



# **INSTITUTO POLITÉCNICO NACIONAL**

Centro de Innovación y Desarrollo Tecnológico en Cómputo

## **“ESTUDIO DE PROCESAMIENTO PARALELO COMPUTACIONAL APLICADO A CÉLULAS DEL SISTEMA INMUNOLÓGICO”**

Tesis que presenta:

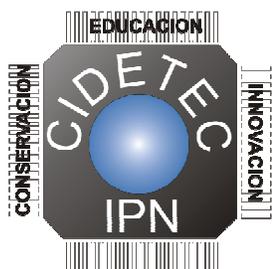
**Ian Ilizaliturri Flores**

Para obtener el grado de Maestro en Ciencias  
en Tecnología de Cómputo

Directores de Tesis:

M. en C. Jesús Antonio Álvarez Cedillo

Dr. José Correa Basurto



México D.F. Diciembre del 2007

**Este trabajo fue realizado en su mayor parte en las instalaciones del CIDETEC y ESM, bajo la dirección del M. en C. Jesús Antonio Álvarez Cedillo y el Dr. José Correa Basurto.**

### **Centro de Innovación y Desarrollo Tecnológico en Cómputo – IPN**

Av. Juan de Dios Bátiz s/n Casi esquina Miguel Othón de Mendizábal, Unidad Profesional Adolfo López Mateos, Edificio CIDETEC. Colonia Nueva Industrial Vallejo, Delegación Gustavo A. Madero, C.P. 07700, México D.F.

Dirección electrónica: [cidetec@ipn.mx](mailto:cidetec@ipn.mx)

URL: [www.cidetec.ipn.mx](http://www.cidetec.ipn.mx)



### **Escuela Superior de Medicina – IPN**

Calle Plan de San Luis y Díaz Mirón, Unidad Profesional Lázaro Cárdenas, Colonia Casco de Santo Tomás, Delegación Miguel Hidalgo, C.P. 11340, México D.F.

Dirección electrónica: [esm@ipn.mx](mailto:esm@ipn.mx)

URL: [www.esm.ipn.mx](http://www.esm.ipn.mx)





# INSTITUTO POLITECNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

## ACTA DE REVISION DE TESIS

En la Ciudad de México, D.F. siendo las 15:00 horas del día 11 del mes de diciembre del 2007 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del CIDETEC para examinar la tesis de grado titulada:

ESTUDIO DE PROCESAMIENTO PARALELO COMPUTACIONAL APLICADO  
A CÉLULAS DEL SISTEMA INMUNOLÓGICO

Presentada por el alumno:

ILIZALITURRI  
Apellido paterno

FLORES  
materno

IAN  
nombre(s)

Con registro: 

A	0	5	0	0	0	7
---	---	---	---	---	---	---

aspirante al grado de:

MAESTRÍA EN TECNOLOGÍA DE CÓMPUTO

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

### LA COMISION REVISORA

  
DR. VICTOR MANUEL SILVA GARCÍA  
Presidente

  
M. EN C. JUAN CARLOS HERRERA LOZADA  
Secretario

  
M. EN C. JESÚS ANTONIO ÁLVAREZ CEDILLO  
Primer Vocal  
(Director de Tesis)

  
DR. JOSÉ CORREA BASURTO  
Segundo Vocal  
(Director de Tesis)

  
M. EN C. EDUARDO VEGA ALVARADO  
Tercer Vocal

  
M. EN C. EDUARDO RODRÍGUEZ ESCOBAR  
Suplente

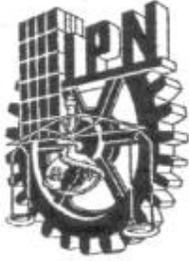
### EL PRESIDENTE DEL COLEGIO

  
DR. VICTOR MANUEL SILVA GARCÍA



S. E. P.

INSTITUTO POLITECNICO NACIONAL  
CENTRO DE INNOVACION Y DESARROLLO



**INSTITUTO POLITÉCNICO NACIONAL**  
**COORDINACIÓN GENERAL DE POSGRADO E INVESTIGACIÓN**

**CARTA CESIÓN DE DERECHOS**

En la Ciudad de México, D.F. el día 14 del mes de Diciembre del año 2007, el que suscribe Ian Ilizaliturri Flores alumno del Programa de Maestría en Tecnología de Cómputo con número de registro A050007, adscrito a el Centro de Innovación y Desarrollo Tecnológico en Cómputo, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del M. en C. Jesús Antonio Álvarez Cedillo y el Dr. José Correa Basurto, cede los derechos del trabajo intitulado ESTUDIO DE PROCESAMIENTO PARALELO COMPUTACIONAL APLICADO A CÉLULAS DEL SISTEMA INMUNOLÓGICO, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección [ferroian@hotmail.com](mailto:ferroian@hotmail.com). Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

---

Ian Ilizaliturri Flores

# **“ESTUDIO DE PROCESAMIENTO PARALELO COMPUTACIONAL APLICADO A CÉLULAS DEL SISTEMA INMUNOLÓGICO”**

## **RESUMEN.**

La mayoría de los problemas “intratables” son problemas en cómputo, que sólo pueden ser resueltos utilizando una exhaustiva búsqueda a través de todas las posibles soluciones. Sin embargo, existe cierta complejidad debido a que la búsqueda es demasiado grande para ser realizada usando determinados elementos de cómputo. Considerando la complejidad funcional y estructural de la célula biológica, la cual depende de la información almacenada en las cadenas de ADN y la facilidad de construcción de muchas copias de ellas, es posible que a partir de éstas se puedan hacer intensas búsquedas de alta complejidad las cuales los sistemas de cómputo uní-procesador no pueden resolver. Un ejemplo de este comportamiento se presenta en el sistema inmunológico; ya que encontrar la inmunoglobulina específica para un antígeno dentro de todas las posibles combinaciones de determinantes antigénicos, es muy difícil, sin embargo la célula lo realiza. Para este fin se requiere de un procesamiento de información que genere inmunoglobulinas que se acoplen a antígenos de una forma muy precisa y que además el tiempo de respuesta sea óptimo para contrarrestar algún daño al organismo.

En este trabajo se analizó al sistema inmunológico específico como un sistema de cómputo con ADN llamado splicing; esto indicaría que el sistema inmunológico se encuentra enlazado a dichas computadoras y por lo tanto a las reglas de la ciencia de la computación. Para cumplir con lo anterior se hizo un análisis bajo un modelo formal de cómputo con ADN que realiza las mismas operaciones que se producen en la recombinación somática en las células del sistema inmunológico.

***Palabras y frases claves:* cómputo molecular, cómputo con ADN, biología molecular, sistema inmunológico, procesamiento paralelo.**

# **“STUDY OF COMPUTATIONAL PARALLEL PROCESSING APPLIED TO CELLS OF THE IMMUNE SYSTEM”**

## **ABSTRACT.**

Most 'intractable' problems are problems in computation, which can only be solved using an exhaustive search through the set of possible solutions. However, there is a kind of complexity because the search is too big to be done using certain elements of computing. Considering the structural and functional complexity of the biological cell, which depends on the information stored in DNA chains and the ease of construction of many copies, it is possible that since they can do intense searches of high complexity which computer systems uni-processor not solve. An example of this behavior is presented in the immune system specific, and to find that specific immunoglobulin for an antigen in all possible combinations of antigenic determinants, it is very difficult, however the cell does. It for this purpose it is required an information processing which produces immunoglobulins to couple the antigens in a very precise way and that the response time is optimal to counter any damage to the body.

This thesis analyses the immune system as a specific computing system with DNA, as it indicates that the immune is linked to the computers and therefore the rules of computing science. To comply with this, the analysis was made under a formal model of computation with DNA named splicing that performs the same operations that occur in the recombination in somatic cells in the immune system.

***Key words and phrases: computational molecular, DNA computing, molecular biology, immunology, parallel processing.***

# AGRADECIMIENTOS

**M. en C. Jesús Antonio Álvarez Cedillo.** Por todo el apoyo que me brindo, por el interés que mostró en el desarrollo de éste trabajo y sus comentarios siempre tan objetivos.

**Dr. José Correa Basurto.** Por guiarme y brindarme las bases necesarias para la elaboración de éste trabajo y por ayudar a mí desarrollo académico.

**M. en C. Juan Carlos Herrera,** por su disposición, por sus sugerencias siempre tan objetivas y por su paciencia.

**Lic. Silvia.** Por su disposición y apoyo administrativo.

**Lic. Irma.** Por su amabilidad y su apoyo administrativo.

A mi padre **Gustavo Ilizaliturri Carmona** por la confianza que deposito en mí, por su apoyo y por sus consejos.

A mi madre **Paula R. Flores Sierra** por su apoyo incondicional y por confiar y creer en mí.

A mis hermanas **Ixtabay, Nicthe y Lol,** por sus consejos y comentarios.

# CONTENIDO

LISTA DE FIGURAS.....	I
ABREVIATURAS .....	III
GLOSARIO.....	IV
LISTA DE TABLAS .....	VI
INTRODUCCIÓN.....	1
1. INTRODUCCIÓN.....	1
2. ANTECEDENTES .....	2
3. JUSTIFICACIÓN .....	4
4. HIPÓTESIS.....	5
5. OBJETIVO GENERAL .....	5
6. OBJETIVOS PARTICULARES.....	5
7. CONTENIDO GENERAL.....	6
CAPÍTULO 1 .....	7
LA CIENCIA DE LA COMPUTACIÓN.....	7
1.1 ALFABETOS, CADENAS Y LENGUAJES .....	7
1.2 LA MÁQUINA DE TURING (MT).....	9
1.3 TEORÍA DE CÓMPUTO.....	10
1.4 TEORÍA DE LA COMPUTABILIDAD.....	10
1.5 EL PROBLEMA DE LA PARADA.....	13
1.6 TEORÍA DE LA COMPLEJIDAD COMPUTACIONAL .....	13
1.7 VARIANTES DE LA MÁQUINA DE TURING .....	16
CAPÍTULO 2 .....	17
BIOLOGÍA MOLECULAR.....	17
2.1 ESTRUCTURA DEL ADN (ÁCIDO DESOXIRRIBONUCLÉICO).....	17
2.2 LA REGULACIÓN EN LA EXPRESIÓN GENÉTICA.....	21
2.3 LAS PROTEÍNAS .....	22
2.4 ENZIMAS DE RESTRICCIÓN.....	23
CAPÍTULO 3 .....	25
CÓMPUTO BIOMOLECULAR .....	25
3.1 CÓMPUTO CON ADN .....	26
3.2 CARACTERÍSTICAS DE LAS OPERACIONES DE CÓMPUTO CON ADN.....	28
3.4 LA COMPLEJIDAD COMPUTACIONAL DE LAS COMPUTADORAS BIOMOLECULARES .....	30
3.5 LIMITACIONES DE LA COMPUTACIÓN CON ADN .....	33
3.6 MODELOS FORMALES DE CÓMPUTO CON ADN .....	35
CAPÍTULO 4 .....	37
LA INMUNOLOGÍA MOLECULAR .....	37
4.1 LA INMUNIDAD ESPECÍFICA.....	37
4.2 LA INMUNOGLOBULINA.....	40
4.3 EL PROCESO DE SELECCIÓN DURANTE LA MADURACIÓN DE LOS LINFOCITOS .....	45
4.4 LA HIPÓTESIS DE SELECCIÓN CLONAL .....	46
4.5 LA INMUNIDAD FRENTE A LOS TUMORES .....	47

<b>CAPÍTULO 5</b> .....	<b>49</b>
<b>RESULTADOS Y CONCLUSIONES</b> .....	<b>49</b>
<b>5.1 SISTEMA SPLICING</b> .....	<b>52</b>
<b>5.2 EL SISTEMA INMUNOLÓGICO CARACTERIZADO COMO UN SISTEMA SPLICING</b> .....	<b>59</b>
<b>REFERENCIAS</b> .....	<b>65</b>
<b>ANEXO I – LA IMPLEMENTACIÓN DE LA FUNCIÓN DE TRANSICIÓN</b> .....	<b>70</b>
<b>ANEXO II. EL PROBLEMA DE SATISFACCIÓN BOOLEANA (SAT)</b> .....	<b>71</b>
<b>ANEXO III. MÁQUINA DE TURING NO DETERMINISTA (MTND)</b> .....	<b>72</b>
<b>ANEXO IV. CLASES DE COMPLEJIDAD COMPUTACIONAL</b> .....	<b>75</b>
<b>ANEXO V. TÉCNICAS MOLECULARES Y OPERACIONES PARA MANIPULAR EL ADN</b> .....	<b>76</b>

# LISTA DE FIGURAS

Figura 1: Pato de Vaucanson.....	2
Figura 2: Bacteria Escherichia coli.....	3
Figura 1.1: Los componentes de la MT.....	9
Figura 1.2: La MT como calculadora de funciones.....	10
Figura 1.3: Una MT que acepta un lenguaje.....	11
Figura 1.4: La clasificación de los lenguajes de acuerdo a una MT que los acepta.....	11
Figura 1.5: Una MT que decide respecto a un lenguaje.....	11
Figura 1.6: La relación entre la MT, lenguajes y computabilidad.....	12
Figura 1.7: La clasificación de los problemas de acuerdo a la computabilidad.....	14
Figura 1.8: Los crecimientos de algunas funciones.....	16
Figura 2.1: Origami ADN.....	18
Figura 2.2: El nucleótido.....	19
Figura 2.3: Enlaces entre nucleótidos.....	19
Figura 2.4: Molécula de ADN de cadena sencilla.....	20
Figura 2.5: Molécula de ADN de doble cadena.....	21
Figura 2.6: Expresión genética.....	22
Figura 2.7: Cambios estructurales de una proteína.....	23
Figura 2.8: Enzima de restricción realizando un corte romo.....	24
Figura 2.9: Enzima de restricción realizando un corte cohesivo.....	24
Figura 3.1: Modelo básico de una enzima como un conmutador.....	25
Figura 3.2: Un modelo de compuertas lógicas basadas en ADN.....	26
Figura 3.3: Cómputo con ADN basado en formaciones de Harpin.....	28
Figura 3.4: AFND POT.....	29
Figura 3.5: Computadora de ADN de Braich.....	30
Figura 3.6: Grafo dirigido.....	31
Figura 3.7: Método de codificación de los caminos de Adleman.....	32
Figura 3.8: Esteganografía genómica.....	35

Figura 4.1: Antígeno particulado.....	37
Figura 4.2: Imágenes de microscopio electrónico del linfocito.....	38
Figura 4.3: Modelo de listón de una molecula de inmunoglobulina.....	38
Figura 4.4: Hematopoyesis.....	39
Figura 4.5: Inmunoglobulina como receptor y en forma secretada.....	40
Figura 4.6: Linfocito B activado y funciones efectoras de las inmunoglobulinas.....	40
Figura 4.7: Diversidad de los genes de los genes de las inmunoglobulinas.....	41
Figura 4.8: La recombinación de los segmentos V-(D)-J.....	42
Figura 4.9: Recombinación somática y expresion de los genes de las inmunoglobulinas ..	43
Figura 4.10: Región variable y constante de las inmunoglobulinas.....	44
Figura 4.11: Selcción positiva y negativa de los linfocitos inmaduros.....	45
Figura 4.12: Selección clonal.....	46
Figura 5.1: Algunos componentes del sistema inmunológico.....	51
Figura 5.2: Recombinación somática como procesamiento de cadenas.....	51
Figura 5.3: Funcionamiento central de los sistemas splicing.....	52
Figura 5.4: La operación de cómputo con ADN del sistema splicing.....	55
Figura 5.5: Funcionamiento general del sistema splicing.....	58
Figura 5.6: Splicing con cadenas circulares.....	59
Figura 5.7: Algoritmo de cómputo con ADN del sistema splicing.....	61
Figura 5.8: Representación del cómputo con ADN.....	62
ANEXO I.....	70
ANEXO II.....	71
ANEXO III.....	72
ANEXO IV.....	75
ANEXO V.....	76

# ABREVIATURAS

**ADN** - Ácido desoxirribonucleico.

**AFD** – Autómata Finito Determinístico

**AFND** – Autómata Finito No Determinístico.

**ARN** – Ácido ribonucleico.

**ARNm** – ARN mensajero.

**Ig** – Inmunoglobulina

**Lc B** – Linfocito B o Célula B

**MT** – Máquina de Turing

**MTN** – Máquina de Turing No determinista

**NP** – Polinomial no determinista

**P** – Polinomial

**pb** – Par base

# GLOSARIO

**Ácido desoxirribonucleico:** (ADN) es una cadena de nucleótidos. Cada nucleótido del ADN tiene como azúcar a la desoxirribosa, un grupo fosfato y una base nitrogenada que puede ser adenina (A), guanina (G), citosina (C) o timina (T). La función principal del ADN es servir como almacén de la información genética.

**Ácido ribonucleico:** (ARN) es una cadena de nucleótidos. Cada nucleótido del ARN tiene como azúcar a la ribosa, un grupo fosfato y una base nitrogenada que puede ser A, G, C o uracilo (U). La función principal del ARN es servir como intermediario de la información que lleva el ADN en forma de genes, generando proteínas.

**Bioingeniería:** aplicación de principios de ingeniería y de procedimientos de diseño para resolver problemas biológicos. Dentro de sus especialidades se incluyen la biomecánica, la ingeniería bioquímica y la bioelectricidad, etc.

**Biomoléculas:** son moléculas que están presentes en los seres vivos. Están constituidas principalmente por carbono, hidrógeno, oxígeno, nitrógeno y algunas pueden contener oligoelementos.

**Catálisis:** es el proceso a través del cual se incrementa la velocidad de una reacción química. El incremento de velocidad se da por la presencia de una sustancia que no forma parte del sistema en reacción. Ésta sustancia se llama catalizador (enzimas).

**Citoplasma:** Es un líquido comprendido entre la membrana plasmática y la envoltura nuclear.

**Cómputo:** obtención de una solución o resultado, a partir de ciertos datos o entradas utilizando para ello un algoritmo.

**Endonucleasas:** (enzimas de restricción) son enzimas que hidrolizan los enlaces fosfodiéster del material genético a partir de una secuencia que reconocen. Su función principal es cortar una cadena de ADN.

**Enzimas:** son proteínas o ARN que catalizan reacciones químicas en los seres vivos, llamadas también catalizadores orgánicos.

**Expresión génica:** es el proceso por medio del cual todos los organismos procariontes y eucariotes transforman la información codificada de los ácidos nucleicos, en las proteínas necesarias para su desarrollo y funcionamiento.

**Gen:** es una cadena de nucleótidos contenida en el genoma involucrado en producir proteínas; comprende regiones que proceden y siguen a la región codificadora (líder y cola) así como secuencias intermedias (intrones) entre segmentos codificadores.

**Genoma:** es todo el material genético contenido en el ADN de un organismo en particular.

**Genómica:** es la rama de la biología que se encarga del estudio de los genomas.

**Ingeniería genética:** puede definirse como un conjunto de técnicas, nacidas de la biología molecular, que permiten manipular el genoma de un ser vivo.

**Inmunidad humoral:** es el principal mecanismo de defensa contra los microorganismos extracelulares y sus toxinas, en el cual, los componentes del sistema inmune que atacan a los antígenos no son las células directamente sino los anticuerpos secretados por activación antigénica.

**Leucocitos:** (glóbulos blancos) son un conjunto heterogéneo de células sanguíneas las cuales participan en la respuesta inmunológica, como defensa del organismo contra sustancias extrañas o agentes infecciosos (antígenos). Se originan en la médula ósea y en el tejido linfático.

**Monómero:** es una molécula de pequeña masa molecular que unida a otros monómeros, por medio de enlaces químicos (generalmente covalentes) forman macromoléculas llamadas polímeros.

**Mutación:** describe cualquier alteración o cambio estructural en la secuencia del ADN.

**Nanotecnología:** es el desarrollo y la aplicación práctica de estructuras y sistemas en una escala nanométrica (entre 1 y 100 nanómetros).

**Nucleótido:** es un monómero formado por un azúcar, un grupo fosfato y una base nitrogenada.

**Oligonucleótido:** polímero corto de nucleótidos (menos de 50).

**Organelos:** son compartimientos localizados en el citoplasma y se encuentran rodeados por una membrana lipídica.

**Par de bases:** es un enlace de A con T o de C con G en una molécula de ADN de doble cadena.

**Plásmido:** pequeño fragmento circular de ADN situado en el citoplasma de algunas bacterias.

**Polímero:** compuesto químico que se forma por la unión de varios monómeros.

**Polipéptido:** larga cadena de aminoácidos unidos por enlaces peptídicos.

**Proteína:** Macromoléculas constituidas por aminoácidos unidos por enlaces peptídicos que intervienen en diversas funciones vitales, como el metabolismo, la contracción muscular o la respuesta inmunológica.

**Receptores:** son glicoproteínas presentes en la membrana plasmática, en la membrana de las organelas o en el citosol celular, a las que se unen específicamente otras sustancias químicas llamadas moléculas señalizadoras, como las hormonas y los neurotransmisores.

**Tejido:** es un conjunto de células de la misma naturaleza, diferenciadas de un modo determinado, ordenadas regularmente, con un comportamiento fisiológico común.

**Sustrato:** es una molécula sobre la que actúa una enzima.

**Big Bang:** Teoría del inicio del universo y su expansión.

**Abstracción:** Es ver un proceso biológico complejo de forma aislada bajo un sistema.

**Estandarización:** Llegar a una sola terminología en diferentes campos de investigación.

**Desacoplamiento:** Que cada especialista se aboque a su área, pero que pueda interactuar con otras.

# LISTA DE TABLAS

Tabla 1. Niveles estructurales de la vida.....	1
Tabla 2. Algunos valores de algunas funciones.....	15
Tabla 3. Tiempo consumido de hipotéticas soluciones a algunos problemas .....	16
Tabla 4. Algunos problemas clásicos resueltos en cómputo biomolecular. . . . .	29
Tabla 5. Características principales de la respuesta inmunológica específica.....	38
Tabla 6. Mecanismos que contribuyen a la generación de la diversidad.. . . . .	42
Tabla 7. Los cuatro principios básicos de la hipótesis de la selección clonal .....	46
Tabla 8. El tamaño de familias S (FL1, FL2) .....	57
Tabla 9. Tabla de verdad para la función F2.....	71

# INTRODUCCIÓN

## 1. INTRODUCCIÓN

Los organismos biológicos, llevan a cabo complejos procesos físicos, bioquímicos, etc., mismos que son regidos por la información genética, la cual es digital<sup>1</sup>. Por ejemplo, las reacciones bioquímicas y las principales operaciones realizadas en las células son reguladas por instrucciones almacenadas en su genoma, las cuales se encuentran codificadas en secuencias de ácidos nucleicos. Cuando las acciones realizadas por la maquinaria biomolecular ubicada dentro de las células que procesan ADN y ARN es comparada con la máquina de Turing, surgen notables semejanzas: los dos sistemas procesan información almacenada en una cadena de símbolos tomados de un alfabeto, y los dos operan por movimiento (paso por paso) a lo largo de esas cadenas, modificando o adicionando símbolos de acuerdo a un conjunto de reglas [1].

Los conceptos actuales de la ciencia de la computación están siendo usados para investigar la abstracción 'moléculas como máquinas de cómputo', en la que entidades de interacciones biomoleculares son descritas y modeladas como entidades de interacciones computacionales. Usando ésta abstracción se pueden generar nuevas posibilidades de entendimiento para los sistemas biomoleculares [2].

Los científicos a través de los años han estado dirigiendo sus estudios a los componentes de menor tamaño que conforman la vida, como son los órganos, tejidos, células y moléculas. El ADN, el ARN y las proteínas son las moléculas básicas de la vida, en forma de código genético (ADN) y otras que participan en los mecanismos para su expresión (proteínas) [Tabla 1]. Existen muchas razones, diversos intereses y trasfondos en esto, porque áreas como la física, la química, la ingeniería de materiales y las biociencias han convergido para trabajar y entender cosas en nanoescala [3].

**Tabla 1.** Niveles estructurales de la vida.

Nivel estructural de la vida	Ejemplo	Ciencia a cargo
Átomo	Carbono,	Física
Molécula	Proteínas, ARN, ADN	Biología Molecular*
Célula	Animal, vegetal	Citología**
Tejido	Tejido muscular, óseo	Histología
Órgano	Pulmón, ojo	Medicina
Sistema	Nervioso, límbico	Medicina
Organismo	Rana, hombre	Fisiología

Escala de medida en \*nanómetro, \*\*micras.

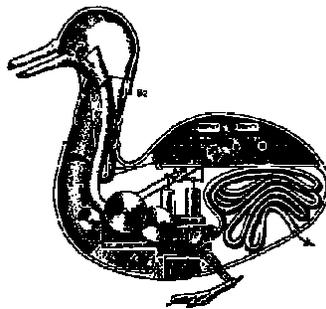
Uno de los sistemas que ha llamado la atención por su organización biológica altamente especializada es el sistema inmunológico. Este es capaz de procesar información con el fin de contrarrestar agentes extraños, de una forma masiva pero ordenada. Es por eso que en esta tesis se ha propuesto analizar, el sistema como un sistema de cómputo con ADN, lo cual nos permitirá determinar si dicho sistema se encuentra enlazado con las reglas de la ciencia de la computación.

<sup>1</sup>Digital es todo aquello que esta representado mediante elementos discretos; en el caso de la información genética esta se representa mediante 4 elementos.

## 2. ANTECEDENTES

La investigación que relaciona a la ciencia de la computación con la biología no es nueva. Por una parte se tiene el estudio de componentes computacionales que tienen los sistemas biológicos, en este segmento se tienen los precedentes siguientes:

En el siglo XVII René Descartes, asombrado con el aumento de autómatas mecánicos que se comenzaban a construir, consideraba a los animales como complejos autómatas, y que en la práctica sus músculos y huesos funcionaban como la mecánica de un artilugio [4], en ese mismo siglo Jacques de Vaucanson, relojero con amplios conocimientos de anatomía y mecánica, quería demostrar mediante autómatas mecánicos la realización de principios biológicos básicos, tales como la circulación, la digestión o la respiración [Figura 1].



**Figura 1:** Pato de Vaucanson. Autómata mecánico de cobre que representaba un pato que podía realizar las funciones básicas de alimentación y de interacción con el exterior.

En 1936, A. Turing, intentaba modelar la acción de la mente humana y presentó el concepto de "computabilidad". Según él, algo (fuera un número, un teorema, una acción o un comportamiento) era computable si existía una máquina capaz de computarlo [5]. En 1943, E. Schrödinger planteó que la diferencia entre un sistema biológico que se podía reducir hasta llegar a sus reacciones químicas elementales y un conjunto de sustancias químicas reaccionando en un vaso de precipitado, era la información contenida en el genoma del sistema biológico que dicta las reacciones químicas [6]. En 1970, el premio Nobel J. Monod identificó algunos procesos que se realizan en el ADN que podrían ser vistos como comportamientos acorde a los principios lógicos: "La lógica de los sistemas de regularización biológica no cumple con las leyes Hegelianas<sup>2</sup> sino, como el funcionamiento de las computadoras, con el álgebra proposicional de George Boole." [7]. En 1995, D. Bray identificó que muchas proteínas en las células presentan como una primera función la transferencia y procesamiento de información, más que la transformación química de intermediarios metabólicos o la construcción de estructuras celulares [8].

---

<sup>2</sup> Las leyes Hegelianas son pensamientos que se centran en su mayor parte en ideas de Europa (Eurocéntrica) y en la dialéctica.

Por otro lado se tienen investigaciones donde se realiza cómputo mediante el empleo de organismos biológicos uni y multicelulares [Figura 2] y también utilizando compuestos sintéticos biomoleculares como primers<sup>3</sup>, dentro de los cuales se cuenta con los antecedentes siguientes:

En 1959, R. Feynman propone la construcción de computadoras a nivel molecular y el uso de arreglos de átomos para guardar información parecido al ADN, pero su aplicación debió esperar el desarrollo de métodos y materiales necesarios para lograrlo [9]. En 1982, C. Bennet realizó similares observaciones y propuso una hipotética máquina de Turing molecular. Interesado en la física de consumo de energía, él especuló que las moléculas podían ser la base de una relación eficiencia/energía mayor en dispositivos computacionales. Propuso el concepto de Computadora Browniana, desarrollando esta idea al observar que una máquina de Turing browniana podía construirse a partir de una macromolécula como el ARN [10]. En 1994 L. Adleman fue el primero en demostrar la capacidad de las moléculas de ADN para ser un instrumento de cómputo, lo anterior fue gracias a que consiguió resolver una instancia concreta del problema del camino Hamiltoniano, comprobando cómo una búsqueda masiva paralela podía ser implementada usando operaciones estándar de cadenas de ADN, mediante el uso de técnicas de biología molecular [11,12], proporcionando un ejemplo de computación a nivel molecular y materializó, en cierto sentido, las ideas que Feynman postuló [9]. Por primera vez se consigue resolver un problema matemático mediante el uso de herramientas bioquímicas, cuando hasta entonces había sido el procedimiento inverso; es decir, el uso de herramientas matemáticas para resolver problemas bioquímicos. En 1995, inspirado en las ideas de Adleman, Lipton realizó una propuesta teórica para resolver instancias del problema de la satisfactibilidad de la lógica proposicional (SAT) [13]. En 1996, D. Beaver [14] demuestra que los primeros modelos de cómputo realizados mediante biomoléculas son completos, en el sentido de tener la misma potencia computacional que la máquina de Turing (modelo general de cómputo y por lo tanto tienen el máximo poder de cómputo).

