PC | PRACTICA 5

EJERCICIO 5

```
PROCEDURE PLAYA IS

    TASK TYPE Persona is
        Entry Ident(id: IN integer);
        Entry comenzar();
        Entry resultado(res: IN integer);
    End Persona;

    TASK TYPE AdminDeGrupo is
        Entry IdentG(id: IN integer);
        Entry llegue(idP: IN integer);
        Entry termine(x: IN integer);
        Entry resultado(res: IN integer);
    End AdminDeGrupo;

    TASK AdminGeneral is
        Entry finalizoGrupo(idG: IN integer; x: IN integer);
    End AdminGeneral;

    arrPersonas: array(1..20) of Persona;
    arrAdmins: array(1..5) of AdminDegrupo;

    TASK BODY Persona is
        nroGrupo, id, res: integer;
        cant: integer := 0;
    Begin
        nroGrupo := getGrupo();
        Accept Ident(id: IN integer);
        arrAdmins(nroGrupo).llegue(id);
        Accept comenzar();
        for i in 1..15 loop
            cant := cant + Moneda();
        end loop;
```

```
        arrAdmins(nroGrupo).termine(cant);
        Accept resultado(res: IN integer);
    End Persona;

    TASK BODY AdminDeGrupo is
        id, idP, x, res: integer;
        tot: integer := 0;
        ids: array(1..4) of integer;
    Begin
        Accept IdentG(id: IN integer);
        for i in 1..4 loop
            Accept llegue(idP: IN integer);
            ids[i] := idP;
        end loop;
        for i in 1..4 loop
            arrPersonas(ids[i]).comenzar();
        end loop;
        for i in 1..4 loop
            Accept termine(x: IN integer) do
                tot := tot + x;
            end termine;
        end loop;
        AdminGeneral.finalizoGrupo(id, tot);
        Accept resultado(res: IN integer);
        for i in 1..4 loop
            arrPersonas(ids[i]).resultado(res);
        end loop;
    End AdminDeGrupo;

    TASK BODY AdminGeneral is
        idMax: integer := -1;
        cantMax: integer := -1;
        idG, x: integer;
    Begin
        for i in 1..5 loop
            Accept finalizoGrupo(idG: IN integer, x: IN integ
            if x > cantMax then
                cantMax := x;
```

```
                idMax := idG;
            end if;
        end loop;
        for i in 1..5 loop
            arrAdmins(i).resultado(idMax);
        end loop;
    End AdminGeneral;

BEGIN
    for i in 1..20 loop
        arrPersonas(i).Ident(i);
    end loop;
    for i in 1..5 loop
        arrAdmins(i).IdentG(i);
    end loop;
END Playa;
```

EJERCICIO 6

```
PROCEDURE CalculoVector is

    TASK TYPE Worker is
        Entry ident(id: IN integer);
        Entry comenzar();
    End Worker;

    TASK Coordinador is
        Entry resultado(prom: IN integer);
    End Coordinador;

    arrWorkers: array(1..10) of Worker;

    TASK BODY Worker is
        id, prom: integer;
        tot: integer := 0;
        vec: array(1..1_000_000_000) of integer;
    Begin
```

```
        vec := getVector();
        Accept ident(id: IN integer) do
            id := id;
        end ident;
        Accept comenzar();
        for i in ((1*id) - 1)..((100_000 * id) - 1) loop
            tot := tot + vec[i];
        end loop;
        prom := tot / 100_000;
        Coordinador.resultado(prom);
    End;

    TASK BODY Coordinador is
        x, prom: integer;
        tot: integer := 0;
    Begin
        for i in 1..10 loop
            arrWorkers[i].comenzar();
        end loop;
        for i in 1..10 loop
            Accept resultado(x: IN integer) do
                x := x;
            end resultado;
            tot := tot + x;
        end loop;
        prom := tot / 10;
    End Coordinador;

BEGIN
    for i in 1..10 loop
        arrWorkers(i).ident(i);
    end loop;
END CalculoVector;
```

