Inteligencia Natural

***Un relato matematico***

***acerca de la existencia del individuo,***

***y el sentido de la vida.***

By

Tony Fabrico

**Title of Your Book**

©Copyright 2022 Author Name, Title

**All rights reserved**

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the express written permission of the author.

**Name of Printer Goes Here**

**ISBN:** 000-1234567890

**Your Organization Title here**

An Example Incorporated Company

0000 Example Street, Sample Suite

State, City & Zip

0123-456-7890

*Free book template downloaded from:*

https://usedtotech.com

admin@usedtotech.com

Contents

[About the Author iv](#_Toc119975481)

[Foreword vi](#_Toc119975482)

[How to use this template? 1](#_Toc119975483)

[Second Chapter Title 5](#_Toc119975484)

[Third Chapter Title 9](#_Toc119975485)

# About the Author

When you create pictures, charts, or diagrams, they also coordinate with your current document look. You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab.

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

## This is level 2 heading!

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command.

You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command.

**Author Name**

Title of the Author Here

# Foreword

When you create pictures, charts, or diagrams, they also coordinate with your current document look. You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

## This is level 2 heading!

To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template.

You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command.

### This is level 3 heading!

You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command.

The Thinking Thing

Chapter 1

# How to use this template?

Looking for a free template of book for writing a book? You can use this professionally formatted Word book template for creating a **nicely formatted book** for online submission or for printing through any self-publishing company. This book template is especially created for preparing small books in 5 x 8 paperback. You will definitely find this Word template useful in formatting your book. The heading above these two paragraphs is a **“Level 1”** heading. Everything has been professionally organized and styled through Ms. Word’s style-sheets. You can start your sections or chapters with this Level 1 heading.

## Perfect formatting for self-authors

If you love writing books then you should definitely use this easy to use yet nicely formatted book layout for your next book. The heading above this paragraph is **“Level 2”** heading. Everything is formatted using predefined styles so instead of making any manual change in the formatting, you should modify the associated styles to let your changes effectively apply at every instance of that formatting. Feel free to personally use this outstanding book template in any of your fiction or non-fiction work. This is more than a blank book template in Microsoft Word; this is a complete book design for commercial printing & publishing.

## Print-ready typesetting & layout

The layout, design, formatting, automatic table of contents –everything has already been included in the template and this is for your **Personal Use Only**. We have included every necessary section into this book writing template. The starting pages are numbered using Roman numerals. However, everything is still modifiable for you. Even if you are not very good with Microsoft Word, still by using this template, you can make your book look professional. Either write your book from scratch or copy your existing content into the template, you’ll definitely love it. Make sure you check the **“Navigation Pane”** to know about the different levels of headings and also check the **“Styles”** group for available styles. This book template is formatted into **“Sections”**. Every Section has its own properties, please also take care of these things when adding new sections into the template.

## Adding new content

For adding new chapters in this template, simply copy these dummy texts entirely and paste them at the insertion point from where you need to start a new chapter. Otherwise, you can write your content from scratch using the predefined styles in the template. Please also note that this template uses **“Different First Page”** & **“Different Odd/Even Pages”** through Microsoft Word’s sections. Insert new chapter in a separate section.

## Using chapter title on the header

If you want to write chapter title on the header instead of using author name then you can modify it on the header of the section you are working on. Using sections is easy in Microsoft Word, if you don’t know about modifying sections then **please Google** about inserting & using sections in Ms. Word, you’ll certainly find many articles on Microsoft Word’s sections.

## Our “Important Notes”

Please first also read our “Important Notes” document attached in the download package. Our free templates normally consist of dummy text throughout the document so before you submit your book for final publishing, make sure there **remains no placeholder text** in the manuscript. You’re allowed to download, modify and use our free resources for your personal use only.

## Help us by accrediting!

If you find this book template helpful in your writings then please spread the word and appreciate us by sharing these awesome and free templates with others as much as you can so that many people will also take benefits of our great free resources. Please remember that this template is for your **PERSONAL USE ONLY**. For any questions, please write to us at the following addresses:

admin@usedtotech.com

https://usedtotech.com

\*\*\*

## This is level 2 heading

You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.

Capitulo 2

# estados de un proceso

Veamos otra vez la suma de un listado de numeros, que hemos utilizado de ejemplo en el capitulo anterior.

Inicialmente la lista contiene 5 numeros, y esa es toda la informacion que tenemos del problema hasta ese momento. Llamaremos al conjunto de datos, "contexto" del problema.

Hemos encontrado la solucion despues de realizar 4 operaciones de suma entre 2 numeros. Han sido 4 pasos.



Observad que, en cada paso, despues de cada suma, el conjunto de datos ha cambiado.

Cada vez que realizamos una operacion sobre los datos, el contexto cambia de estado.

Y entre los posibles estados de un proceso, distinguiremos

- un “estado incial”,

- un “estado final”, y

- un conjunto de “estados intermedios”.

Iniciamos la suma con el listado de numeros que queremos sumar, este es el “estado inicial”.

El objetivo es obtener el resultado total de la suma de todos los numeros de la lista, ese es el “estado final”.

En el caso de una partida de ajedrez.

El “estado incial” sera cuando las fichas estan colocadas en la posicion de salida, antes de que se haya realizado ningun movimiento.

El “estado final” sera cuando se ha dado jaque-mate, la partida termina, este es el estado al que queremos llegar. Y entre el estado inicial y el estado final, la partida cambiara de estado cada vez que un jugador realice un movimiento, estos seran los “estados intermedios”.

Imaginemos ahora que en mitad de una partida de ajedrez, quisieramos dejar el juego por hoy, para continuar manyana desde el mismo punto en que lo hemos dejado.

Para poder continuar manyana desde el mismo punto, sera necesario recordar el estado en que la partida habia quedado.

Para recordar el estado del juego podriamos escribir en un listado la posicion de las fichas sobre el tablero, asi podriamos reconstruir la partida para continuarla en otro momento desde el mismo punto.

En este caso, la informacion del estado sera la posicion de las fichas sobre el tablero.

Utilizaremos la idea de estados del proceso para resolver el problema, buscando un camino que nos lleve desde el estado inicial hasta el estado final a traves de estados intermedios.

Podrian existir un gran numero de estados posibles, pero queremos saber cuales de todos los posibles, son los que forman el camino que va desde el estado incial hasta el estado final.

El juego de ajedrez es de gran complejidad, y si quisieramos generar todos los estados posibles, nos encontrariamos con un elevado numero de estados diferentes. Algo asi como 10 elevado a 120.

Para verlo graficamente utilizaremos un ejemplo mas simple.

Tic-Tac-Toe ( 3 en linea)



El dibujo muestra en el juego de 3 en linea, el camino a traves de distintos estados que va desde el estado inicial hasta uno de los posibles estados finales.

Para resolver problemas, utilizaremos el concepto de "estados", junto con una estructura llamada

"Grafo".

En matemáticas y ciencias de la computación, un "grafo" es una estructura abstracta que representa relaciones entre objetos.

Un grafo consiste en un conjunto de nodos y un conjunto de aristas que conectan pares de nodos.



(grafico grafo)

Los grafos se utilizan de muchas formas diferentes para el analisis de muchos tipos de problemas diferentes, pero nosotros construiremos un grafo del siguiente modo.

Dentro de los nodos, colocaremos un estado del proceso, y las aristas que unen los nodos nos mostraran la relacion entre padres e hijos.

Veamos por ejemplo, como iniciariamos la construccion de un grafo con los estados del juego de 3 en linea.

Empezaremos poniendo el estado incial dentro de un nodo.

Los hijos del estado inicial mostraran todas las jugadas posibles, en este caso el tablero esta vacio y puede colocarse una ficha en cualquier casilla, y por lo tanto existen 9 jugadas posibles, crearemos los 9 hijos del estado inicial y los uniremos al padre con aristas.



