



# Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

División de Electrónica y Computación

Departamento de Ciencias Computacionales

Seminario de Solución a Problemas de Algoritmia

---

## Proyecto final.

---

Diseñar un sistema computacional que simule el comportamiento de entidades “presas” y “depredadoras” en un entorno. A continuación se describen las reglas que definen al comportamiento de estas entidades, el escenario en el que se desenvuelven será representado por un grafo, para la generación del grafo se analizarán imágenes (cada pixel representa un metro) que contienen círculos negros (vértices del grafo) y cualquier cantidad de obstáculos que impiden conectar a un vértice con otro. Los vértices son “zonas seguras temporales”, en un vértice, el depredador no puede atacar a la presa durante algún tiempo. Los escenarios miden 1000 x 1300 píxeles.

### **Presas.**

Entidad que busca llegar a un vértice objetivo. Pueden existir múltiples presas en el mismo entorno, si no existe un objetivo en el grafo, las presas permanecen en un vértice.

La presa tiene un contador que indica la resistencia que tiene, inicia en 1 y gana un contador cada vez que llega a un objetivo. Cada que un depredador la alcanza, le resta un contador. Si una presa es alcanzada por un depredador, y su contador de resistencia es 0, la presa desaparece.

La presa conoce todo el grafo e intentará dirigirse al objetivo utilizando el camino más corto pero si se ve asechada, cambiará su comportamiento para su salvaguarda. La presa puede tomar decisiones de hacia donde dirigirse en cualquier instante, incluso mientras recorre una arista.

La presa sabe cuando un depredador la está acechando, conoce al depredador, de donde viene y hacia donde se dirige en cada instante.

### **Depredadores.**

El objetivo de los depredadores es alcanzar a una presa, conocen a las presas que se encuentran en a un máximo de  $r$  metros a la redonda de él, pero solo pueden acechar a una, siempre y cuando no esté siendo acechada por otro depredador, entre más cerca esté la presa del depredador, mayor velocidad tendrá el depredador (se calcula en cada instante). Si un depredador no detecta presas en su radio, este recorre el grafo de forma aleatoria. Puede haber múltiples depredadores en el entorno.

Un depredador solo puede tener una presa en asecho. El depredador conoce la posición actual de la presa en asecho. El depredador puede cambiar de presa acechada en cualquier momento.

Cuando un depredador toma la decisión de recorrer una arista, debe terminar de recorrerla para tomar una nueva decisión. El depredador puede elegir quedarse en un vértice, en ese caso, el vértice dejará de ser seguro en  $t$  iteraciones, si el depredador sale del vértice, el vértice recuperará su propiedad de “zona segura”.

Cada que un depredador alcanza a una presa, le resta un contador de resistencia.

## Requerimientos Funcionales:

- Paradigma de programación: P.O.O.
- El sistema debe contener una interfaz gráfica intuitiva.
- Capacidad para seleccionar el escenario (imagen).
- Mostrar la imagen que representa al grafo.
- Posibilidad de crear presas y depredadoras:
  - o Seleccionar el vértice en donde se desea colocar cada presa y cada depredadora
  - o Inicializar posiciones de presas y depredadoras de forma aleatoria.
- Posibilidad de seleccionar un vértice “objetivo” (para las presas).
- Capacidad para mostrar, o no, lo siguiente:
  - o Rango de alcance de cada depredadora.
  - o Relación “depredador – presa asechada”.
  - o Representación gráfica de las decisiones de desplazamiento de:
    - presas en peligro.
    - Depredadoras acechando.
  - o Contadores de resistencia de las presas.

## Reporte:

1. Diagrama de clases propuesto.
2. Objetivo.
3. Marco teórico.
4. Desarrollo.
5. Pruebas y resultados.
6. Conclusiones.
7. Apéndice(s).