

# Sistemas Microprocesadores

## Memoria Del Proyecto

### ASIGNACIÓN DE ASIENTOS DE AVIÓN



Alejandro Cano Segovia  
Carlos Soria Mora

GRUPO B3

El trabajo consiste en desarrollar un programa en ensamblador 68k sobre el entorno EASY68K que gestione la asignación de asientos de un avión.

A grandes rasgos, este programa nos permitirá elegir el tamaño del avión, consultar asientos, asignarlos, inicializarlos y generar una interfaz gráfica que nos ayude a visualizar la situación de todos los asientos.

## BLOQUES FUNCIONALES

MENSAJES.....	2
BASEDATOS DE PASAJEROS.....	2
START.....	2
GENERADOR_ASIENTOS.....	4
MENU_PRINCIPAL.....	4
CONSULTAR.....	4
ASIGNAR.....	6
COMPROBADOR_ASIENTO_INICIAL.....	8
CALCULADORA.....	9
INICIALIZAR.....	11
BORRAR_CODIGO_TABLA.....	11
TABLA E INTERFAZ GRÁFICA.....	12
GENERADOR_CODIGO.....	12
GENERADOR_HUECOS_CODIGO.....	13
RECORREDOR_MEMORIA_GEN_COD.....	13
GENERACIÓN DE LA TABLA.....	14
GENERACIÓN DE LA INTERFAZ.....	16

## MENSAJES

Lo primero que haremos será crear los mensajes que saldrán por ventana para poder guiarnos a través del programa, más tarde añadiremos los caracteres ASCII necesarios para realizar la tabla y la interfaz del avión.

Los guardaremos a partir de la posición \$5000

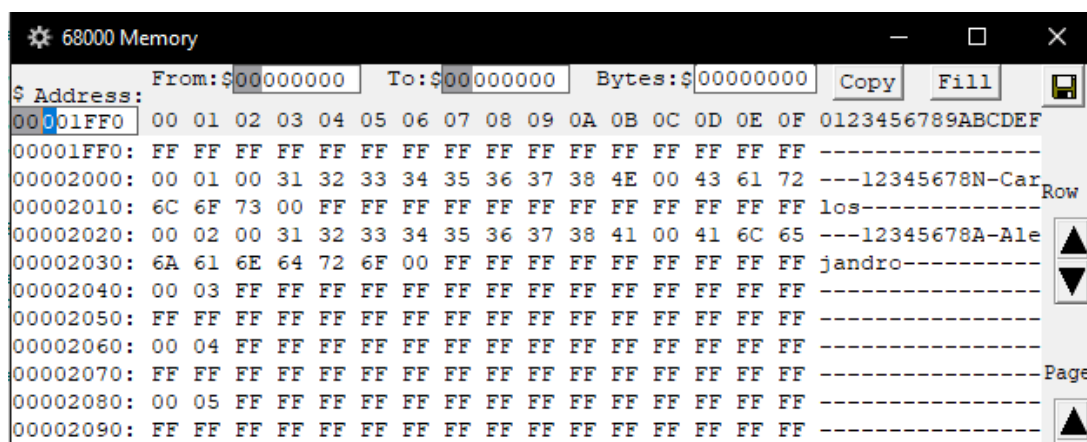
## BASEDATOS DE PASAJEROS

Definimos el espacio de memoria que reservaremos a los asientos, en concreto serán 3000 Bytes, a partir de la posición \$2000.

Contamos con que los datos de cada pasajero van a ocupar un máximo de 32 Bytes, y cada uno tendrá su código en memoria.

Los primeros 2 Bytes almacenarán el número de asiento que ocupa ese pasajero, el siguiente nos dirá si ese asiento está ocupado o no (Si el Byte es \$FF querrá decir que está libre, si es \$00 está ocupado el asiento), los siguientes 9 Bytes corresponden al campo DNI y los 20 que restan al Nombre.

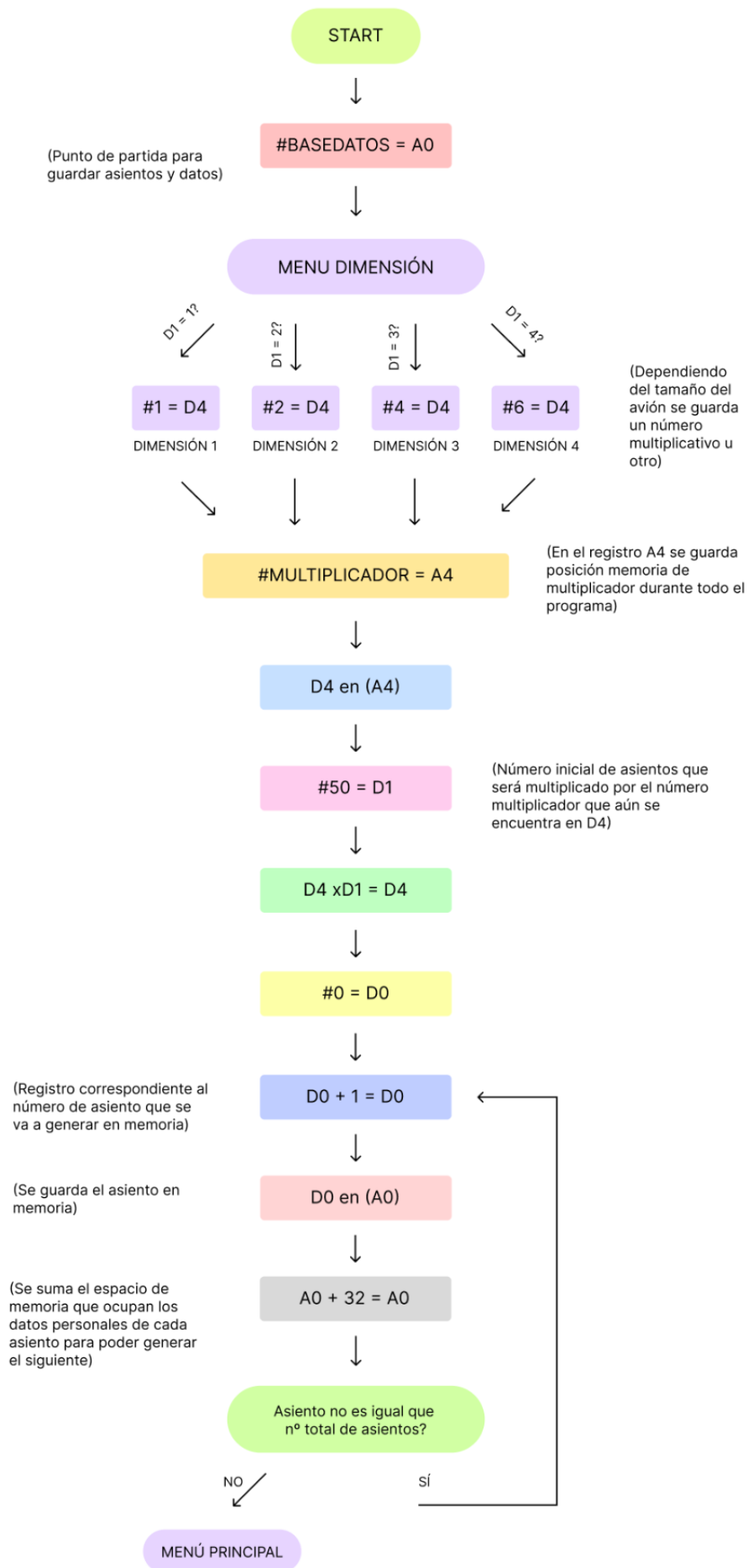
```
ORG $2000
BASEDATOS DS.B 3000
```



## START

Aquí situaremos el comienzo del programa, la parte que no queremos que se vuelva a ejecutar hasta volver a inicializar o hacer un reinicio al programa.

Comenzaremos preguntando por consola el tamaño del avión, según la opción que se haya elegido en memoria, se guardará dicho número en un espacio de memoria, para poder usarlo a lo largo del programa. El registro de dirección A4 apuntará siempre a la etiqueta MULTIPLICADOR que reserva un Byte para este número multiplicador, y podremos ir extrayendolo cuando lo necesitemos.



