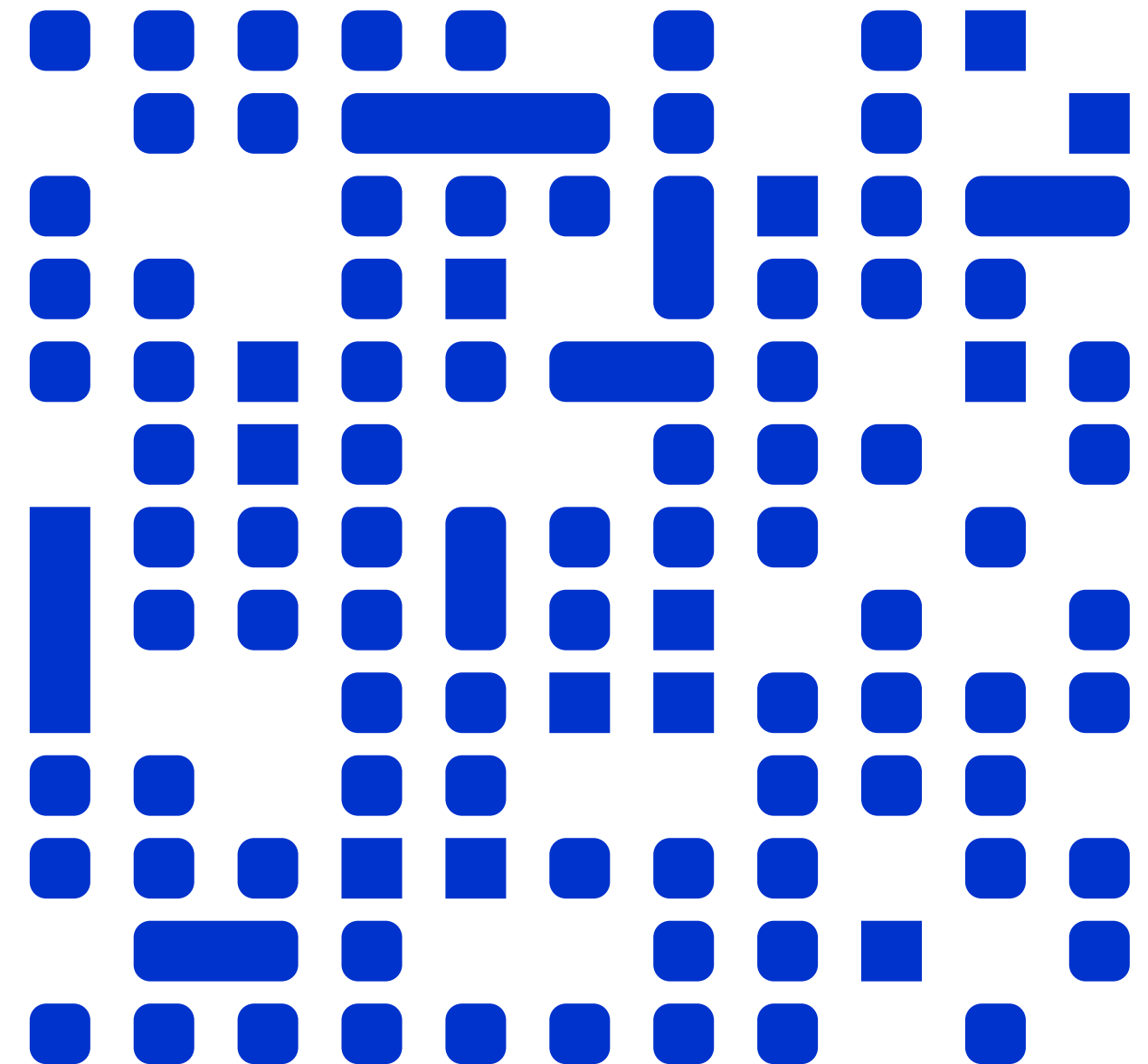


INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO
INTELIGENCIA ARTIFICIAL

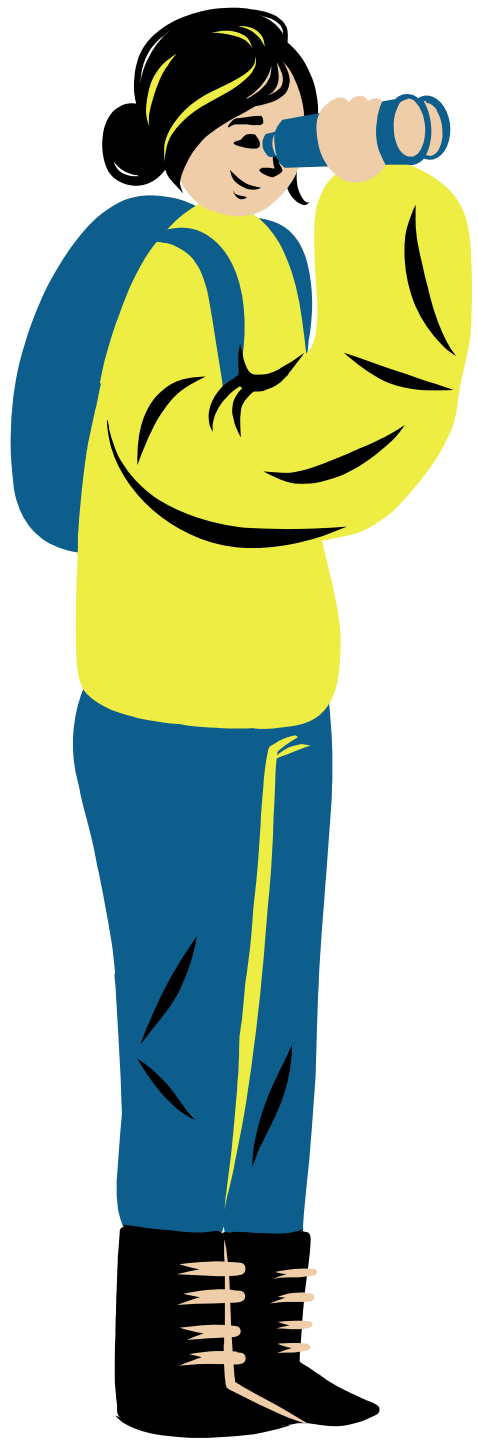
TERCER PROYECTO DE PROGRAMACIÓN

ELISA GARCÍA ZAFRA, 194992
LUCÍA DE MARÍA VARELA GARCÍA, 195239
EMILIO RAMÍREZ GONZÁLEZ 197160
JOSÉ SEBASTIÁN CÓRDOBA QUINTANA 201739



ITAM

OBJETIVOS



- 1. Diferenciar los tipos de algoritmos de búsqueda**
- 2. Identificar las diferencias entre ellos**
- 3. Definir que algoritmo es mejor dependiendo del contexto**

