

**UNIVERSIDAD CATOLICA
DE COSTA RICA**

BACHILLERATO EN INGENIERÍA DE SISTEMAS

IS083 INTELIGENCIA ARTIFICIAL

PROYECTO TORRES DE HANOI

PROFESOR: ANDRÉS JIMENES L.

ALUMNO: DAVID GUTIÉRREZ S.

**SEDE CIUDAD QUESADA
NOVIEMBRE, 2016**

Tabla de contenido

Resumen Ejecutivo	3
Descripción	4
Abstract	5
Objetivo General	7
Objetivos Específico	8
Presentación y Análisis del Problema	15
¿Qué hay que resolver?	15
¿Qué hay que resolver?	15
Análisis del problema.....	16
Análisis de la Solución.....	17
Resultados Obtenidos.....	18
Conclusión	19
Recomendaciones.....	20
Cronograma de Trabajo y Bitácora	21
Bibliografía	23
Anexos	24

Resumen Ejecutivo

Las Torres de Hanói son un rompecabezas o juego matemático inventado en 1883 por el matemático francés Édouard Lucas. Este juego de mesa solitario se trata de un juego con un número de discos de radio creciente que se apilan insertándose en una de las tres estacas de un tablero. El objetivo del juego es crear la pila en otra de las estacas siguiendo ciertas reglas. El problema es muy conocido en la ciencia de la computación y aparece en muchos libros de texto como introducción a la teoría de algoritmos.

En la aplicación vamos a poder ver el problema ya implementado en un sistema informático orientado a juegos en un lenguaje muy conocido, se desarrolló en Visual Studio con el lenguaje C#, y vamos a interactuar con el sistema de forma que podemos tomar decisiones aleatorias para las resoluciones de los problemas que vamos a ir encontrando, sin embargo cabe destacar que ya hay una solución óptima pero en el modo de usuario tendremos la oportunidad de realizarlos nosotros mismos con nuestra propio razonamiento lógico.

El problema nos presenta una temática de matemáticas donde tenemos tres varillas verticales sujetas en un soporte. En una de ellas se encuentran insertados un número arbitrario de discos planos, todos de distinto diámetro, de mayor a menor tamaño. Las otras dos varillas están vacías en su posición inicial.

El juego consiste en pasar todos los discos de la varilla ocupada a una de las otras, también de mayor a menor tamaño, calculando el número de movimientos necesario. Además, existen normas a seguir para su solución pero las veremos más adelante.

Descripción

La presentación de este trabajo, es un medio de facilitamiento académico y práctico para el estudiante quien desea conocer y leer de este trabajo escrito práctico orientado a Inteligencia Artificial.

Tomado en cuenta el desarrollo de la aplicación solicitada por el profesor del curso de Inteligencia Artificial, donde se ejecuta la solicitud de una aplicación orientada a juegos basado en el histórico juego de Torres de Hanói, donde se implementara las normas básicas y algunas restricciones que son necesarias para el trabajo ya que son las solicitudes a realizar, dejando a criterio del estudiante la parte de escogencia del funcionamiento de la misma. Para fin del propósito de aprendizaje por medio de la investigación y desarrollo del sistema o aplicación indicada por este trabajo.

Describiendo la función de esta aplicación, podremos decir que se ha realizado un sistema o aplicación que tiene el fin de realizar las interacciones del popular juego. Dicho esto, sabemos que en este trabajo escrito estaremos aclarando y explicando el proceso de creación y pruebas de dicha aplicación desarrollada bajo el lenguaje C# en el entorno de Visual Studio 2015.

Abstract

Using recursion often involves a key insight that makes everything simpler. Often the insight is determining what data exactly we are recursing on - we ask, what is the essential feature of the problem that should change as we call ourselves?

In our Towers of Hanoi solution, we recurse on the largest disk to be moved. That is, we will write a recursive function that takes as a parameter the disk that is the largest disk in the tower we want to move. Our function will also take three parameters indicating from which tower the disk should be moved, to which tower it should go, and the other tower, which we can use temporarily to make this happen.

In this problem, he will be working on a famous mathematical puzzle called The Tower of Hanoi. There are three disks, and in the first tower there is a stack of disks of different sizes arranged in descending order. The object of the game is to move all the disks to another tower. However, you can only move one disk at a time and you can not place a disk on top of a smaller disk.

When they have managed to move all the discs to another needle their work will be finished, and the tower and the temple will collapse, and with a great thunder, the world will vanish. The simplified version sold in France consisted of eight wooden discs.

In fact, the Tower of Hanoi and Indian legend had been invented by the French mathematician Édouard Lucas. His compatriot, the writer Henri de Parville enlarged and decorated the legend shortly after. Although the challenge posed is relatively simple, Lucas' idea has proved to be one of the most fruitful in the history of recreational mathematics.

At the top level, we want to move the whole tower, so we want to move the disks and always the smaller ones from tower A to tower B and finally tower C. We can use this.

What is the smallest number of moves needed to complete the Tower of Hanoi game with:

- I. One disc?
- II. Two discs?
- III. Three discs?
- IV. Four discs?

Do you notice anything interesting about the way the number of moves increases?

Can you explain any patterns you find?

In the application we will be able to see the problem already implemented in a game-oriented computer system in a well-known language, developed in Visual Studio with the C # language, and we will interact with the system so that we can make random decisions for the Resolutions of the problems that we are going to find, but it is important to emphasize that there is an optimal solution, but in the user mode we will have the opportunity to realize them ourselves with our own logical reasoning.

