

## SEMAFOROS

```

sem sCola = 1;
sem llego = 0;
sem Pasar [200] = ([200] 0);
sem salir = 0;
int cantA = 0; cantB = 0;

```

```

Process Socio [id: 1..200] {
    text cand;
    P(sCola);
    Push(cola, id);
    V(sCola);
    V(llego);
    P(Pasar[id]);
    cand = elegirCandidato();
    IF (cand == A) {
        cantA++;
    } else {
        cantB++;
    }
    V(salir);
}

```

queue cola;

Nieva Juan Pablo

```

Process Secretario {
    int ids; text cand; text ganador;

```

```

    For (i=0; i < 200; i++) {

```

```

        P(llego);
        P(sCola);
        Pop(cola, ids);
        V(sCola);
        V(Pasar[ids]);
        llvota Socio
        P(salir);
    }

```

```

    cand = ElegirCandidato();

```

```

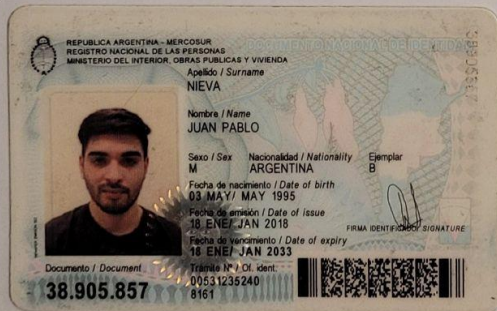
    IF (cand == A) {
        cantA++;
    } else {
        cantB++;
    }

```

```

    ganador = CalcularVotos(cantA, cantB);
}

```



## MONITORES

```

Process Persona [id: 1..40] {
    Complejo.BuscarCancha(nro);
    Cancha[nro].llego();
    Cancha[nro].salir();
}

```

```

Process Partido [id: 1..10] {
    Cancha[id].Inicio();
    delay(3600);
    Cancha[id].Fin();
}

```

```

monitor Cancha [id: 1..4] {
    Cond espera; iniciar; terminar;
    int cant = 0; cantF = 0;
}

```

```

Procedure Llego() {
    cant++;
    IF (cant == 4) {
        signal(iniciar);
    } else {
        wait(espera);
    }
}

```

```

Procedure Inicio() {
    IF (cant < 4) {
        wait(iniciar);
    }
    signal_all(espera);
}

```

```

Procedure Fin() {
    signal_all(terminar);
}

```

Nueva Juan Pablo

```

monitor Complejo {
    int cancha = 1; int cant = 0;
}

```

```

Procedure BuscarCancha(nro = out) {
    cant++;
    IF (cant == 4) {
        cant = 0;
        nro = cancha;
        cancha++;
    } else {
        nro = cancha;
    }
}

```

```

Procedure Salir() {
    wait(terminar);
}

```





## ADA PARTE 1

Procedure Banco is  
Task type cliente;

Nieva Juan Pablo

1

clientes: array (1..N) of Cliente;

Task Empleado is

Entry solicitarP (monto: in Int, c: out text);

Entry solicitarR (monto: in Int, c: out text);

end Empleado;

Task Body Cliente is

Int monto; text comp; bool atendido = false; Int intentos = 1;

Begin

IF (soyRegular()) then

select

Empleado.solicitarR (monto, comp);

or delay (3600).

null;

endselect

else {

while (intentos ≤ 5 AND not atendido) loop

select

Empleado.solicitarP (monto, comp);

atendido = true;

or delay (300)

intentos++;

endselect;

end loop;

endif;

end Cliente;

Sigue



## ADA PARTE 2

Task ~~Body~~ Empleado is

Nieva Juan Pablo

(2)

Begin

loop

select

Accept solicitap(m: in Int, c: out text) do

c = RealizarDeposito(m);

end solicitap;

OR

when (solicitap.count == 0) =>

Accept sollicitar(m: in Int, c: out text) do

c = RealizarDeposito(m);

end sollicitar;

endselect;

end loop;

end Empleado;

Begin

null;

end Banco;