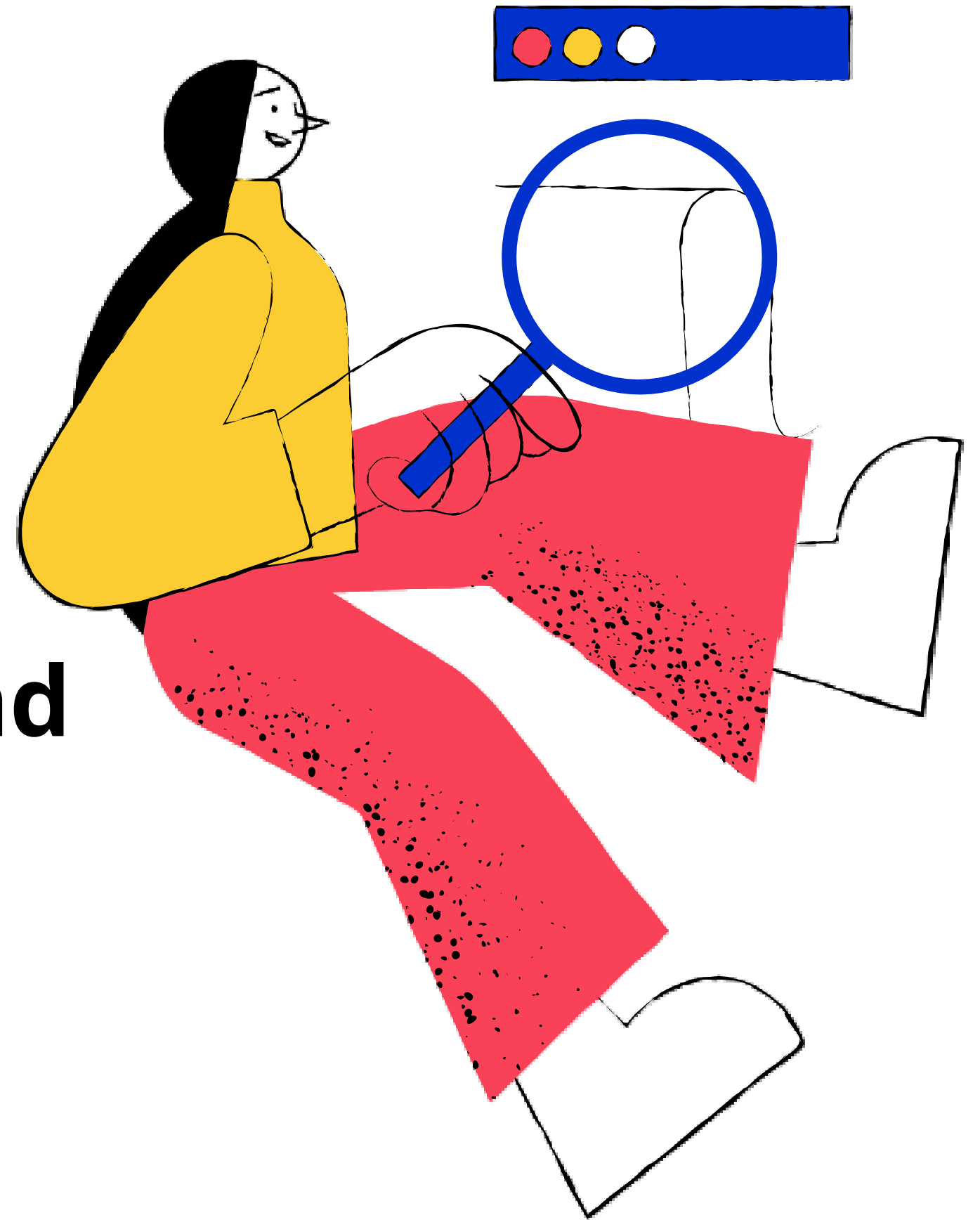
TEORÍA

A*

BFS: Búsqueda en amplitud

DFS: Búsqueda en profundidad

Algoritmo evolutivo



APLICACIÓN

Sabemos que los algoritmos evolutivos funcionan para:

