

Posibles soluciones a los ejercicios del parcial práctico MD del 08-11-22

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

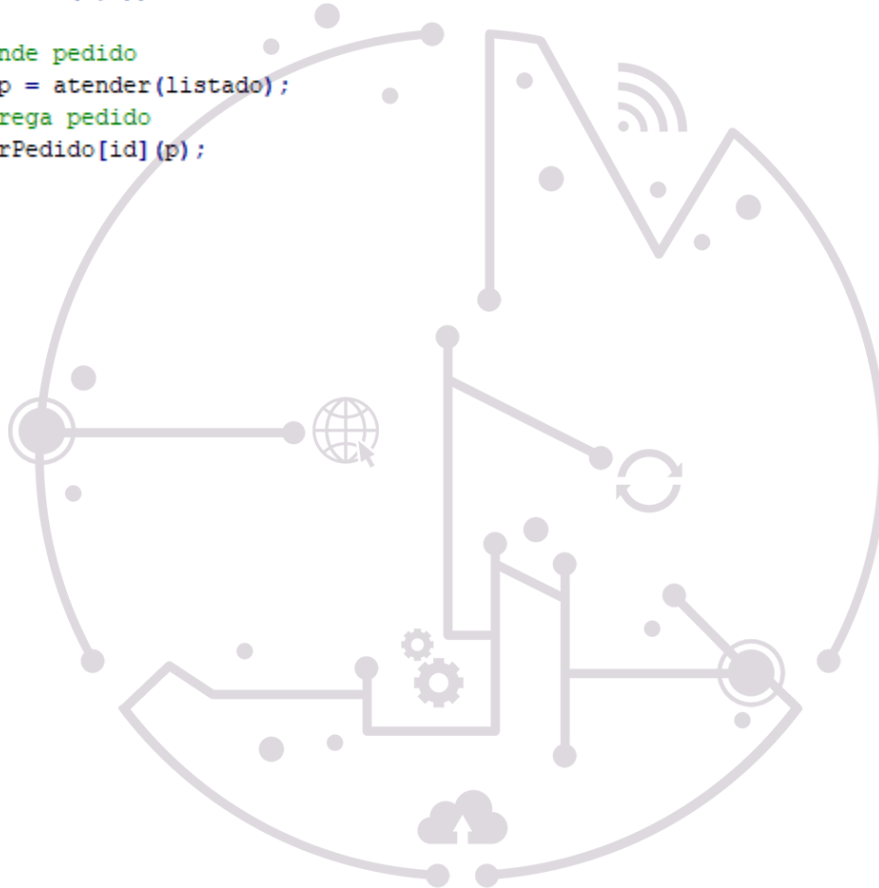
1. Resolver con PMA el siguiente problema. Se debe modelar el funcionamiento de una casa de venta de repuestos automotores, en la que trabajan V vendedores y que debe atender a C clientes. El modelado debe considerar que: (a) cada cliente realiza un pedido y luego espera a que se lo entreguen; y (b) los pedidos que hacen los clientes son tomados por cualquiera de los vendedores. Cuando no hay pedidos para atender, los vendedores aprovechan para controlar el stock de los repuestos (tardan entre 2 y 4 minutos para hacer esto). **Nota:** maximizar la concurrencia.

```
chan realizarPedido (int,string);
chan entregarPedido[C] (Pedido);
chan puedoAtender (int);
chan asignarPedido[V] (int,string);

Process Cliente [i:0..C-1] {
    Pedido p;
    // genera el listado de respuestas
    string listado = generar();
    // realiza pedido
    send realizarPedido (i,listado);
    // queda a la espera de la entrega
    receive entregarPedido[i] (p);
}

Process AdministradorDeVendedores {
    int idC, idV;
    string listado;
    while (true) {
        // recibe aviso de vendedor
        receive puedoAtender(idV);
        // si hay pedidos para atender
        if (not(empty(realizarPedido))) {
            // recibe pedido
            receive realizarPedido(idC,listado);
            // lo asigna a un vendedor libre
            send asignarPedido[idV] (idC,listado);
        } else
            // le indica que no hay pedido para atender
            send asignarPedido[idV] (NULL,NULL);
    }
}
```

```
Process Vendedor [i:0..V-1] {  
    int id;  
    string listado;  
    while (true) {  
        // avisa que está libre  
        send puedoAtender(i);  
        // esperar respuesta para saber si hay pedido para atender  
        receive asignarPedido[i] (id,listado);  
        // si no hay pedido para atender  
        if (listado == NULL)  
            delay(random(2,4)); // controla stock  
        else {  
            // atiende pedido  
            Pedido p = atender(listado);  
            // entrega pedido  
            entregarPedido[id](p);  
        }  
    }  
}
```



2. Resolver con ADA el siguiente problema. Se quiere modelar el funcionamiento de un banco, al cual llegan clientes que deben realizar un pago y llevarse su comprobante. Los clientes se dividen entre los regulares y los premium, habiendo R clientes regulares y P clientes premium. Existe un único empleado en el banco, el cual atiende de acuerdo al orden de llegada, pero dando prioridad a los premium sobre los regulares. Si a los 30 minutos de llegar un cliente regular no fue atendido, entonces se retira sin realizar el pago. Los clientes premium siempre esperan hasta ser atendidos.

```

TASK Type ClienteRegular;
TASK Type ClientePremium;

TASK Empleado IS
  entry atencionRegular (p: in Pago, c: out Comprobante);
  entry atencionPremium (p: in Pago, c: out Comprobante);
end Empleado;

clientesRegulares: array (1..R) of ClienteRegular;
clientesPremium: array (1..P) of ClientePremium;

TASK Body ClienteRegular IS
  pago: Pago;
  comp: Comprobante;
Begin
  SELECT
    -- Pedi atención
    Empleado.atencionRegular(pago, comp);
  OR DELAY 180.0
    -- retirarse;
    Null;
  End SELECT;
End Cliente;

TASK Body ClientePremium IS
  pago: Pago;
  comp: Comprobante;
Begin
  Empleado.atencionPremium(pago, comp);
End Cliente;

```

```
TASK Body Empleado IS
Begin
  loop
    SELECT
      When (atencionPremium'count == 0) ==> Accept atenciónRegular (p: in Pago, c: out Comprobante) do
        -- atender
        c = procesar(p);
      End atenciónRegular
    OR
      Accept atenciónPremium (p: in Pago, c: out Comprobante) do
        -- atender
        c = procesar(p);
      End atenciónPremium
    End SELECT;
  End loop;
End Empleado;
```

