UNIVERSIDAD SAN PABLO DE GUATEMALA

Facultad de Ciencias Empresariales

Escuela de Ingeniería

Ingeniería en Sistemas Ciencias y de la Computación

Examen Final Compiladores

Tarea presentada en el curso de Compiladores

Impartido por: MA. Ing. Walter Oswaldo Ordóñez Sierra

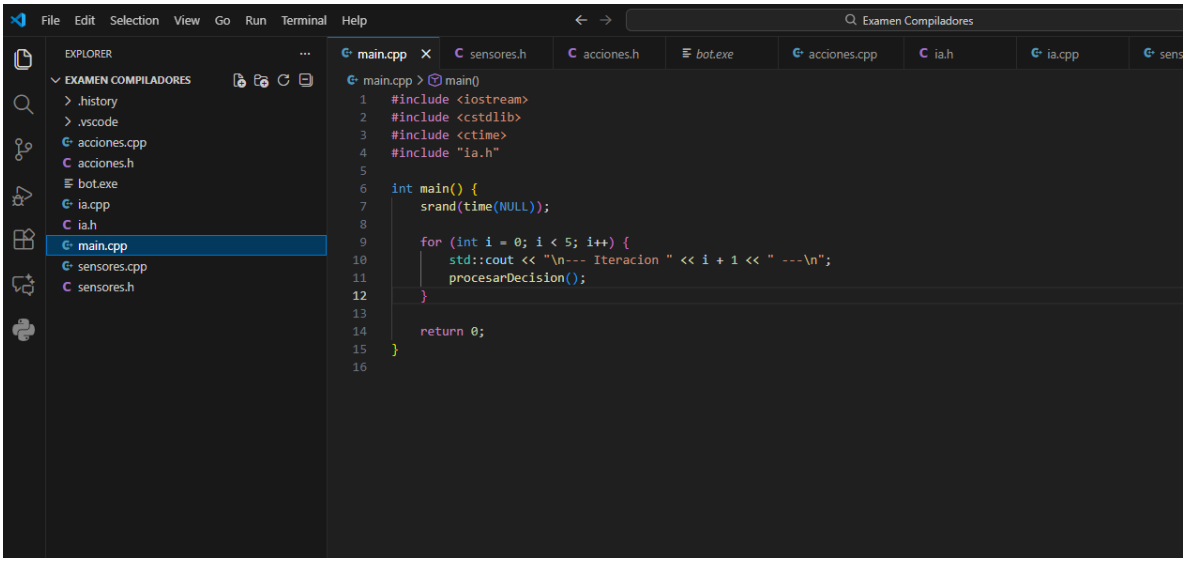
Yordy Estiven Rodríguez Morales - 2300321

