



Universidad Tecnológica Metropolitana  
Facultad de Ingeniería  
Departamento de Informática

# Plataforma Web de Servicios

---

“Montaje de clúster computacional para dar solución a grandes problemáticas de procesamiento de extensos volúmenes de datos: Percolación, Código de la Biblia y FASTA”



Grupo Líder Primer Semestre 2015

Computación Paralela

Oscar Magna Veloso

Miércoles, 24 de junio de 2015

## Resumen

Tras una extensa investigación tanto en bibliografía general, experiencias anteriores y contenidos en la WEB o en bases de conocimiento científico, se llegó a conocer que existen actualmente problemáticas que comprenden inmensos volúmenes de datos, en función de la investigación se plantea dar solución a tres grandes problemas de computo, por ello el proyecto desarrollado y explicitado en el presente informe se elaboró con la finalidad de crear una plataforma de servicios web ambientada en un clúster computacional heterogéneo, para mostrar de manera genérica mediante una interfaz amigable e intuitiva las soluciones a problemas de simulación mediante algoritmo de percolación, búsqueda de patrones mediante algoritmo de código de la biblia y alineamiento de cadenas mediante procesamiento de secuencias de ADN en formato FASTA. Para encontrar las soluciones óptimas a los problemas a abarcar se utilizaron referencias a semestres anteriores obviamente mejorando todo aspecto funcional además de integrar todas las soluciones en solo una plataforma. Una vez modeladas y estructuradas las soluciones se integran en la plataforma web donde el flujo natural de información para la utilización de la plataforma de servicios es el ingreso de los requerimientos de entrada, se procesa la información en el clúster con la finalidad de disminuir el tiempo en el que los algoritmos se completan, por último se entrega una respuesta vía correo en formato PDF, al usuario que hace la petición. Básicamente nuestro proyecto logró hacer un sistema integrado que enlaza servicio web que procesa sus cálculos con un clúster computacional. Finalmente se obtiene una plataforma sustentada en un sistema integrado cuyas principales fortalezas son la capacidad de procesar los requerimientos de manera paralela, su experiencia de usuario y que puede resolver problemas de simulación, búsqueda de patrones y alineamiento de cadenas.

## Abstract

After an extensive research in general bibliography, previous experiences and content on the web or on a basis on general scientific knowledge, it became known that they are currently issues that include huge data volumes, on a basis on the research it was decided give solution to three big problems, and the following project was developed and explained in the current report created in order to present a platform of web services in a heterogeneous computational cluster, to show the solutions in a generic form and in a friendly environment the problems of simulation by percolation, pattern search by the bible code and alignment of chains through processing of DNA in FASTA format. To find the optimal solutions to these problems, the references to previous semester works were used and improved in a functional aspect and putting them together in one platform. Once the solutions were modeled and structured, they were integrated on the web platform where the natural flow on information for the use as the service was de input of the requirements, processing the information on the cluster in order to lower the time of the algorithm processing, and finally, sending the answer via email in a PDF format to the user. Basically our project managed to make an integrated system that links web service that processes calculations with a computer cluster. Finally a strategy based on an integrated platform whose main strengths are the ability to process the requests in parallel, your user experience and can solve problems simulation, pattern matching and alignment of strings is obtained.

## Índice de contenidos

1.	Introducción .....	1
2.	Planteamiento del problema .....	1
3.	Objetivos .....	2
3.1	Objetivo general .....	2
3.2	Objetivos específicos .....	2
4.	Alcances y limitaciones .....	2
5.	Marco teórico .....	3
5.1	Clúster .....	3
5.2	Nodos .....	3
5.3	Clúster Beowulf .....	3
5.4	Open MPI .....	3
5.5	Ruido .....	4
5.6	Ubuntu .....	4
5.7	Python .....	4
5.8	Bootstrap .....	4
5.9	PHP .....	5
5.10	LAMP .....	5
5.11	MailGun .....	5
5.12	Ley de Amdahl .....	5
5.13	SpeedUp .....	5
5.14	Eficiencia .....	6
5.15	Costo .....	7
6.	Modelo propuesto del funcionamiento del sistema .....	7
7.	Metodología de desarrollo .....	8
8.	Evaluación económica .....	10
8.1	Recursos físicos .....	10
8.2	Cuantificación de esfuerzo .....	10
9.	Diseño y arquitectura del software .....	11
9.1	Modelo lógico .....	11
9.2	Modelo físico .....	12
9.3	Herramientas utilizadas .....	12
9.4	Análisis y desempeño de optimización .....	14
9.4.1	Algoritmo de Torah .....	14

---

9.4.2	Algoritmo de Percolación .....	19
9.4.3	FASTA.....	25
9.5	Conclusiones obtenidas.....	28
9.5.1	Algoritmo Torah .....	28
9.5.2	Algoritmo de Percolación .....	28
9.5.3	Algoritmo FASTA.....	29
10.	Conclusiones y trabajo a futuro.....	30
11.	Anexos .....	31
	Anexo A: Manual de instalación .....	31
	<b>Nodo Maestro</b> .....	31
	#Paso 1 .....	31
	#Paso 2 .....	32
	#Paso 3 .....	32
	#Paso 4 .....	33
	#Paso 5 .....	33
	#Paso 6 .....	34
	#Paso 7 .....	34
	<b>Nodo Esclavo</b> .....	35
	#Paso 1 .....	35
	#Paso 2 .....	35
	#Paso 3 .....	36
	Anexo B: Conexión Remota (Túnel SSH) .....	43
	Anexo C: MailGun .....	45
	Anexo D: Modelo de la plataforma .....	46
	Anexo E: Carta Gantt .....	47
	Anexo F: Valor Ganado .....	48
	Interpretación Valor Ganado .....	48
	Anexo G: Evaluaciones.....	49
	Evaluaciones por Etapa .....	49
	Calificación Final del Proyecto.....	49
	Notas Primera Etapa .....	49
	Notas Segunda Etapa.....	50
	Notas Tercera Etapa .....	50
	Notas Cuarta Etapa.....	51
	Anexo H: Detalle de Esfuerzo .....	52

---

Anexo I: Alineamiento de Secuencias de ADN .....	56
1. Presentación general.....	56
2. Objetivos .....	57
Objetivos Generales.....	57
Objetivos Específico.....	57
3. Introducción .....	57
4. Marco Teórico .....	58
4.1 Formato FASTA .....	59
4.2 Nomenclatura de Ácidos.....	60
4.3 Definiciones Indispensables para la contextualización del problema .....	60
4.4 Matrices de sustitución .....	62
5. Estado del Arte.....	63
6. Definición del Problema .....	64
7. Modelo de la Solución Propuesta .....	65
8. Bibliografía .....	68
Anexo J: Búsqueda de patrones del Código de la Biblia .....	69
1. Presentación General .....	69
2. Objetivos .....	69
2.1 Objetivo General.....	69
2.2 Objetivos Específicos.....	69
3. Introducción .....	70
4. Marco Teórico .....	70
4.1. Geometría de SLE.....	72
4.2. Resumen del procedimiento. ....	72
5. Estado del Arte.....	72
5.1. Predicciones reales .....	75
5.2. Críticas.....	76
6. Descripción del modelo de solución .....	76
6.1. Workflow del proceso secuencial .....	77
6.1.1. Lectura de Parámetros .....	77
6.1.2. Conversión .....	77
6.1.3. Búsqueda de Patrones .....	79
6.1.4. Resultados .....	80
6.1.5. Interpretación de Resultados.....	80
6.2. Workflow del proceso paralelo .....	80

---

8.6. Conclusiones .....	81
9. Valor agregado - Algoritmo Implícito .....	82
9.1. Diccionarios ocupados.....	82
9.2. Rendimiento .....	83
9.2.1. Casos de uso .....	83
9.2.2. Tabla de métricas .....	83
9.2.3. Gráficos .....	84
9.3. Estadísticas .....	84
9.3.1. Casos de Uso .....	84
9.3.2. Análisis de PDF .....	85
9.3.3. Tablas de métricas .....	85
9.3.4. Gráficos .....	86
9.3.5. Conclusiones.....	87
Anexo K: Modulo Percolación .....	88
1. Resumen.....	88
2. Introducción .....	88
3. Objetivos .....	88
2.1 Objetivos Generales .....	88
2.2 Objetivos Específicos .....	88
4. Marco teórico.....	89
5. Estado del arte .....	90
5.1 Aplicaciones de la percolación .....	90
5.2 Percolación e incendios forestales .....	90
6. Descripción del modelo de solución .....	93
7. Workflow propagación de incendios forestales .....	93
7.1 Modelo secuencial .....	93
7.2 Modelo paralelo .....	95
8. Análisis de Desempeño y Optimización .....	96
8.1 Estadísticas .....	96
Secuencial .....	96
Paralelo con matriz de tamaño 100 .....	97
Paralelo con matriz de tamaño 1000.....	99
Nodo optimo .....	101
Pruebas clúster heterogéneo .....	102
9. Valor agregado .....	104

Percolación y propagación de enfermedades.....	104
Workflow propagación de enfermedades modelo paralelo.....	107
10. Análisis y conclusiones.....	108
Anexo L: Asana .....	110
Hitos Importantes.....	110
Gráficos de avance .....	113
Anexo M: Carta de amonestación .....	114

## Índice de ilustraciones

Ilustración 1 : Comunicación maestro esclavos. Elaboración propia. ....	4
Ilustración 2: Fórmula para cálculo de SpeedUp. Fuente: Elaboración propia. ....	6
Ilustración 3: Fórmula para cálculo de SpeedUp. Fuente: Elaboración propia. ....	6
Ilustración 4: Fórmula para cálculo de Eficiencia. Fuente: Elaboración propia. ....	6
Ilustración 5: Fórmula para cálculo de costo. Fuente: Elaboración propia. ....	7
Ilustración 6: Modelo propuesto del funcionamiento del sistema. Fuente: Elaboración propia.....	7
Ilustración 7: Distribución de roles en grupos de trabajo para el desarrollo de la problemática. Fuente: Elaboración propia. ....	9
Ilustración 8: Cuantificación de esfuerzo. Fuente: Elaboración propia. ....	10
Ilustración 9: Modelo lógico de la solución del problema. Fuente: Elaboración propia. ....	11
Ilustración 10: Modelo físico del funcionamiento del sistema. Fuente: Elaboración propia. ....	12
Ilustración 11: Tabla cuantitativa de tiempo secuencial y paralelo algoritmo torah. Fuente: Elaboración propia. ....	15
Ilustración 12: Tabla de SpeedUp con diferentes cantidades de procesadores algoritmo torah. Fuente: Elaboración propia. ....	15
Ilustración 13: Tabla de Eficiencia con diferentes cantidades de procesadores algoritmo torah. Fuente: Elaboración propia. ....	15
Ilustración 14: Tabla de Costo con diferentes cantidades de procesadores algoritmo torah. Fuente: Elaboración propia. ....	16
Ilustración 15: Tabla con los tiempos en segundo algoritmo torah. Fuente: Elaboración propia. ....	16
Ilustración 16: Gráfico comparativo de los tiempos de ejecución secuencial y paralela torah. Fuente: Elaboración propia. ....	16
Ilustración 17: Gráfico que representa el SpeedUp alcanzado por el algoritmo paralelo torah. Fuente: Elaboración propia. ....	17
Ilustración 18: Gráfico que representa la eficiencia asociada a la ejecución del algoritmo paralelo torah. Fuente: Elaboración propia. ....	17
Ilustración 19: Gráfico que representa el costo asociado a la ejecución del algoritmo paralelo torah. Fuente: Elaboración propia. ....	18
Ilustración 20: Comparativa de tiempos de ejecución en búsqueda de patrones. Fuente: Elaboración propia. ....	18
Ilustración 21: Tamaño de la matriz v/s tiempo en algoritmo secuencial percolación. Fuente: Elaboración propia. ....	19
Ilustración 22: Tiempos de ejecución algoritmo paralelo percolación matriz tamaño 100.Fuente: Elaboración propia. ....	20
Ilustración 23: Gráfico de tiempos de ejecución algoritmo paralelo percolación matriz tamaño 100. Fuente: Elaboración propia. ....	20
Ilustración 24: SpeedUp Matriz tamaño 100 algoritmo percolación. Fuente: Elaboración propia. ....	21
Ilustración 25: Eficiencia Matriz tamaño 100 algoritmo percolación. Fuente: Elaboración propia. ....	21
Ilustración 26: Gráfico de costo algoritmo percolación, matriz tamaño 100. Fuente: Elaboración propia. ....	22
Ilustración 27: Tiempos ejecución algoritmo paralelo percolación matriz tamaño 1000. Fuente: Elaboración propia. ....	22
Ilustración 28: Gráfico tiempos de ejecución algoritmo paralelo percolación matriz tamaño 1000. Fuente: Elaboración propia.....	23
Ilustración 29: Gráfico de SpeedUp algoritmo percolación, matriz tamaño 1000. Fuente: Elaboración propia. ....	23
Ilustración 30: Gráfico eficiencia algoritmo percolación paralelo matriz tamaño 1000. Fuente: Elaboración propia. ....	24
Ilustración 31: Gráfico de costo, algoritmo percolación paralelo matriz tamaño 1000. Fuente: Elaboración propia. ....	24
Ilustración 32: Nodo óptimo algoritmo percolación. Fuente: Elaboración propia. ....	25
Ilustración 33: Medición de tiempos secuenciales v/s cantidad de datos algoritmo Fasta. Fuente: Elaboración propia. ....	25
Ilustración 34: Tiempos paralelos, comparación v/s número de procesadores. Fuente: Elaboración propia. ....	26
Ilustración 35: SpeedUp algoritmo Fasta. Fuente: Elaboración propia. ....	26
Ilustración 36: Gráfico de eficiencia algoritmo Fasta. Fuente: Elaboración propia. ....	27
Ilustración 37: Gráfico de costo algoritmo Fasta. Fuente: Elaboración propia. ....	27



## 1. Introducción

El software, en su mayoría se sustenta tradicionalmente en el paradigma de la computación secuencial. Por lo que, al momento de resolver una problemática cualquiera, se construye un algoritmo estrictamente paso a paso y se implementa en un flujo de instrucciones sucesivas e incluyentes con la finalidad de dar solución a la problemática. Todas estas órdenes, independientemente de su extensión, son ejecutadas en la CPU de un computador como cualquier otro proceso. Una vez que las instrucciones terminan dan una respuesta al problema a resolver.

Hoy en día, los grandes volúmenes de datos que buscan ser convertidos en información han aumentado de manera considerable y exponencial, y para encontrarles soluciones a éstos grandes volúmenes de procesamiento se suelen implementar técnicas informáticas para poder obtener una respuesta rápida y como paradigma de solución se plantea el uso de procesamiento paralelo en un clúster computacional. En el contexto de la construcción y aplicación de plataformas WEB de alta eficiencia, en el presente documento se presentarán los aspectos teóricos para el desarrollo de un sistema integral el cual, a través de tres distintos módulos brindará solución a tres grandes problemas de cómputo. La plataforma web desarrollada se caracteriza por ser intuitiva, amigable y responsiva, y tiene por objetivo tomar los requerimientos del usuario mediante un formato específico por cada módulo; cadenas de ADN en formato FASTA para el alineamiento de ellas, encontrar una palabra con el código Torah o una simulación basada en percolación. Una vez reconocido el módulo a utilizar, se envían los procesos a la plataforma clúster. Esta plataforma está compuesta por nodos con distintas características (Clúster heterogéneo) permitiendo la escalabilidad gracias a su configuración en MPI. Finalmente, las respuestas serán enviadas por formato PDF al correo electrónico del usuario.

## 2. Planteamiento del problema

Elaborar una plataforma de servicios implementada sobre una plataforma de clúster heterogéneo escalable y entregue los servicios que se indican, asociados a tres problemas a resolver (reales y de amplio uso, con alta necesidad de cómputo tanto en datos y/o procesamiento), en el contexto de las áreas de investigación de química computacional (Química), genómica computacional (Biología), aplicaciones utilizando modelos de ecuaciones simultáneas (en Física y Sistemas MIMO en Ingeniería electrónica) y Data Mining (aplicaciones productivas/comerciales de Informática).

- 1) Acceso a la plataforma de servicios vía una interface Web.
- 2) Permitir el ingreso de datos y una interacción “amistosa”, utilizando un ambiente de trabajo autodocumentado, multiopción y multifuncional.
- 3) Informar al usuario del estado de su trabajo, con datos como el tiempo estimado de término, el tiempo de ejecución actual y el porcentaje de avance.
- 4) Informar al usuario, por algún medio electrónico, del término de su trabajo.

### 3. Objetivos

#### 3.1 Objetivo general

Implementar mediante un clúster computacional de ambiente de proceso paralelo y secuencial, una plataforma web de interfaz intuitiva que permite al usuario final ingresar una problemática del área de las ciencias, que requieran gran procesamiento de datos, entregando mediante e-mail reportes de estado parcial y finales del problema especificado por el usuario.

#### 3.2 Objetivos específicos

- Analizar recursos disponibles en el laboratorio 5 de informática, elaborando un reporte general, y en función de ello establecer la mejor alternativa para la destinación del uso de los recursos.
- Configurar el entorno de desarrollo del proyecto considerando aspectos de paralización (MPI), servidor de nombre de dominio (DNS), y servidor de correo.
- Investigar y definir, en el área de las ciencias biología, física y data mining, problemas que impliquen gran procesamiento de datos.
- Plantear e implementar una solución computacional tanto en ambiente de ejecución secuencial, como en paralelo a los problemas definidos en la investigación.
- Realizar un análisis de métricas obtenidas de los algoritmos, tales como, tiempos de ejecución en modo secuencial y paralelo, SpeedUp considerando una distinta cantidad de procesadores, y la eficiencia propia de las soluciones planteadas a las problemáticas, obteniendo las conclusiones pertinentes con respecto al óptimo de cada uno.

### 4. Alcances y limitaciones

- Este proyecto se limita con respecto a su capacidad, si se utiliza en su totalidad, sólo se puede consultar por un área a la vez, y para pedir al servicio una solución a otra problemática, ésta deberá esperar a que la primera petición termine y luego comenzar la siguiente.
- La plataforma da solución específica a problemas de cómputo pero el usuario debe tener al menos experiencia teórica en los requisitos, no se puede esperar que la solución sea interpretada por la plataforma.

## 5. Marco teórico

### 5.1 Clúster

El término clúster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora. Hoy en día desempeñan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno. La tecnología de clústers ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

### 5.2 Nodos

Pueden ser simples ordenadores, sistemas multiprocesador o estaciones de trabajo (workstations). En informática, de forma muy general, un nodo es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar. Ahora bien, dentro de la informática la palabra nodo puede referirse a conceptos diferentes según el ámbito en el que nos movamos:

- En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.
- En estructuras de datos dinámicas un nodo es un registro que contiene un dato de interés y al menos un puntero para referenciar (apuntar) a otro nodo. Si la estructura tiene solo un puntero, la única estructura que se puede construir con él es una lista, si el nodo tiene más de un puntero ya se pueden construir estructuras más complejas como árboles o grafos.

El clúster puede estar conformado por nodos dedicados o por nodos no dedicados.

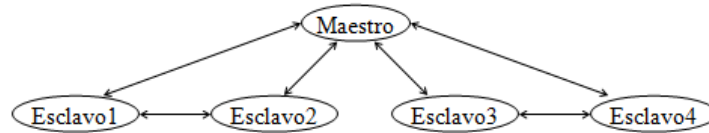
### 5.3 Clúster Beowulf

Para el desarrollo de la plataforma de clúster para el desarrollo del laboratorio, se montó un clúster de tipo Beowulf. Un clúster Beowulf es una colección de ordenadores conectados en una red privada que pueden ser usados para computación paralela. Consisten en una comodidad de hardware utilizando software de uso libre, como Linux o BSD, usualmente junto con PVM (Parallel Virtual Machine) o MPI (Message Passing Interface). El estándar de conexión consiste en un nodo maestro que controla un cierto número de nodos esclavos. Los nodos esclavos tienen un acceso general a todos los archivos del mismo servidor.

### 5.4 Open MPI

MPI significa “Message Passing Interface” y procede del esfuerzo de la comunidad de la computación paralela (industria y grupos de desarrollo de software libre) que creó el MPI Forum con el objetivo de desarrollar un estándar para programación paralela basada en paso de mensajes.

Una aplicación paralela típica sería aquella en la que un proceso maestro crea varios procesos esclavos para realizar, cada uno de ellos, una parte del trabajo global. Gráficamente, este modelo es el siguiente:



*Ilustración 1 : Comunicación maestro esclavos. Elaboración propia.*

Donde el Maestro se comunica con sus esclavos, el Esclavo1 se comunica con el Esclavo2 y el Esclavo3 con el Esclavo4.

### 5.5 Ruido

En comunicación, se denomina ruido a toda señal no deseada que se mezcla con la señal útil que se quiere transmitir. Es el resultado de diversos tipos de perturbaciones que tiende a enmascarar la información cuando se presenta en la banda de frecuencias del espectro de la señal, es decir, dentro de su ancho de banda.

### 5.6 Ubuntu

Ubuntu es un sistema operativo basado en GNU/Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity. Su nombre proviene de la ética homónima, en la que se habla de la existencia de uno mismo como cooperación de los demás.

### 5.7 Python

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

### 5.8 Bootstrap

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

## 5.9 PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

## 5.10 LAMP

LAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Linux, el sistema operativo; En algunos casos también se refiere a LDAP.
- Apache, el servidor web;
- MySQL/MariaDB, el gestor de bases de datos;
- Perl, PHP, o Python, los lenguajes de programación.

La combinación de estas tecnologías es usada principalmente para definir la infraestructura de un servidor web, utilizando un paradigma de programación para el desarrollo.

## 5.11 MailGun

Mailgun es un servicio de email para desarrolladores que cuenta con una poderosa Api que facilita su uso y acceso desde cualquier plataforma de desarrollo. Laravel 5 permite integrar este servicio a nuestra aplicación, facilitando el envío de correos desde cualquier controlador.

## 5.12 Ley de Amdahl

El aumento de rendimiento que puede obtenerse al mejorar alguna parte de una computadora puede calcularse utilizando la Ley de Amdahl.

La Ley de Amdahl establece que la mejora obtenida en el rendimiento al utilizar algún modo de ejecución más rápido está limitada por la fracción de tiempo que se pueda utilizar ese modo más rápido. Define la ganancia de rendimiento o aceleración (speedup) que puede lograrse al utilizar una característica particular.

## 5.13 SpeedUp

En el campo de la arquitectura de computadores, aumento de velocidad es una medida para la mejora relativa de rendimiento al ejecutar una tarea. La noción de aceleración fue establecida por la Ley de Amdahl, que se centra en particular en el contexto de procesamiento en paralelo. Sin embargo, la aceleración puede ser utilizada más generalmente para mostrar el efecto de cualquier mejora del rendimiento.

Para valores de latencia, aceleración se define por la siguiente fórmula:

$$S = \frac{T_{antiguo}}{T_{nuevo}}$$

Dónde:

$$\begin{aligned} S &= \text{SpeedUp resultante} \\ T_{antiguo} &= \text{Tiempo antiguo de ejecución, o sin mejora} \\ T_{nuevo} &= \text{Tiempo nuevo de ejecución, o con mejora} \end{aligned}$$

*Ilustración 2: Fórmula para cálculo de SpeedUp. Fuente: Elaboración propia.*

Para valores de rendimiento, el rendimiento mejorado estará en el numerador y el rendimiento original estará en el denominador. Esto se debe a que es una cantidad relativa, estamos comparando dos casos concretos de ejecución.

$$S = \frac{P_{nuevo}}{P_{antiguo}}$$

$$\begin{aligned} S &= \text{SpeedUp resultante} \\ P_{antiguo} &= \text{performance antiguo, sin mejora} \\ P_{nuevo} &= \text{performance nuevo, con mejora} \end{aligned}$$

*Ilustración 3: Fórmula para cálculo de SpeedUp. Fuente: Elaboración propia.*

Cuando se aplica en el caso paralelo, aumento de velocidad se puede predecir a partir de la ley de Amdahl.

## 5.14 Eficiencia

Llamando a  $S_p$  al SpeedUp para  $p$  procesadores, Aceleración lineal o SpeedUp ideal se obtiene cuando  $S_p = p$ . Cuando se ejecuta un algoritmo con la aceleración lineal, duplicando el número de procesadores se duplica la velocidad. Como esto es ideal, se considera muy buena escalabilidad.

La eficiencia es una medida de rendimiento que se define como:

$$E_p = \frac{S_p}{p}$$

Dónde:

$$\begin{aligned} E_p &= \text{Eficiencia en } p \text{ procesadores} \\ S_p &= \text{SpeedUp para } p \text{ procesadores} \\ p &= \text{procesadores} \end{aligned}$$

*Ilustración 4: Fórmula para cálculo de Eficiencia. Fuente: Elaboración propia.*

## 5.15 Costo

También llamado producto procesador producto procesador-tiempo o trabajo.

Un sistema es óptimo en coste si el coste de resolver un problema en el sistema paralelo tiene el mismo crecimiento asintótico (en función del tamaño del problema) que el coste en el mejor sistema serie equivalente.

$$\text{Costo} = p * T_p$$

Dónde:

$\text{Costo} = \text{Costo del proceso}$

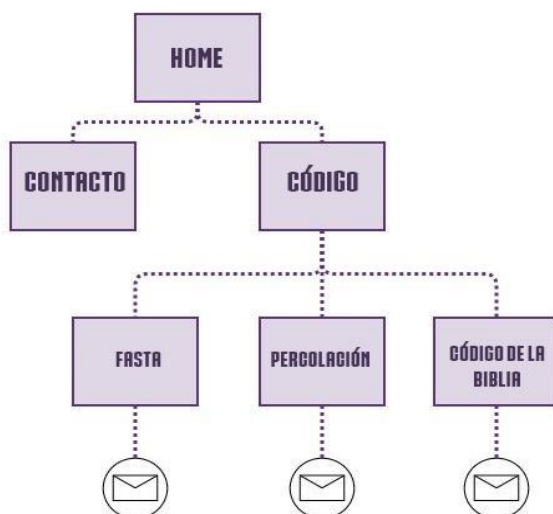
$T_p = \text{Tiempo paralelo}$

$p = \text{procesadores}$

*Ilustración 5: Fórmula para cálculo de costo. Fuente: Elaboración propia.*

## 6. Modelo propuesto del funcionamiento del sistema

En la siguiente ilustración, se puede apreciar el diagrama de flujo del funcionamiento general que se propuso para la implementación de la web de servicios. La plataforma se orienta a resolver tres grandes problemas de computo: FASTA, Código de la Biblia y Percolación. Se debe asumir que cada problema especificado debe ser utilizado por un usuario específico y que tenga un conocimiento y manejo teórico de la problemática. Se puede intuir que el flujo de información es dependiente al conocimiento y el uso que le quiera asignar el usuario por ende se accede al home de la página, y a partir de ella se puede dirigir a la solución de la problemática a evaluar brindando ciertos parámetros de entrada inherentes al problema a resolver obteniendo una respuesta que será enviada mediante e-mail al remitente que realice la visita a la página.



*Ilustración 6: Modelo propuesto del funcionamiento del sistema. Fuente: Elaboración propia*

## 7. Metodología de desarrollo

La metodología que se utilizó fue la de programación extrema o eXtreme Programming (de ahora en adelante, XP), que es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

En base a esta acepción, se puede plantear el objetivo el cual es lograr la solución de software a partir de la metodología planteada de obtener una forma de trabajo idónea, la cual determine una forma limpia de desarrollar el trabajo, con resoluciones coherentes y conclusiones.

Teniendo los objetivos fijos, sabiendo las fechas de entregas (avances y final) y además contemplando que existe un número reducido de personas para realizar el trabajo, la metodología que más se acomoda para este caso es la metodología ágil XP. Aunque se agregarán ítems que no contempla esta metodología, como lo son las reuniones y un tablero Kanban, moderado mediante Asana, para ver el control de avances.

En la planificación de este proyecto se definieron los roles que utiliza esta metodología:

- **Programador:** Escribe las pruebas unitarias y produce el código del sistema.
- **Cliente:** Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar el mayor valor de negocio.
- **Tester:** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Tracker:** Es el encargado de seguimiento. Proporciona realimentación al equipo. Debe verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.
- **Coach:** Responsable del proceso global. Guía a los miembros del equipo para seguir el proceso correctamente.
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Ayuda al equipo a resolver un problema específico.
- **Big Boss:** Es el dueño de la tienda y el vínculo entre clientes y programadores. Su labor esencial es la coordinación.

El equipo de trabajo, se dividió en 4 áreas, que son: Gerencia, Documentación y Control, Plataforma y Desarrollo y Testing. Estas áreas permitieron la movilidad de personas para mantener un equilibrio, pero sin dejar de lado su cargo asignado.



Grupo Gerencia	
- Roberto Oñate Piedras	Subjefe de Proyecto
- Diego Hernández García	Jefe de Proyecto
Grupo Zero	
- Manuel Venegas Solís	Jefe Grupo
- Rodrigo Reyes	Documentación y testing
- Miguel Núñez Gajardo	Programador
Grupo Uno	
- Sergio Abarca Flores	Jefe Grupo
- Claudio Piña	Programador
- Rosa González	Documentación y testing
Grupo Dos	
- Carlos Guerrero	Jefe Grupo
- Daniel Gutiérrez	Programador
- Rodrigo Arratia	Documentación y testing

*Ilustración 7: Distribución de roles en grupos de trabajo para el desarrollo de la problemática. Fuente: Elaboración propia.*

Las reuniones que se agregaron a esta metodología fueron de carácter, realizándose todas las semanas los días martes a las 11:15, con una duración mínima de 30 minutos y de extensión variable dependiendo de lo planificado para esa semana. En dónde se conversaron los avances realizados, complejidades encontradas y evaluaciones personales. Cuando se detectó un progreso lento, se agendó una reunión extra programática para revisar y conocer las razones del por qué, encontrando una solución.

En primera instancia a cada grupo se le asignó una problemática, previamente conversada y luego se llevó a cabo el proceso de recopilación de información acerca del tema que trabajó cada grupo. Una vez se terminó la investigación se llevó a cabo un análisis de esta información, con el fin de entender a nivel general la funcionalidad y de qué trataba cada problema.

Entendido el proceso anterior, se definió el eje que involucra al desarrollo del proyecto; Metodología de grupo. Esta se refiere a cómo los integrantes del grupo curso se relacionan para llevar a cabo la planificación del proyecto y el desarrollo del mismo, tomando en cuenta que todos los integrantes deben responder a las metodologías de desarrollo como también este último a su plan y condiciones de trabajo (tiempo estimado, entre otros).

En cuanto a metodología se pasó por un proceso de planificación, con asignación de roles, y designación de roles y tareas a cada integrante con un proceso de supervisión con seguimiento de todas las tareas asignadas por parte del grupo líder.

## 8. Evaluación económica

Los costos asociados al proyecto son múltiples, entre ellos están el costo de los equipos y costo de desarrollo del proyecto.

### 8.1 Recursos físicos

Dado que el clúster es heterogéneo, se cuenta con 3 tipos distintos de recursos, el de los equipos más potentes y los encargados de la mayor cantidad de cómputo fueron evaluados con un valor de \$563.667, y el Switch de \$276.290. El nodo maestro tiene un valor de \$202.860 aproximado y el computador de menor potencia de cálculo, fue evaluado en \$40.000 dada su generación y estado.

Durante la realización del proyecto, se utilizaron quince computadores HP, un computador armado, un Switch Cisco y un pc reacondicionado, el monto total \$8.974.155.

### 8.2 Cuantificación de esfuerzo

Los siguientes costos son asociados al personal encargado de desarrollo de la solución, control de procesos y calidad, documentación y testing y finalmente el equipo encargado de montar y mantener la plataforma. Se muestra una tabla con la cantidad de horas y la cantidad de UF asociadas por dichas horas, tanto a nivel de planificación como de ejecución (trabajadas).

