



# INSTITUTO TECNOLÓGICO DE IZTAPALAPA

Ingeniería: Sistemas Computacionales

Interaction nets (concurrente)

Presenta

Wakanda Forever

José Eliud Herrera Godinez 181080134 Particip 25%

<https://github.com/EliudH27/LenguajesAutomatas>

Margarita Micaela Hernández Núñez 181080028 Particip 25%

<https://github.com/MargaritaHdz/Lenguajesyautomatas.git>

Miriam Cruz Badillo 181080001 Particip 25%

<https://github.com/MiriamCruz181080001/LENGUAJES-Y-AUT-MATAS-ISC-6AM>

Brandon Martínez Hernández 181080328 Particip 25%

<https://github.com/DERKS20/PROYECTO-AUTOMATAS>

Profesor: Abiel Tomas Parra Hernández

CIUDAD DE MEXICO

Junio/2021



# Índice

Índice.....	2
Resumen.....	3
Introducción.....	4
OBJETIVOS .....	5
Objetivo general.....	5
Objetivos específicos:.....	5
Justificación.....	6
Marco teórico.....	7
Teoría de la Computación.....	7
Teoría de Computabilidad.....	8
Modelo de Computación .....	9
Redes de interacción .....	10
Temas relevantes .....	11
Metodología de trabajo.....	12
Design Thinking. ....	12
Desarrollo e Implementación.....	13
Problema .....	13
Solución .....	13
Conclusiones.....	15
FUENTES.....	16
Anexos (Información adicional relevante) .....	18
SOFTWARE PARA LA INTERACCION DE REDES .....	18
NetFlow Traffic Analyzer.....	18
Wireshark.....	18
Zenmap.....	18

## Resumen

Las redes de interacción son un modelo gráfico de cálculo ideado por Yves Lafont en 1990 como una generalización de las estructuras de prueba de la lógica lineal. Un sistema de red de interacción se especifica mediante un conjunto de tipos de agentes y un conjunto de reglas de interacción.

Los humanos interactuamos con los ordenadores de muchas maneras; la interfaz entre humanos y ordenadores es crucial para facilitar esta interacción. Las aplicaciones, los buscadores de internet y los ordenadores portátiles hacen uso de las prevalentes interfaces gráficas de usuario actuales.<sup>4</sup> Las interfaces mediante voz del usuario se utilizan para el reconocimiento de voz y sintetizan sistemas. Todo esto permite a los humanos comprometerse con caracteres personificados de una forma que nos podría ser conseguida con otros paradigmas de interfaz. El crecimiento del campo de la interacción persona-computadora se ha traducido en mayor calidad de interacción, y en una rama distinta de su historia. En lugar de diseñar interfaces regulares, las distintas ramas de investigación se han enfocado en los conceptos de multimodalidad antes que unimodalidad, interfaces inteligentes adaptativas antes que interfaces basadas en orden/acción e interfaces activas antes que pasivas.

Las interfaces de usuario pueden llevar a muchos problemas inesperados. Un ejemplo clásico es la accidente de Three Mile Island, un accidente de fusión nuclear, en el que las investigaciones concluyeron que el diseño de esta interfaz fue, al menos en parte, responsable del desastre.<sup>78</sup> De igual manera, muchos accidentes de aviación han sido resultado de las decisiones de los fabricantes de utilizar instrumentos de vuelo no estandarizados: a pesar de que los nuevos diseños se propusieron como superiores en interacción humano-máquina básica, los pilotos ya habían interiorizado el plano “estándar” y, así, la idea conceptualmente correcta tuvo resultados indeseables.

## Introducción

El modelado computacional es el uso de computadoras para simular y estudiar sistemas complejos utilizando las matemáticas, la física y la informática. Un modelo computacional contiene numerosas variables que caracterizan el sistema bajo estudio. La simulación se realiza ajustando las variables, solas o combinadas, y observando los resultados. El modelado computacional permite a los científicos realizar miles de experimentos simulados por computadora. Los miles de experimentos por computadora identifican los pocos experimentos de laboratorio que tienen más probabilidades de resolver el problema bajo estudio.