The problem presents a mathematical theme where we have three vertical rods attached to a support. In one of them are inserted an arbitrary number of flat disks, all of different diameter, from greater to smaller size. The other two rods are empty in their initial position.

The norms to follow are:

- I. You can only move one disc at a time.
- II. A larger disk can not rest on a smaller disk than itself.
- III. You can only move the disk on top of each rod.

There are several ways to make the final solution, all of them following different strategies.

Objetivo General

- I. Desarrollar un sistema o aplicación orientado a juegos basado en las normas y estrategias del popular juego llamado Torres de Hanói, donde se implemente los algoritmos necesarios para las soluciones del problema a resolver.

Objetivos Específico

- I. Desarrollar un sistema o aplicación orientado a juegos orientado al juego Torres de Hanói, utilizando la información brindada por el profesor del curso de Inteligencia Artificial.
- II. Desarrollar el sistema o aplicación en el lenguaje C# bajo el entorno de Visual Studio 2015.
- III. Aplicar conocimientos y estructuras orientadas a Inteligencia Artificial.
- IV. Sub-versionar en la plataforma de GitHub dicha aplicación.
- V. Analizar la estructura y documentar los resultados obtenidos.
- VI. Exponer los resultados del proyecto.

Marco Teórico

Las torres de Hanói es un juego algorítmico muy bueno, donde se puede ejercitar la memoria, y nos ayuda a tener respuestas más rápidamente, al cual podemos resolverlo con ayuda matemática, y con el método divide y vencerás. El cual consiste en resolver un problema difícil, dividiéndolo en partes más simples tantas veces como sea necesario, hasta obtener el resultado deseado.

¿Que son las torres de Hanói?

El original juego de las torres de Hanói, es un juego de rompecabezas compuesto por una tabla de madera, tres estacas y ocho discos de tamaños distintos cada uno más pequeño que el otro, también de madera. Los cuales pueden ser de diversos colores, y consiste en insertar los discos en una estaca de forma creciente, y crear una nueva fila en otra estaca pero siguiendo ciertas reglas. En la actualidad hay de diversos materiales, no solo de madera. Y también los hay en páginas de internet para disfrutarlos digitalmente.

Reseña Histórica.

El juego fue creado por el matemático francés Édouard Lucas. En 1883. Hay diversas historias de cómo fue tomada la idea para ser creado como por ejemplo se dice que hace mucho tiempo en un templo de Benarés en la india existía una cúpula que señalaba el centro del mundo y estaba compuesta por tres agujas de diamantes, y un día un rey mando a poner 64 discos de oro siendo ordenados por tamaños de mayor a menor y que los sacerdotes al mover los discos cada día debían hacerlo siguiendo ciertas reglas, iguales a las del juego actual. Y que al pasar el tiempo el templo dejo de existir pero que el juego perduro. Pero en realidad todo ese cuento fue inventado por el mismo francés creador del juego. Llamamos Torre de Hanoi (clásica) unos discos de radio creciente apilados en una de las tres varillas de un tablero. El objetivo es mover la pila en otra de las varillas en el menor número posible de pasos, siguiendo las reglas detalladas a continuación:

- I. Sólo se puede mover un disco a la vez;
- II. No puede haber nunca un disco de mayor tamaño o sobre uno de tamaño menor;

1. Véase por ejemplo C.Madonna, El Teorema de Pitagóras–Papus, Didácticas Específicas, Vol.12 (2015).
2. En el 1995 el matemático ingles Andrew Wiles publica una demostración de la no existencia de ternas (a, b, c) con estas propiedades.
3. El problema clásico fue propuesto en el libro “De Subtilitate publicado” por el matemático italiano Girolamo Cardano (1501–1576), y luego fue introducido en el 1883 por el matemático francés François Edouard Anatole Lucas como juego.

III. Sólo se puede desplazar el disco que se encuentre arriba en cada varilla.

Por lo tanto el problema es el siguiente:

Pregunta 1.

Hallar el menor número de movimientos para mover los 3 discos de una varilla a otra con las reglas de arriba.

Es fácil convencerse que la respuesta correcta en este es 7, y gráficamente se podría describir como en la Fig. 1.

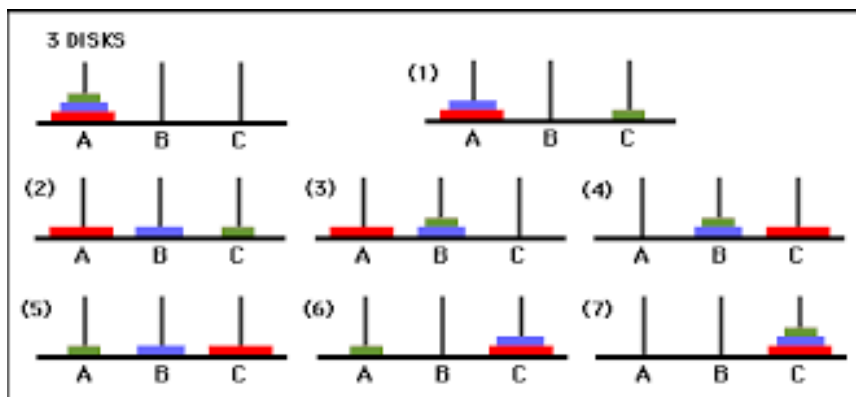


Figura 1. Secuencia de pasos para 3 discos: 7 pasos.

