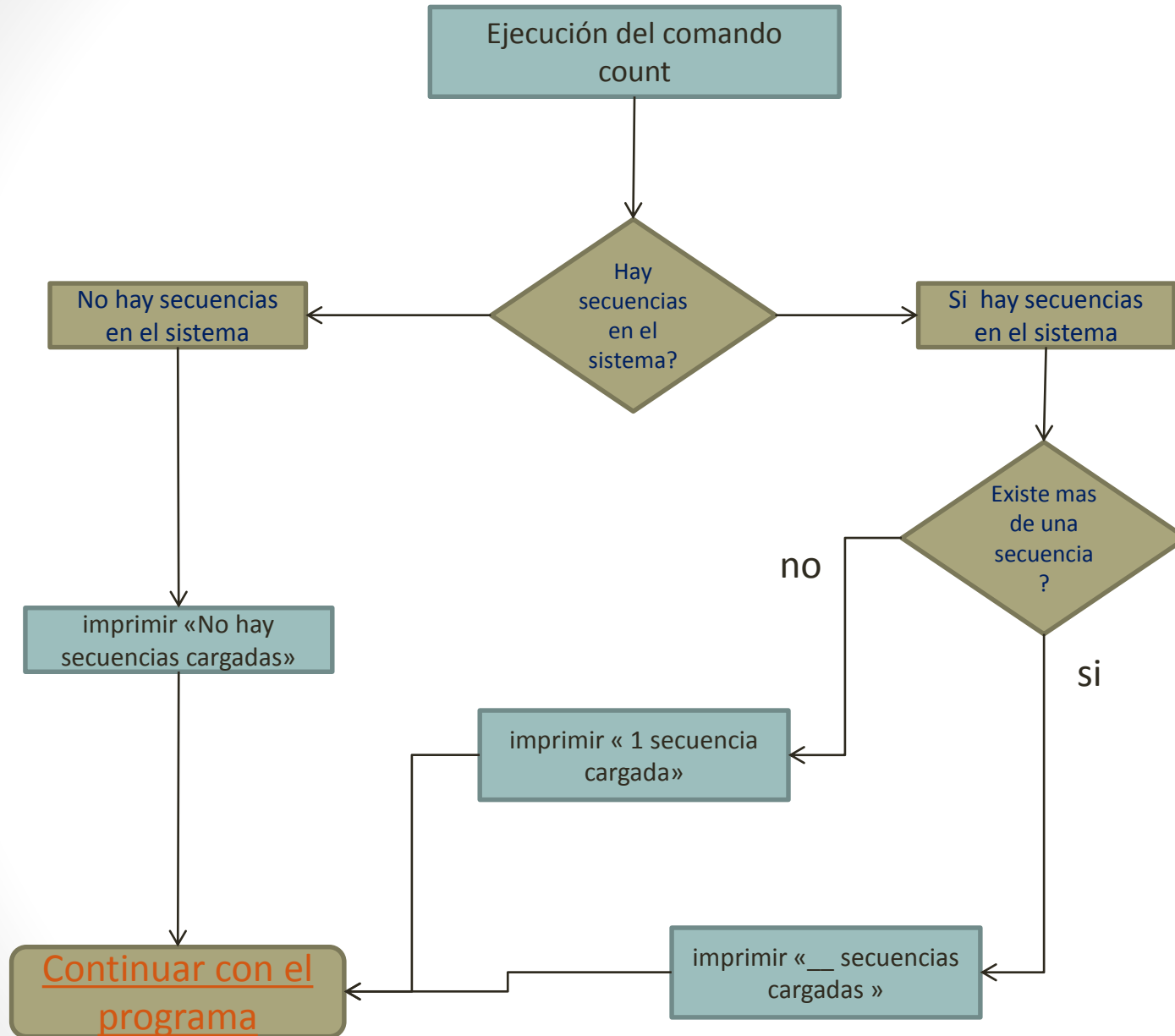


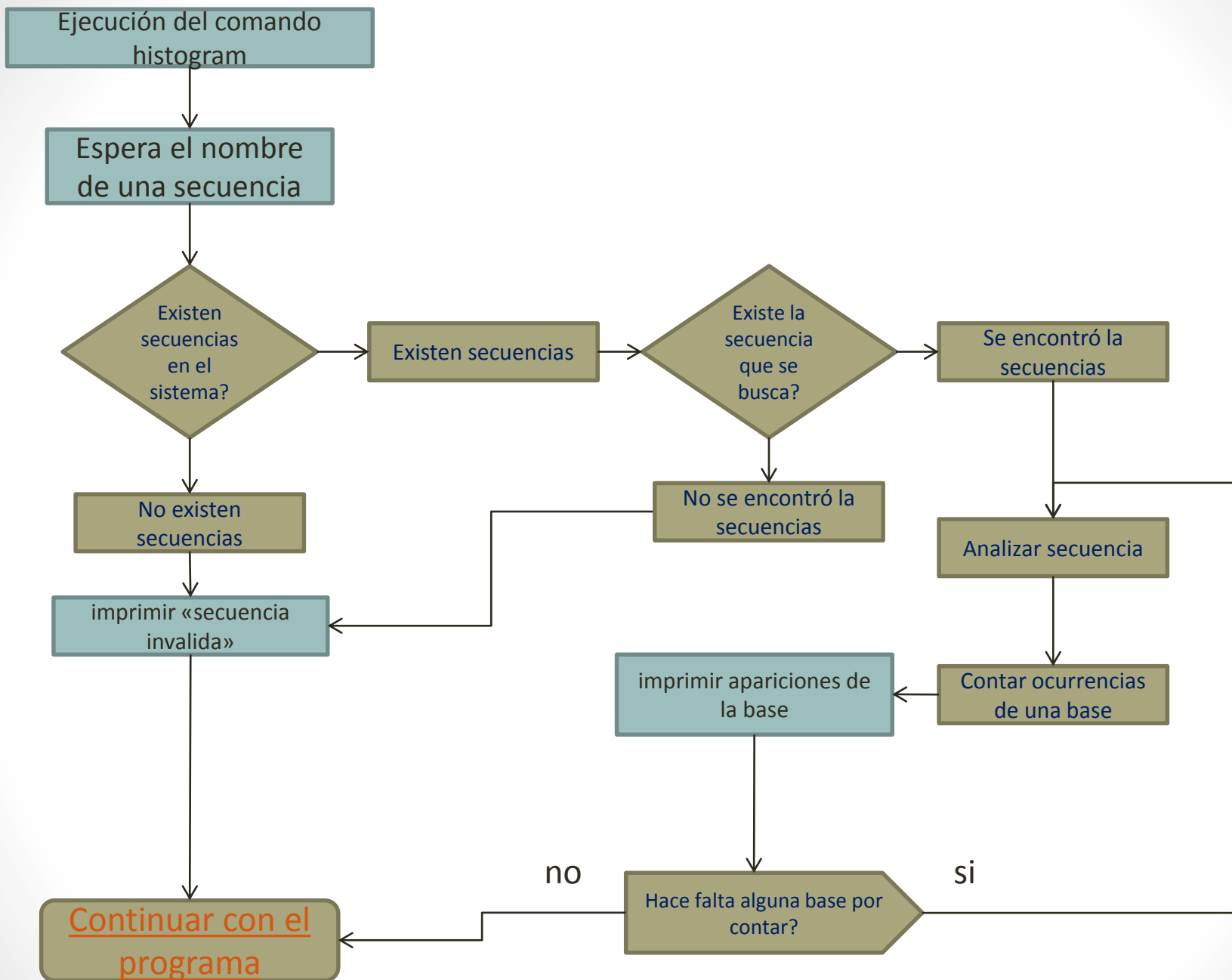


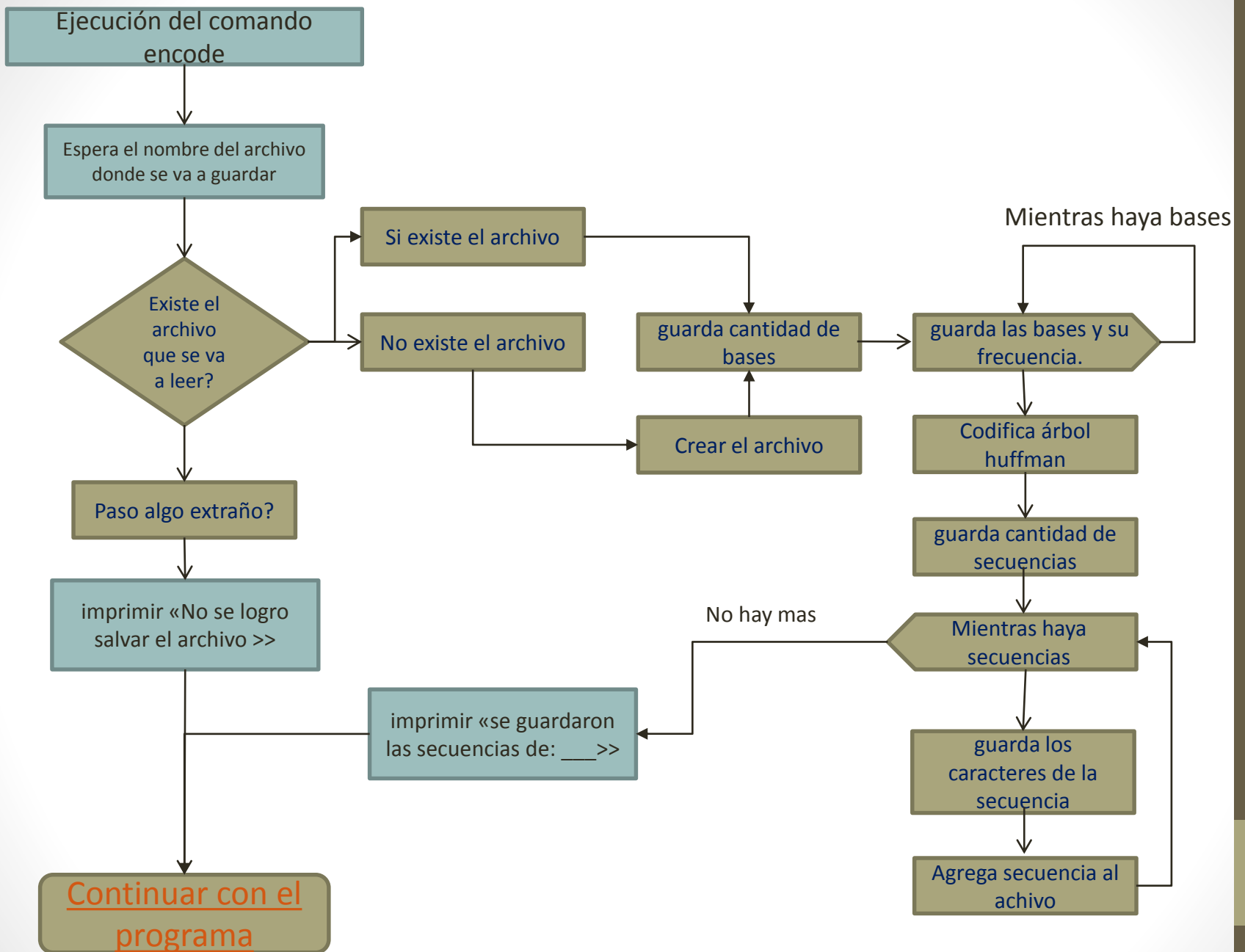