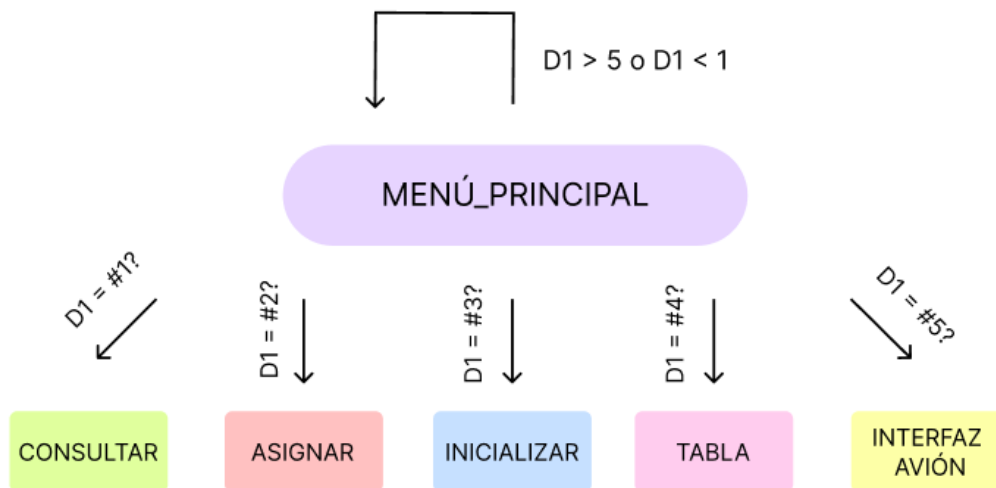
## GENERADOR\_ASIENTOS

Este bloque generará en memoria los números de los asientos en los dos primeros Bytes cada 32 Bytes (Espacio reservado a cada pasajero) a partir de la posición \$2000.

Dependiendo de la dimensión de avión que hayamos seleccionado, este será el primer caso que actuará el multiplicador, si elegimos 50 asientos, el multiplicador es 1 por tanto se queda igual, si es 100 el multiplicador sería 2 y así sucesivamente, con esto podremos establecer la salida del bucle GENERADOR\_ASIENTOS.

## MENU\_PRINCIPAL

Este bloque será el punto de ida y retorno en nuestro programa, para seleccionar las diversas opciones que nos brinda:

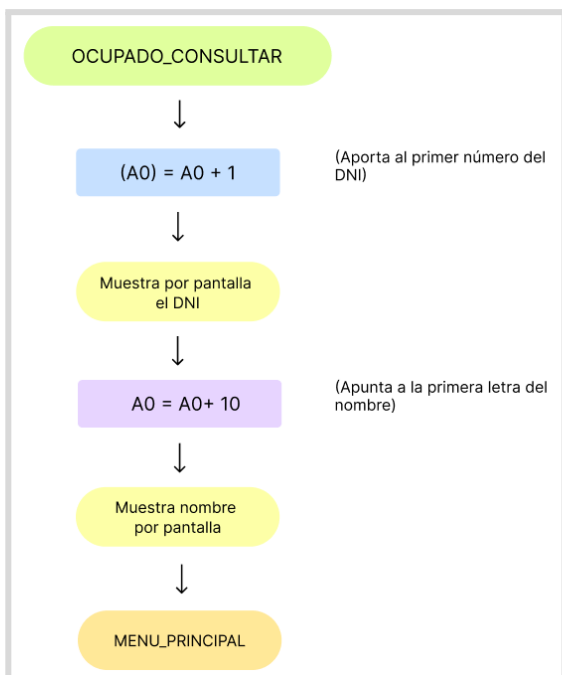
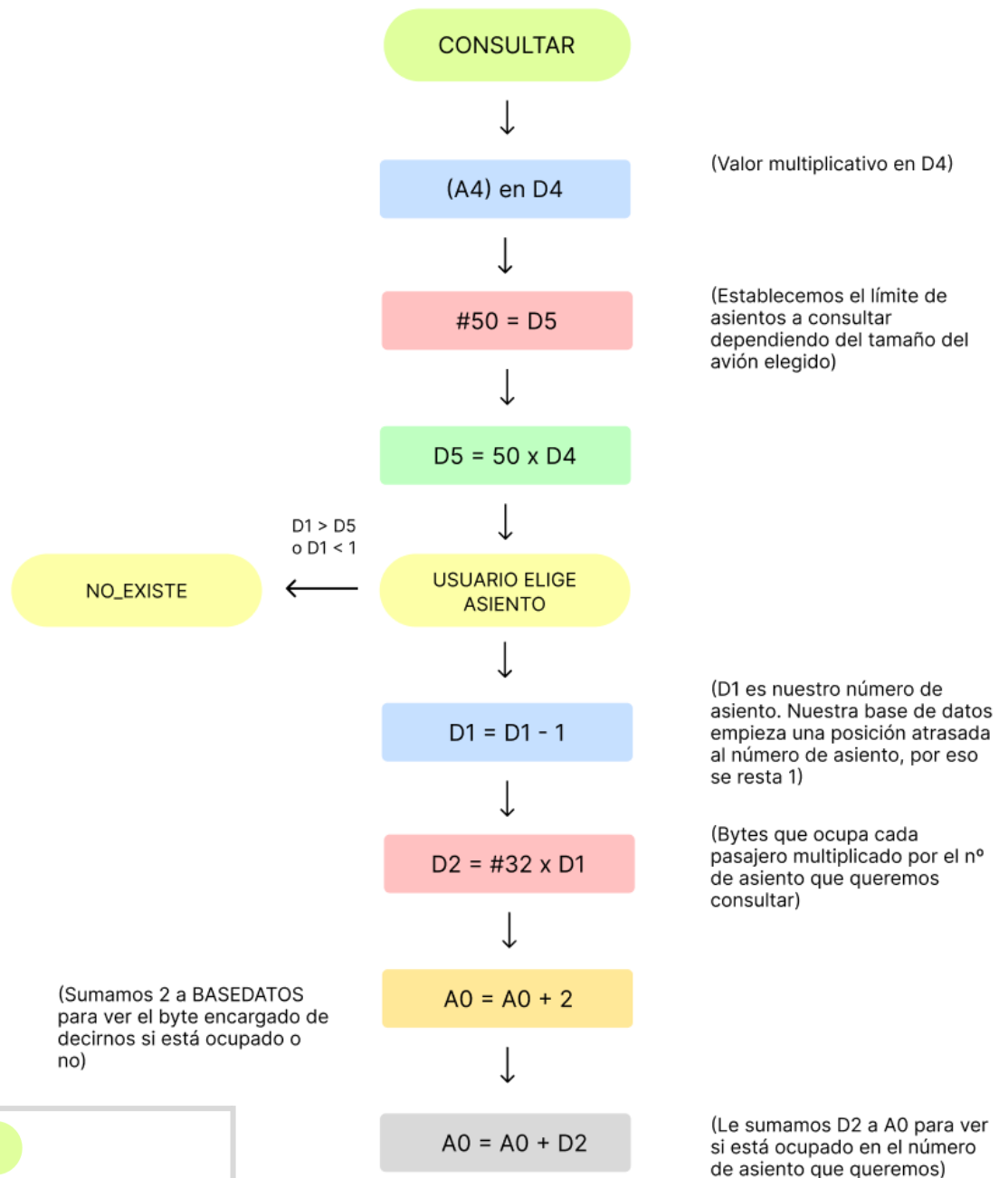


## CONSULTAR

Con este bloque podemos consultar los asientos introduciendo el número de asiento que queramos, si está libre nos devolverá un mensaje comunicándonoslo, si está ocupado nos dirá que está ocupado por el pasajero con DNI...y NOMBRE....

Empezaremos por pedir el número del asiento ( $D1$ ), tras esto comprobaremos si es mayor que 0 y menor que  $D5$  ( $D5 = 50 \times D4(\text{MULTIPLICADOR})$ ), si no está en ese intervalo ese asiento no Existe.

Si ese número de asiento existe, comprobaremos el 3º Byte de su código para ver si está libre o ocupado.  $D1$  guardará el número de asiento, le restamos uno, ya que nuestra base de datos empieza una posición atrasada al número de asiento que le corresponde (Por ejemplo el asiento 1 está en la \$2000) apuntaremos con  $A0$  a la BASEDATOS y procederemos a multiplicar  $D1 \times 32$  (espacios que ocupa cada pasajero), esta multiplicación hará que si sumamos el resultado y BASEDATOS, nos proporcionará la dirección del asiento que buscamos, sumamos 2 Bytes más para llegar al Byte que dice si está ocupado, si lo está el programa pasará a un bucle donde recorrerá el DNI y el Nombre, para mostrarlos por pantalla y volverá al MENU\_PRINCIPAL.



