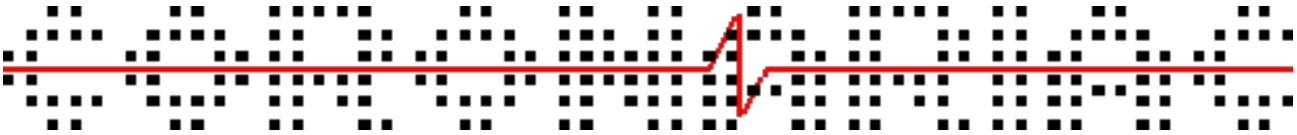
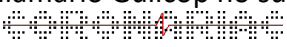


ÍNDICE

¿Qué es Coronariac?

Para poder hablar de Coronariac , primero, hay que hablar de su sistema original, **CARDIAC** (**CARD**board **I**llustrative **A**id to **C**omputation). Se trata de un ordenador de papel, sí, de papel.
Fue desarrollado en los años 60 por David Hagelbarger y Saul Fingerman para los laboratorios Bell, se trata de un sistema pensado para enseñarles a los estudiantes de la época, qué es un ordenador, cómo funciona e incluso, como programarlo.

Por esta parte, Coronariac se trata de un emulador del mismo, aunque tiene un par de diferencias clave que se abordarán más adelante, permite ejecutar los mismos programas que se ejecutarían en el Cardiac original.

En cuanto al nombre de Coronariac, bueno, Cardiac es un ingenioso acrónimo CARDboard Illustrative Aid to Computation es decir, Guía ilustrativa para la computación de papel, no obstante llamarlo Guilcop no suena tan bien y no soy tan ingenioso, de modo, que elegí
 que también hace referencia al corazón y termina en -ac en homenaje a todas esas computadoras antiguas como el ENIAC o UNIVAC .

Iniciando Coronariac.

Dado que Coronariac es un programa escrito en JAVA, usted debe asociar el archivo Coronariac.jar seleccionando JAVA el programa determinado para abrir los archivos .jar.

Otras opciones pueden ser:

Consola de windows:

```
javaw -jar Coronariac.jar
```

Linux/Mac

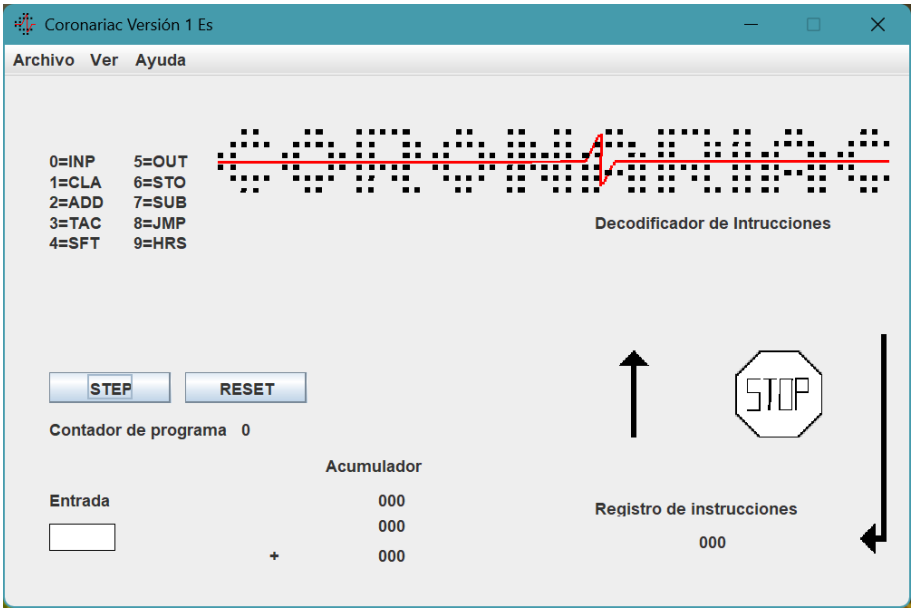
```
java -jar Coronariac.jar &
```

La razón de javaw y no java o poner “&” en Linux y Mac es para ocultar la consola durante la ejecución, aunque usted es libre de tenerla abierta y ver los mensajes de ejecución de cada parte de Coronariac.

