```
/*
Resolver con ADA el siguiente problema. Un programador contrató a 5 estudiantes
para testear los
sistemas desarrollados por él. Cada estudiante tiene que trabajar con un sistema
diferente y debe
encontrar 10 errores (suponga que seguro existen 10 errores en cada sistema) que
pueden ser:
urgentes, importantes, secundarios. Cada vez que encuentra un error se lo
reporta al programador y
espera hasta que lo resuelva para continuar. El programador atiende los reportes
de acuerdo al orden
de llegada pero teniendo en cuenta las siguientes prioridades: primero los
urgentes, luego los
importantes y por último los secundarios; si en un cierto momento no hay
reportes para atender,
durante 5 minutos trabaja en un nuevo sistema que está desarrollando. Después de
resolver los 50
reportes de error (10 por cada estudiante) el programador termina su ejecución.
Nota: todas las
tareas deben terminar.
*/

procedure Testing is

task type Estudiante;

task Programador is
    entry enviarErrorUrgente(error: in-out Log);
    entry enviarErrorImportante(error: in-out Log);
    entry enviarErrorSecundario(error: in-out Log);
end Programador;

Estudiantes = array(0..4) of Estudiante;


task body Estudiante is
    error: Log;
    erroresEncontrados: Integer := 0;
begin
    while (erroresEncontrados < 10) loop
        error := buscarError();

        if (error.level = "urgente") Programador.enviarErrorUrgente(error);
        else if (error.level = "importante") Programador-
enviarErrorImportante(error);
        else Programador.enviarErrorSecundario(error);

        erroresEncontrados += 1;
    end loop;
end Estudiante;


task Programador is
    error: Log;
    erroresTotales: Integer := 0;
begin
    while (erroresTotales < 50) loop
        select
            accept enviarErrorUrgente(error: in-out Log) do
                error.solucionarError();
            end enviarErrorUrgente;
            erroresTotales + = 1;
        or when (enviarErrorUrgente´count = 0) =>
            accept enviarErrorImportante(error: in-out Log) do
```

```
                error.solucionarError();
            end enviarErrorImportante;
            erroresTotales + = 1;
        or when (enviarErrorUrgente´count = 0 and enviarErrorImportante´count =
0)
            accept enviarErrorSecundario(error: in-out Log) do
                error.solucionarError();
            end enviarErrorSecundario;
            erroresTotales + = 1;
        else
            delay(60*5); //trabaja en el nuevo sistema
        end select;
    end loop;
end Programador;


begin
    null;
end Testing;
```