## ASIGNAR

De forma general, la función de asignación se debe realizar sabiendo la clase y la posición del asiento, el programa deberá buscar de forma descendente un asiento que cumpla esas características.

Para asignar los asientos, tras seleccionar la opción en el MENU\_PRINCIPAL, y entrar en ASIGNAR nos pedirá elegir la clase de asiento (Primera clase y Turista).

- Si elegimos Primera clase se guardará en D7 un 4, que es el número de asientos posibles para cada una de las opciones (Ventana y pasillo).
- Si elegimos Clase turista se guardará en D7 un 14, que es el número de asientos posibles para cada una de las opciones (Ventana, centro y pasillo).

Para la ampliación del avión se usará el número Multiplicador que guardamos anteriormente, este multiplicará los asientos posibles para cada opción.

Después de esto nos pedirá según la clase que hayamos elegido que seleccionemos dos o tres opciones (Primera Clase(ventana, pasillo) o Clase Turista(ventana, centro, pasillo))

Las 5 posibilidades guardarán unos valores según la clase y posición que elijamos, se usarán posteriormente en una calculadora de asientos.

Clase	PRIMERA CLASE		CLASE TURISTA		
D7	4		14		
Posición	Ventana	Pasillo	Ventana	Centro	Pasillo
D3	0	1	8	9	10
D5	3	1	5	3	1
D6	1	3	1	3	5

D7 : nº Asientos Disponibles

D3 : nº del Asiento inicial - 1

D5 : nº Saltos desde el primer hasta el segundo asiento de la misma fila

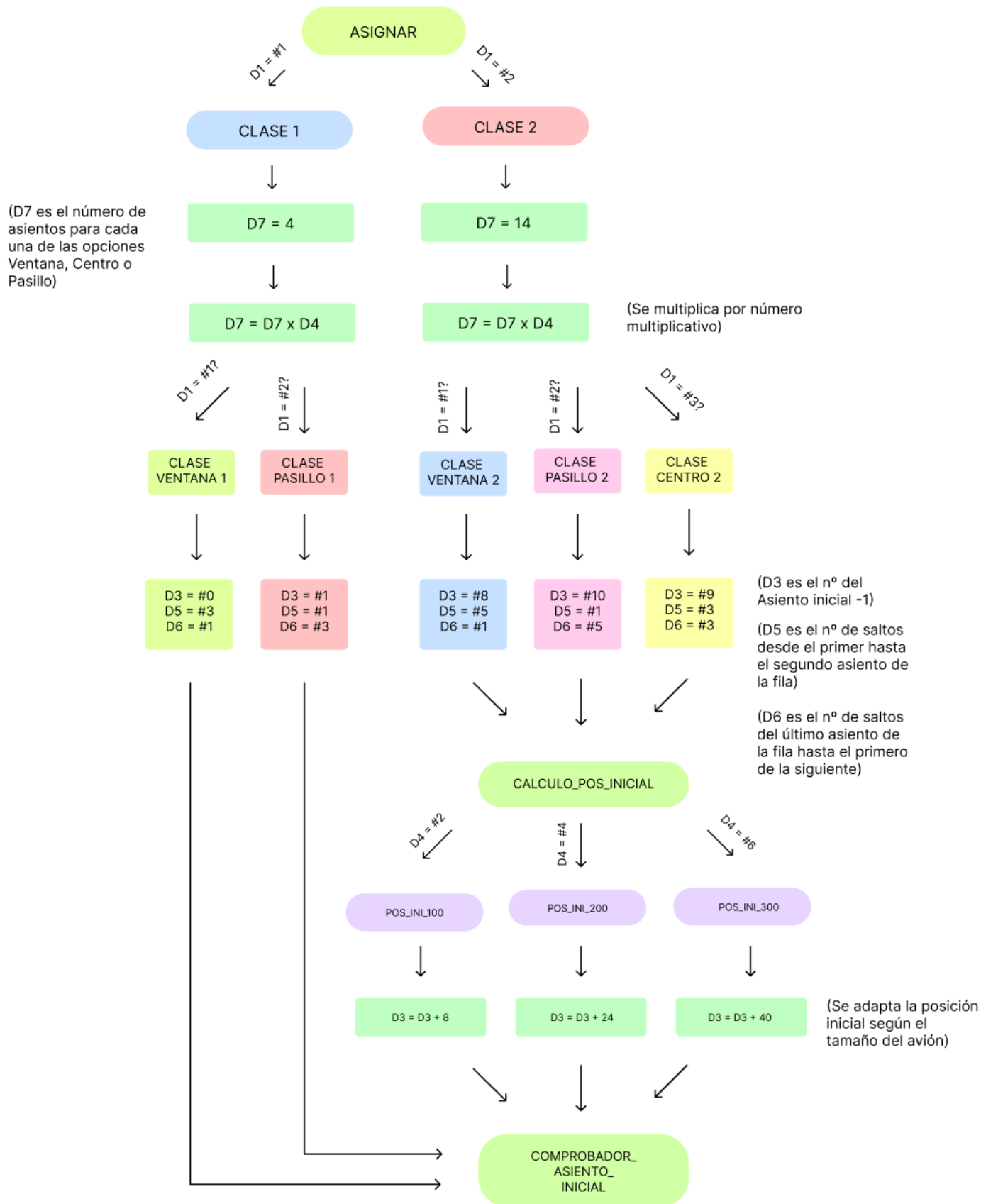
D6 : nº Saltos desde el último asiento de una fila, hasta el primer asiento de la siguiente fila

Respecto a la ampliación de asientos, si son 100,200 o 300, las posiciones Iniciales de la clase Turista no funcionarán ya que están hechas para el tamaño de 50 pasajeros, por lo que habrá que adaptarlo según el tamaño.

Si Multiplicador = 2 <—> 100 Pasajeros —> Posición inicial = D3 + 8

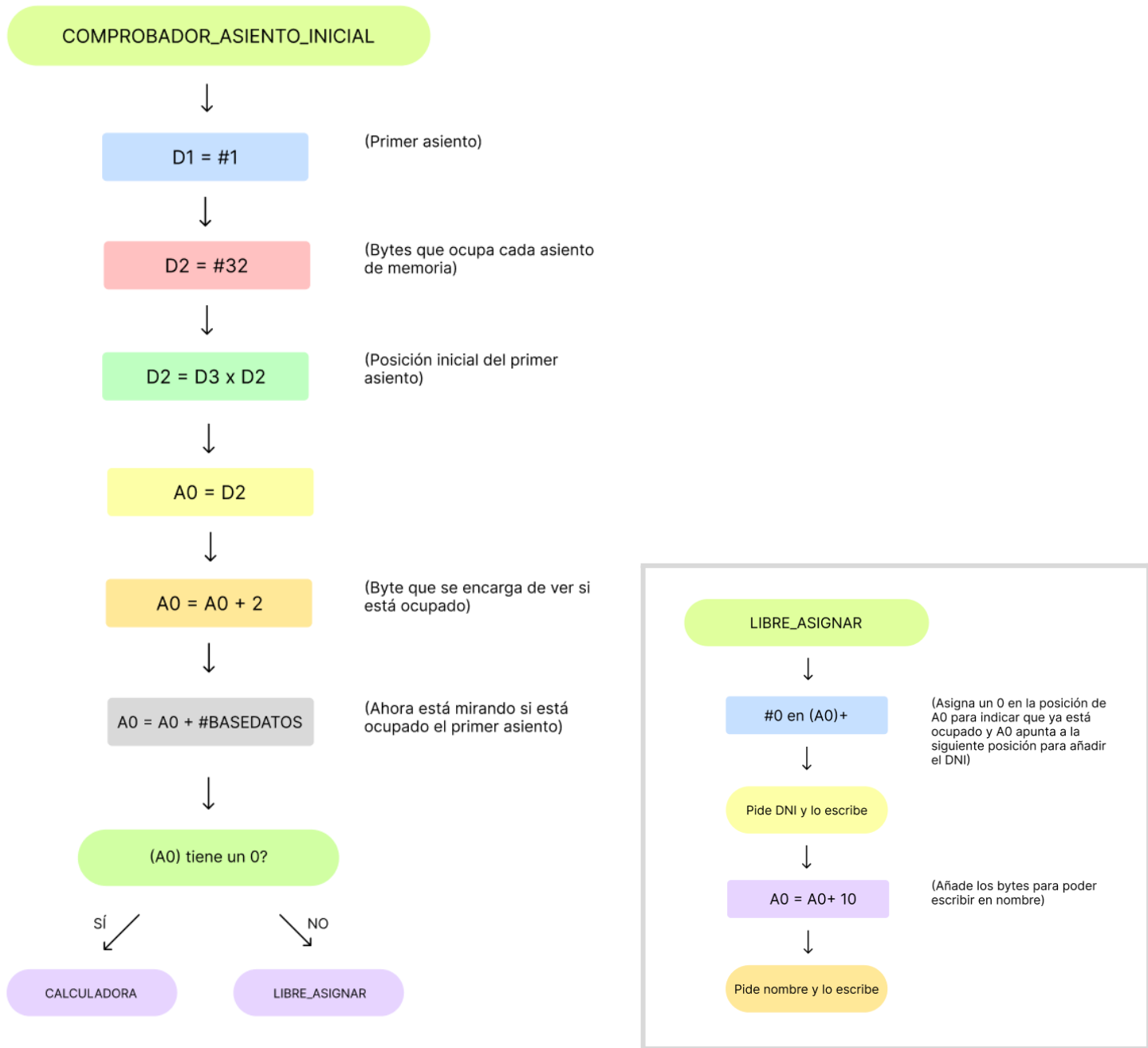
Si Multiplicador = 4 <—> 200 Pasajeros —> Posición inicial = D3 + 24