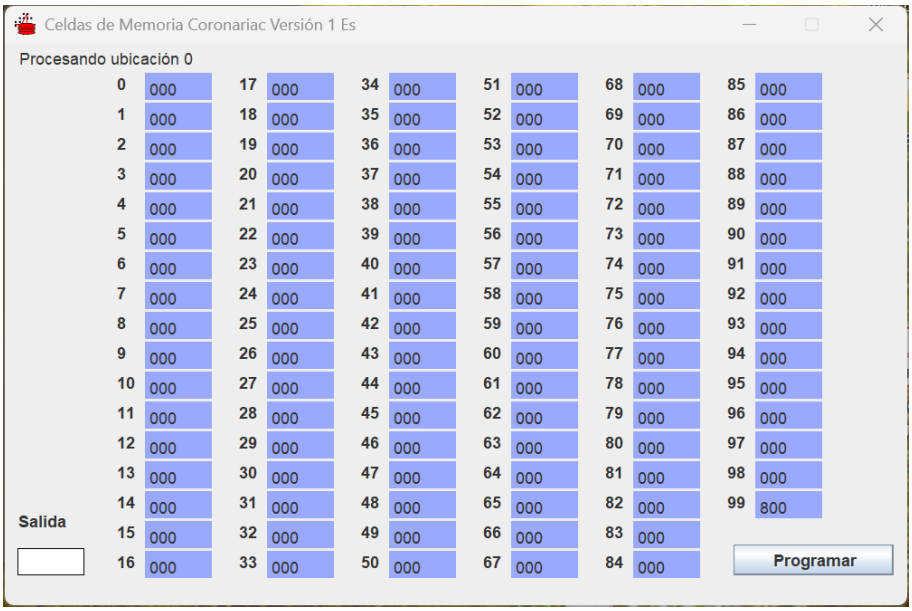
Repaso a la interfaz.

Cuando usted inicia Coronariac por primera vez, se encontrará con 2 ventanas:

ventana principal



ventana de memoria



Desglosemos una a una, el contenido de las mismas, empezando por la ventana principal:

- En primer lugar, se encuentra la barra de herramientas, en ella, puede encontrar:
  - **Archivo:**
    - Establecer archivo de entrada: al pulsarlo, se accede a un menú de exploración de archivos, con él, se selecciona un archivo .tjd que actuará como entrada
    - Establecer ubicación de salida: permite establecer una ubicación en su ordenador donde se generará el archivo "salida.tdj" el cual es la salida del Coronariac
    - Importar memoria: permite la búsqueda en el ordenador de un archivo ROM compatible con Coronariac.
    - Exportar memoria: permite guardar el estado de la memoria de Coronariac en su ordenador a través de un archivo .ROM.
    - Salir: detiene la ejecución de Coronariac y lo cierra.
  - **Ver:**
    - Ver contenido de la entrada: a través de este botón, usted puede ver lo que contiene el archivo .tjd que ha abierto a través de de Establecer archivo de entrada. Si usted se fija, cada vez que se introduce un dato de la entrada a la memoria, este se elimina de la entrada, pudiendo quedar vacía.
    - Ver contenido de la salida: Puede ver como se van agregando datos de salida, en cuanto ciclo de ejecución de su programa se termine, estos serán los datos que verá en salida.tjd.
  - **Ayuda:**
    - Acerca de…: Puede ver una copia de la licencia de Coronariac así como la versión actual.
    - Ayuda: una copia, aunque mucho más escueta de la documentación, para ser consultada en cualquier momento.
- Por otro lado, puede ver los botones Step ,Reset y el contador de programa:
  - **Step:** al pulsar sobre Step, se añade un ciclo de reloj y se ejecuta la instrucción que contenga la celda de memoria señalada,
  - **Reset:** el contador de programa se pone a cero.
  - **Contador de Programa:** permite ver cual será la siguiente posición de memoria que será analizada.

