

**Instituto Nacional de Aprendizaje**

**Título: PROYECTO LOGICA**

**Curso: Programador de aplicaciones informáticas**

**Estudiante:** **Alejandro Quintero Zeledon**

**Año: 2021**

**Profesor: Luis Alonso Bogantes Rodríguez**

**Fecha de entrega: lunes 1° de marzo (Primera parte)**

**Introducción:**

Vamos a resolver problemas relacionados con la lógica, buscando problemas al azar en los cuales vamos a elegir dos soluciones posibles y que escogeremos la solución más eficaz y si es posible la mas eficiente, ya que a la hora de buscar un problema a veces se dificulta encontrar al más eficiente, y a veces pensamos que la respuesta más corta es la mejor y creemos que esa es la más eficiente y fácil de procesar, pero mientras se busca la solución surgen más y mas preguntas a la hora de resolverlo.

También crearemos algoritmos y diagramas de flujos en los cuales pondremos a pruebas los dos problemas escogidos con anterioridad a ver si fueron las mejores ideas, y al final haremos una conclusión del proyecto y que enseñanza nos dejó.

**Objetivos:**

Resolver los problemas de lógica

Realizar algoritmos y diagramas de flujos que resulten correctos

Buscar información y conocer del tema

**Problemas:**

**Problema #1:**  LOS PERROS Y EL ZORRO

**Historia:**

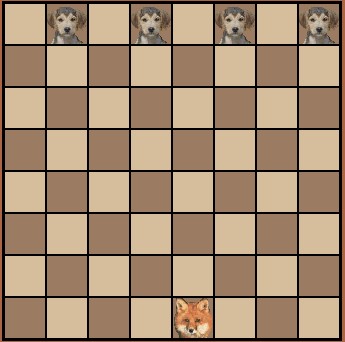
Se trata de un juego de origen vikingo, se han encontrado muchas variantes de él.

**Presentación:**

Juego de bloqueo y escape.

Juego para dos jugadores con fuerzas desiguales.

Se necesita el tablero de juego (matriz de 8x8), una ficha para identificar el zorro y 4 fichas diferentes que van a representar a los perros.

**Reglas del juego:** 

* Se colocan las fichas en la posición inicial, según el dibujo.
* Los jugadores mueven por turno siempre en diagonal y sobre un tipo de color de casillas (blancas o negras).
* El que tiene los perros comienza la partida moviendo una de sus piezas en diagonal y hacia adelante. Los perros no pueden moverse hacia atrás.
* El zorro puede desplazarse en diagonal, en todas las direcciones, a una casilla contigua vacía.
* No se puede comer ni saltar por encima de otra ficha.

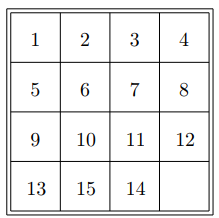
**Objetivos:**

* De los perros: bloquear y arrinconar al zorro.
* Del zorro: abrirse paso y escapar de los perros.

**Problema #2:**  ROMPECABEZAS NÚMERICO

Es un conocido juego que consiste en una cuadrícula de 4x4 en donde se colocan los números del 1 al 15, dejando una casilla vacía, que puede ser ocupada por las piezas adyacentes.

Deberá desordenar las 15 piezas con número de forma aleatoria y volver a reordenar los números con el menor número de movimientos posible.



**Alternativas de solución #1:**

**Planteamiento:** Intentar resolver el problema de como se puede llevar a la victoria a cada un jugador

**Identificación del problema:** Dependiendo de cual equipo este puede ser el bloqueo del zorro o puede ser traspasar detrás de los lobos

**Datos de entrada:** El inicio de cada jugador que es estar en algún lateral, después serian los movimientos de cada jugador

**Datos de salida:** El trapazo detrás de los perros que daría a una victoria, o el encerrar al zorro que daría victoria para los perros

**Observaciones (limitaciones, excepciones, suposiciones):**

Único movimiento es en diagonal

Siempre el primer movimiento es de los perros

Se debe jugar siempre en las mismas casillas (mismos colores)