Equipo de trabajo	UF*Hr	Hr*Tr	UF*Tr	Hr*Plan	UF*Plan	Hr*Dif	UF*Dif
Diego Hernandez	2,35	109,00	256,15	100,50	236,18	8,50	19,98
Roberto Oñate	2,35	118,00	277,30	104,50	245,58	13,50	31,73
Sergio Abarca	1,95	117,50	229,13	106,50	207,68	11,00	21,45
Manuel Venegas	1,95	122,00	237,90	87,50	170,63	34,50	67,28
Carlos Guerrero	1,95	112,50	219,38	82,00	159,90	30,50	59,48
Rosa Gonzalez	1,50	100,50	150,75	85,50	128,25	15,00	22,50
Rodrigo Arratia	1,10	79,50	87,45	69,00	75,90	10,50	11,55
Rodrigo Reyes	0,90	56,50	50,85	69,00	62,10	-12,50	-11,25
Miguel Nuñez	0,90	111,00	99,90	73,00	65,70	38,00	34,20
Daniel Gutierrez	0,70	81,00	56,70	72,50	50,75	8,50	5,95
Claudio Piña	0,70	93,00	65,10	84,00	58,80	9,00	6,30
<b>Presupuesto Proyecto (Horas)</b>	-	<b>1100,50</b>	-	<b>934,00</b>		<b>166,50</b>	
<b>Costo capital humano (UF)</b>	-	-	<b>1730,60</b>		<b>1461,45</b>		<b>269,15</b>

*Ilustración 8: Cuantificación de esfuerzo. Fuente: Elaboración propia.*

Como se puede observar, hay columnas además de la que contiene nombre de los integrantes del grupo. La primera muestra el valor en UF de la hora trabajada de cada integrante. Luego las siguientes indican horas trabajadas (Hr\*Tr) y costo total en UF de dichas horas (UF\*Tr), horas planificadas (Hr\*Plan) y costo total asociado a las horas planificadas (UF\*Plan) y por último, se encuentran las diferencias de horas (Hr\*Dif), entre planificadas y trabajadas, y la diferencias de costos (UF\*Dif), entre costos planificados y costos asociados al trabajo concretado.

En la tabla descrita se observa que las horas trabajadas fueron mayores a las establecidas en la planificación inicial. Por lo tanto se superó el presupuesto inicial, la diferencia fue de 269,45UF. En la última etapa hubo que extremar recursos para lograr suplir el retraso y cumplir con los tiempos establecidos.

## 9. Diseño y arquitectura del software

### 9.1 Modelo lógico

En la siguiente ilustración, se puede apreciar el flujo de información desde que un grupo de usuarios al menos con conocimiento en la resolución de la problemática ingresan a la plataforma y plantean sus requerimientos de entrada dependiendo del problema a resolver. Los requerimientos llegan a un servidor principal luego el servidor en función de los procesadores (o hilos) disponibles le asigna el uso de recursos. Luego en función de los recursos asignados, retornan al servidor las sub-respuestas, y este las unifica para posteriormente enviar la respuesta general mediante e-mail al usuario que realizó la petición.

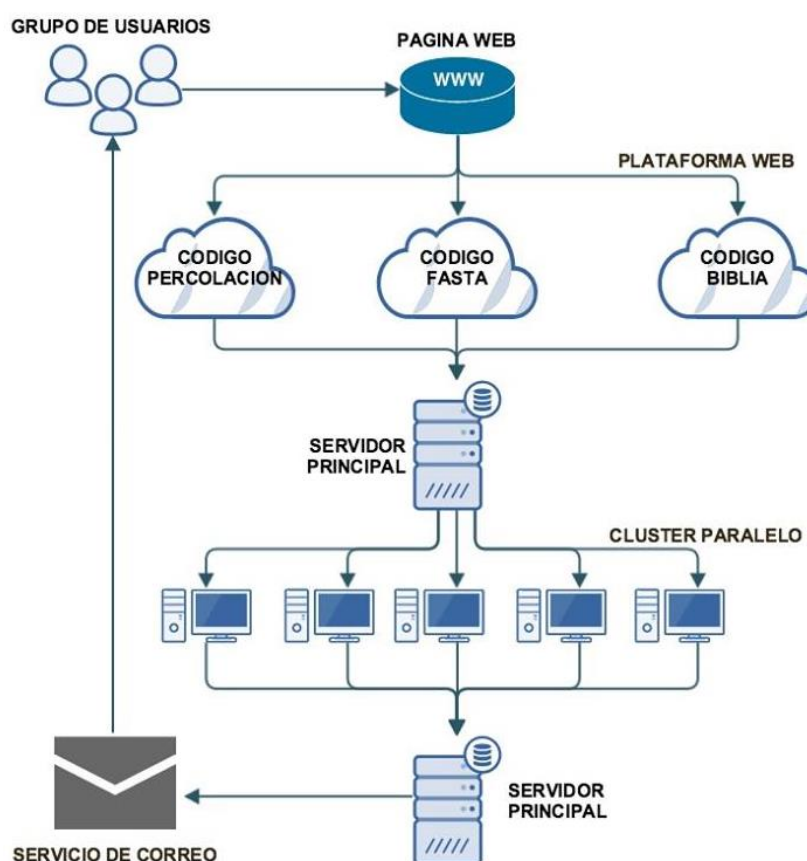


Ilustración 9: Modelo lógico de la solución del problema. Fuente: Elaboración propia.

## 9.2 Modelo físico

En la siguiente ilustración, se puede apreciar cómo se distribuyeron los recursos disponibles para montar la plataforma. Se decidió, para brindarle valor agregado contextualizar el clúster Beowulf heterogéneo, utilizar superhéroes de Marvel para hacer la analogía de ser supercomputadores. A grandes rasgos también se ve como es la estructura de Input/Output aislando el uso de nodos del proceso de conocimiento del usuario.

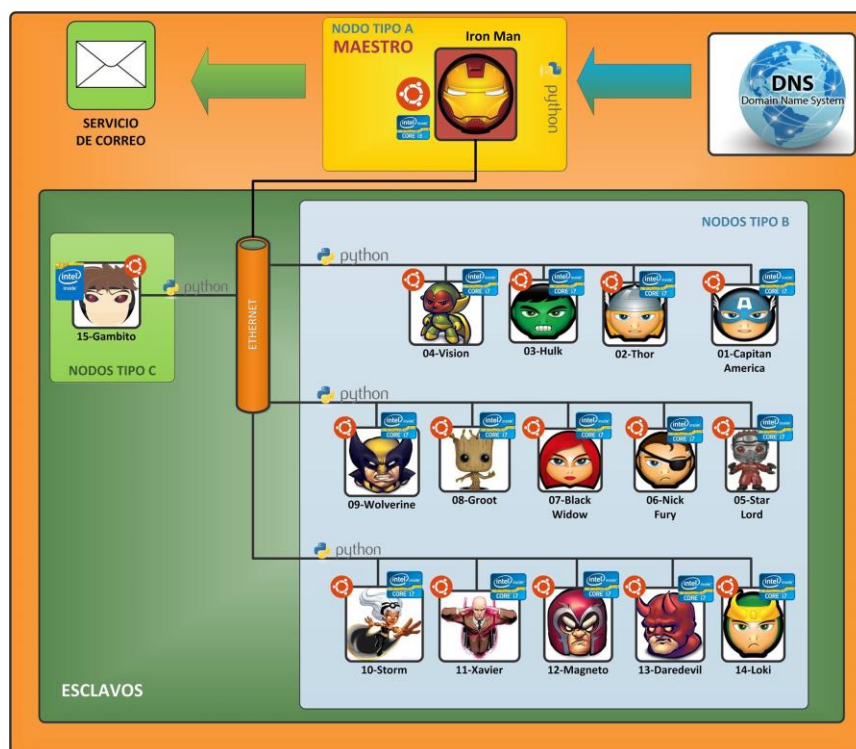


Ilustración 10: Modelo físico del funcionamiento del sistema. Fuente: Elaboración propia.

## 9.3 Herramientas utilizadas

- **Sistema Operativo:** Ubuntu 14.04 LTS, es un sistema operativo basado en GNU/LINUX y que distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity.
- **Lenguaje de Programación:** Python fue utilizado para configurar el Clúster, la elección de este es porque hace hincapié en una sintaxis que favorezca un código legible. Por otro lado PHP se utilizó para poder pasar los parámetros ingresados en la web al código paralelo en Python.
- **Middleware:** MPI es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.

- **Editor de código:** Sublimetext, es un estándar que define la sintaxis y la semántica de las funciones contenidas en una de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.
- **Software de Gestión:** Asana, es una aplicación web y móvil diseñada para mejorar la comunicación y colaboración en equipo.
- **Lugar Físico:** Laboratorio 5 UTEM.
- **Monitor de Procesadores:** Htop, programa de Ubuntu para monitorear la memoria.
- **Servidor WEB:** Apache implementa el protocolo HTTP/1.1 y la noción de sitio virtual.
- **Numpy:** es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.
- **re:** Este módulo cuenta con funciones para trabajar con expresiones regulares y cadenas. La función match comprueba si una cadena es equivalente a una expresión regular. Se basa en el siguiente formato: match (expresión-regular, cadena, [flag]). Para la manipulación y las coincidencias complejas, las expresiones regulares ofrecen soluciones concisas y optimizadas. La manipulación de cadenas como expresiones regulares de Unix se da mediante:
  - ❑ search - busca un patrón en cualquier lugar de la cadena
  - ❑ match - busca sólo al comienzo de la cadena
  - ❑ split - separa una cadena en campos según un delimitador
  - ❑ sub,subn - sustitución de cadenas
- **sys:** Este módulo provee acceso a funciones y objetos mantenidos por el intérprete. Los programas frecuentemente necesitan procesar argumentos de línea de órdenes. Estos argumentos se almacenan en el atributo argv del módulo sys como una lista. El módulo sys también tiene atributos para stdin, stdout, y stderr. Este último es útil para emitir mensajes de alerta y error para que se vean incluso cuando se haya redireccionado stdout. Este módulo interactúa con el sistema de Python mediante las siguientes funciones:
  - ❑ exit() - salida!
  - ❑ argv - acceso a los argumentos de la línea de comando
  - ❑ path - acceso a la ruta del sistema
  - ❑ ps1 - cambia el prompt '>>>' de Python
  - ❑ entre otros.

- **time:** Es un módulo de bajo nivel que proporciona diversas funciones relacionadas con el tiempo.

Funciones de hora y fecha tales como:

- ☐ time - hora actual (expresada en segundos)
  - ☐ gmtime - convertir hora en segundos a UTC (GMT)
  - ☐ localtime - convertir a hora local
  - ☐ mktime - inverso de localtime
  - ☐ sleep - detener el programa por n segundos
  - ☐ entre otras.
- **Biopython:** permite representar secuencias biológicas y anotaciones de genomas y es capaz de comunicar con las bases de datos biológicos en línea del NCBI para hacer cálculos. Además, gracias a diversos módulos, puede ser utilizada para trabajar sobre proyectos relativos al alineamiento de secuencias, cálculo de estructuras proteicas, genética de poblaciones, filogenética e inteligencia artificial.
  - **PyPDF2:** Es una biblioteca PDF para Python capaz de dividir, combinar entre sí, cortar, y transformar las páginas de los archivos PDF. También puede añadir datos personalizados, opciones de visualización, y contraseñas para los archivos PDF. Se puede recuperar el texto y los metadatos de los archivos PDF, así como combinar archivos enteros juntos.
  - **subprocess (common):** El programa diseñado en proyectos anteriores del Código de la Biblia fue construido en base a Python en su versión 2.7.6, sin embargo en un afán de construir código reutilizable y bajo reglas de buenas prácticas, es que establecimos funcionalidades más actualizadas para Python en su versión 3.4.3. Es por ello que módulos como common que ya se encontraban obsoletos, tuvieron que ser reemplazadas por subprocess.

## 9.4 Análisis y desempeño de optimización

### 9.4.1 Algoritmo de Torah

#### 9.4.1.1 Casos de uso

Los casos de usos que se decidieron a probar se diferencian en la cantidad de palabras que contiene un patrón, en donde cada palabra tiene una cantidad arbitraria de caracteres. Los casos de uso son los siguientes:

- chile
- dios existe
- monsanto muerte mundial
- iluminati obama poder mundo

Estos fueron probados a nivel secuencial y en paralelo con diferentes cantidades de nodos para encontrar cual es la cantidad de procesadores donde se encuentra el óptimo en tiempo de ejecución.

#### 9.4.1.2 Tablas de métricas

Patron	Tiempo Secuencial	Tiempo Paralelo (16 P)	Tiempo Paralelo (32 P)	Tiempo Paralelo (48 P)
chile	171.2	31.953	28.767	23.557
dios existe	346.764	61.086	58.274	47.707
monsanto muerte mundial	504.661	99.068	87.222	152.928
iluminati obama poder mundo	680.955	125.487	114.051	118.343

Tiempo Paralelo (64 P)	Tiempo Paralelo (80 P)	Tiempo Paralelo (96 P)	Tiempo Paralelo (112 P)	Tiempo Paralelo (128 P)
21.036	7.507	18.874	52.688	16.655
42.095	14.849	37.107	105.704	33.751
63.82	22.345	56.889	152.928	60.602
83.055	29.339	71.741	212.526	70.238

Ilustración 11: Tabla cuantitativa de tiempo secuencial y paralelo algoritmo torah. Fuente: Elaboración propia.

#### 9.4.1.3 Tablas de métricas de desempeño

Patron	Speed Up P16	Speed Up P32	Speed Up P48	Speed Up P64
chile	5.357869371	5.951263601	7.267478881	8.138429359
dios existe	5.676652588	5.950578303	7.268618861	8.237652928
monsanto muerte mundial	5.09408689	5.785937034	3.299990845	7.90756816
iluminati obama poder mundo	5.426498362	5.970618408	5.754079244	8.198844139

Patron	Speed Up P80	Speed Up P96	Speed Up P112	Speed Up P128
chile	22.80538164	9.070679241	3.249316732	10.27919544
dios existe	23.35268368	9.344975342	3.280519186	10.27418447
monsanto muerte mundial	22.58496308	8.870976814	3.299990845	8.32746444
iluminati obama poder mundo	23.20989127	9.491852637	3.204102086	9.694965688

Ilustración 12: Tabla de SpeedUp con diferentes cantidades de procesadores algoritmo torah. Fuente: Elaboración propia.

Patron	Eficiencia P16	Eficiencia P32	Eficiencia P48	Eficiencia P64
chile	0.3348668357	0.1859769875	0.15140581	0.1271629587
dios existe	0.3547907868	0.185955572	0.1514295596	0.128713327
monsanto muerte mundial	0.3183804306	0.1808105323	0.06874980928	0.1235557525
iluminati obama poder mundo	0.3391561476	0.1865818252	0.1198766509	0.1281069397

Patron	Eficiencia P80	Eficiencia P96	Eficiencia P112	Eficiencia P128
chile	0.2850672705	0.0944862421	0.02901175654	0.08030621435
dios existe	0.291908546	0.09734349314	0.02929034987	0.08026706616
monsanto muerte mundial	0.2823120385	0.09240600848	0.02946420398	0.06505831594
iluminati obama poder mundo	0.2901236409	0.09887346496	0.02860805434	0.07574191944

Ilustración 13: Tabla de Eficiencia con diferentes cantidades de procesadores algoritmo torah. Fuente: Elaboración propia.

Patron	Costo P16	Costo P32	Costo P48	Costo P64
chile	1.9970625	0.89896875	0.4907708333	0.3286875
dios existe	3.817875	1.8210625	0.9938958333	0.657734375
monsanto muerte mundial	6.19175	2.7256875	3.186	0.9971875
iluminati obama poder mundo	7.8429375	3.56409375	2.465479167	1.297734375
Patron	Costo P80	Costo P96	Costo P112	Costo P128
chile	0.0938375	0.1966041667	0.4704285714	0.1301171875
dios existe	0.1856125	0.38653125	0.9437857143	0.2636796875
monsanto muerte mundial	0.2793125	0.59259375	1.365428571	0.473453125
iluminati obama poder mundo	0.3667375	0.7473020833	1.897553571	0.548734375

Ilustración 14: Tabla de Costo con diferentes cantidades de procesadores algoritmo torah. Fuente: Elaboración propia.

Patron	Secuencial	Heterogeneo	Homogeneo
chile	171.2	138.623	7.507
dios existe	346.764	277.386	14.849
monsanto muerte mundial	504.661	406.211	22.345
iluminati obama poder mundo	680.955	559.876	29.339

Ilustración 15: Tabla con los tiempos en segundo algoritmo torah. Fuente: Elaboración propia.

#### 9.4.1.4 Gráficos

##### 9.4.1.4.1 Secuencial vs paralelo

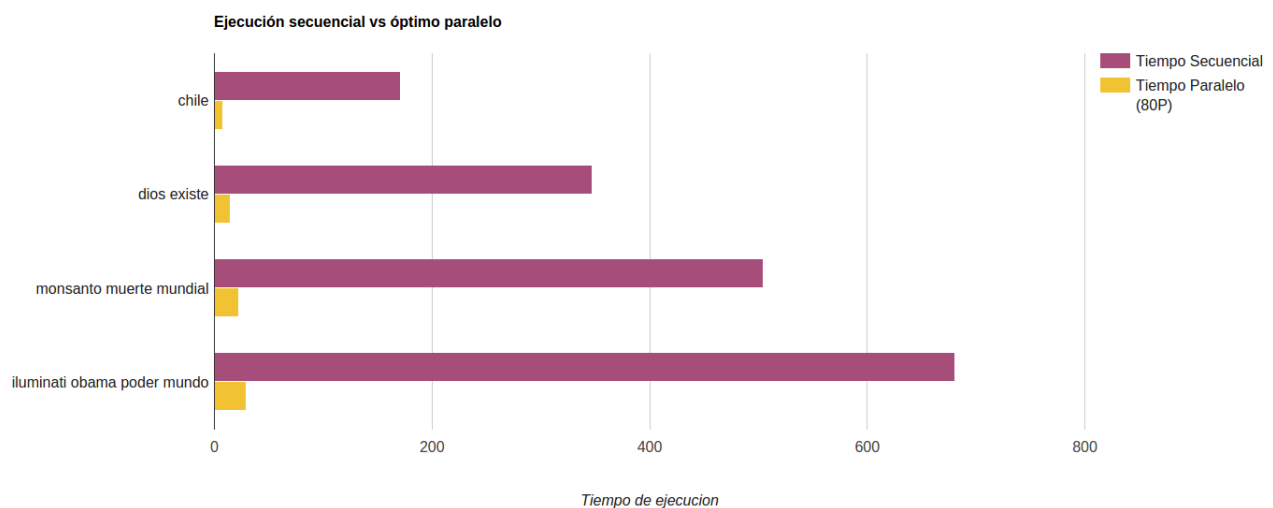


Ilustración 16: Gráfico comparativo de los tiempos de ejecución secuencial y paralela torah. Fuente: Elaboración propia.



#### 9.4.1.4.2 SpeedUp

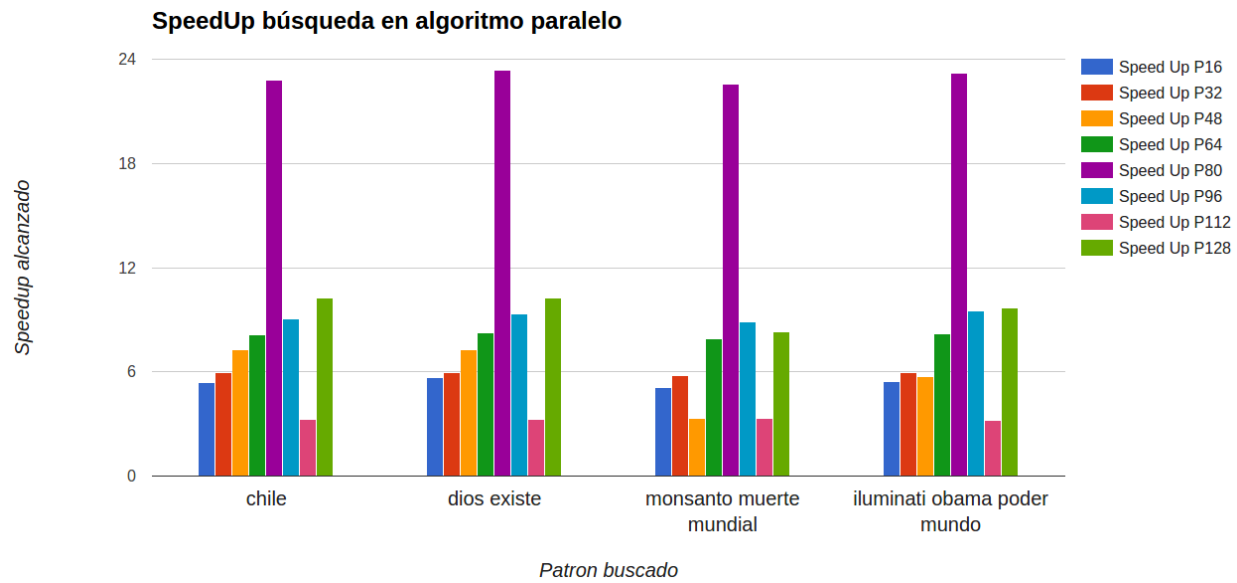


Ilustración 17: Gráfico que representa el SpeedUp alcanzado por el algoritmo paralelo torah. Fuente: Elaboración propia.

#### 9.4.1.4.3 Eficiencia

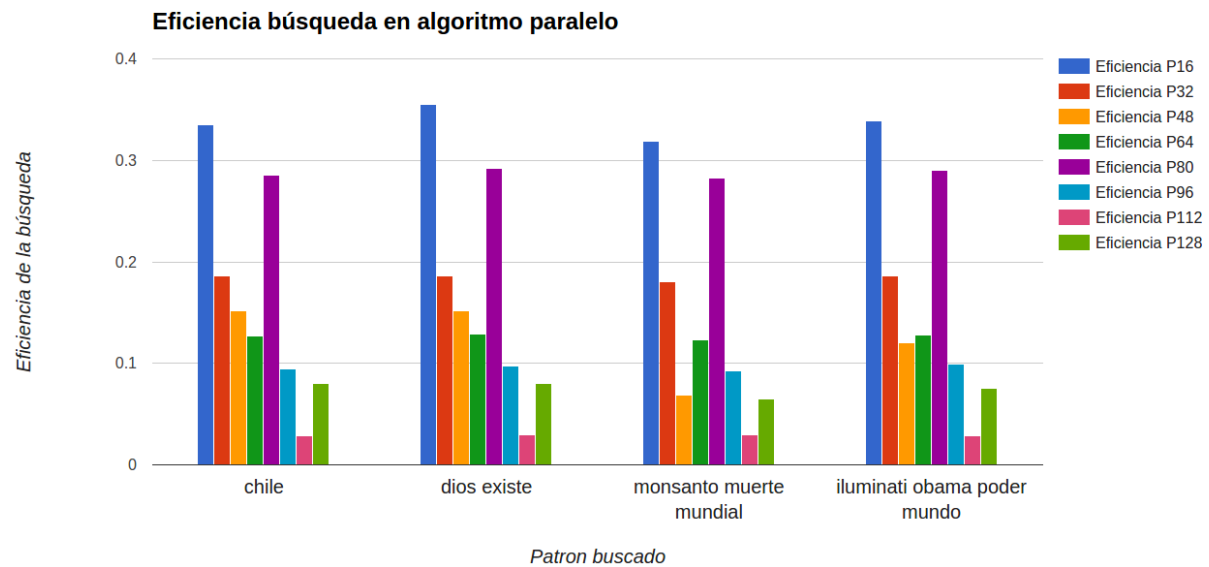


Ilustración 18: Gráfico que representa la eficiencia asociada a la ejecución del algoritmo paralelo torah. Fuente: Elaboración propia.

#### 9.4.1.4.4 Costo

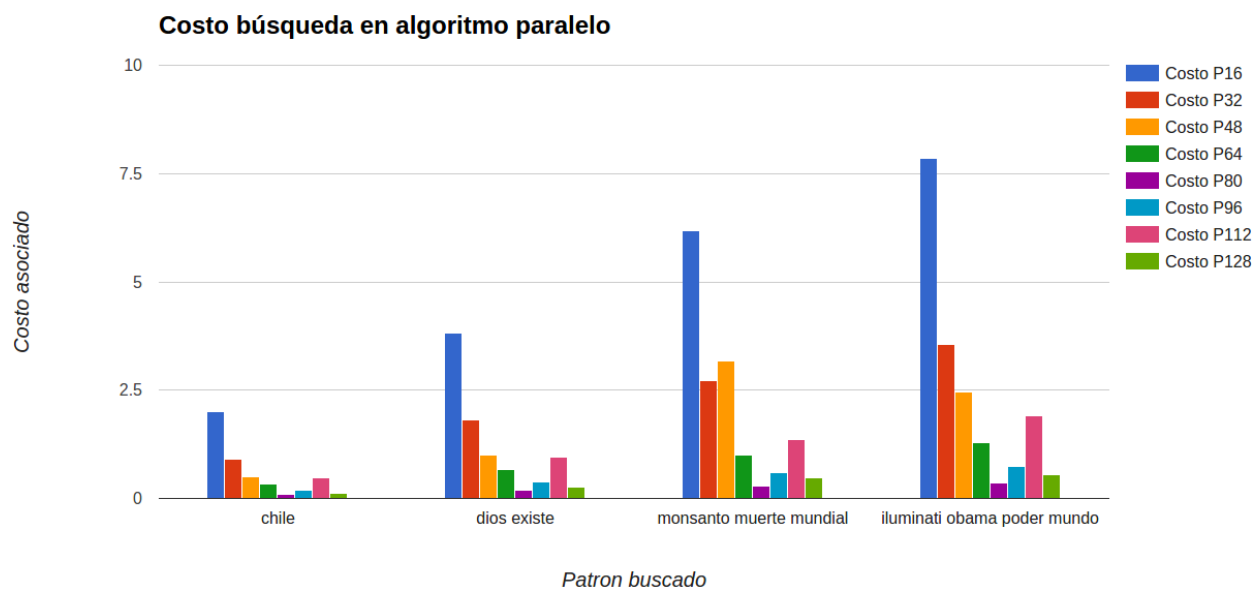


Ilustración 19: Gráfico que representa el costo asociado a la ejecución del algoritmo paralelo torah. Fuente: Elaboración propia.

#### 9.4.1.4.5 Heterogéneos vs Homogéneos

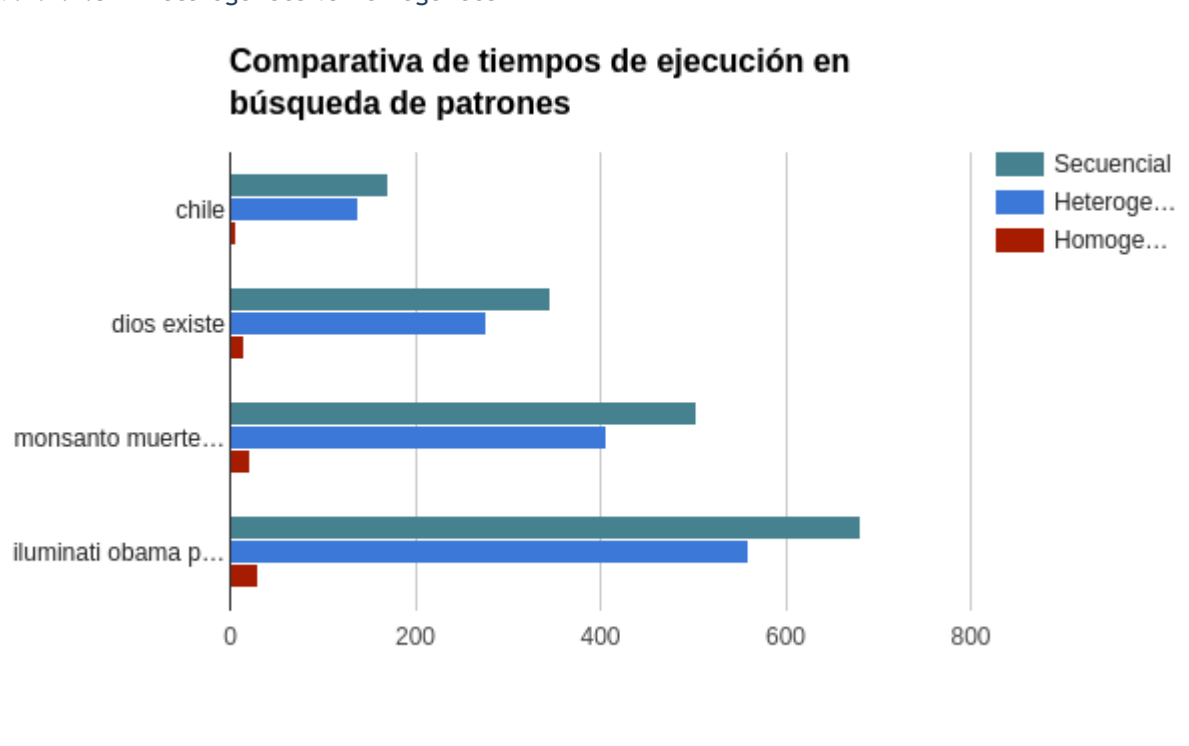


Ilustración 20: Comparativa de tiempos de ejecución en búsqueda de patrones. Fuente: Elaboración propia.

## 9.4.2 Algoritmo de Percolación

### 9.4.2.1 Algoritmo Secuencial

Tamaño Matriz	Tiempo (segundos)
10	0,14914
50	2,38995
100	9,47990
250	67,96358
500	290,57180
1000	1142,07671
1500	3403,16735
2000	7542,09386
5000	37633,59334
10000	150380,62149

Tabla 1. Tiempos ejecución secuencial.  
Elaboración propia.

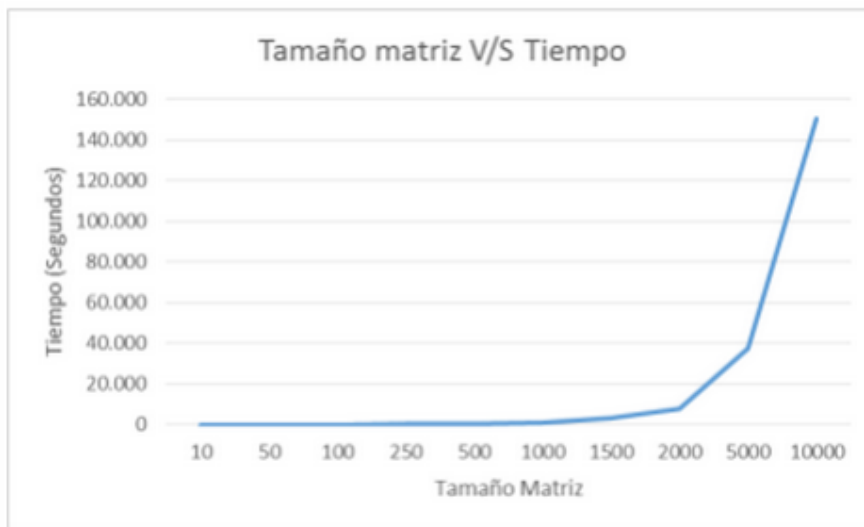


Gráfico 1. Tiempos ejecución secuencial. Elaboración propia.

*Ilustración 21: Tamaño de la matriz v/s tiempo en algoritmo secuencial percolación. Fuente: Elaboración propia.*

Al obtener los tiempos del código secuencial, podemos apreciar que tiene un comportamiento exponencial, de hecho, desde el gráfico se puede obtener la función de comportamiento la cual es:

$$y = 0,1124e^{1,4459x}$$

Cabe destacar que este programa bajo esta modalidad, solo permite utilizar matrices de tamaño 1500, superando este valor, se desborda la memoria RAM del computador.