Los modelos computacionales de hoy en día pueden estudiar un sistema biológico en múltiples niveles. El estudio de sistemas en múltiples niveles se conoce como modelado multiescalar (MSM por sus siglas en inglés).

Los modelos computacionales han jugado un rol importante en el descubrimiento y entendimiento de las complejidades psicológicas de la comprensión textual. Esto ha sido por tres razones principales. En primer lugar, el proceso de transformar teorías de comprensión de textos, descritas verbalmente (teorías conceptuales), en modelos computacionales de comprensión promueve el desarrollo y evolución de las teorías conceptuales, mostrando dónde los modelos concuerdan con información empírica y dónde no lo hacen. En segundo lugar, los modelos computacionales pueden ser aplicados a información de comportamientos con el fin de comprender mejor y evaluar constructos explicativos alternativos, especialmente en casos donde los patrones de comportamiento no resultan como se esperaba a priori. En tales casos, los investigadores aportan explicaciones, muchas de las cuales resultan bastante razonables. Los modelos computacionales pueden así proveer un modo de evaluar tales explicaciones. Por último, y parcialmente, como resultado de los primeros dos beneficios, los modelos computacionales promueven la comunicación entre investigadores dentro y a través de diversas áreas de investigación.

La lógica lineal es una lógica subestructural propuesta por Jean-Yves Girard como un refinamiento de la lógica clásica e intuicionista, uniendo las dualidades de la primera con muchas de las propiedades constructivas de la segunda. Aunque la lógica también se ha estudiado por sí misma, en términos más generales, las ideas de la lógica lineal han influido en campos como los lenguajes de programación, la semántica de juegos y la física cuántica (porque la lógica lineal puede verse como la lógica de la lógica cuántica). teoría de la información), así como lingüística, particularmente debido a su énfasis en la limitación de recursos, la dualidad y la interacción.



## OBJETIVOS

### *Objetivo general.*

La interacción persona-computadora estudia la forma en que los seres humanos hacen o no uso de artefactos, sistemas e infraestructuras computacionales. Debido a esto, gran parte de la investigación en este campo busca mejorar la relación humano-computadora mejorando la usabilidad de las interfaces de los ordenadores

### *Objetivos específicos:*

- Δ Comprender la forma en la que los lenguajes de programación se pueden asociar con la teoría matemática para su mejor entendimiento.
- Δ Comprender la gran cantidad de aplicaciones que pueden tener estos modelos matemáticos, sobre todo en las áreas de la computación y la electrónica.
- Δ Conocer el poder computacional de estas máquinas en el contexto de la solución de problemas de reconocimiento de lenguajes.
- Δ Entender que el diseño de algoritmos presenta limitaciones en ciertos casos, que impiden su representación adecuada.

## Justificación

Los modelos computacionales pueden constituir una forma de probar y aplicar tales explicaciones. Debido a que los modelos computacionales hacen predicciones específicas y, en ocasiones, no obvias, podemos probar y contrastar modelos alternativos y los resultados pueden ayudar a diferenciar entre teorías conceptuales mutuamente competidoras. Finalmente, y en parte como resultado de los dos beneficios anteriores, los modelos computacionales promueven la comunicación entre quienes investigan en la misma área o en diferentes áreas. Estos modelos promueven la consolidación e integración entre teorías y datos empíricos acerca de la comprensión de textos, iluminan aquellas áreas que requieren mayor desarrollo teórico y se integran con otras áreas de investigación, mostrando dónde ciertos mecanismos importantes para la comprensión de textos pueden también ser importantes para comprender otros fenómenos.