Guatemala, 29 de noviembre 2025

Explicación de cada archivo del proyecto

main.cpp

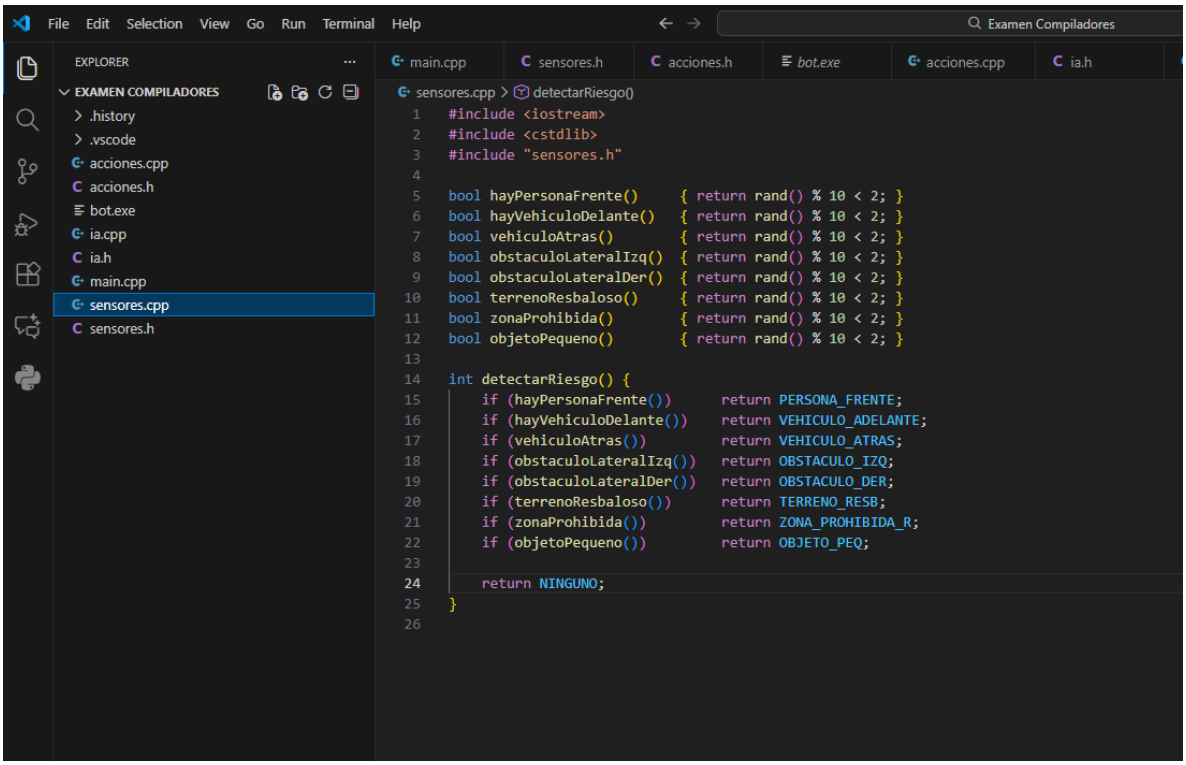
En este archivo está el punto de inicio del programa.
Aquí simplemente se configura el uso de números aleatorios para simular los sensores y se ejecuta un ciclo que corre varias iteraciones. En cada vuelta del ciclo se llama a la función que toma las decisiones del robot. Este archivo no hace nada de lógica compleja, solo organiza el orden en el que todo se ejecuta.



sensores.cpp

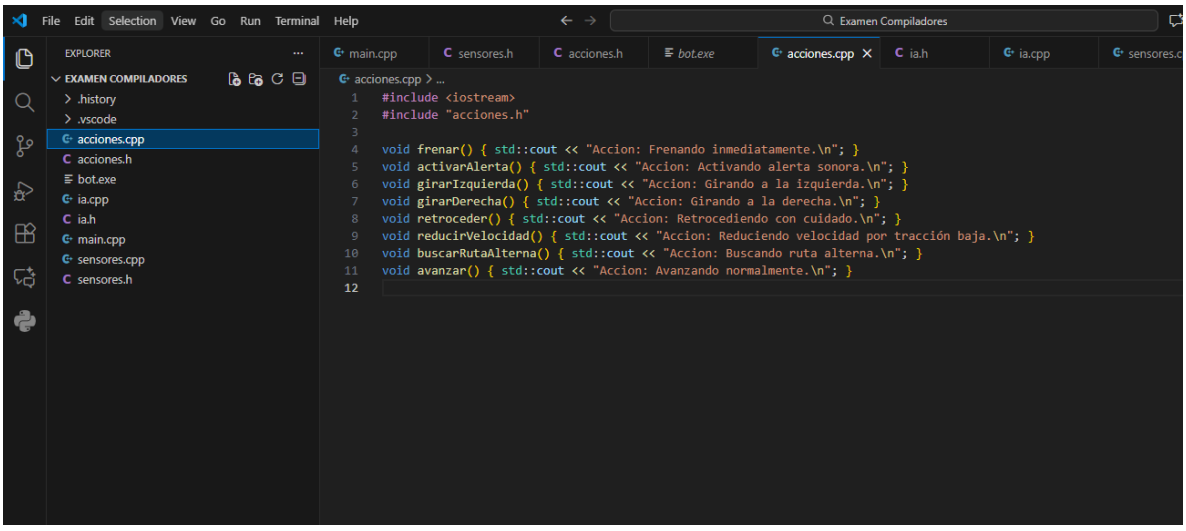
Este archivo contiene todas las funciones que simulan los sensores del robot.
Cada función devuelve un valor aleatorio que representa si el sensor detectó o no un riesgo. No es una lectura real, solo una simulación para poder probar el resto del programa.

