

**Posibles soluciones a los ejercicios del parcial práctico MD del 13-11-23**

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

1. En una oficina existen 100 empleados que envían documentos para imprimir en 5 impresoras compartidas. Los pedidos de impresión son procesados por orden de llegada y se asignan a la primera impresora que se encuentre libre. Implemente un programa que permita resolver el problema anterior usando PASAJE DE MENSAJES ASINCRÓNICO (PMA).

```
chan solicitud_impresion(int,string);
chan aviso_doc_impreso[N](string);

Process Empleados [i=1..N] {
    string documento, docImpreso;
    while (true) {
        documento = generar();
        send solicitud_impresion(i,documento);
        receive aviso_doc_impreso[i] (docImpreso);
    }
}

Process Impresora [i=1..5] {
    string doc, docImp;
    int idEmpleado;
    while (true) {
        receive solicitud_impresion (idEmpleado,doc);
        // docImp = imprimir(doc)
        send aviso_doc_impreso[idEmpleado] (docImp);
    }
}
```

- 2) Resuelva el mismo problema anterior pero ahora usando PASAJE DE MENSAJES SINCRÓNICO (PMS).

```
Process Empleado [i=1..N] {
    string documento, docImpreso;
    while (true) {
        documento = generar();
        Admin!solicitudImpresion(i,documento);
        Impresora[*]?avisoDocImpreso(docImpreso);
    }
}

Process Admin {
    queue cola;
    string doc;
    int idImpresora, idEmpleado;
    while (true)
        if Empleado[*] ? solicitudImpresion(idEmpleado,doc) ->
            push(cola,(idEmpleado,doc));
        [] (not empty(cola)); Impresora [*] ? solicitudTrabajo(idImpresora) ->
            (idEmpleado,doc) = pop(cola);
            Impresora[idImpresora] ! asignacionTrabajo (idEmpleado,doc);
        fi
    }
}

Process Impresora [i=1..5] {
    string doc, docImp;
    while (true){
        Admin!solicitudImpresion(i);
        Admin?asignacionTrabajo(idEmpleado,doc);
        // docImp = imprimir(doc)
        Empleado[idEmpleado]!avisoDocImpreso(docImp);
    }
}
```

- 3) Resuelva con **ADA** el siguiente problema: la página web del Banco Central exhibe las diferentes cotizaciones del dólar oficial de 20 bancos del país, tanto para la compra como para la venta. Existe una tarea programada que se ocupa de actualizar la página en forma periódica y para ello consulta la cotización de cada uno de los 20 bancos. Cada banco dispone de una API, cuya única función es procesar las solicitudes de aplicaciones externas. La tarea programada consulta de a una API por vez, esperando a lo sumo 5 segundos por su respuesta. Si pasado ese tiempo no respondió, entonces se mostrará vacía la información de ese banco.

```
Procedure BCRA is

task type APIBanco is
    entry consultarDolarOficial (compra, venta: OUT String);
end APIBanco;

task TareaProgramada;

arrAPIs: array(1..20) of APIBanco;

task body TareaProgramada is
    int c1, c2;
    cotizaciones: array(1..2, 1..20) of String;
begin
    loop
        for i in 1..20 loop
            select
                arrAPIs(i).consultarDolarOficial(c1,c2);
                cotizaciones(1,i) := c1;
                cotizaciones(2,i) := c2;
            or delay 5.0
                cotizaciones(1,i) := ' ';
                cotizaciones(2,i) := ' ';
            end select;
        end loop;
        mostrar(cotizaciones);
    end loop;
end TareaProgramada;

task body APIBanco is
begin
    loop
        accept consultarDolarOficial(compra,venta: OUT string) do
            compra := cotizaconCompra();
            venta := cotizaconVenta();
        end consultarDolarOficial;
    end loop;
end APIBanco;

Begin
    null;
end BCRA;
```

```
Procedure BCRA is

task type APIBanco is
    entry consultarDolarOficial (compra, venta: OUT String);
end APIBanco;

task TareaProgramada;

arrAPIs: array(1..20) of APIBanco;
cotizaciones: array(1..2, 1..20) of String;

task body TareaProgramada is
    int c1, c2;
begin
    for i in 1..20 loop
        select
            arrAPIs(i).consultarDolarOficial(c1,c2);
            cotizaciones(1,i) := c1;
            cotizaciones(2,i) := c2;
        or delay 5.0
            cotizaciones(1,i) := ' ';
            cotizaciones(2,i) := ' ';
        end select;
    end loop;
    actualizar(cotizaciones);
end TareaProgramada;

task body APIBanco is
begin
    loop
        accept consultarDolarOficial(compra,venta: OUT string) do
            compra := cotizaconCompra();
            venta := cotizaconVenta();
        end consultarDolarOficial;
    end loop;
end APIBanco;
```