#### 9.4.2.2 Algoritmo Paralelo con matriz tamaño 100

N° procesadores	Tiempo (Segundos)	Speed Up	Eficiencia	Costo
4	2,48255	3,81861	0,95465	0,62064
8	1,98546	4,77466	0,59683	0,24818
12	1,39096	6,81538	0,56795	0,11591
16	1,06168	8,92912	0,55807	0,06636
20	0,85846	11,04296	0,55215	0,04292
24	0,73434	12,90947	0,53789	0,03060
28	0,89817	10,55473	0,37695	0,03208
32	0,56962	16,64261	0,52008	0,01780
36	0,92527	10,24559	0,28460	0,02570
40	0,91091	10,40703	0,26018	0,02277
44	1,06932	8,86533	0,20148	0,02430
48	0,41041	23,09865	0,48122	0,00855
52	0,82031	11,55643	0,22224	0,01578
56	1,48513	6,38323	0,11399	0,02652
60	0,95732	9,90258	0,16504	0,01596
64	0,32262	29,38423	0,45913	0,00504
68	1,49546	6,33911	0,09322	0,02199
72	1,00669	9,41688	0,13079	0,01398
76	0,50526	18,76236	0,24687	0,00665
80	1,96712	4,81917	0,06024	0,02459
84	1,57464	6,02037	0,07167	0,01875
88	1,17697	8,05448	0,09153	0,01337
92	0,78301	12,10692	0,13160	0,00851
96	0,25387	37,34164	0,38898	0,00264
100	2,43469	3,89367	0,03894	0,02435
104	2,14301	4,42364	0,04254	0,02061
108	1,83404	5,16885	0,04786	0,01698
112	1,55108	6,11182	0,05457	0,01385
116	1,29224	7,33602	0,06324	0,01114
120	0,97516	9,72135	0,08101	0,00813
124	0,70254	13,49372	0,10882	0,00567

Ilustración 22: Tiempos de ejecución algoritmo paralelo percolación matriz tamaño 100. Fuente: Elaboración propia.

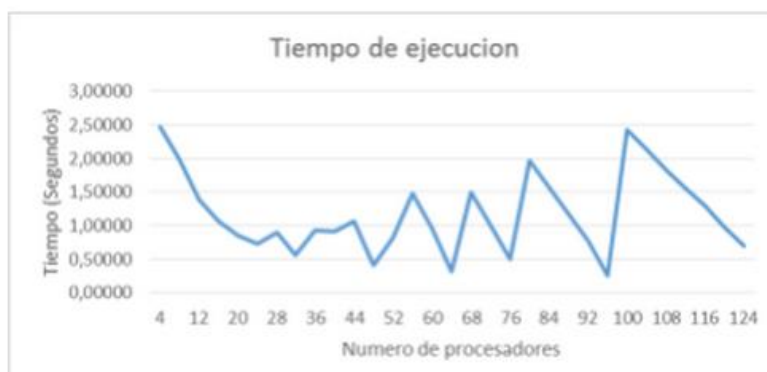


Ilustración 23: Gráfico de tiempos de ejecución algoritmo paralelo percolación matriz tamaño 100. Fuente: Elaboración propia.

#### 9.4.2.2.1 SpeedUp



Ilustración 24: SpeedUp Matriz tamaño 100 algoritmo percolación. Fuente: Elaboración propia.

#### 9.4.2.2.2 Eficiencia

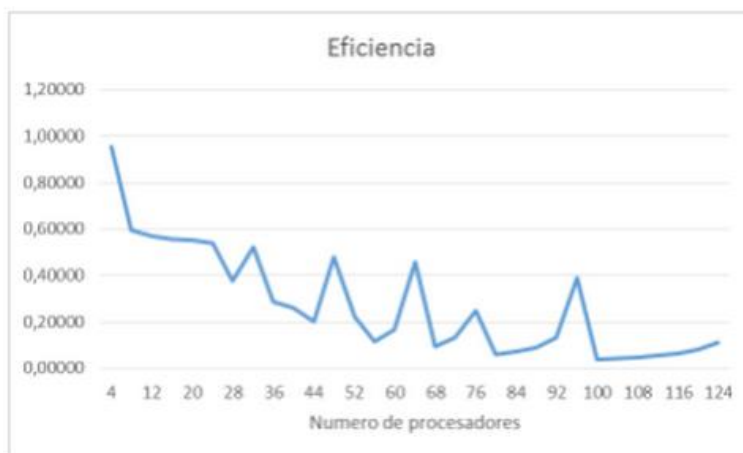


Ilustración 25: Eficiencia Matriz tamaño 100 algoritmo percolación. Fuente: Elaboración propia.

#### 9.4.2.2.3 Costo



Ilustración 26: Gráfico de costo algoritmo percolación, matriz tamaño 100. Fuente: Elaboración propia.

#### 9.4.2.3 Algoritmo paralelo con matriz tamaño 1000

N° procesadores	Tiempo (Segundos)	Speed Up	Eficiencia	Costo
4	311,10135	3,67108	0,91777	77,77534
8	217,85192	5,24244	0,65531	27,23149
12	155,03276	7,36668	0,61389	12,91940
16	115,16685	9,91671	0,61979	7,19793
20	92,80215	12,30658	0,61533	4,64011
24	77,55559	14,72591	0,61358	3,23148
28	111,01886	10,28723	0,36740	3,96496
32	58,35892	19,56987	0,61156	1,82372
36	114,13921	10,00600	0,27794	3,17053
40	110,97567	10,29124	0,25728	2,77439
44	134,17492	8,51185	0,19345	3,04943
48	40,00006	28,55187	0,59483	0,83333
52	90,98662	12,55214	0,24139	1,74974
56	176,83009	6,45861	0,11533	3,15768
60	101,16688	11,28904	0,18815	1,68611
64	30,00008	38,06913	0,59483	0,46875
68	163,95615	6,96574	0,10244	2,41112
72	97,74769	11,68392	0,16228	1,35761
76	32,40725	35,24139	0,46370	0,42641
80	226,71838	5,03742	0,06297	2,83398
84	170,62525	6,69348	0,07968	2,03125
88	121,47308	9,40189	0,10684	1,38038
92	68,40722	16,69527	0,18147	0,74356
96	20,65078	55,30428	0,57609	0,21511
100	289,27081	3,94812	0,03948	2,89271
104	244,54357	4,67024	0,04491	2,35138
108	211,20356	5,40747	0,05007	1,95559
112	170,50004	6,69840	0,05981	1,52232
116	130,22274	8,77018	0,07560	1,12261
120	97,61242	11,70012	0,09750	0,81344
124	64,42974	17,72593	0,14295	0,51959

Ilustración 27: Tiempos ejecución algoritmo paralelo percolación matriz tamaño 1000. Fuente: Elaboración propia.

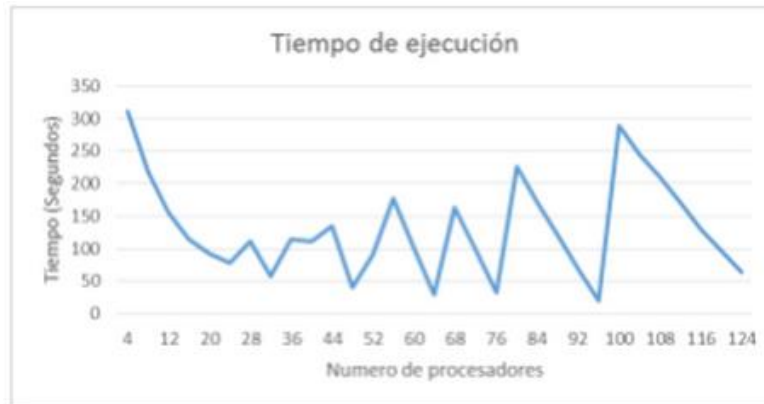


Ilustración 28: Gráfico tiempos de ejecución algoritmo paralelo percolación matriz tamaño 1000. Fuente: Elaboración propia.

#### 9.4.2.3.1 SpeedUp

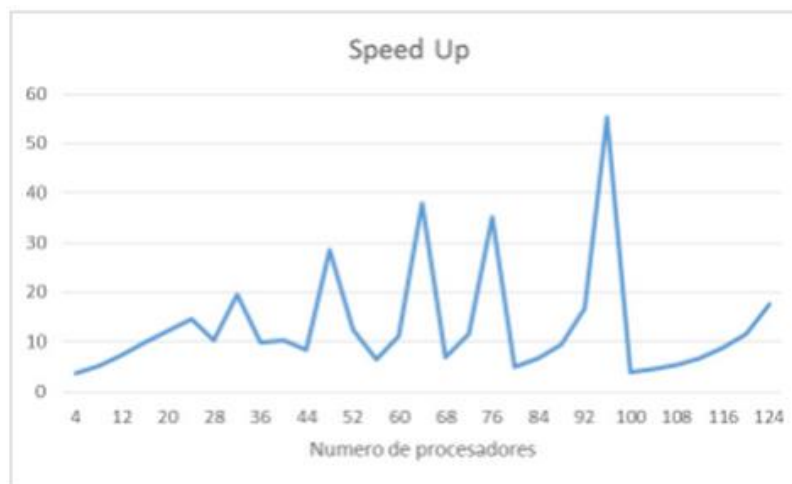


Ilustración 29: Gráfico de SpeedUp algoritmo percolación, matriz tamaño 1000. Fuente: Elaboración propia.

#### 9.4.2.3.2 Eficiencia

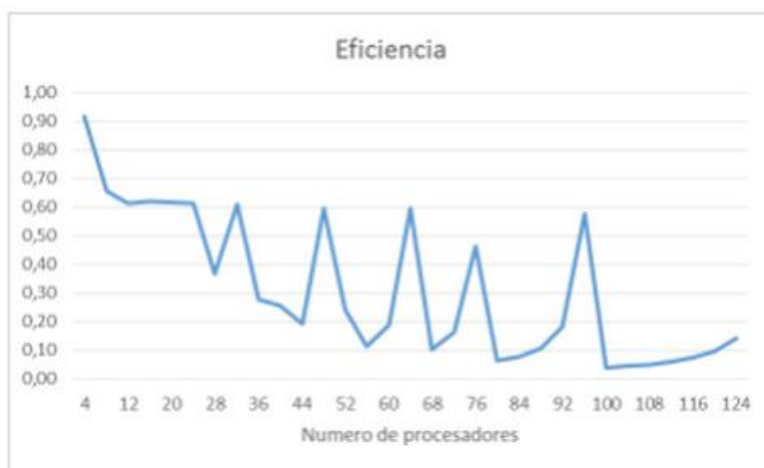


Ilustración 30: Gráfico eficiencia algoritmo percolación paralelo matriz tamaño 1000. Fuente: Elaboración propia.

#### 9.4.2.3.3 Costo



Ilustración 31: Gráfico de costo, algoritmo percolación paralelo matriz tamaño 1000. Fuente: Elaboración propia.



#### 9.4.2.4 Nodo óptimo

Tamaño matriz	Tiempo (Segundos)
100	0,253869375
500	4,827818964
1000	20,65078497
1500	48,16490698
2000	83,95333004

Tabla 4. Tiempos ejecución paralelo usando 96 procesadores. Elaboración propia.

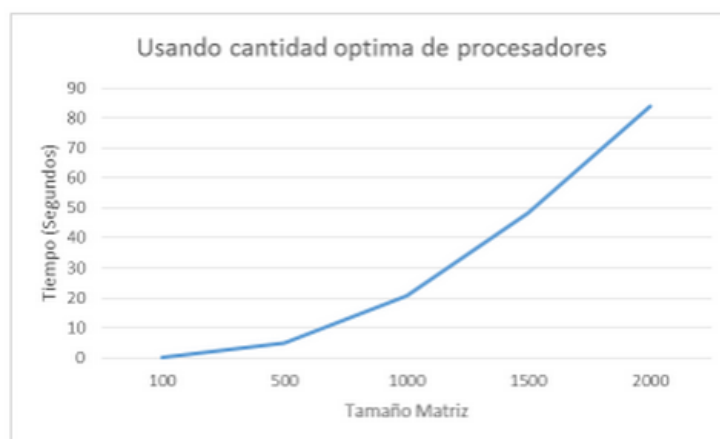


Gráfico 4. Comportamiento nodo óptimo. Elaboración propia.

Ilustración 32: Nodo óptimo algoritmo percolación. Fuente: Elaboración propia.

Realizando pruebas utilizando solo la cantidad optima de procesadores, es decir, ejecutando el programa con 96 procesadores y distintos tamaños de matrices, podemos ver que tiene un comportamiento de tipo polinómica de orden 2. Esto queda demostrado con la siguiente formula:

$$y = 5,2943X^2 - 10,692X + 5,4094$$

#### 9.4.3 FASTA

##### 9.4.3.1 Tiempo secuencial

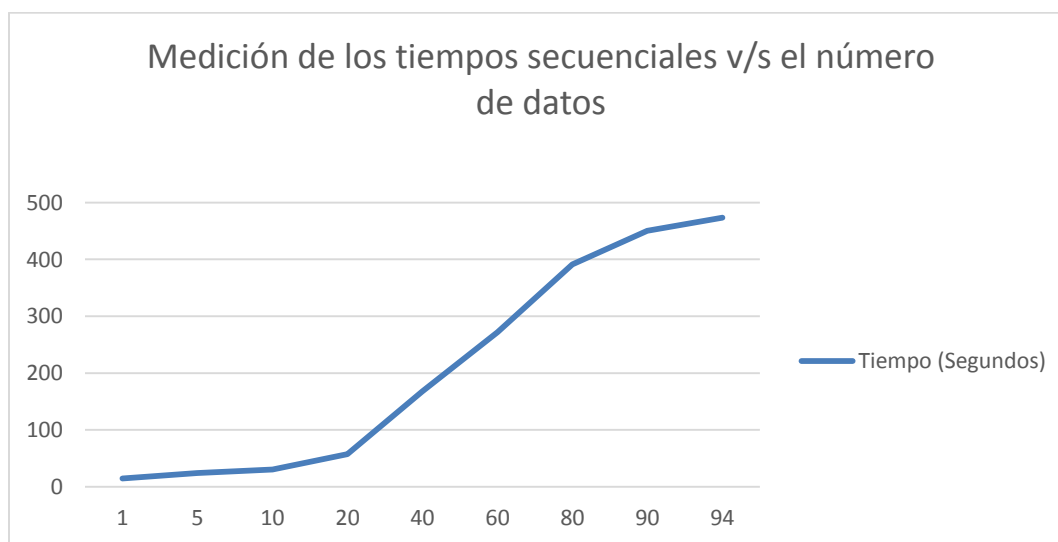


Ilustración 33: Medición de tiempos secuenciales v/s cantidad de datos algoritmo Fasta. Fuente: Elaboración propia.

### 9.4.3.2 Tiempo paralelo

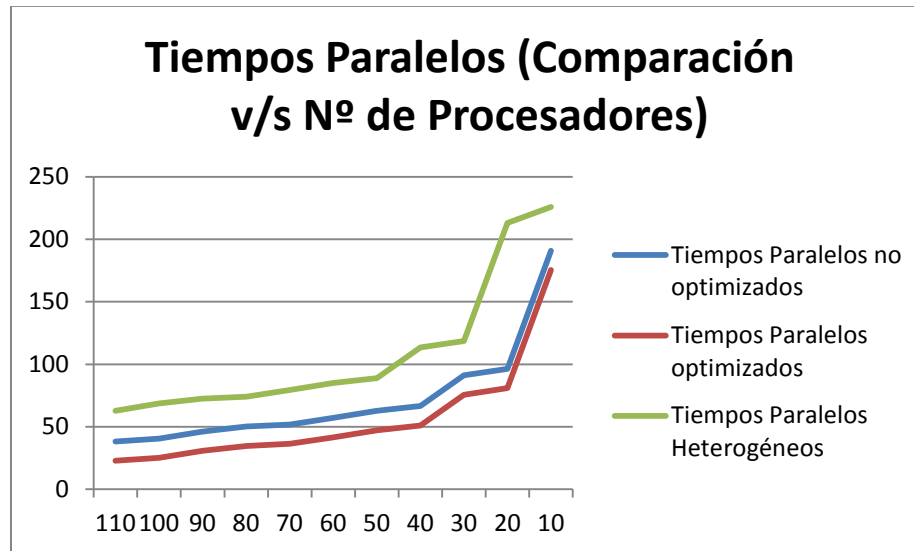


Ilustración 34: Tiempos paralelos, comparación v/s número de procesadores. Fuente: Elaboración propia.

### 9.4.3.2.1 SpeedUp

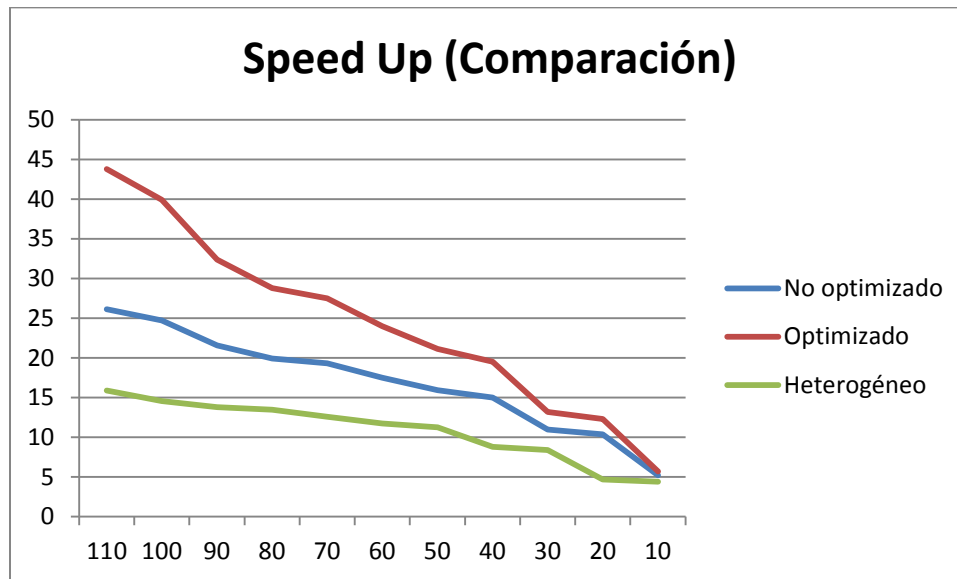


Ilustración 35: SpeedUp algoritmo Fast. Fuente: Elaboración propia.

#### 9.4.3.2.2 Eficiencia

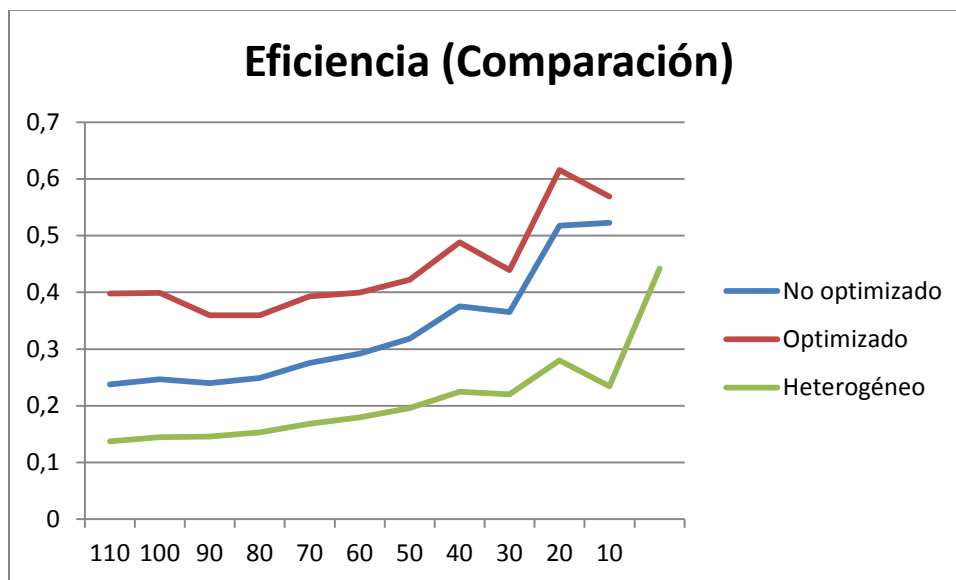


Ilustración 36: Gráfico de eficiencia algoritmo Fasta. Fuente: Elaboración propia.

#### 9.4.3.2.3 Costo

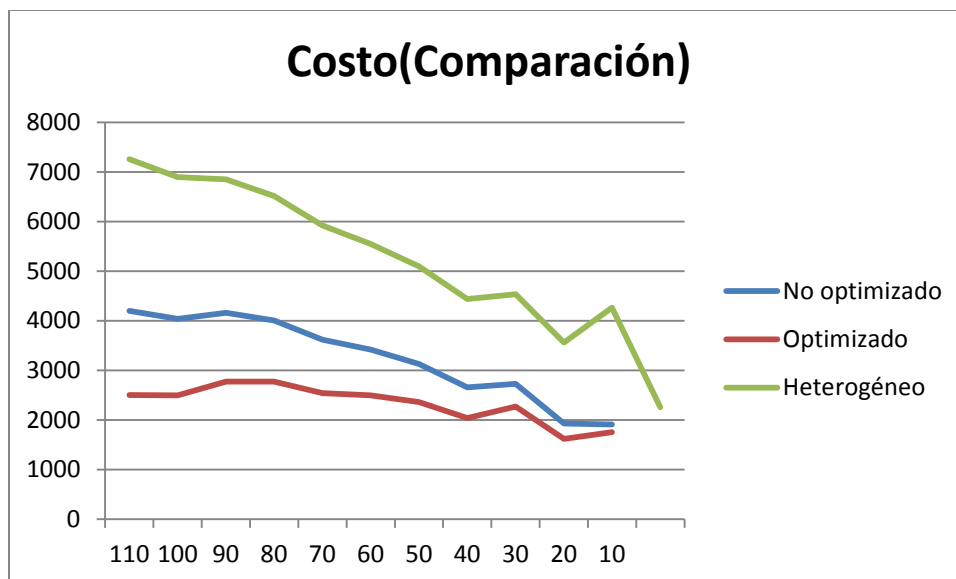


Ilustración 37: Gráfico de costo algoritmo Fasta. Fuente: Elaboración propia.

## 9.5 Conclusiones obtenidas

(Desde el punto de vista del análisis gráfico y comportamiento de las métricas)

### 9.5.1 Algoritmo Torah

- A mayor cantidad de caracteres que tenga una palabra menos probable es encontrar coincidencias del mismo.
- A mayor cantidad de palabras en una frase también disminuye la probabilidad de éxito en la búsqueda de tales patrones, pero no tanto como la diferencia de la cantidad de caracteres.
- El éxito de encontrar coincidencias está condicionado por los caracteres con los cuales está constituida las palabras. Esto es debido a que en todo el texto hay diferentes proporciones de caracteres distintos. Por ese motivo si una palabra está formada por caracteres que tienen escasa participación en el texto, este tendrá pocas probabilidades de ser localizado debido a que ni siquiera puede existir.
- Para las pruebas bajo las condiciones entregadas de largo de palabra y las frecuencias de caracteres se observó que disminuyen las cantidades de coincidencias.

### 9.5.2 Algoritmo de Percolación

- Al observar el gráfico de tiempos de ejecución en paralelo con matriz tamaño 100 se aprecia que el menor tiempo de ejecución se encuentra al utilizar 96 procesadores.
- El gráfico de SpeedUp de matriz tamaño 100 muestra que el procesador óptimo es 96, esta información explica por qué al utilizar el nodo 96 se encuentra el menor tiempo de ejecución.
- La eficiencia del gráfico de matriz tamaño 100 se encuentra al utilizar 4 procesadores, pero también es posible encontrar un alto número de eficiencia en los procesadores 48,64 y 96.
- El costo del gráfico de matriz tamaño 100 se reduce bastante a en el procesador 96, no se logra apreciar bien en el gráfico, pero al revisar la tabla, se observa que es el procesador con menor costo. Al analizar los tiempos de ejecución del gráfico con matriz tamaño 1000 de ejecución, se observa que nuevamente al usar 96 procesadores, se obtiene el menor tiempo de ejecución.
- Con unas curvas más pronunciadas, se puede observar claramente que al usar 96 procesadores es donde se obtiene mejor SpeedUp.
- Al comparar los resultados con las métricas tomadas al usar una matriz de tamaño 100, se observa que la eficiencia se encuentra en los nodos 32, 48, 64,76 y 96 en la ilustración 32.

### 9.5.3 Algoritmo FASTA

- El problema es altamente paralelizable, debido a que los tiempos paralelos en comparación a los secuenciales son considerablemente menores.
- Un clúster heterogéneo para este problema (al menos uno de las características que se usó) resultó ser perjudicial para las aspiraciones de lograr reducir los tiempos de ejecución, debido a que las características de esos procesadores eran muy inferiores.
- La optimización al código paralelo resultó de forma exitosa, debido a que redujo los tiempos en algunos casos en 15 segundos.
- La diferencia de tamaño entre las cadenas a alinear definitivamente influye en los tiempos de ejecución. A mayor diferencia, más tiempo toma alinear las secuencias.
- Otro punto que influye en los tiempos de ejecución son la cantidad de alineaciones que realiza el algoritmo para determinar el óptimo, a mayor cantidad de alineaciones, más tiempo de ejecución toma el programa.
- A nivel de programación, Python resultó ser una herramienta bastante útil para desarrollar los códigos, gracias a su simplicidad y a la cantidad de librerías disponibles para su utilización.
- Añadir más procesadores al óptimo resultó traer consecuencias, ya que los tiempos de ejecución aumentaron. Esto se debe principalmente a los tiempos de comunicación que hay entre los equipos que fueron aumentando a medida que más procesadores se conectaban a la ejecución.

## 10. Conclusiones y trabajo a futuro

En el desarrollo del Proyecto Plataforma Web de Servicios de Computación Paralela se logró implementar, mediante un clúster computacional de ambiente de proceso paralelo y secuencial, una plataforma web de interfaz intuitiva y amigable que permite al usuario final ingresar una problemática de las áreas de simulación de propagación de incendios forestales, cadenas de ADN en formato FASTA para el alineamiento de cadenas de ADN y búsqueda de patrones mediante Torah.

El análisis de los recursos disponibles del Laboratorio 5 de Informática fue fructífero, pues se pudo determinar la mejor alternativa para la destinación de los implementos del laboratorio para el desarrollo de la actividad, descubriendo que el mejor recurso es 4 veces mejor que el peor recurso.

La configuración del entorno de desarrollo del proyecto en los aspectos de paralelización (MPI), servidor de dominio (DNS) y servidor de correo (MailGun) fue exitosa, siendo el acceso al dominio <http://00-ironman.clustermarvel.utem/webParalela/> la cual es la interfaz gráfica del entorno de paralelización configurado y entrega la respuesta mediante correo.

Las investigaciones perseguidas para el correcto desarrollo del laboratorio en las áreas de biología, física y data mining y desenvolverse en problemas específicos de estos, aportaron al grupo el conocimiento necesario el primer paso que involucra el progreso del proyecto en marcha como tal.

## 11. Anexos

### Anexo A: Manual de instalación

#### Plataforma Base

Se trabajará con Ubuntu 14.04 y por tanto, se debe cerciorar de las características de los equipos disponibles para la realización del proyecto, así se puede instalar el sistema operativo con la arquitectura adecuada (x64 x86 bits).

#### Instalación MPI

##### Nodo Maestro

##### #Paso 1

```
sudo apt-get update  
sudo apt-get upgrade -y
```

##### #Utilidades

```
sudo apt-get install nautilus-open-terminal -y  
sudo apt-get install synaptic -y  
sudo apt-get install chromium-browser -y
```

##### #Necesarios

```
sudo apt-get install openmpi-bin -y  
sudo apt-get install openmpi-common -y  
sudo apt-get install libopenmpi1.6 -y  
sudo apt-get install libopenmpi-dev -y  
sudo apt-get install ssh -y  
sudo apt-get install openssh-client -y  
sudo apt-get install nfs-kernel-server -y  
sudo apt-get install nfs-common -y  
sudo apt-get install remotetea -y  
sudo apt-get install python-mpi -y  
sudo apt-get install python-mpi4py-dbg -y  
sudo apt-get install python3-mpi4py-dbg -y  
sudo apt-get install python3-mpi4py -y  
sudo apt-get install python-mpi4py -y  
sudo apt-get install python-mpi4py-doc -y  
sudo apt-get install libpython-all-dev -y  
sudo apt-get install libpython3-all-dev -y  
sudo apt-get install build-essential -y
```

## #Paso 2

#Carpeta ssh con los permisos

```
cd /home/paralela
```

```
mkdir .ssh
```

```
sudo chmod 777 .ssh
```

#Carpeta compartida de procesamiento

```
sudo mkdir /mpi
```

```
sudo chmod 777 /mpi
```

#Cambiar usuario mpi

```
cd /mpi
```

```
sudo chown paralela /mpi
```

## #Paso 3

#Editar Fichero hosts

```
sudo rm /etc/hosts
```

```
sudo touch /etc/hosts
```

```
sudo chown paralela /etc/hosts
```

```
sudo echo '127.0.0.1 localhost'>/etc/hosts
```

```
sudo echo '10.1.10.113 paralela
sudo echo '10.1.10.114 paralela
sudo echo '10.1.10.115 paralela
sudo echo '10.1.10.116 paralela
sudo echo '10.1.10.117 paralela
sudo echo '10.1.10.118 paralela
sudo echo '10.1.10.119 paralela
sudo echo '10.1.10.120 paralela
sudo echo '10.1.10.121 paralela
sudo echo '10.1.10.122 paralela
sudo echo '10.1.10.123 paralela
sudo echo '10.1.10.124 paralela
sudo echo '10.1.10.125 paralela
sudo echo '10.1.10.126 paralela
sudo echo '10.1.10.127 paralela
sudo echo '10.1.10.128 paralela
sudo echo '10.1.10.129 paralela
```

```
00-IronMan'>>/etc/hosts
01-capitanamerica'>>/etc/hosts
02-Thor'>>/etc/hosts
03-hulk'>>/etc/hosts
04-Vision'>>/etc/hosts
05-StarLord'>>/etc/hosts
06-NickFury'>>/etc/hosts
07-BlackWidow'>>/etc/hosts
08-Groot'>>/etc/hosts
09-Wolverine'>>/etc/hosts
10-Storm'>>/etc/hosts
11-Xavier'>>/etc/hosts
12-Magneto'>>/etc/hosts
13-Daredevil'>>/etc/hosts
14-Loki'>>/etc/hosts
15-Gambito'>>/etc/hosts
16-Venom'>>/etc/hosts
```



```
sudo echo '::1 ip6-localhost ip6-loopback'>>/etc/hosts
sudo echo 'fe00::0 ip6-localnet'>>/etc/hosts
sudo echo 'ff00::0 ip6-mcastprefix'>>/etc/hosts
sudo echo 'ff02::1 ip6-allnodes'>>/etc/hosts
sudo echo 'ff02::2 ip6-allrouters'>>/etc/hosts
```

```
#Editar Fichero Exports
sudo chmod 777 /etc/exports
sudo echo '/mpi *(rw,sync)'>>/etc/exports
```

#### #Paso 4

```
sudo service nfs-kernel-server restart
```

```
#esperar a que todos los esclavos monten la carpeta maestra
```

#### #Paso 5

```
ssh-keygen -t rsa
#Darle ENTER a todo
#posicionarse en la carpeta .ssh
cd /home/paralela/.ssh

#generar la clave de autorización
cat id_rsa.pub>>authorized_keys
#Copiar clave publica en cada esclavo e ingresar la clave correspondiente cada esclavo
ssh-copy-id 10.1.10.114
ssh-copy-id 10.1.10.115
ssh-copy-id 10.1.10.116

#Asi sucesivamente
#Confirme que el proceso este correcto

ssh 10.1.10.114

#No debiese pedir contraseña para ingresar
```