Si Multiplicador = 6 <—> 300 Pasajeros —> Posición inicial = D3 + 40



### COMPROBADOR\_ASIENTO\_INICIAL

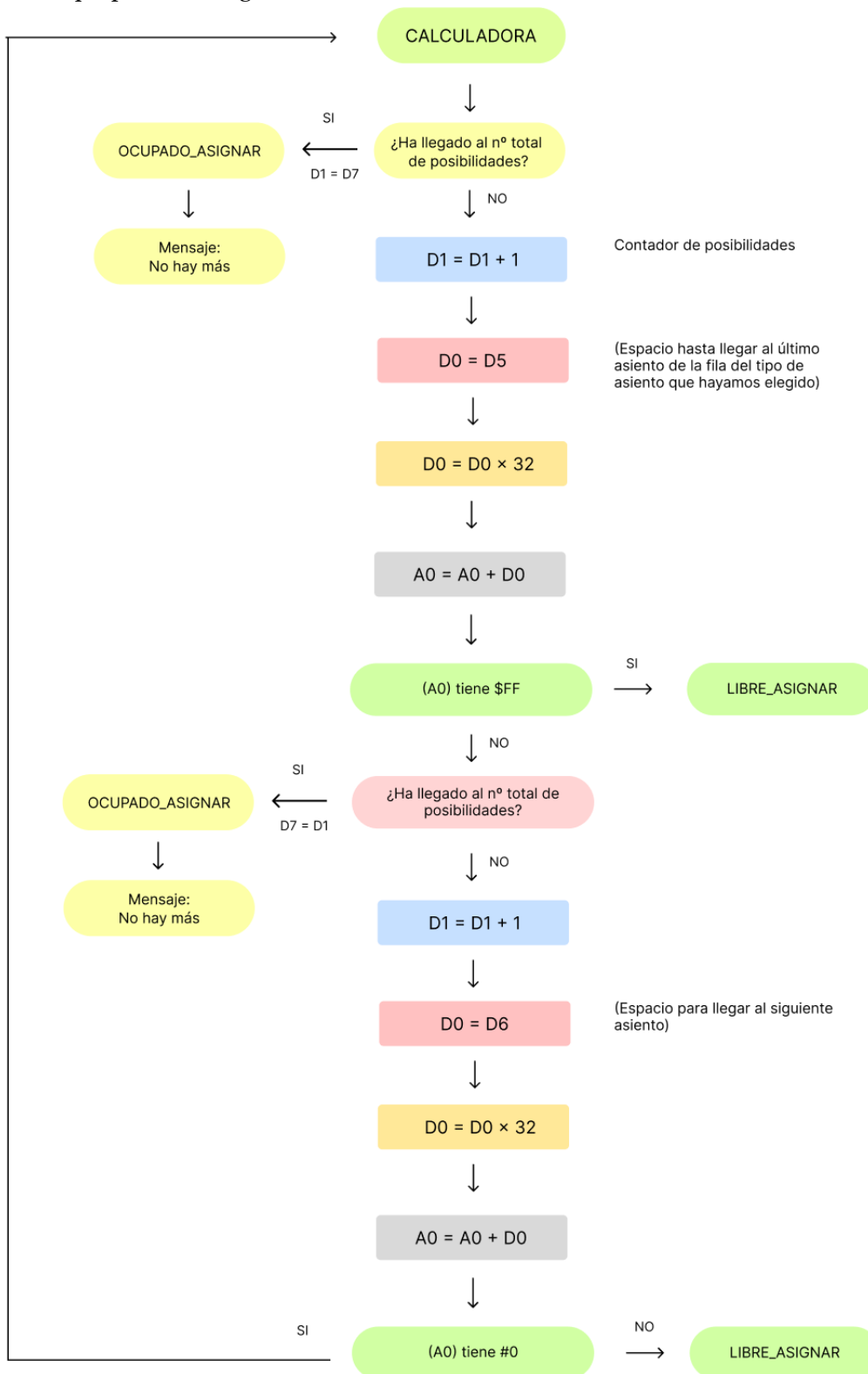
Este bloque determina si el primer asiento para cada opción está libre y si lo está pasa a LIBRE\_ASIGNAR, pero si no lo está (Para que esté libre el 3º Byte del código de cada asiento deberá contener \$FF y si no lo está 00), ese asiento será el punto de partida de nuestro algoritmo y saltaremos a la CALCULADORA.





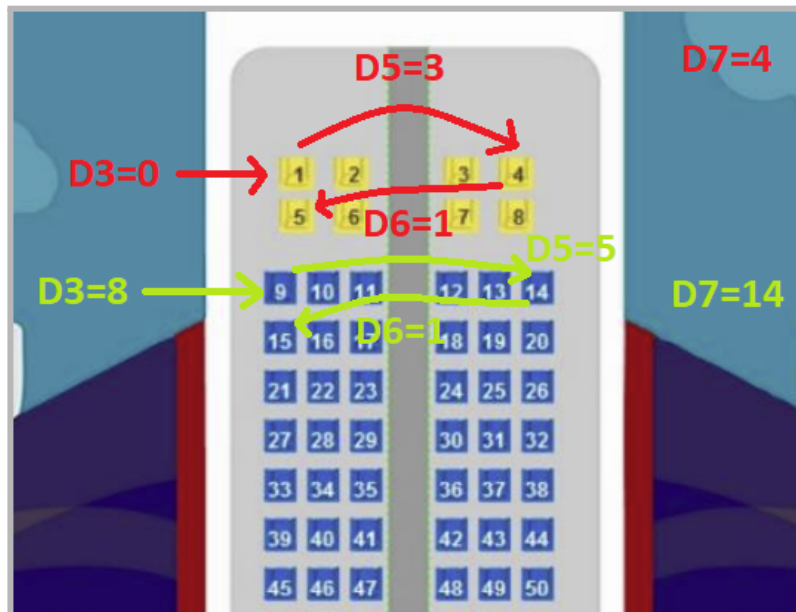
## CALCULADORA

Este bloque es muy importante, ya que nos permitirá hacer el cálculo de todos los asientos con las distintas posibilidades que podemos elegir.



Después de la negativa de que el primer asiento esté libre, mediante los saltos que contienen los registros D5 y D6 iremos saltando a partir de la posición inicial para la opción que hayamos elegido.

Le sumaremos al registro de direcciones  $D5*32$  (Bytes de cada asiento en memoria), si no está libre querrá decir que la primera fila está ocupada, sumaria al registro de direcciones  $D6*32$  y el registro de direcciones apuntaría a la dirección de memoria del primer asiento de la fila de abajo.