En la Figura 2 se ilustra la Bacteria *Escherichia coli*, esta se encuentra frecuentemente en los intestinos de animales incluido el humano, su información genética (ADN) se ha secuenciado (se conoce su genoma) y por lo tanto es utilizada en diversos experimentos biomoleculares y también son empleadas para resolver problemas mediante cómputo (cómputo con ADN), con esto, se han resuelto instancias de los problemas de la mochila y el máximo conjunto independiente en un grafo [15,16].



**Figura 2:** Bacteria *Escherichia coli*.

<sup>3</sup> El primer es ADN producido artificialmente.

### 3. JUSTIFICACIÓN

En el artículo titulado “¿Puede un biólogo componer un radio?, o que aprendí mientras estudiaba apoptosis<sup>4</sup>”, Lazebnik describe las dificultades que tienen los científicos para el entendimiento de los sistemas biológicos y las diversas metodologías para su estudio [17], cuestionando el análisis de un problema en el campo biológico por medio de metodologías de un sistema que no le corresponde, por ejemplo, analizar un problema eléctrico con metodologías de un sistema bioquímico. Esta tesis muestra la importancia que tiene el tomar en cuenta un análisis del componente de cómputo que se encuentra en el sistema inmunológico con metodologías correctas de dicho cómputo.

El funcionamiento de los sistemas biológicos es de una complejidad extraordinaria, debido a que se encuentran involucrados diferentes factores: mecánicos, químicos, eléctricos, computacionales, etc. El presente trabajo permitirá a los investigadores del área biológica y médica dar a conocer la existencia de un componente de cómputo importante en el sistema inmunológico capaz de determinar las direcciones y limitaciones de una investigación, y por lo tanto, esta componente de cómputo debe ser de consideración al igual que lo son otros como los bioquímicos, mecánicos y moleculares en el campo biológico.

La computación ha estado estrechamente ligada a la biotecnología y biomedicina, pero siempre ha sido vista como una herramienta, un ejemplo de esto es la bioinformática. La biotecnología en los últimos años ha crecido exponencialmente, especialmente a través de la biología molecular. El desarrollo mundial de la biotecnología hace que ésta área de actividad esté recibiendo un gran apoyo académico, político y financiero [18]. Sin embargo existen desafíos que limitan en mucho la ingeniería biológica y uno de éstos es la incapacidad para evitar o manejar la abstracción, la estandarización y el desacoplamiento en el estudio y manejo de los sistemas biológicos. [19]. Con este trabajo se pretende ayudar a vencer los desafíos antes mencionados aplicados en el sistema inmunológico, relacionando a la computación como una ciencia inmersa en un sistema biológico (inmunológico) y por lo tanto reafirmando aún más su grado como ciencia.

---

<sup>4</sup> La apoptosis es la muerte programada de una célula.

#### **4. HIPÓTESIS**

Si hablamos de que los procesos de recombinación génica dados en los linfocitos son parecidos al procesamiento de información que se lleva a cabo en una computadora de ADN, entonces podríamos hablar de que el sistema inmunológico realiza operaciones de cómputo con ADN, debido a los procesamientos de cadenas de ADN que se efectúan en los linfocitos.

#### **5. OBJETIVO GENERAL**

Analizar el sistema inmunológico específico como un sistema de cómputo con ADN, lo cual indica que este sistema lleva a cabo procesos de cómputo, encontrándose sujeto a las reglas de la ciencia de la computación.

#### **6. OBJETIVOS PARTICULARES**

1. Comparar la actividad en la producción de inmunoglobulinas que presentan los linfocitos con las operaciones de cómputo con ADN.
2. Analizar la computabilidad que presenta la actividad de los linfocitos.
3. Analizar si la complejidad computacional explica la actividad de los linfocitos.

## 7. CONTENIDO GENERAL

El contenido de esta tesis se encuentra organizado de la manera siguiente:

- **CAPÍTULO 1.** “La Ciencia de la Computación”, aquí se da una breve introducción de esta ciencia para comprender los conceptos de computabilidad y complejidad computacional.
- **CAPÍTULO 2.** “Biología Molecular”, se darán los conocimientos básicos de la biología molecular, principalmente la que corresponden a la estructura y función del ADN, mostrando también algunas técnicas básicas en esta área para procesar las cadenas de ADN y la lectura de sus resultados, al final se da una perspectiva de los avances en esta materia a manera de entender donde se encuentra ubicada esta investigación.
- **CAPÍTULO 3.** “Cómputo Biomolecular”, se explica lo que es cómputo biomolecular con el fin de entender cuales son sus modelos, materiales y métodos para realizar las operaciones de cómputo, también se analizan ejemplos y perspectivas en esta materia.
- **CAPÍTULO 4.** “Inmunología molecular”, se da una introducción de los mecanismos del sistema inmunológico desde un punto de vista molecular. Como principal tema se trata la recombinación genética y maduración de las células que participan en el sistema inmunológico.
- **CAPÍTULO 5.** “Resultados y Conclusiones”.

Finalmente se da la perspectiva, las referencias y anexos.

---

---

# CAPÍTULO 1

## LA CIENCIA DE LA COMPUTACIÓN

La ciencia de la computación se encarga del estudio del procesamiento de la información y su interacción con el mundo. Además se ocupa de la teoría, análisis, diseño, eficiencia, implementación y aplicación de los dispositivos que procesan información, llamados máquinas de cómputo. El objetivo de estas máquinas es resolver problemas.

No todos los científicos de la computación están de acuerdo con que la ciencia de la computación sea llamada como tal [20]. Su opinión en esta cuestión parece depender de la tradición en la cual crecieron. La objeción a que la computación sea una ciencia, es porque estudia objetos hechos por el hombre (tecnologías), lo cual es una falsedad. La ciencia de la computación estudia los procesos de información que se producen de forma artificial o natural. Esta ciencia ayuda a otras áreas a estudiar sus procesos, por ejemplo, los físicos explican el comportamiento de partículas con procesos cuánticos de información y los biólogos explican el ADN como información biológica codificada [21].

La ciencia de la computación utiliza un autómata mecánico como modelo general de máquina de cómputo y su nombre es la Máquina de Turing (MT). Tres cosas pueden suceder al intentar resolver un problema con una MT:

1. Si se resuelve un problema utilizando una MT, entonces también existirá la máquina de cómputo que lo resolverá.
2. El problema no puede ser resuelto implementando una MT, y como consecuencia, ninguna máquina de cómputo lo resolverá.
3. Al iniciar una MT podría nunca parar, ocurriendo lo mismo en una máquina de cómputo.

Existen otros modelos de autómatas que son máquinas de cómputo pero estos tienen el mismo o menor poder de cómputo que la MT. Las cadenas de símbolos son utilizadas por la MT para su funcionamiento, por lo que se presentan a continuación las principales definiciones.

### 1.1 ALFABETOS, CADENAS Y LENGUAJES

Definición 1. Un alfabeto es un conjunto finito y no vacío de símbolos. Se usará la letra griega  $\Sigma$  para denotar alfabetos. Los miembros de un alfabeto son llamados símbolos del alfabeto. Ejemplos:

$$\Sigma_1 = \{0,1\}$$

$$\Sigma_2 = \{0,1,2,3,4,5,6,7,8,9,\div,\times,+,-,=\}$$

Definición 2. Una cadena es una secuencia finita de símbolos tomados de un alfabeto  $\Sigma$ , Si  $\Sigma_1 = \{0,1\}$ , entonces 01001 es una cadena sobre  $\Sigma_1$ . Ejemplos:

Alfabeto  $\Sigma_1 = \{0,1\}$   
 1,10,1111,1100,10001,111000

Alfabeto  $\Sigma_2 = \{0,1,2,3,4,5,6,7,8,9,+,-,=\}$   
 =1-, 4445,2+3=5

Definición 3. Una subcadena w de otra cadena c es  $c = xwz$ , siempre y cuando alguna de las subcadenas x ó z exista. Ejemplos:

10 es una subcadena de 10001  
 000 también lo es.

44 es una subcadena de 4445  
 45 también lo es.

Definición 4. Al conjunto infinito de todas las cadenas sobre el alfabeto  $\Sigma$  se le denota con  $\Sigma^*$  y se le nombra la estrella de Kleen. Ejemplos:

$$\Sigma_1^* = \{\Lambda, 0, 1, 00, 01, 10, 11, 000, 001, \dots\} \quad \Sigma_4^* = \{\Lambda, \#, \#\#, \#\#\#, \#\#\#\#, \#\#\#\#\#, \dots\}$$

Definición 5. Un lenguaje denotado por L, es un conjunto finito o infinito de cadenas formadas de símbolos del alfabeto  $\Sigma$ , por lo que un lenguaje es un subconjunto de  $\Sigma^*$ . Ejemplos:

Lenguaje sobre  $\Sigma_1$  de todas las cadenas de longitud 3  
 $L_1 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

Lenguaje sobre  $\Sigma_2$  compuesto de todas las cadenas formadas por  $\Sigma_2$   
 $L_2 = \Sigma_2^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$

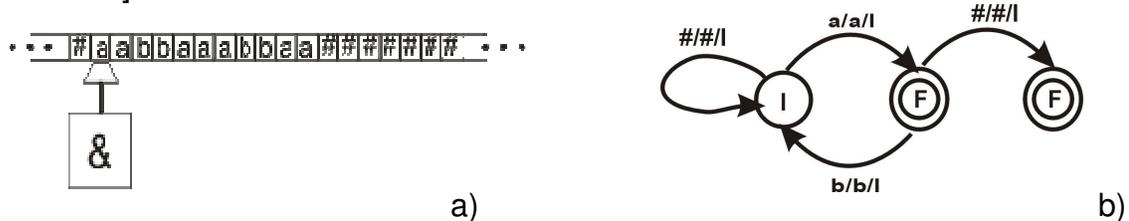
En las cadenas pueden realizarse operaciones como concatenación, potencia, inversa, etc., y como los lenguajes son conjuntos de cadenas, en estos se pueden llevar acabo también una serie de operaciones tales como: complemento, unión e intersección, entre otras.

## 1.2 LA MÁQUINA DE TURING (MT)

La MT es un autómeta que, a pesar de su simplicidad es capaz de hacer cualquier cómputo que pueda ser realizado y por lo tanto no existen modelos con mayor poder de cómputo. Para que una MT cumpla su cometido ésta debe ser diseñada para que decida si una cadena pertenece a un lenguaje.

Los componentes de la MT son los siguientes [Figura 1.1]:

- Una cinta infinita por los dos lados, dividida en espacios, en cada espacio puede alojarse un solo símbolo de un alfabeto.
- Un cabezal capaz de leer/escribir símbolos de un alfabeto en una cinta, y de desplazarse por la misma de la izquierda a la derecha ó viceversa.
- Un conjunto finito de estados internos y dentro de ese conjunto un estado de inicio y al menos un estado de aceptación.
- Una serie de reglas nombrada función de transición (&), que le dice al dispositivo, según lo que este leyendo en ese momento y su estado interno, qué tiene que hacer [ANEXO I].

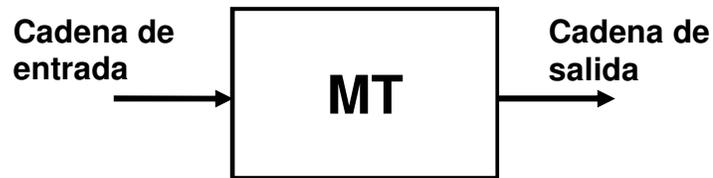


**Figura 1.1:** La Máquina de Turing. a) Representación de una Máquina de Turing, donde & representa a la función de transición. b) Representación de la función de transición con un diagrama, I es el estado de inicio y F los estados de aceptación. En este ejemplo la MT acepta cadenas del alfabeto {a,b} que terminen en el símbolo a.

Su funcionamiento está basado en un paso elemental, la transición, que se compone siempre de tres acciones:

- Transición de un estado (otro o el mismo estado).
- Escritura de un símbolo en la celda de la cinta que examina, reemplazando al que había antes.
- Desplazamiento del cabezal a una posición (izquierda o derecha).

El resultado final de la MT es reconocer si la cadena de la cinta pertenece o no a un lenguaje y para que esto suceda lo primero que debe ocurrir es que la MT se detenga y lo haga en un estado de aceptación (la cadena pertenece al lenguaje) o un estado de no aceptación (la cadena no pertenece al lenguaje), pero también la MT puede funcionar como un calculador de funciones, en este caso se configura la función de transición de manera que la cadena escrita en la cinta al final del proceso sea tomada en cuenta [Figura 1.2].



**Figura 1.2:** La MT como calculadora de funciones. En este caso se toma en cuenta el contenido de la cadena al detenerse la MT.

### 1.3 TEORÍA DE CÓMPUTO

El principal objetivo de la computación es resolver problemas, por lo tanto la Teoría de Cómputo puede caracterizarse como la búsqueda de respuestas para los siguientes problemas.

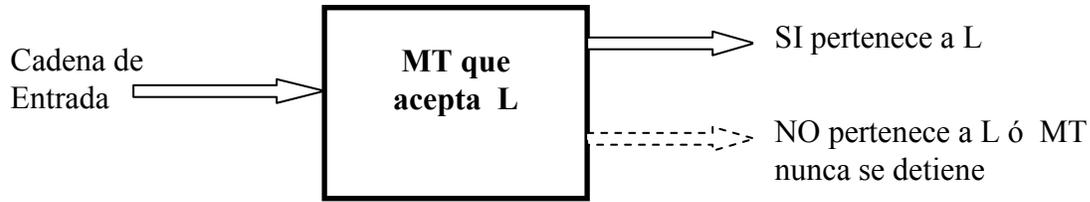
- Dado un problema, ¿Puede su solución ser encontrada utilizando cómputo? (Problema computable).
- Si un problema es computable, puede serlo sólo en una clase de máquina.
- Si un problema es computable, qué cantidad de recursos (tiempo y memoria) son requeridos.
- Existe dependencia entre recursos en un problema computable (más velocidad requiere más memoria).
- Podrían los recursos requeridos para resolver un problema ser tan grandes que hacen imposible resolverlo.
- Pueden ser las clases de problemas tratables e intratables definidos.

La Teoría de Cómputo se sostiene en dos áreas fundamentales: la teoría de la computabilidad y la Teoría de la complejidad computacional.

### 1.4 TEORÍA DE LA COMPUTABILIDAD

La computabilidad se encarga de constatar la existencia o ausencia de un algoritmo para resolver un problema determinado. El algoritmo es fundamental para la ciencia de la computación, si el problema cuenta con un algoritmo, su solución es encontrada utilizando una máquina de cómputo, sin importar los recursos que ella requiera. Así, cualquier limitación de las capacidades de los algoritmos constituye también una limitación de las capacidades de la máquina de cómputo [22]. Para la comprensión de lo que es formalmente un algoritmo se tienen las siguientes definiciones:

**Definición 6.** Se tiene un lenguaje  $L$  de un alfabeto  $\Sigma$ . Se dice que una MT con entrada de cadenas del alfabeto  $\Sigma$ , acepta al lenguaje  $L$  si al introducir cada una de las cadenas del Lenguaje  $L$  éstas llegan siempre a un estado de aceptación, pero si la cadena que se introduce a MT no pertenece a  $L$ , no se asegura que la MT se detenga para dar el resultado [Figura 1.3].

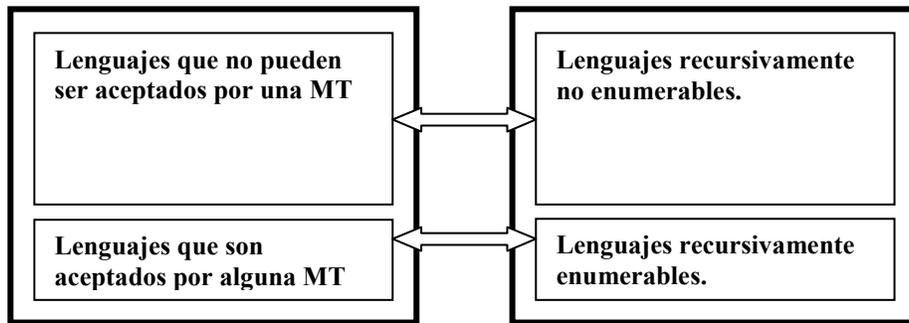


**Figura 1.3:** Una MT que acepta un lenguaje.

El conjunto de todos los lenguajes es mucho mayor que el de MT's y una de estas máquinas puede aceptar sólo un lenguaje, por lo que debe haber un lenguaje que ninguna máquina acepte, lo que significa que no existe la MT que pueda aceptarlo.

Definición 7. Se tiene un lenguaje  $L$  de un alfabeto  $\Sigma$ . Se dice que  $L$  es un lenguaje recursivamente enumerable si es aceptado por una MT y recursivamente no enumerable si no es aceptado por una MT [Figura 1.4].

Se sabe que el conjunto de lenguajes recursivamente enumerables es menor al conjunto de lenguajes recursivamente no enumerables.



**Figura 1.4:** Clasificación de los lenguajes por la existencia de una MT que los acepte.

Definición 8. Se dice que una MT decide respecto de un lenguaje  $L$ , si la MT llega siempre a un estado de aceptación o no aceptación [Figura 1.5].

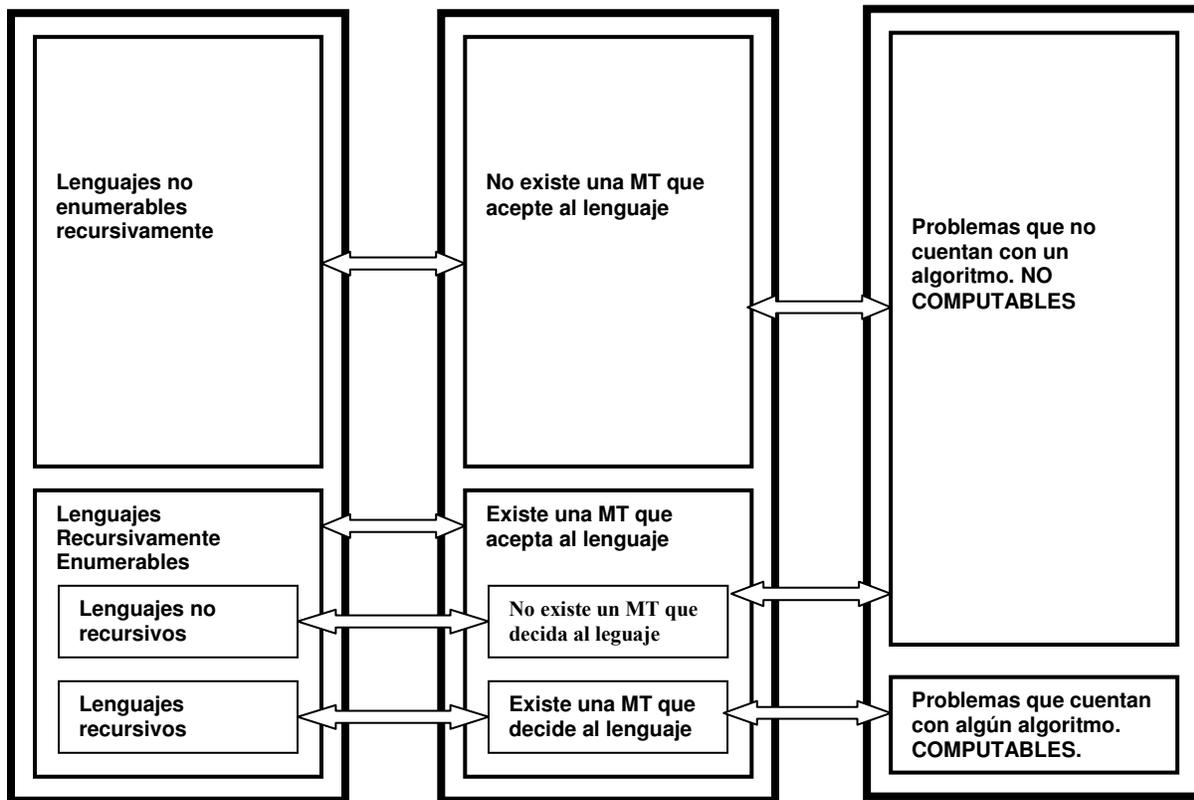


**Figura 1.5:** Una MT que decide respecto a un lenguaje.

Definición 9. Un lenguaje es recursivo si existe una MT que decida respecto al lenguaje [Figura 1.6].

Lo anterior indica que el conjunto de lenguajes recursivos es parte del conjunto de los lenguajes recursivamente enumerables. Ahora bien, se tiene que cualquier problema puede ser descrito como un problema de decisión, por ejemplo, la suma de dos números enteros, se construye una MT que calcula función, ya que la cadena de salida contendrá el resultado de la suma, si la cadena que entra a la MT es correcta, la MT parará en un estado de aceptación y entregará el resultado, si la cadena no es correcta, la MT también parará (el lenguaje recursivo asegurará esto), haciéndolo en un estado de no aceptación, no importando en este último caso la cadena de salida o se podría configurar la función de transición para que entregue una cadena que represente un error, por lo que se tiene lo siguiente:

Un problema de decisión se resuelve si existe un lenguaje recursivo para dicho problema. Si este lenguaje recursivo existe, entonces, existe la MT que lo pueda resolver (MT que decida) y por lo tanto existe el algoritmo [23]. Si existe el algoritmo para resolver un problema entonces el problema es computable [Figura 1.6].



**Figura 1.6:** La relación entre la MT, lenguajes y computabilidad.

---

---

Los problemas por lo tanto se pueden dividir en dos grupos:

- Problemas computables: aquellos que cuentan al menos con un algoritmo para ser resueltos.
- Problemas no computables: aquellos que no cuentan con un algoritmo para ser resueltos.

## 1.5 EL PROBLEMA DE LA PARADA

Existe una gran cantidad de problemas no computables, es decir, que no hay algún algoritmo que resuelva el problema [Figura 1.6]. A continuación se analizará un problema de la computación llamado el problema de la parada o problema de la detención, este es el ejemplo de problema de cómputo sin solución más conocido. Consiste en determinar si una MT se detendrá con cierta cadena de entrada, o bien quedará en un ciclo infinito. Este fue el primer problema donde se demostró formalmente que no tenía solución.

Una de las razones por la que es importante conocer que el problema de la parada no tiene solución, es porque nos permite decidir si otros problemas son computables o no. Turing demostró que no existe una MT que nos informe sobre si otra MT se detendrá o no durante su procesamiento, es decir, si nos dará un resultado final [23]. Esto se puede analizar de la siguiente forma, supongamos que una máquina de cómputo y un algoritmo resuelven un problema A. El algoritmo constantemente se optimiza, por lo que surge un problema B y se debe saber si las modificaciones del algoritmo del problema A crearon un algoritmo o no. Esto es, si se creó un nuevo algoritmo para el problema A entonces la máquina siempre se detiene y entrega el resultado, pero si no es así, entonces no se tendrá la certeza de que la máquina se detendrá (en un minuto, en un minuto antes del Big Crunch o nunca), como consecuencia se decide crear el algoritmo que resuelve el problema B, que nos asegure que la próxima vez que se modifique el algoritmo del problema A, nos informe si obtuvimos un algoritmo. Pero ocurre que el algoritmo del problema B no existe, porque forman parte del problema de la parada y por lo tanto para el problema B no se tendrá una solución.

Sin embargo, el que un problema sea computable no implica que se pueda encontrar su solución cuando esta sea necesaria, pues muchos problemas que disponen de algoritmos para su resolución son inabordables para un elemento de cómputo por el elevado número de operaciones que hay que realizar para resolverlos; el siguiente tema explicará lo anterior.

## 1.6 TEORÍA DE LA COMPLEJIDAD COMPUTACIONAL

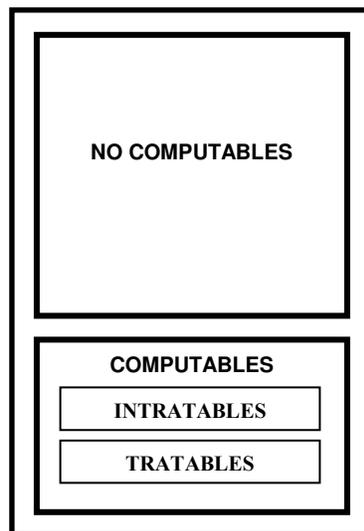
Una vez establecidos los resultados que muestran qué problemas se pueden resolver por medio de un algoritmo, el objetivo siguiente es establecer el precio a pagar, es decir, cuál será el costo asociado al algoritmo y si dicha solución algorítmica será factible, esto es, con un tiempo y espacio de ejecución “razonables”.

La complejidad computacional estudia los recursos (tiempo y espacio) requeridos en cómputo para solucionar un problema.

Tiempo.- Mediante una aproximación al número de pasos de ejecución de un algoritmo para resolver un problema teniendo en cuenta que cada paso se realiza en un tiempo determinado.

Espacio.- Mediante una aproximación a la cantidad de memoria utilizada para resolver un problema.

En la práctica los recursos de cómputo disponibles para la solución de problemas son limitados en tiempo y espacio. Lo anterior es importante en el sentido que un problema, aunque se derive de un buen algoritmo, puede ser inservible para algunas instancias ya que los recursos de memoria pueden sobrepasar la disponible y/o el tiempo requerido para obtener su solución podría ser muy grande. La Teoría de la Complejidad permite estimar la dificultad o, mejor dicho, la tratabilidad o intratabilidad de un problema y establecer si dispone o no de una solución algorítmica con recursos prácticos para su implementación [Figura 1.7].



**Figura 1.7:** La clasificación de los problemas de acuerdo a la computabilidad.

Un ejemplo de la complejidad computacional es cuando se analiza el problema de satisfacción booleana (SAT)[ANEXO II], existen varios algoritmos que resuelven el SAT, el problema de estos algoritmos es su ineficiencia práctica, por ejemplo, resolver el SAT con tablas de verdad consiste en generar todas las posibles valoraciones de las variables de la fórmula, verificando si para alguna de ellas la fórmula se hace cierta. Se tiene que para 2 variables se requiere verificar  $2^2$  valoraciones y para 20 variables,  $2^{20} = 1048576$  valoraciones en el peor de los casos, cada valoración ocupa un tiempo por lo que este algoritmo requerirá de  $2^N$  valoraciones, siendo N el número de variables; por lo tanto éste algoritmo es exponencial en el tiempo. En la práctica es inabordable para un número suficientemente grande de variables, como se puede ver en la Tabla 2.

Existen otros problemas que cuentan con algoritmos de tiempo exponencial como el SAT, los cuales en el mejor de los casos, tienen la función en tiempo  $N!$  o  $N^N$  siendo  $N$  el número de entradas o variables [Tabla 2], a estos problemas por su consumo excesivo de recursos (no razonables), se les denomina intratables. Los problemas que pueden ser resueltos en tiempo polinómico (Algoritmos de tiempo polinómico) se les denominan tratables.

La Figura 1.8, ilustra el crecimiento relativo de algunas de estas funciones y la Tabla 3 da un ejemplo de la cantidad de tiempo consumido en las diferentes soluciones de problemas según su costo computacional tomando un microsegundo por instrucción, en el cual se aprecia que para resolver un problema de SAT de 300 variables en el peor de los casos, requerirá un tiempo tal que la cantidad de siglos sea un número de 75 dígitos.

La teoría de la complejidad da una clasificación de los problemas atendiendo cual es la complejidad del mejor algoritmo conocido<sup>5</sup> para solucionar dicho problema [ANEXO IV].