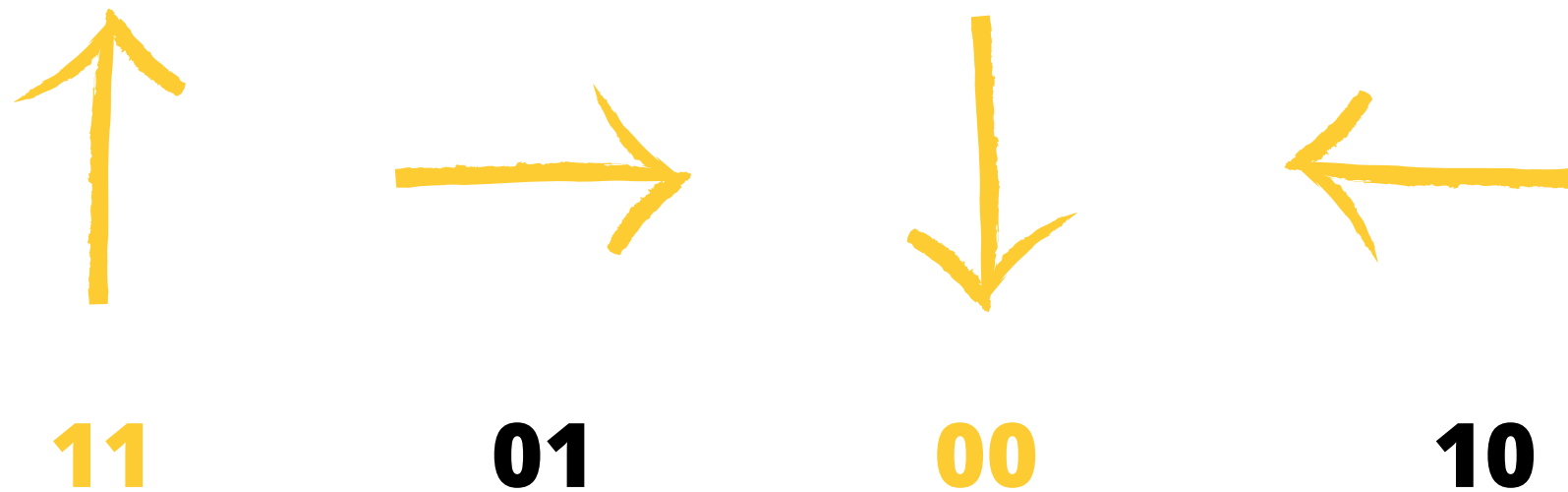
- **Optimizar tareas**
- **Encontrar errores en tareas**
- **Aprendizaje de comportamiento**
- **Sistemas del sector financiero**
- **Automatizaciones**