A partir de estos nuevos estados tambien pueden realizarse jugadas validas, en todos estos tableros existen 8 casillas vacias y podemos colocar una ficha en cualquiera de ellas, generaremos todos los hijos de estos estados, los pondremos dentro de un nuevo nodo y uniremos el padre con el hijo a traves de una arista.

Lo mostrare para uno solo de los hijos.



El grafo empieza a transformarse en una gran red de nodos y aristas.

Si continuamos con este proceso de generar hijos por cada jugada valida desde cada nodo, la red final sera grande y compleja.

En el caso de este juego existen

9 \* 8 \* 7 \* 6 \* 5 \* 4 \* 3 \* 2 = 362.880 estados posibles.

es decir que el grafo tendra 362.880 nodos.

Una vez generados todos los posibles estados podremos ver entre ellos cuales son los estados finales, y podremos ver el camino entre estados que llevan desde el estado inicial hasta un estado final.

Como en el ejemplo anterior.



En resumen, hemos comentado

a que llamamos estados de un proceso,

que cosa es un grafo, y

como es posible encontrar, entre todos los posibles, los "caminos" que nos llevan a la solucion de un problema.

Con estas ideas, estamos ya preparados para iniciar el experimento.

Capitulo 3

# resolver un problema

Es el momento de ver como aplicar las ideas que hemos comentado en la practica. Con este proposito disenyaremos una maquina que resuelva un problema.

El objetivo de este experimento es ver al sistema en funcionamiento para entender como trabaja. Lo pondremos a prueba con un problema simple para evitar distracciones innecesarias, de este modo podremos enfocar toda nuestra atencion en ver como el sistema encuentra la solucion.

Veamos primero el problema que utilizaremos como ejemplo.

## El problema

Sera un juego de mesa que se desarrollara sobre un tablero cuadriculado de ocho por ocho casillas.

Sobre el tablero colocaremos una unica ficha, que se movera una casilla en cada jugada a cualquiera de las casillas contiguas.

El tablero representara un laberinto, con una entrada y una salida.

La entrada estara situada en la casilla 'A1'

y la salida en la casilla 'H8'.

Para referenciar a una casilla determinada utilizaremos un sistema de coordenadas, en donde las columnas del tablero se referencian por una letra desde la 'A', hasta la 'H', y las filas por numeros desde el '1', hasta el '8'.



El problema al que el sistema debe encontrar una solucion es :

"Cual es el camino que la ficha debe recorrer para ir desde la entrada hasta la salida del laberinto ?"

Veamos ahora como sera la maquina que encontrara la solucion.

## La maquina

Hemos comentado que una maquina que resuelve problemas procesando informacion debe disponer de

* una memoria en donde mantener el conjunto de datos que describen el contexto del problema, y
* una forma de modificar la informacion de forma productiva, aplicando transformaciones metodicas que acerquen el conjunto de datos hacia la solucion.

Hemos comentado tambien que la vida dispone de recursos que podemos utilizar con este objetivo.

En particular, el ADN que utilizaremos como memoria del sistema, y la reproduccion que sera el mecanismo que transforme la informacion aplicando una funcion especifica.

Para utilizar estos recursos en la resolucion del problema que hemos planteado, crearemos una nueva especie de seres vivientes, adaptada especialmente al problema que queremos resolver.

Hemos comentado que es posible describir el proceso de resolucion de un problema como una sucesion de "estados" en forma de cadena que va desde el "estado incial" hasta el "estado final" a traves de un conjunto de "estados intermedios" relacionados.

Pensar en terminos de estados nos ayudara en la implementacion.

El problema que queremos abordar es el de un juego de mesa. y el estado sera, la situacion en que se encuentra el juego en un instante puntual del tiempo.

el juego cambiara de estado cada vez que se realice un movimiento.

con todo esto tenemos lo que necesitamos para implementar el sistema.

veamos.

la informacion que debemos mantener y modificar, el contexto, es la informacion que describe un estado cualquiera del juego.

el juego cambia de estado cuando se realiza una jugada valida, de modo que, la funcion que se aplicara para modificar la informacion es la que realiza jugadas validas desde la situacion actual.

Deciamos entonces que, para resolver el problema, crearemos una nueva especie de seres vivientes.

Haremos esto porque,

las criaturas vivientes, de todas las especies, comparten ciertas caracteristicas en comun.

los seres vivos,

nacen, se reproducen y mueren.

todos hacen esto.

otra caracteristica comun es que la descripcion del individuo esta almacenada en ADN.

Todo esto lo proporciona la vida, y es basicamente, una maquina de calculo.

los individuos tienen un ciclo de vida definido, estos nacen, se reproducen y mueren.

los individuos mantienen un set de informacion.

la informacion se transforma con cada generacion.

Y la evolucion de la especie, es la evolucion del proceso de calculo.

La maquina esta hecha, y para utilizarla solo nos queda,

- introducir el conjunto de datos que queremos que sea procesado, y

- la funcion que se utilizara para transformar los datos.

Veamos cuales son los datos.

los datos que queremos procesar son los estados del juego,

partiendo del estado incial la maquina debe generar todos los estados intermedios hasta alcanzar el estado final.

Un individuo de esta especie representa un estado del juego.

para representar un estado de este juego utilizaremos 2 datos.

- la posicion de la ficha en el tablero, y

- el camino que la ficha ha recorrido hasta ahora

Por ejemplo,



Veamos cual es la funcion de transformacion,

durante la reproduccion, la funcion de transformacion debe generar el siguiente estado.

y para generar el siguiente estado del juego, bastara con realizar una jugada valida.

Por ejemplo desde la posicion ‘B3’, es posible realizar 8 jugadas validas, A2, A3, A4, B2, B4, C2, C3, C4,



el mecanismo de reproduccion realizara una jugada valida desde el estado actual, de este modo generara el siguiente estado, y la informacion que describe este nuevo estado generado se almacenara en un hijo.

El siguiente grafico muestra todos los hijos posibles para el individuo del ejemplo anterior, 'B3'



El ADN de estos individuos contiene 2 datos,

* posicion y
* camino

la “posicion” de la ficha en el tablero se obtiene realizando una jugada valida desde la posicion del padre, el “camino” recorrido se obtiene agregando al camino del padre la posicion del hijo.

Como ultimos detalles de configuracion dire tambien que:

* la reproduccion de esta especie es asexual, es decir que los individuos de esta especie se reproducen por si solos.
* En una generacion, un individuo crea a todos los hijos posibles. En el ejemplo anterior, los 8 hijos posibles de B3, seran creados en la primera generacion.
* Una vez generados los hijos, el padre ya no es necesario y debe ser eliminado. Hablaremos con detalle de la necesidad de eliminar a los individuos que no son productivos mas adelante en los siguientes capitulos.

Y con esto el sistema queda definido.

Ahora pongamoslo en marcha y veamos lo que sucede.

## Puesta en marcha

Para arrancar el proceso debemos crear a un individuo que represente el “estado inicial”.

En este juego, el estado incial, es el tablero con una ficha colocada en la entrada del laberinto, posicion ‘A1’, antes de realizar ningun movimiento.

El individuo que representa el estado incial se muestra en el siguiente grafico.



Este individuo no resolvera el problema, porque no es esa su funcion. Desconoce el problema, desconoce el sistema al que pertenece, desconoce el proceso que se esta llevando a cabo.

Ha sido creado para

* nacer,
* reproducirse y
* morir.

Eso es lo que se espera de el, y eso es lo que hara, y ninguna otra cosa.

El primer individuo de esta nueva especie ha sido creado. Y cuando lo dotamos de vida,

sucede lo siguiente ...

‘A1’ tendra hijos, y estos ocuparan las posiciones marcadas con X que se muestran en el dibujo de la izquierda.