## #Paso 6

#Crear fichero hostfile

```
sudo touch  
sudo chown paralela /home/hostfile  
sudo echo '10.1.10.113 slots=4'>>/etc/hosts  
sudo echo '10.1.10.114 slots=4'>>/etc/hosts  
sudo echo '10.1.10.115 slots=4'>>/etc/hosts  
sudo echo '10.1.10.116 slots=4'>>/etc/hosts  
sudo echo '10.1.10.117 slots=4'>>/etc/hosts  
sudo echo '10.1.10.118 slots=4'>>/etc/hosts  
sudo echo '10.1.10.119 slots=4'>>/etc/hosts  
sudo echo '10.1.10.120 slots=4'>>/etc/hosts  
sudo echo '10.1.10.121 slots=4'>>/etc/hosts  
sudo echo '10.1.10.122 slots=4'>>/etc/hosts  
sudo echo '10.1.10.123 slots=4'>>/etc/hosts  
sudo echo '10.1.10.124 slots=4'>>/etc/hosts  
sudo echo '10.1.10.125 slots=4'>>/etc/hosts  
sudo echo '10.1.10.126 slots=4'>>/etc/hosts  
sudo echo '10.1.10.127 slots=4'>>/etc/hosts  
sudo echo '10.1.10.128 slots=4'>>/etc/hosts  
sudo echo '10.1.10.129 slots=4'>>/etc/hosts
```

## #Paso 7

#Nos posicionamos en la carpeta maestra

```
cd /mpi
```

#Final mente probamos el código

```
mpirun -np 80--hostfile /home/hostfile python prueba.py
```

#El 80 sería el número de procesadores q usaremos y obvio q la ejecución del programa dependerá del

#lenguaje en el que se trabaje

## Nodo Esclavo

### #Paso 1

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install nautilus-open-terminal -y
sudo apt-get install checkinstall -y
sudo apt-get install synaptic -y
sudo apt-get install chromium-browser -y
sudo apt-get install openmpi-bin -y
sudo apt-get install openmpi-common -y
sudo apt-get install libopenmpi1.6 -y
sudo apt-get install libopenmpi-dev -y
sudo apt-get install ssh -y
sudo apt-get install openssh-client -y
sudo apt-get install nfs-kernel-server -y
sudo apt-get install nfs-common -y
sudo apt-get install remotetea -y
sudo apt-get install python-mpi -y
sudo apt-get install python-mpi4py-dbg -y
sudo apt-get install python3-mpi4py-dbg -y
sudo apt-get install python3-mpi4py -y
sudo apt-get install python-mpi4py -y
sudo apt-get install python-mpi4py-doc -y
sudo apt-get install libpython-all-dev -y
sudo apt-get install libpython3-all-dev -y
```

### #Paso 2

#Carpeta ssh con los permisos

```
cd /home/paralela
mkdir .ssh
sudo chmod 700 .ssh
```

#Carpeta compartida de procesamiento

```
cd ../../
sudo mkdir /mpi
sudo chmod 777 /mpi
```

#Cambiar usuario mpi

```
cd /mpi
sudo chown paralela /mpi
```

### #Paso 3

#Montar carpeta maestra ....La IP debe ser del maestro

```
sudo mount 10.1.10.113:/mpi /mpi
```

#Montar automático, continuar aquí!..... La IP debe ser del maestro

```
sudo chmod 777 /etc/fstab
```

```
sudo echo '10.1.10.113:/mpi /mpi nfs'>> /etc/fstab
```

```
sudo mount -a
```

### Configuración DNS

1. Actualización de la información de los repositorios con el siguiente comando:

```
sudo apt-get update
```

2. Instalación del servidor DNS Bind9:

```
sudo apt-get install bind9
```

3. Creación de una copia de seguridad del archivo a modificar:

```
sudo cp /etc/bind/named.conf.local{,.original}
```

4. Editar el archivo */etc/bind/named.conf.local* con el siguiente comando:

```
sudo nano /etc/bind/named.conf.local
```

y se añade siguiente contenido:

```
zone "clustermarvel.utem" {  
    type master;  
    file "db.clustermarvel.utem";  
};
```

```
zone "10.1.10.in-addr.arpa" {  
    type master;  
    file "db.10.1.10";  
};
```

La 1era IP se escribe inversa del 1er al 3er n° dejando afuera el 4to valor.

La 2da IP se escribe como corresponde del 1er al 3er n° dejando afuera el 4to valor.

5. Para comprobar la sintaxis de los archivos de configuración:

```
named-checkconf
```

Si no aparece nada, la sintaxis de los archivos de configuración es correcta. Eso no significa que no haya ningún error, sólo que no hay errores de sintaxis.

6. Creación del archivo `/var/cache/bind/db.clustermarvel.utem`:

```
sudo nano /var/cache/bind/db.clustermarvel.utem
```

y se incluye el siguiente contenido:

```
$ORIGIN clustermarvel.utem.
```

```
$TTL 86400 ; 1 dia
```

```
@      IN      SOA    00-IronMan    postmaster (
        1      ; serie
        6H     ; refresco (6 horas)
        1H     ; reintentos (1 hora)
        2W     ; expira (2 semanas)
        3H     ; mínimo (3 horas)
)
        NS     00-IronMan
00-IronMan A    10.1.10.113
```

7. Se comprueba la zona a crear (`clustermarvel.utem`):

```
named-checkzone clustermarvel.utem /var/cache/bind/db.clustermarvel.utem
```

En esta ocasión siempre aparecerá una salida, ya sea para indicar que todo está bien (OK) o algún error.

```
paralela@00-IronMan:~$ named-checkzone clustermarvel.utem /var/cache/bind/db.clustermarvel.utem
zone clustermarvel.utem/IN: loaded serial 1
OK
```

8. A continuación se crea el archivo `/var/cache/bind/db.10.1.10` para la zona inversa:

```
sudo nano /var/cache/bind/db.10.1.10
```

y se incluye el siguiente contenido:

```
$ORIGIN 10.1.10.in-addr.arpa.
```

```
$TTL 86400 ; 1 dia
```

```
@      IN      SOA    00-IronMan    postmaster (
        1      ; serie
        6H     ; refresco (6 horas)
        1H     ; reintentos (1 hora)
        2W     ; expire (2 semanas)
        3H     ; mínimo (3 horas)
)
        NS     00-Ironman.clustermarvel.utem.
113     PTR    00-Ironman.clustermarvel.utem.
```

El número **113** se corresponde con el último dígito de la dirección IP del servidor.

9. Se comprueba la zona inversa recién creada:

named-checkzone 10.1.10.in-addr.arpa /var/cache/bind/db.10.1.10

Al igual que antes, se obtiene un mensaje para indicar tanto si la zona es correcta como si no lo es.

```
paralela@00-IronMan:~$ named-checkzone 10.1.10.in-addr.arpa /var/cache/bind/db.10.1.10
zone 10.1.10.in-addr.arpa/IN: loaded serial 1
OK
```

10. Se reinicia el servicio:

```
sudo service bind9 restart
```

Si todo va bien, se verá que está OK.

```
paralela@00-IronMan:~$ sudo service bind9 restart
* Stopping domain name service... bind9 [OK]
* Starting domain name service... bind9 [OK]
```

11. Revisar el log para comprobar que todo ha ido bien. Aunque se puede hacer con el comando tail, se prefirió less porque permite ver todo el contenido del mismo:

```
less /var/log/syslog
```

El resultado se puede ver en la siguiente captura:

```
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 255.255.255.IN-ADDR.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: D.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 8.E.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 9.E.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: A.E.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: B.E.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: command channel listening on 127.0.0.1#953
Apr 20 16:30:19 servidor named[3403]: command channel listening on ::1#953
Apr 20 16:30:19 servidor named[3403]: zone 0.in-addr.arpa/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone 127.in-addr.arpa/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone 10.1.10.in-addr.arpa/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone 255.in-addr.arpa/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone clustermarvel.utem/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone localhost/IN: loaded serial 2
Apr 20 16:30:19 servidor named[3403]: running
Apr 20 16:30:19 servidor named[3403]: zone 10.1.10.in-addr.arpa/IN: sending notifications (serial 1)
```

Para salir se pulsa la tecla **q**.

12. Editar el archivo `/etc/resolv.conf` para que servidor resuelva las peticiones DNS:

`sudo nano /etc/resolv.conf`

Se cambia el primero de los servidores DNS por la IP propia:

`nameserver 10.1.10.113`

`nameserver 8.8.8.8`

13. Se prueba el servidor de nombres:

`dig clustermarvel.utem`

La respuesta será muy parecida a la siguiente:

```
paralela@00-IronMan :~$ dig clustermarvel.utem
; <<> DiG 9.7.0-P1 <<> clustermarvel.utem
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30475
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
; clustermarvel.utem.          IN      A

;; AUTHORITY SECTION:
clustermarvel.utem.          10800   IN      SOA     00-IronMan.clustermarvel.utem. postmaster.clustermarvel.utem.
1 21600 3600 1209600 10800

;; Query time: 1 msec
;; SERVER: 192.168.2.1#53(192.168.2.1)
;; WHEN: Tue Apr 20 16:32:22 2010
;; MSG SIZE rcvd: 90
```

14. Se prueba la resolución inversa:

`dig -x 10.1.10.113`

Sería la salida esperada del comando anterior:

```
paralela@00-IronMan:~$ dig -x 10.1.10.113
; <<> DiG 9.7.0-P1 <<> -x 10.1.10.113
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 10314
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;          .in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
113.10.1.10.in-addr.arpa. 86400 IN      PTR      00-IronMan.clustermarvel.utem.

;; AUTHORITY SECTION:
113.10.1.10.in-addr.arpa. 86400 IN      NS       00-IronMan.clustermarvel.utem.

;; ADDITIONAL SECTION:
servidor.sliceoflinux.lan. 86400 IN      A        10.1.10.113

;; Query time: 1 msec
;; SERVER: 10.1.10.113#53(192.168.2.1)
;; WHEN: Tue Apr 20 16:32:50 2010
;; MSG SIZE rcvd: 111
```

15. Por último, para poder sacarle partido al servidor que se ha creado, faltaría incluir este servidor DNS en la configuración de los clientes.

## LAMP Server mediante Tasksel

LAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Linux, el sistema operativo
- Apache, el servidor web;
- MySQL/MariaDB, el gestor de bases de datos;
- Perl, PHP, o Python, los lenguajes de programación.

La combinación de estas tecnologías es usada principalmente para definir la infraestructura de un servidor web, utilizando un paradigma de programación para el desarrollo.

Taskel es un sistema de instalación que forma parte del instalador de Debian y que además está incluido en Ubuntu. Taskel agrupa los paquetes de software por tareas y ofrece un modo sencillo de instalar todos los paquetes que son necesarios para dicha tarea. Proporciona la misma funcionalidad que los convencionales meta-paquetes.



### Primero instalar Tasksel

```
sudo apt-get install tasksel
```

### Segundo instalar Lamp Server

```
sudo tasksel install lamp-server
```

## PhpMyAdmin

Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 62 idiomas.

Instalación y configuración de PhpMyAdmin.

```
sudo apt-get install phpmyadmin
```

Transcurridos unos momentos de descarga de paquetes mostrará la ventana de configuración de phpmyadmin. Donde se tienen los siguientes pasos:

- 1.- ¿Desea configurar la base de datos para phpmyadmin con dbconfig-common? Responder “si”.
- 2.- Se ingresa la contraseña de root de mysql, se suministra la misma que se dio en el momento que configurar mysql.
- 3.- Proporcionar una contraseña para el administrador de phpmyadmin y confirmar esta.
- 4.- Elegir el servidor web que a utilizar, seleccionar Apache.
- 5.- Probar, ejecutar primero mysql y apache:

```
sudo service mysql start  
sudo /etc/init.d/apache2 start
```

- 6.- Editar la configuración de apache, sólo en caso de que salga NO FOUND en el paso anterior.

```
sudo nano /etc/apache2/apache2.conf
```

Incluyendo la siguiente línea al final del archivo

```
/etc/phpmyadmin/apache.conf
```

- 7.- Ingresar a la página de login de PhpMyAdmin

```
http://localhost/phpmyadmin/
```

Acceder con usuario y contraseña.

## Montar Pagina Web

Primero cambiamos los permisos de la carpeta /var/www/html

```
sudo chmod 775 -R /var/www/html
```

Luego copiamos la carpeta de la pagina web en la carpeta /var/www/html

Una vez esto esté listo cambiamos el usuario (apache) y damos permisos a la carpeta.

```
sudo chmod 775 -R /var/www/html
```

```
sudo chown www-data -R /var/www/html
```

## Montar Códigos Paralelos

Copiar las carpetas con los códigos a ejecutarse en la carpeta que está designada para la paralización.

```
/mpi
```

## Anexo B: Conexión Remota (Túnel SSH)

La conexión remota es poder ingresar a una red local desde cualquier lugar donde se tenga conexión a internet, contando con algunos requisitos previos. Para ello, en la red local del clúster se necesitaron algunos programas y protocolos para poder hacer realidad la conexión remota, cuales se detallan a continuación.

1. Placa WOL y programa wakeonlan
2. Protocolo de seguridad ssh
3. Servidor con ip pública

### 1.- Placa WOL y programa wakeonlan

La placa WOL es un hardware que permite encender un computador de manera lógica, sin la necesidad de apretar el botón de encendido. Tal placa es un dispositivo que envía señales eléctricas al computador para lograr encenderlo, entregando un denominado “paquete mágico” con varias señales repetitivas para lograr el efecto. En la actualidad, la mayoría de las tarjetas de red, al igual que las que están integradas en las placas bases, incluyen la placa WOL, para disminuir la preocupación cuando se empezó a masificar tal función.

En linux, el programa que arranca esta función se llama WakeOnLan y su instalación es sencilla, *sudo apt-get install wakeonlan*, y su función en términos físicos es, enviar desde algún computador encendido el paquete mágico a un computador apagado, para encenderlo.

### 2.- Protocolo de seguridad ssh

Secure SHell o interprete de ordenes seguras es un protocolo que sirve para acceder a máquinas de manera remota mientras estén en una red en común, que permite manejar por completo un computador tal como estando en la consola de comandos de la misma máquina.

Además de ingresar al computador de manera remota, nos permite copiar archivos de un lado a otro sin mayores complicaciones. Todo con un nivel seguridad alto, con llaves públicas y privadas para encriptar los mensajes.

### 3.- Servidor con IP pública

Para poder conectarse por ssh y enviar el paquete mágico, es necesario tener a los 2 computadores (local y remoto) conectados en la misma red. Para eso se necesita un servidor con IP pública, teniendolo conectado a internet para que cualquier computador que tenga acceso a navegar pueda conectarse mediante ssh al servidor.

Una vez conectado en el servidor, y gracias a interfaces de red virtuales de los servidores, es posible “saltar” al computador que se necesita y está conectado en la misma red que el servidor.

¿Cómo acceder al clúster de manera remota?

Desde un computador que tenga conexión a internet se debe escribir por la consola de comandos (en Unix “konsole o terminal” y en windows “putty”) lo siguiente:

```
ssh nombre-usuario@ip-servidor  
clave de usuario
```

Una vez hecho esto, estamos dentro del servidor, se debe enviar el paquete mágico para encender los computadores que estaremos usando, por la consola lo siguiente

```
wakeonlan mac-cluster
```

Con eso se envía el paquete mágico y el o los computadores se encienden. Esperando un poco se puede ingresar por ssh nuevamente a los computadores que se solicita.

```
ssh nombre-usuario@ip-servidor  
clave de usuario
```

Con esto se puede acceder a los computadores que se necesitan para realizar las tareas pertinentes.

## Anexo C: MailGun

El envío de correos electrónicos es una funcionalidad que no puede faltar en una aplicación web, ya sea para confirmar el registro de un nuevo usuario, enviar un boletín de noticias o notificar sobre alguna transacción en la plataforma. En la actualidad existen muchas plataformas especializadas en realizar estas actividades que permiten manejar grandes campañas de email marketing, ofreciendo ciertas métricas de gran utilidad. En este contexto, Mailgun es un servicio de email para desarrolladores que cuenta con una poderosa Api que facilita su uso y acceso desde cualquier plataforma de desarrollo.

Mailgun es un servicio de correo enfocado para los desarrolladores, que mediante el uso de la API que nos ofrece pueden desarrollar aplicaciones que gestionen correo electrónico. Además desde su compra por parte de Rackspace está teniendo mucho protagonismo en las infraestructuras de cloud computing. En este artículo he presentado las características que yo he usado para mis necesidades, pero evidentemente el servicio nos ofrece muchas más funcionalidades que puedes aprender en su documentación oficial.

Documentación Oficial:

<https://documentation.mailgun.com/>



## Aspectos de Implementación

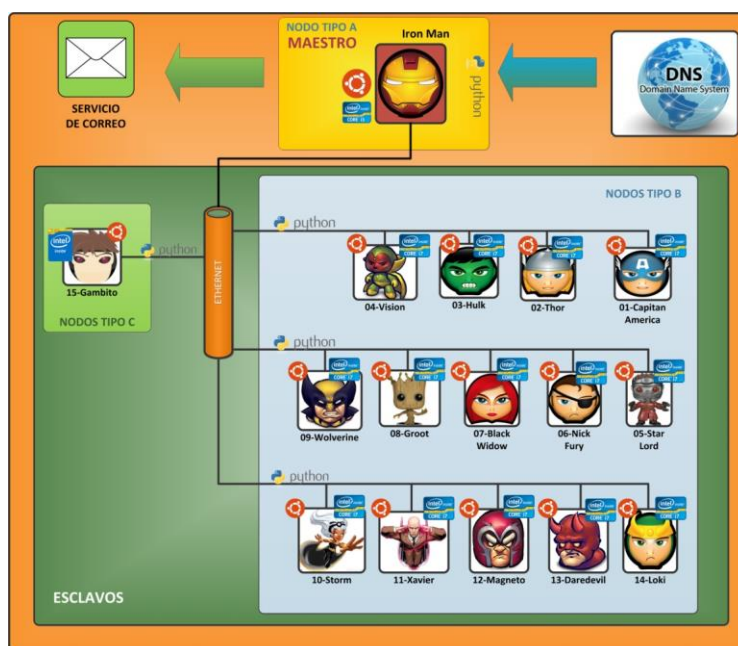
Para hacer una prueba general basta con ejecutar por consola el siguiente comando:

```
curl -s --user 'api:key-3ax6xnjp29jd6fds4gc373sgvjxteol0' \
  https://api.mailgun.net/v2/samples.mailgun.org/messages \
  -F from='Excited User <excited@samples.mailgun.org>' \
  -F to='devs@mailgun.net' \
  -F subject='Hello' \
  -F text='Testing some Mailgun awesomeness!'
```



## Anexo D: Modelo de la plataforma

El esquema a continuación describe los principales componentes del clúster. En él se detallan de manera gráfica la asignación de nombres a los equipos, el rol que cumplen y el tipo de hardware (tipo A, tipo B y tipo C).



Tipo A: Maestro: Procesador i3-3220cpu 3.30GHz x4  
Memoria 3.8Gb  
Grafico Intel Integrado  
S.O. Ubuntu 14.04 x64

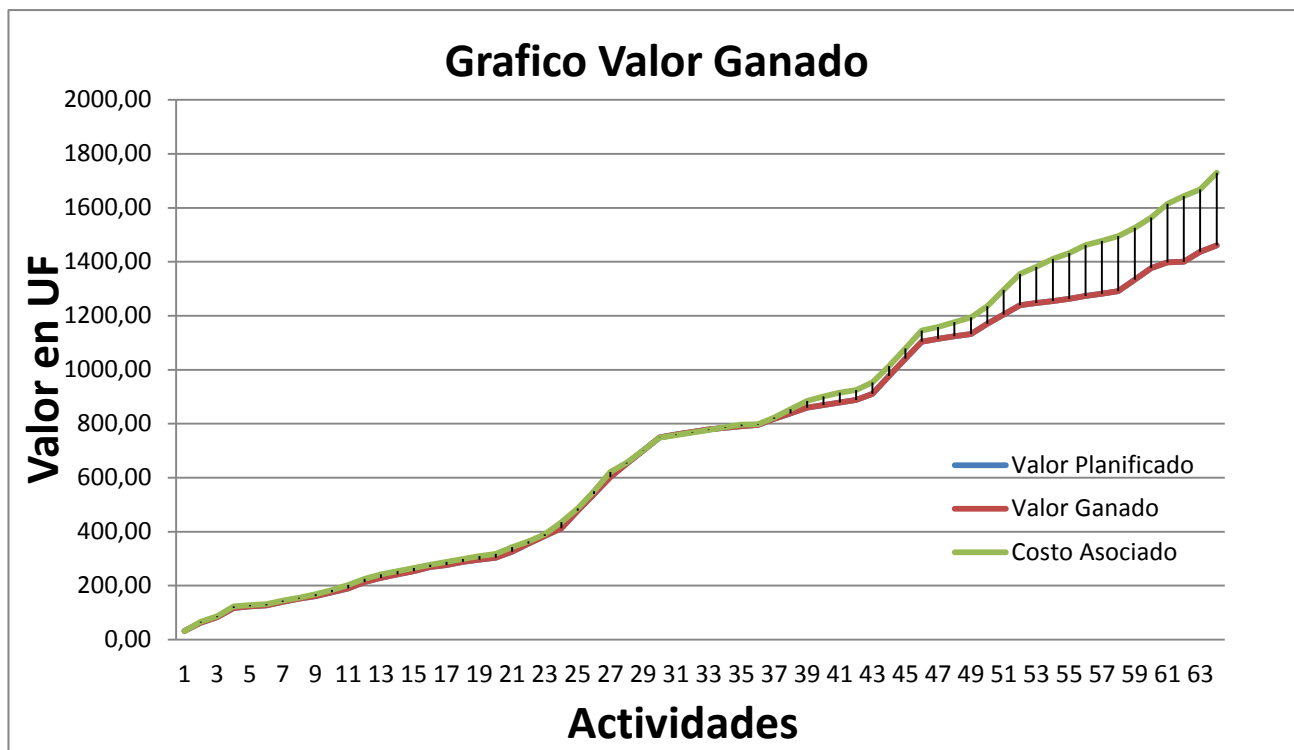
Tipo B: Esclavos: Procesador i7-3730cpu 3.40GHz x8  
| Memoria 7.7Gb  
Grafico Intel integrada  
S.O. Ubuntu 14.04 x64

Tipo C: Esclavos: Procesador Pentium 4 Cpu 3.00GHz x2  
Memoria 2.0Gb  
Grafico intel 945G x86/MMX/SSE2  
S.O. Ubuntu 14.04 x86

## Anexo E: Carta Gantt

## Anexo F: Valor Ganado

A continuación se muestra el gráfico de valor ganado al término del proyecto:



## Interpretación Valor Ganado

A continuación, se muestra una tabla los indicadores que nos dan un diagnóstico de nuestro proyecto al presente. Estos indicadores son:

- Porcentaje de variación del costo(%CV)
- Porcentaje de variación del cronograma(%SV)
- Índice de desempeño del programa(SPI)
- Índice de desempeño del costo(CPI)
- Índice de tiempo-costos(CSI)

CV	SV	%CV	%SV	SPI	CPI	CSI
-269,45	0	-18,416	0	1	0,8444759	0,8444759

Se puede obtener de esta tabla, que se sobrepasó del presupuesto en un 18,42%, pero se logra terminar el proyecto dentro de los plazos establecidos al comienzo de este. Para apoyar esta conclusión, se muestra los índices de desempeño del programa el cual es 1(exactamente lo planeado) y el Índice de desempeño del costo el cual es 0,844, que indica que efectivamente, se tuvo que incurrir en mayor trabajo y mayores costos con el fin de terminar el proyecto dentro de los plazos ideados al comienzo.



## Anexo G: Evaluaciones

### Evaluaciones por Etapa

Estas evaluaciones fueron realizadas por cada jefe de grupo, para su grupo incluyendo una autoevaluación, para luego ser revisados por el Jefe de Proyecto.

Las evaluaciones correspondientes a cada avance se realizaron considerando 13 puntos y la cuarta evaluación entregada por el Jefe de Proyecto considera 13 puntos, que deben ser evaluados en una escala de 1 a 5, cada uno de estos puntos tiene una ponderación específica.

La nota se obtendrá a partir del puntaje total en una escala del 60%.

### Calificación Final del Proyecto

Esta nota es la sumatoria de las evaluaciones obtenidas de las etapas anteriormente mencionadas por una ponderación o peso establecido. La ponderación de las primeras tres calificaciones consiste en el valor porcentual de avance establecido en la planificación menos un 5%.

A lo anterior se le suma la cuarta calificación por la ponderación restante en la planificación mas el 15% reservado de las tres primeras notas.

Y finalmente la sumatoria total se divide por 100, lo que resultara en la Calificación Final del Proyecto.

**Ejemplo:**  $\text{nota1} \times (21.62 - 5) + \text{nota2} \times ((62.71 - 21.62) - 5) + \text{nota3} \times ((95.75 - 62.71) - 5) + \text{nota4} \times ((100 - 95.75) + (5 + 5 + 5)) = \text{Calificación Final}$ .

A continuación se encuentra el detalle de las evaluaciones de todos los integrantes del grupo de desarrollo. Dichas evaluaciones son cuatro, tres correspondientes a cada etapa de avance y una que corresponde a la evaluación final entregada por el Jefe de Proyecto.

### Notas Primera Etapa

Área a evaluar	Factor	Rosa González	Claudio Piña	Sergio Abarca	Rodrigo Aratía	Daniel Gutiérrez	Andrés Guerrero	Manuel Venegas	Miguel Núñez	Rodrigo Reyes	Roberto Oñate	Diego Hernández
1 Realiza un trabajo ordenado.	3	5	4	5	4	4	4	4	3	0	4	3
2 Realiza la tarea asignada en el periodo establecido	5	5	5	5	4	4	4	5	5	0	3	3
3 Cumple con los objetivos de la tarea asignada.	5	5	5	5	4	4	4	5	4	0	5	5
4 Es proactivo a la hora de realizar tareas.	3	3	4	5	5	4	3	5	2	0	5	5
5 Capacidad de trabajo independiente, es decir, sin supervisión.	4	5	5	5	4	3	5	5	3	0	5	5
6 Reacciona adecuadamente ante situaciones conflictivas.	2	5	5	3	5	5	5	4	4	0	3	4
7 Mantiene a su encargado informado del progreso en la tarea asignada.	3	5	5	5	5	5	3	4	5	0	5	5
8 Asume con responsabilidad las consecuencias negativas de sus acciones, demostrando compromisos y deseos de mejorar.	3	5	5	5	4	4	5	5	3	0	3	5
9 Maneja los objetivos y las tareas que realizan los demás .	3	4	3	5	3	4	5	4	3	0	5	5
10 Colabora con compañeros en sus tiempos libres.	5	4	2	5	4	3	4	5	3	0	5	4
11 Colabora con su compañero de tareas armoniosamente sin consideración a la raza, religión, origen nacional, sexo, edad, etc.	1	5	5	5	5	5	5	5	5	0	3	4
12 Respeta las decisiones de su encargado.	2	5	5	5	5	5	5	5	5	0	5	5
13 Asiste a todas las reuniones.	3	5	3	5	5	5	5	5	5	0	5	4
<b>Puntaje Total</b>	-	196	177	206	179	170	180	199	158	0	185	183
<b>Nota</b>	-	6,6	6,1	6,9	6,1	5,9	6,1	6,7	5,5	1,0	6,3	6,2

## Notas Segunda Etapa

Área a evaluar	Factor	Rosa González	Claudio Piña	Sergio Abarca	Rodrigo Arratia	Daniel Gutiérrez	Andrés Guerrero	Manuel Venegas	Miguel Núñez	Rodrigo Reyes	Roberto Oñate	Diego Hernandez
1 Realiza un trabajo ordenado.	3	5	5	5	5	5	5	4	5	4	5	4
2 Realiza la tarea asignada en el periodo establecido	5	5	5	5	5	5	5	4	4	3	5	3
3 Cumple con los objetivos de la tarea asignada.	5	5	5	5	5	5	4	5	5	3	5	4
4 Es proactivo a la hora de realizar tareas.	3	5	5	5	4	4	5	5	4	4	5	5
5 Capacidad de trabajo independiente, es decir, sin supervisión.	4	5	5	5	4	4	5	5	4	2	5	5
6 Reacciona adecuadamente ante situaciones conflictivas.	2	3	4	4	5	5	5	3	4	5	2	5
7 Mantiene a su encargado informado del progreso en la tarea asignada.	3	5	5	4	5	5	4	4	4	5	5	5
8 Asume con responsabilidad las consecuencias negativas de sus acciones, demostrando compromisos y deseos de mejorar.	3	5	4	5	4	4	5	5	4	5	4	5
9 Maneja los objetivos y las tareas que realizan los demás .	3	3	3	4	4	4	5	4	4	1	5	5
10 Colabora con compañeros en sus tiempos libres.	5	2	4	3	4	4	4	5	4	2	3	5
11 Colabora con su compañero de tareas armoniosamente sin consideración a la raza, religión, origen nacional, sexo, edad, etc.	1	5	5	5	5	5	5	4	5	5	3	5
12 Respeta las decisiones de su encargado.	2	5	5	5	5	5	5	5	5	5	5	5
13 Asiste a todas las reuniones.	3	5	3	5	5	5	4	5	5	2	5	5
Puntaje Total	-	185	188	192	192	192	194	191	182	136	189	192
Nota	-	6,3	6,4	6,5	6,5	6,5	6,5	6,5	6,2	4,9	6,4	6,5

## Notas Tercera Etapa

Área a evaluar	Factor	Rosa González	Claudio Piña	Sergio Abarca	Rodrigo Arratia	Daniel Gutiérrez	Andrés Guerrero	Manuel Venegas	Miguel Núñez	Rodrigo Reyes	Roberto Oñate	Diego Hernandez
1 Realiza un trabajo ordenado.	3	5	5	5	5	5	5	5	5	3	5	5
2 Realiza la tarea asignada en el periodo establecido	5	5	5	5	3	3	1	4	4	3	3	4
3 Cumple con los objetivos de la tarea asignada.	5	5	5	5	3	3	1	5	5	4	5	4
4 Es proactivo a la hora de realizar tareas.	3	3	3	3	4	4	5	5	5	4	5	4
5 Capacidad de trabajo independiente, es decir, sin supervisión.	4	5	5	5	4	4	5	5	5	5	5	4
6 Reacciona adecuadamente ante situaciones conflictivas.	2	4	5	5	5	5	5	3	4	5	5	5
7 Mantiene a su encargado informado del progreso en la tarea asignada.	3	5	5	4	5	5	5	4	5	3	5	5
8 Asume con responsabilidad las consecuencias negativas de sus acciones, demostrando compromisos y deseos de mejorar.	3	5	5	5	4	4	5	5	5	5	4	4
9 Maneja los objetivos y las tareas que realizan los demás .	3	4	4	5	4	4	5	4	4	3	5	4
10 Colabora con compañeros en sus tiempos libres.	5	2	4	3	4	4	3	5	5	4	4	5
11 Colabora con su compañero de tareas armoniosamente sin consideración a la raza, religión, origen nacional, sexo, edad, etc.	1	5	5	5	5	5	5	4	5	5	5	5
12 Respeta las decisiones de su encargado.	2	5	5	5	5	5	5	5	5	5	5	5
13 Asiste a todas las reuniones.	3	5	3	5	4	4	5	5	5	5	5	5
Puntaje Total	-	184	190	191	169	169	160	194	200	169	192	187
Nota	-	6,3	6,4	6,5	5,8	5,8	5,6	6,5	6,7	5,8	6,5	6,3

## Notas Cuarta Etapa

Área a evaluar	Factor	Rosa González	Claudio Piña	Sergio Abarca	Rodrigo Arratia	Daniel Gutiérrez	Andrés Guerrero	Manuel Venegas	Miguel Núñez	Rodrigo Reyes	Roberto Oñate	Diego Hernandez
1 Ideas que para mejorar el sistema	3	3	2	5	2	4	5	5	4	2	5	5
2 Compromiso con la Exelencia	4	5	5	5	3	3	4	4	4	2	5	4
3 Reacciona adecuadamente ante situaciones conflictivas.	3	4	3	4	3	5	5	5	3	5	5	5
4 Trabajo en equipo	5	5	3	4	2	4	5	4	4	5	5	5
5 Capacidad de trabajo independiente, es decir, sin supervisión.	4	4	5	5	5	5	4	5	5	3	5	5
6 Habilidad para trabajar bajo presión	3	4	5	3	2	5	3	5	5	5	4	5
7 Iniciativa y disposición	3	3	5	4	3	3	5	5	4	4	5	4
8 Asume con responsabilidad las consecuencias negativas de sus acciones, demostrando compromisos y deseos de mejorar.	3	5	4	5	4	5	5	5	5	5	4	5
9 Maneja los objetivos y las tareas que realizan los demás .	3	3	3	5	4	4	5	5	5	3	5	5
10 Cumple con los tiempos de ejecucion de sus tareas	5	5	5	5	5	3	3	5	5	3	5	4
11 Colabora con su compañero de tareas armoniosamente sin consideración a la raza, religión, origen nacional, sexo, edad, etc.	2	5	5	5	5	5	5	5	5	5	5	5
12 Respeta las decisiones de su encargado.	2	5	5	5	5	5	5	4	5	5	5	5
13 Puntualidad y Asistencia	3	4	2	5	3	3	5	4	5	3	3	4
<b>Puntaje Total</b>	-	184	172	198	150	174	191	201	194	161	203	200
<b>Nota</b>	-	6,1	5,8	6,5	5,2	5,9	6,3	6,6	6,4	5,5	6,7	6,6

## Anexo H: Detalle de Esfuerzo

N°	Actividad	Diego Hernandez				Roberto Oñate			
		Horas	% Esf.	UF*HR	Total UF	Horas	% Esf.	UF*HR	Total UF
1	Recolectar Informacion	2,00	0,09	2,35	4,70	2,00	0,09	2,35	4,70
2	Analizar Informacion	4,00	0,24	2,35	9,40	4,00	0,24	2,35	9,40
3	Fijacion Objetivos	2,50	0,25	2,35	5,88	2,50	0,25	2,35	5,88
4	Confeccion documento propuesta de proyecto	4,00	0,22	2,35	9,40	4,00	0,22	2,35	9,40
	<b>Entrega Propuesta</b>	-	-	-	-	-	-	-	-
5	Analisis de recursos informaticos	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
6	Modificaciones a recursos informaticos	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
7	Instalar Plataforma Ubuntu 14.04	0,00	0,00	2,35	0,00	3,50	0,58	2,35	8,23
8	Configurar Cluster MPI	0,00	0,00	2,35	0,00	4,50	1,00	2,35	10,58
9	Configurar Servidor Web	3,00	0,55	2,35	7,05	2,50	0,45	2,35	5,88
10	Configurar Servidor DNS	3,50	0,54	2,35	8,23	3,00	0,46	2,35	7,05
12	Configurar Servidor de Correos	4,00	0,50	2,35	9,40	4,00	0,50	2,35	9,40
13	Recolectar Informacion Algoritmo N°1	1,00	0,06	2,35	2,35	1,00	0,06	2,35	2,35
14	Analizar Informacion Algoritmo N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
15	Fijacion Objetivos Algoritmo N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
16	Recolectar Informacion Algoritmo N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
17	Analizar Informacion Algoritmo N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
18	Fijacion Objetivos Algoritmo N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
19	Recolectar Informacion Algoritmo N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
20	Analizar Informacion Algoritmo N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
21	Fijacion Objetivos Algoritmo N°3	0,50	0,08	2,35	1,18	0,50	0,08	2,35	1,18
22	Confeccion Avance N°1	4,50	0,39	2,35	10,58	4,50	0,39	2,35	10,58
	<b>Entrega Avance 1</b>	-	-	-	-	-	-	-	-
23	Generar Algoritmo Secuencial N°1	2,00	0,16	2,35	4,70	2,00	0,16	2,35	4,70
24	Generar Algoritmo Secuencial N°2	2,00	0,13	2,35	4,70	2,00	0,13	2,35	4,70
25	Generar Algoritmo Secuencial N°3	3,00	0,10	2,35	7,05	3,00	0,10	2,35	7,05
26	Programacion Algoritmo Secuencial N°1	2,50	0,07	2,35	5,88	2,50	0,07	2,35	5,88
27	Programacion Algoritmo Secuencial N°2	2,50	0,06	2,35	5,88	2,50	0,06	2,35	5,88
28	Programacion Algoritmo Secuencial N°3	2,50	0,05	2,35	5,88	2,50	0,05	2,35	5,88
29	OptimizacionCodigo N°1	1,50	0,07	2,35	3,53	1,50	0,07	2,35	3,53
30	OptimizacionCodigo N°2	1,50	0,05	2,35	3,53	1,50	0,05	2,35	3,53
31	OptimizacionCodigo N°3	1,50	0,05	2,35	3,53	1,50	0,05	2,35	3,53
32	Testing Alg Sec N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
33	Testing Alg Sec N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
34	Testing Alg Sec N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
35	Metricas algoritmos Secuenciales N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
36	Metricas algoritmos Secuenciales N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
37	Metricas algoritmos Secuenciales N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
38	Analisis del Algoritmo secuencial que debe ser	2,00	0,11	2,35	4,70	2,00	0,11	2,35	4,70
39	Analisis del Algoritmo secuencial que debe ser	2,00	0,11	2,35	4,70	2,00	0,11	2,35	4,70
40	paralelizado N°3	3,00	0,16	2,35	7,05	3,00	0,16	2,35	7,05
41	Documentacion Alg Sec N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
42	Documentacion Alg Sec N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
43	Documentacion Alg Sec N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
44	Confeccion Avance N°2	5,50	0,46	2,35	12,93	6,50	0,54	2,35	15,28
	<b>Entrega Avance 2</b>	-	-	-	-	-	-	-	-
45	Construccion Algoritmo Paralelo N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
46	Construccion Algoritmo Paralelo N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
47	Construccion Algoritmo Paralelo N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
48	Testing N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
49	Testing N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
50	Testing N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
51	OptimizacionCodigo Paralelo N°1	2,00	0,07	2,35	4,70	2,00	0,07	2,35	4,70
52	OptimizacionCodigo Paralelo N°2	3,50	0,09	2,35	8,23	4,50	0,12	2,35	10,58
53	OptimizacionCodigo Paralelo N°3	5,00	0,14	2,35	11,75	5,50	0,16	2,35	12,93
54	Metricas algoritmos Algoritmo Paralelo N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
55	Metricas algoritmos Algoritmo Paralelo N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
56	Metricas algoritmos Algoritmo Paralelo N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
57	Documentacion Alg Par N°1	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
58	Documentacion Alg Par N°2	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
59	Documentacion Alg Par N°3	0,00	0,00	2,35	0,00	0,00	0,00	2,35	0,00
60	Pagina Web: FrontEnd	8,00	0,62	2,35	18,80	5,00	0,38	2,35	11,75
61	Pagina Web: BackEnd	8,00	0,50	2,35	18,80	8,00	0,50	2,35	18,80
62	Confeccion Avance N°3	6,50	0,27	2,35	15,28	8,00	0,33	2,35	18,80
	<b>Entrega Avance 3</b>	-	-	-	-	-	-	-	-
63	Documentacion Web	5,50	0,41	2,35	12,93	4,50	0,33	2,35	10,58
64	Comparacion de Paralelo vs Secuencial	3,00	0,25	2,35	7,05	3,00	0,25	2,35	7,05
65	Confeccion informe final	7,00	0,21	2,35	16,45	9,00	0,27	2,35	21,15
	<b>Entrega Informe Final</b>	-	-	-	-	-	-	-	-
	<b>Total Horas</b>	109,00				118,00			

N°	Rosa Gonzalez				Sergio Abarca				Manuel Venegas				Rodrigo Arratia			
	Horas	% Esf.	UF*HR	Total UF	Horas	% Esf.	UF*HR	Total UF	Horas	% Esf.	UF*HR	Total UF	Horas	% Esf.	UF*HR	Total UF
1	2,00	0,09	1,50	3,00	2,00	0,09	1,95	3,90	2,00	0,09	1,95	3,90	2,00	0,09	1,10	2,20
2	3,50	0,21	1,50	5,25	3,50	0,21	1,95	6,83	1,50	0,09	1,95	2,93	0,00	0,00	1,10	0,00
3	2,50	0,25	1,50	3,75	2,50	0,25	1,95	4,88	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
4	3,00	0,16	1,50	4,50	3,50	0,19	1,95	6,83	2,00	0,11	1,95	3,90	1,00	0,05	1,10	1,10
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	0,00	0,00	1,50	0,00	2,50	1,00	1,95	4,88	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
7	0,00	0,00	1,50	0,00	1,50	0,25	1,95	2,93	1,00	0,17	1,95	1,95	0,00	0,00	1,10	0,00
8	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
9	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
10	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
12	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
13	4,00	0,26	1,50	6,00	4,50	0,29	1,95	8,78	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
14	3,00	0,23	1,50	4,50	4,00	0,31	1,95	7,80	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
15	3,00	0,33	1,50	4,50	3,00	0,33	1,95	5,85	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
16	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	4,00	0,50	1,95	7,80	0,00	0,00	1,10	0,00
17	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	4,50	0,53	1,95	8,78	0,00	0,00	1,10	0,00
18	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	3,50	0,50	1,95	6,83	0,00	0,00	1,10	0,00
19	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	3,00	0,33	1,10	3,30
20	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	2,00	0,27	1,10	2,20
21	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	1,50	0,25	1,10	1,65
22	1,50	0,13	1,50	2,25	1,00	0,09	1,95	1,95	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	2,50	0,20	1,50	3,75	3,00	0,24	1,95	5,85	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
25	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	6,00	0,39	1,95	11,70	0,00	0,00	1,10	0,00
26	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	7,00	0,24	1,10	7,70
27	8,00	0,23	1,50	12,00	10,00	0,29	1,95	19,50	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
28	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	20,00	0,50	1,95	39,00	0,00	0,00	1,10	0,00
29	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	15,00	0,29	1,10	16,50
30	3,00	0,13	1,50	4,50	8,00	0,35	1,95	15,60	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
31	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	15,00	0,50	1,95	29,25	0,00	0,00	1,10	0,00
32	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	7,00	0,23	1,10	7,70
33	1,00	2,50	1,50	1,50	3,00	0,40	1,95	5,85	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
34	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,36	1,95	4,88	0,00	0,00	1,10	0,00
35	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	2,00	0,27	1,10	2,20
36	1,50	0,25	1,50	2,25	3,50	0,58	1,95	6,83	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
37	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,50	1,95	6,83	0,00	0,00	1,10	0,00
38	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,50	1,00	1,10	0,55
39	3,00	0,16	1,50	4,50	3,00	0,16	1,95	5,85	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
40	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	8,00	0,44	1,95	15,60	0,00	0,00	1,10	0,00
41	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	3,50	0,19	1,10	3,85
42	7,00	0,64	1,50	10,50	3,00	0,27	1,95	5,85	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
43	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	2,50	0,21	1,95	4,88	0,00	0,00	1,10	0,00
44	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	2,50	0,25	1,10	2,75
45	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
46	11,00	0,23	1,50	16,50	15,00	0,32	1,95	29,25	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
47	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	19,00	0,38	1,95	37,05	0,00	0,00	1,10	0,00
48	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	18,00	0,37	1,10	19,80
49	4,50	0,47	1,50	6,75	2,50	0,26	1,95	4,88	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
50	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	5,50	0,42	1,95	10,73	0,00	0,00	1,10	0,00
51	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	3,50	0,26	1,10	3,85
52	2,00	0,07	1,50	3,00	12,00	0,44	1,95	23,40	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
53	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	13,00	0,34	1,95	25,35	0,00	0,00	1,10	0,00
54	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	5,50	0,16	1,10	6,05
55	6,00	0,32	1,50	9,00	7,50	0,41	1,95	14,63	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
56	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	7,00	0,29	1,95	13,65	0,00	0,00	1,10	0,00
57	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	2,50	0,19	1,10	2,75
58	12,50	0,66	1,50	18,75	5,50	0,29	1,95	10,73	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
59	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
60	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	3,00	0,21	1,10	3,30
61	0,00	0,00	1,50	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
62	5,00	0,20	1,50	7,50	5,00	0,20	1,95	9,75	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
63	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
64	3,50	0,26	1,50	5,25	0,00	0,00	1,95	0,00	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
65	3,00	0,25	1,50	4,50	3,00	0,25	1,95	5,85	0,00	0,00	1,95	0,00	0,00	0,00	1,10	0,00
66	4,50	0,14	1,50	6,75	3,50	0,11	1,95	6,83	1,50	0,05	1,95	2,93	0,00	0,00	1,10	0,00
67	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
68	100,50				117,50				122,00				79,50			

N°	Miguel Nuñez				Carlos Guerrero				Daniel Gutierrez				Rodrigo Reyes			
	Horas	% Esf.	UF*HR	Total UF	Horas	% Esf.	UF*HR	Total UF	Horas	% Esf.	UF*HR	Total UF	Horas	% Esf.	UF*HR	Total UF
1	2,00	0,09	0,90	1,80	2,00	0,09	1,95	3,90	2,00	0,09	0,70	1,40	2,00	0,09	0,90	1,80
2	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
3	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
4	1,00	0,05	0,90	0,90	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
7	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
8	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
9	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
10	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
12	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
13	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
14	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
15	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
16	4,00	0,50	0,90	3,60	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
17	4,00	0,47	0,90	3,60	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
18	3,50	0,50	0,90	3,15	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
19	0,00	0,00	0,90	0,00	3,00	0,33	1,95	5,85	3,00	0,33	0,70	2,10	0,00	0,00	0,90	0,00
20	0,00	0,00	0,90	0,00	3,00	0,40	1,95	5,85	2,50	0,33	0,70	1,75	0,00	0,00	0,90	0,00
21	0,00	0,00	0,90	0,00	2,00	0,33	1,95	3,90	1,50	0,25	0,70	1,05	0,00	0,00	0,90	0,00
22	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
23	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
24	5,50	0,35	0,90	4,95	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
25	0,00	0,00	0,90	0,00	9,00	0,31	1,95	17,55	7,00	0,24	0,70	4,90	0,00	0,00	0,90	0,00
26	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
27	15,00	0,38	0,90	13,50	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
28	0,00	0,00	0,90	0,00	17,00	0,33	1,95	33,15	14,00	0,27	0,70	9,80	0,00	0,00	0,90	0,00
29	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
30	10,00	0,33	0,90	9,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	2,00	0,07	0,90	1,80
31	0,00	0,00	0,90	0,00	13,00	0,42	1,95	25,35	8,00	0,26	0,70	5,60	0,00	0,00	0,90	0,00
32	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
33	2,50	0,36	0,90	2,25	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	2,00	0,29	0,90	1,80
34	0,00	0,00	0,90	0,00	3,50	0,47	1,95	6,83	2,00	0,27	0,70	1,40	0,00	0,00	0,90	0,00
35	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
36	2,50	0,36	0,90	2,25	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	1,00	0,14	0,90	0,90
37	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
38	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
39	3,00	0,17	0,90	2,70	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	3,00	0,17	0,90	2,70
40	0,00	0,00	0,90	0,00	5,00	0,27	1,95	9,75	4,00	0,22	0,70	2,80	0,00	0,00	0,90	0,00
41	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
42	5,50	0,46	0,90	4,95	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	4,00	0,33	0,90	3,60
43	0,00	0,00	0,90	0,00	2,00	0,20	1,95	3,90	5,50	0,55	0,70	3,85	0,00	0,00	0,90	0,00
44	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
45	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
46	25,50	0,52	0,90	22,95	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	5,00	0,10	0,90	4,50
47	0,00	0,00	0,90	0,00	20,00	0,41	1,95	39,00	11,00	0,22	0,70	7,70	0,00	0,00	0,90	0,00
48	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
49	3,00	0,23	0,90	2,70	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	4,50	0,35	0,90	4,05
50	0,00	0,00	0,90	0,00	5,50	0,41	1,95	10,73	4,50	0,33	0,70	3,15	0,00	0,00	0,90	0,00
51	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
52	12,00	0,31	0,90	10,80	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	5,50	0,14	0,90	4,95
53	0,00	0,00	0,90	0,00	12,50	0,36	1,95	24,38	6,00	0,17	0,70	4,20	0,00	0,00	0,90	0,00
54	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
55	8,00	0,33	0,90	7,20	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	9,50	0,39	0,90	8,55
56	0,00	0,00	0,90	0,00	8,00	0,59	1,95	15,60	3,00	0,22	0,70	2,10	0,00	0,00	0,90	0,00
57	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
58	2,00	0,12	0,90	1,80	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	15,00	0,88	0,90	13,50
59	0,00	0,00	0,90	0,00	5,50	0,38	1,95	10,73	6,00	0,41	0,70	4,20	0,00	0,00	0,90	0,00
60	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
61	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
62	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
63	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
64	0,00	0,00	0,90	0,00	0,00	0,00	1,95	0,00	0,00	0,00	0,70	0,00	0,00	0,00	0,90	0,00
65	2,00	0,06	0,90	1,80	1,50	0,05	1,95	2,93	1,00	0,03	0,70	0,70	3,00	0,09	0,90	2,70
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	111,00				112,50				81,00				56,50			

N°	Claudio Piña				HH Planif.	HH Ef. Reales	UF Total por Act.	Porcentaje de Avance	Avance
	Horas	% Esf.	UF*HR	Total UF					
1	2,00	0,09	0,70	1,40	22,00	22,00	32,70	100	22
2	0,00	0,00	0,70	0,00	15,00	16,50	33,80	100	16,5
3	0,00	0,00	0,70	0,00	10,00	10,00	20,38	100	10
4	0,00	0,00	0,70	0,00	18,00	18,50	36,03	100	18,5
-	-	-	-	-	-	-	-	-	-
5	0,00	0,00	0,70	0,00	2,50	2,50	4,88	100	2,5
6	0,00	0,00	0,70	0,00	2,00	2,00	3,90	100	2
7	0,00	0,00	0,70	0,00	6,00	6,00	13,10	100	6
8	0,00	0,00	0,70	0,00	5,50	4,50	10,58	100	4,5
9	0,00	0,00	0,70	0,00	4,00	5,50	12,93	100	5,5
10	0,00	0,00	0,70	0,00	6,00	6,50	15,28	100	6,5
12	0,00	0,00	0,70	0,00	6,00	8,00	18,80	100	8
13	5,00	0,32	0,70	3,50	15,50	15,50	22,98	100	15,5
14	6,00	0,46	0,70	4,20	13,00	13,00	16,50	100	13
15	3,00	0,33	0,70	2,10	9,00	9,00	12,45	100	9
16	0,00	0,00	0,70	0,00	8,00	8,00	11,40	100	8
17	0,00	0,00	0,70	0,00	12,00	8,50	12,38	100	8,5
18	0,00	0,00	0,70	0,00	6,00	7,00	9,98	100	7
19	0,00	0,00	0,70	0,00	9,00	9,00	11,25	100	9
20	0,00	0,00	0,70	0,00	7,00	7,50	9,80	100	7,5
21	0,00	0,00	0,70	0,00	4,50	6,00	8,95	100	6
22	0,00	0,00	0,70	0,00	10,00	11,50	25,35	100	11,5
-	-	-	-	-	-	-	-	-	-
23	3,00	0,24	0,70	2,10	19,00	12,50	21,10	100	12,5
24	0,00	0,00	0,70	0,00	19,00	15,50	26,05	100	15,5
25	0,00	0,00	0,70	0,00	19,00	29,00	44,25	100	29
26	12,00	0,34	0,70	8,40	44,00	35,00	51,65	100	35
27	0,00	0,00	0,70	0,00	44,00	40,00	64,25	100	40
28	0,00	0,00	0,70	0,00	44,00	51,00	71,20	100	51
29	9,00	0,39	0,70	6,30	34,00	23,00	33,45	100	23
30	0,00	0,00	0,70	0,00	34,00	30,00	47,10	100	30
31	0,00	0,00	0,70	0,00	34,00	31,00	45,70	100	31
32	3,50	0,47	0,70	2,45	7,50	7,50	9,80	100	7,5
33	0,00	0,00	0,70	0,00	7,50	7,00	8,93	100	7
34	0,00	0,00	0,70	0,00	7,50	7,50	10,43	100	7,5
35	1,00	0,17	0,70	0,70	4,50	6,00	9,78	100	6
36	0,00	0,00	0,70	0,00	4,50	7,00	9,98	100	7
37	0,00	0,00	0,70	0,00	4,50	0,50	0,55	100	0,5
38	9,00	0,47	0,70	6,30	13,00	19,00	26,05	100	19
39	0,00	0,00	0,70	0,00	13,00	18,00	30,40	100	18
40	0,00	0,00	0,70	0,00	13,00	18,50	30,50	100	18,5
41	1,00	0,09	0,70	0,70	7,50	11,00	17,05	100	11
42	0,00	0,00	0,70	0,00	7,50	12,00	13,43	100	12
43	0,00	0,00	0,70	0,00	7,50	10,00	10,50	100	10
44	0,00	0,00	0,70	0,00	10,00	12,00	28,20	100	12
-	-	-	-	-	-	-	-	-	-
45	21,00	0,45	0,70	14,70	45,00	47,00	60,45	100	47
46	0,00	0,00	0,70	0,00	45,00	49,50	64,50	100	49,5
47	0,00	0,00	0,70	0,00	45,00	49,00	66,50	100	49
48	2,50	0,26	0,70	1,75	7,50	9,50	13,38	100	9,5
49	0,00	0,00	0,70	0,00	7,50	13,00	17,48	100	13
50	0,00	0,00	0,70	0,00	7,50	13,50	17,73	100	13,5
51	9,00	0,33	0,70	6,30	22,50	27,00	42,10	100	27
52	0,00	0,00	0,70	0,00	22,50	38,50	59,90	100	38,5
53	0,00	0,00	0,70	0,00	22,50	34,50	59,30	100	34,5
54	5,00	0,27	0,70	3,50	6,00	18,50	27,13	100	18,5
55	0,00	0,00	0,70	0,00	6,00	24,50	29,40	100	24,5
56	0,00	0,00	0,70	0,00	6,00	13,50	20,45	100	13,5
57	1,00	0,05	0,70	0,70	7,50	19,00	30,18	100	19
58	0,00	0,00	0,70	0,00	7,50	17,00	15,30	100	17
59	0,00	0,00	0,70	0,00	7,50	14,50	18,23	100	14,5
60	0,00	0,00	0,70	0,00	18,00	13,00	30,55	100	13
61	0,00	0,00	0,70	0,00	18,00	16,00	37,60	100	16
62	0,00	0,00	0,70	0,00	10,00	24,50	51,33	100	24,5
-	-	-	-	-	-	-	-	-	-
63	0,00	0,00	0,70	0,00	3,00	13,50	28,75	100	13,5
64	0,00	0,00	0,70	0,00	18,00	12,00	24,45	100	12
65	0,00	0,00	0,70	0,00	12,00	33,00	62,23	100	33
-	-	-	-	-	-	-	-	-	-
	93,00				934,00 Total HH	1100,50 Total HH	1730,60 Total UF		100,00

## Anexo I: Alineamiento de Secuencias de ADN

### 1. Presentación general

En Bioquímica, Genética y Biología Molecular la homología de secuencias se refiere al hecho en el que las secuencias de dos o más proteínas o ácidos nucleídos guardan gran similitud debido a que presentan un mismo origen evolutivo. En función de lo anteriormente expuesto. ¿Cuál será la homología de secuencia para proteínas de especies distintas que tengan la misma función bioquímica? ¿Cuál será la homología entre distintas proteínas de una misma especie pero que realicen alguna función común? Para contestar a estas cuestiones se han desarrollado aplicaciones bioinformáticas que son capaces de encontrar rápidamente las identidades o discrepancias entre distintas secuencias.

Se puede interpretar un alineamiento de secuencias en bioinformática como una forma de representar y comparar dos o más secuencias o cadenas de ADN, ARN, o estructuras primarias proteicas. De este modo, al compararlas, se resaltan sus zonas de similitud, las cuales podrían indicar relaciones funcionales o evolutivas entre los genes o proteínas consultados. Las secuencias alineadas se escriben con letras, que representan aminoácidos o nucleótidos, en filas de una matriz en las que, si es necesario, se insertan espacios para que las zonas con idéntica o similar estructura se alineen de forma paralela en la misma posición.

En los últimos años, la Bioinformática ha atraído una gran atención de varias disciplinas, como la informática, las matemáticas y las ciencias biológicas no tradicionales. Esto se debe a la disponibilidad de enormes cantidades de datos biológicos públicos y privados, y a la necesidad imperiosa de transformar datos en información biológica útil y en conocimiento. Las tareas más importantes en la bioinformática son entender las correlaciones, las estructuras y los patrones en los datos biológicos. La información y el conocimiento de estas disciplinas se pueden utilizar de modo inteligente para aplicaciones que cubran el descubrimiento de fármacos, análisis del genoma y control biológico; esto implica el uso de tecnologías informáticas y métodos estadísticos para manejar y analizar un gran volumen de datos biológicos sobre el ADN, el ARN y las secuencias de proteínas, estructuras de las proteínas, los perfiles de expresión genética y las interacciones de la proteína.

En concreto, la bioinformática abarca el desarrollo de bases de datos para almacenar y recuperar datos biológicos, los algoritmos para analizar y determinar las relaciones de datos biológicos, y las herramientas estadísticas para identificar e interpretar conjuntos de minas de datos.



## 2. Objetivos

### Objetivos Generales

Implementar una solución informática basada en un clúster computacional el cual deberá estar orientado al ámbito de comparación entre una cadena de ADN en una base de datos con otras secuencias y en base a una penalización y una matriz de sustitución asignada por el usuario entregar las mayores similitudes entre la cadena de entrada y las cadenas indexadas a la base de datos.

### Objetivos Específico

- Implementar una solución al problema de alineamiento de cadenas de ADN para su comparación en el contexto de encontrar las mayores similitudes utilizando el algoritmo Heurístico mediante matrices de sustitución y penalización por huecos.
- Implementar una solución secuencial y una paralela del problema de alineamiento de cadenas de ADN.
- Utilizar plataforma clúster con la finalidad de reducir los tiempos de ejecución.
- Analizar algoritmo secuencial y paralelo, analizando su comportamiento y obtener métricas de desempeño (SpeedUp, Desempeño y Costo).

## 3. Introducción

Los datos biológicos siguen creciendo de manera exponencial en tamaño y complejidad, como resultado de ello, se introducen nuevos tipos de datos que antes no habían sido vistos ni siquiera en la biología molecular, por lo que es importante que las tecnologías de información avanzadas se apliquen para apoyar la investigación biológica y sus innovaciones. Las actividades específicas de las tecnologías de la información están echando raíces en algunas partes de la comunidad de investigación biológica, y se espera que se beneficien con la tecnología de la información. Las enormes cantidades de datos biológicos y crecientes demandas de la investigación biológica moderna exigen cada vez más la sofisticación y computación potente de las tecnologías de la información (TI).

Los algoritmos de alta velocidad usan principios simplificados para establecer la similitud entre secuencias, en donde el tiempo que esta tarda en llevarse a cabo depende de la sensibilidad del algoritmo, estando fuertemente influenciado por la longitud de la secuencia y el tamaño de la base de datos. Por su parte, los algoritmos de Smith-Waterman en 1981 están basados principalmente en métodos de programación dinámica, buscando óptimos en alineamientos locales de pares de secuencias. De esta forma el tiempo de cálculo es proporcional al cuadrado del tamaño de las secuencias comparadas, por lo tanto su velocidad es lenta para realizar búsquedas en grandes bases de datos. De otra parte, los programas que utilizan algoritmos FASTA [2] y BLAST [2] fueron desarrollados con el objeto de ser de alta velocidad y baja sensibilidad, respectivamente, ya que estos últimos están basados en estrategias heurísticas que concentran sus esfuerzos en las regiones de la secuencia más probablemente relacionadas (posición de mayor coincidencia entre la secuencias) en un tiempo de ejecución corto, ofreciendo buenos resultados.

Pero para reconocer de una manera más genérica los alineamientos se utilizará para plantear nuestra solución sistemas de puntuación basada en matrices de sustitución como un algoritmo heurístico y trascendente en la cual mediante penalizaciones por huecos necesarios para la alineación y cantidad de áreas coincidentes entregará un score de coincidencia asociado a un porcentaje de similaridad.

## 4. Marco Teórico

La evolución se define como "descendencia con modificación" de un antepasado común. A nivel molecular, la modificación significa cambios en el ADN y la secuencia de proteína, y los cambios correspondientes en función de la proteína. Como las mutaciones se acumulan en secuencias derivadas de una secuencia ancestral, las secuencias derivadas divergen una de otra en el tiempo, pero las secciones de las secuencias todavía pueden retener suficiente similitud de permitir la identificación de un ancestro común.

El cambio evolutivo en una secuencia no siempre tiene que ser grande; ligeros cambios en algunas secciones cruciales de una secuencia puede tener profundas consecuencias funcionales. Como era de esperar, la comparación de secuencias a través de la alineación de secuencias es fundamental para la mayoría de análisis bioinformático. Es el primer paso hacia la comprensión de la relación evolutiva y el patrón de divergencia entre dos secuencias. La relación entre dos secuencias también ayuda a predecir la función potencial de una secuencia desconocida, indicando de este modo la relación familia de proteínas. El inmenso número de secuencias de proteínas que pueden ser consultadas a través de bases de datos públicas en el internet es un recurso invaluable para investigadores que están trabajando en el campo de la biología molecular, química de proteínas diagnóstico clínico. Para optimizar el proceso de alineamiento, estos servidores permiten a los investigadores introducir sus secuencias y escoger varios parámetros, tales como valores de penalidad asociados con la inserción de 'gaps' (espacios) y el tipo de matriz (blosun, pam, entre otras) [2].

La búsqueda en las bases de datos con el objeto de extraer secuencias homólogas es el fundamento para el análisis de secuencias. Para cumplir este propósito una variedad de métodos han sido desarrollados y aplicados en amplios paquetes de programas y servidores de Internet. Los programas de búsqueda en bases de datos difieren en la manera de cómo están diseñados los algoritmos que usan. Lo anterior tiene influencia en el tiempo de ejecución (velocidad) y la sensibilidad a la hora de realizar los alineamientos.

Los alineamientos de secuencias de aminoácidos proporcionan una herramienta poderosa para comparar secuencias relacionadas, permitiendo detectar orígenes evolutivos similares y representar una estructura común y/o un rol catalítico. Las inserciones y sustituciones de residuos singulares están generalmente enfatizadas en los alineamientos. Las inserciones aparecen representadas por caracteres nulos añadidos a una de las secuencias, las cuales pueden ser alineadas con letras en las otras [1]. Existen dos tipos de alineamiento según el número de secuencias en estudio: El alineamiento por pares y el múltiple (ASM).

## 4.1 Formato FASTA

Una secuencia bajo formato FASTA comienza con una descripción en una única línea (línea de cabecera), seguida por líneas de datos de secuencia. La línea de descripción se distingue de los datos de secuencia por un símbolo '>' (mayor que) en la primera columna. La palabra siguiente a este símbolo es el identificador de la secuencia, y el resto de la línea es la descripción (ambos son opcionales). No debería existir espacio entre el '>' y la primera letra del identificador. Se recomienda que todas las líneas de texto sean menores de 80 caracteres. La secuencia termina si aparece otra línea comenzando con el símbolo '>'; esto indica el comienzo de otra secuencia. Un ejemplo simple de una secuencia en el formato FASTA puede ser:

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]  
LCLYTHIGRNIYYGSYLYSETWNTGIMLLITMATAFMGVLPWGQMSFWGATVITNLFSAIPYIGTNLV  
EWIWGGFSVDKATLNRFFAHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG  
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPHLIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL  
GLMPFLHTSKHRSMMLRPLSQALFWLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFPLIAGX  
IENY
```

*Figura 1: Ejemplo de Formato Fasta, Fuente: Elaboración Propia*

## 4.2 Nomenclatura de Ácidos

Código de Ácido Nucleico	Significado	Código de Aminoácido	Significado
A	Adenosina	A	Alanina
C	Citosina	B	Ácido aspártico o Aparagina
G	Guanina	C	Cisteína
T	Timidina	D	Ácido aspártico
U	Uracilo	E	Ácido glutámico
R	G A (puRina)	F	Fenilalanina
Y	T C (pirimidina/pYrimidine)	G	Glicina
K	G T (cetona/Ketone)	H	Histidina
M	A C (grupo aMino)	I	Isoleucina
S	G C (interacción fuerte/Strong interaction)	K	Lisina
W	A T (interacción débil/Weak interaction)	L	Leucina
B	G T C (no A) (B viene tras la A)	M	Metionina
D	G A T (no C) (D viene tras la C)	N	Asparagina
H	A C T (no G) (H viene tras la G)	O	Pirrolisina
V	G C A (no T, no U) (V viene tras la U)	P	Prolina
N	A G C T (cualquiera/aNy)	Q	Glutamina
X	máscara	R	Arginina
-	hueco (gap) de longitud indeterminada	S	Serina
		T	Treonina
		U	Selenocisteína
		V	Valina
		W	Triptófano
		Y	Tirosina
		Z	Ácido glutámico o Glutamina
		X	cualquiera
		*	parada de traducción
		-	hueco (gap) de longitud indeterminada

## 4.3 Definiciones Indispensables para la contextualización del problema

Una **secuencia idéntica**, significa que los mismos residuos estar presente en las posiciones en dos secuencias correspondientes se comparan. Para las proteínas, que significa los mismos aminoácidos; para los ácidos nucleicos, que significa las mismas bases [2].