**Tabla 2.** Algunos valores de algunas funciones [24].

Función \ N		N				
		10	50	100	300	1000
Polinomial	$5N$	50	250	500	1500	5000
	$N \times \log_2 N$	33	282	665	2469	9966
	$N^2$	100	2500	10 000	90 000	1 millón (7 dígitos)
	$N^3$	1000	125000	1 millón (7 dígitos)	27 millones (8 dígitos)	Mil millones (10 dígitos)
Exponencial	$2^N$	1024	Un número de 16 dígitos	Un número de 31 dígitos	Un número de 91 dígitos	Un número de 302 dígitos
	$N!$	3.6 millones (7 dígitos)	Un número de 65 dígitos	Un número de 161 dígitos	Un número de 623 dígitos	Un número de 2568 dígitos
	$N^N$	10 mil millones (11 dígitos)	Un número de 85 dígitos	Un número de 201 dígitos	Un número de 744 dígitos	Un número de 3001 dígitos

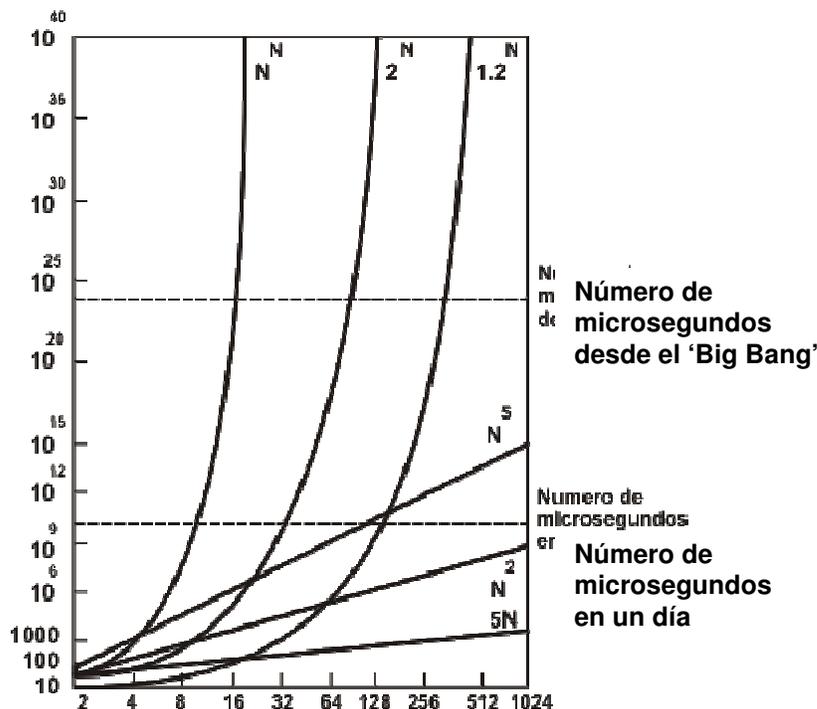
Para comparación: el número de protones en el universo conocido tiene 79 dígitos, el número de microsegundos desde el 'Big Bang' tiene 24 dígitos.

<sup>5</sup> Un algoritmo resuelve un problema, pero para un problema con solución algorítmica, existe más de un algoritmo que lo resuelve.

**Tabla 3.** Tiempo consumido en hipotéticas soluciones a algunos problemas (asumiendo 1 instrucción por microsegundo)[24].

Función \ N		10	20	50	100	300
Polinomial	$2^N$	1/10000 segundos	1/2500 segundos	1/400 segundos	1/100 segundos	9/100 segundos
	$3^N$	1/10 segundos	3.2 segundos	5.2 minutos	2.8 horas	28.1 días
Exponencial	$2^N$	1/1000 segundos	1 segundo	35.7 días	400 billones de siglos	Un número de 75 dígitos de siglos
	$N^N$	2.8 horas	3.3 billones de años	Un número 70 dígitos de siglos	Un número de 185 dígitos de siglos	Un número de 728 dígitos de siglos

Para comparación el 'Big Bang' fue aproximadamente hace 15 mil millones de años.



**Figura 1.8.** Crecimiento de algunas funciones [24].

### 1.7 VARIANTES DE LA MÁQUINA DE TURING

La versión de MT descrita anteriormente no es la única posible. Existen numerosas variantes, en las cuales se cumplen convenciones levemente distintas de configuración inicial, movimientos permisibles, protocolos seguidos para aceptar cadenas, etc. Además, el modelo básico de MT puede mejorarse de diversas maneras, por ejemplo una MT que cuente con cintas adicionales, puede hacer más sencillo describir la ejecución del algoritmo: el análisis puede resaltar los elementos de datos específicos que se almacenan en diversas cintas, sin abrumarse con las técnicas de registro que serían necesarias si todos los datos (cadena) se almacenan en una sola cinta y se recuperan de ella. Sin embargo con las conveniencias adicionales en la MT no se obtiene cambio alguno en el poder de cómputo de la MT original.

---

---

# CAPÍTULO 2

## BIOLOGÍA MOLECULAR

La biología molecular es la ciencia que se ocupa del estudio a nivel molecular de los componentes de los organismos biológicos, es decir, relaciona las estructuras de las biomoléculas con las funciones específicas que se desempeñan en la célula y en el organismo, por lo que la biología molecular es el estudio de la vida a un nivel molecular. Esta área se apoya de otros campos como Biología, Química, Genética y Bioquímica. La biología molecular concierne principalmente al entendimiento de las interacciones de los diferentes sistemas de la célula, entre ellas las del ADN con el ARN y la síntesis de las proteínas, las cuales juegan un papel importante en las vías metabólicas y de señalización, todas esas interacciones son reguladas para conseguir un eficiente funcionamiento de la célula.

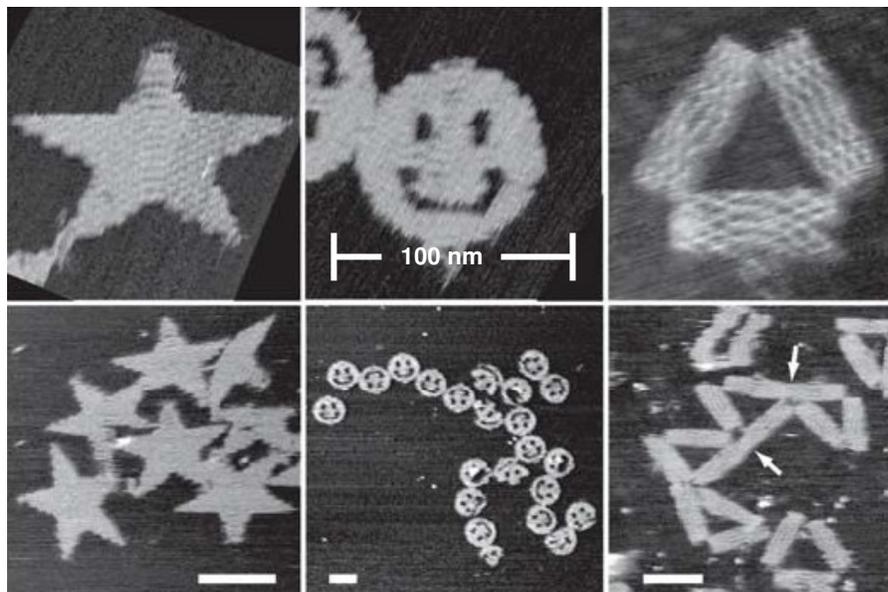
Las células contienen dos tipos de ácidos nucleicos: el ácido desoxirribonucleico (ADN) y el ácido ribonucleico (ARN). El ADN se encuentra en el interior del núcleo y contiene la información genética de la célula, es decir, contiene las instrucciones sobre cómo debe funcionar dicha célula. Tales instrucciones se ejecutan en el citoplasma celular y para ello, la información debe ser llevada desde el núcleo hasta el citoplasma; es aquí donde interviene el ARN, cuya función está relacionada con la expresión del material genético en proteínas. El conocimiento de la estructura y modo de acción de los ácidos nucleicos resulta fundamental para entender el control de la expresión genética y la división celular, así como el crecimiento y desarrollo de los organismos. Sin embargo, gracias a los adelantos tecnológicos el ADN como el ARN pueden ser sintetizados artificialmente en un laboratorio.

### 2.1 ESTRUCTURA DEL ADN (ÁCIDO DESOXIRRIBONUCLEICO)

El ADN es la molécula que juega el papel central en cómputo biomolecular [25], pero también el ADN ha probado ser un medio muy útil y excitante para otros usos en la nanotecnología, entre los que se encuentra la construcción de estructuras y dispositivos como la nanorobótica [26-28]. Ejemplo de esto son las técnicas de construcción de estructuras llamadas ADN Origami<sup>6</sup>, que ayudan en el diseño de nuevas computadoras biomoleculares y medicamentos [29][Figura 2.1].

---

<sup>6</sup> Origami es una palabra japonesa que significa arte de doblar el papel

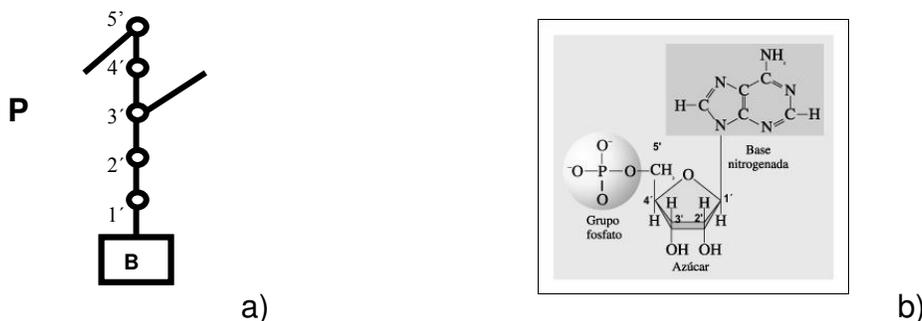


**Figura 2.1:** Origami ADN [29]. Existe un constante desarrollo para la manipulación de el ADN y ARN en diferentes campos como la mecánica, biología y medicina.

El ADN es una molécula crucial en la vida de la célula y tiene una estructura básica, la cual tiene tres de las funciones más importantes de la célula: codifica para la producción de proteínas; la replicación, que consiste en generar una copia exacta de una cadena de ADN la cual es transmitida (ó pasada) a la progenie y la recombinación genética que consiste en generar nuevas secuencias estructuradas de una cadena de ADN; este proceso es muy importante en el sistema inmunológico, ya que en él la célula modifica cadenas de ADN, para realizar esto, existe una maquinaria multienzimática (endonucleasas, exonucleasas, helicasas, polimerasas, recombinasas, etc.) que son las encargadas de poder manipular la información genética de acuerdo a las necesidades de la célula.

Las moléculas de ADN son polímeros construidos a partir de monómeros llamados nucleótidos [30,31]. Cada nucleótido está formado por una estructura que consiste de tres componentes: una molécula de azúcar desoxirribosa, un grupo fosfato y una base nitrogenada. Las bases nitrogenadas son cuatro: adenina, guanina, citosina y timina, abreviadas como A, G, C y T, respectivamente. Cada nucleótido difiere únicamente en término de su base por lo que, se utilizará el nombre abreviado de la base para identificarlos.

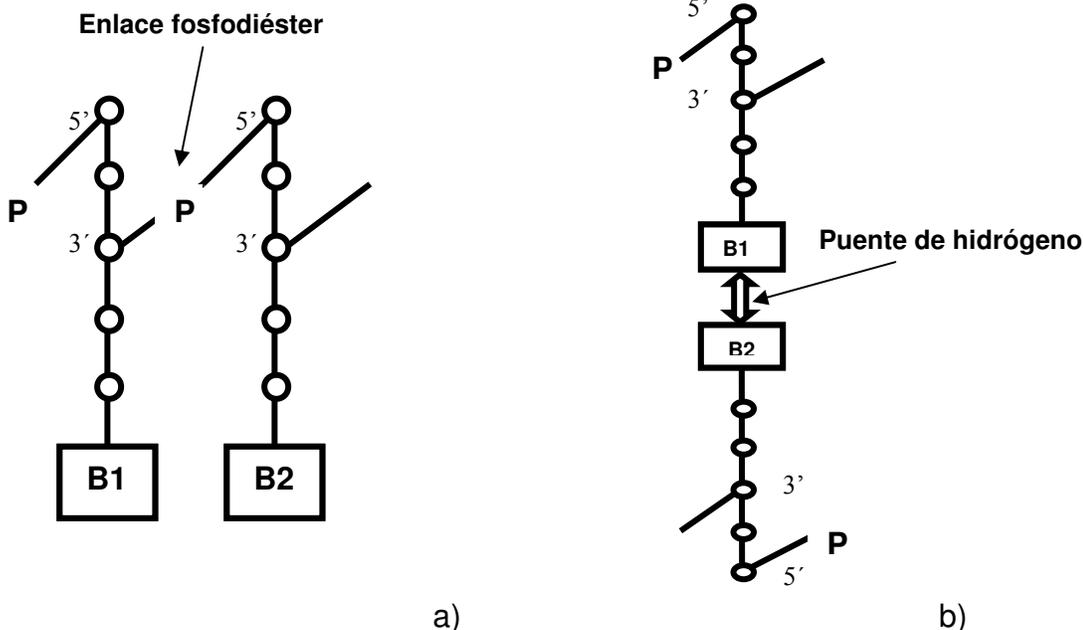
La estructura de un nucleótido es representada en una forma simplista en la Figura 2.2a, donde B es una de las cuatro bases posibles (A, T, C, G), F es un grupo fosfato y la línea vertical grande representa la molécula de azúcar (con sus carbonos enumerados del 1' al 5'). En la Figura 2.2b se muestra la composición química de un nucleótido en su forma desarrollada.



**Figura 2.2:** El nucleótido. a) Representación esquemática. b) La estructura química de un nucleótido con base nitrogenada de adenina [25].

Los nucleótidos pueden enlazarse de dos diferentes maneras:

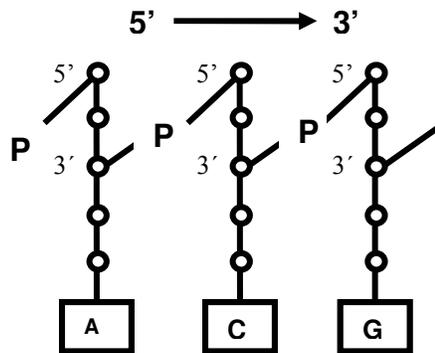
- El grupo fosfato que se encuentra en la posición 5' del azúcar de un nucleótido forma un enlace con el grupo hidroxilo en la posición 3' del azúcar del siguiente nucleótido. A este tipo de enlace se le denomina enlace covalente fosfodiéster. [Figura 2.3a]. Note que el enlace fosfodiéster une el carbono 5' del azúcar de un nucleótido con el carbono 3' del otro nucleótido, por lo que se habla de una dirección 5'-3'. Esta dirección es crucial para entender la funcionalidad y procesamiento de información del ADN.
- La base de un nucleótido interactúa con la base de otro nucleótido formando un enlace de hidrógeno (puente de hidrógeno), esto es llamado un par de bases. Este enlace está sujeto a las siguientes restricciones: A y T pueden formar pares de bases, y C y G pueden formar pares de bases, otros pares no son posibles [Figura 2.3b].



**Figura 2.3:** Enlaces entre nucleótidos. a) Enlace fosfodiéster. b) Enlace de hidrógeno. B1 y B2 representan una base respectivamente, las que se encuentran enlazadas respetando la complementariedad de Watson-Crick [25].

El principio de formación de pares de bases es llamado complementariedad de Watson-Crick [Figura 2.3b], donde la doble flecha entre las bases representa el enlace de hidrógeno. Este tipo de enlace es mas débil que el enlace fosfodiéster.

A partir de la unión de los nucleótidos por medio de los enlaces fosfodiéster se puede formar una cadena sencilla de ADN [Figura 2.4]. Note la dirección con la que se encuentran unidos los nucleótidos.



**Figura 2.4:** molécula de ADN de cadena sencilla [25].

Usando complementariedad de Watson-Crick, se pueden formar a partir de cadenas sencillas de ADN, moléculas de ADN de doble cadena [Figura 2.5], en este caso la cadena cuenta con tres pares de bases. La Figura 2.5 ilustra las reglas generales para unir dos moléculas de ADN de cadena sencilla por medio de un enlace de hidrógeno. En la molécula de doble cadena las dos cadenas sencillas tienen dirección contraria: el nucleótido final del lado 5' de una de las cadenas es enlazado al nucleótido final del lado 3' de la otra cadena. Existe una convención estándar que cuando la molécula de doble cadena es dibujada, la cadena que se encuentra arriba corre de izquierda a derecha en la dirección 5'-3' y la cadena de abajo corre de derecha a izquierda en la dirección 5'-3'. Por lo tanto la cadena de arriba de la Figura 2.5 es 5'-ACG y la cadena de abajo 3'-TGC.

El otro ácido nucléico que se encuentra en la célula, el ARN, al igual que el ADN está formado por cadenas de nucleótidos. Cada uno de los nucleótidos está formado por una molécula de azúcar ribosa, un grupo fosfato y una de cuatro posibles bases nitrogenadas: adenina, guanina, uracilo y citosina. Estos nucleótidos se unen igual que en el ADN. El ARN se diferencia químicamente del ADN por dos cosas: la molécula de azúcar del ARN contiene un átomo de oxígeno que falta en el ADN; y el ARN contiene la base uracilo (U) en lugar de la timina (T) del ADN.

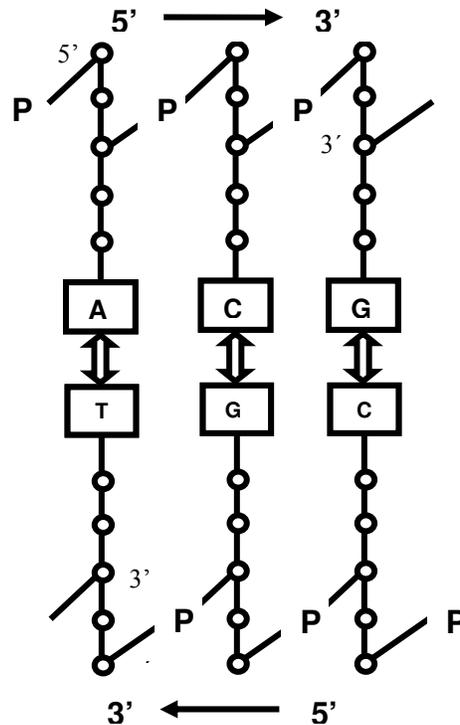


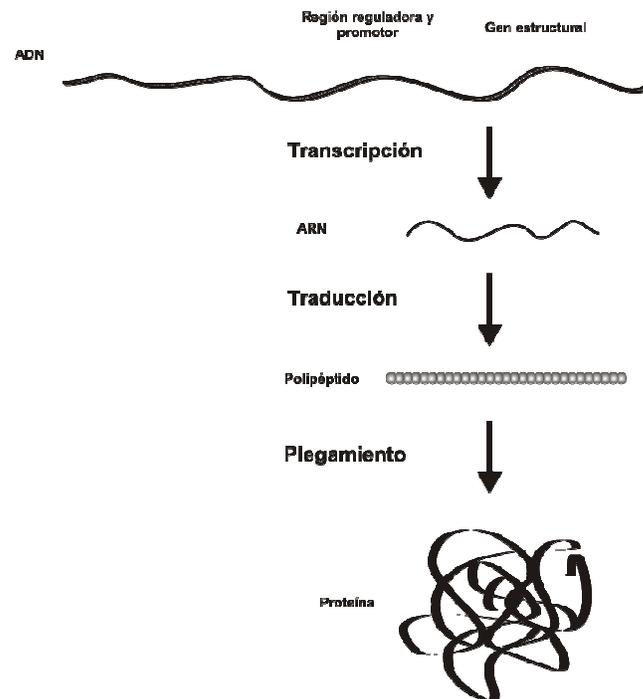
Figura 2.5: Molécula de ADN de doble cadena [25].

## 2.2 LA REGULACIÓN EN LA EXPRESIÓN GENÉTICA

La información genética contenida en las cadenas de ADN, está codificada en la secuencia de los pares de bases dentro de un segmento determinado. La célula humana tiene aproximadamente 3300 millones de estos pares de bases. La información debe expresarse (proteínas) antes de poder ser utilizado por la célula. Un gen es una secuencia lineal de pares de bases en la molécula de ADN, que contiene la información necesaria para la síntesis de una macromolécula con una función celular específica, por ejemplo: proteínas, ARN. El gen es considerado como la unidad de almacenamiento de información de las células biológicas. Los genes se disponen a lo largo de cada uno de los cromosomas. Cada gen ocupa en el cromosoma una posición determinada llamada locus. Una célula humana tiene 100 000 genes.

La estructura y función del gen las propuso Monod [32], quien a partir de sus observaciones permitió elaborar una hipótesis de la constitución y contenido del gen, postulando que está construido por diversas partes o secuencias, generalmente contiguas, dentro de una molécula de ADN: consta de una región reguladora donde se encuentra un gen activador, uno operador y uno represor, los cuales establecen cuando y en que cantidad se expresará el gen. Seguido de la región reguladora se encuentra el gen estructural, que está formado por la secuencia de nucleótidos que, al traducirse determina la secuencia de aminoácidos de la proteína [Figura 2.6]. Al principio y final del gen hay señales constituidas por secuencias de ADN, que determinan el inicio y la finalización del

proceso de transcripción. Todas esas señales son “leídas” y decodificadas por interacciones establecidas por medio de proteínas específicas.



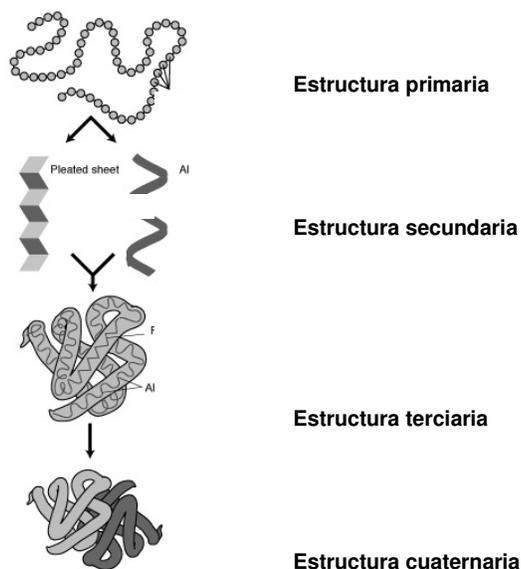
**Figura 2.6:** Expresión genética.

La mayoría de los genes codifican proteínas las cuales son responsables de la mayor parte de las funciones de un organismo. Para ello, la célula utiliza un proceso llamado transcripción, en el cual por medio de enzimas (polimerasas) y el ARN leen el gen estructural, generan una molécula de ARN que posteriormente será traducido por enzimas (ribosomas) a una secuencia de aminoácidos, generándose una proteína. Muchos genes se encuentran constituidos por regiones codificantes (exones) interrumpidas por regiones no codificantes (intrones) que son eliminadas en el procesamiento del ARN. La secuencia de bases presentes en el ARN determina la secuencia de aminoácidos de la proteína por medio del código genético. Un gen estructural es, simplemente, un gen que codifica un producto proteínico (o ARN). Un gen regulador es un gen estructural que codifica una proteína implicada en la regulación de la expresión de otros genes [Figura 2.6].

## 2.3 LAS PROTEÍNAS

Las proteínas están formadas por polímeros de aminoácidos (polipéptido) que ejecutan la mayor parte de las funciones vitales de las células. La función biológica de una proteína está determinada por la secuencia de aminoácidos que la componen y por la conformación espacial que esta presenta (estructura plegada en el espacio) [Figura 2.7]: el reconocimiento molecular, el transporte de moléculas, la formación de estructuras, la catálisis de las reacciones químicas, inclusive la regulación de la expresión de los genes

está determinada por proteínas que interactúan con el ADN. Las proteínas pueden estar formadas por uno o varios polipéptidos.



**Figura 2.7:** Cambios estructurales de una proteína. Cada proteína presenta una secuencia específica de aminoácidos, esta secuencia también llamada estructura primaria contiene la información necesaria para que estas biomoléculas se plieguen o adquieran una estructura tridimensional específica para desarrollar una función. [89]

## 2.4 ENZIMAS DE RESTRICCIÓN

Las enzimas de restricción también llamadas endonucleasas, pueden ser producidas por diferentes organismos tales como células y bacterias. Existen diferentes tipos de estas enzimas con nombres constituidos por letras y números romanos, ejemplo EcoRI, Hind III, Sau III y Hpa I, Pst I, Pvu II. El nombre es derivado del organismo de donde la enzima fue aislada. Las enzimas de restricción tienen la propiedad de cortar en secuencias específicas al ADN, lo que las hace ser indispensables en la biotecnología. Estas secuencias son llamadas secuencias de reconocimiento porque son identificadas por enzimas de restricción específicas. Las células del sistema inmunológico utilizan enzimas de restricción para procesar secuencias de ADN [33].

Las enzimas de restricción se agrupan en:

### Sistemas tipo I

Reconocen la secuencia específica de ADN, pero el corte no ocurre en el sitio de reconocimiento, sino que es al azar y en sitios distantes al de reconocimiento.

### Sistemas tipo II

El corte ocurre en el sitio de reconocimiento ó adyacente. Estas enzimas tipo II son utilizadas en genética y biología molecular puesto que permiten romper el ADN en sitios específicos, y así, recuperar secuencias conocidas. Las secuencias de

reconocimiento de las enzimas tipo II son de tipo palíndromo. Esta enzima puede realizar dos tipos de corte, romos [Figura 2.8] y cohesivos [Figura 2.9].

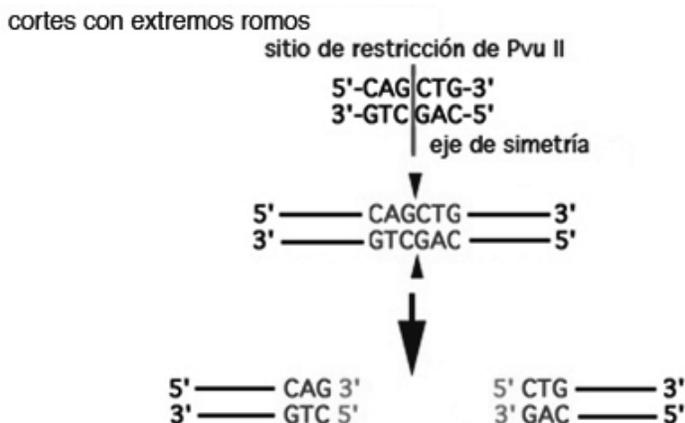


Figura 2.8: Enzima de restricción realizando cortes romos.

extremos pegajosos o cohesivos (sticky ends)

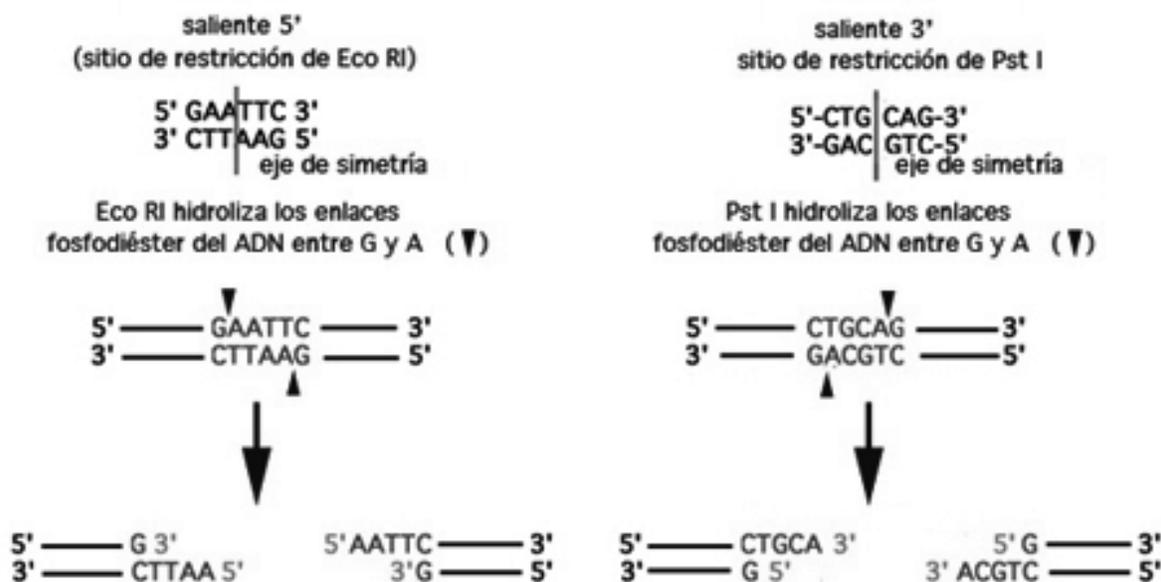


Figura 2.9: Enzima de restricción realizando cortes cohesivos [89].

### Sistemas tipo III

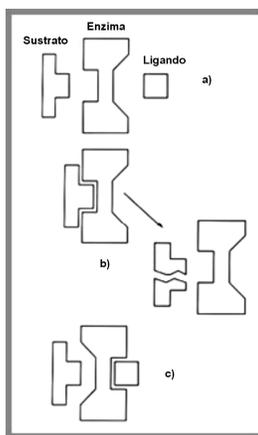
Es similar al sistema tipo I, utilizan una enzima oligomérica que realiza todas las actividades enzimáticas y corta el ADN entre 25-27 pares de bases, más allá del sitio de reconocimiento.

Muchas enzimas de restricción han sido purificadas de diferentes especies de bacterias, varias de las cuales reconocen diferentes secuencias de nucleótidos y están disponibles comercialmente hoy día.

