**CLUSTER PROCESADOR SALA DE ESPERA PROCESOS**

Privat: Privat: Privat: Privat:

?<PROCESADOR> ?<PROCESOS> string id\_prioridades int cant\_memoria

String id\_procesador int num\_prioridades Int tiempo

?<PROCESOS> Int id\_procesos

Públic: Públic: Públic: Públic:

Configurar\_cluster() existeix(id\_procesos) iniciar\_sala(num\_prioridades) en\_ejecuccion(id\_procesador)

Modificar\_cluster(id\_procesador,clúster) alta\_prioridad (id\_prioridad) existeix (id\_procesos)

Alta\_proceso\_procesador (id\_procesador, id\_procesos,tiempo,cant\_memoria) Baja\_prioridad (id\_prioridad)

existeix(id\_procesador) Alta\_procesos\_espera (id\_procesos,tiempo,cant\_memoria,id\_prioridad)

baja\_procesos\_procesador(id\_procesador, id\_procesos) Existeix (id\_prioridad)

enviar\_procesos\_cluster (int n, Procesos) imprimir\_prioridad (id\_prioridad)

avanzar\_tiempo(int t) imprimir\_area\_espera()

imprimir\_procesador (id\_procesador)

imprimir\_procesadores\_cluster()

imprimir\_estructura\_cluster() optimitzar\_memoria();

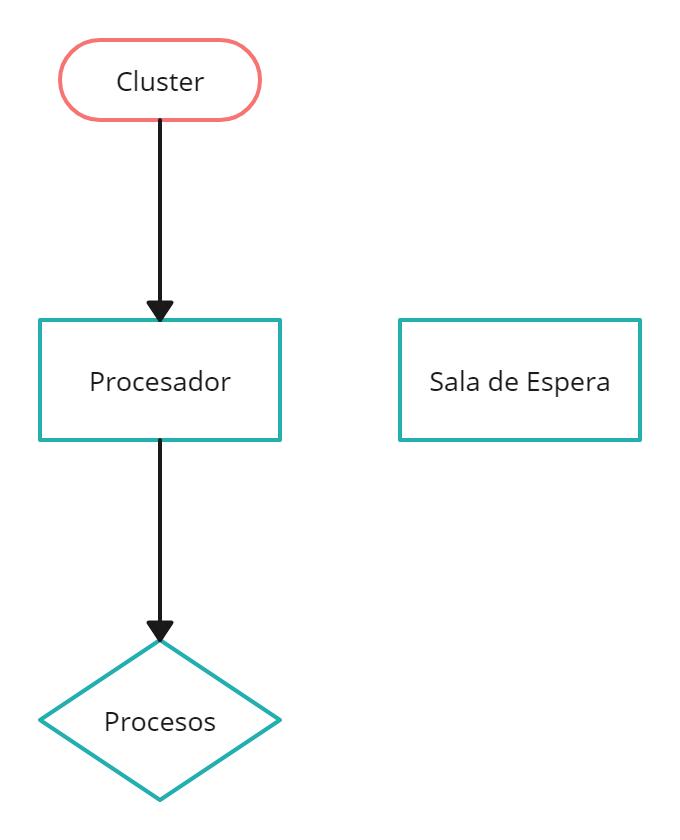
Compactar\_memoria\_procesador(id\_procesador)

compactar\_memoria\_cluster()

**MAIN**

Clúster.configurar\_cluster();

Sala.iniciar\_sala(ident\_prioridades);



While (comando != ‘fin’) {

1. Cluster.Configurar\_cluster();
2. Clúster.modificar\_cluster(ident\_procesador,clúster);
3. Sala.alta\_prioridad (id\_prioridad);
4. Sala.baja\_prioridad (id\_prioridad);
5. Sala.alta\_proceso\_espera ((id\_procesador,id\_prioridad);
6. Cluster.alta\_proceso\_procesador ((id\_procesador, id\_procesos,tiempo,cant\_memoria);
7. Cluster.baja\_procesos\_procesador(id\_procesador, id\_procesos);
8. \*Cluster.enviar\_procesos\_cluster(int n, Procesos);
9. \*Procesador.avanzar\_tiempo (int t);
10. Sala.imprimir\_prioridad (id\_prioridad);
11. Sala.imprimir\_area\_espera();
12. \*Cluster. imprimir\_procesador (id\_procesador);
13. Cluster. imprimir\_procesadores\_cluster();
14. Cluster. imprimir\_estructura\_cluster();
15. \*Cluster.compactar\_memoria\_procesador (id\_procesador)
16. Cluster. compactar\_memoria\_cluster();
17. Fin;