EJERCICIO 7

```
PROCEDURE Policia is

    TASK TYPE Servidor is
        Entry pedido(test: IN img);
    End Servidor;

    TASK Especialista is
        Entry resultado(codigo: IN integer; valor: IN integer
    End Especialista;

    arrServidores: array(1..8) of Servidor;

    TASK BODY Servidor is
        test: img;
        codigo, valor: integer;
    Begin
        while true loop
            Accept pedido(test: IN img) do // No hago el proc
                test := test;
            end pedido;
            Buscar(test, codigo, valor);
            Especialista.resultado(codigo, valor);
        end loop;
    End Servidor;

    TASK BODY Especialista is
        test: img;
        codigo, valor: integer;
        codMax: integer := 0;
        valorMax: integer := -1;
    Begin
        while true loop
            test := getHuella();
            for i in 1..8 loop
                arrServidores(i).pedido(test);
            end loop;
            for i in 1..8 loop
                Accept resultado(codigo: IN integer; valor: I
```

```
                    if valor > valorMax then
                        codMax := codigo;
                        valorMax := valor;
                    end if;
                end resultado;
            end loop;
            codMax := 0;
            valorMax := -1;
        end loop;
    End Especialista;

BEGIN
    NULL;
END Policia;
```

EJERCICIO 8

```
PROCEDURE CamionesDeBasura is

    TASK TYPE Persona is
        Entry ident(id: IN integer);
        Entry pasoCamion();
    End Persona;

    TASK TYPE Camion is
        Entry identC(id: IN integer; idP: IN integer);
        Entry recoleccion(direc: IN String; idP: IN integer);
    End Camion;

    TASK Coordinador is
        Entry pedido(idP: IN integer);
        Entry camionLibre(idC: IN integer);
        Entry recoleccionRealizada(idFin: IN integer);
    End Coordinador;

    arrPersonas: array(1..P) of Persona;
    arrCamiones: array(1..3) of Camion;
```

```
TASK BODY Persona is
    id: integer;
    ok: boolean := false;
Begin
    Accept ident(id: IN integer);
    SELECT
        Coordinador.pedido(id);
    OR DELAY 900.0
        while !ok loop
            SELECT
                Accept pasoCamion() do
                    ok := true;
                end pasoCamion;
            ELSE
                SELECT
                    Coordinador.pedido(id);
                OR DELAY 900.0
                    NULL;
                END SELECT;
            END SELECT;
        end loop;
    END SELECT;
End Persona;

TASK BODY Camion is
    id, idP: integer;
    direc: String;
Begin
    Accept identC(id: IN integer);
    while true loop
        Coordinador.camionLibre(id);
        Accept recoleccion(direc: IN String; idP: IN inte
        RealizarRecoleccion(direc);
        Coordinador.recoleccionRealizada(idP);
    end loop;
End Camion;
```

```
    TASK BODY Coordinador is
        pedidos: array(1..P) of integer := ([P] 0);
        idP, idC, idMax, idFin: integer;
        pedidosPendientes: integer := 0;
    Begin
        while true loop
            SELECT
                Accept pedido(idP: IN integer) do
                    pedidos[idP] := pedidos[idP] + 1; //Incre
                    pedidosPendientes := pedidosPendientes + 
                end pedido;
            OR
                WHEN (pedidosPendientes > 0) => Accept camion
                    idMax := getPosMaximo(pedidos); //Toma el
                    arrCamiones[idC].recoleccion(getDireccion
                end camionLibre;
            OR
                Accept recoleccionRealizada(idFin: IN integer
                    pedidosPendientes := pedidosPendientes - 
                    arrPersonas[idFin].pasoCamion(); //Avisa 
                end recoleccionRealizada;
            END SELECT;
        end loop;
    End Coordinador;

BEGIN
    for i in 1..P loop
        arrPersonas(i).ident(i);
    end loop;
    for i in 1..3 loop
        arrCamiones(i).identC(i);
    end loop;
END CamionesDeBasura;
```

PC | PRACTICA REPASO MD