Una **secuencia similar**, significa residuos similares estar presente en las posiciones en las dos secuencias que se comparan correspondiente. Para los ácidos nucleicos, la similitud de secuencia y de identidad de secuencia son los mismos. Sin embargo, para las proteínas, la similitud de secuencia implica aminoácidos con fisicoquímicas similares y propiedades funcionales [2].

Una **secuencia homóloga**, es un término de la evolución que se ha perdido en gran parte de la literatura para referirse a la similitud de secuencia o identidad. Las secuencias se denominan homólogas si tienen un origen evolutivo común, es decir, si se derivan de una secuencia ancestral común. Así, las secuencias son homólogas o no homóloga y no hay cuantificación de homología. Sin embargo, incluso ahora, expresiones como "alta homología", "homología significativa", e incluso especificar un "% de homología" son muy utilizados. Tal uso no tiene ninguna referencia al apuntalamiento evolutivo de la homología plazo. [2].

El **Alineamiento Global**, compara dos secuencias a lo largo de toda su longitud, y viene con la mejor alineación que muestra el número máximo de nucleótidos o aminoácidos alineados. El algoritmo de alineación global más reconocido es el de NeedlemanWunsch. Alineación Global funciona mejor cuando las secuencias son similares en carácter y longitud. Debido alineación global muestra la mejor alineación entre dos secuencias utilizando toda la secuencia, puede perderse una pequeña región de importancia biológica. [2]

El **Alineamiento Local**, en contraste con la alineación global, la alineación de secuencias locales se pretende encontrar la región más similar dos secuencias alineadas. El algoritmo que impulsa la alineación local es el algoritmo de Smith-Waterman. Un algoritmo de alineamiento local encuentra a la región f mayor similitud entre dos secuencias y construye la alineación hacia el exterior de esta región. Si hay re múltiples regiones de muy alta similitud, se aplica el mismo principio. Obviamente, la alineación local es útil para las secuencias que no son la misma naturaleza y duración, pero re sospechosos de contener pequeñas regiones de similitud, como motivos biológicamente importantes. [2]

```
Global FTFTALILLAVAV
      F--TAL-LLA-AV

Local  FTFTALILL-AVAV
      --FTAL-LLAAV--
```

*Figura 2: Ejemplo para diferenciar el alineamiento global del local. Fuente: Elaboración Propia*

Ahora en el contexto de nuestra solución, aun cuando existan soluciones algorítmicas como Needleman-Wunsch para alineamiento global y Smith-Waterman para alineamiento local, para reconocer de una manera más genérica los alineamientos se utilizará para plantear nuestra solución sistemas de puntuación basada en matrices de sustitución como un algoritmo heurístico y trascendente en la cual mediante penalizaciones por huecos necesarios para la alineación y cantidad de áreas coincidentes entregará un score de coincidencia asociado a un porcentaje de similaridad.

#### 4.4 Matrices de sustitución

Para producir buenos alineamientos es importante la elección de una función de puntuación que refleje observaciones biológicas o estadísticas sobre secuencias conocidas. Las secuencias de proteínas son alineadas usando frecuentemente matrices de sustitución que reflejan las probabilidades de particulares sustituciones carácter por carácter. En biología evolutiva una matriz de sustitución, o de puntuación, describe el ritmo al que un carácter en una secuencia cambia a otro carácter con el tiempo. Las matrices de sustitución se ven usualmente en el contexto de alineamiento de secuencias de aminoácidos ADN, donde la similitud entre secuencias depende del tiempo desde su divergencia y de los ritmos de sustitución según se representan en la matriz [7]. Estas matrices se utilizan como parámetros de los algoritmos de alineamiento (por ejemplo los de Needleman-Wunsch o Smith-Waterman), en los cuales cumplen el papel de asignar una determinada puntuación a cada emparejamiento entre los aminoácidos de las secuencias a alinear, contribuyendo así a la puntuación global del alineamiento.

¿Cómo interpretar a grandes rasgos la alineación de dos secuencias?

Si dos secuencias en un alineamiento proceden de un ancestro común, las no coincidencias pueden interpretarse como mutaciones puntuales (sustituciones), y los huecos como mutaciones de inserción o delección introducidas en uno o ambos linajes en el tiempo que transcurrió desde que divergieron. En el alineamiento de secuencias proteicas, el grado de similitud entre los aminoácidos que ocupan una posición concreta en la secuencia puede interpretarse como una medida aproximada de conservación en una región particular, o secuencia motivo, entre linajes. La ausencia de sustituciones, o la sustitución de aminoácidos cuya cadena lateral tiene propiedades químicas similares en una región particular de la secuencia (lo que se denomina como sustituciones muy conservadas) indica que esta zona tiene importancia estructural o funcional.

## 5. Estado del Arte

### *Grandes Hitos Históricos de la Bioinformática*

El inicio de la biología molecular, la bioquímica y la genética, ciencias vinculadas a la informática, tuvo lugar aproximadamente hace 50 años, cuando en 1953 Watson y Crick los padres del ADN [4] propusieron el modelo de la doble hélice para explicar su estructura. Watson y Crick no percibieron la gran cantidad de volumen de información que se generaría y que crecería en forma exponencial a partir de ese momento.

En forma contraria a lo que podría suponerse, las herramientas computacionales comenzaron a aplicarse en la biología molecular mucho antes del comienzo de la era de la Internet o de los proyectos de secuenciación del genoma [5] y con la constante producción de información biológica, la cual crecía a un ritmo lento comparado con el actual volumen de generación de datos, se creó la necesidad de recopilar y organizar toda la información generada a partir de dichos proyectos de secuenciación.

En 1965, Margaret Dayhoff creó la primera base de datos de secuencias biológicas, en la cual almacenó y puso a disposición de la comunidad científica todas las secuencias de ADN y proteínas descritas hasta la fecha. Ocho años más tarde, en 1973, se anunció y creó la base de datos más antigua que se conoce y la cual sigue vigente, el Protein Data Bank.

En la década de los años 80, la bioinformática ya tenía nombre en el ámbito de la investigación científica, y teniendo conocimiento de ello, varios grupos de investigación de prestigio, dieron origen a la base de datos más conocida en el mundo, el GenBank [6]. Dicho proyecto fue financiado por los National Institutes of Health de los Estados Unidos y otras instituciones gubernamentales, como el United States Department of Energy y el United States Department of Defense.

Actualmente quienes mayormente aportan al conocimiento son las universidades de estados unidos ya que cuentan con grandes aéreas de investigación aunque resaltan muchos países como el Reino unido, Japón, Alemania y sorprendiendo en avances de investigación India. Sin embargo no se puede extrapolar que aun cuando Estados Unidos y el Idioma inglés sea el dominante en papers, artículos y tesis, las nacionalidades de los investigadores sean referentes al país, mayormente los apellidos más recurrentes son de descendencia asiática.

## 6. Definición del Problema

Consiste en la alineación de dos secuencias de ácidos nucleicos o dos proteínas para encontrar el mejor partido. El alineamiento múltiple realiza la misma función utilizando más de dos secuencias. El propósito de la alineación es identificar regiones de similitud que pueden tener consecuencia estructural, funcional y evolutiva.

El genoma humano es un término que se utiliza para describir el contenido total de ADN (siglas del ácido desoxirribonucleico, la molécula portadora de la información genética) en cada célula humana, y que contiene el conjunto de instrucciones (código genético) para la síntesis de las proteínas, las cuales son, en última instancia, las responsables de la fisiología y morfología de las células que, a su vez, son los componentes básicos de todos nuestros tejidos y órganos.

Todas y cada una de los trillones de células que componen nuestros tejidos y nuestros órganos (excepto las células sexuales: espermatozoides y el óvulo, que cuentan con la mitad del contenido de ADN, y determinadas células como los glóbulos rojos de la sangre) son portadoras del mismo conjunto de instrucciones genéticas, del mismo genoma. Por tanto, independientemente del órgano o tejido del que extraigamos el ADN de un mismo individuo, el perfil genético que obtengamos será siempre el mismo para identificar a tal individuo.

Algunos autores afirman que existe la necesidad imperiosa de poseer un conocimiento general en bioinformática que abarque a estudiantes, investigadores y a la industria, ésta necesidad surgió debido a que se requería una evolución a partir de la interacción activa de dos disciplinas de rápido desarrollo como la biología y las tecnologías de la información. La solución de problemas biológicos modernos requiere métodos computacionales avanzados y sofisticados, ya que el volumen de los datos en estos dominios sigue creciendo de manera exponencial en tamaño y complejidad. Un ejemplo de esto es el centro de investigación CENBIOTEP [3], donde se han presentado problemas con la gran cantidad de datos biológicos que desean procesar, debido a que no poseen el conocimiento informático suficiente para llevar a cabo su investigación de manera eficaz y en el tiempo deseado.<sup>3</sup> Es indispensable y urgente que las tecnologías de información avanzadas sean aplicadas para apoyar la investigación biológica basada en el análisis de secuencias de ADN. ¿Qué tecnologías informáticas se usan para la solución de problemas biológicos relacionados con el análisis de secuencias de ADN?



## 7. Modelo de la Solución Propuesta

Dada la naturaleza del problema, enfocado principalmente a la alineación de ácidos nucleicos mediante el uso de programación dinámica, en donde a lo largo todo el proceso en esta tarea en particular es secuencialmente dependiente, la paralelización de la alineación de cadenas propiamente tal no resulta una alternativa factible, dicho lo anterior, se ha optado por otro enfoque para la subdivisión de procesos.

Una vez definida nuestra base de datos y un archivo de entrada correspondiente a una cadena en formato .fasta, se procede a realizar la comparación según el funcionamiento del algoritmo de alineación de secuencias de ADN, como mencionamos anteriormente, este utiliza programación dinámica para dicho propósito; y además una matriz de 4x4, debido a las 4 letras que se comparan para cadenas de ADN: A de adenina, G de guanina, C de citosina y T de Timina. Al terminar de comparar la cadena, el algoritmo asigna un puntaje a la alineación, el cual indica que a mayor valor, mayor similitud existe entre la comparación y la secuencia original. Cuando se encuentra una puntuación más alta que la que ya se encontraba en ese momento, se guardan las cadenas alineadas junto con la descripción de ambas hasta que termine el programa o se encuentre otra alineación con mayor puntaje. Después de todo ese proceso, la salida del programa entrega tanto las cadenas alineadas, el valor de similitud entregado por el algoritmo (Score) y el porcentaje de similitud en relación a la cadena que ha proporcionado el usuario para el análisis, todo esto, para las “n” secuencias con mayor similitud que estime conveniente el usuario para sus propósitos.

El algoritmo correrá en una plataforma web, la cual estará desarrollada en un clúster de computadores que utilizan Python como lenguaje de programación, en Ubuntu.

Si lo que deseamos es encontrar las “n” secuencias con mayor similitud en base a la cadena proporcionada por el usuario, resulta lógico suponer que el enfoque idóneo para la subdivisión de las tareas es la segmentación de la base de datos, de tal manera que cada nodo esclavo opere con su fragmento de datos previamente proporcionado por el nodo maestro. Sin duda, se trata de un problema altamente paralelizable si lo abordamos de esa forma.

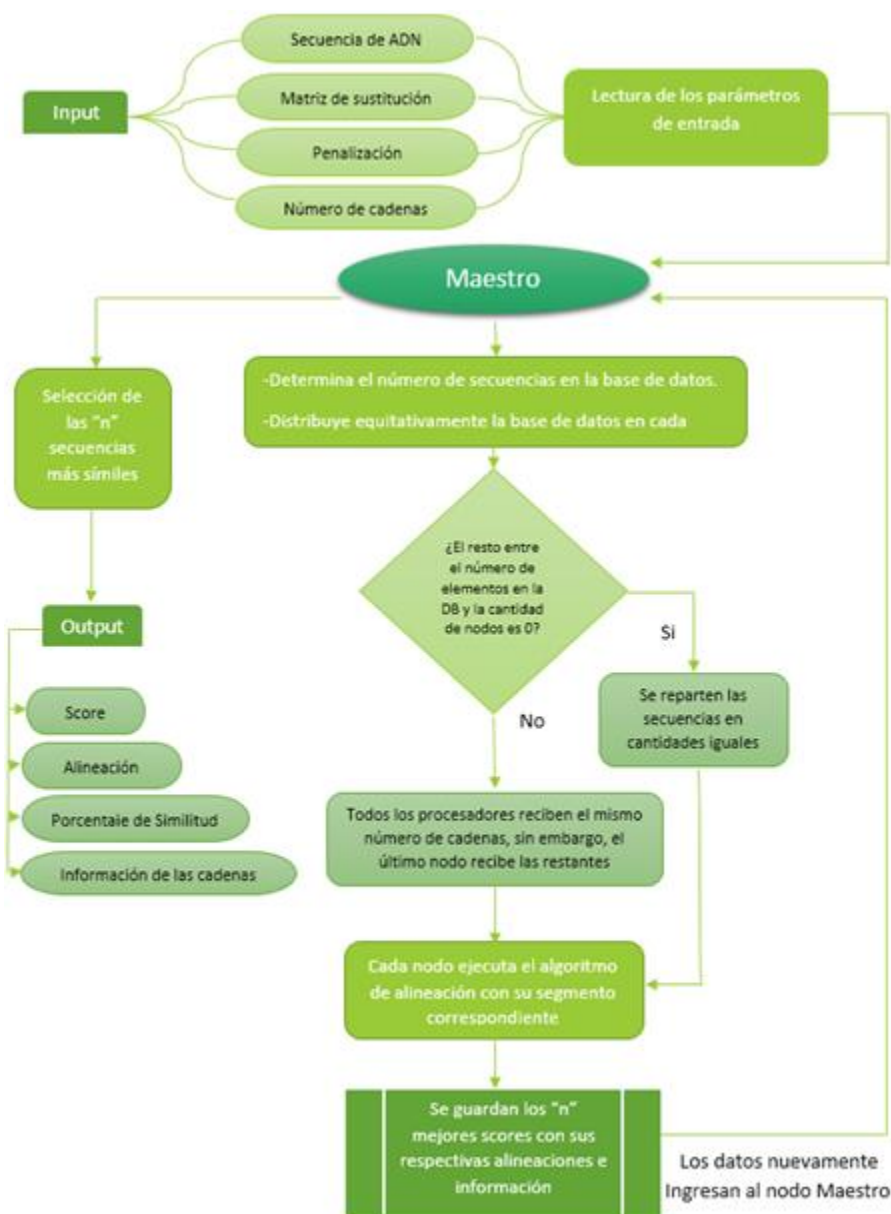


Figura 3: Modelo de workflow del proceso. Fuente: Elaboración Propia

Los procesos para llevar a cabo la tarea anteriormente mencionada se inician con la entrada de los parámetros otorgados por el usuario, los cuales serán descritos con mayor detalle a continuación:

- Cadena a comparar: Secuencia de ADN contenida en un fichero en formato “.fasta”. El fichero .fasta debe respetar las normas correspondientes a su formato con la finalidad de asegurar el correcto funcionamiento del software. Si el archivo de entrada posee más de una secuencia de ADN, se trabajará con la última cadena contenida en el fichero.
- Matriz de sustitución: Con la finalidad de otorgar un mayor control y precisión de los resultados, el cliente tendrá a su disposición 33 diferentes matrices de sustitución, que podrá elegir mediante un menú de selección Individual.
- Penalización: Parámetro de entrada de tipo entero que indica el valor otorgado al no existir coincidencia entre nucleótidos en un segmento de la secuencia, es decir, un espacio vacío.
- N° de cadenas de salida: Dependiendo de la finalidad del estudio, el cliente necesitará una cantidad variable de alineamientos otorgados por el software, en base a lo anterior, se ha decidido facultar al cliente definir el número de secuencias con mayor similitud que desee. Este parámetro de entrada será un número entero distinto de cero.

Una vez que son leídos estos parámetros de entrada, el nodo maestro comienza la lectura de la base de datos, calculando el número de secuencias totales contenidas en esta, esto se realiza con el objetivo de asignar una cantidad equitativa de secuencias de ácidos nucleicos a cada nodo esclavo.

Una vez definida la cantidad de secuencias con las que debe operar cada nodo, se procede a ejecutar el algoritmo de alineación entre cada cadena y la secuencia otorgada por el usuario, obteniendo un Score que indica el grado de concurrencia entre ambas cadenas producto de la alineación.

Cada nodo esclavo le entregará al nodo maestro las 5 secuencias con mejor grado de concurrencia que ha calculado, esto se realiza con la finalidad de otorgarle al cliente un resultado con mayor precisión.

El nodo maestro realiza un ordenamiento descendente del Score, junto con la información otorgada por todos los nodos esclavos. Dependiendo del número de alineamientos que desee el usuario, estos serán entregados en un fichero en formato .txt que contendrá la información de la cadena, su Score, porcentaje de similitud y alineamiento, esto se realizará en forma independiente para cada cadena que solicite el usuario.

Para una mayor comprensión de los datos de salida, se detallarán a continuación:

- Score: Número entero que representa el grado de concurrencia entre 2 secuencias de ADN. La puntuación es directamente proporcional al grado de similitud entre las cadenas comparadas.

- **Porcentaje de similitud:** Con el objetivo de facilitar los análisis de nuestro cliente, el software entregará un porcentaje de similitud entre la cadena de entrada (archivo .fasta) y la secuencia de ADN encontrada en la base de datos de mayor concurrencia. En el caso de que el cliente solicite más de una cadena de ADN, se le otorgará un porcentaje de similitud para cada cadena y/o alineación solicitada. Este porcentaje está relacionado en base al Score otorgado entre las alineaciones.
- **Cadenas con mayor similitud:** Junto a la información de cada secuencia de ADN, el software generará tantos archivos como secuencias solicite nuestro cliente, de esta forma, la información otorgada por el programa garantizará un resultado limpio, claro y ordenado para su posterior análisis.
- **Alineación de Cadenas:** La alineación correspondiente entre las cadenas que presenten mayor similitud también será entregadas en archivos independientes junto a la secuencia de ADN asociada.

## 8. Bibliografía

[1] Rehm, B.H.A. (2001). Bioinformatic tools for DNA/protein sequence analysis, functional assignment of genes and protein classification. Appl. Microbiol. Biotechnol. 57(5-6):579-592.

[2] Pearson, W. R. (2014). BLAST and FASTA similarity searching for multiple sequence alignment, Bioinformatics for Beginners, paginas 133-155 Retrieved from scopus.com

[3] Centro de Biología molecular y Biotecnología de la Universidad Tecnológica de Pereira.

[4] La Jornada. 20 de octubre de 2007. Watson y Crick, los padres del ADN [en línea]. México, D.F. Disponible desde Internet en: [citado en 7 de octubre de 2010].

[5] Colombia Médica [en línea]. Universidad del Valle: Cali, 2008 – [citado en 7 de octubre de 2010]. Vol. 39, No. 001. (enero-marzo 2008). Disponible desde Internet en: ISSN 1657-9534.

[6] GenBank® es la base de datos de secuencias genéticas del Instituto Nacional de Salud (en inglés National Institutes of Health, NIH), una colección anotada de todas las secuencias de ADN a disposición del público.

[7] Attwood, T. K. (2002). «6». Introducción a la bioinformática. Prentice Hall. ISBN 84-205-3551-6.

## Anexo J: Búsqueda de patrones del Código de la Biblia

### 1. Presentación General

Como uno de los 3 módulos pertenecientes a la plataforma de servicios se entrega al usuario una herramienta de búsqueda de patrones, la cual se conoce puntualmente como el código de la biblia.

Desde tiempos remotos se desea tener algún indicio de lo que depara el destino, de obtener alguna premonición, de poder prever y adelantarse a sucesos que puedan cambiar el transcurso de la historia. Suena interesante entonces la hipótesis de que todo este conocimiento se encuentre almacenado dentro de las páginas de la Torah, o biblia hebrea, este planteamiento cuenta tanto con una facción que lo avala como de otra que lo desmiente, donde el mayor interés para nosotros se encuentra en la base computacional del estudio de la problemática.

### 2. Objetivos

#### 2.1 Objetivo General

Lograr la correcta implementación de los algoritmos computacionales, tanto secuenciales como paralelos, utilizados dentro del proceso de búsqueda de patrones en el Código de la Biblia.

#### 2.2 Objetivos Específicos

- Conseguir que los algoritmos de carácter genérico, es decir, una estructura que permita el análisis de cualquier tipo de texto.
- Establecer métricas adecuadas al caso, que muestren claramente el rendimiento de la solución presentada.
- Implementar el código paralelo de manera tal que consiga encontrar patrones en más de un texto.
- Conseguir una correcta integración con la plataforma web, de modo que permita la recepción y envío de datos por parte del algoritmo.
- Implementar salidas de respuesta cómodas para el usuario vía email u plataforma.
- Completar la integración del módulo con la plataforma web.

### 3. Introducción

El código de la Biblia, también conocido como el código de la Torá, consiste en un supuesto código oculto en la Torá judía (el Pentateuco del Antiguo Testamento) que relata acontecimientos del pasado, presente y futuro.

En este proyecto se trató la búsqueda de patrones dentro de textos tanto bíblicos como de cualquier otra índole, aplicando técnicas de minería de datos a una gran cantidad de información.

Al trabajar con una gran cantidad de información, el factor tiempo es uno de los temas con mayor consideración para investigar la problemática. Esto es lo que consumió la mayor cantidad de tiempo en el procesamiento de los datos, al ejecutar un algoritmo de procesos secuenciales, es por ello que se implementa un algoritmo paralelo que corre en varios equipos computacionales involucrados en el cálculo.

### 4. Marco Teórico

SLE es una secuencia de letras extraída de un texto formada por una letra inicial la cual representa la primera posición del SLE, las siguientes letras que lo conforman están separadas por el mismo número de letras sucesivas, a esta separación equidistante se le llama “salto”.

$$n, n+d, n+2d, \dots, n+(k-1)d.$$

n: Posición inicial  
d: Salto  
k: Largo en caracteres del SLE.

*Fórmula que describe una secuencia de letras equidistantes.*

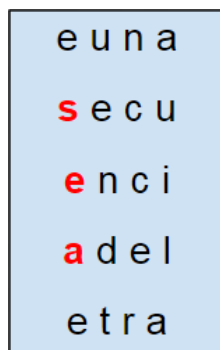
Para lograr determinar esta secuencia, el primer paso es convertir un texto en una cadena de caracteres, en este caso dentro de un arreglo unidimensional en el cual se eliminan los caracteres especiales y espacios, además de estandarizar la grafía en minúsculas. Una forma de ilustrarlo es la siguiente:

"Ejemplodeunasecuenciadeletrasequidist  
antes"

Palabra "Sea" - n:13 , d:4 , k : 3 .

*SLE ilustrado con la palabra "sea".*

Luego la cadena de caracteres es “enrollada” en un cilindro, de esta forma podremos buscar otros SLE que pudieran formarse en su cercanía dado un determinado paso. La forma gráfica del cilindro se muestra utilizando la cadena de caracteres del ejemplo anterior:



*Cilindro de tamaño 4 para la palabra “sea”.*

Logramos apreciar un par de palabras en la siguiente imagen: פִּרְטִי (Privado) aparece como un SLE (de salto 855) junto a נִנְזָן (nombres) en el texto (Gn 26:18).



*Fragmento de la tabla que contiene el texto citado, privado y nombres*

#### 4.1. Geometría de SLE

Pasos:

- Si se utiliza un salto de tamaño  $n$  en un cilindro de  $n$  columnas aparecerá verticalmente.
- Si se utiliza un salto de tamaño  $n$  en un cilindro de  $n/2$  columnas aparecerá verticalmente.
- Si se utiliza un salto de tamaño  $n+1$  |  $n-1$  sobre un cilindro de  $n$  columnas aparecerá diagonalmente.

Para esta definición de geometría es necesario que algunos cálculos sean redondeados, dado que no se puede encontrar unas palabras con pasos expresados con coma flotante.

Cabe mencionar además que esta es una forma de mostrar cómo se puede leer el supuesto “código” cifrado en la biblia hebrea, específicamente en el pentateuco. Al traducirlo a un lenguaje extranjero perdería su carácter “predictivo”, por ende la muestra del algoritmo no tiene otro fin que el mostrar la mecánica del algoritmo.

#### 4.2. Resumen del procedimiento.

- Definir el concepto de “salto” entre dos palabras, esto nos dará la idea de la proximidad que tienen las palabras con el fin de determinar qué palabras son las más “cercanas” entre sí.
- Definir estadísticas para crear una muestra de palabras en relación a la proximidad antes mencionada, la cual se compone de pares de palabras asociadas.
- Seleccionar un elemento de la muestra de pares de palabras relacionadas, con el cual se realizarán pruebas.
- Realizar el análisis estadístico correspondiente a la muestra seleccionada.

### 5. Estado del Arte

Aunque fue primero intuido por el rabino H.M. D. Weissmandel, hace más de 50 años, fue en realidad el Dr. Eliyahu Rips, experto mundial en teoría de grupos (modelo matemático en que se basa la física cuántica), quien lo descubrió, y posteriormente otro físico israelí, Doron Witztum, fue quien completó el modelo matemático para descifrar el código. Este fue corroborado por matemáticos de la Universidad de Harvard, Yale y de la Universidad Hebrea. Este descubrimiento fue anunciado primero en la revista norteamericana especializada Statistical Science (Ciencia Estadística), no sin antes someterlo al análisis de otros expertos, procedimiento habitual de verificación en revistas de este tipo. El experimento original fue llamado “Secuencias equidistantes de letras en el libro del Génesis”, en cuyo resumen se leía: “El análisis randomizado (aleatorio) señala la existencia de información oculta en el texto del Génesis, imbricada en forma de secuencias equidistantes de letras. Su nivel de acierto es del 99.998%”.



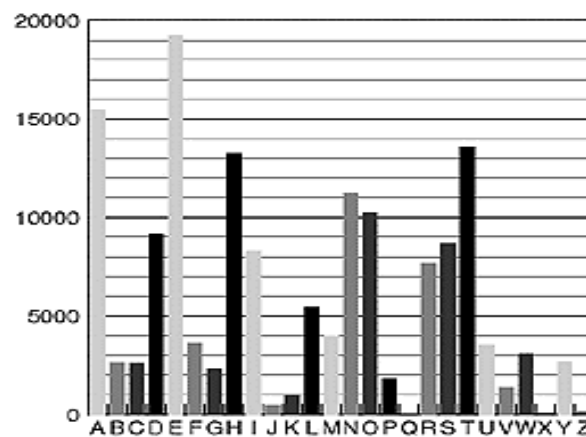
El periodista norteamericano Michael Drosnin publicó el descubrimiento en 1997 su libro El Código Secreto de la Biblia, señalando que para ello "tuvo que aprender hebreo y estudiar y comprobar el código durante cinco años". Explica Drosnin que "la Biblia tiene la forma de un gigantesco crucigrama. Está codificada de principio a fin con palabras que al conectarse entre sí, revelan una historia oculta".

S	A	N	D	M	Y	D	A	U	G	H	T	E	R	S	T	H	O	U	H
M	Y	D	A	U	G	H	T	E	R	S	T	H	O	U	H	A	S	T	N
U	G	H	T	E	R	S	T	H	O	U	H	A	S	T	N	O	W	D	O
E	R	S	T	H	O	U	H	A	S	T	N	O	W	D	O	N	E	F	O
H	O	U	H	A	S	T	N	O	W	D	O	N	E	F	O	O	L	I	S
A	S	T	N	O	W	D	O	N	E	F	O	O	L	I	S	H	L	Y	I
O	W	D	O	N	E	F	O	O	L	I	S	H	L	Y	I	N	S	O	D
N	E	F	O	O	L	I	S	H	L	Y	I	N	S	O	D	O	I	N	G
O	L	I	S	H	L	Y	I	N	S	O	D	O	I	N	G	I	T	I	S
H	L	Y	I	N	S	O	D	O	I	N	G	I	T	I	S	I	N	T	H

*Matriz para Roswell/UFO encontrada en el libro del Génesis.*

Rips eliminó los espacios entre palabras y convirtió la totalidad del texto bíblico original en una "hebra" continua compuesta por 304.805 letras. Al hacerlo, la estaba devolviendo a la forma primigenia que los grandes sabios le atribuyen. Según la historia, ésa es la forma en que Moisés habría recibido la Biblia de Dios: continua, sin separación de palabras.

El computador explora esta hebra en busca de nombres, palabras y frases codificadas. Comienza por la primera letra de la Biblia y verifica todas las secuencias alternas posibles: palabras formadas por saltos de 1, 2, 3 y así, hasta varios miles de espacios. Luego repite la búsqueda empezando por la segunda letra, luego por la tercera, la cuarta, hasta llegar a la última letra del texto. Como en un crucigrama, la información codificada puede aparecer en forma vertical, horizontal o diagonal. Cuando localiza una palabra clave, el computador se dedica a buscar información relacionada con ella que encuentra generalmente en el mismo tramo del texto. Como resultado del estudio se concluye que es más fácil encontrar palabras cortas, así como es más fácil armar palabras que se compongan con las letras de mayor frecuencia relativa dentro del texto utilizado. A modo de ejemplo se muestra el siguiente gráfico:



Conteo de letras en el libro del Génesis.

En el año 1999, también publicado Statical Science, en el cual refutan sus postulados acusando que este es fatalmente deficiente en sus bases, partiendo por el tipo de distribución escogida y por consiguiente de los resultados obtenidos.

A pesar de esta negativa en el año 2015 nuevamente un rabino de nombre Aharón David Shlezinger viraliza un video donde afirma lo siguiente: “Según la Torá, el apellido de Nisman aparece en Génesis 47:29, saltando 77 letras. Es decir, se señala una letra, se saltan 77 letras, y se señala otra letra, se vuelven a saltar 77 letras y se señala otra letra. Así aparece (Alberto) Nisman...” Y en la última letra de Nisman, el texto que la atraviesa dice: ‘Por favor no me entierres en Egipto, pues he de yacer junto con mis padres’ (Génesis 47:29-30)”.

Cabe destacar que Alberto Nisman (1963-2015), fue el fiscal a cargo de la causa por el atentado contra la AMIA. Tras denunciar a importantes funcionarios del gobierno por supuesto encubrimiento, apareció muerto en su departamento el 18 de enero de 2015 con un disparo en la cabeza. Faltaban horas para que expusiera su denuncia frente a la Cámara de Diputados. La opinión del país se encuentra dividida entre los partidarios del suicidio y los del asesinato, dada la falta de pruebas sobre el trágico suceso.

## 5.1. Predicciones reales

Se dice que en la Biblia están codificados centenares de hechos cruciales para el mundo, desde la segunda guerra mundial hasta el escándalo del Watergate, el Holocausto, la bomba de Hiroshi-ma, la llegada del hombre a la Luna o el reciente impacto de un cometa en Júpiter.