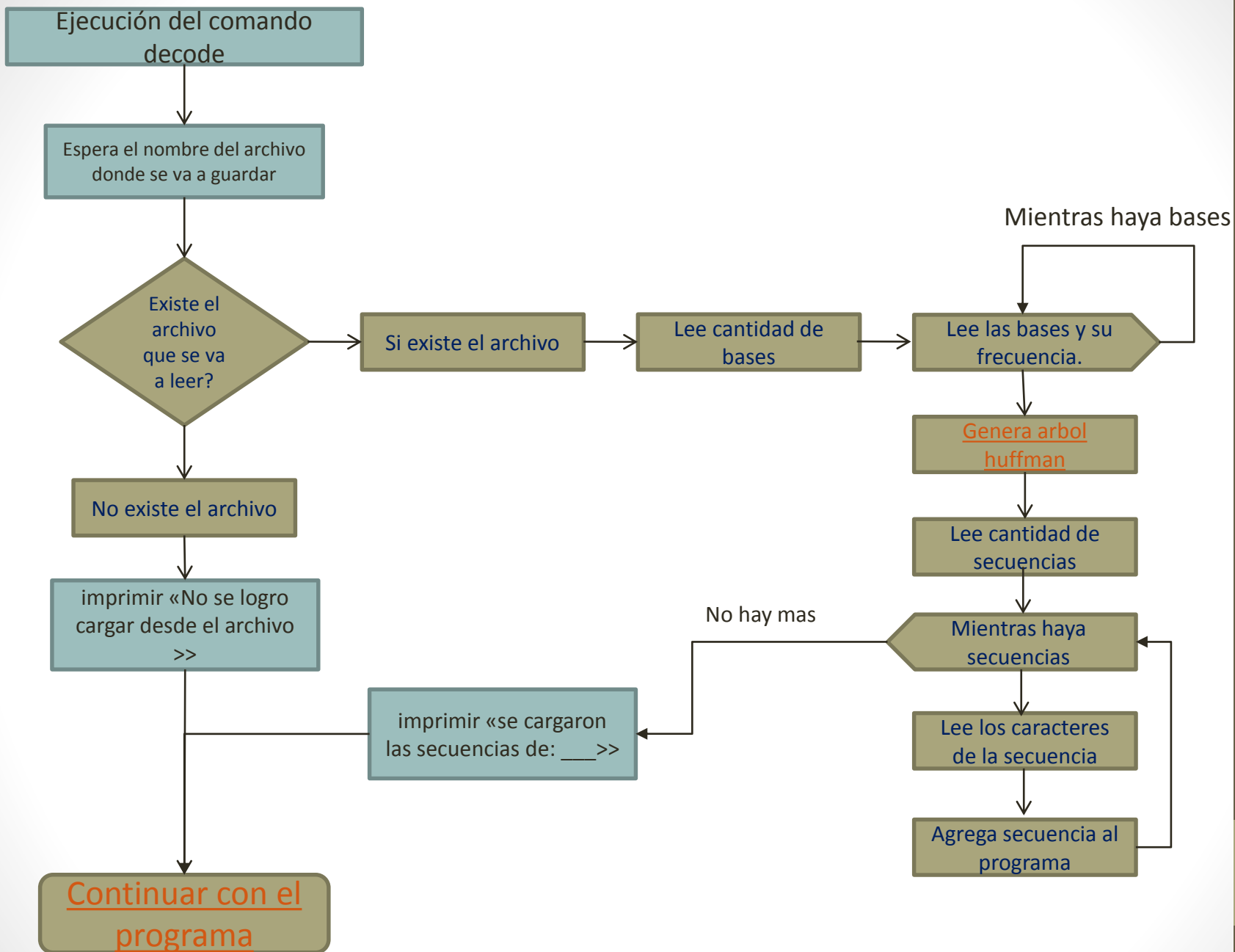




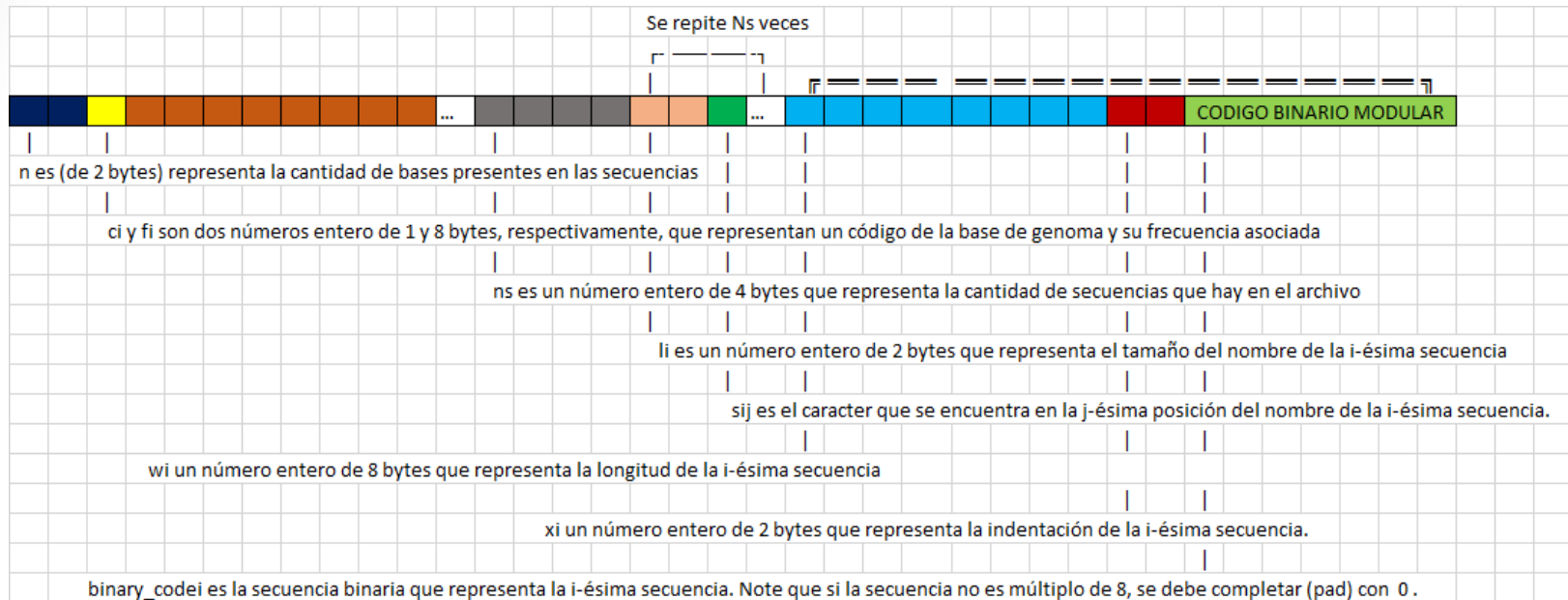








Como esta codificado el archivo binario:

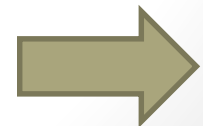


Nota: con código binario modular nos referimos a la secuencia de bits `binary_code`, donde un modulo individual representa cada letra dentro de una secuencia

[Volver](#)

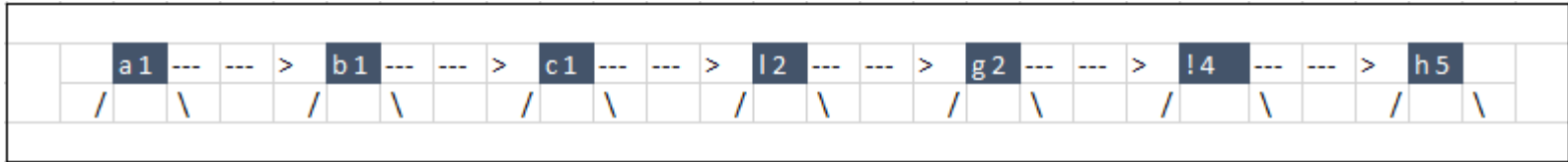
Como se genera el árbol de Huffman:

