

## Posibles soluciones a los ejercicios del parcial práctico de memoria distribuida del 9-12-24

*Importante: las soluciones que se muestran a continuación no son las únicas que se pueden considerar correctas para los ejercicios planteados.*

1) Existe un médico que debe auditar los estudios médicos de 15 pacientes que tienen un turno asignado con él. Los pacientes son atendidos de a uno por vez (nunca hay dos pacientes al mismo tiempo en el consultorio) y el orden de atención lo define su ID (1..15). El proceso de atención consiste en que el paciente entrega el estudio y el médico le responde con el resultado luego de revisarlo. Implemente un programa que permita resolver el problema anterior usando **PMS**. **Notas:** la función `obtenerEstudio()` retorna el estudio a revisar para el paciente que la invoca; la función `revisarEstudio(estudio)` retorna el resultado de revisar el estudio recibido como entrada.

```
Process Paciente [i=1..15] {  
    estudio est;  
    resultado res;  
    // obtiene estudio  
    qr = obtenerEstudio();  
    // solicita atención y envía estudio  
    Medico!solicitarAtencion(est);  
    // recibe resultado  
    Medico?entregarResultado(res);  
}  
  
Process Medico {  
    estudio est;  
    resultado res;  
    int i;  
  
    for i in 1..15 {  
        // entrega paquete al cliente  
        Paciente[i]?solicitarAtencion(est);  
        // revisa el estudio  
        res = revisarEstudio(est);  
        // entrega resultado  
        Paciente[i]!entregarResultado(res);  
    }  
}
```

2) La UNLP está organizando su maratón anual y  $P$  personas que deben pasar a retirar su remera y chip antes de la carrera. Para ello, los corredores deben dirigirse al edificio de Rectorado, donde habrá un organizador que los atenderá. Los corredores son atendidos de acuerdo con el orden de llegada, teniendo prioridad los corredores ancianos sobre los jóvenes y adultos. Adicionalmente, los corredores con discapacidad tienen prioridad sobre todos los anteriores. Para ser atendidos, los corredores entregan su DNI al organizador, quien como respuesta les entrega la remera oficial y el chip asociado. Implemente un programa que permita resolver el problema anterior usando **ADA**. Notas: para conocer su prioridad, cada corredor puede llamar a la función *obtenerPrioridad()*, la cual retorna 0 si la persona tiene alguna discapacidad, 1 si es anciana, o 2 en otro caso; las funciones *obtenerRemera(dni)* y *obtenerChip(dni)* retornan la remera y el chip asociado para el DNI recibido como entrada, respectivamente; la función *obtenerDNI()* retorna el DNI para el corredor que la invoca; todas las tareas deben terminar.

**Procedure** Maraton **IS**

**TASK Type** Corredor;

**TASK** Organizador **IS**

```
entry atencionDisc (in dni: integer, rem: out Remera, ch: out Chip);
entry atencionAnc (in dni: integer, rem: out Remera, ch: out Chip);
entry atencionGral (in dni: integer, rem: out Remera, ch: out Chip);
```

**end** Organizador;

corredores: array (1..P) of Corredor;

**TASK Body** Corredor **IS**

```
dni: integer;
prioridad: integer;
rem: Remera;
ch: Chip;
```

**Begin**

```
dni = obtenerDNI();
prioridad = obtenerPrioridad();
if (prioridad = 0) then
    Organizador.atencionDisc(dni, r, c);
elsif (prioridad = 1) then
    Organizador.atencionAnc(dni, r, c);
else
    Organizador.atencionGral(dni, r, c);
end if;
```

**end** Corredor;

```
TASK Body Organizador IS
Begin
  for i in 1..P loop
    SELECT
      Accept atenciónDisc (dni: in integer, rem: out Remera, ch: out Chip) do
        -- atender corredor con discapacidad
        rem = obtenerRemera(dni);
        ch = obtenerChip(dni);
      End atenciónDisc;
    OR WHEN (atencionDisc'count = 0) =>
      Accept atenciónAnc (dni: in integer, rem: out Remera, ch: out Chip) do
        -- atender corredor anciano
        rem = obtenerRemera(dni);
        ch = obtenerChip(dni);
      End atenciónAnc;
    OR WHEN (atencionDisc'count = 0) AND (atencionAnc'count = 0) =>
      Accept atenciónGral (dni: in integer, rem: out Remera, ch: out Chip) do
        -- atender corredor general
        rem = obtenerRemera(dni);
        ch = obtenerChip(dni);
      End atenciónGral;
    end SELECT;
  End loop;
End Organizador;
begin
  null;
end Maraton;
```