Las actuales teorías acerca de la comprensión textual admiten esta complejidad en supuestos acerca de las complejas interrelaciones entre los varios niveles y sistemas del lenguaje

Los avances en la teoría de la comprensión de textos se han logrado gracias a los esfuerzos por traducir las formulaciones teóricas que postulaban estas variables a modelos computacionales manejables. Al mostrar de qué se podía dar cuenta y de qué no se podía, los esfuerzos computacionales han estimulado la evolución de teorías de comprensión textual.

Todos estos esfuerzos de modelamiento computacional han ayudado a definir un tema importante aún no resuelto en la comprensión de textos o han presentado evidencia convincente de la utilidad del modelo computacional particular que estaba siendo probado. Al hacerlo, estos esfuerzos de modelamiento computacional han estimulado el desarrollo de teorías de comprensión de textos. En el presente contexto destacamos dos esfuerzos de modelamiento en la familia del CI

Desarrollamos esta posición, describiendo primero el trabajo de Walter Kitsch y sus colegas, debido a que ellos han elaborado una teoría de comprensión muy influyente en la que el modelamiento computacional ha jugado un papel importante.

## Marco teórico

### *Teoría de la Computación*

Se ocupa de determinar qué problemas pueden ser resueltos computacionalmente y con qué eficiencia. La teoría considera distintos modelos de cómputo, como los autómatas finitos (que son los más sencillos), las máquinas de Turing (que son las computadoras usuales de hoy en día) y las computadoras cuánticas (cuyo funcionamiento no es digital). Las lógicas y los lenguajes formales juegan un rol central en la teoría de la computación porque permiten expresar propiedades de los programas y razonar sobre su comportamiento. La teoría de la computación también se encarga de entender el límite entre los problemas computables y los no-computables y, dentro del mundo de lo computable, clasificarlos de acuerdo a su grado de simpleza o dificultad.

En particular, estudiamos lógicas con buen comportamiento computacional, como las lógicas modales, tanto desde el punto de vista de la teoría de modelos como desde el de la teoría de prueba. Analizamos lenguajes eficientes de consultas que permiten razonar sobre distintas estructuras de representación del conocimiento. A un nivel más abstracto, investigamos las propiedades teóricas de los sistemas de reescritura y los modelos de cómputo fuertes, como el cálculo lambda. A la inversa, estudiamos modelos de cómputo débiles, como los autómatas finitos y sus numerosas variantes. Nos ocupamos también de la noción de aleatoriedad en relación a los distintos modelos de cómputo y a los grados de dificultad de los problemas. Por último, introducimos nociones provenientes de la teoría de funciones computables y de la teoría de la aleatoriedad en el procesamiento cuántico de la información y en algunos modelos computacionales que intentan explicar ciertas características de la cognición humana.

Es una rama de la matemática y la computación que centra su interés en las limitaciones y capacidades fundamentales de las computadoras. Específicamente esta teoría busca modelos matemáticos que formalizan el concepto de hacer un cómputo (cuenta o cálculo) y la clasificación de problemas

## *Teoría de Computabilidad*

es una de las cuatro partes que constituyen la lógica matemática, siendo las otras tres, la teoría de conjuntos, la teoría de modelos y la teoría de la demostración, y se ocupa del estudio y clasificación de las relaciones y aplicaciones computables. Además, la teoría de la computabilidad, junto con la teoría de autómatas, lenguajes y máquinas, es el fundamento de la informática teórica y esta, a su vez, de la industria de los ordenadores.

Desde tiempo inmemorial se sabe que cierta clase de problemas, e.g., la determinación del máximo común divisor de dos números enteros, mediante el algoritmo de Euclides, o la determinación de los números primos, mediante la criba de Eratóstenes, son algorítmicamente solubles, i.e., hay algoritmos o procedimientos mecánicos que permiten obtener la solución del problema en cuestión. De manera que hasta principios del siglo XX se daba por hecho que existían algoritmos y que el único problema residía en determinarlos. Así pues, si lo que se desea es determinar un algoritmo, no hay ninguna necesidad de definir la clase de todos los algoritmos; eso sólo es necesario si se pretende demostrar que algún problema no es algorítmicamente soluble, i.e., que para dicho problema no hay ningún algoritmo que lo resuelva.