 Y entre ellos nace ‘B2’

Ahora que ‘B2’ esta vivo, también tendrá hijos, y estos ocuparan las posiciones marcadas con X que se muestran en el dibujo de la izquierda.  Y entre ellos nace ‘C3’

Y ‘C3’ también tendrá hijos, y ocuparan las posiciones marcadas con X en el dibujo de la izquierda. Y entre ellos nace ‘D4’

‘D4’ también tendrá hijos, y ocuparan las posiciones marcadas con X en el dibujo de la izquierda. Y entre ellos nace ‘E5’



‘E5’ también tendrá hijos, y ocuparan las posiciones marcadas con X en el dibujo de la izquierda. Y entre ellos nace ‘F6’

‘F6’ también tendrá hijos, y ocuparan las posiciones marcadas con X en el dibujo de la izquierda. Y entre ellos nace ‘G7’



‘G7’ también tendrá hijos, y ocuparan las posiciones marcadas con X en el dibujo de la izquierda. Y entre ellos nace ‘H8’



Este ultimo individuo, que nace con posicion ‘H8’, representa el ‘estado final’ que queriamos alcanzar.

Y tiene la solucion al problema.

Este individuo conoce el camino que va desde la entrada hasta la salida del laberinto.

Esta grabado en su ADN, y por lo tanto, es un conocimiento que heredaran todos sus descendientes.



Capitulo 4

# caracteristicas del sistema

En el capitulo anterior hemos visto como funciona el sistema, y el mecanismo que ha utilizado para encontrar la solucion al problema.

Lo hemos visto en accion y ya tenemos una primera impresion.

Pero,

que es exactamente esta cosa?,

que tipo de problemas puede resolver?,

que podemos esperar de un sistema como este?

En los siguientes capitulos, analizaremos algunas de sus caracteristicas, y en particular,

* su eficiencia, en terminos de, cuantos calculos necesita para encontrar la solucion a un problema?

es mas eficiente que otros sistemas, o menos ?

* su potencia, en terminos de, cual es la dificultad de los problemas que puede abordar ?

puede resolver problemas mas complejos que otros sistemas, o menos ?

* su coste, en terminos de, cual es el coste de resolver un problema utilizando un sistema de este tipo ?

Con esta informacion podremos entender mejor lo que esta cosa es. Podremos encontrar su lugar dentro del universo de los sistemas inteligentes. Y podremos entender la necesidad de ciertos detalles que ahora pasan desapercibidos.

Creo que en este punto, y antes de continuar con el analisis, no esta de mas adelantar algun resultado de lo que viene a continuacion.

Recordemos que estamos hablando de vida.

Y la vida no tiene nada de "ordinario".

Cualquier aspecto de la vida que analicemos resultara siempre sorprendente.

El experimento que hemos realizado ha sido casi como un juego de ninos.

ha pasado rapido, quizas no significa nada, quizas es una broma, un truco.

Ahora vamos a verlo un poco mas despacio, prestando mas atencion a los detalles.

Empecemos hablando de eficiencia.

Capitulo 5

# eficiencia

Los sistemas computacionales, asi como los procedimientos que se utilizan para resolver problemas logicos, pueden ser medidos y comparados.

No todos los computadores son iguales, algunos pueden abordar problemas de mayor complejidad que otros, o consumir menos recursos.

Tambien podemos encontrar diferencias entre los metodos que utilizamos para resolver problemas. Un mismo problema puede ser abordado de diferentes formas, utilizando distintos procedimientos.

En este capitulo, nos proponemos obtener una medida de la “Eficiencia Algoritmica” del sistema computacional basado en especies de seres vivientes que estamos analizando.

No tendremos problemas en establecer una medida aproximada pero rigurosa de la eficiencia del sistema, razonando sobre los hechos que hemos podido observar en el experimento del capitulo anterior.

Hablaremos de Eficiencia Algoritmica.

Un algoritmo es la descripción de el procedimiento utilizado para resolver un problema, como un conjunto ordenado de pasos.

Por ejemplo, si queremos ordenar una baraja de naipes, como lo haremos ?

Es posible que 2 personas propongan formas diferentes de conseguir el mismo resultado. Porque existen diferentes formas de ordenar una baraja, diferentes procedimientos, algoritmos.

Sin embargo, algunos procedimientos ordenaran la baraja con mas operaciones que otros.

Ante la posibilidad de utilizar distintos algoritmos para realizar el trabajo, preferiremos aquel que consiga el objetivo con un menor numero de pasos.

Porque hacer el trabajo con menos operaciones se traduce en un menor consumo de recursos, como tiempo y energia.

Realizar el trabajo con un menor consumo de recursos es lo que llamaremos, Eficiencia.

Diremos, por lo tanto que, el algoritmo que hace el trabajo con menor numero de operaciones es, el mas eficiente.

Para dejar clara esta idea, veamos, por ejemplo, 2 algoritmos diferentes que podriamos utilizar para ordenar una baraja.

* En el primer algoritmo, reviso toda la baraja para extraer el naipe de menor valor, y una vez localizado lo separo de la baraja. Repito el procedimiento con los naipes que han quedado en la baraja para obtener el segundo naipe de menor valor y separarlo de la baraja, y repito el procedimiento para obtener la tercera, y otra vez para la siguiente, y repito el proceso una y otra vez hasta tener la baraja ordenada.
* En el segundo algoritmo, preparo espacio sobre la mesa para cada naipe en una cuadricula. Extraigo el primer naipe de la baraja y lo coloco directamente en el lugar que tiene reservado en la cuadricula, y lo mismo hago con el segundo, el tercero y todos los demas.

Veamos ahora cual de estos 2 algoritmos es mas eficiente.

1. El primer algoritmo, revisa cuidadosamente la baraja comparando uno a uno todos los naipes para extraer el de menor valor.

Si la baraja tiene 50 naipes, se habran realizado 50 comparaciones. Despues de realizar 50 comparaciones tenemos 1 naipe ordenado, y 49 sin ordenar.

Para extraer el segundo naipe de menor valor repetiremos el procedimiento anterior con los 49 naipes que todavia estan sin ordenar, y eso requerira otras 49 comparaciones. Repetiremos el procedimiento para extraer el tercer naipe con los 48 todavia sin ordenar, y esto requerira otras 48 comparaciones, repetiremos el procedimiento una y otra vez hasta que no queden naipes sin ordenar.

cuantas comparaciones seran necesarias ?

50 + (50 - 1) + (50 - 2) + (50 - 3) + (50 - 4) + ... + (50 - 49)

Si al numero de elementos que queremos ordenar le llamamos 'N', en este caso N=50, el numero de operaciones necesarias seran aproximadamente

N \* (N-1)

Podemos afirmar que el coste en tiempo de ejecutar este algoritmo es de "el orden de" N cuadrado.

Veamos ahora el segundo algoritmo.

1. Sobre una superficie, dibujamos una cuadricula, reservando un lugar para cada naipe de la baraja.

Extraemos un naipe, digamos que es el cinco de treboles, y lo colocamos dentro de la cuadricula, en el lugar reservado para el cinco de treboles. Extremos el segundo naipe, digamos que es el dos de diamantes, lo colocaremos dentro de la cuadricula, en el lugar reservado para el dos de diamantes.

Hemos extraido 2 naipes y tenemos 2 naipes ordenados.

Repetimos el procedimiento con el tercero, y con los restantes. Cuantas operaciones son necesarias para ordenar la baraja ? Una operacion por naipe.

Si el numero de naipes es N, entonces el coste en tiempo de este algoritmo es del orden de N.

Ahora que tenemos una medida del coste en tiempo de ambos algoritmos podemos compararlos.

* el primero es del orden de N cuadrado.
* el segundo es del orden de N.

el segundo algoritmo hace el trabajo con menos operaciones, y por lo tanto es mas eficiente.

Ahora que hemos visto como pueden compararse 2 metodos, volvamos al sistema inteligente formado por seres vivientes que estamos analizando, y veamos el metodo que ha utilizado el sistema en terminos de eficiencia algoritmica.

Como es de eficiente ?