Un hecho curioso que ocurre cuando Drosnin le envía una carta a Itzhak Rabin a través de su amigo Chaim Hurí, advirtiéndole de un supuesto peligro:

«Si me he permitido escribirle es porque la única vez que su nombre completo -Itzhak Rabin-aparece codificado en la Biblia, las palabras "asesino que asesinara" lo cruzan. Esto no debería tomarse a la ligera, toda vez que los asesinatos de Anwar al-Sadat y de John y Robert Kennedy también aparecen codificados en la Biblia; en el caso de Sadat, con el nombre completo del homicida, la fecha y el lugar del atentado, y el modo de perpetuar. Creo que corre usted un grave peligro, pero también que el peligro puede ser evitado.»

La advertencia no es tomada en serio y el 4 de noviembre de 1995 llegó la terrible confirmación. Un hombre que se creía encomendado por Dios acababa de disparar a Rabin por la espalda. Durante tres mil años, el atentado había permanecido oculto en el código secreto de la Biblia.

Después del atentado, el nombre del asesino -«Amir»-saltó de inmediato a la vista. Siempre había estado allí, encima del de su víctima, perfectamente descifrable.

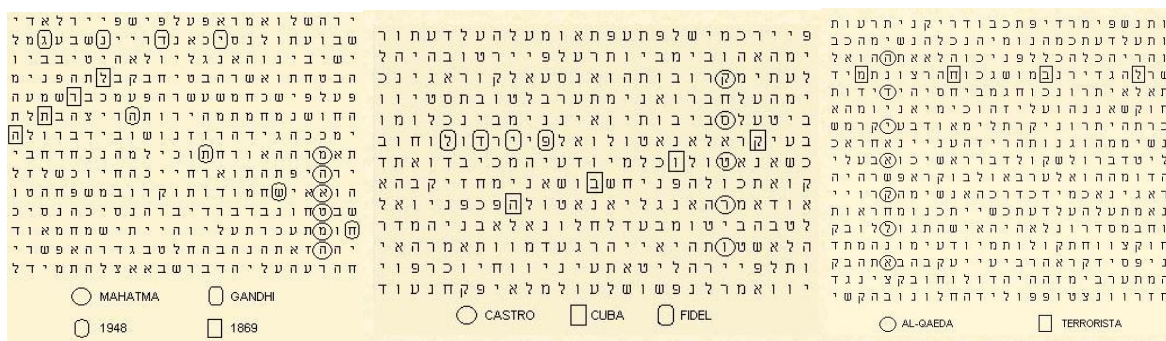
Por otra parte se habla de que al conocer una predicción esta puede modificarse, basándose en su poder de predicción. Entre ellas es interesante mencionar el patrón guerra mundial / holocausto atómico / fin de los días / 5766 (año hebreo que simboliza el 2006 para nuestra cultura), esta predicción data del año 2001. Se cree que esta predicción fue evadida dada la intervención estadounidense.

## 5.2. Críticas

Contrario a lo que intentaron probar WRR, el grupo de investigadores compuesto por Brendan Mckay, Dror Bar-Natan, Maya Bar-Hillel, y Gil Kalai, buscan refutar lo presentado en las investigaciones previas respecto al código de la biblia. Por esta razón es totalmente propio tomarlo como parte de nuestro estudio ya que la técnica empleada para buscar patrones es la misma en ambas investigaciones.

Para una contra-argumentación fue utilizada la versión hebrea del libro “Moby Dick”, los patrones encontrados se muestran en las imágenes inferiores con el siguiente orden:

- mahatma / ghandhi / 1948 / 1889
- castro / cuba / fidel
- al-qaeda / terrorista



SLE encontrados en el libro Moby Dick

Puesto que es factible encontrar no solo en libros distintos a la torah, sino que además hacerlo en otras lenguas, es que el carácter premonitorio del código de la biblia solamente se sustenta en la fe profesada a la palabra de Dios, ya que expresamente según lo que indica la torah estos mensajes no tienen significancia a no ser que sean extraídos de ella misma, hecho que cae en tela de juicio cuando se pueden encontrar y demostrar resultados en otras fuentes.

## 6. Descripción del modelo de solución

El usuario entrega como parámetro al algoritmo el nombre del fichero con el texto a buscar (en formato PDF), el cual es transformado en una lista de cadenas de texto con el contenido de cada una de sus páginas. Otro parámetro ingresado por el usuario es la palabra o cadena de palabras (frase) que se busca dentro del documento, cuyas ocurrencias se irán registrando a lo largo de la ejecución. El último valor que ingresado es el del maximo salto que se probado sobre la extensión de la cadena ingresada.

Al arrancar se divide, dependiendo del número de procesadores esclavos disponibles, un intervalo de saltos el cual es asignado a cada nodo para que cada uno individualmente los procese de la misma forma que lo hace la búsqueda de coincidencias de forma secuencial, esto respondiendo a una arquitectura donde una misma instrucción se ejecuta sobre una gran cantidad de datos.

Una vez terminada la lectura del texto completo, cada nodo devuelve al maestro las coincidencias encontradas en el texto si es que existiesen, teniendo en cuenta los saltos con el cual se localizaron y donde inicia y termina cada 'keyword'. El nodo maestro se encarga entonces de reunir todas las respuestas y llevarlas a un formato en serie (en un archivo JSON) para que estas sean integradas a la plataforma web.

## 6.1. Workflow del proceso secuencial

El desarrollo del algoritmo involucra diferentes etapas de ejecución. Para establecer con más claridad su funcionamiento, estas etapas del programa fueron agrupadas en diferentes etapas, las cuales se muestran a continuación:



### 6.1.1. Lectura de Parámetros

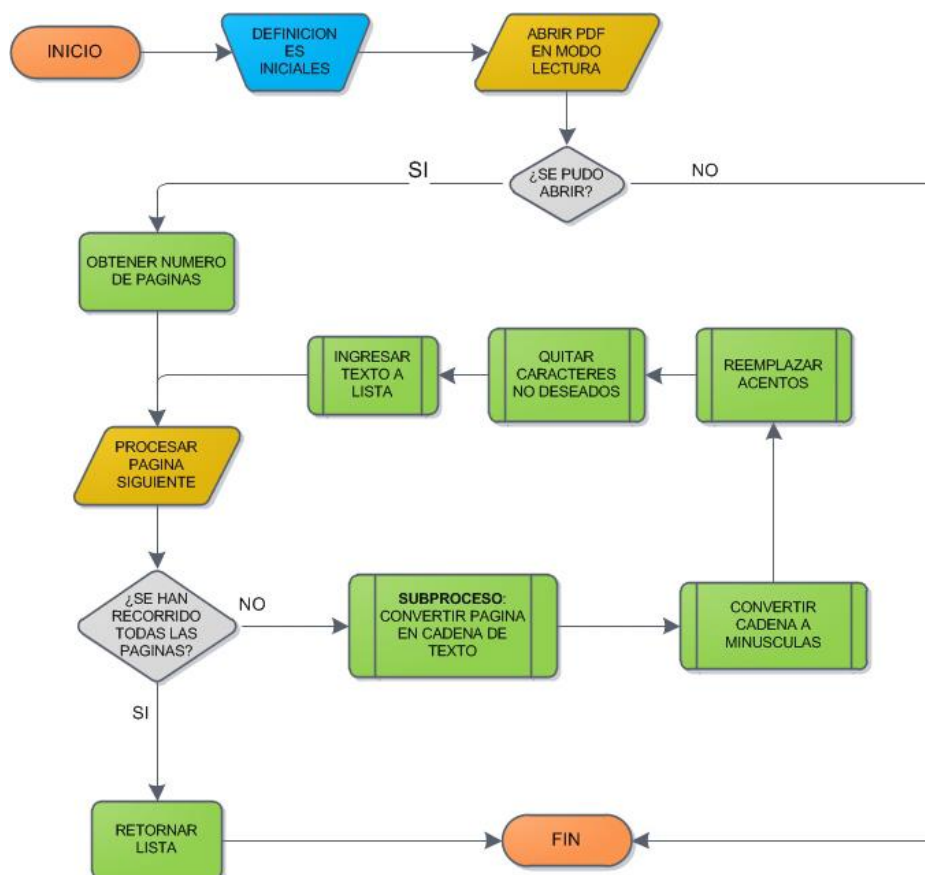
En esta fase del programa se establecen todos los datos esenciales para la ejecución, tales como la dirección del documento en formato PDF que se va a utilizar, el salto máximo que se probará en cada iteración del código secuencial, se inicializan todas las variables auxiliares que ayudan a una correcta conversión del código y a encontrar las coincidencias en el texto.

También, en esta fase del programa se importan los módulos anteriormente descritos, los cuales le dan funcionalidades especiales a cada etapa del código.

### 6.1.2. Conversión

Luego de ser ingresados los parámetros iniciales, comienza la etapa de conversión de la información contenida en los documentos que serán utilizados. Esta etapa define los elementos necesarios para almacenar como una cadena de caracteres (solo letras) que se extrae del PDF.

También se eliminan del texto todos los espacios, guiones, puntos, comas y cualquier elemento que no sea una letra. Los pasos que se ejecutan en esta etapa se ven ejemplificados mediante el siguiente diagrama de flujo:

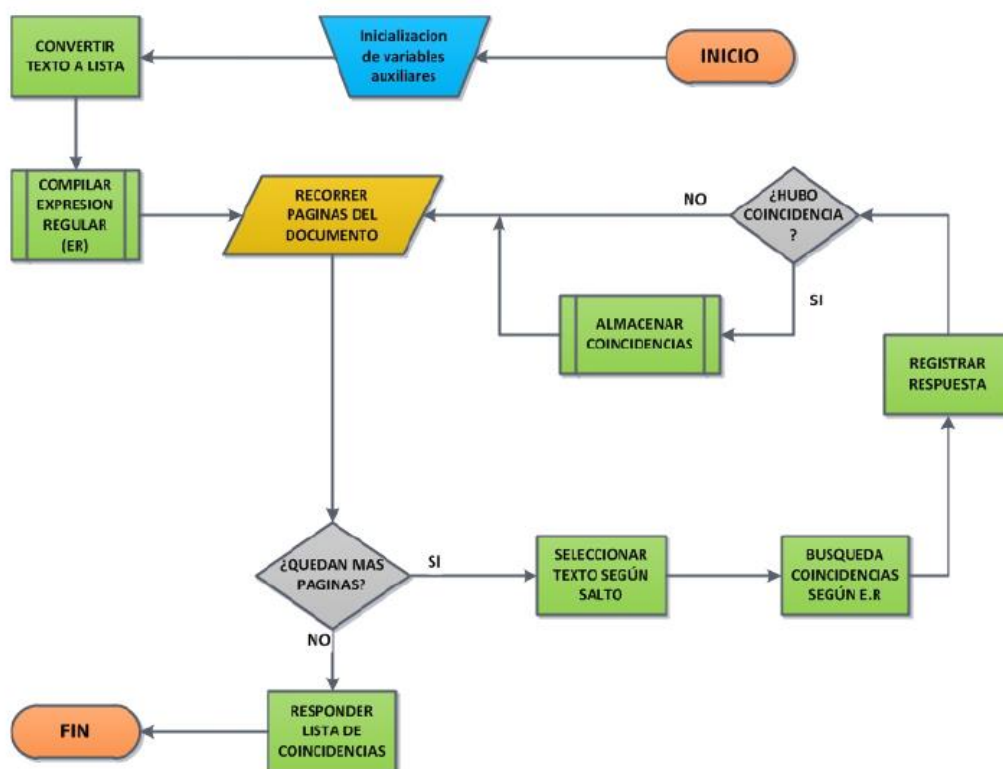


*Proceso de conversión de las cadenas de texto*

### 6.1.3. Búsqueda de Patrones

Esta parte del algoritmo creado es la más importante, ya que en esta se desarrolla la comparación fundamental del programa de cada una de las casillas que contienen el texto del documento.

Se aplica en esta parte el algoritmo de búsqueda de coincidencias en el texto como una secuencia equidistante de letras (ELS). Es también la parte que almacena en un arreglo todos los detalles de los resultados encontrados para en la última etapa mostrarlos al usuario. A continuación se muestra el diagrama correspondiente a esta etapa del proyecto:



*Proceso de búsqueda de patrones*

Cabe destacar también, que esta etapa del programa es altamente paralelizable, lo que se describirán posteriormente en las etapas a desarrollar.

#### 6.1.4. Resultados

Esta es la etapa final del programa secuencial en la cual los resultados son interpretados y mostrados al usuario del sistema por pantalla con todas las coincidencias encontradas en el documento PDF.

Se incluyen también los resultados que se obtienen buscando la misma palabra de forma inversa, por ejemplo: si la palabra que buscamos es “biblia”, los resultados serán respecto, también, a la palabra “ailbib”. De esta manera se abarcan todos los resultados posibles de coincidencias de la palabra en el texto según establece el Código de la Biblia.

#### 6.1.5. Interpretación de Resultados

Luego de que se haya finalizado la búsqueda de coincidencias en todo el documento, se retorna un arreglo con diccionarios de Python dentro de cada casilla. Cada diccionario contiene los atributos:

- Keyword: Palabra ingresada para buscar coincidencias.
- Página: Indica la página en donde se encontró la coincidencia.
- Salto: Es el salto con el cual se encuentra la coincidencia.
- Inicio: Es la posición del carácter en donde comienza la ‘keyword’ encontrada.
- Fin: Es la posición donde termina la ‘keyword’ encontrada.

### 6.2. Workflow del proceso paralelo

El usuario deberá ingresar al algoritmo el nombre del fichero con el texto a buscar (en formato PDF), el cual será transformado en una lista de cadenas de texto con el contenido de cada una de sus páginas. También se deberá ingresar la palabra o cadena de palabras (frase) que se buscará en el documento, cuyas ocurrencias se irán registrando a lo largo del algoritmo. El último valor que se deberá ingresar será el del máximo salto que se probará a lo largo del texto.

El algoritmo secuencial establece que este salto se debe ir recorriendo mediante iteraciones sucesivas desde 1 hasta dicho salto, luego se utilizarán las funciones implementadas de forma secuencial, es decir que para cada salto el algoritmo recorre el texto buscando patrones por todas las páginas que pueda tener el documento pdf.

Al arrancar se dividirá, dependiendo del número de procesadores esclavos disponibles, un intervalo de saltos que se le asignará a cada nodo para que cada uno individualmente los procese de la misma forma que lo hace la búsqueda de coincidencias de forma secuencial.

Una vez terminada la lectura del texto completo, cada nodo devolverá al maestro las coincidencias que encontraron en el texto, teniendo en cuenta los saltos con el cual se localizaron y donde inicia y termina cada ‘keyword’. El nodo maestro tendrá como objetivo reunir todas las respuestas y agruparlas en serie (en un archivo JSON) para luego en la siguiente etapa integrarlas a la plataforma web.



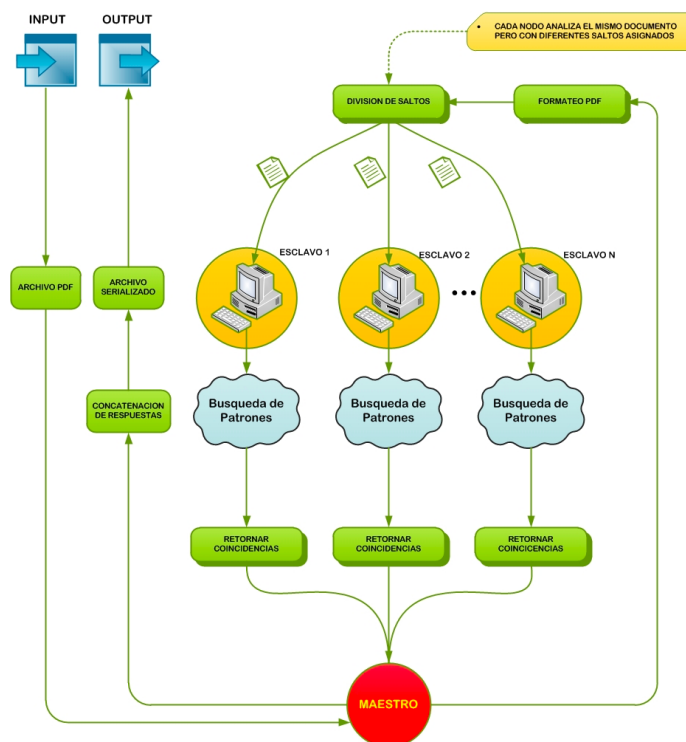


Diagrama representativo de la ejecución paralela del código de la biblia

## 8.6. Conclusiones

- En el gráfico de rendimiento que refleja el SpeedUp encontramos que el óptimo se encuentra en el uso de 80 procesadores, dado que en todos los casos de uso es con cual se alcanza la mayor aceleración.
- Para el caso del gráfico de eficiencia podemos determinar que la mayor encontrada es la que representa el uso de 16 procesadores, seguida por la de 80 procesadores. Podemos
- El menor costo asociado a las métricas reflejadas en su correspondiente gráfico se encuentra también en el uso de 80 procesadores, por lo que para las condiciones en las cuales las pruebas fueron medidas podemos concluir un óptimo para el uso de 80 procesadores.
- Las condiciones anteriores se ven totalmente alteradas al modificar la arquitectura de la plataforma, en el gráfico que representa el testing con las opciones de cluster implementadas observamos como el tiempo es casi 20 veces menor con nodos homogéneos y solo 1.2 veces menor en el caso de los nodos heterogéneos.

## 9. Valor agregado - Algoritmo Implícito

El algoritmo implícito, a diferencia del explícito, no necesita ingreso de patrones para realizar una búsqueda ya que este se basa en diccionarios definidos con una serie de palabras clave. Esto da la exploración de un universo mucho más extenso de coincidencias en los textos.

Este algoritmo es el más cercano a cómo trabaja el código de la biblia en sí, aunque la diferencia es que se ha optimizado para trabajar con cualquier clase de texto en formato PDF. También trabaja con intervalos de de saltos para lograr una iteración completa y poder ampliar el universo de soluciones obtenidas.

Debido a que el uso de diccionarios de palabras implica mucho costo en tiempo dependiendo del tamaño de este, al llevarlo a la plataforma paralela se logra trabajar con tiempos mucho más cortos.

### 9.1. Diccionarios ocupados

Para el estudio del algoritmo implícito se utilizaron 5 diccionarios distintos, los cuales se relacionan entre sí para la entrega de un patrón. Los diccionarios básicamente son una lista de palabras relacionadas a un tema común, cada una de ellas se encuentra guardada en un archivo de texto.

Diccionarios creados:

- nombres.txt: Contiene nombres de personajes históricos mundialmente conocidos.
- apellidos.txt: Contiene apellidos históricos, además de otros que comunes dentro del país.
- apocalipsis.txt: Contiene términos relacionados con el fin del mundo anunciado en la biblia.
- marca.txt: Contiene nombres de empresas o marcas presentes a nivel nacional.
- moda.txt: Contiene palabras de tendencia actual nacional.
- paises.txt: Contiene un listado de los países del mundo
- verbos.txt: Contiene una lista de verbos
- animales.txt: Contiene una lista de animales comunes.
- enfermedades.txt: Contiene un listado de enfermedades actuales.

## 9.2. Rendimiento

### 9.2.1. Casos de uso

Para las pruebas realizadas con el algoritmo implícito se ha escogido un texto relevante y mundialmente conocido para que los patrones encontrados causen una mayor impresión. Otro factor relevante al momento de escoger el texto fue la extensión del mismo, ya que esta búsqueda es la que cuenta con el tiempo más extenso para su ejecución, por lo que se buscó utilizar un documento que respondiera a estos requerimientos.

El detalle de las entradas que se ocuparon en estas pruebas son las siguientes:

- Título: Manifiesto Comunista
- Autor: Carl Marx
- Genero: Política
- Idioma: Español
- Número de páginas: 89
- Todos los diccionarios anteriormente dichos.
- 

### 9.2.2. Tabla de métricas

NP	Tiempos
16	759.13
32	536.325
48	440.036
64	375.413
80	163.548
96	393.575
112	1042.656
128	418.803

Tiempos de ejecución tomados con salto 100

### 9.2.3. Gráficos

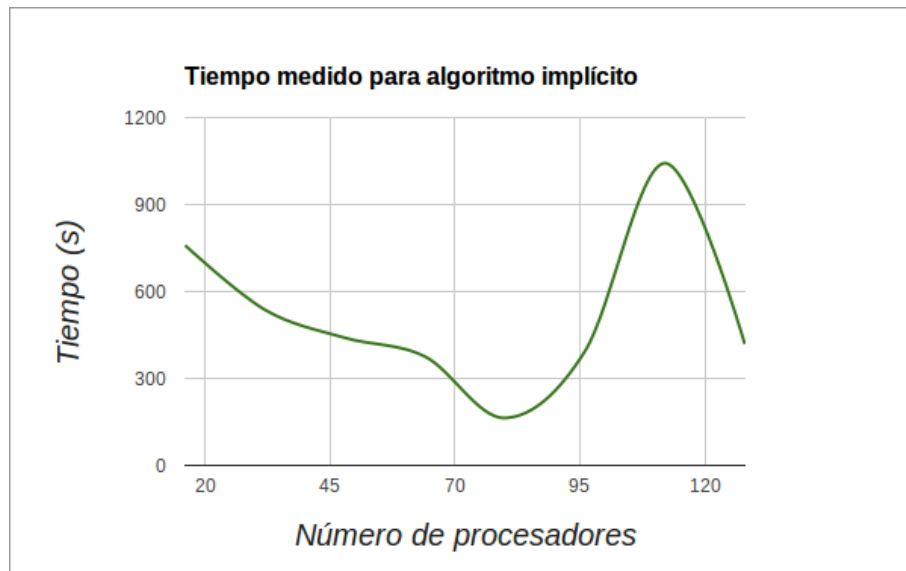


Gráfico representativo de la ejecución del algoritmo implícito

## 9.3. Estadísticas

### 9.3.1. Casos de Uso

Se ingresa un texto para analizarlo con una función de conteo, la cual retorna las repeticiones de cada letra a lo largo del mismo.

Para el conteo de caracteres definiremos 3 clasificaciones o grupos de letras, las de alta repetición (A), mediana repetición (M) y baja repetición (B), pues este es el indicador que nos refleja con mayor certeza la posibilidad de que cierta palabra clave aparezca o no dentro del texto. Es un factor relevante dado que se puede intuir fácilmente que una palabra compuesta por algún carácter inexistente será imposible de ubicar en la muestra analizada.

### 9.3.2. Análisis de PDF

Título: La evolución de las especies

Autor: Charles Darwin

Número de páginas: 1213

Cantidad de caracteres: 1016904

### 9.3.3. Tablas de métricas

Carácter	Coincidencias	Carácter	Coincidencias
a	114,947	n	74,783
b	10,115	o	89,220
c	45,291	p	28,279
d	52,801	q	9,429
e	143,327	r	64,752
f	8,835	s	91,551
g	11,778	t	45,884
h	9,399	u	42,805
i	66,668	v	9,895
j	2,647	w	178
k	124	x	2,506
l	47,984	y	7,932
m	33,020	z	2,754

*Tabla cuantitativa de cantidad de coincidencias por carácter*

Carácter	Coincidencias (%)	Carácter	Coincidencias (%)
e	14.1	i	6.6
a	11.3	r	5.4
s	9	d	5.2
o	8.8	l	4.7
n	7.4	t	4.5

*Tabla porcentual de los 10 caracteres más recurrentes en el texto*

Carácter	Coincidencias (%)	Carácter	Coincidencias (%)
v	0.78	z	0.271
q	0.869	j	0.26
h	0.924	x	0.246
f	0.927	w	0.018
y	0.973	k	0.012

*Tabla porcentual de los 10 caracteres menos recurrentes en el texto*

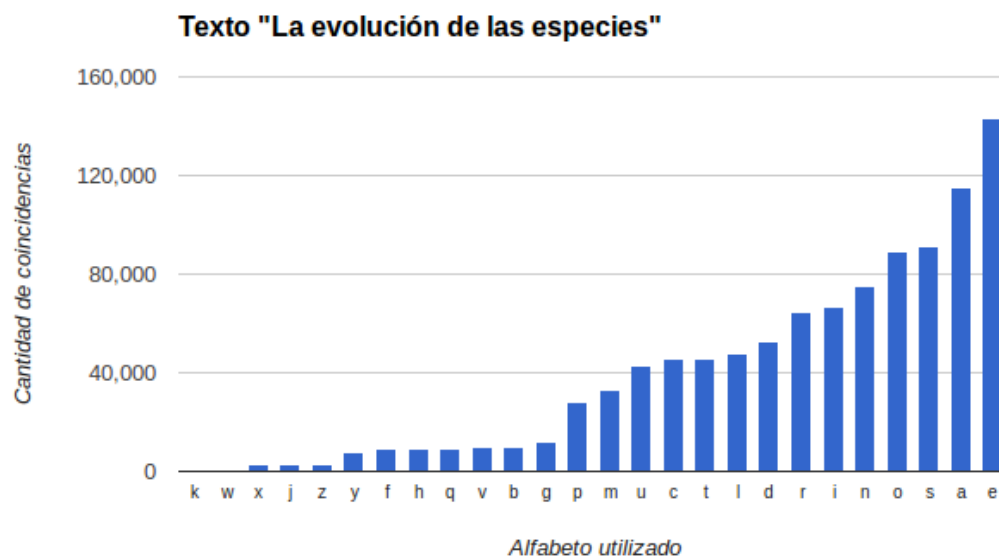
Keyword	Coincidencias	Grupo "alta"	Grupo "media"	Grupo "baja"
jesus	20	3	1	1
olimpo	9	4	2	0
satanas	7	7	0	0

*Tabla cuantitativa de coincidencias por palabra y clasificación de caracteres*

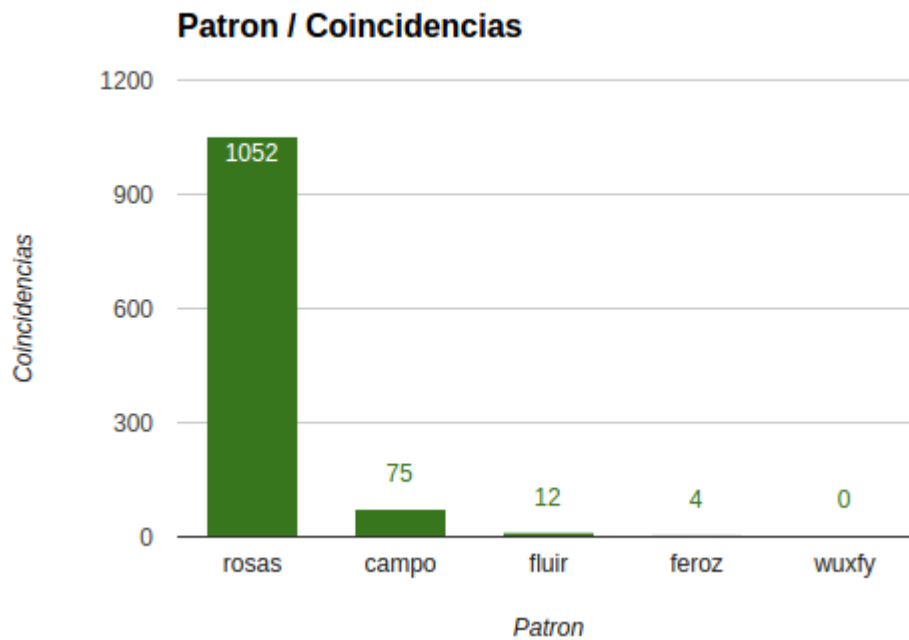
Patron	Coincidencias	Grupos
rosas	1052	<b>AAAAA</b>
campo	75	<b>MAMMA</b>
fluir	12	<b>BAMAA</b>
feroz	4	<b>BAAAB</b>
wuxfy	0	<b>BMBBB</b>

*Tabla de búsqueda de patrones según la frecuencia de caracteres*

### 9.3.4. Gráficos



*Conteo de letras en texto "La evolución de las especies"*



*Coincidencias de caracteres para patrones según las coincidencias*

### 9.3.5. Conclusiones

- A mayor cantidad de caracteres que tenga una palabra menos probable es encontrar coincidencias del mismo.
- A mayor cantidad de palabras en una frase también disminuye la probabilidad de éxito en la búsqueda de tales patrones, pero no tanto como la diferencia de la cantidad de caracteres.
- El éxito de encontrar coincidencias está condicionado por los caracteres con los cuales está constituida las palabras. Esto es debido a que en todo el texto hay diferentes proporciones de caracteres distintos. Por ese motivo si una palabra está formada por caracteres que tienen escasa participación en el texto, este tendrá pocas probabilidades de ser localizado debido a que ni siquiera puede existir.
- Para las pruebas bajo las condiciones entregadas de largo de palabra y las frecuencias de caracteres se observó que disminuyen las cantidades de coincidencias.

## Anexo K: Modulo Percolación

### 1. Resumen

Percolación es una de las problemáticas tratadas en la plataforma web de servicios, esta se refiere a la probabilidad que tiene un objeto de llegar de un extremo a otro, en nuestro caso, enfocado en la propagación de incendios, siendo el fuego nuestro objeto y los árboles como nuestro medio de propagación.

Para realizar esta simulación, se usara una plataforma clúster la cual está configurada por MPI y será de carácter heterogéneo. Se estimó que es necesario realizar 385 repeticiones de esta simulación para así tener un error menor al 5%. Una vez finalizada esta simulación, se enviara vía correo electrónico, un documento en formato PDF con las respuestas encontradas.

### 2. Introducción

Se creó un módulo de percolación perteneciente a la plataforma web, la cual permite la solución a dos problemáticas reales, las cuales son: propagación de incendios y propagación de enfermedades. Estas problemáticas son tratadas bajo la teoría de la percolación, que en palabras simples, es la probabilidad de cruzar desde un extremo a otro independiente del medio. La percolación entrega un umbral, el cual sirve para determinar si es factible o no el cruce por ese medio. Si la probabilidad de cruzar es menor al umbral de percolación, no será factible el cruce. Para este desarrollo, se contó con tres personas, Sergio Abarca, Rosa Gonzales y Claudio Piña.

### 3. Objetivos

#### 2.1 Objetivos Generales

Implementar una solución informática basada en un clúster computacional el cual deberá estar orientado al ámbito de procesamiento de un algoritmo paralelo que realice una búsqueda de un intervalo de confianza, con un nivel de significancia del umbral de percolación y que genere un resultado a través de una matriz.

#### 2.2 Objetivos Específicos

- Implementar una solución al problema utilizando el umbral de percolación.
- Implementar una solución secuencial y una paralela del problema de percolación.
- Analizar algoritmo secuencial y paralelo, analizando su comportamiento y obtener métricas de desempeño.
- Dar respuestas a los alcances de un incendio forestal.



#### 4. Marco teórico

Según la Real Academia Española, percolar es:

*(Del lat. percolāre, filtrar, colar). 1. intr. Dicho de un líquido: Moverse a través de un medio poroso.*

Para complementar esta definición. La percolación es un concepto físico-químico con el que se designa la filtración de un fluido en un medio poroso. Está determinado por la porosidad del medio. Cuando la porosidad es alta (por arriba de cierto umbral) el fluido se cuela por toda la estructura. Si por el contrario es baja, el líquido no pasa. Este fenómeno es considerado crítico, de todo o nada.

La teoría de la percolación consiste en la búsqueda y el análisis de modelos matemáticos para explicar el comportamiento aleatorio de un fluido en un medio poroso. Fue introducida por Broadbent y Hammersley en 1957. La percolación es el modelo más sencillo para un número considerable de fenómenos físicos, en los cuales el desorden está presente.

El modelo más sencillo para simular este tipo de fenómenos es considerar un retículo cuadrado infinito en el plano, dividido en rectas paralelas horizontales y verticales (una malla). Una molécula se desplaza en cada paso con una determinada probabilidad  $p$ ; si no, no se puede mover. Se estudia entonces la posibilidad de que partiendo del origen se tenga una trayectoria infinita. Ese comportamiento depende del valor de  $p$ , para el que existe un valor crítico (llamado de transición de fase). Esta teoría, de creciente interés en la actualidad, emplea conceptos geométricos y de cálculo de probabilidades.