Es posible que el primero en afirmar la no existencia de un algoritmo fuera Tietze en 1908, quién dijo de los grupos de presentación finita

Pero parece ser que fue, por una parte, el problema de la decidibilidad de la lógica de predicados planteado por Hilbert y Ackermann en su libro sobre lógica, publicado en 1928, y, por otra, el asunto de la solubilidad de todo problema matemático, lo que indujo, en aras a resolverlos, a diversos investigadores a partir de 1930, y entre los que cabe mencionar a Gödel, Church y Turing, a proponer diversas formalizaciones del concepto informal de función mecánicamente computable. Debido a que de todas esas formalizaciones, y de otras propuestas por Kleene, Post y Markoff, se demostró que eran dos a dos equivalentes, se propuso la hipótesis, conocida como Hipótesis de Church-Turing-Post-Kleene, que afirma la coincidencia entre el concepto informal de función parcial mecánica o algorítmicamente computable, y el concepto formal, matemático, de aplicación parcial recursiva. Naturalmente, esa hipótesis, de carácter similar a otras hipótesis propuestas en las ciencias empíricas, no es demostrable, y su fundamento último reside en las equivalencias antes mencionadas.



## *Modelo de Computación*

El modelado computacional es el uso de computadoras para simular y estudiar sistemas complejos utilizando las matemáticas, la física y la informática. Un modelo computacional contiene numerosas variables que caracterizan el sistema bajo estudio. La simulación se realiza ajustando las variables, solas o combinadas, y observando los resultados. El modelado computacional permite a los científicos realizar miles de experimentos simulados por computadora. Los miles de experimentos por computadora identifican los pocos experimentos de laboratorio que tienen más probabilidades de resolver el problema bajo estudio.

es un modelo matemático en las ciencias de la computación que requiere extensos recursos computacionales para estudiar el comportamiento de un sistema complejo por medio de la simulación por computadora. El sistema bajo estudio es a menudo un sistema complejo no lineal para el cual las soluciones analíticas simples e intuitivas no están fácilmente disponibles. En lugar de derivar una solución analítica matemática para el problema, la experimentación es hecha con el modelo cambiando los parámetros del sistema en la computadora, y se estudian las diferencias en el resultado de los experimentos. Las teorías de la operación del modelo se pueden derivar/deducir de estos experimentos de computacionales

Los modelos computacionales de hoy en día pueden estudiar un sistema biológico en múltiples niveles. Los modelos de cómo se desarrolla la enfermedad incluyen procesos moleculares, interacciones intercelulares, y cómo dichos cambios afectan los tejidos y los órganos. El estudio de sistemas en múltiples niveles se conoce como modelado multiescalar (MSM por sus siglas en inglés)

En los últimos años, otros modelos de computación han alcanzado notoriedad y éxito, como el aprendizaje automático o las redes neuronales, mientras ahora son otros, como la computación cuántica, los que comienzan a mostrar avances significativos. Cada uno tiene sus desafíos y su marco de aplicación.

- Δ Computación algorítmica
- Δ Aprendizaje automático e IA
- Δ Computación cuántica

## *Redes de interacción*

Son un modelo gráfico de cálculo ideado por Yves Lafont en 1990 como una generalización de las estructuras de prueba de la lógica lineal. Un sistema de red de interacción se especifica mediante un conjunto de tipos de agentes y un conjunto de reglas de interacción. Las redes de interacción son un modelo de cálculo intrínsecamente distribuido en el sentido de que los cálculos pueden tener lugar simultáneamente en muchas partes de una red de interacción y no se necesita sincronización. Esto último está garantizado por la fuerte propiedad de reducción de confluencia en este modelo de cálculo.