- Justo debajo de esta sección previamente descrita, se encuentra **la entrada**, en ella podrá ver cual es el siguiente dato que será introducido en la máquina, si es que se ejecuta la instrucción pertinente, claro.

- En el **acumulador** se puede ver la operación que se va a realizar y sus operandos, cabe destacar que aunque la suma de números de 3 dígitos puede dar un número de 4 dígitos como resultado, NO serán almacenados las unidades de millar, solo las unidades, decenas y centenas, por otro lado, también permite observar si el resultado de la operación da como resultado un número positivo o negativo.

- **Unidad de control**, justo debajo del logotipo, se encuentra la unidad de control, si se fija, puede observar unas flechas, que representan el flujo de ejecución del programa, también puede ver:

    - **Registro de instrucciones:** muestra al usuario la instrucción que se va a ejecutar

    - **Decodificador de instrucciones**: muestra una traducción a lenguaje humano, lo que realizan las instrucciones.

Por otro lado, en la ventana de memoria :

- **Celdas de memoria:** todos los rectángulos azules que ve, son las celdas de memoria. En ellas se almacenan las instrucciones y datos, junto a estas, puede ver el número, la dirección, asociada a estas.

- **Salida:** se muestra el dato que está expulsando coronariac para que el usuario lo vea, si usted suma 3+5, el 8, se mostrará ahí.

- **Botón Programar:** Cuando usted escribe un programa en la memoria del coronariac,debe pulsar aquí para guardar los cambios y que coronariac pueda procesarlos.

Archivos .tjd y .ROM

Coronariac utiliza dos tipos de archivos, por un lado, tenemos los archivos .tjd y por el otro, tenemos los .ROM

El sistema Cardiac original, la entrada y la salida utiliza una serie de tiras en las cuales se escriben o se toman datos de ellas según las instrucciones ejecutadas, así pues, la instrucción 0 por ejemplo, tomaría los datos de la tarjeta de entrada , mientras que la instrucción 5 escribiría en la tarjeta.

Estas tarjetas , están representadas mediante los archivos .tjd(tjd significaría **t**ar**j**eta de **d**atos). Realmente los .tjd son archivos de texto con una extensión florida.

Para crear un archivo .tjd bien se puede crear creando un programa en Coronariac que genere una salida.tjd, aunque también existe otra posibilidad, puede crearlos usted.

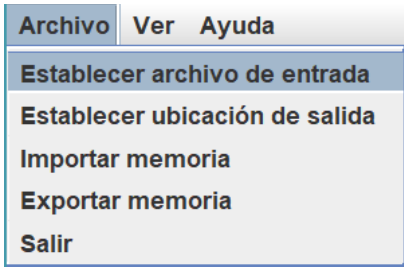
Un arhivo .tdj válido sería el siguiente:

[0] 154
[1] 343
[2] 342
[3] 076
[4] 099
[5] 999
[6] 145
[7] 645
[8] 001
[9] 453
[10] 101

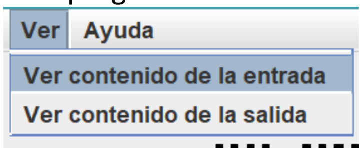
Como podemos ver, tenemos entre corchetes el orden de la entrada. Estos números no representan nada para Coronariac y serán ignorados al almacenarlos en su memoria, son enrealidad un indicativo para usted sin embargo, se recomienda encarecidamente su incorporación ya que Coronariac no reconocerá los datos a menos que delante tengan el número de orden.

En su archivo .tjd puede almacenar hasta simples datos hasta programas enteros que pueden ser cargados mediante un programa que cargue los datos a la memoria(puede consultar esto en el manual oficial de Cardiac).