Este problema de muy fácil resolución ha sido objeto de numerosos estudios y diferentes aplicaciones en muchas otras ramas de conocimiento: en medicina, en psicología, en pedagogía, entre otras aplicaciones que no vamos aquí a recordar visto que nuestro interés principal está centrado en un problema mucho más general, que tampoco en matemáticas ha sido resuelto hasta hoy, y de sus aplicaciones al ámbito de la formación y del desarrollo de las capacidades de razonamiento y pensamiento lógico–Matemático.

La pregunta natural que nos podemos poner es la siguiente:

Pregunta 2.

¿Y si en lugar de tres discos hay que mover $n - 1$ discos?

- Esta actividad ofrecida a alumnos de Educación Infantil no ofrece ninguna dificultad y se puede adaptar en lugar de utilizar discos utilizando fichas geométricas de diferentes tamaños y formas, por ejemplo polígonos de 3, 4 o 5 lados. Es una actividad que no tiene un tiempo definido para ser terminada y el objetivo en la escuela infantil será enfocado más al razonamiento que al resultado. El objetivo fundamental es que los mismos estudiantes lleguen a proponerse la pregunta de encontrar el número mínimo de movimientos una vez resuelto el problema preparatorio que podría formularse de la siguiente manera: "Mover los tres discos de una varilla a otra siguiendo las reglas dadas". Se puede consultar el texto citado en la nota más abajo de A. Zvonkin para ver cuál podría ser el tiempo aconsejable para desarrollar esta actividad.
- En la escuela infantil y primaria se podrá presentar la pregunta con 4 o 5 fichas en lugar de pasar directamente al caso de un número arbitrario de fichas.

6. Así se encuentra formulado el problema por ejemplo, cuando $n = 3$, en A. Zvonkin, *Math from Three to Seven. The Story of a Mathematical Circle for Preschoolers*, MSRI-AMS, 2011.

Para contestar a la pregunta 1 procedemos de la siguiente manera. Primero creamos una tabla de ayuda y vamos apuntando el número de movimientos en función del número de discos que tenemos, en algunos casos especiales, que luego nos servirán como ejemplos para encontrar una solución general para toda $n \geq 1$.

Con 1 disco está claro que el número de pasos que necesitamos es igual a 1. Para 2, y 4 discos, en las figuras Fig. 2 y Fig. 3 están representadas las secuencias de pasos a seguir, mientras que para 3 discos ya lo hemos visto en la Fig. 1.

Nº Discos	Nº Pasos
0	0
1	1
?	?
3	7
?	?

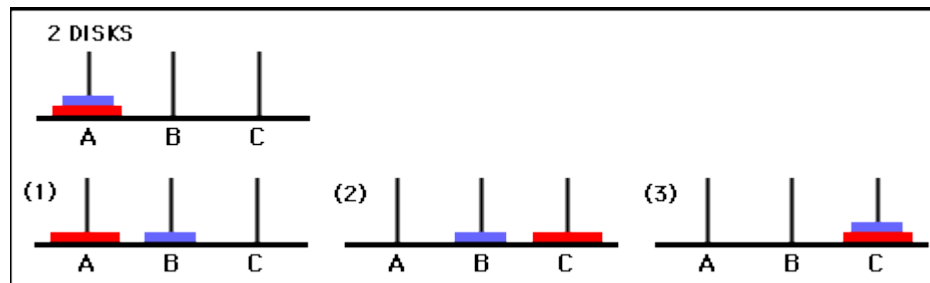


Figura 2. Secuencia de pasos con dos discos.

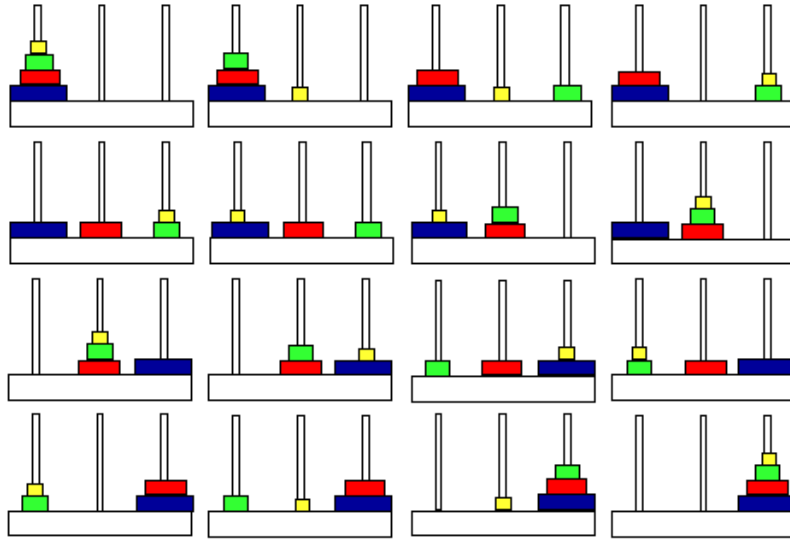


Figura 3. Secuencia de pasos con 4 discos.