Si no hay ninguno libre volverá hacer el ciclo una y otra vez comprobando, hasta que el número de veces que se haya comprobado sea igual a D7 ya que querrá decir que se han acabado todos los posibles asientos y saltará a OCUPADO\_ASIGNAR, devolviendo el mensaje pertinente y volviendo al MENU\_PRINCIPAL .

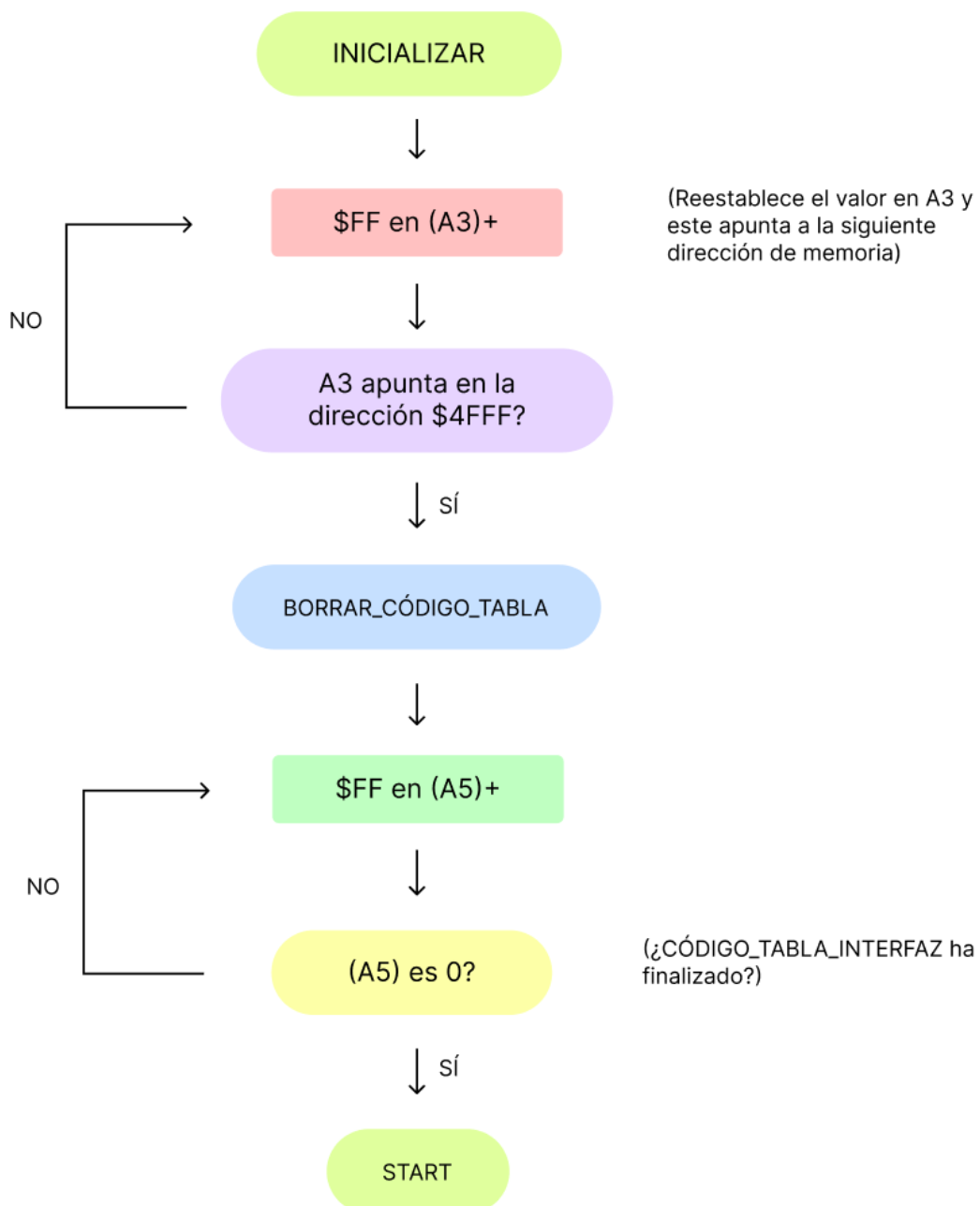
Si encuentra un asiento libre, el programa pasará a LIBRE\_ASIGNAR donde marcará en memoria el Byte 2 del código de ese asiento con un \$00, además le pedirá al usuario sus datos personales, se escribirán a partir del Byte 3 hasta el Byte 12 el DNI y el Nombre a partir del Byte 14 hasta el 32 como máximo.

## INICIALIZAR

El objetivo de este bloque es borrar toda la memoria ocupada por los pasajeros. En el registro A3 inicialmente se encuentra la posición de memoria de #BASEDATOS. Para ello se sustituirá cada byte de la memoria por \$FF hasta que A3 llegue a la posición de memoria \$4FFF ( porque en \$5000 guardan todos los mensajes), entonces saltará a la etiqueta BORRAR\_CODIGO\_TABLA.

## BORRAR\_CODIGO\_TABLA

En este apartado se borra el CODIGO\_TABLA\_INTERFAZ empleado para generar la tabla ASCII y la interfaz gráfica. Inicialmente A5 indica la posición de memoria de CODIGO\_TABLA\_INTERFAZ. Para ello se sustituirá cada byte de memoria por \$FF hasta que detecte #0 dentro de la posición de (A5) lo que indicará que el código ASCII de la tabla ha terminado. Entonces saltará a START para que el usuario pueda volver a elegir lo que hacer.



# TABLA E INTERFAZ GRÁFICA

Esta opción del menú, nos permitirá generar una Tabla de caracteres ASCII representativa sobre el estado actual del avión.

FILA	ASIENTO A	ASIENTO B	ASIENTO C	ASIENTO D	ASIENTO E	ASIENTO F
1		#	#	#	#	
2		#	#	#	#	
3	X	#	#	#	#	#
4	#	#	#	#	#	#
5	#	#	#	#	#	#
6	#	#	#	#	#	#
7	#	#	#	#	#	#
8	#	#	#	#	#	#
9	#	#	#	#	#	#

Para crearla, lo primero que haremos será generar un código (CODIGO\_TABLA\_INTERFAZ) que describa la situación del avión y nos ayude a crearla, ya que programarla a la vez que se comprueban los estados de los asientos sería complicado.

00007FF0	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00007FF0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	-----
00008000:	30	23	23	23	23	30	30	23	23	23	23	30	58	23	23	23	0####00####0X###
00008010:	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00008020:	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	#####
00008030:	23	23	23	23	23	23	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	#####
00008040:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	-----

## GENERADOR\_CODIGO

Lo definiremos como una subrutina, ya que lo usaremos para generar el código que usará la tabla y a parte el que usará la interfaz que explicaremos posteriormente.

D1=0 (contador de asientos)

D3=0 (contador de asientos+ casillas en blanco)

A1 apuntará al espacio reservado para CODIGO\_TABLA\_INTERFAZ en la dirección \$8000

D7 = MULTIPLICADOR(A4) x 54 (Huecos totales, si la dimensión del avión es 50, la tabla tendrá 4 huecos libres a los lados debido a la separación entre la Primera Clase y la Clase Turista)

En base a estos datos iniciales, comenzaremos por un bucle que consulte el 3º Byte de todos los asientos para comprobar si están libres (#) o ocupados(X), además las casillas en blanco tanto de la tabla como la interfaz, estarán representadas por '0' en el código.

## GENERADOR\_HUECOS\_CODIGO

Lo primero que haremos será crear un filtro a base de comparar el contador D3 con las posiciones en las que queremos que aparezcan esos 'o', para poder crear esos huecos, además se usarán mas o menos CMP según la dimensión del avión (MULTIPLICADOR).

Si D3 coincide con alguno, se añadirá un o en la secuencia del CODIGO\_TABLA\_INTERFAZ y

$D3 = D3 + 1$  (Para pasar al siguiente símbolo que representará la siguiente casilla de la tabla o interfaz).