Podriamos sugerir cambios en el procedimiento para mejorar su eficiencia ?

Para responder a estas preguntas volvamos a ver como el sistema ha encontrado la solucion al problema del laberinto.

El tablero representa un laberinto, con una entrada y una salida, y sobre el tablero tenemos una unica ficha.

El objetivo del juego es llevar la ficha desde la entrada hasta la salida.

La ficha puede desplazarse a una distancia de una sola casilla cada vez, y este es el unico movimiento valido.

Y lo que queremos saber es,

cual es el camino que la ficha debe recorrer para ir desde la entrada hasta la salida ?

La entrada la hemos situado en la casilla 'A1', y la salida en la casilla 'H8'. La distancia que separa la entrada de la salida es de 7 casillas.



La ficha solo puede moverse una casilla cada vez, por lo tanto, para alcanzar la salida seran necesarios, al menos, 7 movimientos.

7 movimientos como minimo, y solo en el caso de que encuentre el camino mas corto.

Pero pueden ser mas, dependiendo del metodo que utilice para explorar el tablero.



Como ejemplo se muestran 2 posibles caminos.

Recorrer el primero requerira de 14 movimientos, mientras que recorrer el segundo requerira de 56 movimientos.

Recordemos ahora como ha resuelto el problema.

Hemos creado un individuo, el primero de su especie, representando el estado inicial del problema.

El tablero con una ficha colocada en la entrada del laberinto, 'A1'

Todos los individuos de esta especie, cumplirán con un ciclo de vida que consiste en,

nacer, reproducirse y morir.

Cuando hemos dotado de vida a este primer individuo, el ha llevado a cabo las etapas de su ciclo de vida. Se ha reproducido, generando todos los hijos posibles, e inmediatamente después ha muerto, eliminándose del sistema.

Al cumplir con su ciclo de vida, ha iniciado un proceso mas complejo, que va mas allá de su propia vida, y que continuara aun cuando el ya no forma parte del sistema.

El proceso que ha desencadenado con su existencia le sobrevivirá.

Y sucederá lo siguiente...

Los posibles hijos de ‘A1’ ocuparan las posiciones marcadas con X. Todos ellos nacerán con la primera generación, y entre ellos ‘B2’.

Los posible hijos de ‘B2’ ocuparan las posiciones marcadas con X. Todos ellos nacerán con la segunda generación, y entre ellos ‘C3’.



Los posibles hijos de ‘C3’ ocuparan las posiciones marcadas con X. Todos ellos nacerán con la tercera generación, y entre ellos ‘D4’.

Los posibles hijos de ‘D4’ ocuparan las posiciones marcadas con X. Todos ellos nacerán con la cuarta generación, y entre ellos ‘E5’.

Los posibles hijos de ‘E5’ ocuparan las posiciones marcadas con X. Todos ellos nacerán con la quinta generación, y entre ellos ‘F6’. 

Los posibles hijos de ‘F6’ ocuparan las posiciones marcadas con X. Todos ellos nacerán con la sexta generación, y entre ellos ‘G7’.



Los posibles hijos de ‘G7’ ocuparan las posiciones marcadas con X. Todos ellos nacerán con la séptima generación, y entre ellos ‘H8’.



Este individuo, ‘H8’, representa el estado final que queríamos alcanzar, y tiene la respuesta al problema, el camino que va desde la entrada hasta la salida del laberinto.

El sistema ha encontrado la solución en la séptima generación.

Nos preguntábamos como podemos mejorar el procedimiento que utiliza el sistema para hacerlo mas eficiente.

Veamoslo.

Con cada generación la ficha realiza un movimiento, una jugada valida.

Hemos establecido que una jugada valida consiste en desplazar la ficha a una distancia de una casilla.

También hemos establecido que la distancia entre la entrada y la salida es de 7 casillas, y por lo tanto, no puede ser recorrida en menos de 7 movimientos.

El sistema ha alcanzado la salida en la séptima generación, es decir, con 7 movimientos.

Y si como hemos dicho, esa distancia no puede ser recorrida con menos movimientos, entonces, la eficiencia del sistema no puede ser mejorada.

lo que ha sucedido es que, el sistema con cada generación, avanza un paso en la dirección correcta.

No se pierde, no duda, no se equivoca.

Se mueve hacia el objetivo como si supiera adonde esta.

No existirá jamas un sistema mas eficiente que este, porque el problema no puede ser resuelto con un menor numero de pasos.

Capitulo 6

# POTENCIA

En este capitulo nos proponemos hacer una estimación de la potencia de calculo que un sistema de este tipo podría tener.

La potencia puede ser definida como la cantidad de trabajo entregada por unidad de tiempo. Pero esta definición no es suficiente para medir la potencia de un sistema de computo. Podemos pensar en el numero de operaciones que un procesador puede realizar por unidad de tiempo. Pero eso no es exactamente lo que queremos saber.

La velocidad esta bien, es un buen indicador, pero ademas, nos gustaría saber que tipo de problemas es capaz de abordar. De modo que, en cuanto a potencia, queremos saber,

* cual es la complejidad de los problemas que puede resolver,
* cual es la complejidad de las operaciones que puede realizar por unidad de tiempo, y finalmente,
* a que velocidad puede realizar el calculo.

Para iniciar la discusión, veamos como podemos determinar de forma aproximada, cual podría ser la complejidad de un problema.

## Complejidad de un problema

No todos los problemas son iguales. Algunos de ellos podrán resolverse con facilidad, otros serán mas difíciles de resolver, mientras que otros podrían considerarse inabordables o incluso irresolubles. Así podríamos clasificar a los problemas de acuerdo con su complejidad.

Y cuando hablamos de complejidad, un buen indicador del grado de complejidad de un problema, es el numero de decisiones que deben tomarse para encontrar la solución.

Podemos ver a una decisión como, una bifurcación en el camino.

por donde debo continuar ?

En lo que sigue pensaremos que, la complejidad de un problema aumenta con el numero de decisiones que deben tomarse para resolverlo.

Las decisiones tienen coste. Con cada decisión el camino se bifurca y a medida que aumenta el numero de decisiones que deben tomarse aumentan el numero de caminos que debemos recorrer para encontrar la solución al problema.

Rápidamente el numero de caminos tiende a infinito, o dicho de otro modo, el numero de caminos se vuelve demasiado grande para recorrerlos todos con el tiempo y recursos disponibles.

Digamos que queremos saber como ganar una partida de ajedrez.

Partiendo del estado inicial, esto es, el tablero con las fichas en su posición de salida, cuantas posibles jugadas pueden hacerse ?

Desde la posición inicial, puede moverse alguno de los peones o alguno de los caballos.

Y esto significa, 20 posibles movimientos diferentes.

Cual sera el mejor de todos ellos ?

Tendré que probarlos todos para decidir.

Solo para comenzar la partida, desde el punto de salida pueden verse 20 posibles caminos.

Cuantos movimientos diferentes podrían hacer las negras para responder a esta primera jugada de las blancas ?

Otros 20.

Esto creara 20 por 20 = 400 escenarios diferentes.

Y solo se han movido 2 fichas, una blanca y una negra.

Y con cada nuevo movimiento el numero de posibles caminos se multiplica.

Cual de todos estos caminos me llevara al estado final deseado ?

Cual de todos ellos me llevara a ganar la partida ?

Tendré que recorrerlos todos para saberlo. Resolver este problema requerirá recursos como tiempo y memoria.

Tiempo, para recorrer todos los posibles caminos, y memoria, para recordarlos todos y poder compararlos.

Solo así sabre cual es el mejor.

Las decisiones, incrementan la complejidad del problema, y los recursos necesarios para resolverlo.

Y eso se traduce en un incremento en el coste.

Podemos decir que las decisiones son el problema, y cuantas mas decisiones, mayor es el problema.

Los problemas se vuelven mas complejos a medida que aumenta el numero de decisiones que deben tomarse para resolverlo.