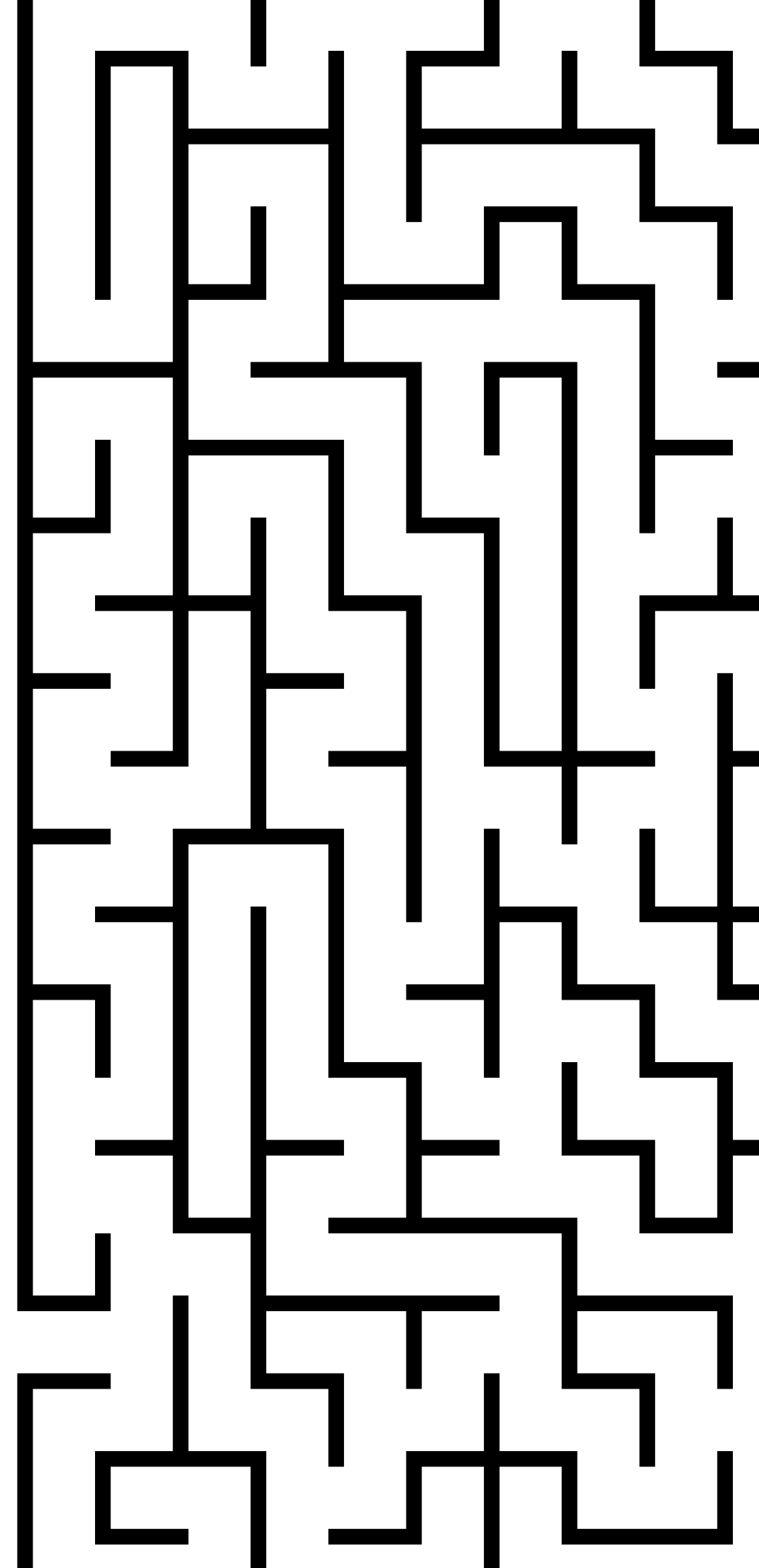
¿Se puede encontrar un camino con
algoritmos evolutivos?