También está la función `detectarRiesgo()`, que revisa cada "sensor" en un orden específico. Si uno de los sensores activa un riesgo, la función devuelve un código que identifica ese peligro. Si ninguno detecta nada, devuelve un valor que indica que está todo normal. Esta parte es la base para que la IA sepa qué decisión tomar después.



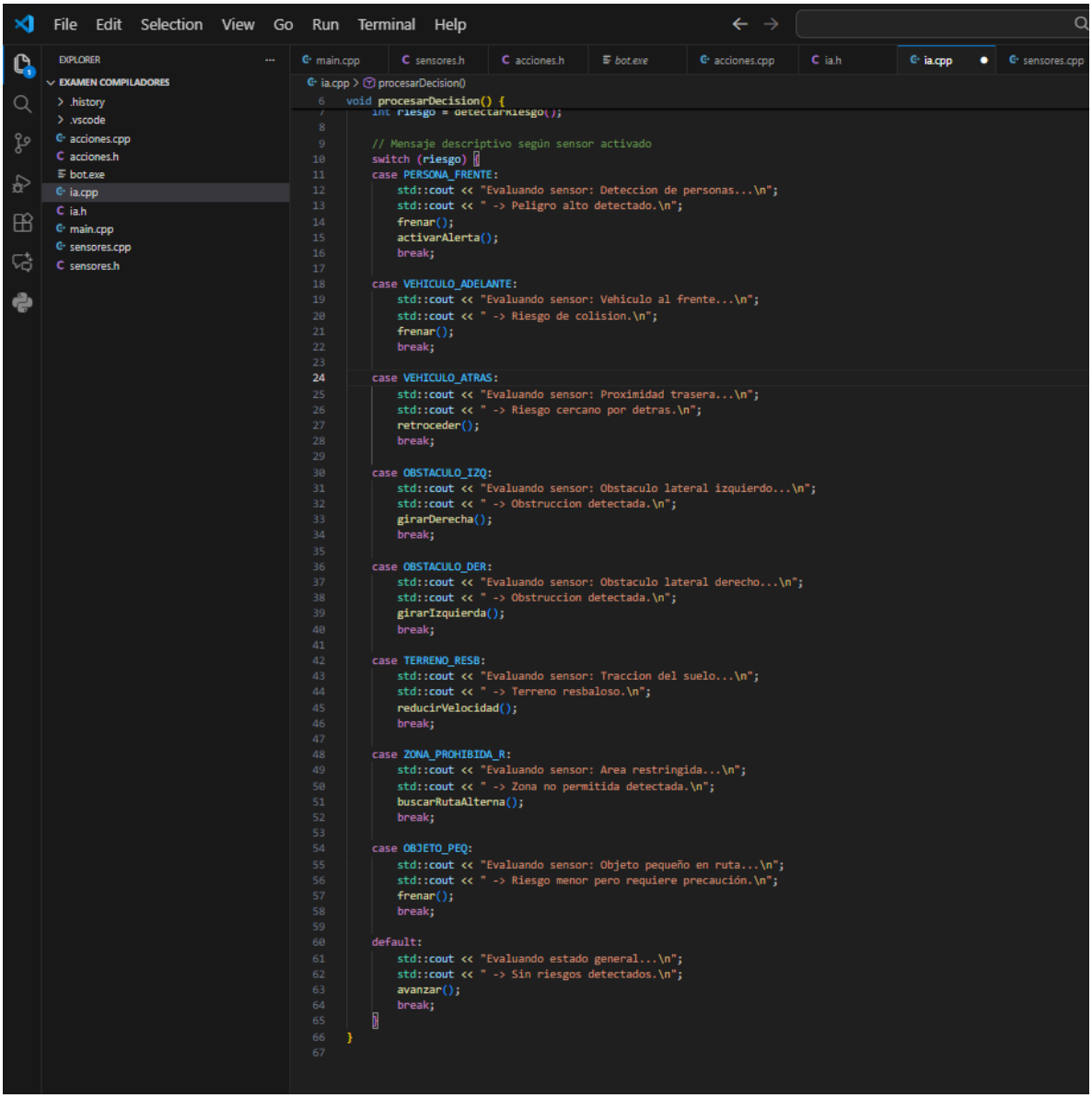
acciones.cpp

Aquí están las funciones que representan lo que el robot puede hacer.
Cada acción solo imprime un mensaje indicando lo que ejecutaría el robot en ese caso: frenar, girar, retroceder, reducir velocidad, avanzar, etc.
No controla motores ni nada así; simplemente muestra las acciones que corresponderían según el riesgo que se detectó.