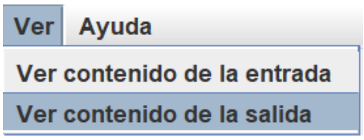
Para poder cargar su archivo .tjd en su copia de Coronariac, debe ir a Archivo,Establecer archivo de entrada, se le abrirá un menú explorador de archivos para localizar y abrir su .tjd



Si todo ha sido exitoso, puede ir a Ver,Ver contenido de la entrada y podrá ver una ventana con el contenido de la entrada, no se asuste si a lo largo de la ejecución de su programa desarrollado en Coronariac , los datos que puede ver aquí se van reduciendo, esto es normal y se corresponde con la utilización de esos datos por parte del programa.



Sin embargo, si lo que usted quiere es crear un archivo .tjd mediante coronariac, usted debe, lo primero establecer una ubicación donde se creará su .tjd, y esto se hace mediante archivo, establecer ubicación de salida y se abrirá un menú de exploración de ficheros donde usted debe seleccionar la carpeta deseada.



Establecida una ubicación, puede correr su programas normalmente, al llegar a la instrucción 900 y parar la máquina, se generará en la carpeta deseada un archivo de nombre “salida.tjd” con todos los datos que ha ido enviando a la salida durante la ejecución del programa, solo asegúrese de utilizar la intrucción 5 para guardar los datos deseados en la salida.

Por otro lado, como se mencionó anteriormente ademas de los .tjd, existen los .ROM.
Normalmente, los .ROM son archivos binarios destinados a ser cargados en un circuito integrado de memoria ROM, no obtante, decidí que se compartiera el nombre de la extensión ya que el objetivo del mismo es en la práctica el mismo, una representación del contenido de la memoria de Coronariac que puede ser cargado en Coronariac.

Al igual que los .tjd, en este caso, los .ROM son .txt con nombre florido, echemos un vistazo a su estructura:

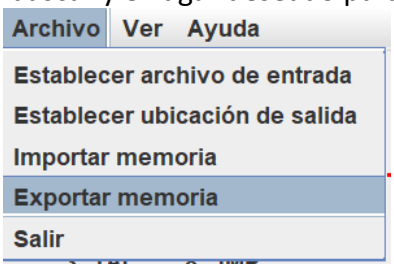
[024, 025, 124, 225, 626, 526, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 800]
(este programa es un derivado del que aparece en el manual oficial de Cardiac)

Este es un ejemplo de un programa escrito en Coronariac que toma dos datos de la entrada los suma y lo muestra en la salida , usted puede copiarlo, pegarlo en su procesador de texto favorito,

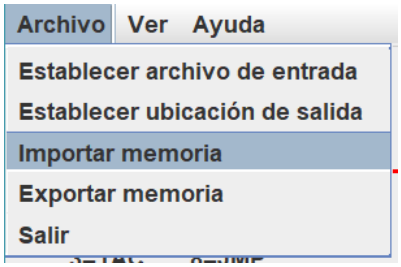
guardarlo como .ROM e importarlo en su copia de Coronariac y debería funcionar tal y como me funcinó a mí.

Pero, se debe aclarar algo y es que , como se mencionó previamente, Coronariac se toma unas libertades frente a su sistema original y eso es que el Cardiac sí puede tener espacios de memoria vacíos mientras que en Coronariac siempre van a tener “000”, además, Cardiac tiene hardcodeados las celdas 0 y 99 con “001” y “8__” respectivamente, mientras que en Coronariac esas mismas celdas tienen “000” y “800” pero sin hardcodear, usted es libre de modificarlo perfectamente. Sin embargo en cuanto a la celda 99, está reservada para ser usada con la instrucción 8, la cual almacenará en la 99 la ubicación de la memoria desde la cual se dió un salto, por lo cual se recomienda no modificar esta celda a menos que vaya a escribir un programa sin la instrucción 8.

Para poder generar sus archivos .ROM sólo vaya a archivo, exportar memoria y se le abrirá un menú de exploración donde podrá buscar y el lugar deseado para guardar su archivo.