- Al momento de leer los datos, estos se agregan a un Map de carácter-frecuencia, donde se cuenta la cantidad total de todas las bases presentes en todas las secuencias del archivo que se lee.
- Por medio de una función, cada dato del Map se convierte en un NodoHuffman (remítase al documento adjunto del diseño del proyecto).
- Estos nodos se insertan en una Priority queue, que por medio de un comparator organiza todos los elementos al momento que se ingresa un nuevo nodo.
- A continuación se muestra como se forma el árbol a partir de esta priority queue, después de que todos los datos han sido insertados.



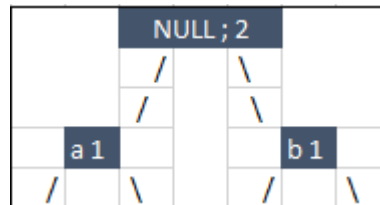
Generar el árbol a partir de la priority queue

Tenemos nuestra priority queue ya ordenada, el elemento mas a la derecha es El tope de la cola

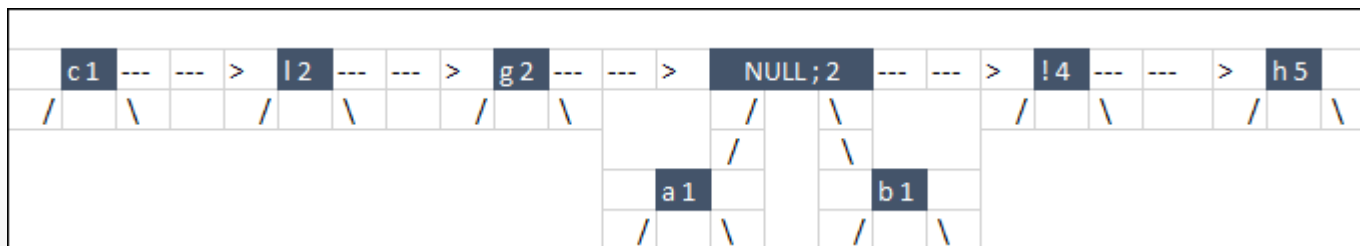


Los 2 primeros elementos de la cola se extraen y se les asigna un padre de carácter que nunca va a ser una base, para la explicación se usara un NULL para mostrar esta idea.

