

```

/*
Resolver con Pasaje de Mensajes Sincrónicos (PMS) el siguiente problema. En un
torneo de programación
hay 1 organizador, N competidores y S supervisores. El organizador comunica el
desafío a resolver a cada
competidor. Cuando un competidor cuenta con el desafío a resolver, lo hace y lo
entrega para ser evaluado.
A continuación, espera a que alguno de los supervisores lo corrija y le indique
si está bien. En caso de
tener errores, el competidor debe corregirlo y volver a entregar, repitiendo la
misma metodología hasta
que llegue a la solución esperada. Los supervisores corrigen las entregas
respetando el orden en que los
competidores van entregando.
Nota: maximizar la concurrencia y no generar demora innecesaria.
*/

```

```

process Organizador {
    int idC;
    String desafio;

    for i = 0 to N-1 {
        Competidor[*]?pedidoDeDesafio(idC);
        Competidor[idC]!enviarDesafio(desafio);
    }
}

process Competidor [id: 0 to N-1] {
    String desafio, codigo;
    boolean aprobado = false;

    Organizador!pedidoDeDesafio(id);
    Organizador?enviarDesafio(desafio);

    while (!aprobado) {
        codigo = resolverDesafio(desafio);
        Coordinador!enviarEjercicio(codigo, id);
        Supervisor[*]?devolverCorreccion(aprobado);
    }
}

process Supervisor [id: 0 to S-1] {
    int idC;
    String codigo;
    boolean nota;

    while (true) {
        Coordinador!pedirEjercicio(id);
        Coordinador?enviarParaCorregir(codigo, idC);
        nota = corregirEjercicio(codigo);
        Competidor[idC]!devolverCorreccion(nota);
    }
}

process Coordinador {
    int idC, idS;
    String codigo;
    Queue entregas;

    do Competidor[*]?enviarEjercicio(codigo, idC) -> push(entregas, codigo,
idC);
    □ (!empty(entregas)) ; Supervisor[*]?pedirEjercicio(idS) ->
Supervisor[idS]!enviarParaCorregir(pop(entregas));
    od

```

}