Por otro lado, para hacerlo a la inversa e importar un .ROM previamente creado, sólo debe pulsar archivo , importar memoria, donde se le abrirá otro menú de exploración, esta vez, para buscar y cargar su archivo .ROM

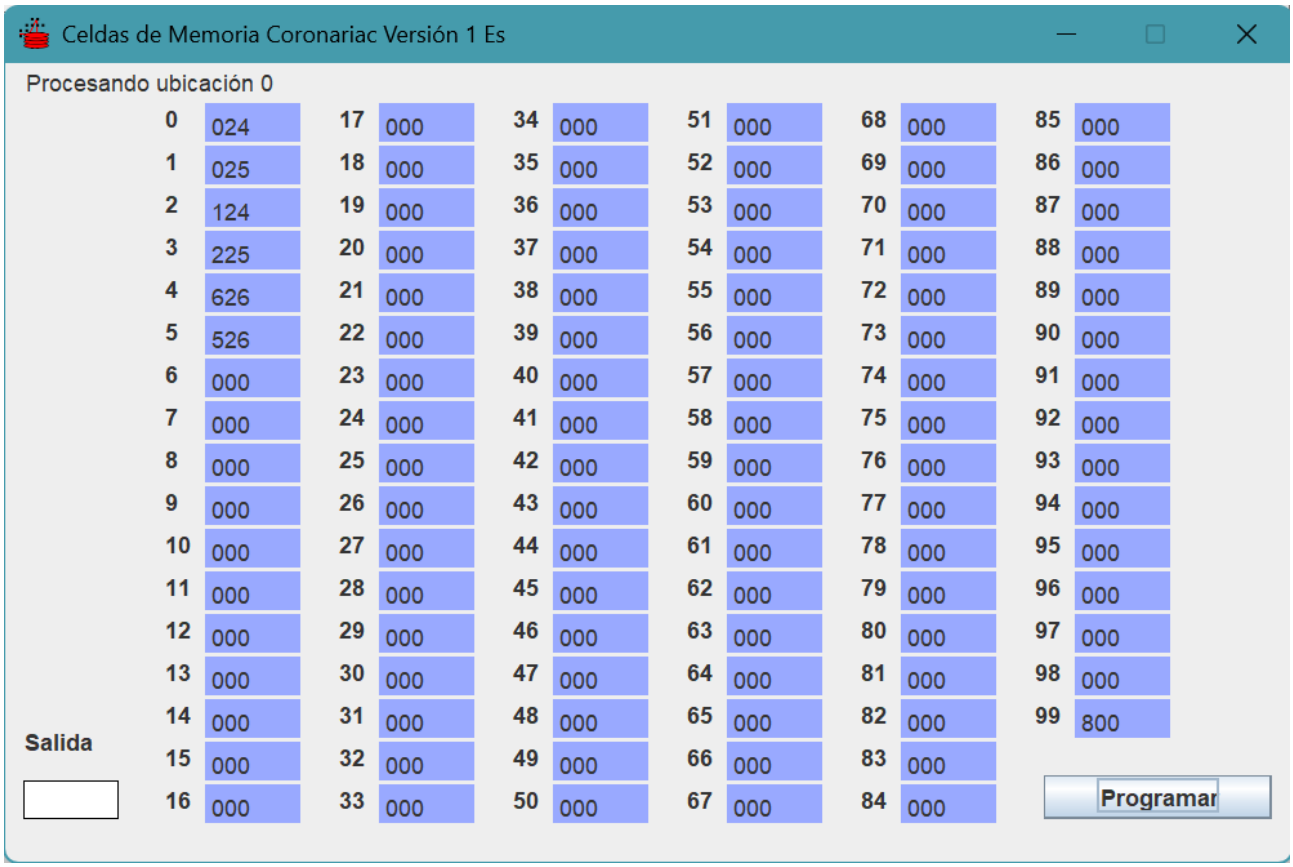


Tras esta no tan breve explicación usted debería ser capaz de generar archivos tanto .ROM como .tjd

Escribiendo un programa en Coronariac.

Ahora que usted ya sabe navegar por Coronariac, es hora de crear su primer programa escrito en este ordenador.

