



# Tecnológico de Monterrey

**Instituto Tecnológico y de  
Estudios Superiores de  
Monterrey**

TE3003B.501

**Integración de robótica y sistemas inteligentes (Gpo 501)**

Semestre: febrero - junio 2024

**Actividad 5 (Navegación reactiva)**

**Alumno:**

Daniel Ruán Aguilar

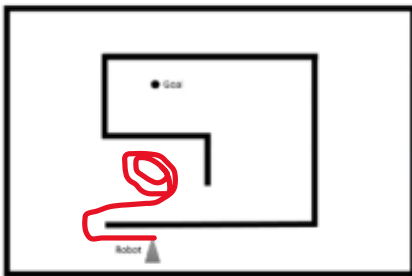
A01731921

## Objetivo:

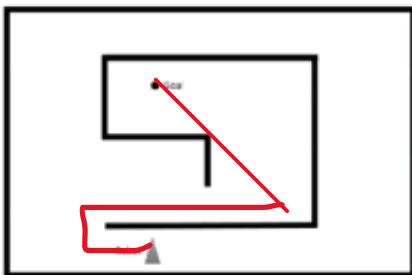
En este reporte se expresará el análisis de los posibles casos para cada uno de los algoritmos de evasión de obstáculos aplicados en tres escenarios. Se argumentará en cada caso cuales son los algoritmos que garantizan que siempre el robot alcance el objetivo planteado.

## Desarrollo:

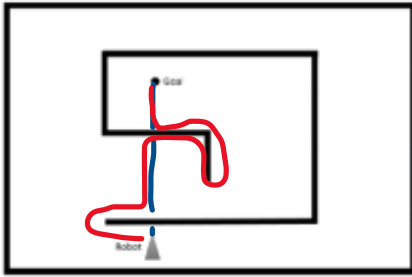
### Escenario 1:



Bug 0: Usando el algoritmo más básico. El robot simplemente rodea la pared del laberinto hasta que llega al objetivo. Sin embargo, al estar buscando siempre el objetivo, se crearía un bucle al querer rodear este tipo de obstáculos que cierran el paso.



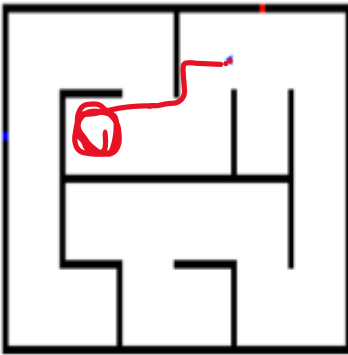
Bug 1: Este algoritmo utiliza rutas rectas cada que tiene camino libre para optimizar la trayectoria del robot, además tiene mayor alcance de proximidad para evitar los callejones sin salida. Por lo que soluciona este escenario.



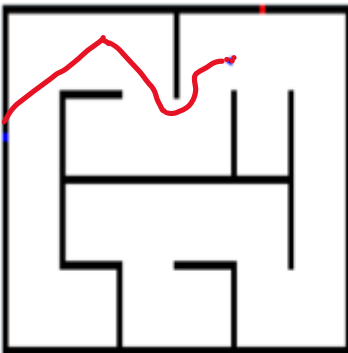
Bug 2: Este algoritmo siempre traza una línea recta al objetivo y al evitar obstáculos reanuda el recorrido de la línea recta. Por lo que también soluciona el laberinto, pero gasta más recursos y es más largo el recorrido.

### **Solución: Bug 1 & Bug 2**

#### **Escenario 2:**



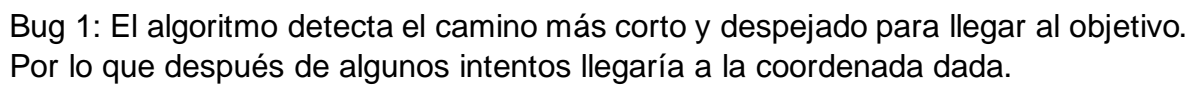
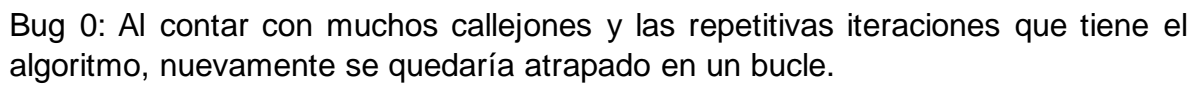
Bug 0: Nuevamente al contar con una programación básica se generaría un bucle al encontrar un callejón sin salida.

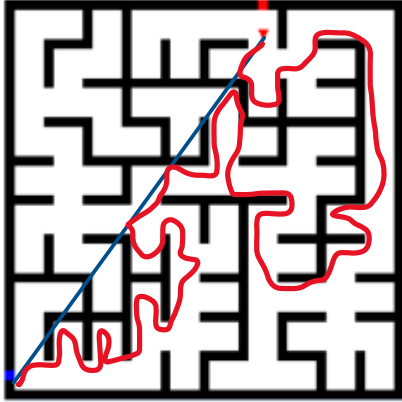


Bug 1: Al encontrar la ruta más corta y de manera lineal, puede evitar los callejones y encontrar la ruta más óptima para resolver el laberinto.



### Escenario 3:





Bug 2: Este algoritmo también funcionaría para llegar al destino ya que los callejones no son tan profundos y podría el robot salir de ellos fácilmente para seguir buscando la línea.

**Solución: Bug 1 & 2**

## Conclusión

Cómo podemos ver en los diagramas de todos los posibles casos de navegación reactiva, el bug 0 es un sencillo algoritmo que sirva para obstáculos más pequeños e independientes, pero cuando se trata de una serie de obstáculos que pueden estar conectados, es mejor recurrir a la complejidad de los otros dos algoritmos, los cuales se emplean dependiendo del caso de aplicación.