

1 | B  
2 | B  
3 | B

(A)

① PMA.

Chon espesa (int, int);

Chon Sip (int),  
Chon proxcliente [3] (int, int);

Chon tu entrada [P] (int);

Process PERSONA [id: 1 .. P] {  
int DNI, INT telefono, TXT entrada;

DNI = obtenerDNI();

Send espesa (ID, DNI);

Receive tu entrada [ID3] (entrada);

Process Empleado [ID: 1 .. 3] {  
Bool seguir; int C, DNI; TXT entrada;

seguir = true;

while (seguir) {

Send Sip (ID);

Receive proxCliente [ID] (C, DNI);

if (C > 0) {

entrada = obtenerEntrada (DNI);

Send tu Entrada [C] (entrada);

else {

seguir = false;

Process Admin {  
int cont; case, Persona, DNI; i.

For i: 1 .. P {

receive Sip (case);

receive espesa (Persona, DNI);

Send proxCliente [case] (Persona, DNI);

For i: 1 .. 3 {

receive Sip (case);

Send proxCliente [case] (-1, 0);

1)

(2) PNs.

Process Entrenador {

Int i;

For i : 1..P {

Deportista [i]? QuienUsa( );

Deportista [i]! PodesUsar( );

Deportista [i]? terminaUsar( );

}

3

Comienzo

Iniciar

Process DEPORTISTA [id: 1..P] {

Entrenador! quienUsa( );

Entrenador? PodesUsar( );

USAR! laPista( );

Entrenador! terminaUsar( );

3

2)

③ ADA.

Program vuelo IS;

TASK TYPE PASAJERO;

TASK EMPLEADO;

BEGIN

Entry consulta\_inmediata (cod: IN INT; Puenta: OUT INT, Itono: OUT DOUBLE);  
Entry consulta\_reducida (cod: IN INT, Puenta: OUT INT, Itono: OUT DOUBLE);  
Entry consulta\_otros (cod: IN INT, Puenta: OUT INT, Itono: OUT DOUBLE);

END;

VECP = array (1..P) OF PASAJERO;

TASK BODY EMPLEADO IS,

INT Poblado, INT P;

BEGIN

Pasajero := 0;

P := P; → cont Poblado;

WHILE (Pasajero < P) Loop FOR 1 → R

SELECT

ACCEPT CONSULTA\_INMEDIATA (cod: IN INT, Puenta: OUT INT, Itono: OUT DOUBLE) DO  
Puenta := obtenerPuenta (cod);  
Itono := obtenerItono (cod); Poblado++;  
END CONSULTA\_INMEDIATA;

OR WHEN (CONSULTA\_REDUCIDA'count == 0) => ACCEPT consulta\_reducida (cod: IN INT,  
Puenta: OUT INT, Itono: OUT DOUBLE) DO  
Puenta := obtenerPuenta (cod);  
Itono := obtenerItono (cod);  
Poblado++;  
END CONSULTA\_REDUCIDA;

OR WHEN (CONSULTA\_OTSOS'count == 0) AND (CONSULTA\_INMEDIATA'count == 0) =>  
ACCEPT consulta\_otros (cod: IN INT, Puenta: OUT INT, Itono: OUT DOUBLE) DO  
Puenta := obtenerPuenta (cod);  
Itono := obtenerItono (cod);  
Poblado++;  
END CONSULTA\_OTSOS;

END SELECT;

END LOOP;

END EMPLEADO;