Coronariac, basado en Cardiac, tiene 10 opcodes, o instrucciones pero no usa binario para procesar sus datos. Debida a la naturaleza educativa de Cardiac, se eligió el sistema decimal para facilitar su lectura e interpretación, la memoria, es capaz de almacenar a la vez tanto datos como instrucciones, esto significa que si no se tiene cuidado(o si se pretende) un número que usted ha guardado como número puede ser interpretado como instrucciones.



En esta ventana podemos ver que hay algo en la memoria desde las celdas 0 a la 5, es realmente el mismo programa que se se mostró previamente. Analicemos la celda 3

en la celda 3 podemos encontrar escrito “225” , a priori esto no nos dice nada pero enrealidad 225 se corresponde más bien con 225, siendo 2 la instrucción o más apropiadamente Opcode y 25 que sería el dato, en este caso, 225 quiere decir “sumar lo que haya en la celda 25”.

Aunque en la ventana principal hay un recordatorio de lo que hace cada instrucción y muestra lo que hace cuando se ejecuta

0=INP	5=OUT	Llevar al acumulador el contenido de la celda [25]
1=CLA	6=STO	
2=ADD	7=SUB	
3=TAC	8=JMP	
4=SFT	9=HRS	

Conviene igualmente desglosarlos uno a uno

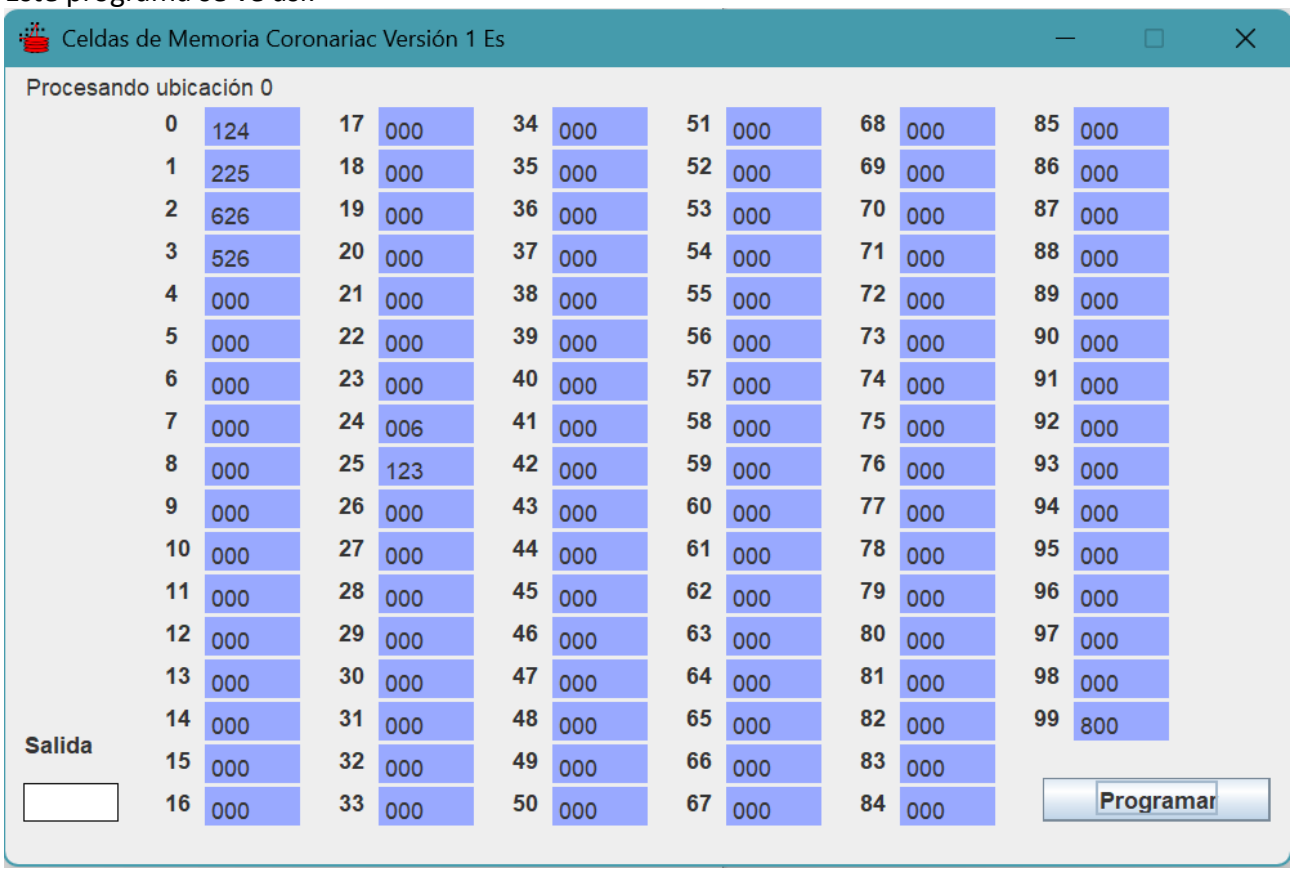
0. **INP**: toma el dato de entrada y lo lleva a la posición de memoria deseada(osea 024 llevaría el dato de entrada a la celda 24).
1. **CLA**: borra el contenido del acumulador y añade como primer operando el contenido de la celda dada (124 vendría a indicar que se tome lo que sea que haya en la celda 24).
2. **ADD**: suma el contenido del acumulador con lo que haya en la celda dada(224 sumaría el contenido del acumulador con lo que haya en la celda 24).