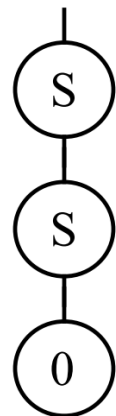
Interaction Nets es un lenguaje de programación basado en la reducción de gráficos. Programación consiste en especificar las formas en que se pueden construir los gráficos y definir reescritura de reglas. Los gráficos se conocen como redes; una red es un gráfico no dirigido con vértices etiquetados, que satisfacen ciertas otras condiciones. En las redes de interacción puras idioma, no hay definiciones integradas, por lo que, por ejemplo, los números deben definirse mediante el programador. Una forma de hacer esto es usar una representación unaria y considerar redes

Se deben dar definiciones de símbolo adecuadas para S y 0 para permitir que se conecte de esta manera. Observe que se permite que una red tenga "bordes colgantes" que solo están conectados en un extremo. Tales aristas pueden considerarse como variables libres de la red. y suelen estar etiquetados. De hecho, también es posible tener bordes que estén conectados en ni fin; tales aristas son necesarias para describir algunas reglas de reescritura.

Cada agente (vértice) en una red es una instancia de algún símbolo. Una definición de símbolo asocia el nombre de un símbolo con una especificación de cómo un agente con ese nombre se puede conectar con otros agentes. Un agente tiene varios puertos a los que los bordes pueden Adjuntado.

Los puertos se escriben, una especificación de tipo que consta de un nombre de tipo y una dirección. Las direcciones son de entrada, representadas por -, y de salida, representadas por +.

Solo se pueden conectar dos puertos si tienen el mismo tipo y direcciones opuestas.



## *Temas relevantes*

### Δ Funciones de idioma adicionales

Los pseudo-agentes son el equivalente de las redes de interacción de las macros en la programación tradicional. Un pseudo-agente tiene una definición de símbolo como para un agente, y una expansión regla que es como una regla de interacción pero con solo el pseudo-agente en la mano izquierda lado. La regla de expansión especifica una red por la cual el pseudo-agente es reemplazado cuando interactúa con otro agente. Un ejemplo en el que los pseudo-agentes son útiles es definir la función Invertir para listas.

### Δ El sistema

El sistema proporciona un entorno para experimentar con Interaction Nets programación. Permite que los programas se escriban y ejecuten, es decir, acepta tipos, definiciones de símbolos y reglas, permite ingresar redes y reducirá una red dada a forma normal aplicando las reglas de interacción dadas. El sistema comprueba la escritura, linealidad y partición a medida que procesa las definiciones. También permite reducir las redes paso a paso y examinado en cada etapa, como alternativa a la realización del conjunto secuencia de reducciones sin parar. Otras facilidades permiten escribir programas. Una descripción completa de la interfaz de usuario aparece en un Apéndice.

### Δ Programación en redes de interacción

En los programas de ejemplo que siguen, los tipos de puerto de los agentes no siempre son especificados antes de que se utilicen en las reglas de interacción; Por lo general, se pueden deducir de la reglas. La parte importante de un programa es la colección de reglas, que siempre son mostrado. Los primeros tres ejemplos son para la versión simplemente mecanografiada de Interaction Nets; el final es polimórfico.

### Δ Direcciones futuras

Dado que las reglas de interacción son locales, existe la posibilidad de una reducción paralela de enlaces activos, sin interferencias. En ausencia de efectos secundarios, el orden de reducción no es importante; cualquier agente que cause efectos secundarios deberá diseñarse cuidadosamente. A

La implementación paralela explotaría automáticamente las posibilidades de paralelismo en algoritmos

## Metodología de trabajo

### *Desing Thinking.*

Esta metodología se centra en realizar un diseño personalizado para cada proyecto o cliente, teniendo la mentalidad de que cada situación es diferente, se busca empatizar y descubrir las necesidades para adaptarlas lo más próximo y así obtener resultados más adecuados.