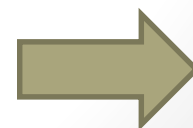
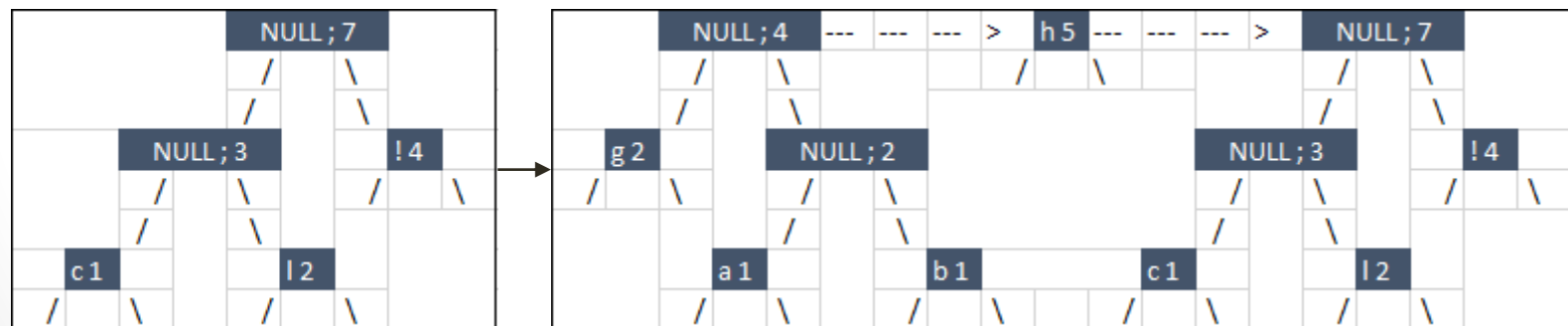
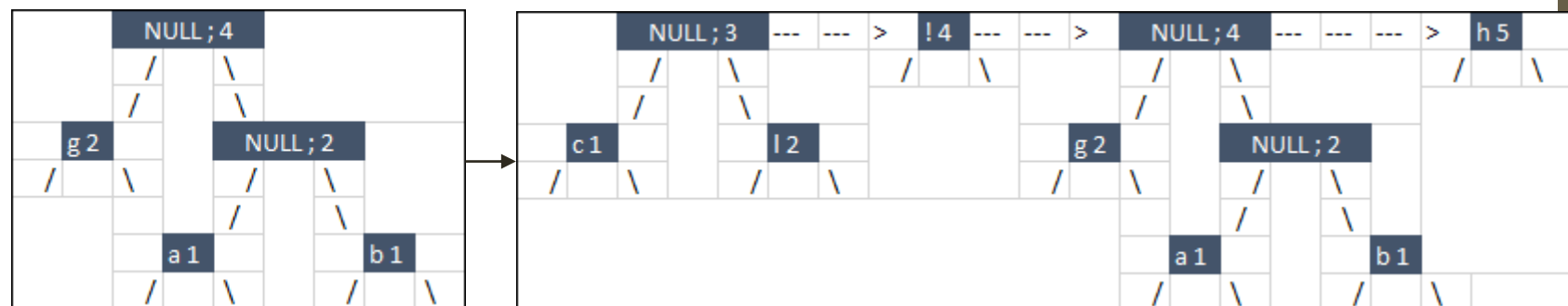
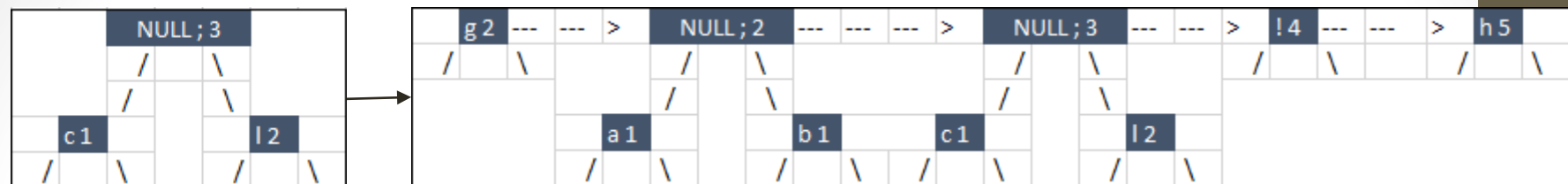
la frecuencia de este nodo padre es la suma de la frecuencia de sus 2 hijos:

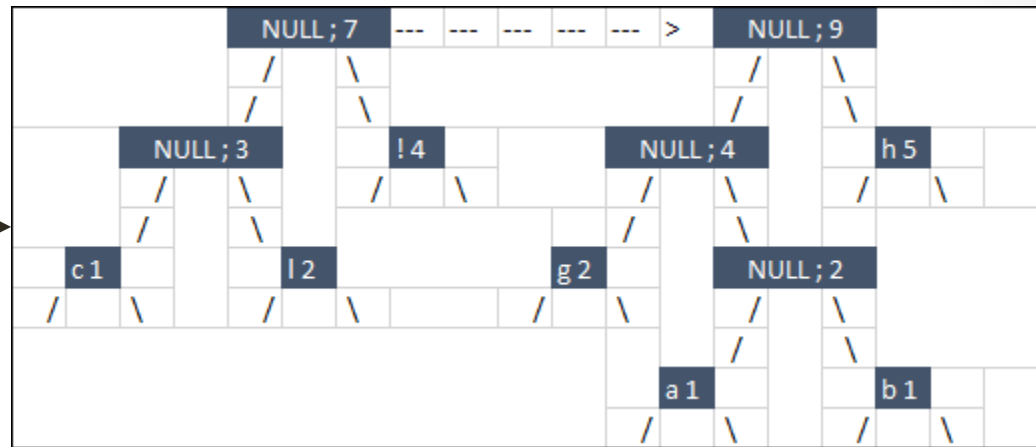
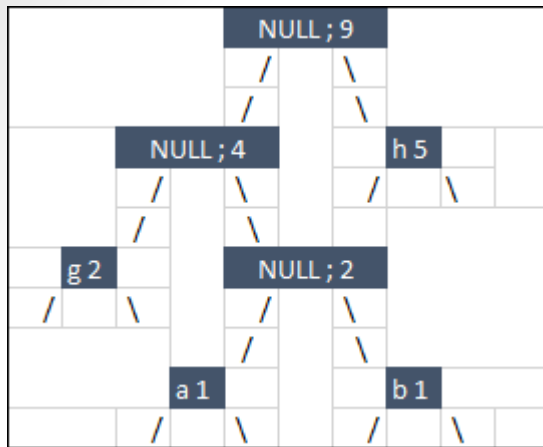


Este nuevo nodo se vuelve meter a la cola, y esta se ordena sola basado en la frecuencia:

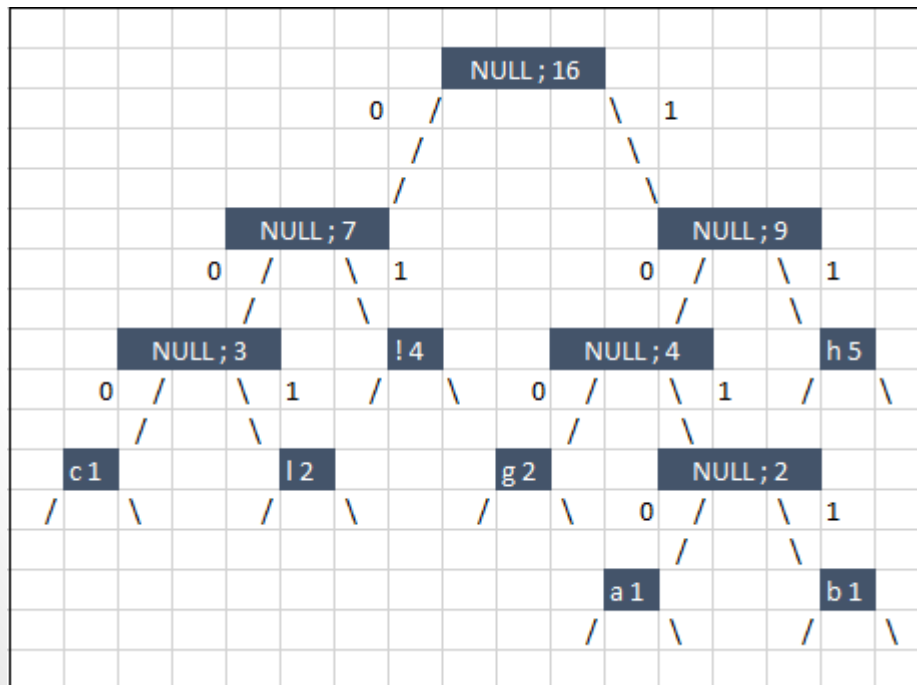


Se repite este proceso N veces:





Cuando quedan solo 2 nodos, se crea un padre que se convierte en la raíz de mi ArbolHuffman.



Cada letra tiene un código:

A	1010
B	1011
C	000
!	01
G	100
H	11
L	001

Y un mensaje como

a l ! ! c

Quedaría codificado así:

1010|001|0
1|01|000|00

[Volver](#)