		- 2							
								7	5
								12	
	-	de		de				4.1	
						0.5			
						0.2		F	
	0	de		de		0.1		-	
	0	g.		de		G	T	1.3	
	0	d.		de				1.2 1.3	
-	0	ē		de				-	
	0	rp		<u>6</u>				100	Tiempo
	turn	රි	(5	-	J.	٥	-	,-

Figura 3.3. Diagrama de estados para el algoritmo de Hyman.

3.3.5. Algoritmo de Eisenberg-McGuire

Como ya comentábamos anteriormente, el algoritmo de Eisenberg-McGuire [Eisenber and McGuire, 1972] está inspirado en el algoritmo original de Dijkstra. Este algoritmo utiliza las siguientes estructuras de datos:

- Los elementos del array indicador son del tipo enumerado, con tres posibles • indicador: array[0..n-1] of (restoproceso,quiereentrar,enSC) valores cuyo significado es el siguiente:
- restoproceso: indica que el proceso está ocioso con respecto a la sección
 - crítica, es decir, no está ni desea entrar en la sección crítica.
- quiereentrar: refleja la situación de que el proceso desea entrar en su sec
 - ensc: significa que el proceso se encuentra ejecutando su sección crítica.
- indice: 0..n-1, que indica qué proceso puede entrar en la sección crítica.

Todos los elementos del array tendrán inicialmente el valor restoproceso y el valor inicial de la variable indice es indistinto, es decir, puede ser cualquier entero

```
if indicador[j]≠restoproceso
                                    indicador[i]: =quiereentrar;
                                                  (1) j: =indice;
                                                               while(j≠i)
process P<sub>i</sub>
                                                                               begin
                          repeat
            repeat
```

© ITES-Paraninfo

```
(indicador[indice]=restoproceso))
                                                                                                                                      while ((j<n) and ((j=i) or (indicador[j] #enSC)))
                                                                                                                                                                                                                                                                                                                                            while (indicador[j]=restoproceso)
                                                                                                                                                                                                 (a) until ((j>n) and ((indice=i) or
                                                                                                                                                                                                                                                                                                                                                                                                                                indicador[i]: =restoproceso;
                          else j: =(j+1)mod n
(2) then j: =indice
                                                                                     indicador[i]: menSC;
                                                                                                                                                                                                                                                                                                                 | = (indice+1) mod n;
                                                                                                                                                                                                                                                                                   Section Critica;;
                                                                                                                                                                                                                                                                                                                                                                           j: = (j+1)mod n;
                                                                                                                                                                                                                                                                                                                                                                                                       (a) indice: =j;
                                                                                                                                                                                                                                                            indice: =i;
                                                                                                                                                                           1 = 1+1;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                Resto
```

ominuación, se describe qué es lo que hacen cada una de las partes del

- Comprueba circularmente que entre el que tenía permiso para entrar y él, los procesos se encuentran en el resto del proceso.
- Il alguno no está en el resto del proceso (Resto_i), vuelve a comprobar el el principio.
 - Si hay algún proceso en la sección crítica, espera.
- Hasta que ninguno está en la sección crítica e i tiene el turno o el que lo está en Resto_i.
- Da turno al primero que no está en el resto del proceso, es decir, al pri que quiere entrar. Si ninguno quiere entrar, se queda con el turno, ya que está en el resto del proceso.

Than probar que el algoritmo es correcto es necesario ver:

- Que satisface el requerimiento de la exclusión mutua: para ello se puede o var que cada proceso P_i , entra en su sección crítica solamente si indicado es distinto de enSC, para todo j≠i. Debido a que solamente el proceso P. de fijar indicador[i]=enSC, y que el proceso Pi inspecciona indicado wolamente cuando indicador[i]=enSC, el resultado es correcto.
- Que se satisface el requerimiento de progreso en la ejecución: se puede var que el valor de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando un presenta de la variable indice puede ser modificado cuando de la variable de so entra en su sección crítica y cuando deja la sección crítica. Así, si no ningún proceso en su sección crítica o dejándola, el valor de indice pen ce constante. El primer proceso que desee entrar en el orden cíclico (1, ..., n-1, 0, ..., i-1) entrará en la sección crítica.