Si observamos cómo hemos conseguido mover los 3 discos del primer caso nos damos cuenta que hemos tenido que mover 2 discos desde una columna a otra, luego mover el disco más grande en la posición final, y finalmente mover otra vez 2 discos en la varilla donde ya está el tercer disco. Para mover 2 discos hemos tenido que mover dos veces 1 disco de una varilla a otra más y mover el segundo disco en su posición final. Observamos también que si el número de movimientos para los 2 discos es el mínimo entonces también lo será para el caso de 3 discos.⁷ Es decir, si denotamos a $f(n)$ el número mínimo de movimientos para mover los $n - 1$ discos en la varilla inicial con las reglas dadas, se puede observar que:

$$f(1) = 0 + 1 + 0$$

$$f(2) = 1 + 1 + 1$$

$$f(3) = 3 + 1 + 3$$

$$f(4) = 7 + 1 + 7$$

....

$$f(n) = f(n-1) + 1 + f(n-1)$$

$$= 2f(n-1) + 1$$

Sustituyendo en cada ecuación los valores obtenidos en la anterior e iterando conseguimos:

$$f(n) = 2[2f(n-2) + 1] + 1 = 2(2[2f(n-3) + 1] + 1) + 1$$

$$= 2^3 f(n-3) + 4 + 2 + 1 = \dots$$

$$= 2^{n-1} f(n - (n-1)) + 1 + 2 + 2^2 + 2^3 + \dots + 2^{(n-1)-1}$$

$$= 2^{n-1} f(1) + 2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{n-2}$$

$$= 2^{n-1} + \sum_{k=0}^{n-2} 2^k = \sum_{k=0}^{n-1} 2^k = \sum_{k=1}^{n-1} 2^k + 1$$

Resumiendo con un procedimiento recursivo es fácil demostrar que:

$$f(n) = 2f(n-1) + 1 = \sum_{k=1}^{n-1} 2^k + 1, \quad f(1) = 1$$

7. En la escuela infantil y primaria se podrá resaltar el razonamiento general: quitar todos los discos exceptuado el más grande, ponerlos en una sola varilla para poder mover el disco más grande en la posición final para luego volver a poner todos los demás discos encima de éste repitiendo el procedimiento hecho antes para quitarlos de la varilla inicial.
8. Utilizando por ejemplo el Principio de Inducción Matemática, véase por ejemplo p.44: A.Barcia, P.Caressa, C.Madonna, *Matemática escolar desde un punto de vista superior, I: conjuntos y números*,
9. Ediciones UAM, 2011.

Como ya hemos observado, desde la educación infantil se puede proponer el problema de la Torre de Hanói con 3 discos y 3 varillas como actividad para el desarrollo matemático de las habilidades de razonamiento lógico matemático de los niños, siguiendo por ejemplo pautas y objetivos como hemos indicado a lo largo del texto. Hemos visto también como luego se puede pasar al caso de $n \geq 3$ discos en una torre de 3 varillas.

Por otro lado, también hemos visto como el solo pasar de 3 varillas a 4 varillas lleva a un problema que solo recientemente se ha conseguido resolver en una torre con 4 varillas, y con métodos no elementales, quedando todavía abierto el problema de establecer el número mínimo de movimientos necesarios en el caso de 5 o más varillas o también deducir una demostración elemental del caso de 4 varillas.

El lector interesado puede consultar la amplia bibliografía (más de 300 artículos citados) redactada por Paul K. Stockmeyer¹² y otras referencias en Klavžar, Milutinović, y Petrb¹³.

De toda forma, resulta claro como la Torre de Hanói (generalizada) puede ser muy útil en entornos educativos para introducir, por ejemplo:

- I. El concepto de recursividad (o bien el Principio de Inducción Matemática);
- II. Los problemas de optimizar y/o los problemas de mínimo;
- III. Como resolver problemas y al mismo tiempo formular otros problemas;
- IV. Un ejemplo de lo que puede ser la investigación matemática.

Introducción

Este proyecto ha sido desarrollado en el área de aplicaciones orientado a juegos para la interacción entre usuario y la plataforma de modo que el usuario pueda disfrutar de un momento de diversión.

Básicamente este proyecto está basado en una aplicación de juegos muy popular que simplemente posee tres torres y varios discos dependiendo de su nivel de dificultad, donde el reto a tomar es la solución del algoritmo, para pasar todos los discos de la primera torre a la tercera torre siempre usando como auxiliar la segunda torre, que podremos usar para descansar los discos temporalmente mientras resolvemos el problema.

Este proyecto fue desarrollado en el lenguaje C# con el entorno de Visual Studio 2015, y está diseñado para ser entendido por cualquier persona que se interese en leerlo y modificarlo, por lo tanto, esta aplicación es totalmente funcional, y está libre para que pueda ser analizada por cualquier grupo de estudiantes que quieran poder comprender su código y realizar sus propias modificaciones a dicha aplicación, ya que esta fue desarrollada para el área estudiantil y la comunidad virtual de internet, que quieran poner usarla para sus propios intereses de aprendizaje, entre otros.

Presentación y Análisis del Problema

¿Qué hay que resolver?

Se debe implementar el algoritmo recursivo para la solución de los problemas que se presenten a la hora de la búsqueda de una respuesta en el juego. Además, la identificación de pequeños problemas a la hora de la escogencia o desarrollo del sistema, ya que siempre surgen implicaciones a la hora de ejecutar ordenes de código en el desarrollo.