El proyecto debe desarrollarse siempre mediante cinco fases no lineales:

- △ Empatía: Se busca entender la situación para llegar a lograr una profunda comprensión, que fomente la creación de ideas adecuadas para el proyecto que se va a ejecutar.
- △ Definición: En esta fase hay que analizar todas las ideas de la anterior fase y escoger las ideas que aportan mayor valor o cuyas soluciones serán determinantes para el buen desarrollo del proyecto.
- △ Ideación: Aquí las ideas toman protagonismo, ya que con las conclusiones obtenidas en las anteriores fases se pueden observar todas las posibles vías de desarrollo, buscando así soluciones para ver cual es la mejor opción.
- △ Construcción: Convertir las ideas en realidad es el objetivo de esta fase, donde este proceso ayude a refinar el resultado final y vislumbrar las posibles soluciones.
- △ Testeo: Consiste en probar todas las ideas desarrolladas en las fases anteriores. El objetivo es obtener los resultados adecuados por parte del cliente y los miembros del equipo.

## Desarrollo e Implementación

### *Problema*

Diseñar un algoritmo que calcula el número consecutivo de un número dado en binario.

### *Solución*

Si el número es par, su último bit es 0. El algoritmo sólo tiene que cambiar este 0 por un 1.

Si el número es impar, su último bit es 1. En este caso, se tiene que cambiar por 0's todos los 1's seguidos que haya escritos de derecha a izquierda hasta llegar al primer 0, que se cambia por un 1. Si no hay ningún 0, entonces se tiene que añadir un 1 delante del número (añadir un bit). Es decir, escribir un 1 en la casilla en blanco (B) a la izquierda del número.

Vamos a considerar tres estados:  $q_0, q_1, q_2$

- Inicialmente, la MT está en el estado  $q_0$  con la cabeza señalando la primera cifra del número.

La MT recorre todo el número para ver si es par o impar sin modificar su cinta.

$$\delta(q_0, 0) = (q_0, 0, R) \quad \delta(q_0, 0) = (q_0, 0, R)$$

$$\delta(q_0, 1) = (q_0, 1, R) \quad \delta(q_0, 1) = (q_0, 1, R)$$

- Notemos que, por ahora, la MT se detiene al llegar al primer símbolo en blanco a la derecha del número.

La MT vuelve a la anterior casilla (último número). Si es un 0, lo cambia por un 1 y pasa al estado final que es  $q_2$ . Para hacer esto usaremos el estado  $q_1$ :

$$\delta(q_0, B) = (q_1, B, L) \quad \delta(q_0, B) = (q_1, B, L)$$

$$\delta(q_1, 0) = (q_2, 1, R) \quad \delta(q_1, 0) = (q_2, 1, R)$$

- Si el número es impar, la MT no ha cambiado el último número, pero está en el estado  $q_1$ . Tiene que cambiar todos los 1's consecutivos que haya de derecha a izquierda.

$$\delta(q_1, 1) = (q_1, 0, L) \quad \delta(q_1, 1) = (q_1, 0, L)$$

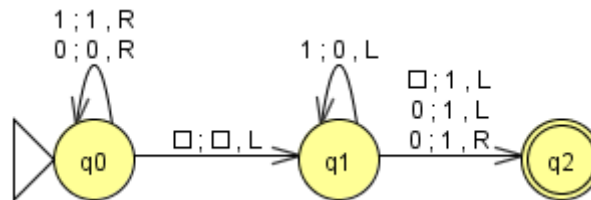
- Por ahora, la MT se para cuando llega al primer 0 (de derecha a izquierda) ó en un símbolo en blanco. Si es un 0, lo cambia por un 1 y el proceso finaliza:

$$\delta(q_1, 0) = (q_2, 1, L) \quad \delta(q_1, \square) = (q_2, 1, L)$$

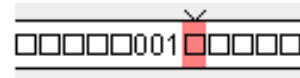
(Hemos escrito un desplazamiento a la izquierda, pero esto no tiene importancia ya que la MT ha llegado al estado final).

- Si lo que señala la cabeza es un blanco en vez de un 1, tiene que cambiarlo por un 1 y finalizar el proceso.