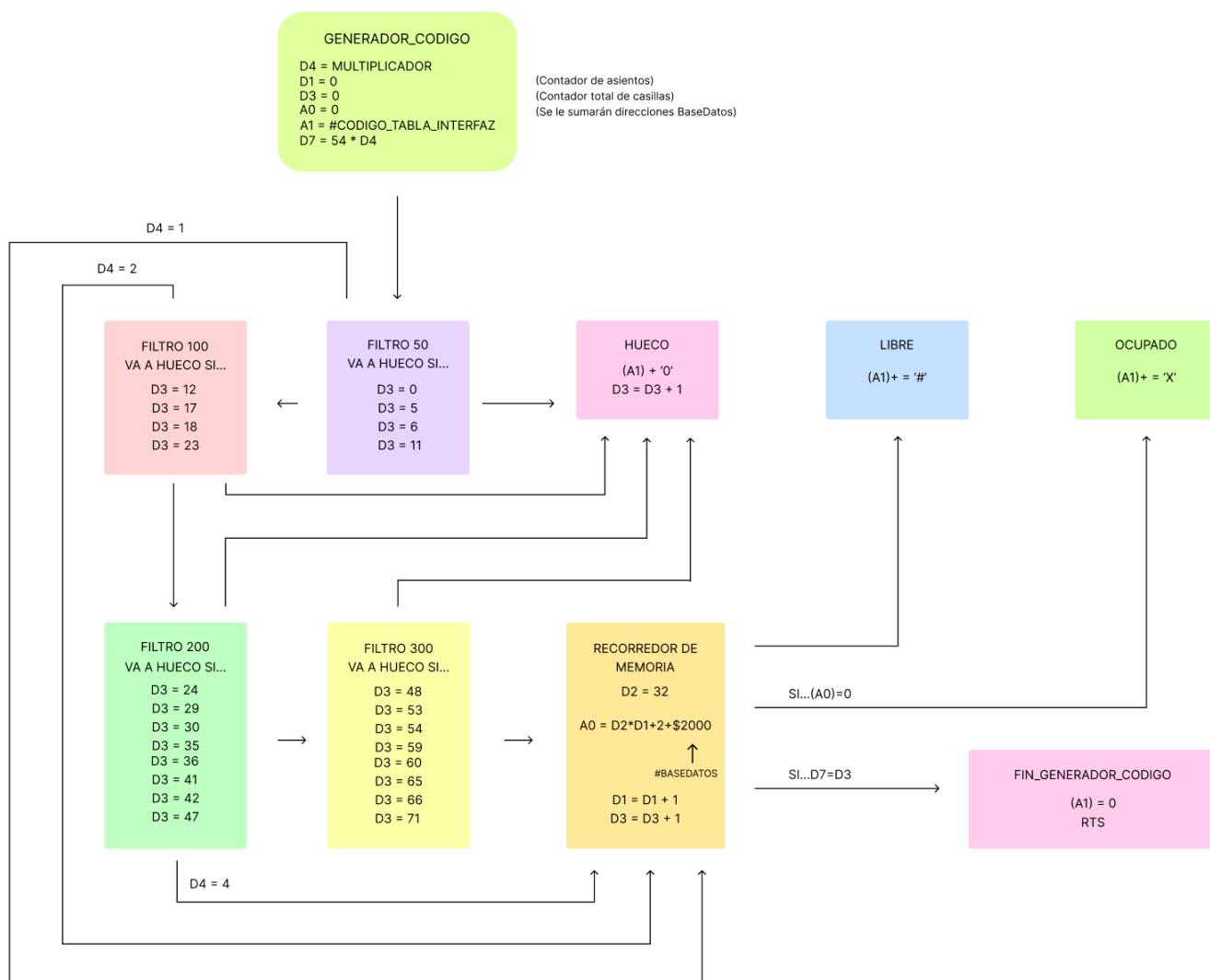
Si no coincide con ningún CMP pasará RECORREDOR\_MEMORIA\_GEN\_COD

## RECORREDOR\_MEMORIA\_GEN\_COD

Este bloque consultará BASEDATOS y verá qué asientos están libres u ocupados y asignará según estén una # o X a CODIGO\_TABLA\_INTERFAZ y volverá a GENERADOR\_HUECOS\_CODIGO

Haremos la típica operación realizada en CONSULTAR y ASIGNAR, para poder recorrer cada código de asiento fijándonos en el 3<sup>o</sup> Byte para ver el estado del asiento,  $A0 = (D1 * 32 + \$2000)$  comprobaremos y según lo que haya pondremos X o # en A1(CODIGO\_TABLA\_INTERFAZ) con postincremento y fijamos  $D1 = D1 + 1$  y  $D3 = D3 + 1$  para el siguiente ciclo.

Hará ciclos , mientras que D7 y D3 no sean iguales, cuando lo sean, saltará a la etiqueta FIN\_GENERADOR\_CODIGO y colocará un o al final de CODIGO\_TABLA\_INTERFAZ.



## GENERACIÓN DE LA TABLA

Ahora que tenemos nuestra secuencia de caracteres lista en CODIGO\_TABLA\_INTERFAZ, podemos crear la tabla más fácilmente.

EJEMPLO 50 0####0####0#####...  
EJEMPLO 100 0####0####0####0####0#####...  
EJEMPLO 200 0####0####0####0####0####0####0####0#####...

Guardaremos tras todos los mensajes usados durante el programa, los caracteres ASCII y espacios, para poder generar la tabla.

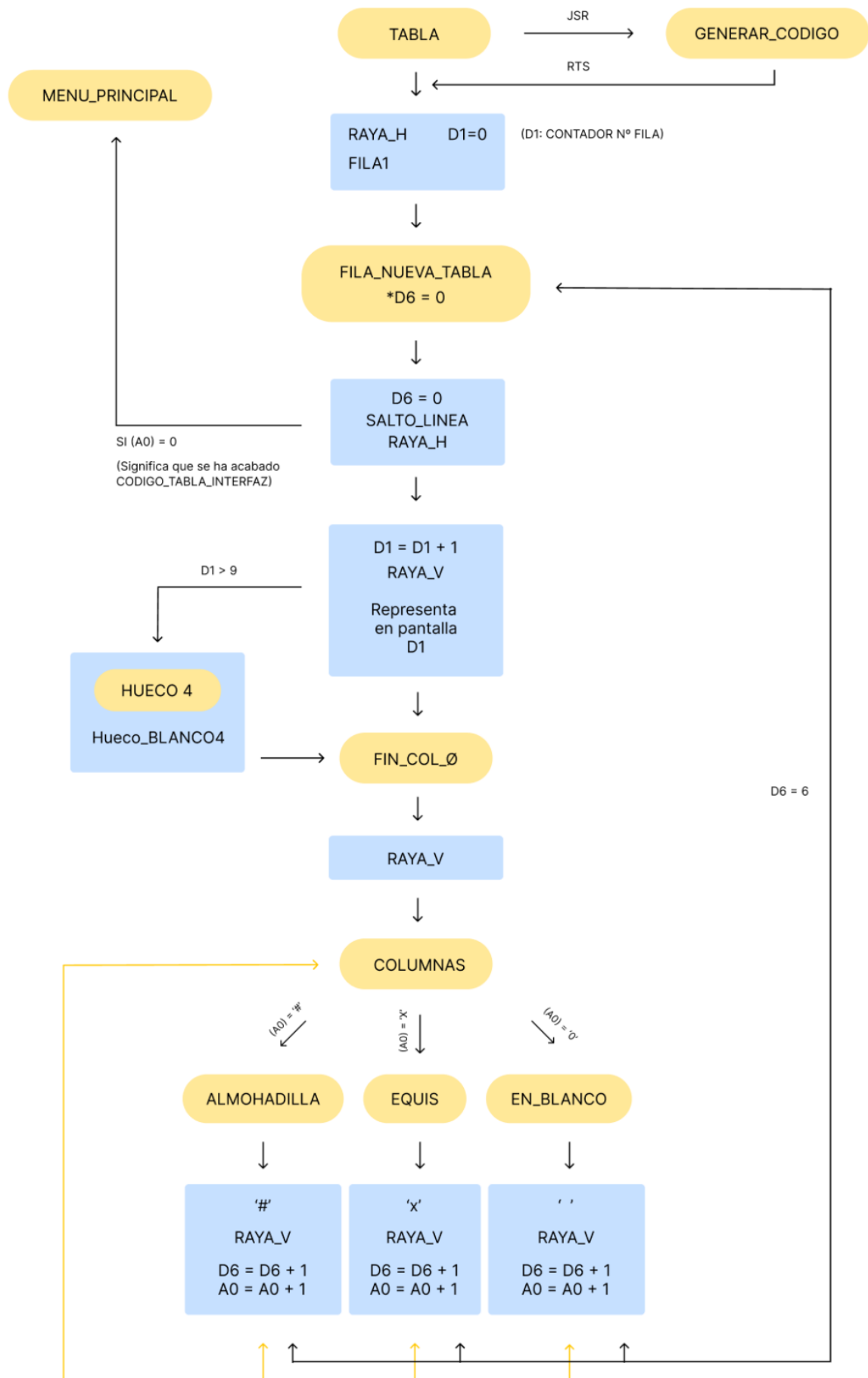
```
SALTO_LINEA DC.B ' ',13,10,0
ESPACIO_BLANCO DC.B ' ',0
RAYA_V DC.B '|',0
RAYA_H DC.B '-----',13,10,0
FILAI DC.B '| FILA | ASIENTO A | ASIENTO B | ASIENTO C | ASIENTO D | ASIENTO E | ASIENTO F |',0
SIGNO1 DC.B ' #' ,0
SIGNO2 DC.B ' X ' ,0
HUECO_BLANCO11 DC.B ' ' ,0
HUECO_BLANCOS DC.B ' ' ,0
HUECO_BLANCO4 DC.B ' ' ,0
```