Se debe de revisar bien los algoritmos del sistema para saber si son los adecuados y tienen el mejor rendimiento que no provoque un problema en el sistema ya terminado. La utilización de estructuras de datos para facilitar el manejo de datos con mejor eficiencia y estabilizad. El lenguaje tiene sus propias aplicaciones así que debemos saber aprovecharlas al máximo para que este nos dé su mayor potencial.

La lógica de estructura debe estar basada en el código limpio y con una legible interpretación para quien dese leer el código para fines prácticos o educativos.

Se debe resolver el modo de juego solo usuario y solo demostración que no haya conflictos entre ambos y que este orientado a inteligencia artificial en el caso de las demostraciones.

¿Qué hay que resolver?

Ya que estamos iniciando, comenzaré con algo simple, la solución con cuatro discos solamente, pero el “método” es fácilmente aplicable a cualquier número de discos.

Para empezar llamaremos “Torre A” al origen en la que se encuentran inicialmente todos los discos, “Torre C” al destino al cual se desean mover los discos y “Torre B” al queda en el medio que servirá como auxiliar. También llamaremos a los discos, del menor al mayor (de arriba a abajo), según los números del uno al cuatro, siendo el que está más abajo (y el más difícil de mover) el número cuatro.

Bien, dados cuatro discos en Torre A, cero en Torre B y cero en Torre C, considérese que:

- Para mover legalmente el disco 4 hasta Torre C, será necesario mover los primeros tres hasta Torre B.
- De la misma manera, para mover el disco 3 de Torre A hasta Torre B, será necesario mover los primeros dos hasta Torre C usándolo como auxiliar.
- Igualmente, para mover el disco 2 hasta Torre C, es necesario mover el disco 1 hasta Torre B, ahora es evidente que: ¡Se ha de mover el primer disco hasta Torre B!

Luego, el único disco que es razonable mover ahora es el 2, y el único lugar a moverlo es Torre C, después, según el razonamiento anterior se ha de mover el disco 1 hasta Torre C (sobre el disco 2), dejando Torre B libre para colocar el disco 3.

El algoritmo utilizado en este procedimiento es el siguiente:

```
public void solucionViaRecurrencia(int numeroDiscos, string origen, string auxiliar, string destino)
{
    if (numeroDiscos == 1)
    {
        count++;
        listaSolucion = Convert.ToString(count + ". De Torre " + origen + " a " + destino + "\n");
        dataGridView_Solucion.Rows.Add(listaSolucion);
    }
    else
    {
        solucionViaRecurrencia(numeroDiscos - 1, origen, destino, auxiliar);
        count++;
        listaSolucion = Convert.ToString(count + ". De Torre " + origen + " a " + destino + "\n");
        dataGridView_Solucion.Rows.Add(listaSolucion);
        solucionViaRecurrencia(numeroDiscos - 1, auxiliar, origen, destino);
    }
}
```

Análisis del problema

Analizamos un sistema o aplicación informática, que debemos implementar y que genera operaciones matemáticas, resultados y que facilita el procedimiento de realizar un cálculo automático utilizando algoritmos recursivos e iterativos para la solución del problema en torno a dicho proyecto el cual es el solicitado y que debemos analizar los problemas a resolver para su buen funcionamiento, sin embargo, siempre debemos de ingresar valores manuales porque así se dicta en los requerimientos del uso y funcionamiento.

Se realizara por cada calculo un resultado nuevo, por cada operación que se necesite tendremos un resultado final, ya que la aplicación nos dará la solución cada vez que ingresemos un valor y queramos convertirlo a otro resultado.

La implementación de las torres de Hanói, lleva un complejo procedimiento si nos basamos en las fórmulas matemáticas, pero si recurrimos a la recurrencia podremos ver la diferencia de simplicidad que abarca este problema. Al final, mediante los algoritmos ya indicados anteriormente podremos realizar la resolución de las implicaciones presentadas con los diferentes niveles de juego.

Finalmente, el análisis da como resultado el requerimiento de las funciones establecidas en los procesos de interacción con el usuario, y que es necesaria la implementación con las constantes de código limpio, amigable y de fácil compresión para el lector que este aprovechando esta investigación.

Análisis de la Solución

Mediante la preparación del entorno de desarrollo especializado para que dicha aplicación en lenguaje C# del entorno de Visual Studio 2015, se desarrolló este sistema de juego orientado al juego Torres de Hanói, siendo funcional y dinámico, se debió de realizar algunas configuraciones básicas para este resultado final del sistema o aplicación.

Algunas configuraciones son:

- I. Actualización del entorno de desarrollo Visual Studio 2015.
- II. Ediciones de iconos para el entorno gráfico en <http://www.iconarchive.com/>.

La solución para el desarrollo fue preparada y funcional satisfactoriamente, ya que al principio había problema para desarrollar una aplicación con resolución de los algoritmos de solución al problema, donde indica la lista de movimientos a utilizar pero fue resuelto con aplicaciones propias del IDE de Visual Studio 2015.

El sistema tiene dos modalidades las cuales son las demostraciones y el modo usuario para que se interactúe en modo manual para resolver el problema dependiendo los niveles de dificultad.

Resultados Obtenidos

Tomando en cuenta los objetivos específicos, podemos decir que se ha cumplido con los objetivos establecidos y propuestos por el docente del curso de Inteligencia Artificial, donde nos indica cómo y cuándo realizar los criterios.