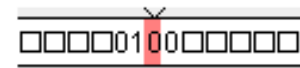
$$\delta(q_1, B) = (q_2, 1, L)$$



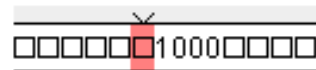
**Entrada: 000; Resultado esperado: 001.**



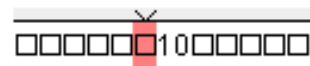
**Entrada: 0011; Resultado esperado: 0100.**



**Entrada: 111; Resultado esperado: 1000.**



**Entrada: 1; Resultado esperado: 10.**



## Conclusiones

Sobre el presente trabajo se llegó a distintas conclusiones las cuales se mencionan a continuación.

Las redes cumplen un papel muy importante en la vida actual, ya que, el uso de las tecnologías se ha hecho más común y de mayor utilidad, y es que facilitan la comunicación de forma global con nuestras familias, amigos, etc. Esto se logra por los medios de los diferentes procedimientos que utilizan estas redes, haciendo que la comunicación llegue al destino y a tiempo.

Las redes funcionan con protocolos y estándares, esto es para que sea fácil su implementación de acuerdo a las necesidades de cada individuo.

Con la implementación de la tecnología tenemos la facilidad de organizar nuestro día a día, ya sea tanto laboral, profesional o personal. Gracias a que las tecnologías nos proporcionan herramientas para la realización de las distintas actividades que requerimos.

Al utilizar redes se hace posible el que podamos tener una comunicación constante a costos razonables.

Como ya se mencionó, existen diferentes herramientas que se pueden usar para la gestión de redes. Por ejemplo, las que mencionamos en el presente documento son algunas, estas son fáciles de usar ya que su interfaz es práctica. Así mismo, estas herramientas fueron diseñadas para, aquellos administradores en redes, tengan un control del funcionamiento de las redes en equipos ya sea de empresas o personales, cabe mencionar que estas herramientas pueden ser utilizadas por el público en general que disponga de una computadora, esto para verificar que no halla fallas en sus ordenadores, incluso para los estudiantes de esta área, resulta ser una herramienta muy útil para realizar prácticas y ir aprendiendo a realizar este proceso.

Dentro de las herramientas que se investigaron, la más recomendada es Wireshark, ya que, es más fácil y eficaz de usar, permitiendo tener una mejor gestión de la seguridad de la red, así mismo esta funciona mediante protocolos.

## FUENTES

- ▽ *CC209 TEORIA DE LA COMPUTACIÓN*. (s. f.). CC209 TEORIA DE LA COMPUTACIÓN. Recuperado 25 de mayo de 2021, de [http://mate.cucei.udg.mx/depto/Materias/CC209\\_Teoria\\_de\\_la\\_Computacion.html](http://mate.cucei.udg.mx/depto/Materias/CC209_Teoria_de_la_Computacion.html)
- ▽ *Programación concurrente*. (s. f.). Programación Concurrente. Recuperado 26 de mayo de 2021, de [http://ferestrepoca.github.io/paradigmas-de-programacion/progconcurrente/concurrente\\_teoría/index.html](http://ferestrepoca.github.io/paradigmas-de-programacion/progconcurrente/concurrente_teoría/index.html)
- ▽ (Wikipedia contributors, 2019)  
Wikipedia contributors. (2019, December 13). Interaction nets. Retrieved June 8, 2021, from Wikipedia, The Free Encyclopedia website:  
[https://en.wikipedia.org/w/index.php?title=Interaction\\_nets&oldid=930588877](https://en.wikipedia.org/w/index.php?title=Interaction_nets&oldid=930588877)
- ▽ (Wikipedia contributors, 2021)  
Wikipedia contributors. (2021, April 6). Linear logic. Retrieved June 8, 2021, from Wikipedia, The Free Encyclopedia website:  
[https://en.wikipedia.org/w/index.php?title=Linear\\_logic&oldid=1016400872](https://en.wikipedia.org/w/index.php?title=Linear_logic&oldid=1016400872)
- ▽ (Wikipedia contributors, 2021b)  
Wikipedia contributors. (2021b, April 20). Model of computation. Retrieved June 8, 2021, from Wikipedia, The Free Encyclopedia website:  
[https://en.wikipedia.org/w/index.php?title=Model\\_of\\_computation&oldid=1018905744](https://en.wikipedia.org/w/index.php?title=Model_of_computation&oldid=1018905744)
- ▽ (Goldman, Golden, & van den Broek, 2007)  
Goldman, S., Golden, R., & van den Broek, P. (2007). ¿Por qué son útiles los modelos computacionales de comprensión de textos? *Revista Signos (Impresa)*, 40(65), 545–572.