# CAPÍTULO 3

## CÓMPUTO BIOMOLECULAR

El estudio de las moléculas de la vida (ADN, ARN y proteínas) ha transformado la práctica de la biología completamente, numerosas disciplinas están siendo influenciadas por la revolución biotecnológica [34]. En las operaciones de cómputo biomolecular se utiliza la naturaleza bioquímica de las moléculas para la solución de problemas, ya que estas pueden contener y procesar información en diferentes maneras; las operaciones de procesamiento de información son producidas por las interacciones entre estas moléculas [35]. Un ejemplo de ello son algunas proteínas que actúan en las células, las cuales parecen tener como función primaria la transferencia y procesamiento de información, más que ser utilizadas en la transformación metabólica, como intermediarias o constructores de estructuras celulares. Las proteínas que procesan información funcionan como un conmutador bioquímico, estando funcionalmente enlazadas a través de regulaciones alostéricas<sup>7</sup> u otros mecanismos dentro de circuitos bioquímicos que desempeñan una variedad de tareas computacionales incluyendo amplificación y almacenamiento de información [4, 36]. La Figura 3.1 ilustra como las enzimas tienen características computacionales similares a las compuertas lógicas, en el que el sustrato será catalizado por la enzima siempre y cuando el ligando adecuado contacte a la enzima para que realice un cambio en su conformación espacial de sus átomos (cambio alostérico), a) Los componentes del conmutador enzimático. b) La enzima reconoce al sustrato. c) El ligando actúa uniéndose a la enzima, cambiando su conformación. En ese momento, la enzima no reconoce más al sustrato [4].

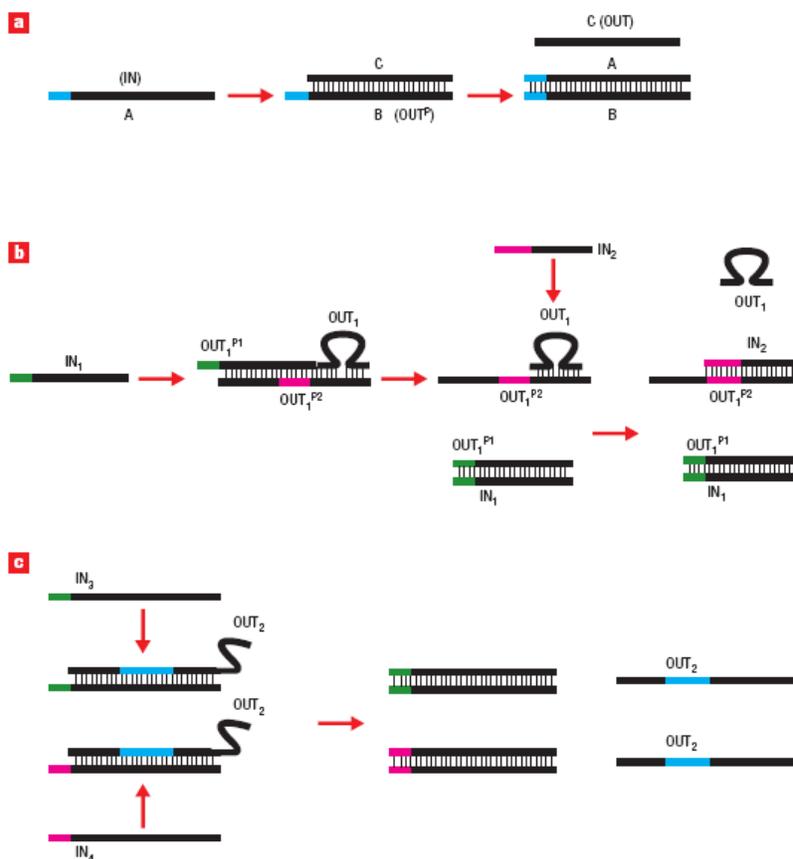


**Figura 3.1:** Modelo básico de una enzima como un conmutador.

<sup>7</sup> La regulación alostérica es un modo de control de las enzimas por el cual la fijación de una molécula en una ubicación en la enzima modifica las condiciones de fijación de otra molécula.

### 3.1 CÓMPUTO CON ADN

La computación con ADN es parte de la computación biomolecular, donde se considera también el uso del ADN y el ARN para propósitos de cómputo. La computación es ejecutada utilizando las técnicas y operaciones de la biología molecular tales como hibridación, desnaturalización, ligación, PCR, clonación molecular<sup>8</sup>, secuenciación, etc., Existe un gran estímulo entre la biología molecular y la ciencia de la computación para producir nuevas y mejores ideas en este campo de la computación [37]. La naturaleza interdisciplinaria tanto para su diseño y aplicación de cómputo con ADN hace interaccionar campos como la genética, bioquímica, farmacología, ciencia de la computación, biología, etc. La Figura 3.2 ilustra un proceso por el cual moléculas diseñadas de ADN conforman compuertas lógicas, Se muestran diversas cadenas de ADN interactuando entre si, a) A (señal de entrada) desplaza a C (señal de salida), ya que A y B forman un enlace más estable termodinámicamente que A y C, la cadena C podrá interactuar como señal de entrada de otra compuerta biomolecular. b) Compuerta AND, cadenas  $IN_1$  e  $IN_2$  son señales de entradas, si existen las dos da como salida la cadena  $OUT_1$ . c) Compuerta OR,  $IN_3$  e  $IN_4$  son entradas, si existe alguna de las dos da como salida alguna cadena de  $OUT_2$  [38].

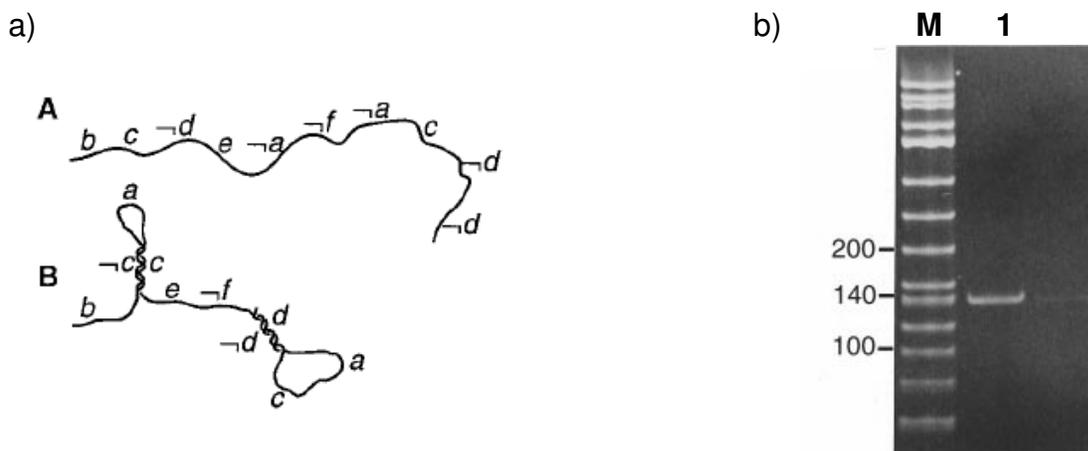


**Figura 3.2:** Un modelo de compuertas lógicas basadas en ADN.

<sup>8</sup> La clonación molecular es el proceso de aislar una secuencia de ADN de interés, insertarlo en un plásmido y obtener múltiples copias de ella en un organismo (generalmente procarionota).

Después de que Adleman L. resolvió una instancia del camino Hamiltoniano [11,12], se creó una gran expectativa ya que era la primera vez que se solucionaba un problema de este tipo empleando moléculas [11, 39]. Las investigaciones en este nuevo campo avanzaron creando diferentes modelos prácticos y teóricos. Pronto se pusieron de manifiesto las principales limitaciones de esta nueva forma de computación. A pesar del enorme paralelismo que se consigue al trabajar en algunos métodos sobre un tubo de ensayo con  $10^{20}$  cadenas de ADN, existen dificultades ya que este procesamiento paralelo computacional masivo se encuentra acotado [40]. Se han demostrado modelos de cómputo con ADN acordes con las teorías de la computación y con el mismo o menor poder que la MT, algunos de estos modelos verifican el poder de cómputo de algunas computadoras ADN experimentales, otros modelos están en espera del desarrollo tecnológico como es la producción de proteínas sintéticas, ADN modificado o nuevas técnicas en biología molecular [25, 41, 42]. La Figura 3.3 ilustra un ejemplo de cómputo con ADN con base en la formación de horquillas que forma la molécula de ADN, a) Se muestran dos cadenas de ADN (A y B) de las  $3^{10}$  configuraciones de cadenas posibles para la solución de un problema de FNC-SAT de 24 variables, 10 cláusulas disyuntivas de tres variables cada una, con este método su complejidad en tiempo es  $O(n)$ , se utilizaron  $10^4$  moléculas por cada configuración de cadena por lo que sería más  $30^{14}$  cadenas de ADN de 140 pares de bases cada una.

Para entender como funciona este procedimiento, se da un ejemplo sencillo con el siguiente problema FNC-SAT, la expresión lógica  $(a \vee b) \wedge (\neg a \vee \neg c)$  la cual puede ser pasada fácilmente a la siguiente forma  $(a \wedge \neg a) \vee (a \wedge \neg c) \vee (b \wedge \neg a) \vee (b \wedge \neg c)$  llamada FND-SAT (SAT de la forma normal disyuntiva), esta será codificada por un conjunto de cuatro cadenas de ADN,  $a \rightarrow a$ ,  $a \rightarrow c$ ,  $b \rightarrow a$  y  $b \rightarrow c$  las cuales pueden ser formadas por la misma técnica de Adleman, cada variable es representada por una secuencia de nucleótidos fija y la negación por complemento Watson-Crick de la variable, una fórmula se satisface si hay una cadena que no involucre ninguna variable junto con su negación, esto es, que no exista una cadena donde una subcadena que representa una variable tenga también su complemento, entonces si en la cláusula conjuntiva existe una variable y su negación, para cualquier valor en esa cláusula dará siempre FALSO y si existe alguna que no lo cumpla la función se satisface. En el caso de las cadenas de ADN formarán un harpin implicando que existe una variable con su negación, la solución puede ser extraída de la siguiente forma, se introducen endonucleasas que cortan las uniones de los harpin y si al final queda alguna cadena sin cortar (140 pares de bases y verificado por electroforesis) la fórmula se satisface. b) Se muestra un gel de agarosa en donde se corren las muestras, en el carril M se muestra un patrón de bandas que indica el número de pares de bases, en el carril 1 se observa una banda de 140 pares de bases, en este caso la expresión lógica se satisface, lo que indica que existen cadenas que no se cortaron [44]. Existen otros métodos para resolver instancias de FNC-SAT [13,43-46].



**Figura 3.3:** Cómputo con ADN basado en formaciones de Hairpin (horquilla).

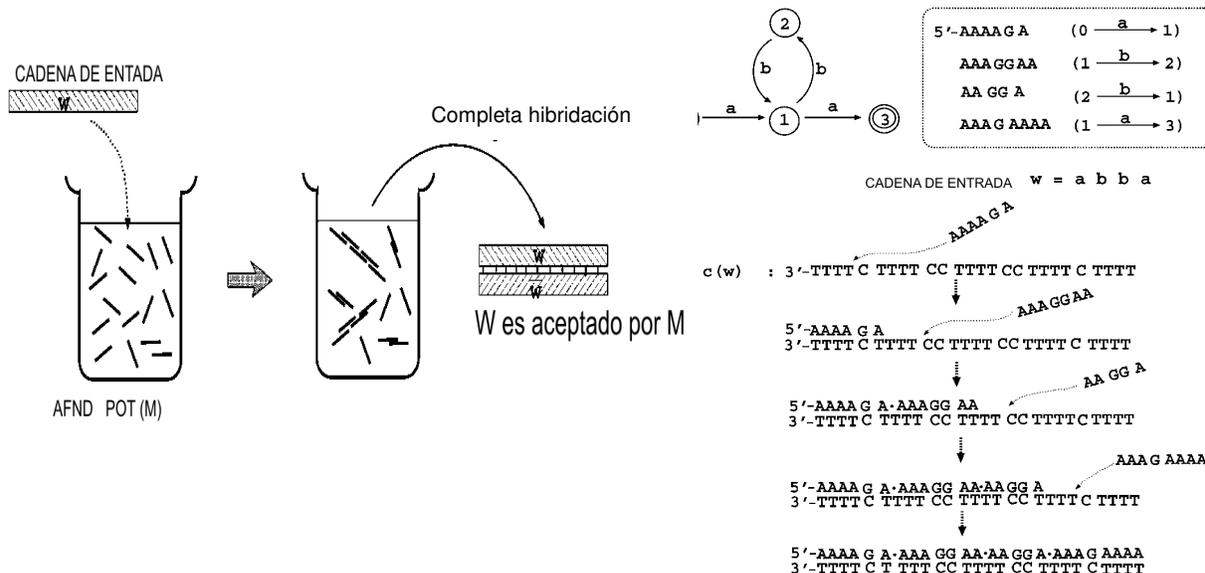
### 3.2 CARACTERÍSTICAS DE LAS OPERACIONES DE CÓMPUTO CON ADN

Los siguientes puntos son características principales de la computación con ADN.

- El procesamiento masivo paralelo de las cadenas de ADN: La mayoría de los problemas intratables pueden ser resueltos por una búsqueda a través de todas las posibles soluciones. Un ejemplo de ello es el criptoanálisis, todas las llaves posibles pueden ser probadas simultáneamente [47]. Otro ejemplo de este tipo de procesamiento masivo fue el que Adleman utilizó, aunque su método empleó la fuerza bruta, realiza unas  $1,2 \times 10^{18}$  operaciones por segundo ya que cada operación se hace en paralelo sobre todas las moléculas de ADN de un tubo de ensayo [48].
- Consumo de energía bajo: Algunos modelos de computadoras de ADN realizan unas  $2 \times 10^{19}$  operaciones por joule<sup>9</sup> (Ejemplo de una operación es una ligación), esto es cercano al límite teórico de  $3.4 \times 10^{20}$  operaciones por joule dictado por la segunda ley de la termodinámica. La información bioquímica para sus requerimientos es usualmente procesada por vía hidrólisis del ATP. Tales sistemas son notablemente eficientes con la energía [11, 48, 49].
- La densidad: Se puede llegar a representar 2 bits por cada base de una cadena de ADN. En un tubo de ensayo caben  $10^{20}$  cadenas de ADN. Suponiendo que las hebras tienen una longitud de 100 bases y que por cuestiones tecnológicas y de aplicación codificamos un bit mediante una secuencia de 10 pares de bases, se tendrá un total de  $10^{21}$  bits.

<sup>9</sup> Un joule equivale a utilizar un Watt en un segundo.

La Figura 3.4 ilustra un modelo sencillo de un AFND hecho con cadenas de ADN.



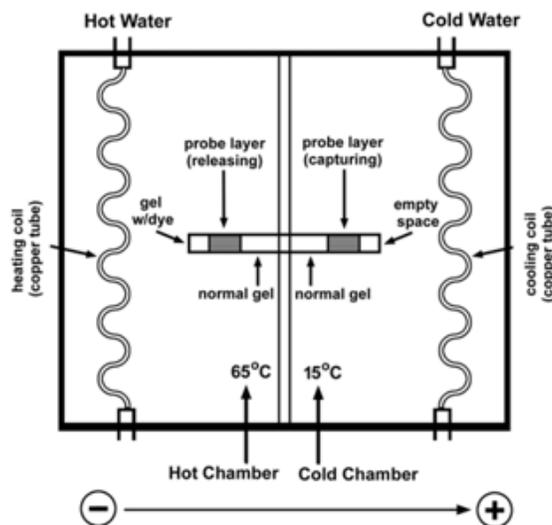
**Figura 3.4:** AFND POT. Cadenas sencillas de ADN son introducidas al AFND POT, el cual consiste de oligonucleótidos adecuados en un compuesto líquido que tendrá una hibridación con la cadena de entrada. La cadena introducida es aceptada si sale enlazada con su cadena complemento [50].

### 3.3 OPERACIONES Y PROBLEMAS LLEVADOS AL CÓMPUTO CON ADN

Se han realizado cálculos con ADN para realizar operaciones aritméticas [51] y lógicas [38, 52, 53], se han obtenido soluciones de instancias a problemas clásicos, la mayoría de estos problemas son de la clase NP, algunos son mostrados en la Tabla 4. Pero también se han resuelto problemas de ajedrez [54], creado juegos como el conocido gato y competido contra estas computadoras [55], Este tipo de cálculo se ha aplicado también para resolver problemas de procesamiento digital de señales [56, 57] y de lógica difusa [58], etc. Todas estas operaciones y soluciones se han realizado bajo diferentes modelos prácticos de computadoras de ADN, con diversas herramientas y operaciones de la biología molecular, pero también se han utilizado los paradigmas de computación actuales, como la computación evolutiva, que en el caso de las computadoras biomoleculares esta es llamada evolución in vitro [40]. Un dispositivo acoplado para realizar cálculo con ADN se muestra en la Figura 3.5.

**Tabla 4.** Algunos problemas clásicos resueltos en cómputo biomolecular. Algunos de estos problemas se resolvieron utilizando diferentes computadoras con ADN y operaciones en la biología molecular.

Problema	Año	Referencia
Camino Hamiltoniano	1994	[11]
FNC-SAT	1995	[13, 43-46]
3-colores	1995	[48]
Máximo Clique	1997	[59]
Conjunto máximo independiente	2000	[16]
correspondencia de Post acotado	2000	[60]
Producto de subconjuntos	2004	[61]
Mochila	2005	[15]

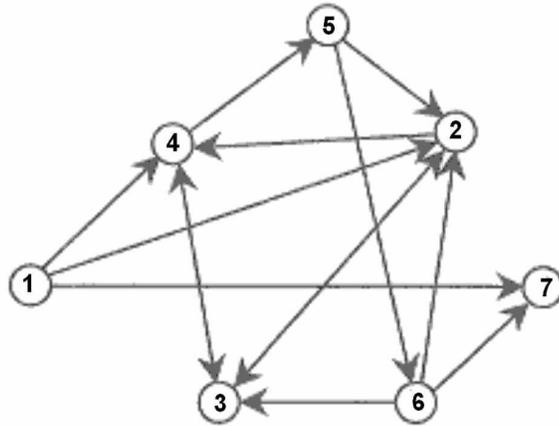


**Figura 3.5:** Computadora de ADN de Braich. Es una caja de electroforesis de 30 x 15 x 8 cm., dividida en dos partes, una caliente y otra fría. Utilizada para resolver instancias del problema FNC-SAT de 20 variables y 24 cláusulas, con complejidad computacional  $O(s)$  siendo  $s$  igual al número de cláusulas [45].

### 3.4 LA COMPLEJIDAD COMPUTACIONAL DE LAS COMPUTADORAS BIOMOLECULARES

La frontera entre los problemas ‘tratables’, ‘intratables’ o sin solución ha permanecido prácticamente inalterada durante los últimos 45 años, estas limitaciones se encuentran fundamentadas por las teorías de la computación, en el caso de las computadoras ADN el factor limitante en los recursos no es el tiempo sino el espacio, ya que el número de moléculas de ADN necesarias para codificar las posibles soluciones crece exponencialmente con el tamaño del problema [40]. A continuación se muestra brevemente el procedimiento que realizó Adleman en 1994 para resolver una instancia del camino Hamiltoniano, aunque su método utiliza la fuerza bruta y se han construido métodos prácticos y teóricos más eficientes para la solución de este problema con computadoras biomoleculares, se toma como ejemplo para ilustrar cómo la complejidad espacial aumenta exponencialmente.

En el campo matemático de la teoría de grafos, un camino Hamiltoniano en un grafo es una sucesión de aristas adyacentes, que visita todos los vértices del grafo una sola vez. El problema consiste en saber si existe el camino Hamiltoniano para un grafo y cual es su ruta, en el caso del grafo que resolvió Adleman [11,12] [Figura 3.6] el camino es 1->2, 2->3, 3->4, 4->5, 5->6.



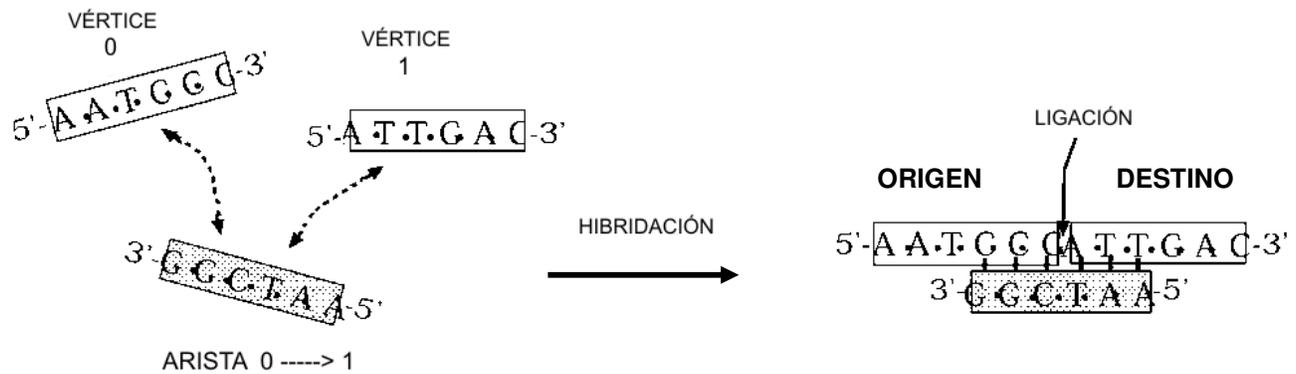
**Figura 3.6:** Grafo dirigido.

El siguiente algoritmo (no determinístico) resuelve el problema.

1. Generar caminos aleatorios en grandes cantidades a través del grafo dirigido.
2. Mantener solamente los caminos que comiencen con el vértice de INICIO (1) y terminen en el vértice de FIN (7).
3. Si el grafo tiene  $n$  vértices, en este caso 7, mantenga sólo los caminos que pasan por exactamente  $n$  vértices.
4. Mantenga los caminos que pasan por todos los vértices del grafo y que pasan por cada uno de ellos solo una vez.
5. Si al final de la búsqueda queda algún camino, responda 'SI' (con la posibilidad de secuenciar la cadena para determinar el camino) de otra manera responda 'NO'.

Operaciones realizadas en biología molecular:

- Para implementar el paso 1, cada vértice del grafo se codificó como una cadena de ADN aleatoria. Entonces por cada unión entre vértice, origen y destino que forman una arista en el grafo, se crea una secuencia de ADN (dirección 5  $\rightarrow$  3) cuya segunda mitad está compuesta de la secuencia complementaria que codifica la segunda mitad del vértice de origen y la primera mitad de la secuencia complementaria que codifica para el vértice de destino. Usando los complementos de los vértices como molde, las secuencias que corresponden a los vértices compatibles se 'ligan' y de esta forma se construyen moléculas de ADN que codifican caminos aleatorios a través del grafo, teniendo en cuenta que las cadenas que codifican los vértices y las cadenas que codifican los moldes se introducen masivamente en un tubo de ensayo [Figura 3.7].



**Figura 3.7:** Método de codificación de los caminos de Adleman. Cada vértice fue codificado por una cadena de 20 nucleótidos [50].

- Para implementar el paso 2, las cadenas de ADN producidas en el paso 1 se amplifican mediante la reacción en cadena de la polimerasa (PCR) [ANEXO V] solo las cadenas que inicien con la secuencia que codifica el vértice INICIO y terminen con la secuencia que codifica el vértice FIN. A partir de este paso solo quedarán los vértices que comienzan con INICIO y terminan en FIN.
- Para implementar el paso 3, se usa la técnica de electroforesis [ANEXO V], que hace posible la separación de las cadenas de ADN de acuerdo a su tamaño; las moléculas de mayor tamaño se desplazan de manera más lenta en el gel y después de determinado tiempo las moléculas quedan separadas en bandas de acuerdo a su tamaño, por lo que se escogen solo las cadenas que tengan el tamaño que codifica 7 vértices.
- El paso 4 se realiza aplicando iterativamente un proceso llamado purificación por afinidad. Este proceso permite que solamente las cadenas que contengan una determinada secuencia que corresponde al vértice  $v$  sean filtradas y separadas del resto de las moléculas. Previamente es necesario sintetizar cadenas complementarias a  $v$  y enlazarlas a la matriz de filtrado del proceso de purificación; solo aquellas cadenas que contienen la secuencia  $v$  quedaran adheridas a las cadenas complementarias sintetizadas, el resto de las cadenas no será retenida en la matriz una vez que el flujo de cadenas heterogéneas pasa a través de la matriz de purificación. Este proceso se repite con cada resultado de la purificación para cada uno de los  $n$  vértices de nuestro problema inicial.
- Para implementar el paso 5 es necesario averiguar la existencia de una molécula que represente el camino Hamiltoniano. Esto se hace amplificando los resultados del paso 4 mediante el PCR y determinando las secuencias de ADN de las secuencias amplificadas para ver si alguna representa la solución buscada.

---

---

Al analizar la complejidad computacional del método de Adleman se tiene que la complejidad en tiempo con  $n$  vértices es del orden de  $O(n^2)$  pasos de laboratorio. La complejidad en espacio se deduce de la siguiente aproximación: el número de todos los caminos posibles del grafo anterior con  $n$  vértices es aproximado a  $(\log n)^n$ , cada vértice fue codificado por una cadena de ADN de 20 nucleótidos, por lo que cada camino tendrá un promedio  $20n$  nucleótidos, por lo tanto tal experimento involucra  $20n(\log n)^n$  nucleótidos, esto sin tomar en cuenta las cadenas que son utilizadas como molde, si expandimos el grafo a 70 vértices esto nos dará aproximadamente  $10^{25}$  kilogramos (tomando en cuenta el promedio de la masa de un nucleótido) [62, 63, 64]. La masa del sol es aproximadamente de  $1.9 \times 10^{30}$  kg.

### 3.5 LIMITACIONES DE LA COMPUTACIÓN CON ADN

En cómputo biomolecular existen limitaciones que dificultan la puesta en práctica de algunos cómputos [65]

- Número total de cadenas de ADN en un tubo de ensayo: Esta podría ser la limitación más importante, ya que el tubo de ensayo que es generalmente utilizado puede contener una cantidad de mililitros, a menos que las operaciones sean realizadas en forma industrial. Por lo tanto el número de cadenas contenidas en un tubo de ensayo se encuentra limitado a no más de  $10^{24}$ . Esto reduce los límites del tamaño del problema NP que se puede solucionar y más si el método utilizado se basa en su mayoría en fuerza bruta.
- Tiempo de duración de cada paso de laboratorio. Mientras que algunas técnicas que se realizan en biología molecular pueden ser realizadas en pocos segundos, otras están relacionadas con factores como una operación bioquímica lenta.
- Número de pasos de laboratorio requeridos por un algoritmo molecular deberá ser extremadamente corto.
- Errores inducidos por las operaciones. Dependiendo de factores como la técnica en biología molecular, herramientas empleadas, tecnología y factores en los materiales, los cómputos pueden contener un porcentaje de error.

Para una percepción práctica de las limitaciones en cómputo con ADN, se ilustran los requerimientos, que con la tecnología de 1996 en el área de biología molecular, se plantearon métodos con base en modelos formales en cómputo con ADN para atacar el sistema de cifrado DES (Data Encryption Standard). Uno de los métodos de ataque pretende encontrar la clave empleada en el cifrado de un texto. Una de las propuestas plantea el denominado texto-plano/texto-cifrado. Se supone que el criptoanalista<sup>10</sup> dispone de (ha interceptado) un texto sin cifrar y su versión codificada y pretende determinar cuál

---

<sup>10</sup> El criptoanálisis es el estudio de los métodos para obtener el sentido de una información cifrada

---

---

es la clave empleada en ese cifrado. La propuesta que utiliza cómputo con ADN examina las  $2^{56}$  claves posibles, por lo que sigue un esquema de fuerza bruta aunque con moléculas de ADN [66, 67].

Los requerimientos totales son:

- En total se necesitan 1271 tubos de ensayo.
- el número de pasos de laboratorio que es necesario para realizar un ataque contra el algoritmo DES es de 6719.
- Durante el procesamiento, se calcula que nunca van a tener más de 96 tubos de ensayo activos al mismo tiempo.
- Un humano, que realice una operación por día, en total, tardará 18 años.
- Si las operaciones tardan una hora cada una, ocupando las 24 h en realizarse, el ataque durará 9 meses.
- Si usamos una máquina dedicada a las operaciones en biología molecular, suponiendo un minuto por operación, se tiene la clave en 5 h, (estas máquinas actualmente son desarrolladas para el procesamiento de cadenas de ADN en cuestiones de biotecnología, son computadoras híbridas las cuales emplean tecnologías de cómputo de ADN, silicio, así como robótica, óptica, etc.).
- Cada uno de esos pasos, es aplicado a todas las combinaciones posibles que puedan existir, por lo tanto, según la cantidad de la llave con la que se encripte, varía la cantidad de líquido necesario.
- Para romper el algoritmo DES, se usarán 140 mL de líquido por cada tubo de ensayo empleado.
- Si se quiere romper algoritmos que usen claves de mayor tamaño, es lógico pensar que necesitarán más ADN para codificar las soluciones.