D1=0 : Contador nº de Fila

Ao : Apunta a CODIGO\_TABLA\_INTERFAZ

Empezaremos generando la primera fila que corresponde a los campos de nuestra Tabla, a partir de ahí entramos en FILA\_NUEVA (CONTADOR D6=0) generamos el número de de esa fila (Guardado en D1), rellenando de los caracteres para que quede adecuadamente la casilla.

Después pasará a comprobar el CODIGO\_TABLA\_INTERFAZ y rellenará la segunda casilla con espacios si es 0 , # si está libre, o X si está ocupado, y avanzaremos el contador de columnas D6=D6+1 volverá a repetir el proceso hasta llegar a que D6=6, entonces deberemos crear otra fila (FILA\_NUEVA) y así hasta que lea en el código un 0, entonces parará y volveremos al MENU\_PRINCIPAL



## GENERACIÓN DE LA INTERFAZ

Guardaremos tras todos los caracteres ASCII y espacios de la tabla, todo lo necesario para poder generar la interfaz.

Tiene prácticamente el mismo funcionamiento que la generación de la Tabla, pero con la diferencia que ahora no usaremos símbolos ASCII para formarla, si no espacios con rellenos de color y los numeros de los asientos rellenos de color, verde si está libre (# en CODIGO\_TABLA), rojo si está ocupado(X) y blanco para rellenar el hueco(o).

D6 será otra vez nuestro contador de columnas, pero esta vez saltará cuando D6 valga 7 ya que el hueco del pasillo lo detectamos como una columna y se generará cuando D6=3, D7 guardará el número que le pondremos al asiento en la interfaz.

La complicación viene cuando queremos representar D7 en los cuadrados de colores, como D1 se usa para almacenar el código de los colores que se usa para pintar los cuadrados, decirle que represente directamente D7=D1 con un trap modo 3, haría que se perdiese esos colores, entonces lo mejor es no usar el trap modo 3, si no sacar el numero como si fuese una cadena. Para ello habrá que sumarle \$30 a ese número para que lo saque en ASCII.

Otro problema es que en ASCII obviamente solo va a contar hasta el 9, entonces hay que hacer 3 operaciones, similares a las realizadas en prácticas , para representar números en 7 segmentos, SWAP y DIVU.

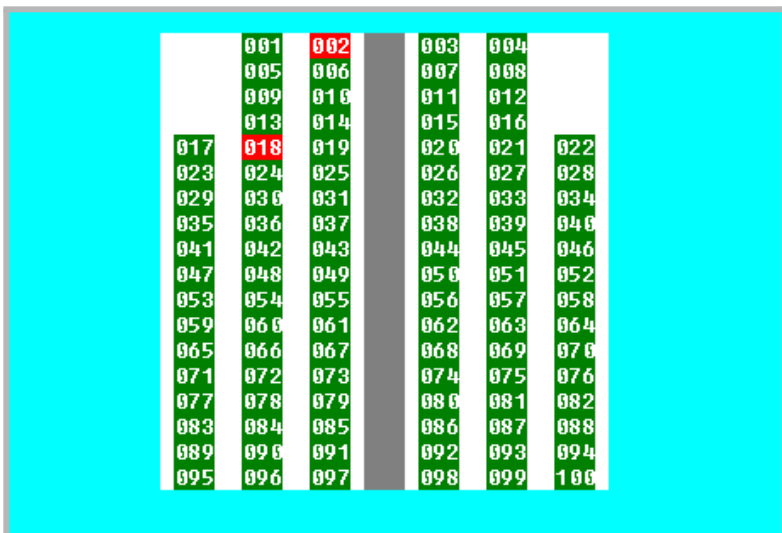
Crearemos una Subrutina que haga esta operación

Usaremos otros registros para hacer las operaciones y no alterar D7.

El primer DIVU será entre 100 y extraerá las decenas (resto) y centenas (cociente), mostraremos por pantalla el cociente (Parte baja del registro), usaremos SWAP y guardamos el resto (ahora la parte baja) y lo volvemos a dividir, pero entre 10 y extraemos y mostramos del mismo modo unidades y decenas.

D7=D7+1 y avanzamos también el registro de direcciones y una posición para leer el siguiente carácter de CODIGO\_TABLA\_INTERFAZ.

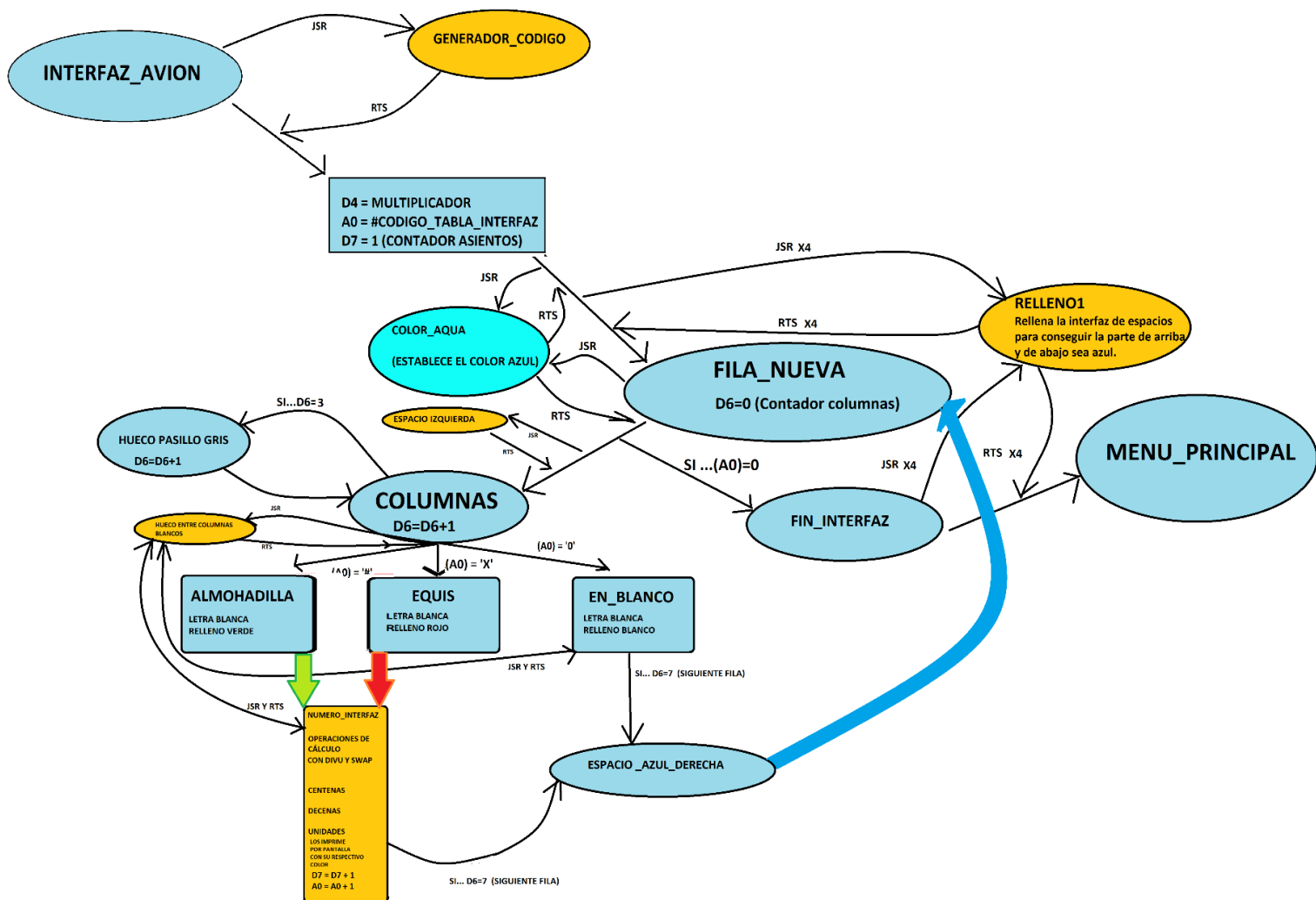
Iremos a NUEVA\_FILA o volveremos a columnas según el valor de D6. La interfaz acabará cuando se acabe nuestro código Tabla detectando un 0, como en la tabla y volverá al menú principal