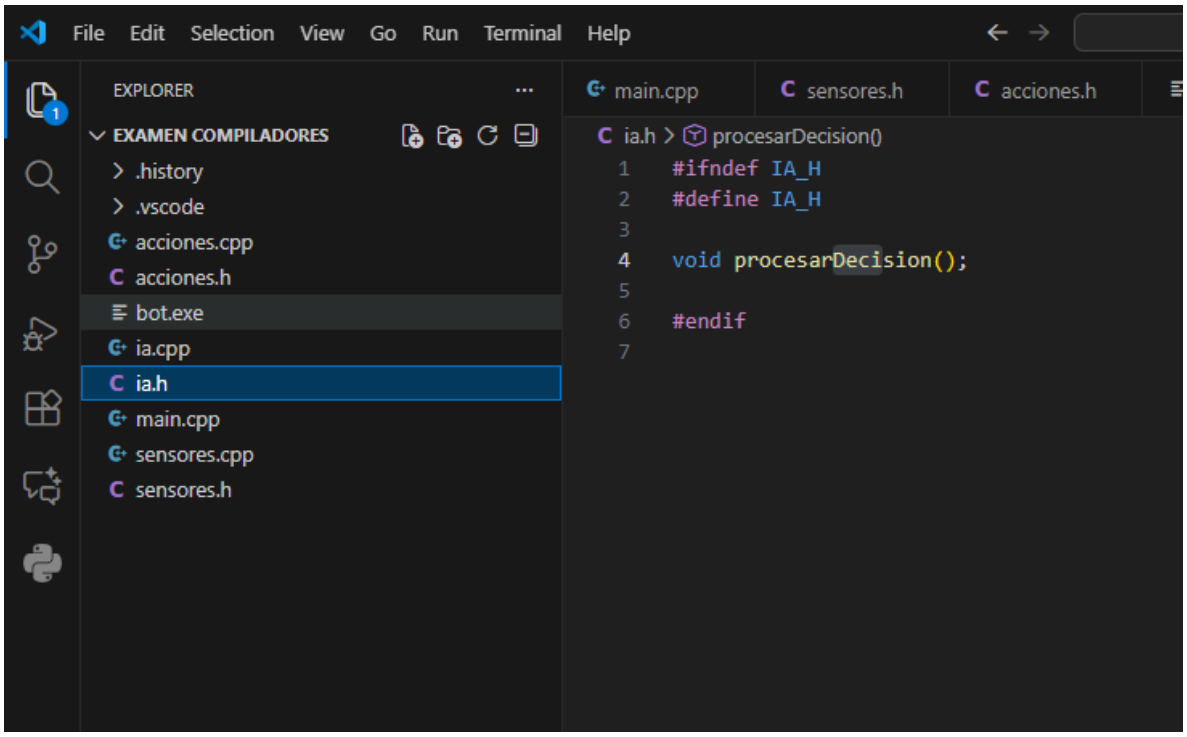
ia.cpp

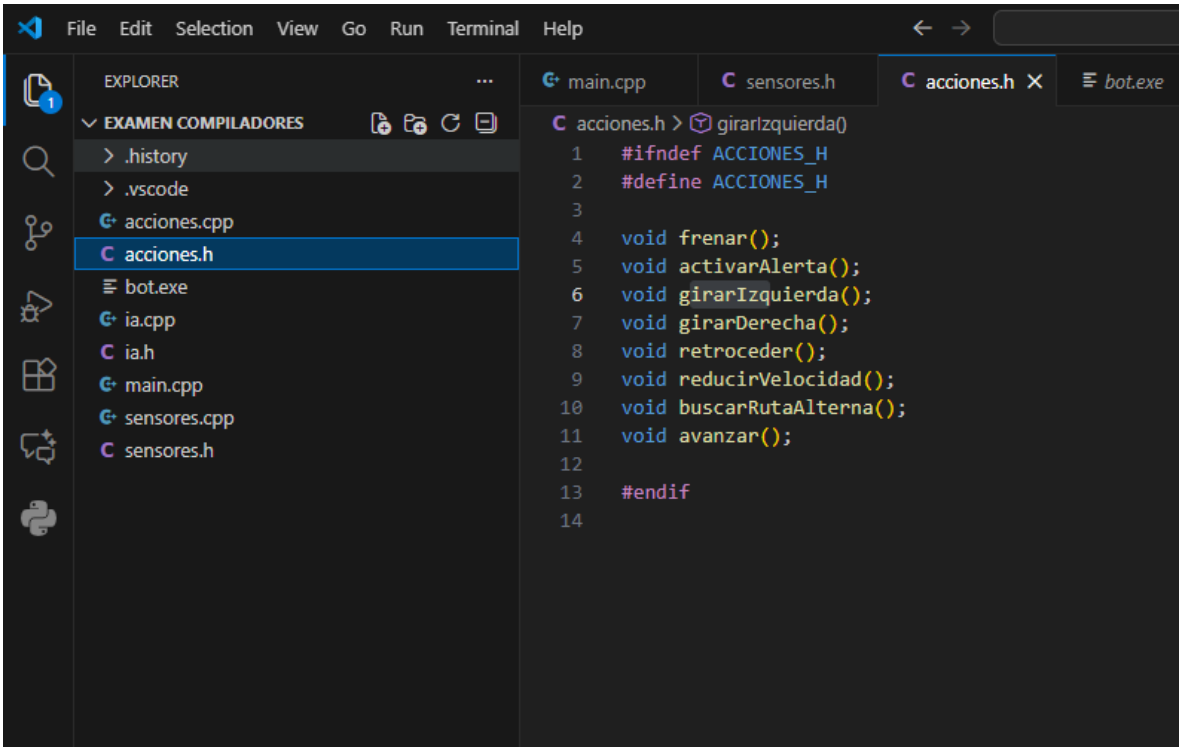
En este archivo está la lógica que decide qué hacer dependiendo del riesgo que detectan los sensores.
Lo primero que hace es llamar a la función `detectarRiesgo()`. Luego compara el resultado con una lista de casos posibles.
Cada caso activa una o varias acciones diferentes. Si el riesgo es una persona, frena y enciende la alerta. Si es un vehículo adelante, frena. Si el peligro está atrás, retrocede, y así con los demás.
Si no hay ningún riesgo, hace que el robot avance con normalidad.
Este archivo es donde se unen los sensores con las acciones.