**Planteamiento de alternativas:**

**Planteamiento #1:** Puede crear un cuadro 8x8 en el que los perros deben tener siempre un solo movimiento y en diagonal también se debe tomar en cuenta que no se pase del tablero o que no pueda saltarse ninguna pieza sobre otra, también se debe tomar en cuenta que el zorro se puede mover en cualquier movimiento en forma diagonal también que no puede saltar ni un solo objeto, se debe tomar en cuenta cuando cruza el zorro hacia el otro lado y cuando es encerrado el zorro, para averiguar la victoria de uno de los equipos, eso puede hacerse comparando cada movimiento en el tablero, así sabremos cuando el zorro no tiene opción de movimientos o ya paso al otro lado

**Planteamiento #2:** La segunda opción que se me ocurre seria moverse hacia donde hace el primer movimiento el perro, pensando que uno es el zorro, así podría saber que movimiento esta intentando hacer el perro e intentar pasar por ellos, si uno fuera el perro lo mejor que puede hacer es llevar a todos los perros de la forma mas recta posible, a todos al mismo tiempo

**Elección de Alternativa**

**Alternativa seleccionada: #1**

**Justificación:** La numero 1 es con la que es mas posible conseguir la victoria, si se lleva por la esquina sin perder muchos movimientos siendo el zorro se lograría ganar, y siento que es la que tiene la mejor eficacia

**Alternativas de solución #2:**

**Planteamiento:** Intentar resolver el rompecabezas con el número menor al inicio

**Identificación del problema:** El desorden de los datos y cantidad de movimientos a realizar deben ser cortos

**Datos de entrada:** Los números y movimientos

**Datos de salida:** Datos ordenados de mayor a menor

**Observaciones (limitaciones, excepciones, suposiciones):**

El juego puede ser de diferentes formas, o mas bien la cuadricula puede ser de 5x5 y con 24 piezas para hacer los movimientos

**Planteamiento de alternativas:**

**Planteamiento #1:** Intentar empezar con el numero menor moviendo el “hueco” o “vacío” hasta la esquina superior luego ir moviendo poco a poco hasta armarlo de tras hacia adelante, rotando movimientos hasta llegar a la respuesta

**Planteamiento #2** se puede verificar donde se encuentran los números, después hacer los movimientos correspondientes en el cual vamos revisando cada movimiento como va quedando el cuadro, después verificamos que el cuadro quedo ordenado y así ganaría, la mejor opción seria estar verificando en cada movimiento que el cuadro este ordenado, después solo los movimientos se pueden hacer y ya se encuentra la respuesta

**Elección de Alternativa**

**Alternativa seleccionada:** #2

**Justificación:** La segunda opción es la que parece ser la mas eficiente entre las dos ideas, ya que minimiza los movimientos hechos por el usuario así llega a una respuesta más rápida de lo que se imaginaria