Efectivamente, la funcionalidad de dicha aplicación es totalmente apegada a como se planteó anteriormente en la propuestas del problema, y así siguiendo las normas hasta la finalización de la misma. Finalizada la aplicación, los datos obtenidos de nuestra aplicación, son importantes para una futura optimización del software y experiencia en el desarrollo de software.

Se ha logrado aplicar bien los fundamentos aprendidos los cuales han sido de mucha importancia para este desarrollo. Además, los error que surgieron durante el desarrollo se han solucionado dejando de un lado la inseguridad de que nuestros sistema, que pueda emitir algún tipo de error y así perjudicar el usuario final.

Conclusión

Se da la conclusión de que el juego de las Torres de Hanói es un juego matemático en el cual podemos ver o demostrar las habilidades que tenemos y al igual esto nos sirve mucho para resolver algún problema ya que con la lógica y la razón llegaron a la solución.

Con esta actividad se buscó que el estudiante pueda visualizar las características primordiales que se necesita, para poder desarrollar un sistema de información, ya que necesita recolectar datos específicos que a simple vista pueden resultar insignificante pero que son indispensables para el buen desarrollo de sistemas lógicos matemáticos.

Finalmente se debe mencionar el uso que se le dará, estará destinado solamente a la parte experimental y de aprendizaje por parte de los estudiantes que quieran repasar el código fuente de la aplicación para así entender como funciona y su desarrollo.

Se espera que cualquier problema presentado en un futuro con dicha información, pueda ser modificada en pro del aprendizaje y sus preocupaciones ya que está enfocada al bien del estudiante.

Recomendaciones

Las recomendaciones siempre serán un punto de modificación de algún proyecto o sistema que este en desarrollo o ya haya sido desarrollado. Pero se recomienda cualquier problema de compilación en la aplicación, sea solucionado sin problemas por la sencillez del código que existe en la aplicación. La mayor recomendación que podemos dar como punto fuerte, es una mejor personalización de la interface gráfica y el estilo de notificaciones de dicho sistema para que el usuario sienta una atracción mayor por el estilo y modo de comunicación entre ambos.

En cuanto al entorno grafico sea bienvenida una implementación de un mejor estilo visual ya que no es muy elegante pero es estable.

Cronograma de Trabajo y Bitácora

Bitácora de Trabajo	
Fecha	Observaciones
17/10/2016	Se hizo investigacion sobre el problema a resolver en internet.
18/10/2016	Se creo el proyecto en en el lenguaje C#, solamente parte de formularios.
19/10/2016	Se hizo investigacion sobre el problema a resolver en internet.
20/10/2016	Se hizo investigacion sobre el problema a resolver en internet.
21/10/2016	Se hizo investigacion sobre el problema a resolver en internet.
22/10/2016	Se hizo investigacion sobre el problema a resolver en internet.
23/10/2016	Se realizaron practicas de la implementacion de algoritmos que resolvieran problemas mediante recursividad.
24/10/2016	Se hizo una pequeña implementacion de graficos para la comprension del funcionamiento.
25/10/2016	Se trabajo en la parte escrita, como la portada, introduccion, y objetivos.
06/11/2016	Se hizo un pequeño avance en la estructura del sistema, implementando las variables a utilizar y logica a usar.
07/11/2016	Se avanzo el trabajo escrito.
08/11/2016	Se crearon otros formularios del proyeceto para normas, soporte y vistas del entorno.
10/11/2016	Se hizo la implementacion de objetos y creacion de graficos de las vistas del formulario.
11/11/2016	Se realizo validaciones ya con algunos detalles programados.
12/11/2016	Personalizacion del entorno grafico y validaciones del sistema.
13/11/2016	Adelanto del trabajo en el marco teorico, desarrollo y problemas a resolver.
14/11/2016	Trabajoi en marco teorico, resúmenes, analisi solucion.
17/11/2016	Trabajo en la parte teorica.
18/11/2016	Revision de funcion de los demos en el desarrollo.
19/11/2016	Revision y solucion de errores pequeños en el sistema.
20/11/2016	trabajando en soluciones y problemas del escrito
21/11/2016	trabajo en el entorno grafico del sistema
22/11/2016	trabajo en el entorno grafico del sistema
23/11/2016	finalizando el trabajo del sistema
24/11/2016	finalizando el trabajo escrito
25/11/2016	finalizando el trabajo del sistema
26/11/2016	finalizando el trabajo escrito
27/11/2016	finalizando el trabajo del sistema
27/11/2016	finalizando el trabajo escrito