Cuando hablamos de computadoras, las mas potentes serán las maquinas que llamamos

"super-computadoras".

Estas maquinas están diseñadas para realizar una gran cantidad de operaciones por unidad de tiempo. Disponen también de mucha memoria para poder almacenar y utilizar una gran cantidad de información.

En comparación, el sistema de calculo basado en especies de seres vivientes que estamos analizando, como sera de potente?

Sera comparable en potencia a los super-computadores ?

Podrá entregar resultados a mayor o menor velocidad ?

Podrá abordar problemas de mayor o menor complejidad ?

Para responder a estas preguntas tengamos presente lo que hemos dicho acerca de las decisiones.

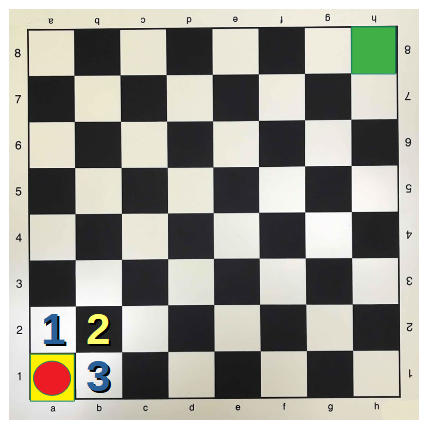
* las decisiones pueden verse como una bifurcación en el camino
* el numero de decisiones es un buen indicador de la complejidad
* las decisiones incrementan la complejidad

Con esto en mente, volvamos a ver como ha sido resuelto el problema del laberinto, y en especial, cuantas decisiones han debido ser tomadas, o cuantos diferentes caminos posibles existen en ese problema.

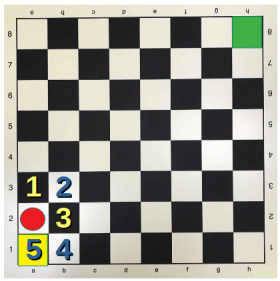
La posición de salida, el comienzo del juego, esta en 'A1'. Cuantos posibles caminos se ven desde este punto ?

Desde 'A1', pueden verse 3 posibles caminos.

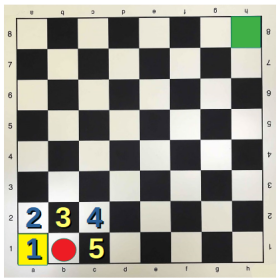
Podria ir por 'A2', 'B1', o 'B2'.



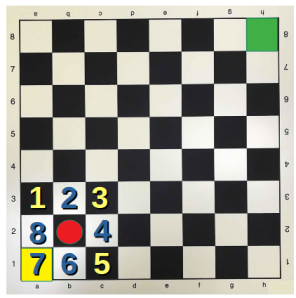
Iniciando la marcha por ‘A2’, encontrara 5 nuevos caminos posibles.



Iniciando la marcha por ‘B1’, encontrara 5 nuevos caminos posibles.



Iniciando la marcha por ‘B2’, encontrara 8 nuevos caminos posibles.



Y en cualquiera de estas 18 posibilidades, se encontrara con la necesidad de decidir nuevamente entre varias opciones posibles para seguir adelante. Y lo mismo sucederá con cada paso que avance en cualquier dirección.

Cuantas decisiones deberán tomarse hasta encontrar la solución ?

El numero de decisiones que deberán tomarse para encontrar la solución crece de forma exponencial, es decir que tendera a infinito rápidamente.

Como crece de rápido ?

Aproximadamente en un dígito cada vez que la ficha avanza una casilla. Veremos esto con mas detalle mas adelante cuando hablemos de costes.

Existen distintas soluciones al problema, así como existen distintos caminos que unen la entrada con la salida del laberinto, unos caminos serán mas cortos que otros. El camino mas corto se compone de 7 casillas, y esta sera la primera solución que encontrara el sistema.

Observad que para avanzar una casilla desde la entrada exiten 3 posibles caminos, 3 decisiones (un dígito)

Para avanzar una segunda casilla, deberán tomarse, 3 + 5 + 5 + 8 decisiones, esto es 21 (2 dígitos)

El calculo se complica a partir de aquí, pero adelantare que para recorrer 7 casillas deberán tomarse alrededor de 2 millones de decisiones.

Para recorrer una casilla mas, el numero de decisiones se habrá incrementado hasta los 16 millones aproximadamente.

Y si quisiéramos recorrer 100 casillas, el numero de decisiones que habría que tomar seria aproximadamente un dígito seguido por 100 ceros, y este es un numero difícil de expresar con palabras.

En los computadores que utilizamos a diario el método utilizado para resolver problemas de este tipo seria explorar de forma secuencial los diferentes caminos posibles, y después de recorrerlos todos, compararlos para decidir cual de todos es el mejor.

Pero para recorrerlos todos uno tras otro, se necesita tiempo.

Y cuando el numero de caminos posibles tiende a infinito, el tiempo necesario para recorrerlos todos también, tiende a infinito.

Y cuando el tiempo necesario fuera mas del disponible, entonces el problema debería considerarse, inabordable.

Para compararlos sera necesario almacenar la información que vamos recogiendo a medida que los recorremos.

Y para esto es necesario tener memoria.

Una vez mas, cuando el numero de caminos posibles tiende a infinito, la memoria necesaria para recordarlos todos también, tiende a infinito.

Para que los computadores electrónicos puedan abordar problemas de este tipo sera necesaria una estrategia que evite recorrer todos los caminos de principio a fin, porque no sera posible dotar a una maquina de una cantidad de memoria infinita.

Y en cuanto a la complejidad de los problemas.

Un sistema inteligente basado en familias de seres vivientes, que clase de problemas podría resolver?, o dicho de otra forma, problemas de que complejidad podría abordar?

Veamos otra vez como ha resuelto el problema del laberinto.

Inicialmente creamos un individuo que representa el estado inicial. Un tablero con una ficha en la entrada del laberinto, 'A1'.

Desde esta posicion, existen 3 caminos posibles, que empiezan en 'A2', 'B1', 'B2'.

Cual de ellos sera el mejor ?

Como decidir por cual de ellos continuar la exploración ?

El individuo genera tantos hijos como caminos posibles existen, coloca a un nuevo individuo en la entrada de cada camino, y encarga a cada uno de ellos la exploración de 1 único camino.

El sistema no decide.

Simplemente recorre todos los caminos a la vez, en paralelo. No importa cuantos nuevos caminos aparezcan, la maquina continuara avanzando como si no existiera mas que un único camino.

Hasta llegar a la salida habrán aparecido una gran numero de caminos diferentes posibles. Habra sido necesario tomar una gran cantidad de decisiones.

Para un computador que explora los diferentes caminos de forma secuencial, el coste en tiempo y memoria seria elevado.

Para un sistema inteligente basado en familias de criaturas vivientes, las decisiones no incrementaran el tiempo necesario para resolver el problema.

La familia ha encontrado la solución después de 7 generaciones. En 7 movimientos ha llegado a la salida. Como si no existiera mas que un único camino.

El efecto es el de no existir la complejidad. La forma en que el sistema trabaja disuelve la complejidad, haciendo que las decisiones, no existan.

La exploración se realiza en paralelo, de forma concurrente, como si cada individuo fuera una nueva maquina que inicia la exploración por su cuenta, independientemente del resto de la colonia.

Y el trabajo de cada individuo no encontrara complejidad, porque el individuo no recorrerá todo el camino, el individuo no tendrá que enfrentarse a todas las decisiones que se presentaran en el futuro, porque el trabajo de un individuo se reduce a dar un único paso simple.

Así, todos los problemas son simples, triviales, independientemente de su complejidad intrinseca. El problema se desmenuza en pequeños problemas triviales, y se crean tantas computadoras como pasos elementales tenga el problema. Y cada computadora realizara una única operación simple.