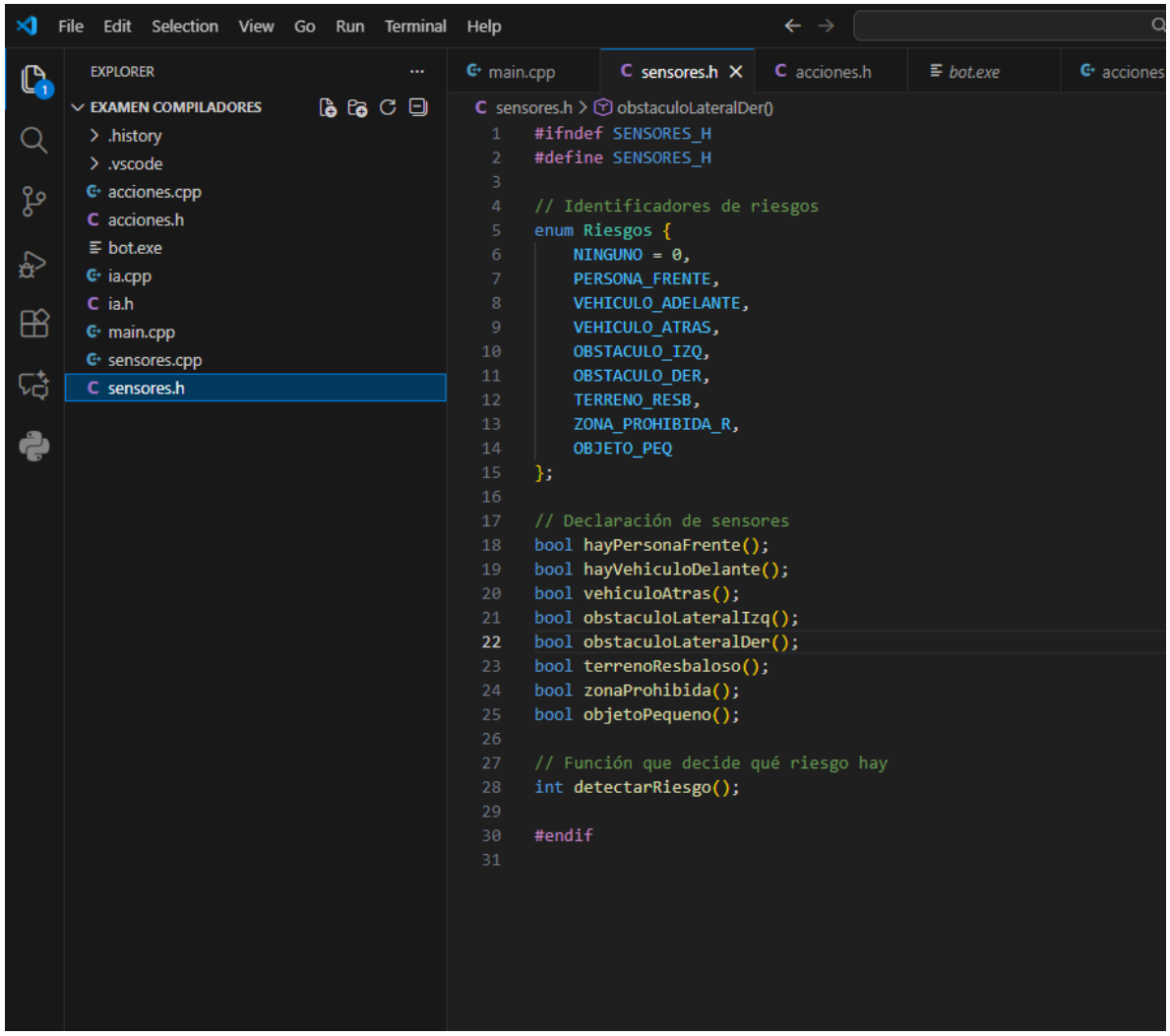


ia.h, acciones.h y sensores.h

Estos archivos son los encabezados donde solo se declaran las funciones y los valores necesarios para que los otros archivos puedan trabajar juntos.
Sirven para organizar mejor el proyecto y mantener el código más claro.
No contienen lógica interna, solo las definiciones que permiten que cada archivo conozca las funciones y los nombres que necesita.






File Edit Selection View Go Run Terminal Help
EXPLORER
EXAMEN COMPILADORES
.history
.vscode
acciones.cpp
acciones.h
bot.exe
ia.cpp
ia.h
main.cpp
sensores.cpp
sensores.h
main.cpp
sensores.h
acciones.h
bot.exe
acciones
sensores.h > obstaculoLateralDer()
#ifndef SENSORES_H
#define SENSORES_H
// Identificadores de riesgos
enum Riesgos {
NINGUNO = 0,
PERSONA_FRENTE,
VEHICULO_ADELANTE,
VEHICULO_ATRAS,
OBSTACULO_IZQ,
OBSTACULO_DER,
TERRENO_RESB,
ZONA_PROHIBIDA_R,
OBJETO_PEQ
};
// Declaración de sensores
bool hayPersonaFrente();
bool hayVehiculoDelante();
bool vehiculoAtras();
bool obstaculoLateralIzq();
bool obstaculoLateralDer();
bool terrenoResbaloso();
bool zonaProhibida();
bool objetoPequeno();
// Función que decide qué riesgo hay
int detectarRiesgo();
#endif