Dado que el ADN es un dispositivo primordial de codificación, su potencial en criptografía (codificando y decodificando información) y esteganografía (encubrimiento de la información) [68] ha sido examinado desde muchas perspectivas [69]. En la Figura 3.8 muestra una cadena de ADN conteniendo un mensaje secreto fue camuflageado con el genoma humano. a) La estructura de un mensaje secreto en una cadena de ADN. b) Llave usada para codificar un mensaje. c) Análisis de un gel, en la línea 3-7 se muestra una relación del peso de (ADN mensaje / ADN genoma humano), se aprecia que no se distingue el patrón de bandas del ADN mensaje el cual cuenta con 100 pares de bases. c) Secuenciación del ADN mensaje, una vez obtenida la amplificación por PCR con las llaves que son los primers utilizados para su amplificación [68].

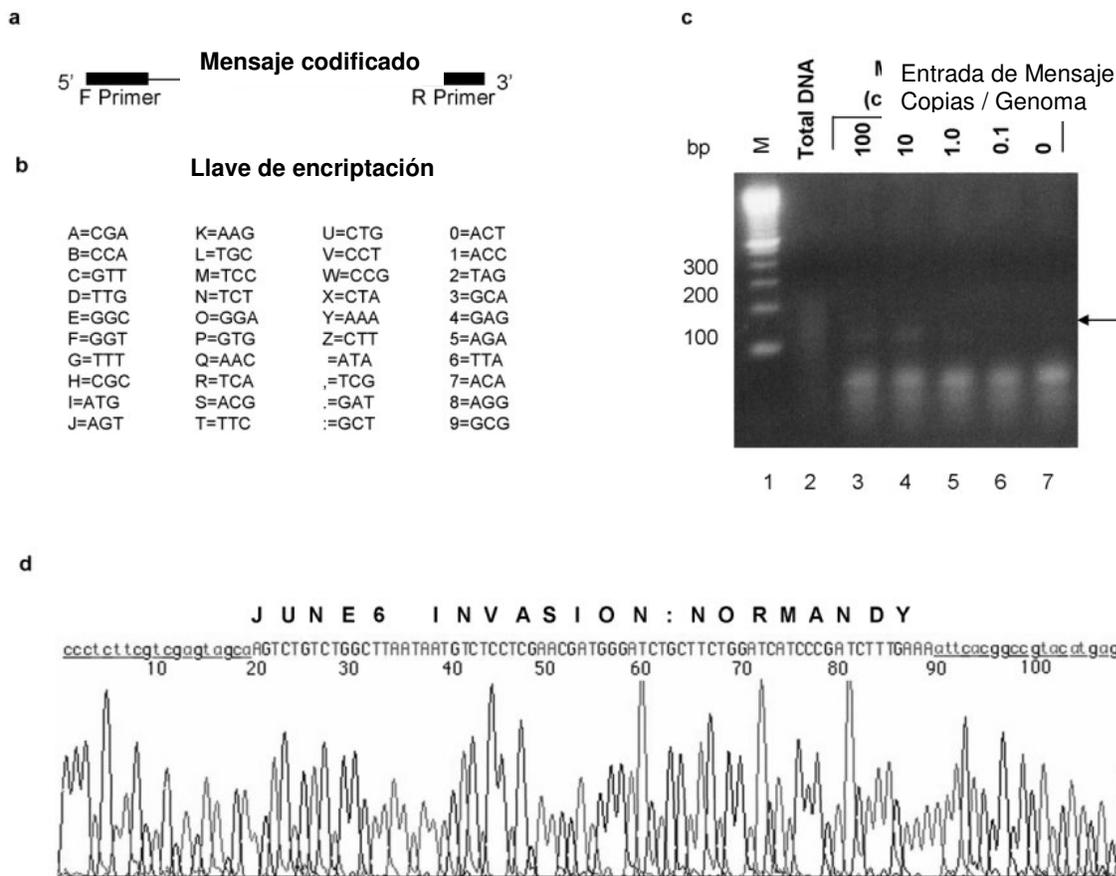


Figura 3.8: Esteganografía genómica.

### 3.6 MODELOS FORMALES DE CÓMPUTO CON ADN

Con los primeros resultados prácticos y teóricos de cómputo con ADN se han creado áreas nuevas a las que tradicionalmente existían, algunas de estas áreas son la biología matemática y la computación biológica (en ésta se encuentran sumergidas las operaciones de cómputo con ADN). Las áreas existentes anteriormente son la matemática biológica y la biología computacional (apoyada por la bioinformática).

La idea central en la que se apoya el nuevo campo de la computación biológica es la analogía entre los dos procesos siguientes:

- La compleja estructura de un ser vivo no es más que el resultado de aplicar operaciones simples (copiado, borrado, etc) a una información inicial codificada en secuencias de ADN.
- El resultado  $f(w)$  de aplicar una función computable a un argumento  $w$  se puede obtener aplicando una serie de funciones básicas y simples sobre  $w$ .

Ante cualquier modelo de cómputo existen dos preguntas básicas formuladas también en la computación con ADN, y son las siguientes:

- ¿Que tipo de problemas se pueden resolver con computación con ADN? O mejor dicho, ¿es el modelo de computación con ADN completo en el sentido de que la acción de cualquier MT puede ser llevada a cabo con manipulaciones de ADN?
- ¿Es posible diseñar computadoras de ADN programables?

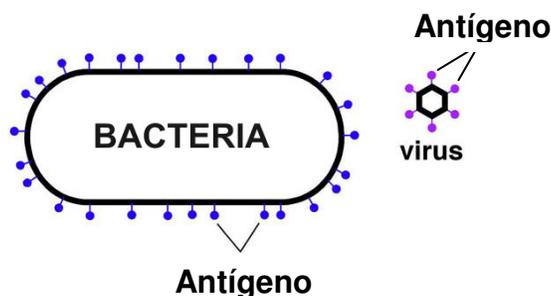
Existen varias direcciones de investigación que emplean las herramientas de la teoría de la computación para modelar las operaciones específicas de una serie de operaciones con ADN. En varios casos se obtienen modelos universales y completos. Los modelos principales son los sistemas sticker, los sistemas Inserción-delección y los sistemas de splicing.

El modelo de cómputo con ADN que se utiliza en esta investigación es el sistema splicing, también llamado de cortar y pegar porque las operaciones que se realizan a nivel molecular son por medio de endonucleasas que cortan en lugares específicos dentro de una cadena de ADN y ligasas que unen cadenas de secuencias de ADN. El sistema splicing tiene varios modelos de computadoras de ADN, algunos de estos tienen la misma potencia que la MT.

# CAPÍTULO 4

## LA INMUNOLOGÍA MOLECULAR

El sistema inmunológico está formado por un conjunto de mecanismos que se encuentran en un organismo biológico para contrarrestar cualquier material extraño y la aparición de células tumorales. El material extraño o parte de éste es reconocido como antígeno; existen diversos tipos de moléculas que pueden actuar como antígenos, tales como las proteínas, los polisacáridos<sup>11</sup> y más raramente otras moléculas como los ácidos nucleicos. El antígeno se puede encontrar soluble (moléculas dispersas como una toxina) o particulado (forma parte de una partícula mayor, como células, bacterias, virus, etc.) [Figura 4.1].



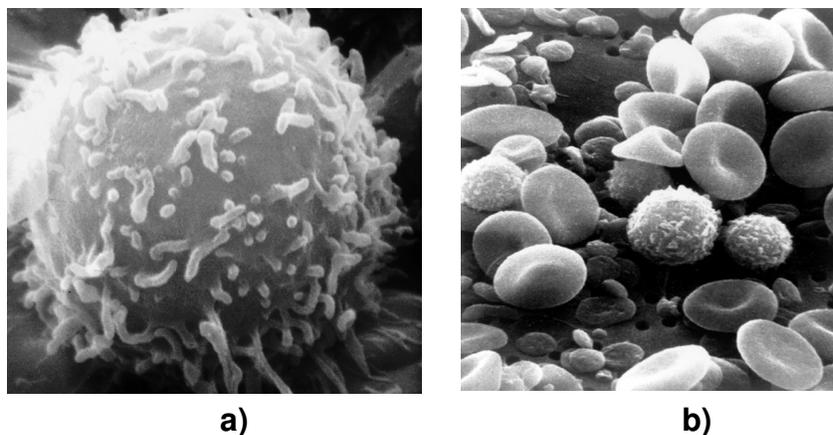
**Figura 4.1:** Antígeno particulado. Una bacteria o virus puede contener varios antígenos, por ejemplo la membrana de bacterias cuenta con muchos compuestos protéicos extraños al organismo huésped.

La respuesta global y coordinada del organismo tras la introducción del antígeno es la respuesta inmunológica. El tipo de respuesta que el sistema inmunológico pone en marcha depende del antígeno, teniendo muchas maneras de destruirlo, pero resulta útil considerar estas maneras en dos clases generales: inmunidad innata e inmunidad específica. Para propósitos de esta investigación solo se estudiará la inmunidad específica.

### 4.1 LA INMUNIDAD ESPECÍFICA

El sistema inmunológico específico evolucionó en los vertebrados primitivos y permite una respuesta inmunológica mayor, así como el establecimiento de la denominada memoria inmunológica, donde el antígeno es recordado. Las características de la respuesta inmunológica específica son esenciales para las funciones del sistema inmunológico [Tabla 5]. Las células que participan en la inmunidad específica son los linfocitos [Figura 4.2], éstas células cuentan con varios productos protéicos que son utilizados en la respuesta inmunológica, uno de estos productos son las inmunoglobulinas [Figura 4.3].

<sup>11</sup> Los polisacáridos son biomoléculas formadas por la unión de muchos monosacáridos (moléculas de azúcares simples).



**Figura 4.2:** Imágenes de microscopio electrónico del linfocito. a) Su tamaño se encuentra entre 5 y 20  $\mu\text{m}$ . b) Linfocitos junto a otras células [85].

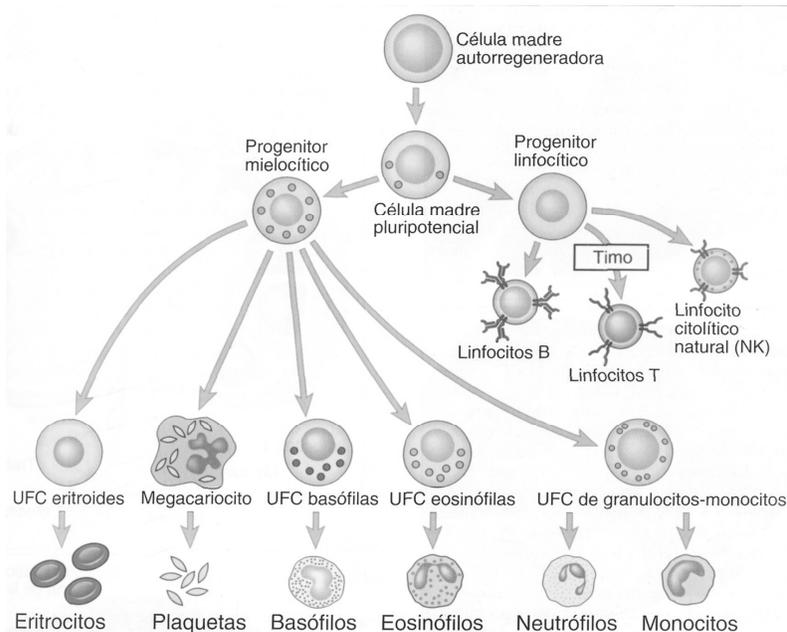


**Figura 4.3:** Modelo de listón de una molécula de inmunoglobulina. Cada inmunoglobulina está compuesta por cuatro cadenas polipeptídicas (tetrapéptido) [70].

**Tabla 5.** Características principales de la respuesta inmunológica específica.

Características	Función
Especificidad:	Las respuestas inmunológicas son específicas para los diferentes antígenos y, de hecho, para diferentes componentes estructurales de un complejo protéico o polisacárido.
Diversidad:	Es la capacidad del sistema inmunológico para responder a una amplia variedad de antígenos.
Memoria:	La exposición del sistema inmunológico a un antígeno extraño mejora su capacidad para responder al mismo antígeno en un segundo encuentro.
Especialización:	El sistema inmunológico genera respuestas que son adecuadas para defender al organismo contra distintos tipos de microorganismos.
Autolimitación:	Todas las respuestas inmunológicas normales disminuyen con el tiempo después de la eliminación del antígeno.
Discriminación entre lo propio y lo no propio:	Una de las propiedades más importantes del sistema inmunológico normal es su capacidad para reconocer, responder y eliminar antígenos extraños a la vez que no reaccionar perjudicialmente frente a materiales del organismo (antígenos propios), esto es, no despertar una respuesta inmunológica contra compuestos proteínicos u otros compuestos propios del individuo.

Los linfocitos son producidos en la médula ósea y en forma continua, estas son las únicas células capaces de reconocer antígenos de una forma específica y por ello son las células principales de la inmunidad específica. Las dos poblaciones más importantes de linfocitos son las células B (linfocitos B) y las células T (linfocitos T), y derivan de células madre hematopoyéticas<sup>12</sup> pluripotenciales<sup>13</sup>, en la Figura 4.4 se ilustran los dos progenitores de las células hemáticas, el mielocítico y el linfocítico.



**Figura 4.4:** Hematopoyesis. En este árbol hematopoyético se refleja el desarrollo de los diferentes linajes de células hemáticas, las cuales provienen de las células madre [70].

La mayoría de los antígenos, sobre todo los de naturaleza proteínica, son timo-dependientes; es decir, requieren de la participación de los linfocitos T, estos a su vez necesitan de la cooperación de otras células como los macrófagos que provienen de los monocitos, ya que la inmunogenicidad<sup>14</sup> de estos antígenos depende de su procesamiento por estas células o por otras células presentadoras<sup>15</sup> de antígeno para inducir una respuesta inmunológica completa.

Hay, por el contrario, un grupo de antígenos relativamente simples que sin la participación de células presentadoras de antígeno ni de linfocitos T, son capaces de inducir una respuesta inmunológica, estos son llamados antígenos timo-independientes; estos tipos de antígenos estimulan directamente a los linfocitos B y los inducen a producir inmunoglobulinas, además, son moléculas con unidades estructurales de repetición, usualmente polímeros lineales o poco ramificados, tanto naturales como sintéticos.

<sup>12</sup> La hematopoyesis es la producción y desarrollo de nuevas células de la sangre.

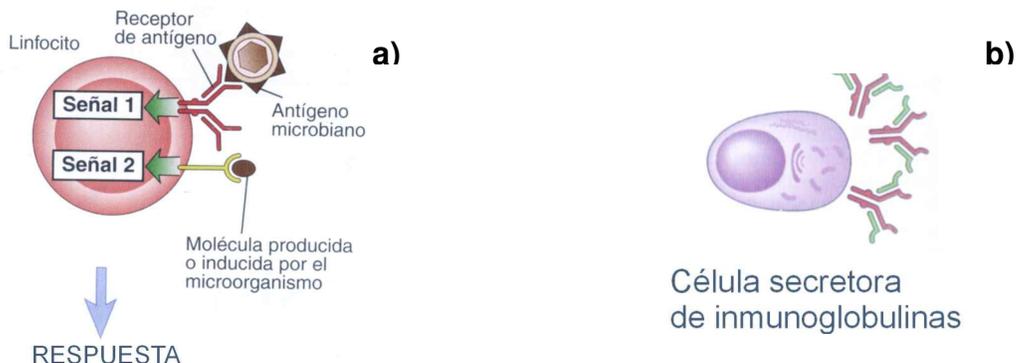
<sup>13</sup> Las células madre pluripotenciales pueden dar origen a progenitores que forman cualquiera de las tres capas germinales embrionarias.

<sup>14</sup> La inmunogenicidad es la propiedad que permite a una sustancia inducir una respuesta inmunológica.

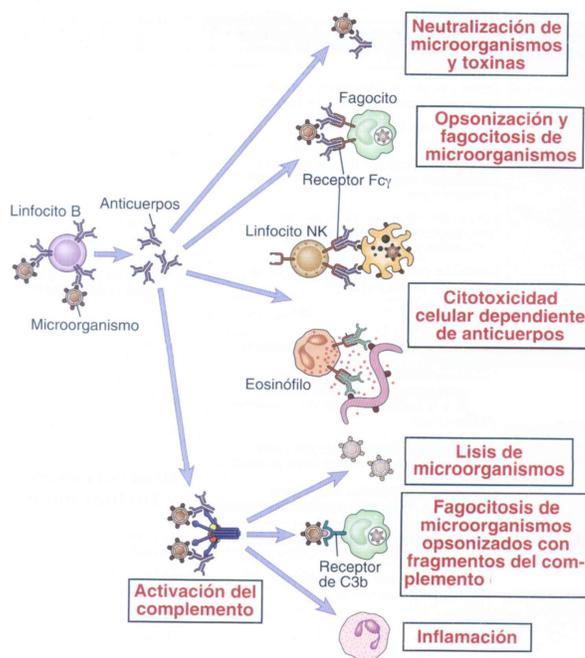
<sup>15</sup> Las células presentadoras de antígeno tienen como misión captar, procesar y presentar el antígeno a los linfocitos T.

## 4.2 LA INMUNOGLOBULINA

Las inmunoglobulinas son moléculas con una estructura en forma de Y, que pertenecen a las glicoproteínas y las producen los linfocitos B en forma secretada (las liberan) o unidas a su membrana plasmática de las células que la producen [Figura 4.5]. Las inmunoglobulinas unidas a membrana actúan como receptores que median la activación de los linfocitos B, la cual es desencadenada por el antígeno correspondiente. Las inmunoglobulinas secretadas actúan como mediadores específicos de diversos mecanismos efectores que sirven para eliminar a los antígenos fijados a las inmunoglobulinas [Figura 4.6].



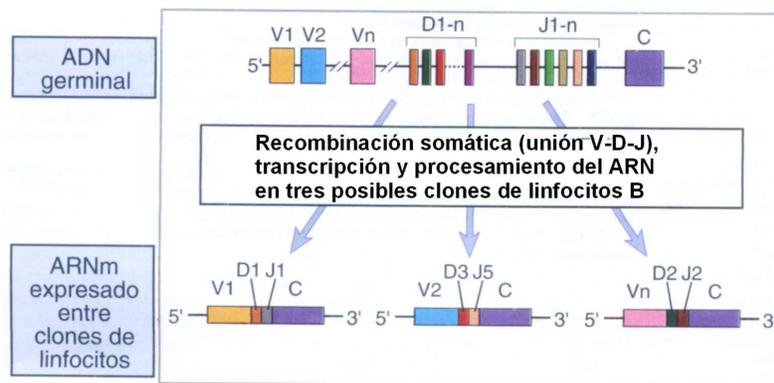
**Figura 4.5:** Inmunoglobulina como receptor y en forma secretada. a) El antígeno es identificado por algún linfocito que cuente con el receptor específico para detectarlo, dando una señal y formando una respuesta. b) Célula plasmática, es un linfocito B convertido a este tipo de célula para producir inmunoglobulina en grandes cantidades y secretarla [70].



**Figura 4.6:** Linfocito B activado y funciones efectoras de las inmunoglobulinas. Un linfocito B activado por el antígeno específico de sus receptores, secreta inmunoglobulina la cual dependiendo del tipo de antígeno realiza diversas funciones efectoras [70].

Cada molécula de inmunoglobulina está compuesta por cuatro cadenas polipeptídicas (tetrapéptido) unidas por puentes disulfuro, dos de estas cuatro cadenas son llamadas cadenas ligeras y las otras dos son llamadas cadenas pesadas. La formación de estas cadenas se da a través de la combinación aleatoria entre múltiples segmentos génicos germinales bien definidos y contenidos en estructuras, llamadas V, D, J y C de tal forma que las distintas combinaciones de cada uno de los segmentos génicos de cada estructura, generan la diversidad de inmunoglobulinas, este proceso es llamado recombinación somática [Figura 4.7]. El ADN de los segmentos génicos para codificar la cadena pesada se encuentra en el cromosoma 14 y los segmentos de la cadena ligera en el cromosoma 2, por lo que las estructuras no comparten segmentos entre cadenas pesadas y ligeras. La composición de cada par de cadenas es la siguiente:

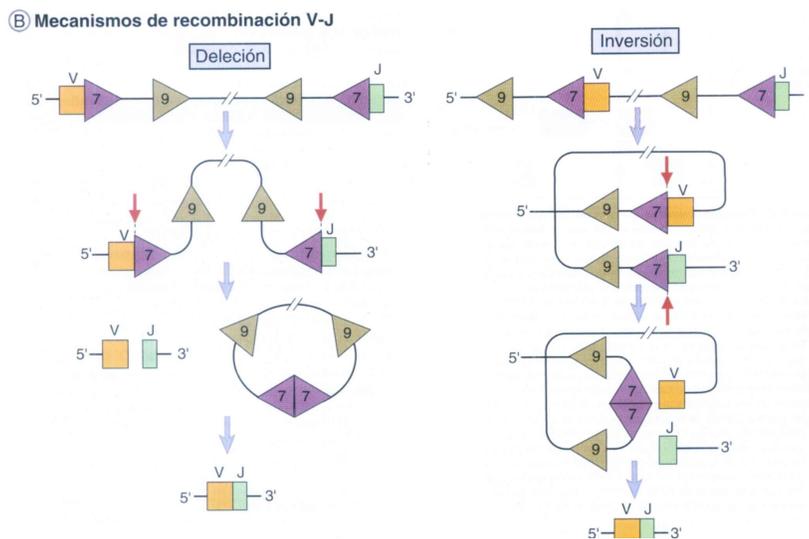
- Cadenas ligeras: Se dan a través de la expresión génica a partir de la recombinación somática de cada uno de los segmentos génicos de V, J y C. Se pueden dar dos tipos de cadenas ligeras, estas son  $\kappa$  y  $\lambda$ .
- Cadenas pesadas: Se dan a través de la expresión génica a partir de la recombinación somática de cada uno de los segmentos génicos de V, D, J y C.



**Figura 4.7:** Diversidad de los genes de las inmunoglobulinas. A partir del mismo ADN germinal se generan secuencias reordenadas de ADN y ARNm. En el ejemplo que se muestra, se producen tres posibles ARNm distintos a partir del mismo ADN germinal mediante el empleo de la combinación de los distintos segmentos génicos estructurados [70].

La recombinación somática es un procesamiento a nivel molecular que requiere de un mecanismo de regulación muy estricto, el cual se encuentra mediado por funciones coordinadas de enzimas llamadas recombinasas, que identifican, cortan y ligan los segmentos génicos para formar una secuencia bien estructurada de ADN, que posteriormente se transcribirá a un ARN primario y esta cadena tendrá otro mecanismo que reconoce, corta y liga el segmento C adecuado; realizado lo anterior se tendrá un ARNm que se traducirá a un polipéptido que formará a la cadena correspondiente de la inmunoglobulina. La Figura 4.8 ilustra como se realiza este reordenamiento.

En la Figura 4.8 se muestra la recombinación puede ocurrir por dos procesos llamados deleción o inversión. Si las secuencias de reconocimiento se orientan en la dirección opuesta, las recombinasas acercan los segmentos génicos V y J, y forman un bucle de ADN intercalado, este bucle no cumple ningún papel y es eliminado. El ADN intercalado se corta y los extremos de los segmentos V y J se empalman para completar el proceso de reordenamiento. Si las secuencias de reordenamiento se orientan en la misma dirección, el ADN se invierte y los segmentos V y J se empalman en un extremo. El mismo proceso es responsable de las recombinaciones DJ y VDJ en el locus de la cadena pesada [70].



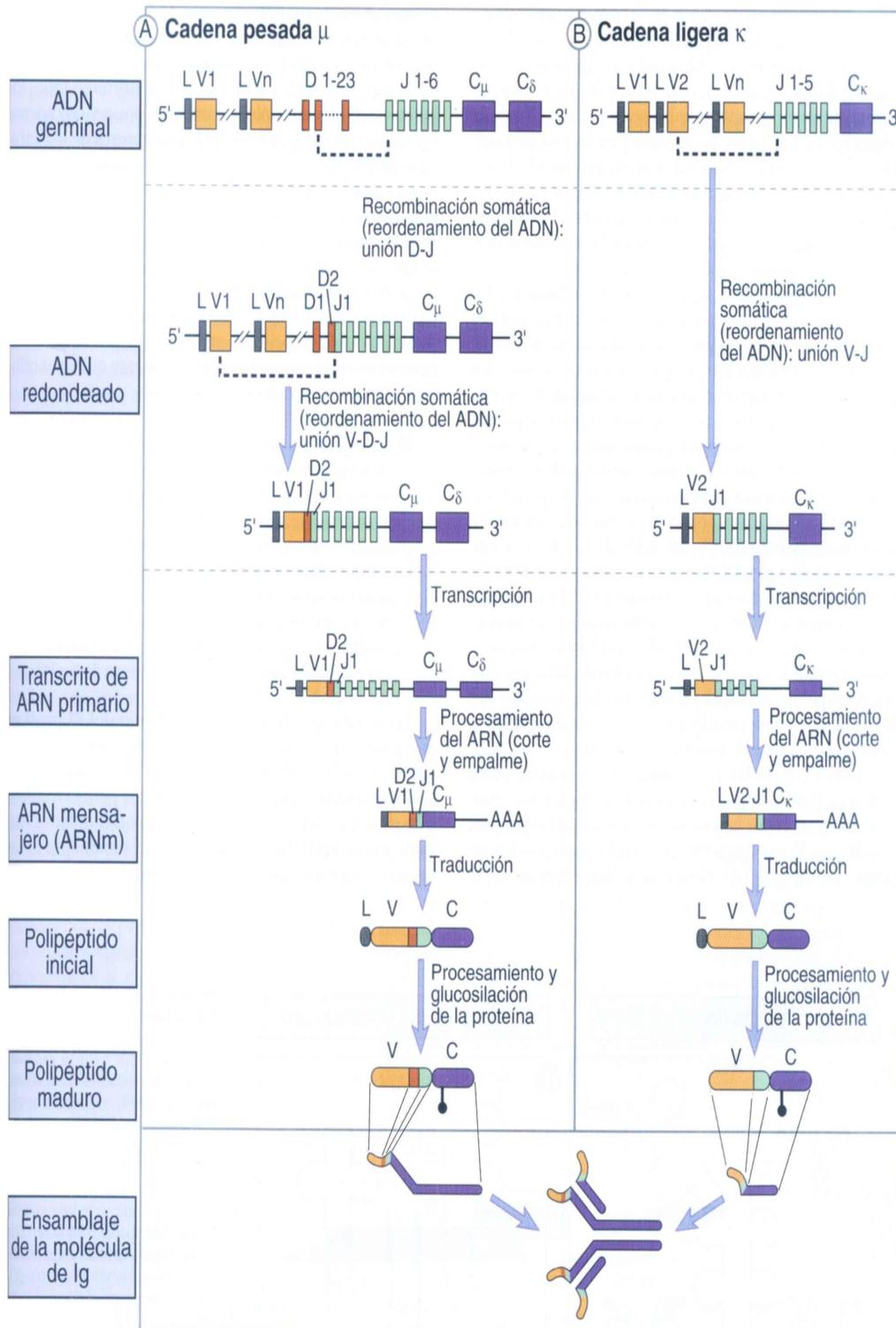
**Figura 4.8:** La recombinación de los segmentos V-(D)-J.

La Tabla 6 muestra el número de segmentos génicos de las estructuras V, D y J que le corresponden a las combinaciones de la cadena pesada y ligera (tipo K y λ) y el total de combinación al unirse las dos cadenas. Al traducirse el ARNm formado por la transcripción de los segmentos seleccionados se ensambla en una molécula proteínica, la cual es la inmunoglobulina, que tendrá una región variable (V) y otra constante (C) [Figura 4.9].

**Tabla 6.** Mecanismos que contribuyen a la generación de la diversidad en el repertorio de inmunoglobulinas [70].

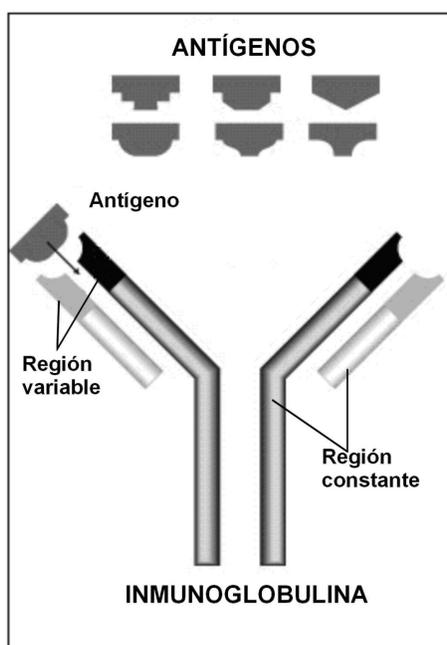
Segmentos	Cadena pesada	Cadena ligera K	Cadena ligera λ
ESTIMADO NÚMERO DE SEGMENTOS GÉNICOS PARA LA FORMACIÓN DE INMUNOGLOBULINAS EN EL HUMANO			
V	48	41	34
D	23	0	0
J	6	5	5
Combinación V-D-J	$48 \times 23 \times 6 = 6624$	$41 \times 5 = 205$	$34 \times 5 = 170$
Posibles combinaciones de cadenas pesadas y ligeras*	$6624 \times (205 + 170) = 2.48 \times 10^6$		

\*En la diversidad contribuyen también las regiones N y P que se dan solo en la cadena pesada.