Si pensamos en los individuos de la colonia como en computadores, entonces, el procesador no necesita conocer un gran conjunto de operaciones, porque solo realiza una única operación, siempre la misma, que consiste en generar los siguientes estados posibles.

Tampoco es necesario que sea capaz de realizar una gran cantidad de cálculos por unidad de tiempo, porque solo realizara una única operación, una.

En el caso de la memoria, no es necesario que sea capaz de recordar todos los infinitos posibles caminos en los que pueda descomponerse el problema para tomar una decisión después de haberlos recorrido todos.

Eso podría requerir una gran cantidad de memoria, en cambio, solo necesita recordar el camino que su familia ha recorrido hasta ese momento.

Y eso es un único camino finito.

Volviendo a la comparación entre los computadores electrónicos y los sistemas inteligentes basados en familias de seres vivientes. (en adelante llamaremos a estos sistemas, N.I.S., o Natural Intelligence Systems).

Hemos establecido que, la complejidad de los problemas no es importante para un “N.I.S”, y podría abordar problemas de cualquier complejidad, entonces en este sentido, no puede ser superado.

Hemos establecido también que la memoria tampoco es un problema, porque un individuo no tiene que recordar todas las posibles ramificaciones del calculo, sino solo su camino, esto hace que la cantidad de memoria necesaria siempre sea discreta, independientemente de la complejidad que pueda tener el problema.

Ademas, en el momento de crear a un hijo, ya sabe cuanta memoria necesitara, porque ya tiene los datos que debe almacenar, y la memoria una vez creada no se modificara. Esto le confiere la capacidad de abordar problemas de complejidad que tiende a infinito, con una cantidad de memoria discreta. De modo que, en este aspecto, tampoco puede ser superado.

Veamos ahora la velocidad de proceso.

En el caso de un computador electrónico, una medida de la velocidad de proceso nos la daría el numero de operaciones que puede ejecutar por unidad de tiempo.

En el caso de un N.I.S, no podemos medirlo con la misma facilidad, sin embargo podemos aproximarlo.

Hemos visto ya, en el capitulo dedicado a la “Eficiencia”, que resolverá los problemas en el menor numero de operaciones posibles, y por lo tanto, en este sentido, no puede ser superado, pero veamos a que velocidad ejecutaría esas operaciones y cuanto tiempo sera necesario para resolver el problema.

Un N.I.S. resultara mas rápido que un computador electrónico, o mas lento ?

Un NIS ejecutara un paso en el proceso de resolución del problema con cada generación.

Es en la reproducción cuando la información se modifica, y dicho asi, podría parecer lento.

En el caso de algunas especies de mamíferos, el tiempo parece enorme, una generación en la especie humana requiere unos 20 anyos de media.

Existen especies que se reproducen mas rapidamente, como las bacterias que podrían producir una nueva generación cada 15 minutos.

Pero digamos que utilizamos humanos, con una generación cada 20 anyos.

Y por otro lado, tenemos un computador electrónico que puede ejecutar billones de operaciones por segundo.

A simple vista puede parecer mas rápido el computador electrónico, sin embargo, lo realmente importante es,

Cual de los 2 resuelve el problema antes ?

Cual de los 2 encontrara la solución en menos tiempo ?

Recordemos que, un sistema basado en familias de seres vivientes, un N.I.S, recorrerá todos los caminos a la vez, en paralelo.

Y aun cuando, los computadores electrónicos pueden utilizar cierto grado de concurrencia, esta siempre tendrá un limite.

En el caso de un N.I.S, la concurrencia tiende a infinito, de modo que, no importa cuantos caminos posibles existan, los recorrerá todos en el mismo tiempo que recorre uno.

Esto no puede ser igualado por un computador electrónico.

Y aunque al principio puede parecer lento, el numero de individuos crece de forma exponencial con el tiempo.

Al principio, partimos de un único individuo, pero el numero de individuos de la familia crece rápidamente, y en poco tiempo el numero de individuos de la familia tendera a infinito.

De modo que, después de un tiempo de "puesta en marcha", la velocidad de proceso tiende a infinito, y sera difícilmente superada.

No todas las especies requerirán 20 anyos para producir una generación, pero aun así, cuando hablamos de problemas de una complejidad tal que un computador electrónico necesitaría muchos millones de anyos para encontrar una solución, en ese caso, un N.I.S. tendria ventaja.

Cuantos individuos existen hoy en el mundo, sumando todos los que forman las diferentes especies de criaturas vivientes ?,

Las plantas, los peces, las bacterias, los insectos, las aves, los reptiles, todos ellos, todas los ejemplares de todas las cosas que están vivas.

Todos ellos son descendientes del primer organismo que ha cobrado vida. Y por lo tanto, todos forman parte del mismo proceso.

Todos ellos buscan respuesta a las mismas preguntas, pero todos ellos por caminos diferentes.

Todos los individuos son diferentes, no existen 2 iguales para asegurar que no hay 2 realizando el mismo trabajo.

Y todos ellos inician una nueva búsqueda, creando una nueva familia que buscara la respuesta por su propio camino.

Cada individuo que nace representa un paso hacia adelante en el proceso, una nueva operación sobre los datos.

Y todos ellos representas operaciones diferentes. Dos operaciones jamas darán el mismo resultado.

Entonces, cuantas operaciones por segundo podría ejecutar esta maquina viviente ?

Tantas como nuevos individuos nacerán en 1 segundo.

Como referencia dire que, dentro del organismo de un único ser humano, viven varias decenas de billones de bacterias.

Un computador electrónico podría realizar billones de operaciones por segundo.

Ahora, mirando alrededor, e intentando adivinar cuantos individuos nacerán en un segundo en la tierra, es evidente que ningún computador electrónico conseguirá nunca competir con la potencia de calculo que la vida tiene.

Un computador electrónico, no puede abordar problemas del universo real, cuando este no tiene limites.

Solo podrá abordar problemas que pertenezcan a un dominio discreto, acotado.

Las principales limitaciones de los computadores electrónicos se encuentran en

la memoria disponible,

y el tiempo disponible.

La realidad es demasiado grande, inabarcable.

La vida, sin embargo, como sistema inteligente, debe ser capaz de realizar cálculos, que le permitan, entender, aprender, mejorar, adaptarse, sobrevivir, dentro de un dominio que a priori, se muestra,

infinito.

Capitulo 7

# cOsTE

Continuando con el análisis comparativo del sistema veremos ahora otra de sus características.

El coste.

Intentaremos establecer de forma aproximada cual podría ser el coste de un sistema inteligente de este tipo.

Por un lado nos gustaría estimar cual podría ser el coste de la maquina.

Y por otra parte, quisiéramos saber cual podría ser el coste de resolver un problema utilizando esta tecnología.

Es natural que un problema complejo requiera una mayor consumo de recursos que un problema simple, lo que tendría impacto en el coste.

Por ejemplo, un problema mas complejo podría requerir,

mayor consumo de energia, de tiempo, de memoria, un hardware mas especializado, infraestructuras mas costosas, etc.

En algunas arquitecturas, el coste de la maquina es, hasta cierto punto independiente del coste de resolver un problema.

En esta arquitectura, sin embargo, en un sistema de inteligencia natural, un N.I.S., el coste de la maquina y el coste de resolver un problema estan estrechamente relacionados.

Si pensamos en que por cada paso elemental en el calculo crearemos un nuevo individuo, entonces esta claro que cuanto mas cálculos requiera la resolución del problema, mayor sera la colonia de individuos necesaria para encontrar la solución.

Pero veamoslo con mas detalle.

Veamos como el sistema a encontrado la solución al problema del laberinto, y cual podría ser el coste de resolver ese problema.

Recordemos el problema.

Se trata de averiguar cual es el camino que una ficha debe recorrer en un tablero de juego cuadriculado, para ir desde una casilla determinada hasta otra.

Para esto hemos creado una especie de seres vivientes a la medida del problema, en donde,

los individuos de la especie representan los estados del juego.

