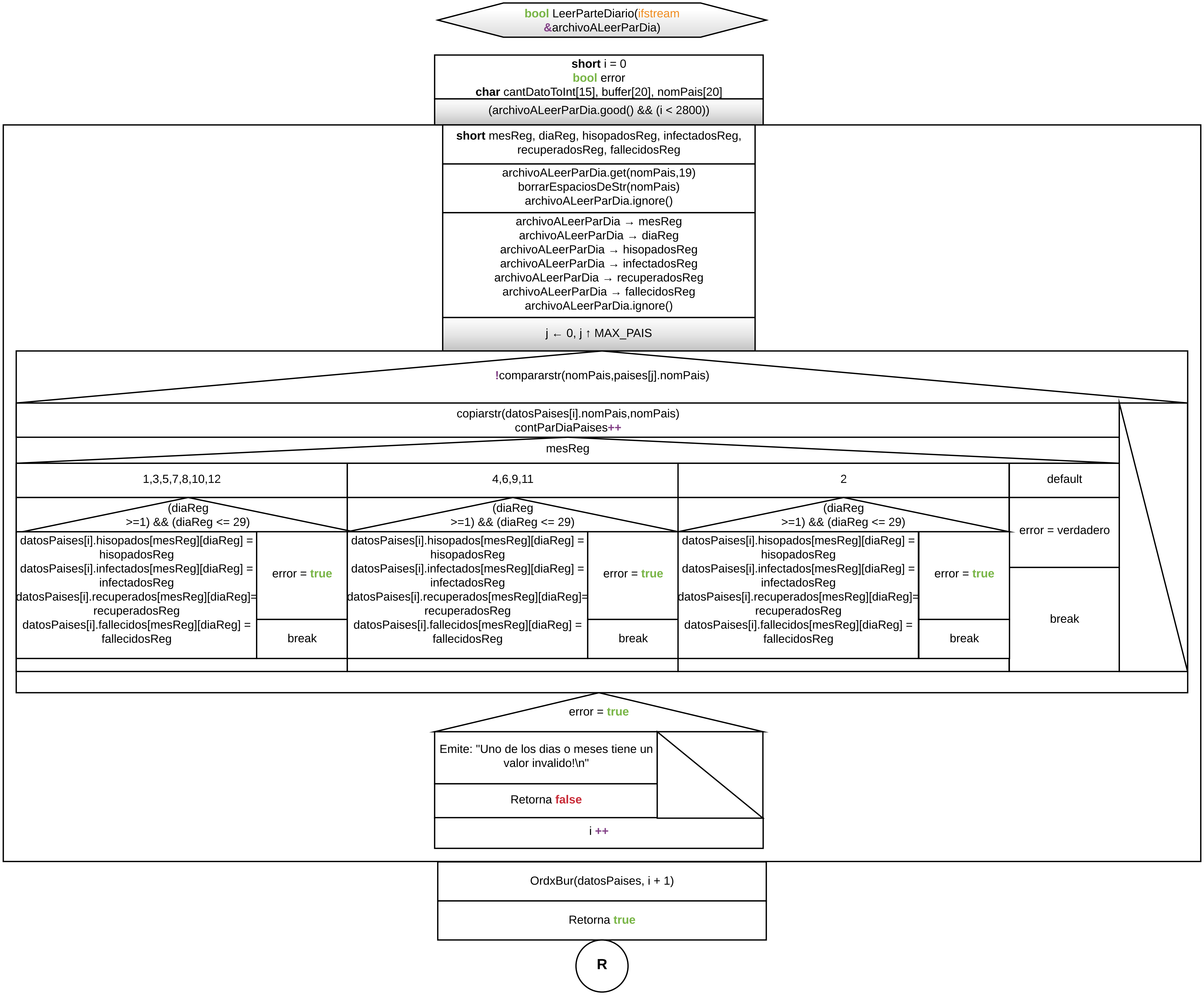
bool LeerParteDiario(ifstream &archivoALeerParDia)
short i = 0
bool error
char cantDatoToInt[15], buffer[20], nomPais[20]
(archivoALeerParDia.good() && (i < 2800))
short mesReg, diaReg, hisopadosReg, infectadosReg, recuperadosReg, fallecidosReg
archivoALeerParDia.get(nomPais,19)
borrarEspaciosDeStr(nomPais)
archivoALeerParDia.ignore()
archivoALeerParDia → mesReg
archivoALeerParDia → diaReg
archivoALeerParDia → hisopadosReg
archivoALeerParDia → infectadosReg
archivoALeerParDia → recuperadosReg
archivoALeerParDia → fallecidosReg
archivoALeerParDia.ignore()
j ← 0, j ↑ MAX_PAIS
!compararstr(nomPais,paises[j].nomPais)
copiarstr(datosPaises[i].nomPais,nomPais)
contParDiaPaises++
mesReg
1,3,5,7,8,10,12
4,6,9,11
2
default
(diaReg >=1) && (diaReg <= 29)
datosPaises[i].hisopados[mesReg][diaReg] = hisopadosReg
datosPaises[i].infectados[mesReg][diaReg] = infectadosReg
datosPaises[i].recuperados[mesReg][diaReg]= recuperadosReg
datosPaises[i].fallecidos[mesReg][diaReg] = fallecidosReg
error = true
break
(diaReg >=1) && (diaReg <= 29)
datosPaises[i].hisopados[mesReg][diaReg] = hisopadosReg
datosPaises[i].infectados[mesReg][diaReg] = infectadosReg
datosPaises[i].recuperados[mesReg][diaReg]= recuperadosReg
datosPaises[i].fallecidos[mesReg][diaReg] = fallecidosReg
error = true
break
(diaReg >=1) && (diaReg <= 29)
datosPaises[i].hisopados[mesReg][diaReg] = hisopadosReg
datosPaises[i].infectados[mesReg][diaReg] = infectadosReg
datosPaises[i].recuperados[mesReg][diaReg]= recuperadosReg
datosPaises[i].fallecidos[mesReg][diaReg] = fallecidosReg
error = true
break
error = verdadero
break
error = true
Emite: "Uno de los dias o meses tiene un valor invalido!\n"
Retorna false
i ++
OrdxBur(datosPaises, i + 1)
Retorna true
R