3. **TAC**: este Opcode es interesante, comprueba si el resultado del acumulador es positivo o negativo, si es positivo, se sigue con el flujo normal del programa, si es negativo,el contador de programa saltará a la ubicación dada(324, si es negativo, llevará al contador de programas a la posición 24).
4. **SFT**: Este opcode hace un desplazamiento lateral, en lugar de tener instrucción y dato, se divide en instrucción, dato 1 y dato 2 es decir 424 en este caso, el primer dato corresponderían al numero de posiciones a la izquierda que se desplazaría el resultado del acumulador y el segundo son las posiciones a la derecha, los **números desplazados se pierden para siempre**.
5. **OUT**: simplemente transporta a la tarjeta de salida , el contenido de la celda señalada(524 llevaría a la salida el contenido de la celda 24).
6. **STO**: guarda el resultado del acumulador a una celda deseada(624 transporta el contenido del acumulador a la celda 24).
7. **SUB**: es similar a la función 2, pero con una resta(724 restaría al acumulador el contenido de la celda 24).
8. **JMP**: realiza un salto incondicional a la celda que señale el dato y guarda la ubicación original a la celda 99(osea si usted se encuentra en la posición 5 y se encuentra con 824, usted saltaría a la posición 24 pero se guardaría su posición original en la celda 99 quedando 809 en la celda original)
9. **HRS**: Para la máquina y genera su archivo salida.tjd si tenía una ubicación establecida.

Una vez escrito su programa, debe pulsar sobre el botón programar, entonces lo escrito se almacenará en la memoria.

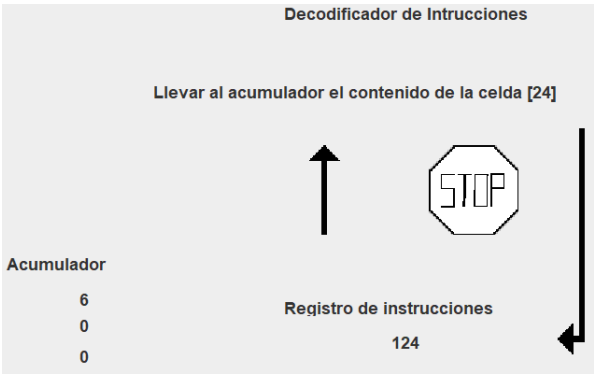
Ejecutando programas.