Los individuos, nacen se reproducen y mueren.

Y en la reproducción se generaran los siguientes estados posibles, que ese obtienen realizando jugadas validas.

La maquina tiene una estructura celular, en donde el individuo es la unidad estructural o célula.

La maquina esta formada por un tejido de individuos, formando una colonia en forma de red,

y no existe ningún otro componente, solo individuos.

Los individuos se reproducirán, creando mas y mas individuos, hasta que se encuentre la solución al problema.

Y cuanto mas complejo sea el problema, mas individuos deberán ser creados.

Hasta ahora hemos visto el funcionamiento de la maquina sobre el papel, es decir, de forma teórica.

En el ejemplo del laberinto hemos creado individuos en nuestra imaginación, y solo algunos de ellos, no todos los que habrían sido creados, sino solo los relevantes para el análisis.

Pensaremos ahora en como podemos llevar a cabo estas ideas en el mundo real.

En como podríamos construir una maquina física, real.

Esto nos ayudara a establecer, al menos de forma aproximada, cual podría ser el coste de construir una maquina de estas características.

Cualquier implementacion que construyamos debería incluir las siguientes características:

-la maquina esta compuesta por individuos

-los individuos cumplirán con un ciclo de vida que consiste en : nacer-reproducirse-morir

-los individuos contienen un conjunto inmutable de datos único (ADN)

-en la reproducción los datos son modificados para crear otro conjunto de datos único que define al individuo hijo

Digamos que para implementar a los individuos utilizamos bacterias.

Digo bacterias porque intento reducir los costes.

Y construir un mamifero, seguramente, tendra un coste muy superior al de una bacteria.

Las bacterias parecen una solución económica cuando hablamos de seres vivos, y sin embargo, el coste sigue siendo elevado.

No puedo cuantificar el coste exacto, pero puedo afirmar sin temor a equivocarme que,

diseñar y construir una bacteria esta fuera de mi alcance en este momento.

Veamos una solución mas económica.

Digamos que grabo la informacion de un individuo en un disco de plastico como un CD.

Un CD cumplirá la función del ADN almacenando la informacion.

Y grabare un CD por cada individuo que sea necesario en el proceso de calculo. Estos individuos no están vivos como seria deseable, son de plastico, por lo tanto no podrán reproducirse por si solos, y tendré que simular la reproducción manualmente,

sin embargo, esta solución es mucho mas económica que la creación de seres vivientes, y tiene la ventaja de que podremos cuantificar en dinero el valor de cada CD,

lo que nos ayudara en el análisis de costes.

Veamos ahora como el sistema ha encontrado la solución al problema del laberinto.

Para iniciar el proceso hemos creado un primer individuo que representa el estado inicial del juego.

Es decir, una ficha sobre el tablero colocada en la entrada del laberinto, que hemos situado en la casilla 'A1'.

Cuando llegue el tiempo de producir la siguiente generación, el individuo se reproducirá creando todos los hijos posibles.

Todos los hijos son diferentes, y representan movimientos validos desde el estado del juego que representa el padre.

Los hijos representan los estados del juego alcanzables después de realizar un único movimiento.

Después de crear a toda su descendencia el padre ya no es necesario y es eliminado.

En la siguiente generación, los nuevos individuos creados a su vez se reproducirán creando todos los hijos posibles, moviendo el juego hacia el siguiente estado.

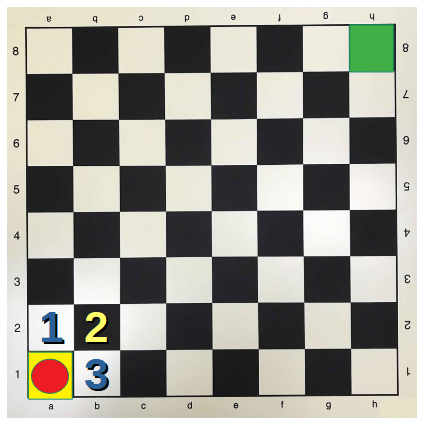
Y generación tras generación, el proceso continuara avanzando hacia su resolución.

Recordemos también, que han sido necesarias 7 generaciones para encontrar la solución.

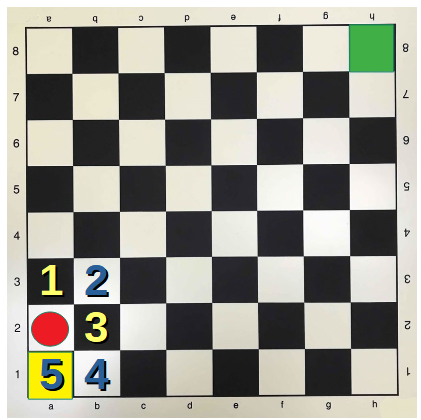
Cuantos individuos han sido creados durante el proceso ?

Para realizar este calculo debemos tener en cuenta que

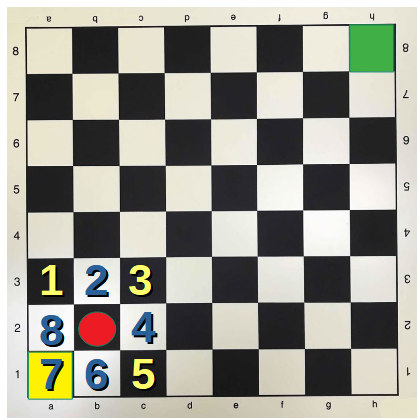
Cuando la ficha se encuentra en un angulo del tablero, son posibles 3 movimientos diferentes, y por lo tanto, un individuo representando un estado de este tipo, tendría 3 hijos.



En otro caso, cuando la ficha se encuentra en un borde del tablero, son posibles 5 movimientos validos, y por lo tanto un individuo de este tipo tendría 5 hijos.



El ultimo caso posible en este problema, es cuando la ficha se encuentra en una casilla alejada de los bordes del tablero, este sera el caso mas común, la gran mayoría de las casillas del tablero son de este tipo, y en esta situación serán posibles 8 distintos movimientos validos, y por lo tanto, un individuo que representa un estado de este tipo tendrá 8 hijos.



Sin embargo el calculo en detalle es complicado y no es importante tener una cifra exacta en este momento, sino que, sera suficiente con tener una idea aproximada, de modo que simplificaremos el calculo. Porque lo que necesitamos ahora es hacernos una idea de como crecen las poblaciones de seres vivientes.

Con este objetivo, tomaremos como base del calculo el caso mas común, es decir, la ficha esta alejada de los bordes del tablero.

Imaginaremos que el tablero no tiene bordes sino que se extiende sin fin en todas direcciones.

En esta situación, una ficha en cualquier posición podría realizar 8 movimientos diferentes.

Y un individuo que representa uno de estos estados tendrá, por lo tanto, 8 hijos.

Veamos ahora como crece la descendencia de un individuo, es decir, cuantos otros serán creados a partir de un único individuo con el paso de las generaciones.

Con la primera generación, nacen 8 hijos.

En la segunda generación, los 8 hijos tendrán 8 hijos cada uno, y por lo tanto, nacen 64 nuevos individuos con la segunda generación.

En la tercera generación, los 64 nuevos individuos tendrán 8 hijos cada uno, y por lo tanto, nacen 512 nuevos individuos con la tercera generación.

En la cuarta generación, los 512 nuevos individuos tendrán 8 hijos cada uno, y por lo tanto, nacen 4096 nuevos individuos con la cuarta generación.

El numero de individuos que son creados en cada generación no se mantiene constante, sino que crece mucho, de una generación a la siguiente.

Hemos dicho que serian necesarias 7 generaciones para encontrar la solución al problema del laberinto.

Pues bien, en la primera generación nacen 8 individuos, mientras que en la séptima generación nacen

1 \* 8 \* 8 \* 8 \* 8 \* 8 \* 8 \* 8, aproximadamente unos 2.000.000

Por este motivo, diremos que el crecimiento de la población es EXPONENCIAL.