▽ (karoma, n.d.)

karoma. (n.d.). Teoría de la Computación. Retrieved June 21, 2021, from Uba.ar website: <https://icc.fcen.uba.ar/teoria-de-la-computacion/>

▽ (“TEORIA DE LA COMPUTACION - CIENCIAS DE LA COMPUTACION,” n.d.)

TEORIA DE LA COMPUTACION - CIENCIAS DE LA COMPUTACION. (n.d.).

Retrieved June 21, 2021, from Google.com website:

<https://sites.google.com/site/cienciasdelacompuacion/algoritmos-y-estructuras-de-datos/teoria-de-la-computacion>

▽ (Computing, 2020)

Computing, R. (2020, May 8). ¿Cuáles son los nuevos modelos de computación?

Retrieved June 22, 2021, from Computing.es website:

<https://www.computing.es/analytics/opinion/1118527046201/cuales-nuevos-modelos-de-computacion.1.html>

▽ (n.d.)

(N.d.). Retrieved June 22, 2021, from Gla.ac.uk website:

<http://www.dcs.gla.ac.uk/~simon/publications/diploma.pdf>

▽

## Anexos (Información adicional relevante)

### *SOFTWARE PARA LA INTERACCION DE REDES*

Los administradores de redes necesitan contar con programas útiles y eficientes para desarrollar su actividad laboral en perfectas condiciones los administradores de redes informáticos son esos profesionales que desarrollan, soportan y mantienen las redes informáticas que conforman una empresa. De hecho, muchas empresas no trabajan con una única red, sino que tienen varias; por ejemplo, una empresa puede disponer de una red de área local (en una oficina) y una red de área extensa (que conecta todas las áreas de la empresa de todo un territorio).

Por tanto, dichos administradores necesitan contar con programas efectivos que les ayuden a desarrollar su trabajo de manera rápida e impecable

#### *NetFlow Traffic Analyzer*

Éste es un programa de administración de redes de SolarWinds que permite el monitoreo del ancho de banda de las redes, además de hacer un análisis del tráfico de red. Es perfecto para administradores de redes porque además del análisis también ofrece un panel de análisis de desempeño, y con él se pueden detectar flujos de tráfico malicioso o mal formado. Es, en definitiva, un completo programa que cualquier administrador ha de tener en cuenta

#### *Wireshark*

Este programa es uno de los más potentes en el análisis de protocolos de red, por lo que es un aliado perfecto para administradores que necesitan mantener la seguridad en redes. Éste es un programa muy interesante, ya que es multiplataforma, y cuenta tanto con captura en vivo como con análisis fuera de línea. Aquí todos los datos capturados pueden ser consultados a través de una interfaz gráfica, lo que lo convierte en un programa muy útil e intuitivo.

#### *Zenmap*

Zenmap es una interfaz gráfica oficial para la Nmap Security Scanner, y está disponible para diversos sistemas operativos, como Mac, BSD, Windows o Linux. Este programa ayuda a los administradores de redes a mejorar las características de Nmap, ya que permite hacer un inventario de red y el seguimiento de host. Cabe destacar que éste ofrece un escaneo rápido y que las capturas pueden leerse en vivo o ser almacenadas para leerse después.