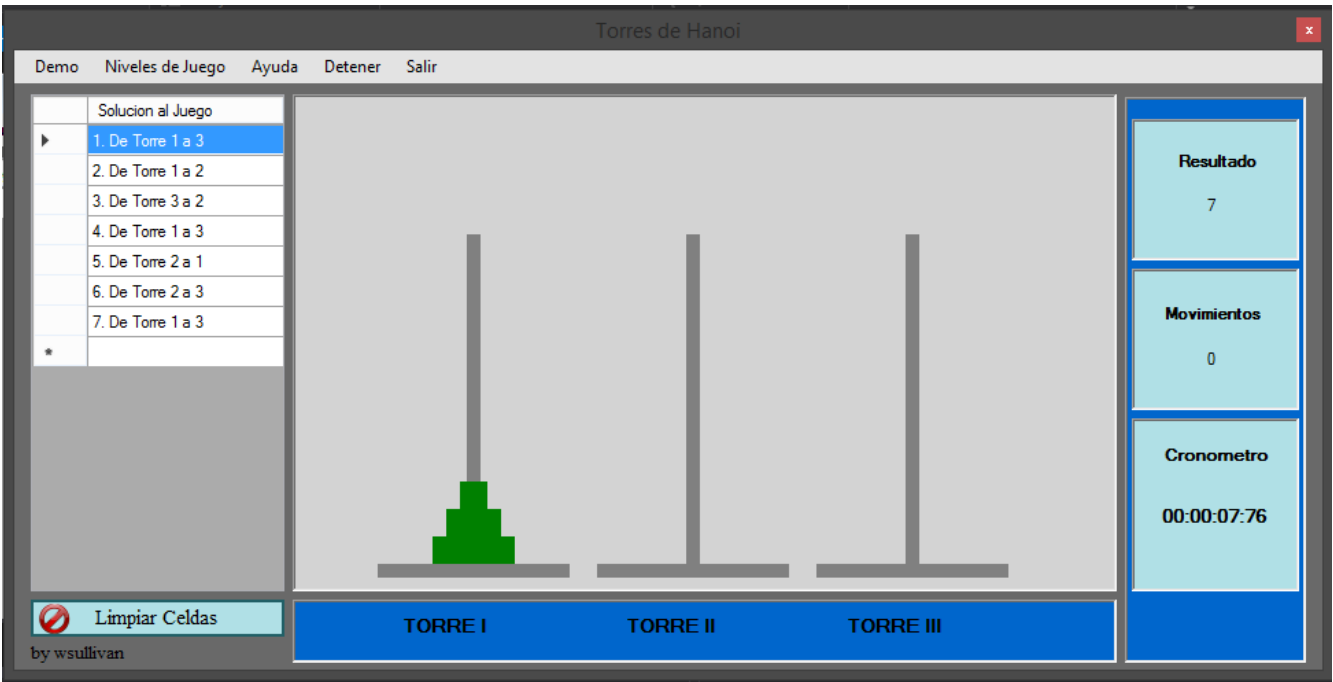
Cronograma de Trabajo			
Fecha	Hora Inicial	Hora Final	Observaciones
17/10/2016	21:00	22:00	Se hizo investigacion sobre el problema a resolver en internet.
18/10/2016	18:10	19:30	Se creo el proyecto en en el lenguaje C#, solamente parte de formularios.
19/10/2016	10:00	13:45	Se hizo investigacion sobre el problema a resolver en internet.
20/10/2016	12:00	14:15	Se hizo investigacion sobre el problema a resolver en internet.
21/10/2016	15:15	15:35	Se hizo investigacion sobre el problema a resolver en internet.
22/10/2016	18:00	19:45	Se hizo investigacion sobre el problema a resolver en internet.
23/10/2016	19:00	22:45	Se realizaron practicas de la implementacion de algoritmos que resolvieran problemas mediante recursividad.
24/10/2016	17:30	22:38	Se hizo una pequeña implementacion de graficos para la comprension del funcionamiento.
25/10/2016	19:15	23:52	Se trabajo en la parte escrita, como la portada, introduccion, y objetivos. Y se trabajo en el sistema
06/11/2016	20:30	23:20	Se hizo un pequeño avance en la estructura del sistema, implementando las variables a utilizar y logica a usar.
07/11/2016	20:00	01:52	Se avanzo el trabajo escrito y en el sistema
08/11/2016	20:00	01:20	Se crearon otros formularios del proyeceto para normas, soporte y vistas del entorno.
10/11/2016	21:50	23:10	Se hizo la implementacion de objetos y creacion de graficos de las vistas del formulario.
11/11/2016	21:00	22:40	Se realizo validaciones ya con algunos detalles programados.
12/11/2016	20:45	22:45	Personalizacion del entorno grafico y validaciones del sistema.
13/11/2016	19:50	23:00	Adelanto del trabajo en el marco teorico, desarrollo y problemas a resolver.
14/11/2016	19:35	22:16	Trabajoi en marco teorico, resúmenes, analisis solucion.
17/11/2016	21:00	22:45	Trabajo en la parte teorica.
18/11/2016	21:10	23:36	Revision de funcion de los demos en el desarrollo.
19/11/2016	20:10	01:30	Revision y solucion de errores pequeños en el sistema.
20/11/2016	20:00	22:00	trabajando en soluciones y problemas del escrito
21/11/2016	19:55	22:30	trabajo en el entorno grafico del sistema
22/11/2016	19:50	22:15	Trabajo en el entorno grafico del sistema
23/11/2016	20:16	23:26	finalizando el trabajo del sistema
24/11/2016	20:10	23:00	finalizando el trabajo escrito y el sistema
25/11/2016	20:00	22:50	finalizando el trabajo del sistema y el escrito
26/11/2016	19:45	22:36	finalizando el trabajo escrito
27/11/2016	11:00	14:35	finalizando el trabajo del sistema
27/11/2016	17:00	20:25	finalizando el trabajo escrito