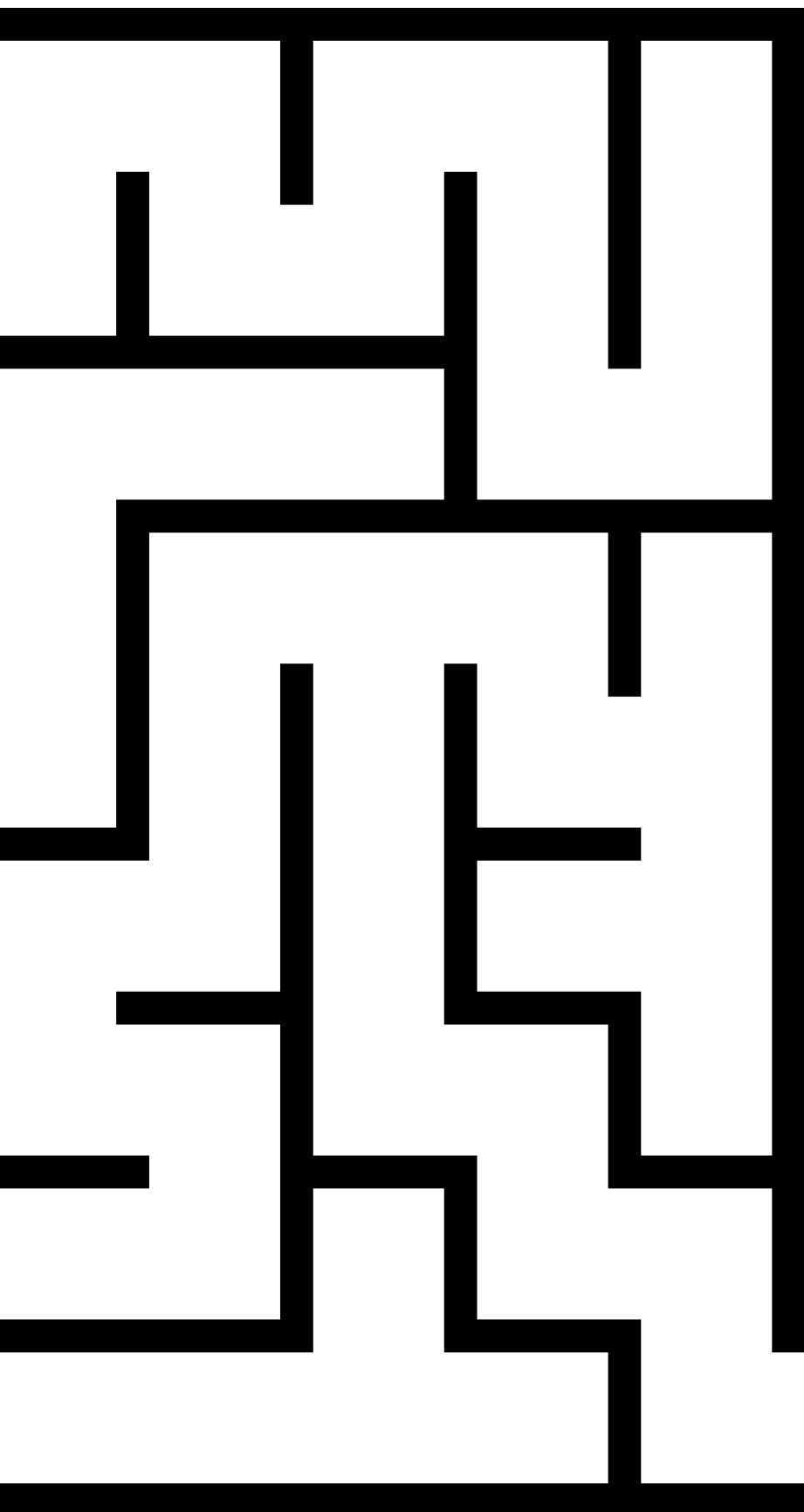
PROCEDIMIENTO

1. Representamos las soluciones como movimientos en el laberinto y los representamos con código binario



3. La solución (ADN) es la serie de movimientos convertidos en un **string de binarios**





4. Seguimos los movimientos y los evaluamos con una función heurística (distancia del último movimiento al objetivo).

5. Tomamos en cuenta

- **Si choca con una pared, se descarta la solución**
- **Las soluciones que repiten movimientos varias veces se dejan a un lado**

*** La función heurística:**

- **distancia(posición final, objetivo)**
- **Entre menor sea el valor, mejor**

Selección natural

De toda la población se elegirá el que salga mejor en su función heurística y todos los demás en la población se volverán iguales. Después cada uno será candidato a mutar.

Mutación

Se elegirá un miembro de la población aleatoriamente y habrá una probabilidad del 20% de que mute (Cambiar un de sus movimientos):

ADN inicial -> '011010100111'

ADN después de mutar -> '011010100101'

CÓDIGO

```
// Inicializar la población del algoritmo genético con secuencias de ADN aleatorias
for (let i = 0; i < pobSize; i++) {
  let dna = randomDNA(cromosomas);
  genetic_dna.push(dna);
  let path = show_genetic(dna, x, y);
  genetic_pob.push(path);
  let last = path[path.length - 1];

  // Calcular la aptitud de cada secuencia de ADN en función de la distancia desde el nodo final
  genetic_fitness.push(distance_last(last, endNode));

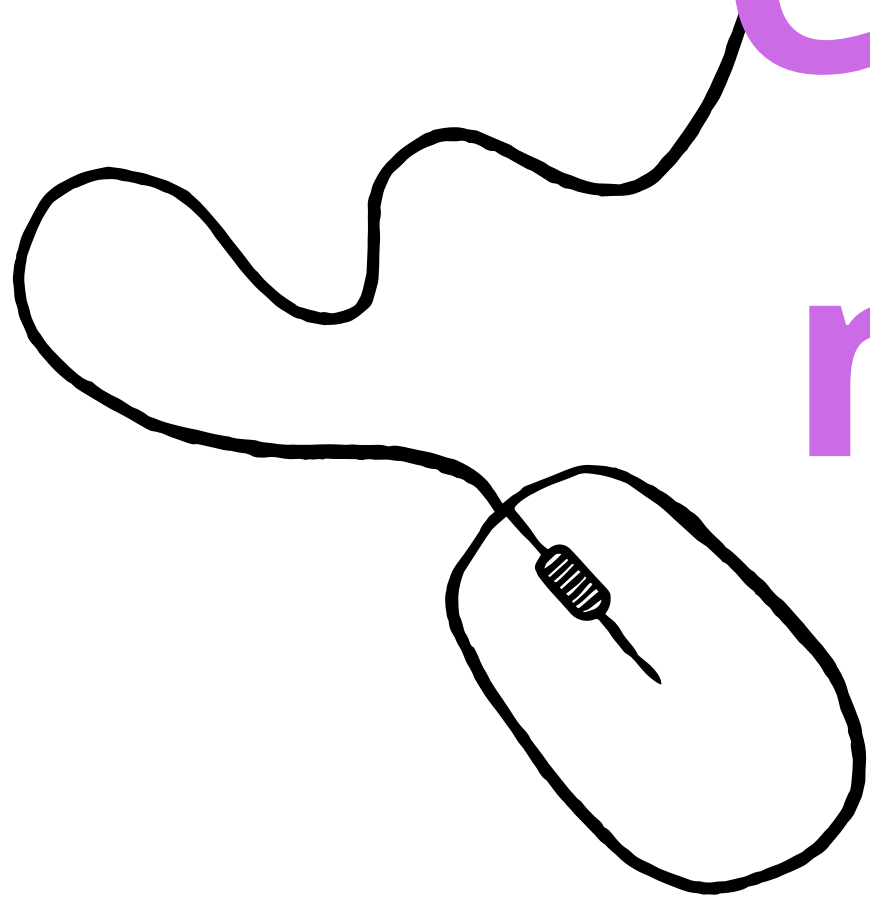
  // Colorea el último nodo del camino en rojo
  last.color = color(255, 0, 0);
}
```



```
function generate() {  
  // Restablecer los colores de todos los nodos en la población  