**Algoritmos**

**Problema #1**

Crear matriz 8x8, creamos las piezas que serian perros y zorro

Llenamos la matriz de espacios en blanco, luego le pedimos al usuario en que color desea usar para jugar, si negros o blancos, los blancos se encuentran en la primera posición 0,0 entonces nos guiamos en esa posición, luego colocaríamos en esa posición 0,0 después sumamos 2 y colocamos en la posición 0,2 y así sucesivamente hasta encontrar los blancos, en los negros lo colocamos desde la posición 0,1 y sumamos 2 hasta encontrar todos los blancos, en esas posiciones colocaríamos a los perros, y el zorro debemos decirle donde desea colocarlo, si escogieron negros empezamos mostrándole en que posiciones desea colocarlos empezando de la opción 7,1 y así sucesivamente, igualmente si fuera color blanco le mostramos la opción 7,0 y las que se puedan colocar una vez colocado las posiciones de los usuarios, empezamos con los perros, ósea que el usuario debe escoger entre alguno de los perros, que serán 4 perros llamados 1 2 3 4 y así sucesivamente entonces colocamos un \*según\* en el que si escoge uno de esos perros pondremos los movimientos posibles para ese perro, sin hacer que se salga, ósea, en el perro 1 estando en la posición 0,0 tendrá un solo movimiento en el que se puede mover a la posición 1,1, entonces sucede que siempre hay un iterador que se llama \*i\* ese valdrá el primer 1, ese siempre se sumara una vez cada vez que haga movimiento ese perro, entonces que sucede? Simple, en el siguiente movimiento el primer 1 valdrá 2, seria mas bien 2,1 pero también necesitamos que el segundo uno haga los movimientos hacia los lados porque sino siempre ira hacia abajo, pero aquí es a lo que el usuario decida, colocaremos donde se encuentre ese perro que escogió el 1 o 2 que serán los posibles movimientos que podrá hacer el perro, y para que surta efecto siempre al \*i\* se le sumara 1 para que vaya bajando, y al segundo número, se le puede sumar o restar dependiendo lo que escoja el usuario, si el usuario escoge el número 2 que es a la izquierda entonces el primer numero valdrá 1 y el segundo se le restara 1 si desea ir a la izquierda y se le sumara 1 si desea ir a la derecha, y así sucesivamente, igualmente con los demás perros, y para comprobar que no se pase, se puede hacer un \*si\* en el que evaluaremos si es \*i\*=0 no seguirá restando y caso contrario que \*i\*=7 no seguirá sumando, todo esto para el movimiento de los perros… pero que sucede si hay un perro donde desea mover, primero debemos evaluar que no haya nada, lo hacemos un \*si\* en el cual evaluamos lo siguiente, \*si\* en la posición que deseo hacer el movimiento vale algo diferente a “Espacio” que eso es lo que vale el total del tablero porque anteriormente lo habíamos llenado de vacíos espacios vacíos, si hay algo diferente a “espacio” significa que hay algo ahí, y no podrá hacer un movimiento hacia esa posición, y así sucesivamente en todos los perros y en este caso igualmente el zorro, no podrá hacer ningún movimiento hacia donde haya algo

Luego con los movimientos del zorro seria lo mismo que el perro, solo que si desea ir hacia atrás se debe restar esta vez el primer numero ósea el \*1\*,0 si desea ir a una posición más abajo, también evaluamos que si llega a la posición 7,X significa que ya está abajo ósea que no se puede restar el primer número, y así sucesivamente, después tendremos que evaluar quien gano y quien perdió.

Si el primer índice del zorro vale mas que cualquiera de los perros entonces el zorro logro ganar, ejemplo, si zorro vale 4,3 y el zorro con la posición mas arriba vale 3,1 significa que el zorro gano, también puede darse el caso de que llegue al final sin que un perro se haya movido así que podemos hacer que si llegase a la fila 0 gano, o si sucede que el ultimo perro valga lo mismo que el zorro significa que gano ya que el siguiente movimiento será de victoria, caso de que el perro gane, se puede comparar los movimientos del zorro, si zorro no tuviera movimientos creamos una variable que sea verdadero o falso, una vez que entre en el anterior si que mostraba los movimientos del zorro esta vez si no entrara en ninguna de esas esa variable se vuelve falso, y el zorro perderá

A la hora de rotar movimientos quien va y quien sigue, se hace con un falso y verdadero, el primer movimiento será verdadero para cualquiera de los dos, el segundo movimiento solo será verdadero para el segundo jugar y se vuelve falso para el primer jugador y una vez haga el segundo jugador su movimiento el primer se vuelve verdadero y así sucesivamente