Bibliografía

- Wikipedia. Historia sobre las torres de Hanói: https://es.wikipedia.org/wiki/Torres_de_Hanói
- Explicación desde diapositivas: <http://es.slideshare.net/Vanessajrp/torres-de-hanoi-15334421>
- Leyenda del problema de Hanói:
http://www.ceibal.edu.uy/UserFiles/P0001/ODEA/ORIGINAL/090628_torres_hanoi.elp/leyenda.html
<https://eltrasterodepalacio.wordpress.com/2014/04/02/las-torres-de-hanoi-una-leyenda-y-un-juego/>
- Torres de Hanói y curiosidades: https://books.google.com/books?id=7HvQV8PhskC&pg=PA141&lpg=PA141&dq=tower+of+hanoi+investigation&source=bl&ots=6wzyEbgs_s_&sig=KOQHnrZyKG5nh6gE1zB98aROM2I&hl=es&sa=X&ved=0ahUKEwikpo7qvMLQAhVCtRoKHRN8CgQ4ChDoAQhUMAc#v=onepage&q=tower%20of%20hanoi%20investigation&f=false
- <http://www.rodoval.com/heureka/hanoi/hanoi2.html>
- The Tower of Hanoi: A Bibliography, <http://www.cs.wm.edu/pkstoc/biblio2.pdf>
- 13On the Frame–Stewart algorithm for the multi-peg Tower of Hanoi problem, Discrete Applied Mathematics 120 (2002) 141–157.
- <http://stackoverflow.com/questions/18820151/how-to-populate-a-datagridview-with-an-array-of-user-generated-integers>
- <http://stackoverflow.com/questions/12535722/what-is-the-best-way-to-implement-a-timer>
- [https://msdn.microsoft.com/es-es/library/system.timers.timer\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.timers.timer(v=vs.110).aspx)
- <http://stackoverflow.com/questions/1302741/how-do-i-draw-simple-graphics-in-c>
- http://www.deitel.com/articles/csharp_tutorials/20051025/IntroTo2DGraphics.html

Anexos

1. Pantalla Principal.



2. Normas del Juego.

Normas del Juego

DESCRIPCION

El juego, en su forma más tradicional, consiste en tres varillas verticales.

En una de las varillas se apila un número indeterminado de discos (elaborados de madera) que determinará la complejidad de la solución, por regla general se consideran ocho discos. Los discos se apilan sobre una varilla en tamaño decreciente. No hay dos discos iguales, y todos ellos están apilados de mayor a menor radio en una de las varillas, quedando las otras dos varillas vacantes.

El juego consiste en pasar todos los discos de la varilla ocupada (es decir la que posee la torre) a una de las otras varillas vacantes.

Para realizar este objetivo, es necesario seguir tres simples reglas:

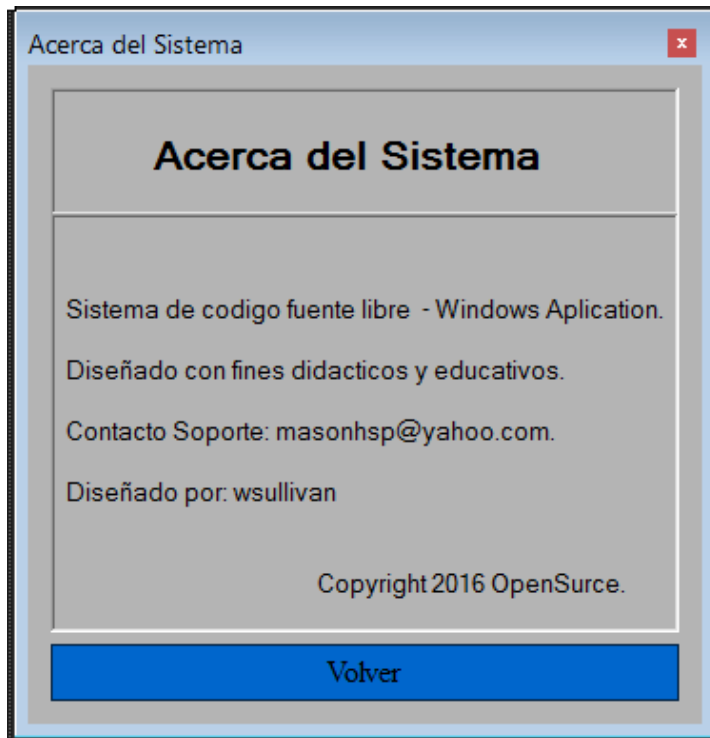
- I. Sólo se puede mover un disco cada vez.
- II. Un disco de mayor tamaño no puede descansar sobre uno más pequeño que él mismo.
- III. Sólo puedes desplazar el disco que se encuentre arriba en cada varilla.

Existen diversas formas de realizar la solución final, todas ellas siguiendo estrategias diversas.

Ejemplo de 4 discos:



Acerca del Sistema.



Algoritmo recursivo para la solución.

```
public void solucionViaRecurrencia(int numeroDiscos, string origen, string auxiliar, string destino)
{
    if (numeroDiscos == 1)
    {
        count++;
        listaSolucion = Convert.ToString(count + ". De Torre " + origen + " a " + destino + "\n");
        dataGridView_Solucion.Rows.Add(listaSolucion);
    }
    else
    {
        solucionViaRecurrencia(numeroDiscos - 1, origen, destino, auxiliar);
        count++;
        listaSolucion = Convert.ToString(count + ". De Torre " + origen + " a " + destino + "\n");
        dataGridView_Solucion.Rows.Add(listaSolucion);
        solucionViaRecurrencia(numeroDiscos - 1, auxiliar, origen, destino);
    }
}
```