  for (let i = 0; i < genetic_pob.length; i++) {  
    for (let j = 0; j < genetic_pob[i].length; j++) {  
      genetic_pob[i][j].color = color(255, 255, 255);  
    }  
  }  
  
  // Seleccione la secuencia de ADN con la aptitud (distancia) más baja como padre  
  bestDNA = indexOfMin(genetic_fitness);  
  
  // Generar una nueva población mediante mutación en la secuencia de ADN original  
  for (let i = 0; i < genetic_pob.length; i++) {  
    newDNA = genetic_dna[bestDNA];  
    newDNA = mutacion(newDNA);  
    genetic_dna[i] = newDNA;  
    let path = show_genetic(newDNA, x, y);  
    genetic_pob[i] = path;  
    let last = path[path.length - 1];  
  
    // Recalcular la aptitud de cada secuencia de ADN en función de la distancia desde el nodo final  
    genetic_fitness[i] = distance_last(last, endNode);  
  
    // Colorea el último nodo del camino en rojo  
    last.color = color(255, 0, 0);  
  }  
}
```

```
function mutacion(adn) {  
  // Convertir la cadena a un array para poder modificarla  
  let adnArray = adn.split("");  
  
  // Calcular la probabilidad de mutación (10%)  
  const probabilidadMutacion = 0.2;  
  
  // Seleccionar un índice aleatorio del array de ADN  
  const indiceMutacion = Math.floor(Math.random() * adnArray.length);  
  
  // Si se cumple la probabilidad de mutación, cambiar el caracter en el índice seleccionado  
  if (Math.random() < probabilidadMutacion) {  
    // Si el caracter es 0, cambiarlo a 1, y viceversa  
    if (adnArray[indiceMutacion] === "0") {  
      | adnArray[indiceMutacion] = "1";  
    } else {  
      | adnArray[indiceMutacion] = "0";  
    }  
  }  
  
  // Unir el array mutado en una nueva cadena y devolverla  
  return adnArray.join("");  
}
```