Porque podemos calcular el numero de individuos que nacen en cada generación como una potencia.

En este caso es 8 elevado a 7.

Observad que estos no son todos los individuos que han sido creados después de 7 generaciones, solo son los que nacen en la séptima.

El numero total es, (8 elevado a 7) + (8 elevado a 6) + (8 elevado a 5) + (8 elevado a 4) + (8 elevado a 3) + (8 elevado a 2) + (8 elevado a 1)

Lo importante aquí es que son muchos en pocas generaciones.

Cuantos nacen en la octava generación ? pues 8 elevado a 8, que serán aproximadamente unos 16 millones.

La idea que quiero transmitir es que las poblaciones de seres vivientes crecen exponencial mente con el tiempo.

En el caso del problema que hemos planteado como ejemplo, solo han sido necesarias 7 generaciones para encontrar la solución.

Pero que pasaría con problemas mas complejos que requieran mas cálculos ?

Que pasaría si hubieran sido necesarias 100 generaciones para encontrar la respuesta ?

Pues que en la generación numero 100 el numero de individuos que nacerán se contarían como un numero de 100 dígitos.

Intentad expresar un numero como ese con palabras, para entender la magnitud.

Volvamos ahora al punto adonde yo intento materializar el sistema, construirlo físicamente.

He propuesto grabar a cada individuo en un disco de plástico como un CD, buscando una solución económica.

Así, habrían sido necesarios 2 millones de CDs para recorrer 7 casillas del tablero. Esto nos permite hacernos una idea del elevado coste del sistema.

Y cuantos CDs habrían sido necesarios para recorrer 10 casillas ?

Pues, 8 elevado a 10, es decir, alrededor de 1000 millones de CDs.

Solo incrementando en 3 casillas la distancia a recorrer hemos pasado de 2 millones a 1000 millones.

Y si las casillas a recorrer fueran 20 ?

En ese caso, 8 elevado a 20 es 1.152.921.504.606.846.976

El problema del laberinto es extremadamente simple.

Que pasaría con un problema mas complejo ? Cuantos individuos deberíamos construir para el caso de una partida de ajedrez ?

Muchos.

Porque el numero de individuos necesarios (igual que el coste del sistema) se incrementa exponencial mente cuando aumenta la complejidad del problema que se quiere resolver.

No olvidemos que el sistema esta vivo.

Los individuos son seres vivientes.

He utilizado la solución con CDs porque es económica.

Un CD es plástico, es barato.

Me resulta imposible estimar el coste que podría tener crear una especie de seres vivientes.

Quiero que se vea claramente el contraste.

Porque lo que intento decir es que

no importa lo económica que pueda ser la solución que encontremos,

aun cuando el coste de una unidad tiende a cero,

el coste del sistema tiende a infinito rápidamente.

Capitulo 8

# redundancia

La redundancia es una técnica ampliamente utilizada en ciencias de la computación.

Consiste en hacer replicas de componentes criticos, software o hardware, o de informacion importante, con el objetivo de incrementar la disponibilidad y protegerlos contra fallos o perdidas.

Veamos un ejemplo.

Digamos que tenemos un negocio que vende productos en Internet.

Para vender nuestros productos tenemos un sistema compuesto por una tienda (comercio electrónico), que esta alojada en un computador que llamamos "Servidor".

El servidor esta alojado en un edificio de oficinas.

El negocio va bien hasta que un día un terremoto destruye el edificio, y con el nuestro servidor.

Todo nuestro negocio estaba en ese servidor, la plataforma software que mostraba la tienda en Internet, la informacion de los clientes, proveedores, las ventas, en fin, todo.

Destruido el servidor, todo se ha perdido y ya no tenemos negocio.

Para evitar situaciones como esta, podríamos replicar el sistema, y colocar copias de la tienda en otros servidores, en otras ciudades.

Digamos que tenemos un servidor en San Francisco, otro servidor en Orlando y otro servidor en Cleveland.

Los 3 proporcionan los mismos servicios y trabajan juntos como una unidad. Y si alguno de ellos tuviera algún problema, podría ser eliminado de la red sin que el negocio se vea afectado.

Los clientes que compran nuestros productos en la tienda no saben cual de los servidores les esta atendiendo.

Con esta estrategia, habiendo redundado el sistema, la probabilidad de que un desastre natural destruya nuestro negocio es mas baja.

aragraphs is a **“Level 1”** heading. Everything has been professionally organized and styled through Ms. Word’s style-sheets. You can start your sections or chapters with this Level 1 heading.

## Perfect formatting for self-authors

If you love writing books then you should definitely use this easy to use yet nicely formatted book layout for your next book. The heading above this paragraph is **“Level 2”** heading. Everything is formatted using predefined styles so instead of making any manual change in the formatting, you should modify the associated styles to let your changes effectively apply at

ooking Si ‘A1’ tiene todos los hijos posibles,

estos ocuparían las posiciones (marcadas con X)

que se muestran en el dibujo 7. for a free template of book for writing a book? You can use this professionally formatted Word book template for creating a **nicely formatted book** for online submission or for printing through any self-publishing company. This book template is especially created for preparing small books in 5 x 8 paperback. You will definitely find this Word template useful in formatting your book. The heading above these two paragraphs is a **“Level 1”** heading. Everything has been professionally organized and styled through Ms. Word’s style-sheets. You can start your sections or chapters with this Level 1 heading.

## Pcomplejidad de un problemaspuesta en marcha erfect formatting for self-authors

If you love writing books then you should definitely use this easy to use yet nicely formatted book layout for your next book. The heading above this paragraph is **“Level 2”** heading. Everything is formatted using predefined styles so instead of making any manual change in the formatting, you should modify the associated styles to let your changes effectively apply at every instance of that formatting. Feel free to personally use this outstanding book template in any of your fiction or non-fiction work. This is more than a blank book template in Microsoft Word; this is a complete book design for commercial printing & publishing.

## Print-ready typesetting & layout

The layout, design, formatting, automatic table of contents –everything has already been included in the template and this is for your **Personal Use Only**. We have included every necessary section into this book writing template. The starting pages are numbered using Roman numerals. However, everything is still modifiable for you. Even if you are not very good with Microsoft Word, still by using this template, you can make your book look professional. Either write your book from scratch or copy your existing content into the template, you’ll definitely love it. Make sure you check the **“Navigation Pane”** to know about the different levels of headings and also check the **“Styles”** group for available styles. This book template is formatted into **“Sections”**. Every Section has its own properties, please also take care of these things when adding new sections into the template.

## Adding new content

For adding new chapters in this template, simply copy these dummy texts entirely and paste them at the insertion point from where you need to start a new chapter. Otherwise, you can write your content from scratch using the predefined styles in the template. Please also note that this template uses **“Different First Page”** & **“Different Odd/Even Pages”** through Microsoft Word’s sections. Insert new chapter in a separate section.

## Using chapter title on the header

If you want to write chapter title on the header instead of using author name then you can modify it on the header of the section you are working on. Using sections is easy in Microsoft Word, if you don’t know about modifying sections then **please Google** about inserting & using sections in Ms. Word, you’ll certainly find many articles on Microsoft Word’s sections.

## Our “Important Notes”

Please first also read our “Important Notes” document attached in the download package. Our free templates normally consist of dummy text throughout the document so before you submit your book for final publishing, make sure there **remains no placeholder text** in the manuscript. You’re allowed to download, modify and use our free resources for your personal use only.

## Help us by accrediting!

If you find this book template helpful in your writings then please spread the word and appreciate us by sharing these awesome and free templates with others as much as you can so that many people will also take benefits of our great free resources. Please remember that this template is for your **PERSONAL USE ONLY**. For any questions, please write to us at the following addresses:

admin@usedtotech.com

https://usedtotech.com

\*\*\*

## This is level 2 heading

You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.