**Figura 4.9:** Recombinación somática y expresión de los genes de las cadenas pesadas y ligeras de las inmunoglobulinas. Se muestra la secuencia de acontecimientos que tienen lugar en el reordenamiento del ADN y la expresión de los genes de la cadena pesada (A) y la cadena ligera (B). En el ejemplo A, la región variable (V) de la cadena pesada está codificada por los segmentos génicos V1, D1 y J1. En el ejemplo B, la región V de la cadena ligera está codificada por los segmentos génicos V1 y J1 [70].

La región variable de las dos cadenas de la inmunoglobulina contiene un fragmento de unión al antígeno, que es el sitio donde podrá unirse una parte específica de un antígeno, esta variabilidad se produjo como consecuencia de la recombinación somática, para adaptarse a cada uno de los posibles antígenos, [Figura 4.10], por lo tanto estas secuencias de aminoácidos constituyen los puntos de unión antigénica. A la región del antígeno reconocida por esa parte variable de la inmunoglobulina se le denomina epítipo o determinante antigénico. Un antígeno puede presentar un número variable de epítopos de estructura única o repetitiva. La complejidad estructural de las proteínas favorece que éstas presenten por lo general un número elevado de epítopos distintos, mientras que los ácidos nucleicos y los polisacáridos, dada su repetitividad estructural, poseen un número escaso de epítopos diferentes.



**Figura 4.10:** Región variable y constante de las cadenas pesadas y ligeras de la inmunoglobulina.

Cada linfocito cuenta con múltiples receptores de inmunoglobulina en su membrana plasmática pero todos estos receptores son específicos para un solo epítipo; al igual que la inmunoglobulina secretada por cada linfocito B, ya que tienen la misma región variable. Las inmunoglobulinas secretadas tendrán afinidad<sup>16</sup> al epítipo del antígeno que activó al linfocito, por lo que se fijarán en él, y la parte constante del antígeno servirá para diversas funciones efectoras.

<sup>16</sup> Afinidad es el grado de fuerza adhesiva entre moléculas.

### 4.3 EL PROCESO DE SELECCIÓN DURANTE LA MADURACIÓN DE LOS LINFOCITOS.

Una vez que los linfocitos B inmaduros localizados en la médula ósea (series de clones de linfocitos inmaduros) expresan receptores antigénicos, son sometidos a procesos de selección positiva y negativa. En la selección positiva, los linfocitos inmaduros con receptores antigénicos se unen a antígenos propios, estos no tendrán receptores específicos para los epítomos de estos antígenos, por lo que se seleccionan para sobrevivir y continuar su maduración. Estos linfocitos seleccionados positivamente alcanzan los tejidos linfáticos periféricos donde responden a antígenos extraños. En la selección negativa, los linfocitos que se unen con una avidéz elevada a los epítomos de antígenos propios del organismo, presentes en estos tejidos, reciben señales que les conducen a la muerte (apoptosis). Como resultado, el repertorio de linfocitos maduros carece de células capaces de responder a antígenos propios. La Figura 4.11 ilustra la selección de los linfocitos B.

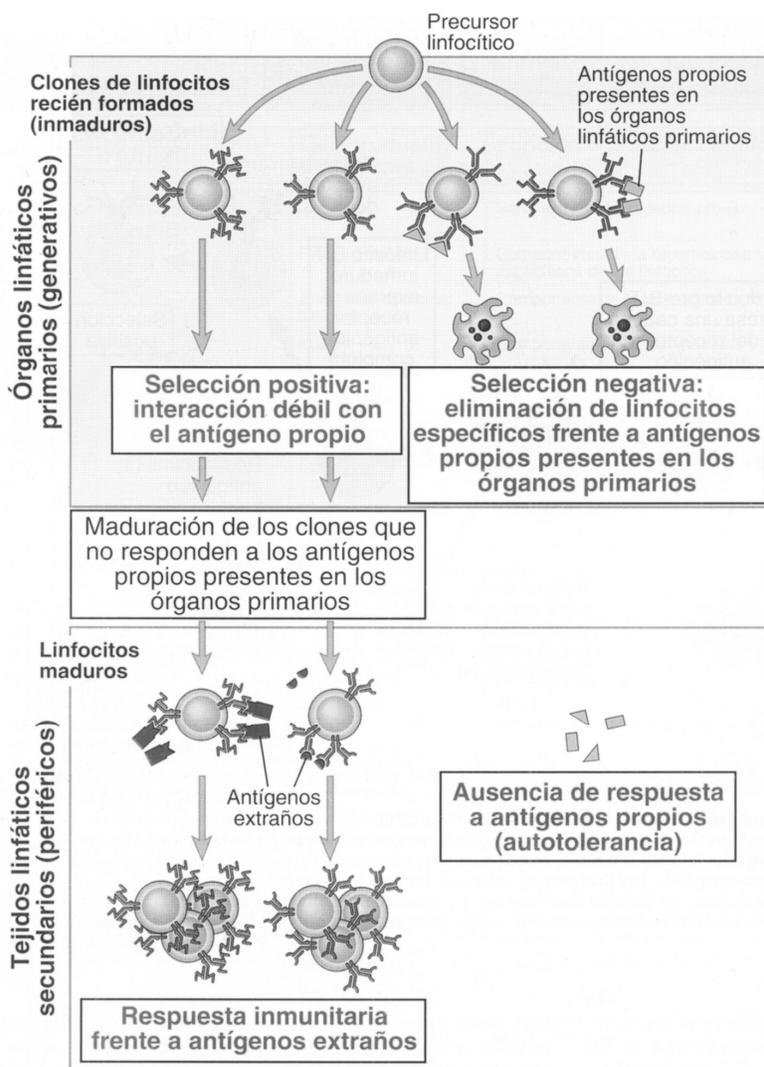


Figura 4.11: Selección positiva y negativa de los linfocitos inmaduros [70].

## 4.4 LA HIPÓTESIS DE SELECCIÓN CLONAL

Los linfocitos B maduros que se encuentran en los tejidos linfáticos periféricos tienen contacto con epítopos de antígenos que son transportados a ese tejido, se realiza una selección a una serie de linfocitos B con receptores de inmunoglobulina específicos a los epítopos de los antígenos transportados, y los activan, produciendo clones de esos linfocitos; algunos de estos linfocitos se cambian a células efectoras diferenciadas, como células de memoria o células plasmáticas (secretoras de inmunoglobulina). A esto se le llama la hipótesis de selección clonal, los principios básicos de esta teoría son mostrados en la **¡Error! No se encuentra el origen de la referencia..**

La Figura 4.12 ilustra cómo cada antígeno (X o Y) selecciona un clón de linfocitos específicos preexistentes y estimula la proliferación y diferenciación de dicho clón, dando lugar a las células plasmáticas, que secretarán inmunoglobulinas que tendrán afinidad con los epítopos de los antígenos produciendo distintas funciones efectoras de la inmunoglobulina, como se ilustra en la Figura 4.6.

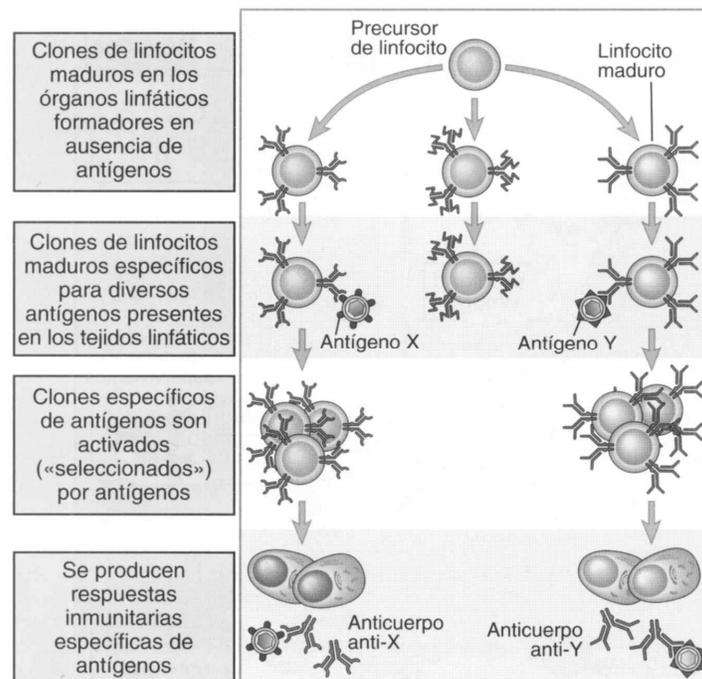


Figura 4.12: Selección clonal [70].

Los cuatro principios básicos de la hipótesis de la selección clonal:

- Cada linfocito posee un solo tipo de receptor de especificidad única.
- La interacción entre un epítipo de un antígeno y un receptor del linfocito capaz de unir esa molécula con alta afinidad conduce a la activación del linfocito.
- Las células efectoras diferenciadas derivadas de un linfocito activado tendrán receptores con especificidad idéntica a los de la célula progenitora que les dio origen.

- Los linfocitos que poseen receptores específicos para moléculas propias son eliminados en las etapas tempranas del desarrollo de las células linfoides y por lo tanto estarán ausentes en el repertorio de los linfocitos maduros.

#### 4.5 LA INMUNIDAD FRENTE A LOS TUMORES

Cada célula mantiene estables sus características morfológicas y fisiológicas gracias a la operación de diversos mecanismos de control o regulación que son sutiles y muy estrictos. La mayoría de estos mecanismos de control se originan pero operan en el núcleo, cuando alguno de los mecanismos falla, se altera la actividad genómica y, consecuentemente, la homeóstasis<sup>17</sup> celular. Un ejemplo de esto podría ser el cáncer.

Las células cancerosas pueden exhibir cambios de todo tipo, estas alteraciones incluyen modificaciones en su apariencia: forma y tamaño, alteraciones en la distribución de la cromatina nuclear, alteraciones en la forma y número de sus cromosomas: copias en exceso de cromosomas, pérdida de cromosomas y cromosomas aberrantes, alteraciones bioquímicas, en sus procesos moleculares, etc. Todos estos cambios tienen una causa genética, es decir, aparecen cuando ocurren alteraciones en los genes que no son corregidos por los mecanismos de reparación que contiene la célula de manera normal, cuando estos genes se alteran en su estructura o función, ocurre su pronta detección y reparación, usualmente a través de proteínas codificadas por otros genes; así la célula se mantiene en un orden. Algunas veces sin embargo, estas alteraciones en los genes activos no son detectadas y los genes afectados dejan de funcionar o funcionan de manera anormal, produciendo cantidades anormales de proteínas normales (subexpresión y sobre expresión) o produciendo proteínas anormales sin actividad fisiológica definida. Este funcionamiento anormal de los genes es lo que caracteriza a las células cancerosas y es la base de todas las alteraciones arriba señaladas.

La posibilidad de que respuestas inmunitarias específicas puedan erradicar los cánceres ha constituido el impulso para realizar numerosos trabajos en el campo de la inmunología. El concepto de vigilancia inmunitaria establece que una función fisiológica del sistema inmunológico consiste en reconocer y destruir a los grupos de células que han sido alteradas (llamadas células transformadas) antes de que produzcan más células con la misma característica y formen tumores, así como también eliminar los tumores una vez formados. Los resultados de algunos experimentos cuestionan la importancia e incluso la existencia de la vigilancia inmunológica, frente a distintos tipos de tumores. No obstante aún hoy no está claro si el sistema inmunológico reacciona frente a muchos tumores, pero el estudio para destruir los cánceres de una manera específica sigue siendo una meta importante para los inmunólogos que trabajan en el tema. Además, uno de los factores que influyen en el crecimiento de los tumores malignos es la capacidad de sus células para evadir los mecanismos de defensa del organismo.

Las células transformadas expresan antígenos que no aparecen en células normales. El sistema inmunológico específico considera a estos antígenos como extraños, lo que ocasiona que las células inmunitarias ataquen a las células transformadas. Los antígenos

---

<sup>17</sup> La homeóstasis es el conjunto de reacciones que realiza una célula o un conjunto de células que le permiten estar en equilibrio con el medio que las rodea.

---

---

expresados por los tumores pueden tener varios orígenes; algunos derivan de virus oncógenos como el virus del papiloma humano, que ocasiona cáncer del cuello uterino mientras que otros se forman por sobre expresiones y subexpresiones génicas en células tumorales. Un ejemplo de sobre expresión es una enzima llamada tirosinasa que, cuando se expresa en altos niveles, transforma a ciertas células de la piel (melanocitos) en tumores llamados melanomas.

Las observaciones clínicas y los experimentos en animales han demostrado que, aunque las células tumorales proceden de las propias células del huésped, pueden desencadenar respuestas inmunológicas. Sin embargo a menudo estas respuestas inmunológicas no pueden evitar el crecimiento de los tumores. Son varias las razones por las que la inmunidad antitumoral es incapaz de erradicar las células transformadas, en primer lugar las células tumorales proceden de las células huésped y, por lo tanto son iguales en muchos aspectos. En otras palabras, la mayoría de los tumores solo expresan unos pocos antígenos que pueden ser reconocidos como extraños, de modo que la capacidad inmunógena de gran parte de un conjunto de células transformadas tiende a ser escasa. Los tumores que desencadenan respuestas inmunológicas intensas son los producidos por virus oncógenos, en los que las proteínas del virus son antígenos extraños, y los tumores inducidos en animales por carcinógenos potentes que suelen provocar mutaciones génicas en las células normales. La respuesta inmunológica estimulada por diversos tumores es débil e indetectable y los estudios realizados con estos tumores dieron lugar a un escepticismo considerable sobre el concepto de vigilancia inmunitaria. En realidad, como ya se ha comentado, es probable que la importancia de la vigilancia inmunológica frente a los tumores varíe según el tipo de estos. En segundo lugar, el rápido crecimiento y la propagación de los tumores puede sobrepasar la capacidad del sistema inmunológico para erradicar las células tumorales y la eliminación del tumor requiere que todas las células tumorales sean eliminadas. En tercer lugar, muchos tumores disponen de mecanismos especializados para evitar las respuestas inmunológicas del huésped.

---

---

# CAPÍTULO 5

## RESULTADOS Y CONCLUSIONES

En este capítulo se presentan los resultados obtenidos sobre el estudio de procesamiento paralelo computacional aplicado a células del sistema inmunológico.

Dos preguntas que surgen en éste trabajo son:

- ¿Tiene el sistema inmunológico una componente de cómputo?
- Si tiene esta componente, ¿Que clase es?

Estos cuestionamientos también se han realizado en cada una de las áreas que conciernen a este trabajo: la ciencia del cómputo y la inmunología. Para responder a estas preguntas, se hizo una clasificación de los diversos estudios realizados que permiten ver las semejanzas y diferencias entre estas dos áreas, la clasificación es de acuerdo al lugar de la investigación y los elementos que se utilizan para su puesta en práctica, teniendo como base, los términos *In Silico*, *In Vitro*, *In Vivo* que se usan principalmente en la biología para indicar el tipo de análisis de sus investigaciones, la clasificación es la siguiente:

- Cómputo *In Silico* – La expresión *In Silico* hace referencia a las acciones que son realizadas en una computadora basada en el silicio. Son tres las disciplinas importantes que abarcan lo realizado con cómputo *In Silico* y la inmunología:
  - a) Simulaciones del sistema inmunológico – Las simulaciones son realizadas principalmente para la dinámica de las estructuras moleculares de las biomoléculas que constituyen los elementos del sistema inmunológico, tienen como meta predecir el comportamiento de dicho sistema ante diferentes situaciones, la unión con algunas áreas como la bioinformática y la biología molecular incrementa el poder de predicción [72].
  - b) Sistema inmunológico artificial (SIA) – Los principios biológicos modernos del sistema inmunológico han proveído las bases para esta creciente área de investigación, el SIA soluciona problemas de cómputo e ingeniería, usando las estrategias del sistema inmunológico, construyendo algoritmos que son implementados *In Silico* basados en los mecanismos del sistema inmunológico. Actualmente el SIA es representado por implementaciones que se realizan en conjunto con los algoritmos heurísticos, cómputo evolutivo, etc., unido con la explotación de varios conceptos de cómputo

emergentes como son las redes neuronales artificiales, algoritmos genéticos, etc. [73-76].

- c) **Inmunocómputo** – Tiene como antecedente al SIA y a los mecanismos biomoleculares del sistema inmunológico. El inmunocómputo explora los principios de procesamiento de información de las proteínas y la teoría de las redes inmunológicas<sup>18</sup>, para resolver complejos problemas, este concepto de computación se basa principalmente en los modelos matemáticos de las acciones que realizan las biomoléculas del sistema inmunológico. Se han desarrollado rigurosas pruebas en aplicaciones que utilizan el concepto de las redes inmunológicas junto con el cómputo *In Silico*, encontrando que son capaces de aprender y reconocer, lo cual permite resolver problemas como el reconocimiento de patrones, con similares características al neurocómputo (redes neuronales artificiales) [77].
- **Cómputo *In Vitro*** – El término *In Vitro* se refiere a realizar experimentos biológicos en tubos de ensayo o en un medio externo controlado, tratando de proporcionar las condiciones semejantes a las que ocurren en un organismo biológico. Muchos experimentos en biología celular y molecular acontecen fuera de los organismos o de las células; porque las condiciones de prueba pueden no corresponder a las condiciones dentro del organismo, ésto puede dar lugar a resultados inexactos. Por lo tanto, tales resultados experimentales se denominan como resultados *In Vitro*, en contraste con el *In Vivo*. Es aquí donde la mayor parte de los experimentos de cómputo con ADN se producen, por lo que nos referimos al término de cómputo *In Vitro* a todo procedimiento de cómputo biomolecular que se realiza en este medio. Los trabajos en el cómputo *In Vitro* y el sistema inmunológico consisten en aplicar las teorías y algoritmos del SIA y el inmunocómputo en las computadoras de biomoléculas. Un ejemplo de estos trabajos sería el inmunochip, que son microarreglos biológicos (biochips) en este caso construidos para resolver problemas de cómputo [78].
  - **Cómputo *In Vivo*** – El término *In Vivo* se refiere a los experimentos producidos dentro de un organismo biológico, así como a la interacción entre las biomoléculas que se encuentran funcionando normalmente en el organismo, también a la experimentación o acción propia del organismo hecha en el tejido vivo, en comparación con uno parcial o muerto. Las pruebas animales y los ensayos clínicos son formas de investigación de este tipo. Por lo tanto, el término cómputo *In Vivo* es referido a todo procedimiento de cómputo biomolecular que se realiza *In Vivo*. Esta tesis abarca este campo

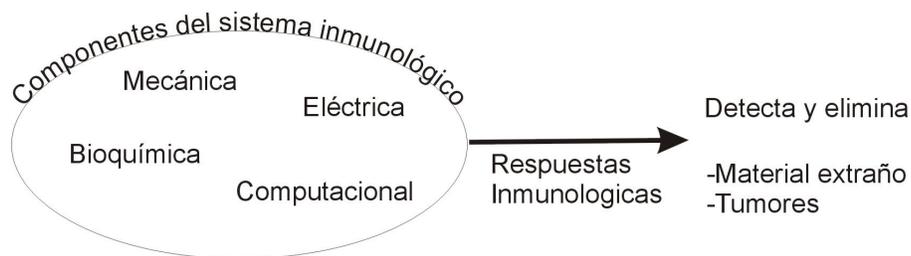
De aquí en adelante nos referiremos como sistema inmunológico al sistema inmunológico real, aquel que forma parte de un organismo biológico.

<sup>18</sup>La red inmunológica es el sistema inmunológico visto como una red regulada de moléculas y células (red idiotípica)

De acuerdo a los objetivos de esta tesis y a la clasificación antes mencionada, se sitúa a este trabajo en el estudio de las operaciones de cómputo *In Vivo* que se realizan en el sistema inmunológico [Figura 5.1].

Para realizar este estudio lo primero es responder a las dos preguntas del inicio de este capítulo.

- ¿Tiene el sistema inmunológico una componente de cómputo? [Figura A1]



**Figura 5.1:** Algunos componentes del sistema inmunológico.

Para responder a lo anterior, se propuso analizar el procesamiento de información a nivel molecular en el sistema inmunológico, realizando una abstracción del funcionamiento de este sistema como procesos de cadenas de ADN.

Obteniendo, que en las células linfoides (linfocitos) se realiza un procesamiento de cadenas de ADN, llamado recombinación somática, éste procesamiento puede ser mostrado con cadenas de un lenguaje en la Figura 5.2.



**Figura 5.2:** Recombinación somática como procesamiento de cadenas.

Lo cual se explica de la siguiente forma:

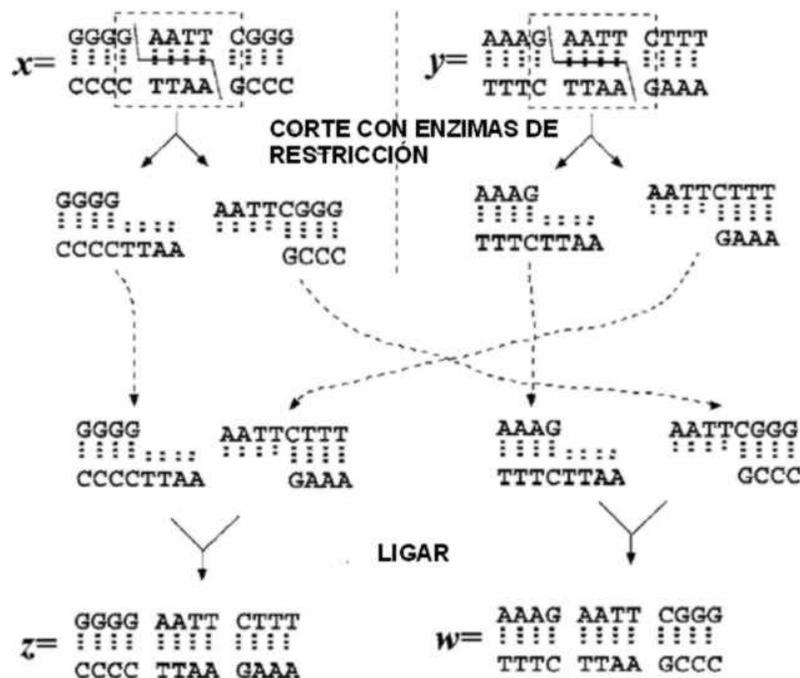
A partir de un gen (cadena A, ADN de longitud limitada) contenido en todas las células precursoras linfocíticas, se obtienen por la recombinación somática secuencias ARNm (cadenas B, ARN de longitud limitada) contenidas en los linfocitos inmaduros. Cada linfocito inmaduro contendrá solamente alguna cadena B.

El siguiente punto a considerar es reglamentar el procesamiento de información, esto es, formalizar las operaciones que se producen en el procesamiento de las cadenas de ADN en los linfocitos con los conceptos de cómputo, para lo cual se estudiaron diversos modelos teóricos de cómputo con ADN. Comprobando que las operaciones que se producen en las cadenas de ADN y ARN son iguales a las operaciones del modelo de cómputo con ADN, denominado sistema splicing, para entender esto se explicará a continuación dicho modelo.

## 5.1 SISTEMA SPLICING

La teoría del sistema splicing propuesta por T. Head [79], permite contar con un modelo para realizar cómputo con biomoléculas, esta teoría fue originalmente introducida para modelar fenómenos bioquímicos de recombinación de ADN y analizar sus propiedades como un lenguaje, mas tarde fue reconocida como una base fundamental del cómputo con ADN [80].

El funcionamiento de este sistema radica en operaciones que son llamadas splicing, que consisten en la acción de enzimas que cortan y ligan moléculas de ADN, la endonucleasa (enzima de restricción) es la encargada de cortar una molécula de ADN en un sitio específico con base en una secuencia de nucleótidos específica y la ligasa une dos moléculas de ADN en sus extremos. La Figura 5.3 ilustra un ejemplo en el cual se tienen dos moléculas de ADN de doble cadena (X, Y) que son cortadas por la misma endonucleasa, y posteriormente sus fragmentos son ligados mediante algunas condiciones, obteniéndose moléculas con nuevas secuencias de nucleótidos (Z, W).



**Figura 5.3:** Funcionamiento central de los sistemas splicing. Generación de nuevas secuencias de nucleótidos por corte y ligado de moléculas de ADN por medio de endonucleasas y ligasas.

En el sistema splicing solo se utiliza una cadena de las dos cadenas de una molécula de ADN de doble cadena, esto sin afectar la operación. Del ejemplo de la Figura 5.3, se representan las cadenas de ADN X y Y de la forma siguiente:

$$X = \text{GGG G|AATTC GGG}$$

$$Y = \text{AAA G|AATTC TTT}$$

En donde la cadena GAATTC es la secuencia que reconoce la endonucleasa y la barra vertical (|) indica el lugar donde la cadena es cortada.

Para formalizar las operaciones de splicing, las cadenas y subcadenas son representadas con el formato siguiente:

$$X = x_1u_1u_2x_2$$

$$Y = y_1u_3u_4y_2$$

donde:

u<sub>1</sub>, u<sub>3</sub> son subcadenas de X y Y respectivamente y son las secuencias de reconocimiento anterior al corte

u<sub>2</sub>, u<sub>4</sub> son subcadenas de X y Y respectivamente y son las secuencias de reconocimiento posterior al corte

Una regla de splicing especifica la secuencia de nucleótidos que la enzima reconoce y el lugar donde realiza el corte es descrita como:

$$(u_1, u_2 ; u_3, u_4)$$

donde los cortes se realizan entre las secuencias u<sub>1</sub> - u<sub>2</sub>, y la secuencia u<sub>3</sub> - u<sub>4</sub>.

De lo anterior se obtiene:

$$x_1 = \text{GGG}, u_1 = \text{G}, u_2 = \text{AATTC}, x_2 = \text{GGG}$$

$$y_1 = \text{AAA}, u_3 = \text{G}, u_4 = \text{AATTC}, y_2 = \text{TTT}$$

Aplicando una operación de splicing se obtienen las siguientes cadenas de dos moléculas de ADN:

$$Z = \text{GGGG AATT CTTT} = x_1u_1u_4y_2$$

$$W = \text{AAAG AATT CGGG} = y_1u_3u_2x_2$$

Una operación de splicing es denotada por  $\vdash_r$  y los parámetros completos son escritos de la forma siguiente:

$$(X, Y) \vdash_r (Z, W) \text{ si}$$

$$\begin{aligned} X &= x_1 u_1 u_2 x_2 \\ Y &= y_1 u_3 u_4 y_2 \\ \\ Z &= x_1 u_1 u_4 y_2 \\ W &= y_1 u_3 u_2 x_2 \\ \\ &\text{Para alguna } x_1, x_2, y_1, y_2 \\ &\text{Donde } r = (u_1, u_2 ; u_3, u_4) \end{aligned}$$

Formalizando el sistema splicing con base en la teoría de lenguajes propuesta por T. Head [79] se determina lo siguiente:

Una regla de splicing  $r$  ahora será formalizada a una cadena y consiste del formato:

$$r = u_1 \# u_2 \$ u_3 \# u_4$$

donde:

$u_1, u_3$  son subcadenas de  $X$  y  $Y$  respectivamente, y son las secuencias de reconocimiento anterior al corte  
 $u_2, u_4$  son subcadenas de  $X$  y  $Y$  respectivamente, y son las secuencias de reconocimiento posterior al corte

donde  $u_1, u_2, u_3, u_4 \in V^*$ , siendo  $V$  un alfabeto.

$\#, \$$  son símbolos especiales que no pertenecen al alfabeto  $V$  y utilizados para delimitar.

Si lo que se manipula son cadenas de ADN, el alfabeto es  $V = \{ A, G, C, T \}$ , por lo que cualquier cadena (secuencia) se encuentra contenida en  $V^*$ .

Al realizar un cómputo con el sistema splicing, este puede utilizar más de una regla, el conjunto de todas las reglas de splicing es representado por  $R$ :

$$R \subseteq V^* \# V^* \$ V^* \# V^*, \quad (\text{note que } R \text{ puede ser infinito}).$$

Por lo que  $r_n \in R$  con  $n \geq 1$

Sea  $L$  el lenguaje formado por las cadenas iniciales llamados axiomas siendo estas cadenas un subconjunto de  $V^*$ , en el ejemplo de la Figura 5.3,  $X$  y  $Y$  forman el lenguaje  $L$  con cadenas del alfabeto  $V = \{ G, A, T, C \}$ .