Cosas extra que necesitamos...

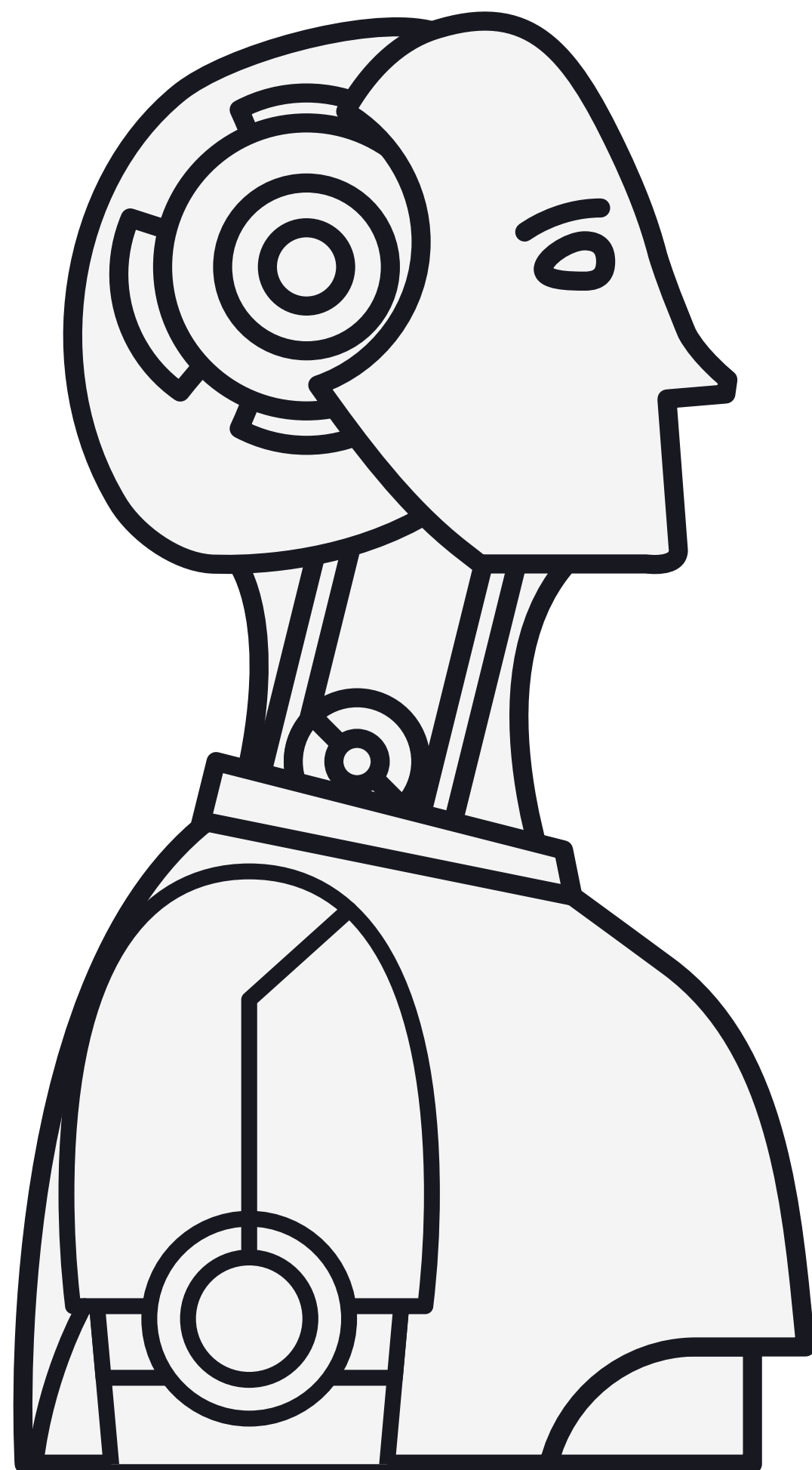


Interfaz gráfica en JavaScript

Una interfaz describe la estructura del objeto, en otras palabras, todas las propiedades y métodos de un objeto.

Servidores locales

Es un servidor de nuestra propia máquina que corremos con el fin de trabajar offline y online. Es una dirección IP que se utiliza para el desarrollo web.



CONCLUSIONES

- Este proyecto fue una excelente oportunidad para reforzar lo aprendido en clase y aprender más a fondo, igualmente, aprendimos a **implementar los algoritmos para usarlos en un programa**
- Por medio de un trabajo extenso de investigación, aprendimos nuevas cosas como: **implementar una interfaz gráfica en JavaScript y utilizar un servidor local**

- **Esta investigación nos permitió distinguir entre distintos tipos de algoritmos de búsqueda y entender las diferencias entre ellos. Además, se determinó que la elección del algoritmo más adecuado depende del contexto y de los requisitos específicos de cada problema.**
- **Nos ayudo a visualizar la implementacion de un algoritmo evolutivo para encontrar un objetivo y ver que en cada generación va mejorando.**
- **Estos hallazgos pueden servir como base para futuras investigaciones y aplicaciones de algoritmos de búsqueda en problemas similares.**

REFERENCIAS



(2022). Retrieved from <https://enzyme.biz/blog/algoritmos-geneticos-y-sus-aplicaciones-para-soluciones>

(N.d.-a). Retrieved from <https://desarrolloweb.com/articulos/definicion-interfaces-typescript.html>

(N.d.-a). Retrieved from <http://scielo.sld.cu/scielo.php>

(N.d.). Retrieved from <https://www.cognizant.com/es/es/glossary/evolutionary-algorithm#:~:text=Un%20algoritmo%20evolutivo%20es%20una,la%20reproducci%C3%B3n%20mutaci%C3%B3n%20y%20recombinaci%C3%B3n.>

The Black Box Lab Agencia de desarrollo de negocio, & negocio, A. de desarrollo de. (2020). Retrieved from <https://theblackboxlab.com/2020/06/22/que-son-los-algoritmos-evolutivos-y-para-que-se-usan/>