

1) Sem mutex(cola = 1;

Sem mutexTropical = 1,

sem mutexNOTropical = 1;

Cola colaData [365];

Cola colatropical;

Cola colANotropical;

(A)

1	B
2	R+
3	B

Process Worker [id: 0..4] {

text+data;

P(mutexCola);

while (not colaData.isEmpty()) {

c.pop (data);

V(mutexCola);

boolean tropical = ProcessX(data);

if (tropical) {

P(mutexTropical);

colaTropical.push (data);

V(mutexTropical);

} else {

P(mutexNOTropical);

colaNOTropical.push (data);

V(mutexNOTropical);

}

P(mutexCola);

}

V(mutexCola);

}

2) sem mutexPerido = 1;
 Sem HayPerido = 0;
 sem Espera[N] = ([N], 0);
 Cola c;
 boolean pude[N];

```

Process Cliente [ID=0..N-1] {
    while (true) {
        P(mutexPedir);
        c.push(id);
        V(mutexPedir);
        V(hayPedido);
        P(espera[id]);
        WriteIn (puce[id]);
    }
}

```

Process Imposoraf [Point id;]

while (true) {

P (nayPedido);

P (mutexPedido);

c.pop (ID);

V (mutexPedido);

txtPedido = ObtenerPedido (ID);

bandok = verificarPizza (pedido);

pure [ID] = OK;

V (espera [ID]);

to hacer al
enviarlo

debe cliente la cote
pr

to cliente la cote

to hacer al
enviarlo

to hacer al
enviarlo

to recibir la
pizza

to recibir la
pizza

3) Process Persona [id: 0..49]

int ead = obtenerEad();

AdminCabina.llegar(id, ead),
TomarFoto(),

AdminCabina.Salir()

}

Monitor AdminCabina {

cond Espera[50]; int id;

boolean libre = true;

int esperando = 0;

colaDrenada c,

Procedure Llegar(id: in int, ead: in int){

if (not libre){

esperando++;

c.push(id, ead); //Anadir a persona mayor
wait(espera[id]);

else{

libre = false;

}

Procedure Salir(){

if (esperando > 0){

esperando -;

c.pop(id);

Signal(espera[id]),

} else{

libre = true;

}

}