La Figura 5.4 ilustra una operación de splicing con los parámetros siguientes:

$$(X, Y) \vdash_r Z$$

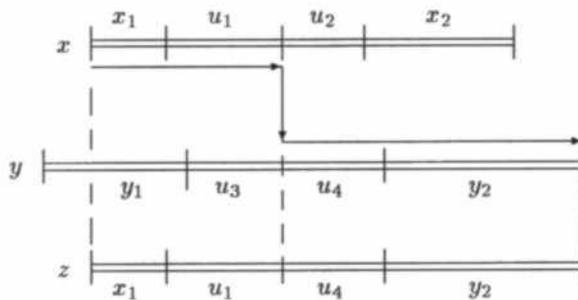
$$X = x_1u_1u_2x_2$$

$$Y = y_1u_3u_4y_2$$

$$Z = x_1u_1u_4y_2$$

Para  $r = u_1\#u_2\$u_3\#u_4$

Donde X, Y y Z son cadenas sobre el alfabeto V y r una regla de splicing.



**Figura 5.4:** La operación de cómputo con ADN del sistema splicing [80].

Una combinación H es denotada por  $\sigma$  y es un par  $(V, R)$

$$\sigma = (V, R)$$

Sea  $\sigma(L)$  el lenguaje obtenido por una primera operación de splicing no iterativo de algún par de cadenas (X y Y) que pertenecen a L con respecto a alguna regla de splicing r que pertenece a R.

$$\sigma(L) = \{ Z \in V^* \mid (X, Y) \vdash_r Z, \text{ para algún } X, Y \in L, r \in R \}$$

Cuando alguna endonucleasa y ligasa se encuentran presentes ya sea In Vivo o In Vitro, estas enzimas no se inactivan después de una operación de corte y pegado, por lo que las enzimas actúan iterativamente.

Sea  $\sigma^0(L)$  el conjunto de cadenas iniciales.

$$\sigma^0(L) = L$$

Sea  $\sigma^1(L)$  el lenguaje que resulta de la operación de la primera iteración, este se encuentra dado por las cadenas que pertenecen a L, junto con un primer splicing no iterativo:

$$\sigma^1(L) = \sigma^0(L) \cup \sigma(\sigma^0(L)) = L \cup \sigma(L)$$

El lenguaje obtenido de la iteración  $i$  está dado por la expresión

$$\sigma^{i+1}(L) = \sigma^i(L) \cup \sigma(\sigma^i(L)), \text{ para } i \geq 1$$

En consecuencia,

$$\sigma^*(L) = \bigcup_{i \geq 1} \sigma^i(L)$$

El ejemplo siguiente muestra un splicing no iterativo e iterativo, donde  $L$  es un lenguaje que consiste de una sola cadena  $\{ abba \}$  sobre el alfabeto  $\Sigma = \{ a, b \}$ ,  $r_1$  y  $r_2$  son un conjunto de reglas de splicing, las cuales son un subconjunto de  $R$ :

$$\{r_1 : a\#b\$b\#ba, r_2 : b\#a\$b\#ba\} \text{ donde } r_1 \text{ y } r_2 \in R$$

Por lo que se obtiene.

$$(a|bba, ab|ba) \vdash_{r_1} aba$$

$$(abb|a, ab|ba) \vdash_{r_2} abbba$$

$$\sigma(L) = \{ aba, abbba \}$$

Splicing iterativo

$$\sigma^0(L) = \{ abba \}$$

De la primera iteración se obtiene:

$$(a|bba, ab|ba) \vdash_{r_1} aba$$

$$(a|ba, ab|ba) \vdash_{r_1} aba$$

$$(ab|a, ab|ba) \vdash_{r_2} abba$$

$$(abb|a, ab|ba) \vdash_{r_2} abbba$$

$$\sigma^1(L) = \{ aba, abba, abbba \}$$

$$\sigma^i(L) = \{ ab^n a \mid 1 \leq n \leq i+2 \}, \text{ Para } i \geq 1$$

En consecuencia,

$$\sigma^*(L) = \{ ab^n a \mid n \geq 1 \}$$

La operación de splicing se emplea como herramienta para la construcción de los mecanismos principales de estos sistemas, dado un conjunto de axiomas, los cuales son cadenas que pertenecen a  $L$  y un conjunto de reglas de splicing  $R$ , el lenguaje generado estará formado por todas las cadenas que se puedan obtener al aplicar las reglas de splicing, tanto al conjunto de axiomas iniciales como a las cadenas resultantes obtenidas en operaciones de splicing precedentes.

Se representan por FIN, REG, LIN, LC, SC y RE las familias de lenguajes generados por gramáticas arbitrarias donde REG representa los lenguajes regulares, LIN lineales, LC libres de contexto, SC sensibles al contexto y RE recursivamente enumerables, FIN representa la familia de lenguajes finitos.

Cumpliendo con la jerarquía Chomsky, que es definida como:

$$FIN \subset REG \subset LIN \subset LC \subset SC \subset RE$$

Dadas 2 familias de lenguajes FL1, FL2, donde L pertenece a FL1 y R pertenece a FL2,  $S(FL1, FL2)$  es la familia del lenguaje generado a la cual pertenece el sistema splicing no iterativo; la Tabla 7 muestra las diferentes combinaciones:

$$S(FL1, FL2) = \{ \sigma(L) \mid L \in FL1 \text{ y } \sigma = (V, R) \text{ con } R \in FL2 \}$$

**Tabla 7. El tamaño de familias  $S(FL1, FL2)$  [25].**

FL1 \ FL2	FIN	REG	LIN	LC	SC	RE
FIN	FIN	FIN	FIN	FIN	FIN	FIN
REG	REG	REG	REG, LIN	REG, LC	REG, RE	REG, RE
LIN	LIN, LC	LIN, LC	RE	RE	RE	RE
LC	LC	RE	RE	RE	RE	RE
SC	RE	RE	RE	RE	RE	RE
RE	RE	RE	RE	RE	RE	RE

Se le llama a un sistema splicing no iterativo cuando los lenguajes que forman el conjunto de axiomas y reglas pertenecen a FIN, esto tendrá la capacidad de generar lenguajes que pertenezcan a FIN, por lo que el poder de cómputo de este dispositivo será menor al de un autómata finito (AF). De la Tabla 7 se obtiene:

$$S(FIN, FIN) = FIN \subset REG$$

De otra manera, para generar lenguajes RE conforme a los axiomas y reglas, requerirá que alguno de estos últimos se encuentre en una familia de lenguajes con una jerarquía Chomsky más alta a FIN, por lo que se requerirá de lenguajes no finitos.

$$S(SC, FIN) = REG$$

La familia de SC contiene lenguajes que no son finitos, por tanto, habrá al menos un lenguaje que represente a un conjunto de axiomas que será infinito, por lo que este sistema no sería práctico.

La familia de lenguajes generados por el sistema splicing iterativo esta dada por la expresión siguiente:

$$H(FL1, FL2) = \{ \sigma^*(L) \mid L \in FL1 \text{ y } \sigma = (V, R) \text{ con } R \in L2 \}$$

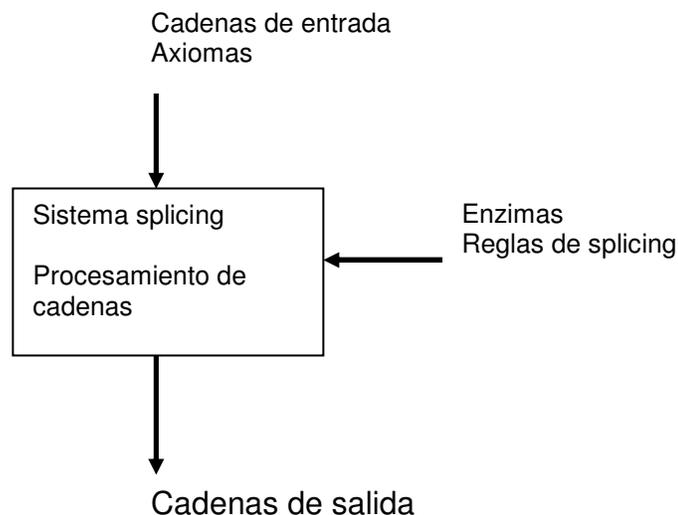
La caracterización de este sistema ante axiomas y reglas finitas es:

$$H(FIN, FIN) = REG$$

Se ha demostrado la existencia de modelos del sistema splicing que bajo una serie de controles y tienen el mismo poder de cómputo que la MT, por lo que cualquier lenguaje RE puede generarse con un número finito de axiomas y un número finito de reglas mediante este sistema de operaciones de corte y ligado con moléculas de ADN [25].

$$H(FIN, FIN) = RE$$

El funcionamiento general del sistema splicing se ilustra en la Figura 5.5.



**Figura 5.5:** Funcionamiento general del sistema splicing.

Una vez entendidas las operaciones de cadenas de ADN que se realizan en el sistema inmunológico y en el sistema splicing,. Se realizarán las comparaciones que identifican al sistema inmunológico con operaciones que se efectúan en el sistema splicing.

## 5.2 EL SISTEMA INMUNOLÓGICO CARACTERIZADO COMO UN SISTEMA SPLICING

Dentro de las variantes del sistema splicing, existen algunas que utilizan cadenas lineales y circulares para realizar sus operaciones, estas variantes son mostradas en la Figura 5.6,

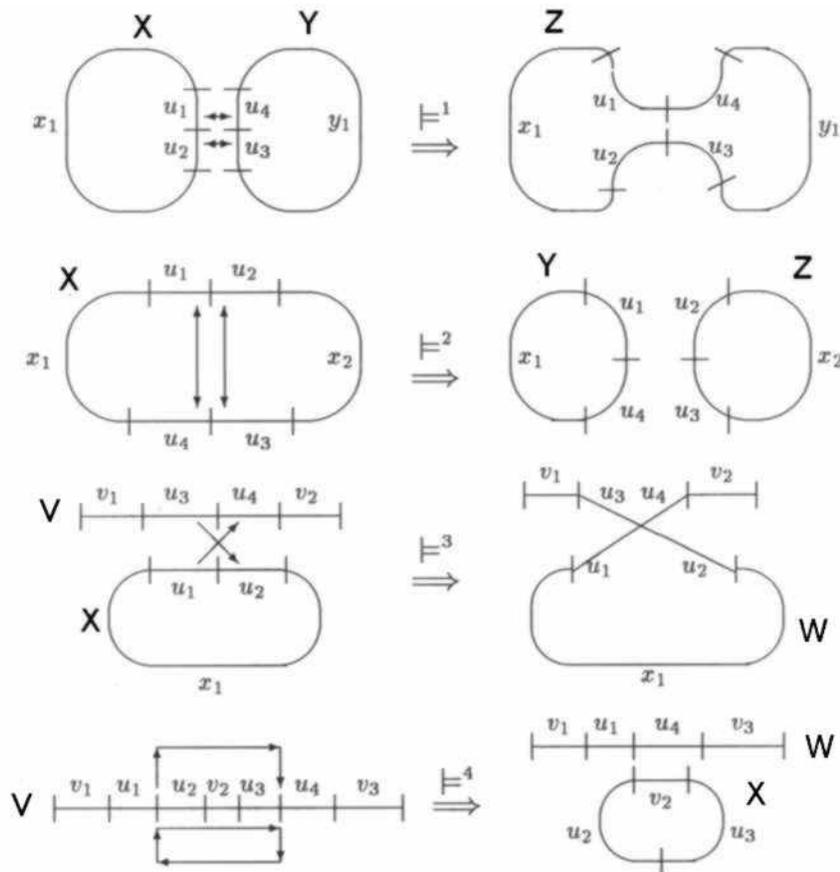


Figura 5.6: Splicing con cadenas circulares [25].

Las consideraciones del sistema splicing con cadenas circulares son las siguientes:

Se utilizan dos clases de cadenas, las cadenas lineales que pertenecen a  $V^*$  y las cadenas circulares que pertenecen a  $V^\circ$  (estrella de Kleen de cadenas circulares) todas ellas sobre el alfabeto  $V$ ; el sistema splicing con cadenas circulares cuenta con las mismas características del sistema splicing de cadenas solamente lineales, como estas tienen modelos que cuentan con el mismo poder de cómputo al de la MT, las operaciones de splicing de la Figura 5.6 son formalizadas a continuación:

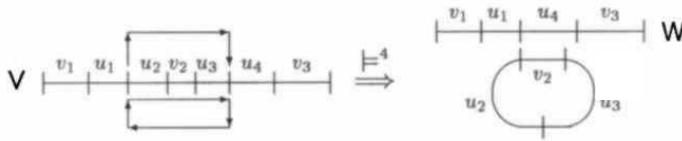
La regla de splicing se denota igual que el splicing con cadenas lineales, esto es  $r = u_1 \# u_2 \$ u_3 \# u_4$  sobre  $V$ . Para  $X, Y$  y  $Z \in V^\circ$  y  $V, W \in V^*$

Donde  $V^\circ$  es el conjunto de todas las cadenas circulares  
 $V^*$  es el conjunto de todas las cadenas lineales

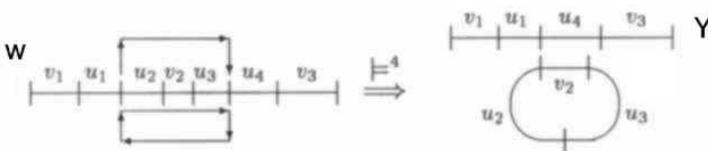
$$\begin{aligned}
 (X, Y) \vdash_r^1 Z & \text{ si solo si } & X &= x_1 u_1 u_2, \\
 & & Y &= y_1 u_3 u_4, \\
 & & Z &= x_1 u_1 u_4 y_1 u_3 u_2, \\
 & & & \text{para alguna } x_1, y_1 \in V^*, \\
 X \vdash_r^2 (Y, Z) & \text{ si solo si } & X &= x_1 u_1 u_2 x_2 u_3 u_4, \\
 & & Y &= x x_1 u_1 u_4, \\
 & & Z &= x_2 u_3 u_2, \\
 & & & \text{para alguna } x_1, x_2 \in V^*, \\
 (X, V) \vdash_r^3 W & \text{ si solo si } & X &= x_1 u_1 u_2, \\
 & & V &= v_1 u_3 u_4 v_2, \\
 & & W &= v_1 u_3 u_2 x_1 u_1 u_4 v_2, \\
 & & & \text{para alguna } v_1, v_2, v_3 \in V^*, \\
 V \vdash_r^4 (X, W) & \text{ si solo si } & V &= v_1 u_1 u_2 v_2 u_3 u_4 v_3, \\
 & & X &= u_2 v_2 u_3, \\
 & & W &= v_1 u_1 u_4 v_3, \\
 & & & \text{para alguna } v_1, v_2, v_3 \in V^*
 \end{aligned}$$

En la Figura 5.7 se da el algoritmo de cómputo con ADN, que realiza el sistema inmunológico para obtener las secuencias de ARNm de la cadena pesada, en el cual se realizan 3 operaciones del sistema splicing con cadenas circulares; por lo tanto se concluye que el repertorio total de linfocitos B crea un sistema de cómputo con ADN.

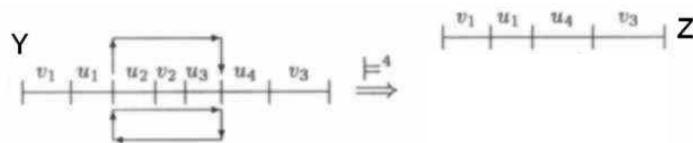
V = ADN GERMINAL  
 Operación de splicing 1  
 u1 = D  
 u4 = J  
 u2,u3 = Intrones



W = ADN CON RECOMBINACIÓN D-J  
 Operación de splicing 2  
 u1 = LV  
 u4 = DJ  
 u2, u3 = Intrones



Y = ADN CON RECOMBINACIÓN V-D-J  
 Operación de splicing 3  
 u1 = LVDJ  
 u4 = C  
 u2,u3 = Intrones



Z = LVDJC = ARNm

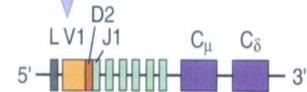
**Cadena pesada**



Recombinación somática (reordenamiento del ADN):  
 unión D-J



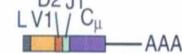
Recombinación somática (reordenamiento del ADN):  
 unión V-D-J



Transcripción



Procesamiento del ARN (corte y empalme)



Traducción



Procesamiento y glucosilación de la proteína

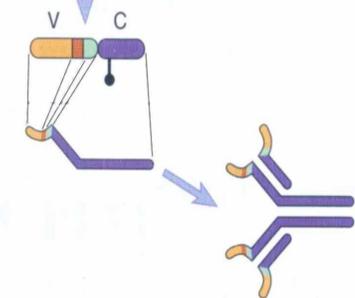
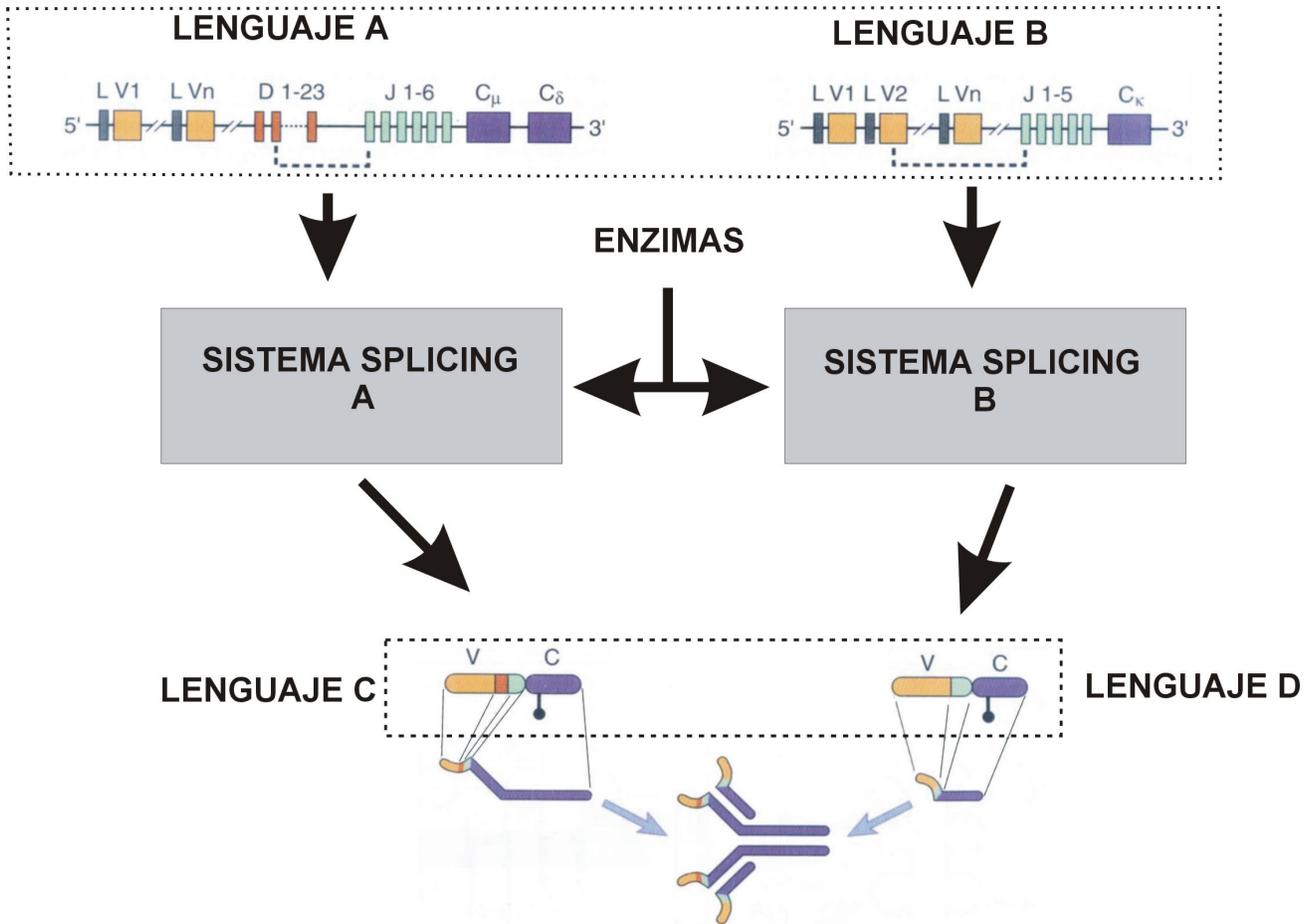


Figura 5.7: Algoritmo de cómputo con ADN del sistema splicing [70].

Las operaciones mostradas en la Figura 5.7 pueden representarse de la forma mostrada en la Figura 5.8, la cual ilustra que el lenguaje A y B son los axiomas, la acción de las enzimas son las reglas de splicing para la formación del lenguaje C y D.



**Figura 5.8:** Representación del cómputo con ADN mediante el modelo de sistema splicing que realiza una célula del sistema inmunológico [70].

Bajo las bases de la teoría de selección clonal que forma parte del sistema inmunológico y la teoría del cómputo con ADN, se concluye lo siguiente: los linfocitos B inmaduros contienen el repertorio total de soluciones, las cuales en el proceso de selección que se realiza durante la maduración de los linfocitos B y la selección clonal es igual al filtrado que se realiza también en el cómputo con ADN in vitro, dando como resultado la solución requerida para atacar el problema, y comprobando también que el tipo de computadora que es el sistema inmunológico es no determinística.

Por lo tanto, como el sistema inmunológico en el componente de procesamiento de cadenas realiza cómputo con ADN y este cómputo es un sistema de procesamiento paralelo masivo computacional, los estudios que se realicen en el cómputo con ADN ayudarán a la comprensión del sistema inmunológico:

Referente a la computabilidad y complejidad computacional, se concluye lo siguiente:

Se tiene que las diferentes sobre expresiones que realiza la célula de manera anormal causada por diversas razones, como es un código genético corrupto, coinciden con un sistema de procesamiento de información que no cuenta con un control (no se detiene) y el sistema inmunológico (ultima barrera de ataque de un mal funcionamiento respecto a estas sobre expresiones), no tendrá la certeza de que este problema pueda ser resuelto, porque el sistema inmunológico se enfrenta a un problema de la parada y al igual que otros sistemas tanto artificiales creados por el hombre (como son los marcadores de pronostico) o natural creados por un organismo biológico tendrán esa certeza de resolverlo, el problema simplemente es no computable.

Si de acuerdo a los análisis de cómputo con ADN a mayor velocidad de procesamiento, mayor masa requerida, el sistema inmunológico tiene por lo tanto limites en la complejidad computacional.



---

---

# REFERENCIAS

- [1] Shapiro E. and Benenson Y. (2006) Bringing DNA computers to life. *Sci. Amer., May*, 45-51.
- [2] Regev A. and Shapiro E. (2002) Cells as computation. *Nature*, 419, 343.
- [3] Thrall J. (2004) Nanotechnology and Medicine. *Radiology*, 230, 315-318.
- [4] Amos M. (2004) Cellular Computing. Oxford University Press.
- [5] Turing A. (1950) Computing machinery and intelligence. *Mind*, 59, 433-460.
- [6] Schrödinger E. (1943) What is life?
- [7] Monod J. (1970) El Azar y la Necesidad
- [8] Bray D. (1995) Protein molecules as computational elements in living cell. *Nature*, 376, 307-312.
- [9] Feynman R. (1961) There's plenty of room at the bottom. *Miniaturization. Editor D. Gilbert. Editions Reinhold, New York*, 282-296.
- [10] Bennett C. (1982) The Thermodynamic of computation – a review. *Intl. J. Theor. Phys.*, 21, 905-940.
- [11] Adleman L. (1994) Molecular computation of solutions to combinatorial problems. *Science*, 266, 1021-1024.
- [12] Adleman L. (1998) Computing with DNA. *Sci. Amer., August*, 54-61.
- [13] Lipton R. (1995) DNA Solution of Hard Computational Problems. *Science*, 268, 542-545.
- [14] Beaver D. (1996) Universal molecular computer, DNA based computers. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 27, 29-36.
- [15] Henkel C., Bäck T., Kok J., Rozenberg G. and Spaink H. (2007) DNA computing of solution to Knapsack problems. *BioSystems*, 88, 156-162.
- [16] Head T., Rozenberg G., Bladergroen R., Breek C., Lommerse P. and Spaink H. (2000) Computing with DNA by operating on plasmids. *BioSystems*, 57, 87-93.
- [17] Lazebnik Y. (2002) Can a Biologist Fix a Radio? or, What I Learned while Studying Apoptosis. *Cancer Cell*, 2, 179-82.
- [18] Ulaby F. (2006) Engineering in the age of biology. *Proc. IEEE*, 94, 863-864.
- [19] Endy D. (2005) Foundations for engineering biology. *Nature*, 438, 449-453
- [20] Abelson H., Sussman G. (1996) Structure and Interpretation of Computer Programs - 2nd Edition, The MIT press.

- 
- 
- [21] Denning P. (2005) Is computer science science? *Communications of the ACM*, 48, 27-31.
- [22] Church A. (1936) An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58, 345-363.
- [23] Turing A. (1936) On computable numbers with an application to the Entscheidungsproblem. *Proceedings of London Mathematical Society*, 42, 230-265.
- [24] Harel D. (1992) *Algorithmics, The Spirit of Computing*. Addison Wesley Publishing, second edition.
- [25] Paun G. (1998) *DNA Computing, New Computing Paradigms*. Springer-Verlag.
- [26] Chworos A., Severcan I., Koyfman A., Weinkam P., Oroudjev E., Hansma H. and Jaeger L. (2004) Building Programmable Jigsaw Puzzles with RNA.
- [27] Yan H. (2004) Nucleic Acid Nanotechnology. *Science*, 306, 2048-2049.
- [28] Ding B. and Seeman N. (2006) Operation of a DNA Robot Arm Inserted into a 2D DNA Crystalline Substrate. *Science*, 314, 1583 – 1585.
- [29] Rothemund P. (2006) Folding DNA to create nanoscale shapes and patterns. *Nature*, 440,297-302.
- [30] Watson J. and Crick F. (1953) Genetical implications of the structure of deoxyribonucleic acid. *Nature*, 171, 964-967.
- [31] Watson J. and Crick F. (1953) Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171, 737-738.
- [32] Monod J. and Jacob F. (1961) Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, 3, 318-356.
- [33] Lewin B. (2000) *Genes VII*. Oxford University Press.
- [34] Wolynes P. (1998) Computational biomolecular science. *Proc. Natl. Acad. Sci. USA*, 95, 5848.
- [35] Henkel C. (2005) Experimental DNA computing. *Doctoral thesis. Institute of Biology and Leiden Institute of Advanced Computer Science, Faculty of Mathematics & Natural Sciences, Leiden University*.
- [36] Bray D. (1995) Protein molecules as computational elements in living cell. *Nature*, 376, 307-312.
- [37] Rozen D. (1996) Molecular computing: Does DNA compute?. *Current Biology*, 6, 254-257.
- [38] Shapiro E. and Gil B. (2007) Logic goes in vitro. *Nature nanotechnology*, 2, 84-85.
- [39] Pool R. (1995) A Boom in Plans for DNA computing. *Science*, 268, 498-499.
- [40] Rodríguez-Patón A. (1999) Variantes de la concatenación en computación con ADN. Tesis Doctoral, Facultad de informática de la Universidad Politécnica de Madrid.
- [41] Benner S. (2004) Redesigning Genetics. *Science*, 306, 625-626.

- 
- 
- [42] Dirks R., Lin M., Winfree E. and Pierce N. (2004) Paradigms for computational nucleic acid design, *Nucleic Acids Research*, 32, 1392-1403.
- [43] Liu Q., Wang L., Frutos A., Condon A., Corn R. and Smith L. (2000) DNA Computing on Surface. *Nature*, 403, 175-178.
- [44] Sakamoto K., Gouzu H., Komiya K., Kiga D., Yokoyama S., Yokomori T. and Hagiya M. (2000) Molecular Computation by DNA Hairpin Formation. *Science*, 288, 1223-1226.
- [45] Braich R., Chelyapov N., Johnson C., Rothmund P. and Adleman L. (2002) Solution of a 20-Variable 3-SAT Problem on a DNA Computer. *Science*, 296, 499-502.
- [46] Liu W., Gao L., Xu G., Zhu X., Liu X. and Xu J. (2005) A Random Walk DNA Algorithm for the 3-SAT Problem. *Current Nanoscience*, 1, 85-90.
- [47] Adleman L., Rothmund P., Roweis S. and Winfree E. (1996) On Applying Molecular Computation To The Data Encryption Standard, Proceedings of the Second Annual Meeting on DNA Based Computers, Princeton University, 28-48.
- [48] Adleman L. (1996) On constructing a molecular computer. DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 27, 1-22.
- [49] Schneider T. (1994) Sequence Logos, Machine/Channel Capacity, Maxwell's Demon, and Molecular Computers: a Review of the Theory of Molecular Machines. *Nanotechnology*, 5, 1-18.
- [50] Yokomori T. (2002) Molecular computing paradigm – toward freedom from Turing's charm. *Natural Computing*, 1, 333-390.
- [51] Guarnieri F., Fliss M. and Bancroft (1996) Making DNA Add. *Science*, 273, 220-223.
- [52] Stojanovic M., Mitchell T. and Stefanovic D. (2002) Deoxyribozyme-Based Logic Gates. *J. Am. Chem. Soc.*, 124, 3555-3561.
- [53] Fontana W. (2006) Pulling String. *Science*, 314, 1552-1553.
- [54] Faulhammer D., Cukras A., Lipton R. and Landweber L. (2000) Molecular computation: RNA solution to chess problems, *Proc. Natl. Acad. Sci.*, 97, 1385-1389.
- [55] Stojanovic M. and Stefanovic D. (2003) A deoxyribozyme-based molecular automaton. *Nature Biotechnology*, 21, 1069-1074.
- [56] Tsiftaris S., Katsaggelos A., Pappas T. and Papoutsaki E. (2004) DNA Computing from a Signal Processing Viewpoint. *IEEE Signal Processing Magazine*, September, 100-105.
- [57] Tsiftaris S., Katsaggelos A., Pappas T. and Papoutsaki E. (2004) How Can DNA Computing Be Applied to Digital Signal Processing?. *IEEE Signal Processing Magazine*, November, 57-61.
- [58] Deaton R. and Garzon M. (2001) Fuzzy Logic with Biomolecules. *Soft Computing*, 5, 2-9.