Ahora que ya sabemos programar en Coronariac, vamos a proceder a la ejecución de un programa. Usaremos de ejemplo, un programa que tenga ya almacenados 2 números, los sume y los muestre por la salida.

Este programa se ve así:

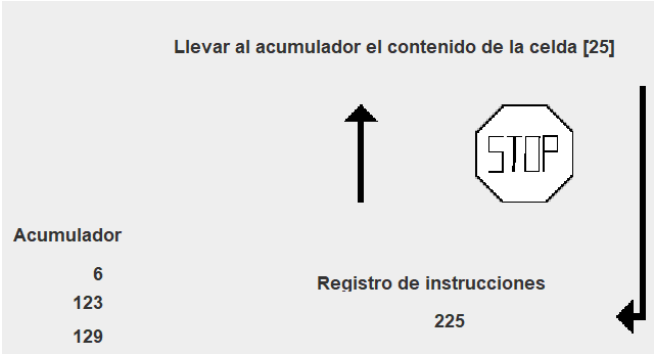


Una vez escrito y programado, en la ventana principal se encuentra el botón [STEP]



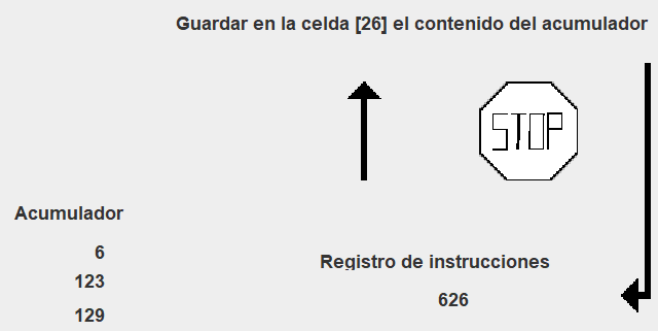
Con el primer cilo, podemos ver que se muestra la instrucción a ejecutar, con su significado y efectivamente, se ha llevado al acumulador el número de la celda 24, osea el 6

volviendo a pulsar [STEP] vemos lo siguiente:



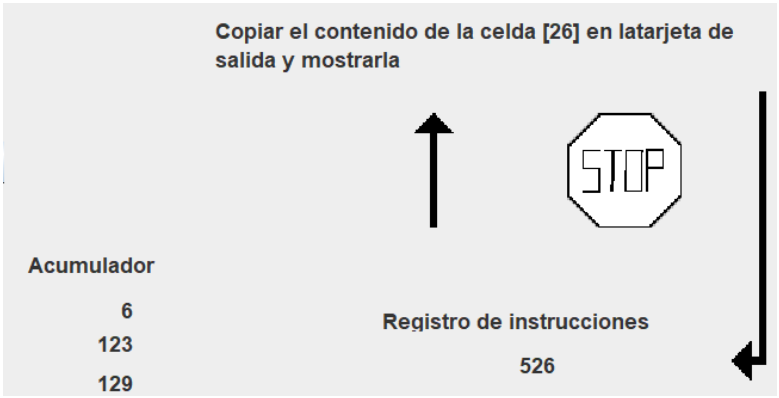
Ahora observamos que se ha cargado el número de la celda 25, es decir 123 y se ha sumado, dando como resultado el número 129

A continuación, el botón [STEP] de nuevo y se nos muestra lo siguiente:



Aquí vemos que la instrucción 626 indica que se va a guardar en la celda 26 el contenido del acumulador y si nos vamos a la ventana de memoria, vemos que efectivamente en la celda 26, se ha guardado el 129:

Finalmente, pulsando [STEP] de nuevo :



Si vemos lo que dice, vamos a la ventana de memoria y nos encontramos que efectivamente en la salida se encuentra el contenido de la celda 26:

Y ahora...¿Qué?.

El desarrollo de Coronariac está terminado, sin embargo, es posible que en el futuro se puedan añadir cambios, siempre surgen errores imprevistos, de modo que cualquier cambio se añadirá aquí.

MUCHAS GRACIAS POR UTILIZAR CORONARIAC