	001	002		003	004	
	005	006		007	008	
	009	010		011	012	
	013	014		015	016	
017	018	019		020	021	022
023	024	025		026	027	028
029	030	031		032	033	034
035	036	037		038	039	040
041	042	043		044	045	046
047	048	049		050	051	052
053	054	055		056	057	058
059	060	061		062	063	064
065	066	067		068	069	070
071	072	073		074	075	076
077	078	079		080	081	082
083	084	085		086	087	088
089	090	091		092	093	094
095	096	097		098	099	100

En el proceso realizaremos varios rellenos de color Azul a los lados, arriba y abajo, pero no los queremos comentar porque no los consideramos relevantes en la explicación, es una cuestión de estética.





# MANUAL DEL USUARIO

## DIMENSIÓN AVIÓN

```
Usted se encuentra en el menu principal de la aeronave ALECAR
Selecciona el número de pasajeros de su avión
Pulse " 1 " para elegir dimensión de 50
Pulse " 2 " para elegir dimensión de 100
Pulse " 3 " para elegir dimensión de 200
Pulse " 4 " para elegir dimensión de 300
_
```

Seleccionamos la dimensión del avión que queramos presionando 1,2,3 o 4 , si no se pulsa ninguno de estos, se inicializa el programa.

## MENÚ PRINCIPAL

```
Seleccione la operación que desee realizar
Pulse " 1 " para Consultar los asientos
Pulse " 2 " para asignar asientos
Pulse " 3 " para inicializar la base de datos
Pulse " 4 " para mostrar la disponibilidad de todos los asientos
Pulse " 5 " para mostrar la interfaz gráfica del avión
_
```

Seleccionamos la operación que queramos realizar presionando 1,2,3 o 4 , si no se pulsa ninguno de estos, vuelve al mismo menú.

## CONSULTAR

```
Elija el número de asiento que desea consultar >0
```

```
El asiento no Existe
```

Si consultamos un asiento no existente para nuestra dimensión de avión o para ninguna, nos dirá que no existe y nos devolverá al Menú Principal.

```
Elija el número de asiento que desea consultar >1
```

```
El asiento esta ocupado por el pasajero, 12345678N Carlos
```

Si consultamos un asiento que está ocupado nos devolverá el mensaje, los datos personales del pasajero y nos devolverá al Menú Principal.

```
Elija el número de asiento que desea consultar >49
```

```
El asiento esta disponible
```

Si consultamos un asiento que está libre nos devolverá el mensaje y nos devolverá al Menú Principal.

## ASIGNAR

```
Seleccione la clase del asiento que desea ocupar  
Pulse " 1 " para Primera Clase  
Pulse " 2 " para Clase Turista
```

Al asignar asientos , lo primero que nos pedirá es la clase que queremos elegir.

Según la opción podremos elegir entre ventana y pasillo o ventana centro y pasillo.

```
Seleccione la ubicación del asiento que desea ocupar  
Pulse " 1 " para un asiento al lado de la ventana  
Pulse " 2 " para un asiento al lado del pasillo
```

```
Seleccione la ubicación del asiento que desea ocupar  
Pulse " 1 " para un asiento al lado de la ventana  
Pulse " 2 " para un asiento al lado del pasillo  
Pulse " 3 " para un asiento en el centro
```

Tras seleccionar la ubicación de nuestro asiento, nos pedirá nuestro DNI y nombre

```
Introduzca los datos personales del pasajero  
Escriba el DNI >  
12345678A  
  
Escriba el nombre >  
Alejandro
```

Tras esto se nos habrá asignado el asiento y nos volverá a enviar al Menú Principal

## INICIALIZAR

Esta opción permite inicializar el avión, borrar el espacio de memoria que se ha generado para lectura y escritura, y permitir volver ejecutar el programa desde la elección de la dimensión del avión.

Nota: Esta opción estaría deshabilitada para un usuario si este proyecto se destinase a un caso real y solo estaría habilitada para un Supervisor.

TABLA

FILA	ASIENTO A	ASIENTO B	ASIENTO C	ASIENTO D	ASIENTO E	ASIENTO F
1		#	#	#	#	
2		#	#	#	#	
3	#	#	X	#	#	#
4	#	#	#	#	#	#
5	#	#	#	#	#	#
6	#	#	#	#	#	#
7	#	#	#	#	#	#
8	#	#	#	#	#	#
9	#	#	#	#	#	#

Nos generará una tabla con símbolos ASCII donde se podrá consultar rápidamente el estado de los asientos, las primeras dos filas representan la 1ºClase, de la tercera hasta la última representa la clase turista.

Dependiendo de la dimensión del avión elegida esta tabla podrá ser más grande y a su vez el tamaño de ambas clases aumentará, pero la 1ºClase siempre se caracterizará por ir desde ASIENTO B hasta el E y por tener el A y F libres.

INTERFAZ AVIÓN

	001	002		003	004	
	005	006		007	008	
009	010	011		012	013	014
015	016	017		018	019	020
021	022	023		024	025	026
027	028	029		030	031	032
033	034	035		036	037	038
039	040	041		042	043	044
045	046	047		048	049	050

Esta opción tiene la misma funcionalidad que la tabla, solo que se representa el estado de los asientos de una forma más visual, marcando de color verde los asientos que están libres y los que están ocupados en rojo, con su respectivo número, además de separarlos con el pasillo.

Pasa lo mismo otra vez que dependiendo la dimensión del avión que hayamos elegido, obtendremos un dibujo distinto

	001	002		003	004	
	005	006		007	008	
	009	010		011	012	
	013	014		015	016	
	017	018		019	020	
	021	022		023	024	
	025	026		027	028	
	029	030		031	032	
	033	034		035	036	
	037	038		039	040	
	041	042		043	044	
	045	046		047	048	
049	050	051		052	053	054
055	056	057		058	059	060
061	062	063		064	065	066
067	068	069		070	071	072
073	074	075		076	077	078
079	080	081		082	083	084
085	086	087		088	089	090
091	092	093		094	095	096
097	098	099		100	101	102
103	104	105		106	107	108
109	110	111		112	113	114
115	116	117		118	119	120
121	122	123		124	125	126
127	128	129		130	131	132
133	134	135		136	137	138
139	140	141		142	143	144
145	146	147		148	149	150
151	152	153		154	155	156
157	158	159		160	161	162
163	164	165		166	167	168
169	170	171		172	173	174
175	176	177		178	179	180
181	182	183		184	185	186
187	188	189		190	191	192
193	194	195		196	197	198
199	200	201		202	203	204
205	206	207		208	209	210
211	212	213		214	215	216
217	218	219		220	221	222
223	224	225		226	227	228
229	230	231		232	233	234
235	236	237		238	239	240
241	242	243		244	245	246
247	248	249		250	251	252
253	254	255		256	257	258
259	260	261		262	263	264
265	266	267		268	269	270
271	272	273		274	275	276
277	278	279		280	281	282
283	284	285		286	287	288
289	290	291		292	293	294
295	296	297		298	299	300