- 
- 
- [59] Ouyang Q., Kaplan P., Liu S. and Libchaber A. (1997) DNA Solution of the Maximal Clique Problem. *Science*, 278, 446-449.
- [60] Kari L., Gloor G. and Yu S. (2000) Using DNA to solve the Bounded Post Correspondence Problem. *Theor. Comput. Sci.*, 231, 193-203.
- [61] Ho M., (2005) Fast parallel molecular solutions for DNA-based supercomputing: the subset-product problem. *BioSystems*, 80, 233-250.
- [62] Linial M. and Linial N. (1995) On the potential of Molecular Computing. *Science*, 268, 481.
- [63] Lo Y., Yiu K. and Wong S. (1995) On the potential of Molecular Computing. *Science*, 268, 481-482.
- [64] Bunow B. (1995) On the potential of Molecular Computing. *Science*, 268, 482-483.
- [65] Reif J. (1995) Parallel molecular computation. *Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures*, 213-223.
- [66] Adleman L, Rothmund P., Roweis S. and Winfree E. (1999) On Applying Molecular Computation to the Data Encryption Standard. *Journal of Computational Biology*, 6, 53-64.
- [67] Boneh D., Dunworth C. and Lipton R. (1996) Breaking DES using a molecular computer. *DNA Based Computers, The AMS DIMACS Series in Discrete Mathematics and TCS*, 27, 35-65.
- [68] Clelland T., Risca V. and Bancroft C. (1999) Hiding message in DNA microdots. *Nature*, 399, 533-534.
- [69] Fu P. (2007) Biomolecular computing: Is it ready to take off?. *Biotechnol. J.* 2, 91-101.
- [70] Abbas K., Andrew H. (2003) *Inmunología Celular y Molecular*, Quinta edición, Elsevier.
- [71] Espinosa R. (1996) *Inmunología de memoria* 2da Edición. Panamericana.
- [72] Flower D. (2007) Immunoinformatics, Predicting Immunogenicity *In Silico*. *Methods in Molecular Biology*. Humana Press.
- [73] Farmer J., Packard N. and Perelson A. (1986) The Immune System, Adaptation, and Machine Learning. *Physica D*, 2, 187-204.
- [74] Cohen I. (2007) Real and artificial immune systems: computing the state of the body. *Nature Reviews Immunology*, 7, 569- 574.
- [75] Carter J. (2000) The Immune System as a Model for Pattern Recognition and Classification. *J. Am. Med Inform. Assoc.*, 7, 28-41.
- [76] Cortés N. (2004) Sistema inmune artificial para solucionar problemas de optimización. *Tesis de doctorado, Cinvestav-IPN México*.
- [77] Tarakanov A., Akormin V. and Sokolova S. (2003) *Immunocomputing. Principles and applications*. Springer.
- [78] Goncharova L., Jacques Y., Vide C., Tarakanov O. and Timmis J. (2005) Biomolecular Immune-Computer: Theoretical Basis and Experimental Simulator.

---

---

Springer/ Heidelberg.

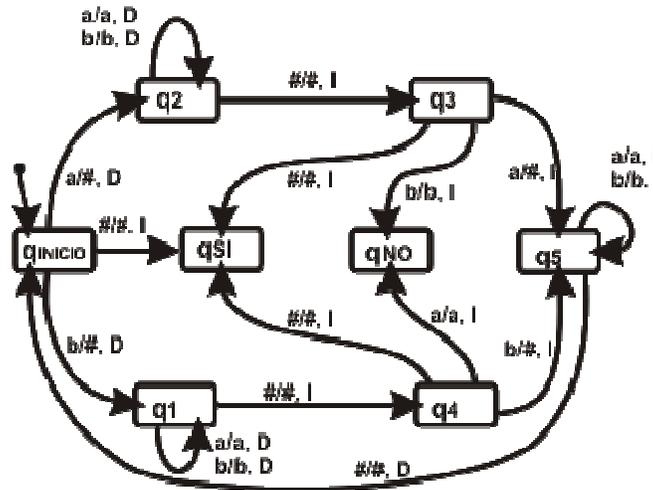
- [79] Head T. (1987) Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. *Bullet. Math. Biol.* 49, 737-759.
- [80] Hoogeboom H. and Vugt N. (1998) The power of H systems: does representation matter?. *Computing with Bio-Molecules. Theory and Experiments.*
- [81] Amos M. (1997) DNA Computation. Thesis for the degree of Doctor of Philosophy. *Department of Computer Science of The University of Warwick, UK.*
- [82] Benenson Y., Paz-Elizur T., Adar R., Keinan E., Livneh Z. and Shapiro E. (2001) Programmable and autonomous computing machine made of biomolecules. *Nature*, 414, 430-434.
- [83] Benenson Y., Gil B., Ben-Dor U., Adar R. and Shapiro E. (2004) An autonomous molecular computer for logical control of gene expression. *Nature*, 429, 423-429.
- [84] Marin J. (2004) Lenguajes formales y teoría de la computación. Mc Graw Hill, Tercera edición.
- [85] Triche (1976) National Cancer Institute
- [86] Amos M., Paun G., Rozenberg G. and Salomaa A. (2002) Topics in the theory of DNA computing. *Theoretical Computer Science*, 287, 3-28.
- [87] Bolívar F. (2004) Fundamentos y casos exitosos de la biotecnología moderna. Colegio Nacional, México.
- [88] López M., Romero G., Mallorquín P. y Vega M. (2006) Biología Sintética. GENOMA ESPAÑA/CIBT-FGUAM. Fundación Española para el desarrollo de la investigación en Genómica y Proteómica/Fundación General de la Universidad Autónoma de Madrid.
- [89] Weiss, R., *et al.*, (2003). Genetic circuit building blocks for cellular computation, communications, and signal processing. *Natural computing*, 2, 47-84.
- [89] (<http://www.nhgri.nih.gov/DIR/VIP/>) National Human Genome Research Institute. <http://www.genome.gov/>

---

---

## ANEXO I – LA IMPLEMENTACIÓN DE LA FUNCIÓN DE TRANSICIÓN

La Figura 1 ilustra una función de transición de un MT que reconoce cadenas de palíndromos del alfabeto  $\Sigma = \{a,b\}$  y el símbolo # para identificar el espacio en blanco. El conjunto de estados son  $q_{INICIO}$ ,  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$ ,  $q_5$ ,  $q_{SI}$ ,  $q_{NO}$ , siendo  $q_{INICIO}$  el estado de inicio,  $q_{SI}$  el estado de aceptación y  $q_{NO}$  el estado de no aceptación.



**Figura 1.** Función de transición de una MT representada por un diagrama. Esta decide si la cadena puesta es un palíndromo [24].

Se pone el cabezal de la MT en el primer símbolo del lado izquierdo de la cinta, que no identifique el espacio en blanco y en el diagrama se inicia en el estado  $q_{INICIO}$ . Para que se lleve acabo la siguiente transición de estado, el movimiento del cabezal y el símbolo que será escrito antes del movimiento serán dados por el símbolo que lee el cabezal y el estado de transición actual. Un ejemplo de la lectura del paso de transición es el siguiente  $a/\#, D$  = (símbolo que se lee) / (símbolo que se escribe) , (movimiento del cabezal). Cuando la transición llega a un estado final  $q_{SI}$  o  $q_{NO}$  la MT cumplió su cometido, pero existen máquinas en las que no sucede esto.

---

---

## ANEXO II. EL PROBLEMA DE SATISFACCIÓN BOOLEANA (SAT)

El problema de satisfacción booleana (SAT) es uno de los problemas más importantes, existe mucho interés en resolverlo de manera eficiente, ya que ese método de resolución sería aplicable a un gran número de problemas de diferente naturaleza.

Una instancia del problema SAT es una expresión booleana que combina variables booleanas con operadores booleanos (AND, OR, NOT). Una expresión es satisfactible si existe una asignación de valores booleanos (cierto, falso) para las variables de esa expresión haciendo que la expresión completa sea verdadera. Para este problema existen muchas variantes, una de ellas es el FNC-SAT (SAT de la Forma Normal Conjuntiva) que será descrito a continuación:

Sea una expresión lógica, la cual contiene variables  $x_i$  y los operadores booleanos  $\wedge$ ,  $\vee$  y  $\neg$  (AND, OR y NOT, respectivamente). Se supone que la expresión está en forma FNC, lo cual significa que es una conjunción de subexpresiones  $C_i$  llamadas cláusulas disyuntivas:

$$F = C_1 \wedge C_2 \wedge \dots \wedge C_n$$

Por ejemplo:

$$F_1 = (x_1 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee x_4 \vee \neg x_2) \wedge (\neg x_3) \wedge (x_2 \vee \neg x_4)$$

Es una expresión FNC con cinco elementos conjuntivos.

El problema de FNC-SAT es el siguiente: Dada una expresión en forma normal conjuntiva, ¿existe una asignación de verdad que la satisfaga?

Por ejemplo:

Una instancia de problema SAT con 2 variables y su correspondiente tabla de la verdad [Tabla 8].

$$F_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$$

**Tabla 8.** Tabla de verdad para la función  $F_2$ .

$x_1$	$x_2$	$x_1 \vee x_2$	$\neg x_1 \vee \neg x_2$	$F_2$
Cierto	Cierto	Cierto	Falso	Falso
Cierto	Falso	Cierto	Cierto	Cierto
Falso	Cierto	Cierto	Cierto	Cierto
Falso	Falso	Falso	Cierto	Falso

Se puede verificar fácilmente que la expresión precedente se satisface (se vuelve cierto) con la asignación de ciertas combinaciones de valores booleanos en las variables. El problema de SAT pertenece a NP (No determinista) dado que puede ser resuelto en tiempo polinómico con una máquina de Turing no-determinista que genera todas las posibles combinaciones de valores para las variables de la expresión y, en forma no-determinista, intenta verificar si alguna de ellas hace que la expresión se evalúe en verdadero, en cuyo caso acepta la entrada.

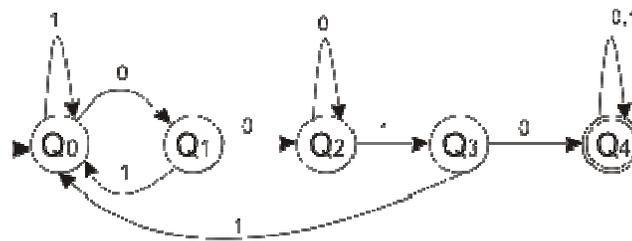
---

---

### ANEXO III. MÁQUINA DE TURING NO DETERMINISTA (MTND)

Para entender de manera práctica la máquina de Turing no determinista (MTND) explicaremos un modelo de menor poder computacional que la MT, el autómata finito determinístico (AFD), que en términos prácticos se describe como una MT que no tiene la capacidad de escribir en la cinta y únicamente el cabezal se puede mover de izquierda a derecha. El siguiente ejemplo ilustrará un problema, el cual sí tiene el poder de resolver el AFD. Ejemplo:

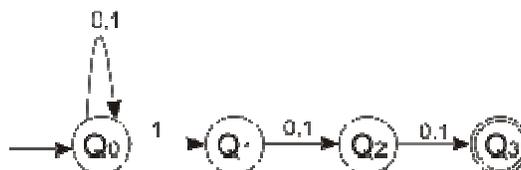
Se tiene el alfabeto  $\Sigma = \{0,1\}$ , construir un AFD cuyo lenguaje aceptado sean todas cadenas que siempre y cuando tenga una longitud de por lo menos 3 símbolos y un 1 en la antepenúltima posición de la cadena. El diagrama de la función de transición es mostrado en la Figura 2, donde:  $Q_0$  es el estado inicial y  $Q_4$  el estado de aceptación y cualquier otro estado es un estado de no aceptación.



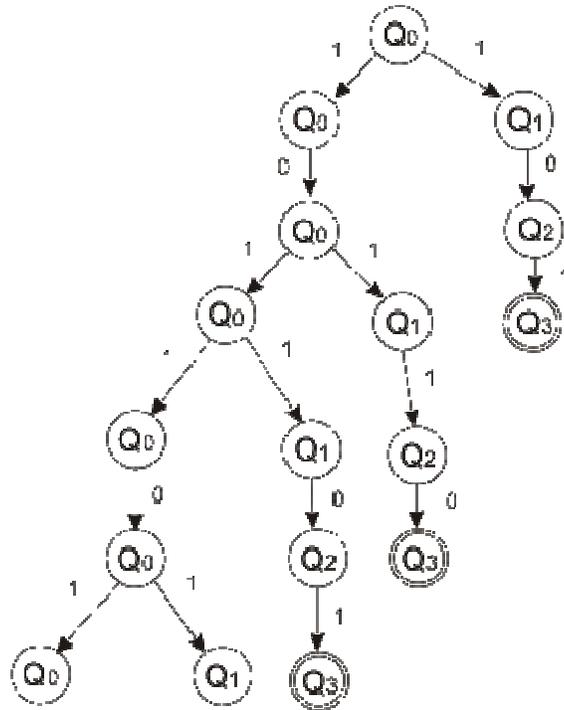
**Figura 2.** Diagrama de una función de transición de un Autómata Finito Determinístico.

Por ejemplo si introducimos la cadena 100, ésta es aceptada por que es de longitud tres y tiene un 1 en la antepenúltima posición de la cadena, si introducimos 0110101 es aceptada también.

Ahora mostraremos un autómata finito no determinístico (AFND) que resuelve el mismo problema y acepta el mismo lenguaje del problema anterior. En la Figura 3 se muestra el diagrama de la función de transición no determinista. Se puede observar que para el estado  $Q_0$  existen dos transiciones para el símbolo 1, Esto se ejemplifica de la siguiente manera; se da la cadena 101101 al AFND, en la Figura 4 se muestra el árbol de transiciones resultante, esto es, todas las transiciones que ocurren del inicio de la cadena hasta el final.



**Figura 3.** Diagrama de una función de transición de un Autómata Finito No Determinístico.



**Figura 4.** Árbol del AFND de la Figura 3.

El determinismo radica en lo siguiente: en lugar de preguntarse si el trayecto correspondiente a una cadena lleva a un estado de aceptación como lo haría con un AFD, hay que preguntarse si algún trayecto correspondiente a la cadena tiene dicho efecto. En el seguimiento del diagrama [Figura 3, Figura 4], cuando se está en el estado  $Q_0$  y se recibe el símbolo de entrada 1, se adivina cuál flecha con el rotulo 1 es la que debe seguir. La adivinación incorrecta podría originar una respuesta “no” para una cadena que sea parte verdadera del lenguaje. Ello no invalida la estrategia, ya que para una cadena del lenguaje existe al menos una secuencia de adivinación que lleva a la aceptación, y para una cadena que no sea parte del lenguaje, ninguna secuencia de adivinación hace que se acepte la cadena [84]. Debe tenerse en cuenta que todos los trayectos deben llegar a un final y sobre todo, el no determinismo no aumenta la potencia de la MT, pero puede reducir el menor número de estados en relación a un AF, como en el caso de este ejemplo, además que puede variar la complejidad del cómputo, esto es, el tiempo y espacio.

Ahora veamos un ejemplo de cómo sería la MTND:

Una forma determinista de probar si un entero es una potencia de 2 es verificar en primer término si el entero es 1 y en caso de que no lo sea, realizar unas secuencias de divisiones entre dos. Si en cualquier paso antes de llegar a uno se obtiene un residuo distinto de cero, la respuesta es negativa. Será positiva en caso de que se obtenga al final, el cociente 1 sin en alguna de las divisiones el residuo sea cero. Sin embargo, normalmente se diría que multiplicar por 2 es más fácil que dividir entre 2. Así pues, un enfoque más sencillo sería comenzar con 1 y realizar una secuencia de multiplicaciones por 2, Se compararía el resultado de cada una de las multiplicaciones contra la entrada

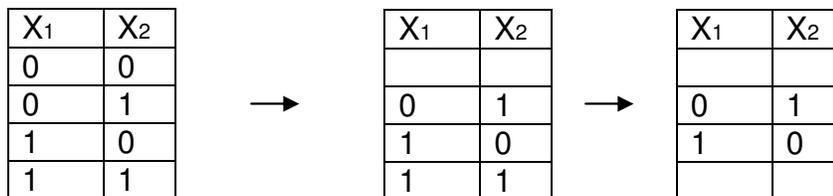
original (el entero a verificar), que se aceptaría eventualmente si se obtiene un número igual a la entrada, mientras que se rechaza en caso de que se obtenga en última instancia un número mayor que la entrada o simplemente se deja que las iteraciones continúen indefinidamente. En la MTND se adivinará un número  $i$ , se obtendrá  $2i$  y se someterá a prueba este valor con el entero a verificar, pero existen simultáneamente otras adivinaciones de  $i$  que se evaluarán.

Ahora analizaremos un algoritmo que resuelve el SAT en forma no determinista. Tenemos el problema de FNC-SAT anterior.

$$F_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$$

Por lo que su algoritmo no determinista sería el siguiente [Figura 5].

- Se generan soluciones.
- Atraemos las soluciones en un tiempo donde  $x_1$  sea igual a cierto o  $x_2$  sea igual a cierto.
- Atraemos las soluciones en un tiempo donde  $x_1$  sea igual a falso o  $x_2$  sea igual a falso.



**Figura 5.** Algoritmo SAT no determinístico.

Se puede observar que la complejidad de FNC-SAT cambia de un algoritmo determinista exponencial con complejidad  $2^N$  (donde  $N$  es el número de variables) a un algoritmo no determinista polinómico con complejidad  $N$ , donde  $N$  es el número de elementos conjuntivos.

Cualquier lenguaje que pueda ser aceptado por una MTND también es aceptado por una MT, por lo que el poder de la MTND no cambia, solo lo hace su complejidad.

---

---

## ANEXO IV. CLASES DE COMPLEJIDAD COMPUTACIONAL

Una clase de complejidad es un conjunto de problemas para los que se observa una determinada propiedad relacionada con su complejidad. El objetivo es ahora agrupar los problemas de decisión en clases, atendiendo a su dificultad, es decir, a la complejidad computacional de mejor algoritmo capaz de solucionarlo. Se debe tener en cuenta que la complejidad no cambia entre los mismos modelos de máquinas (MT o MTND).

- La clase de complejidad P (Polinómico): Los problemas de decisión que pueden ser resueltos con un algoritmo en tiempo polinómico calculado a partir de la entrada de una cadena por una MT determinista.
- La clase de complejidad NP (Polinómico No Determinista): Es el conjunto de problemas que pueden ser resueltos en tiempo polinómico por una MT no determinista. La importancia de esta clase de problemas de decisión es que contiene muchos problemas de búsqueda y de optimización para los que se desea saber si existe una cierta solución o si existe una mejor solución que las conocidas. Dada su importancia, se han hecho muchos esfuerzos para encontrar algoritmos que decidan algún problema de NP en tiempo polinómico. Sin embargo, parecerá que para algunos problemas de NP (los del conjunto NP-completo) no es posible encontrar un algoritmo mejor, que simplemente realizar una búsqueda exhaustiva. Se puede decir que la mayoría de los problemas interesantes de computación en el mundo real pertenecen a la clase NP.
- La clase de complejidad NP-COMPLETOS: En la teoría de la complejidad computacional, la clase de complejidad NP-completo es el subconjunto de los problemas de decisión en NP, de manera que todo problema en NP se puede reducir en cada uno de los problemas de NP-completo. Se puede decir que los problemas de NP-completo son los problemas más difíciles de NP y muy probablemente no formen parte de la clase de complejidad P. La razón es que de tener una solución polinómica para un problema de NP-completo, todos los problemas de NP tendrían también una solución en tiempo polinómico.

El problema SAT es un problema que ha demostrado ser NP-completo, de manera, que si se encontrara un algoritmo polinómico que resolviera este problema, muchos otros problemas NP serían resueltos también en tiempo polinómico.

Otros ejemplos de problemas NP-completo son la suma de conjuntos, el problema de mochila, el camino Hamiltoniano, el máximo claque, etc.

---

---

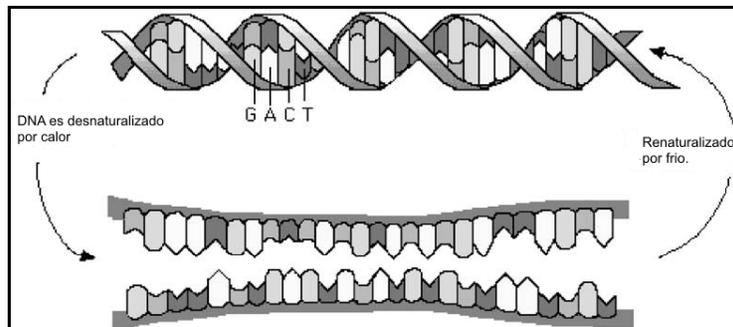
## ANEXO V. TÉCNICAS MOLECULARES Y OPERACIONES PARA MANIPULAR EL ADN

Las técnicas y operaciones moleculares permiten la manipulación del material genético de los organismos vivos o de fragmentos sintéticos de ADN. Todos los modelos de cómputo con ADN aplican secuencias específicas de operaciones biológicas a un conjunto de moléculas de ADN de cadena sencilla o doble [86]. Algunas de estas herramientas y operaciones son explicadas a continuación:

- Síntesis de ADN: Es la creación de un oligonucleótido, que es una secuencia específica y corta de ADN o ARN, con cincuenta o menos pares de bases.
- Separación y fusión de cadenas de ADN; que consiste en las técnicas siguientes:

La desnaturalización es separar las cadenas de una molécula de ADN de doble cadena para obtener dos cadenas sencillas [Figura 6].

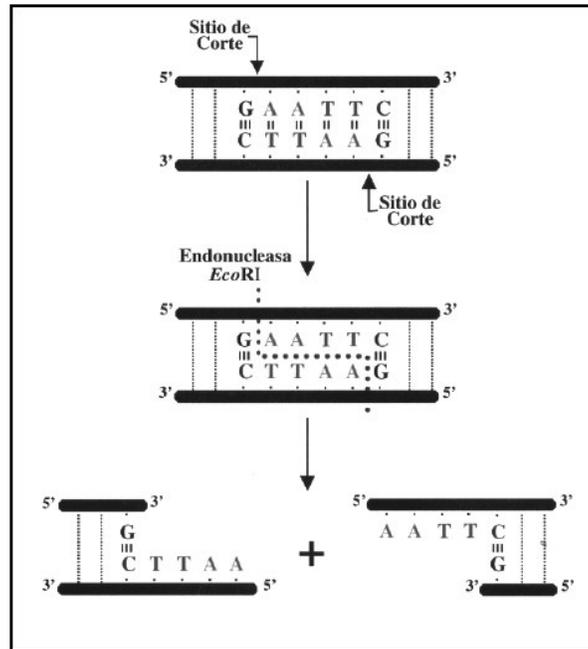
La renaturalización es la unión de una molécula de ADN de cadena sencilla con su correspondiente molécula de ADN de cadena sencilla complementaria.



**Figura 6.** Desnaturalizando y renaturalizando ADN [81].

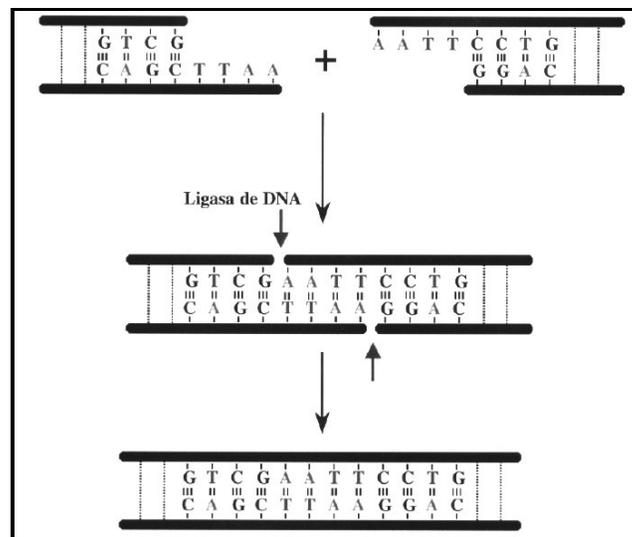
Hibridación: es la unión de una molécula de ADN de cadena sencilla con otra molécula de ADN o ARN la cual varía en tamaño y en donde no todos los nucleótidos son complementarios (por lo tanto son menos estables termodinámicamente).

- Cortar ADN: En este proceso se corta el ADN a través de enzimas de restricción. [Figura 7].



**Figura 7.** Digestión del ADN por la endonucleasa EcoRI [87].

- **Ligar ADN:** Las moléculas de ADN pueden ser unidas por sus extremos a través de un proceso llamado ligación, el cual es realizado por enzimas llamadas ligasas [Figura 8].



**Figura 8.** Mecanismo de acción de la enzima ligasa de ADN [87].

- **Modificar nucleótidos de ADN:** Se utilizan enzimas que modifican las moléculas de ADN adicionando o eliminando ciertos componentes químicos que son muy útiles para controlar varias operaciones en el ADN.

- 
- 
- La técnica de reacción en cadena de la polimerasa: llamada también PCR, por sus siglas en inglés, permite obtener muchas copias de un segmento de ADN específico. La esencia de la técnica es relativamente simple y consiste de los siguientes pasos:
    1. Se calienta el ADN hasta que las dos cadenas de la molécula se separen, dando como resultado 2 hebras separadas.
    2. Se adicionan nucleótidos, ADN polimerasa y secuencias promotoras de ADN que dan las señales de inicio de la replicación del ADN.
    3. Tan pronto como la mezcla comienza a enfriarse los segmentos promotores de ADN se unirán a las hebras separadas y el ADN polimerasa unirá los nucleótidos libres con los nucleótidos de los segmentos de ADN de la molécula inicial. En esta fase se logra la duplicación del ADN.
    4. Se repite el procedimiento anterior.

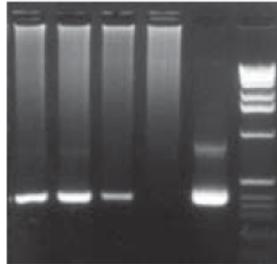
Al cabo de 10 ciclos se obtienen 1000 copias de ADN y en 20 ciclos se tendrá un millón de copias. En muchos casos de medicina forense, por ejemplo, se requieren grandes cantidades de ADN por lo que la técnica es bastante útil.

- La electrofóresis: es una técnica que permite comparar el tamaño y la secuencia de bases de varios segmentos de ADN. El ADN es una molécula que tiene una carga negativa (dada por los grupos fosfatos), y por lo tanto es atraída hacia cargas positivas. En la electrofóresis se utiliza un gel de agarosa, a través del cual el ADN puede pasar [Figura 9]. La técnica se puede resumir de la siguiente manera:
  1. La muestra de ADN de varias células del mismo organismo o de diferentes oligonucleótidos se mezcla con una determinada enzima de restricción, lo que producirá varias copias del segmento 1, varias copias del segmento 2, etc.
  2. Se coloca el gel de agarosa dentro de un recipiente que a su vez está dentro de una solución que conduce electricidad.
  3. Luego se coloca el ADN, ya sea en segmentos, en el extremo del gel opuesto al extremo que tiene la carga positiva.
  4. Al dejar pasar el ADN a través del gel los segmentos más cortos migrarán hacia el polo opuesto a una velocidad mayor que los segmentos más largos.
  5. El tamaño de los fragmentos se puede conocer comparando la distancia a la que migraron con la migración de fragmentos de referencia de longitud ya conocida.

---

---

Esta técnica nos permitirá conocer el número y el tamaño de los segmentos de determinado ADN y con ciertas operaciones la secuencia de cadenas de ADN.



**Figura 9. Gel de Agarosa.**

- **Secuenciación:** Esta operación permite determinar el orden preciso de bases nucleótidas (secuencia) de un fragmento de ADN. La mayoría de los tipos de secuenciación de ADN se basan en una técnica denominada extensión de oligonucleótido.