**Problema #2**

Primero creamos una matriz 4x4

Llenamos esa matriz con números al azar del 1 al 15 sin que se repitan los números, entonces cada vez que se repita no lo metemos a la matriz y colocamos el siguiente, también se pueden crear varios números al azar luego colocarlos, pero eso siempre se debe tomar en cuenta que no se repitan los números se puede ir verificando sucesivamente hasta que no haya números repetidos, se puede hacer en un mientras, hasta que se llene la matriz, el 4,4 será la posición en la que vale “espacio” ósea que nada, una vez lleno la matriz, se empiezan a crear los movimientos, como el ultimo dato de la matriz esta en blanco, es el que usaremos para mover todos los datos, hay varias formas para rotar los números , primero le decimos al usuario digite A si desea subir el espacio en blanco I por si desea ir a la izquierda D por si desea ir a la derecha AB por si desea ir abajo así una vez eso suceda hacemos el movimiento, dado caso que fuera hacia arriba el numero que se encuentra arriba lo guardamos y lo ponemos en blanco y el numero guardado lo ponemos donde anteriormente estaba el blanco, para eso debimos haber guardado en que posición estaba el blanco, así podemos hacer un movimiento, pero que sucede si no hubiera nada arriba, por eso para eso primero debemos hacer una comparación, si el primer iterador fuera -1 significa que esta mas arriba de lo deseado igualmente con la parte de abajo si fuera 5 se paso y así sucesivamente con los lados, si dado caso que fuera así simplemente no hará nada ni un solo movimiento, ya después con cada movimiento que se haga se guarda el dato en una variable que ira guardando que mostrara cuanto tiempo le costo terminar el cuadro, también con cada movimiento que se haga se va comparando que el cuadro este terminado, ósea de menor a mayor, una vez compruebe eso, dirá si gano o no, una vez ganado se reiniciar el juego

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lista de Rasgos Observables** | | | **Criterios** | | |
| **CC**  **2** | **IC**  **1** | **II**  **0** |
| **En cuanto a la estructura** | | |  |  |  |
| 1. La portada tiene todos los datos requeridos | | | 2 |  |  |
| 1. La introducción es acorde al tema asignado | | | 2 |  |  |
| 1. Define los objetivos del proyecto | | |  | 1 |  |
| 1. Describe el problema empresarial | | | 2 |  |  |
| 1. Respecta el formato solicitado | | | 2 |  |  |
| 1. Incluye la bibliografía | | | 2 |  |  |
|  | **En cuanto al desarrollo.** | |  |  |  |
| 1. Analiza el problema dado según la metodología | | |  | 1 |  |
| 1. Identifica el problema mediante la metodología | | |  | 1 |  |
| 1. Brinda soluciones lógicas al problema | | |  | 1 |  |
| 1. La solución es pertinente para el problema planteado según   requerimientos de la clientela | | |  | 1 |  |
| 1. La solución planteada no presenta errores de lógica | | |  | 1 |  |
| 1. Determina los datos de entrada para el proceso 1. | | |  | 1 |  |
| 1. Los datos de entrada se ajustan a lo planteado para el proceso 1. | | |  | 1 |  |
| 1. La descripción del proceso 1 es completo. | | |  | 1 |  |
| 1. El proceso 1 se ajusta a la necesidad del caso. | | |  | 1 |  |
| 1. Determina los datos de salida para el proceso 1. | | |  | 1 |  |
| 1. Los datos de salida cumplen con lo descrito en el proceso 1. | | |  | 1 |  |
| 1. Determina los datos de entrada para el proceso 2. | | |  | 1 |  |
| 1. Los datos de entrada se ajustan a lo planteado para el proceso 2. | | |  | 1 |  |
| 1. La descripción del proceso 2 es completo. | | |  | 1 |  |
| 1. El proceso 2 se ajusta a la necesidad del caso. | | |  | 1 |  |
| 1. Determina los datos de salida para el proceso 2. | | |  | 1 |  |
| 1. Los datos de salida cumplen con lo descrito en el proceso 2. | | |  | 1 |  |
| 1. Toma en cuenta las limitaciones y excepciones de los procesos | | |  | 1 |  |
| **Total** | | | **48 puntos** | | |
| **Puntos Obtenidos 30** | | **Calificación 63** | **Porcentaje**  **Obtenido:** | | |

|  |  |  |
| --- | --- | --- |
| **I Fase del Proyecto Final** | | |
|  | Plantilla | 63 |
|  | Algoritmo 1 | 93 |
|  | Algoritmo 2 | 87 |
|  | **Nota** | **81** |