Este algoritmo realizará 385 repeticiones. Se estimó usando inferencia estadísticas, que para tener un error menor al 5%, la cantidad mínima de repeticiones es la mencionada anteriormente. La fórmula utilizada es la siguiente:

$$N = \frac{Z^2 P(1 - P)}{E^2}$$

Dónde:

- N: tamaño de muestra que se desea calcular.
- Z: Es la desviación del valor medio que aceptamos para lograr el nivel de confianza deseado. En función del nivel de confianza que busquemos, usaremos un valor determinado que viene dado por la forma que tiene la distribución de Gauss. Para este problema se usó un nivel de confianza 95%, es decir,  $Z=1,96$ .
- P: Es la proporción que esperamos encontrar. Para este problema  $p = 0,5$ .
- E: Es el margen de error máximo admitido. Para este problema  $e=0,05$ .

Al ingresar todos los valores a la formula nos da como resultado: 384,16. Lo cual se aproxima al número entero siguiente, es decir, a 385.

Este módulo entrega como resultado, la probabilidad de propagación de incendios en un documento formato PDF que contiene los requerimientos, también entrega imagen de los 3 focos más importantes. Y la propagación de enfermedades, entrega un documento formato PDF que contiene la información si es factible considerar esa enfermedad como una epidemia o no.

## 5. Estado del arte

### 5.1 Aplicaciones de la percolación

- Contagio y propagación de epidemias en humanos.
- Diseminación de plagas en campos de cultivo.
- Propagación de incendios forestales.
- Estudio de redes eléctricas aleatorias y semiconductores.
- Simulaciones de la formación de estrellas.
- Estudio de terremotos y patrones de fallas.
- Reacciones químicas en medios porosos.
- Estudio de mercado publicitario.

Este grupo tiene por objetivo la modelación, creación y optimización de un algoritmo que sea capaz de entregar resultados sobre una simulación de propagación de incendio.

### 5.2 Percolación e incendios forestales

¿En qué condiciones un incendio se propaga en un bosque? Preguntando de otra forma, ¿cómo se establece conexión de punta a punta en un conjunto de elementos unidos parcial y aleatoriamente? Los científicos disponen de un término genérico para estos fenómenos: la percolación. El concepto fue introducido en 1956 por los ingleses J.M. Hammersley (matemático) y S.R. Broadbent (ingeniero).

La percolación en este problema juega un papel muy importante, debido a que nos permite simular hacia qué lugares se irá propagando este incendio, donde, un fuego sólo se puede propagar en el bosque si existen pocas regiones de este bosque sin árboles, de lo contrario solo afecta a los arboles más cercano, sin quemar al resto.

Cabe preguntarse si la probabilidad de cortar el incendio es proporcional a la superficie de las regiones sin árboles que tiene el bosque. Al contrario, sorprendentemente, la probabilidad que tiene un incendio forestal de ir de extremo a extremo del bosque es prácticamente nula por debajo de una cierta proporción de árboles, y es prácticamente del 100% por encima de este valor. Con esto, vemos que la percolación funciona según una ley de todo o nada. A lo que los físicos llaman un fenómeno de umbral.

Viéndolo más de cerca, con un ejemplo geométrico, cómo surge este umbral de percolación. Consideremos una red cuadrada, las casillas pueden ser azules o rojas; pero estos colores, en vez de estar regularmente distribuidos como en un tablero de ajedrez, son elegidos al azar. La proporción de casillas azules es  $p$ , la de casillas rojas  $1-p$ . Nuestra red presenta grupos azules y grupos rojos de variados tamaños. Concentrémonos en los grupos rojos. Cuando las casillas azules son muy minoritarias, es muy poco probable que aparezca un grupo grande. Pero cuando la proporción de casillas azules es lo bastante alta, se encuentran grupos cada vez mayores, y algunos de ellos atraviesan toda la red (aunque ésta sea muy grande). Así, al aumentar la proporción  $p$  de casillas azules, el tamaño de los grupos crece brutalmente: el grupo mayor percola cuando  $p$  rebasa el valor umbral  $p_u$ . El valor crítico que marca el umbral, la frontera, entre los grupos disjuntos y el grupo percolante depende, de la forma de la red. Si ésta está formada por triángulos, y no por cuadrados, el resultado será distinto. También será distinto si en vez de construir los grupos coloreando casillas y reuniendo casillas del mismo color

trazamos puentes entre casillas elegidas al azar (en tal caso lo que cuenta es la proporción de casillas unidas a sus vecinas).

El tamaño y la forma del grupo mayor, el que percola, siempre son muy irregulares. No obstante, los investigadores han encontrado notorias semejanzas entre los distintos ejemplos en las proximidades de la transición. Expliquémonos. Se trata de ciertas características de los grupos cuando la proporción de casillas azules y rojas se aproxima al umbral de la transición. En promedio, el tamaño del grupo mayor aumenta al hacerlo el número de casillas azules de acuerdo con una ley matemática muy simple: es inversamente proporcional a la distancia en relación al umbral de la proporción casillas azules/casillas rojas, es decir, a  $(p-p_u)$ , elevada a una cierta potencia, que los científicos llaman exponente crítico.

Más aún: los investigadores han demostrado que este exponente crítico es muy general, ya que toma el mismo valor cualquiera que sea el tipo de percolación considerado. Otras varias propiedades de los grupos varían de modo análogo en las cercanías del umbral, como por ejemplo la probabilidad de que una casilla elegida al azar esté en el grupo mayor el que va a percolar o la probabilidad de que dos casillas pertenezcan al mismo grupo. Así, en las cercanías del umbral de percolación, las diferencias entre los distintos sistemas se difuminan y surge un comportamiento universal. Las propiedades ya no dependen del detalle de la estructura, sino que obedecen a unas leyes globales llamadas leyes de escala. Pero aunque esta universalidad está bien establecida, todavía no se ha dado de ella ninguna formulación matemática completa y para los investigadores no siempre es obvio cómo distinguir a priori en un sistema cuales son las magnitudes universales y cuales los detalles que se pueden despreciar. Esta cuestión es el tema de buena parte de los trabajos sobre la percolación. Un aspecto al que el fenómeno es ciertamente sensible es el siguiente

Otro exponente universal que aparece muy cerca del umbral. Justo por encima de éste, la velocidad es proporcional a la distancia al umbral elevada a una potencia de aproximadamente 0,17. En el umbral exacto, claro está, la velocidad tiende a cero porque el grupo percolante desaparece y con él la posibilidad de transmisión. Este modelo se aplica, por ejemplo, a una epidemia en la que los enfermos mueren o se inmunizan, o en todo caso a una epidemia que agota su medio y que por lo tanto tiene que propagarse para subsistir. Un fenómeno análogo se produce en el crecimiento de ciertas especies de setas.

Para ilustrar las características principales del fenómeno de la percolación, basta examinar un objeto geométrico muy simple una red de casillas cuadrados coloreadas de azul y de rojo. En (A), el color de cada casilla se elige al azar y hoy tantas casillas azules como casillas rojas. Ambos tipos de casillas forman grupos más o menos irregulares (las casillas de igual color que no comparten una arista no se cuentan como pertenecientes al mismo grupo). Uno de los grupos azules, a modo de ejemplo se ha coloreado con un color más claro. Ningún grupo azul se extiende de punta a punta de la red: la proporción de casillas azules no es suficiente para que un grupo percole. La probabilidad de hallar un grupo de casillas azules que se extienda de punta a punta de la red no aumenta regularmente con la proporción de casillas azules por debajo de un cierto umbral, esta probabilidad es prácticamente nula. Vemos en (B) un motivo formado con un 58 % de casillas azules. Pintamos con un color más claro tres de los grupos azules de mayor tamaño. Ninguno de ellos percola. Aumentando un poco más todavía la proporción de casillas azules se sobrepasa el umbral (que según la teoría, es de aproximadamente

59,27 %) y se está prácticamente seguro de encontrar un grupo que percola. Es lo que vemos en (C), donde la proporción de casillas azules es de 60 %.

Un último problema de cara a la comparación entre teoría y experiencia con sistemas en que las predicciones teóricas sólo son válidas en las proximidades inmediatas del umbral. Si el sistema observado está lejos del umbral, entonces la teoría no predice nada en absoluto.

Propagación de un incendio forestal se puede abordar por medio del concepto de percolación, es importante tener en cuenta el agotamiento de los recursos del medio por el fenómeno: los árboles se queman, lo cual, no puede el fuego volver inmediatamente a lugares ya alcanzados. Pero al cabo de un cierto tiempo los árboles vuelven o crecer, con lo cual el fuego reaparece.

Podemos imaginar dos situaciones límite. La primera corresponde al caso en que los árboles no vuelven a crecer en la tierra quemada y el incendio se propaga según lo que hemos llamado proceso de epidemia general. La segunda corresponde al caso en que los árboles vuelven a crecer muy deprisa, antes incluso de que el fuego haya acabado de propagarse (lo cual no es demasiado realista y se aplica más bien a una epidemia); el incendio puede volver a sitios anteriormente visitados y estamos en el caso de la epidemia simple, unida a la percolación dirigida. Entre estos dos extremos, la nueva situación que nos interesa aparece cuando los árboles vuelven a crecer poco a poco. Lentamente en lugares aislados, esperando en cierto modo a que los árboles hayan crecido lo bastante en los lugares ya visitados para volver a quemarlos. La dinámica de la propagación implica así un delicado equilibrio entre la extinción y la propagación.

¿Habrán dado por fin los investigadores con el buen modelo? Subsisten pues muchas cuestiones abiertas en la teoría de la percolación. Pero, pese a las dificultades de la teoría pura, ésta sigue teniendo un gran adelanto sobre las aplicaciones. Las principales dificultades son de dos órdenes: en primer lugar, es difícil proceder a experimentos conformes a las condiciones teóricas, en las cuales la escala microscópica, en la que se produce la transmisión entre elementos de la red, es muy pequeña con respecto a la escala macroscópica, la de la red entera. Por otra parte, los aspectos dinámicos son todavía muy difíciles de caracterizar en la práctica, y en este campo quedan muchos progresos metodológicos por hacer. No habría que concluir de todo ello que la teoría es inútil. Tiene un indiscutible poder predictivo y ha logrado grandes éxitos.

Ha tenido también el mérito de forjar unos conceptos generales aplicables a fenómenos muy variados de la incumbencia de distintas ciencias. Tal vez sea este teniendo un gran adelanto sobre el aspecto unificador lo que hace de la percolación un fenómeno tan fascinante.

## 6. Descripción del modelo de solución

Aproximemos el bosque o la población por una red cuadrada en la que cada celda está ocupada por un árbol o persona con probabilidad  $p$ , o vacía con probabilidad  $p - 1$ . Esta distribución de objetos y huecos será el estado inicial. Evidentemente, si  $p$  es pequeño es de esperar que los objetos estén en grupos bien distribuidos y poco conectados entre sí. En cambio, si  $p$  es próximo a 1, casi todas las celdas estarán ocupadas por un objeto, el bosque se extenderá de lado a lado de la red y su alta conectividad hará que acabe por propagar casi completamente. Se sabe que existe un valor crítico de  $p$ , ( $p_c$ ), tal que si  $p > p_c$  y en el límite de red infinita, el fuego se extiende por toda la red (lo que no quiere decir que se propague por toda la región), mientras que si  $p < p_c$  sólo afecta a una porción de él. Se define pues  $p_c$  como aquel valor de  $p$  a partir del cual, en una red infinita, aparecen agregados de percolación, es decir, agregados que se extienden a toda la red.

Primero se definirá el tamaño de la matriz, luego la cantidad de repeticiones de la prueba para obtener un  $p_c$  promedio y así compararlo con las condiciones iniciales ( $p_c$  mayor o menor que  $p$ ). Se utilizará la percolación para encontrar cuáles serán los árboles afectados y hasta qué punto llegará dicho incendio forestal, lo mismo sucederá con la propagación de enfermedades.

El algoritmo toma por parte del usuario los siguientes parámetros:

- Tipo de árbol.
- Tipo de suelo.
- Cantidad de metros.
- Distribución porcentual del espacio.

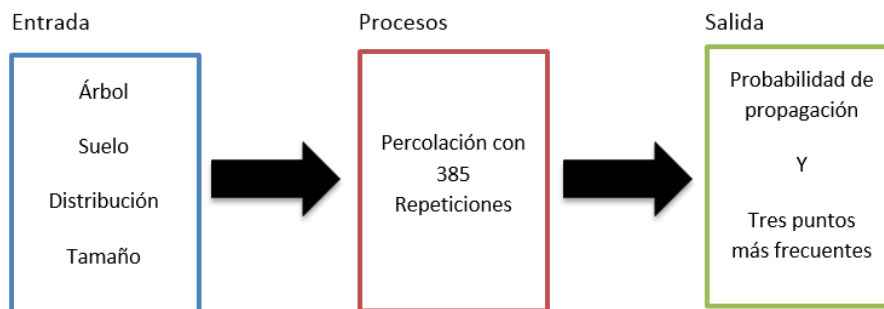
Con esta información, entra al algoritmo de percolación junto a los siguientes parámetros:

- Porcentaje de combustión del árbol.
- Porcentaje de combustión del suelo.

## 7. Workflow propagación de incendios forestales

### 7.1 Modelo secuencial

El desarrollo del algoritmo involucra distintas etapas para su correcta ejecución. Este, involucra los pasos agrupados que se presentarán a continuación:



**Selección de tipo de árbol:** Esta etapa involucra los distintos tipos de árboles para realizar la simulación y demostrar qué tanto se puede propagar el fuego, ya que no todos cuentan con el mismo % de combustión. Los árboles a trabajar son:

- Quillay con un 61% de combustión.
- Peumo con un 65% de combustión.
- Boldo con un 78% de combustión.
- Roble con un 90% de combustión.
- Raulí con un 80% de combustión.

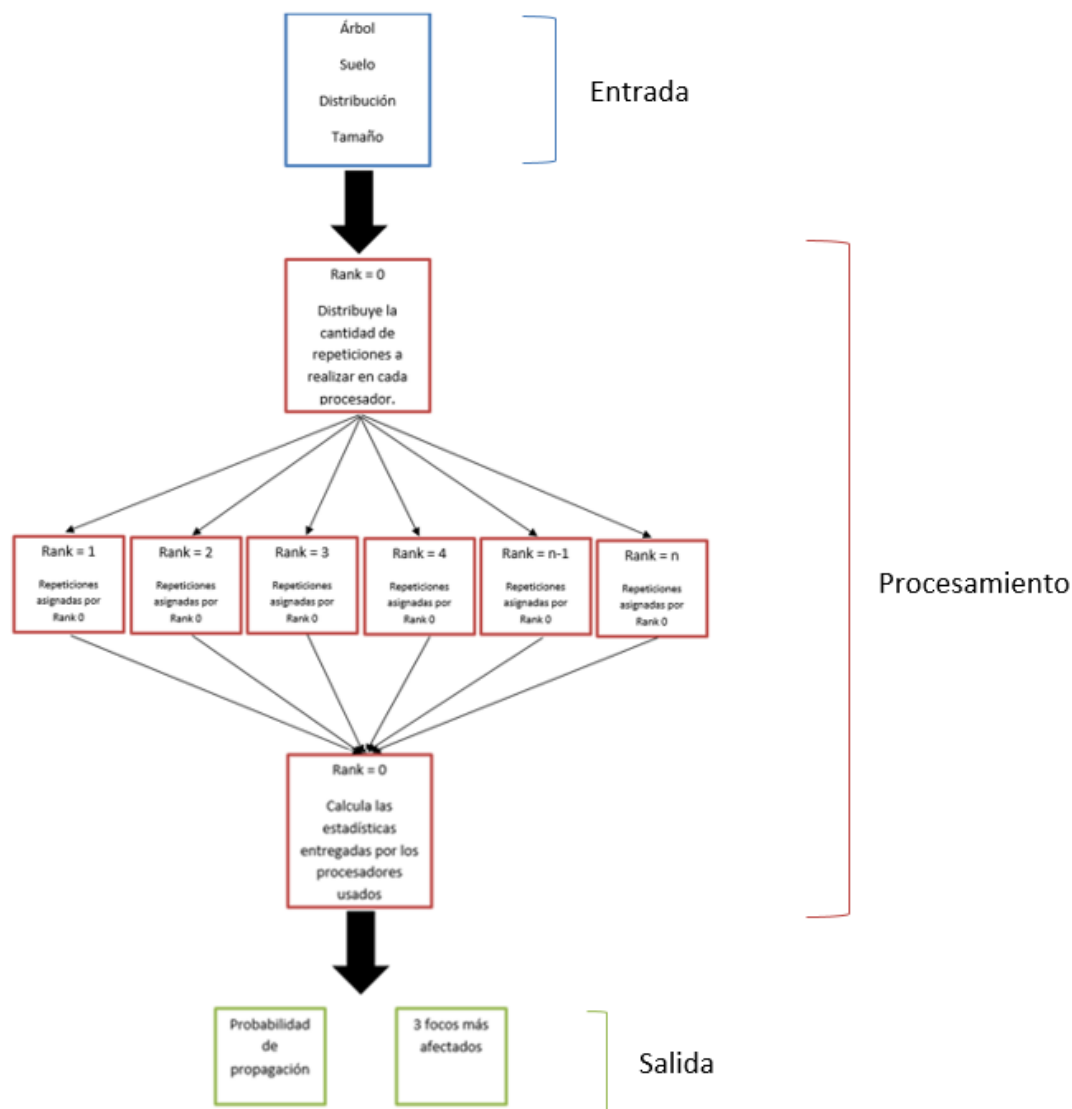
**Selección de tipo de suelo:** Esta etapa involucra el tipo de suelo, que también incide en la propagación del fuego, ya que influye en qué medida puede continuar avanzando el fuego, si es que se lo permite. Los tipos de suelo a trabajar son:

- Serranías áridas y semiáridas con un 97% de combustión.
- Graníticos de la costa con un 80% de combustión.
- Vertisoles con un 53% de combustión.
- Aluviales del valle central con un 78% de combustión.

**Distribución:** Distribución porcentual del espacio, indica qué tan separados o cerca están los árboles unos de otros en relación a la cantidad de metros.

**Tamaño:** Indica el área total en la que se está trabajando, se toman como unidad de medida, 1 cuadrado de la matriz = 1 hectárea.

## 7.2 Modelo paralelo



El procesador maestro es quien recoge la información del usuario, repartiendo la información e indicando cuantas repeticiones debe hacer cada procesado, una vez terminado los procesadores esclavos, el procesador maestro recopila la información y calcula las estadísticas, para así entregar los 3 puntos más frecuentes en que se puede iniciar el incendio y la probabilidad de propagación de estos.

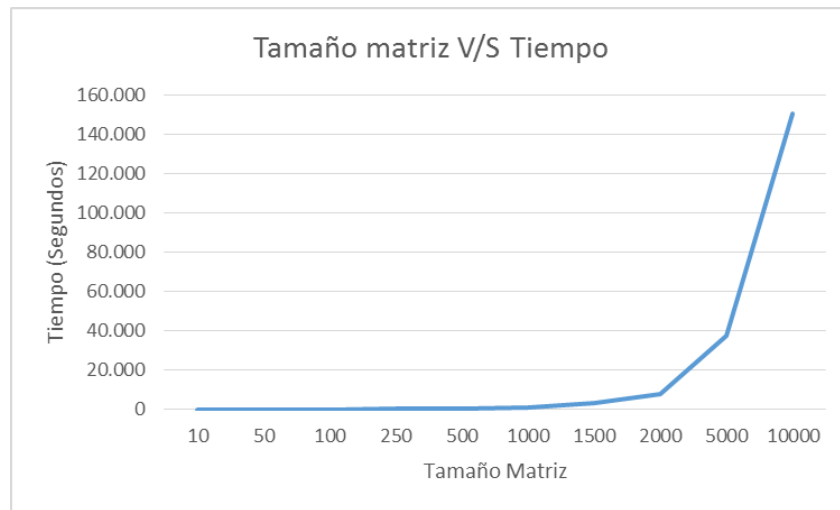
## 8. Análisis de Desempeño y Optimización

### 8.1 Estadísticas

#### Secuencial

Tamaño Matriz	Tiempo (segundos)
10	0,14914
50	2,38995
100	9,47990
250	67,96358
500	290,57180
1000	1142,07671
1500	3403,16735
2000	7542,09386
5000	37633,59334
10000	150380,62149

Tabla 1. Tiempos ejecución secuencial.  
Elaboración propia.



Al obtener los tiempos del código secuencial, podemos apreciar que tiene un comportamiento exponencial, de hecho, desde el gráfico se puede obtener la función de comportamiento la cual es:

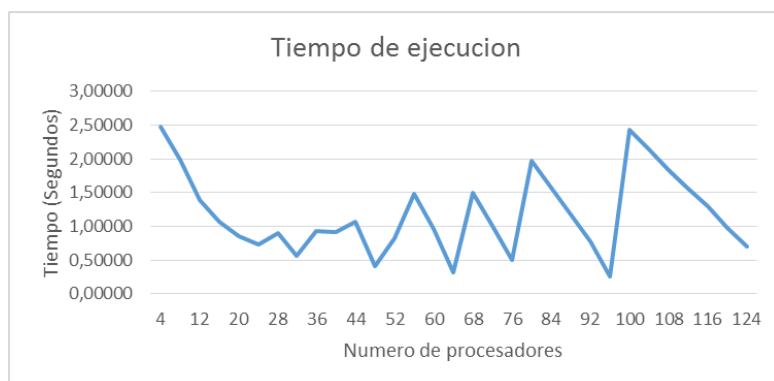
$$y = 0,1124e^{1,4459x}$$



## Paralelo con matriz de tamaño 100

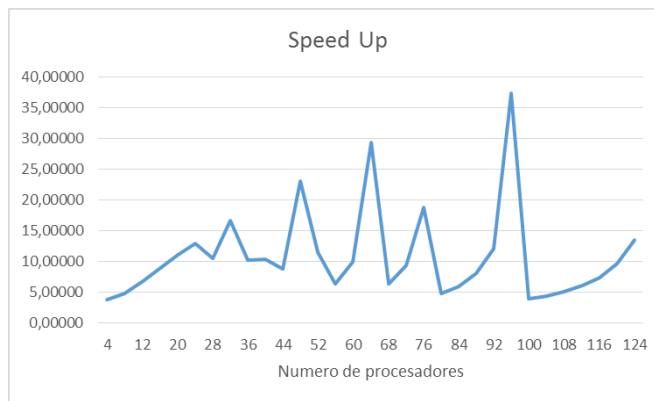
N° procesadores	Tiempo (Segundos)	Speed Up	Eficiencia	Costo
4	2,48255	3,81861	0,95465	0,62064
8	1,98546	4,77466	0,59683	0,24818
12	1,39096	6,81538	0,56795	0,11591
16	1,06168	8,92912	0,55807	0,06636
20	0,85846	11,04296	0,55215	0,04292
24	0,73434	12,90947	0,53789	0,03060
28	0,89817	10,55473	0,37695	0,03208
32	0,56962	16,64261	0,52008	0,01780
36	0,92527	10,24559	0,28460	0,02570
40	0,91091	10,40703	0,26018	0,02277
44	1,06932	8,86533	0,20148	0,02430
48	0,41041	23,09865	0,48122	0,00855
52	0,82031	11,55643	0,22224	0,01578
56	1,48513	6,38323	0,11399	0,02652
60	0,95732	9,90258	0,16504	0,01596
64	0,32262	29,38423	0,45913	0,00504
68	1,49546	6,33911	0,09322	0,02199
72	1,00669	9,41688	0,13079	0,01398
76	0,50526	18,76236	0,24687	0,00665
80	1,96712	4,81917	0,06024	0,02459
84	1,57464	6,02037	0,07167	0,01875
88	1,17697	8,05448	0,09153	0,01337
92	0,78301	12,10692	0,13160	0,00851
96	0,25387	37,34164	0,38898	0,00264
100	2,43469	3,89367	0,03894	0,02435
104	2,14301	4,42364	0,04254	0,02061
108	1,83404	5,16885	0,04786	0,01698
112	1,55108	6,11182	0,05457	0,01385
116	1,29224	7,33602	0,06324	0,01114
120	0,97516	9,72135	0,08101	0,00813
124	0,70254	13,49372	0,10882	0,00567

Tabla 2. Tiempos ejecución paralelo matriz tamaño 100 Elaboración propia.



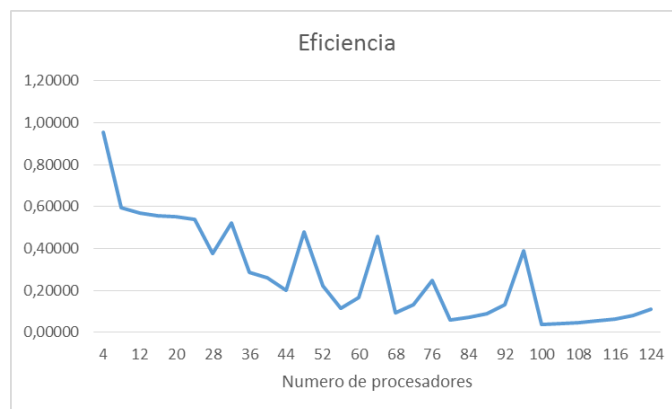
Al ver el grafico podemos apreciar que el menor tiempo de ejecución se encuentra al utilizar 96 procesadores.

Grafico 2-1. Tiempos ejecución paralelo, matriz tamaño 100. Elaboración



El grafico de Speed Up nos muestra que el procesador óptimo es 96, esta información explica por qué al utilizar el nodo 96 se encuentra el menor tiempo de ejecución.

Grafico 2-2. Speed Up, matriz tamaño 100. Elaboración propia.



La eficiencia se encuentra al utilizar 4 procesadores, pero también es posible encontrar un alto número de eficiencia en los procesadores 48,64 y 96.



El costo se reduce bastante a en el procesador 96, no se logra apreciar bien en el gráfico, pero al revisar la tabla, podemos ver que es el procesador con menor costo.

Grafico 2-4. Costo, matriz tamaño 100. Elaboración propia.

## Paralelo con matriz de tamaño 1000

N° procesadores	Tiempo (Segundos)	Speed Up	Eficiencia	Costo
4	311,10135	3,67108	0,91777	77,77534
8	217,85192	5,24244	0,65531	27,23149
12	155,03276	7,36668	0,61389	12,91940
16	115,16685	9,91671	0,61979	7,19793
20	92,80215	12,30658	0,61533	4,64011
24	77,55559	14,72591	0,61358	3,23148
28	111,01886	10,28723	0,36740	3,96496
32	58,35892	19,56987	0,61156	1,82372
36	114,13921	10,00600	0,27794	3,17053
40	110,97567	10,29124	0,25728	2,77439
44	134,17492	8,51185	0,19345	3,04943
48	40,00006	28,55187	0,59483	0,83333
52	90,98662	12,55214	0,24139	1,74974
56	176,83009	6,45861	0,11533	3,15768
60	101,16688	11,28904	0,18815	1,68611
64	30,00008	38,06913	0,59483	0,46875
68	163,95615	6,96574	0,10244	2,41112
72	97,74769	11,68392	0,16228	1,35761
76	32,40725	35,24139	0,46370	0,42641
80	226,71838	5,03742	0,06297	2,83398
84	170,62525	6,69348	0,07968	2,03125
88	121,47308	9,40189	0,10684	1,38038
92	68,40722	16,69527	0,18147	0,74356
96	20,65078	55,30428	0,57609	0,21511
100	289,27081	3,94812	0,03948	2,89271
104	244,54357	4,67024	0,04491	2,35138
108	211,20356	5,40747	0,05007	1,95559
112	170,50004	6,69840	0,05981	1,52232
116	130,22274	8,77018	0,07560	1,12261
120	97,61242	11,70012	0,09750	0,81344
124	64,42974	17,72593	0,14295	0,51959

Tabla 3. Tiempos ejecución paralelo matriz tamaño 1000  
Elaboración propia.



Al analizar los tiempos de ejecución, se puede observar que nuevamente al usar 96 procesadores, se obtiene el menor tiempo de ejecución.

Gráfico 3-1. Tiempos ejecución paralelo, matriz tamaño 1000. Elaboración

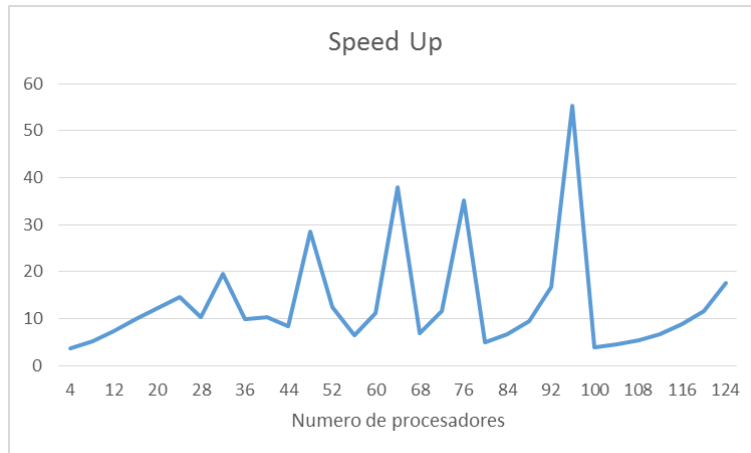


Grafico 3-2. Speed Up, matriz tamaño 1000. Elaboración propia.

Con unas curvas más pronunciadas, se puede observar claramente que al usar 96 procesadores es donde se obtiene mejor Speed Up.

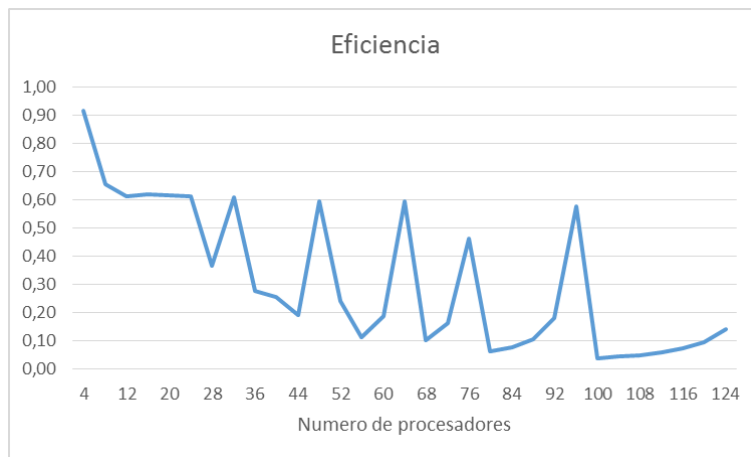


Grafico 3-3. Eficiencia, matriz tamaño 1000. Elaboración propia.

Al comparar los resultados con las métricas tomadas al usar una matriz de tamaño 100, podemos observar que la eficiencia se encuentra en los nodos 32, 48, 64, 76 y 96.

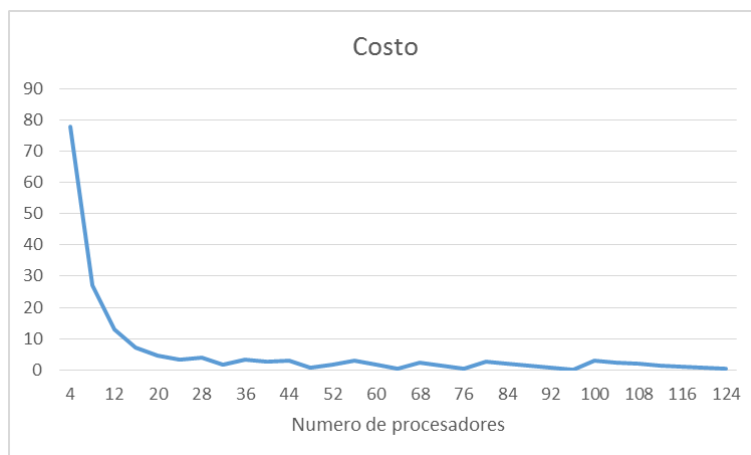


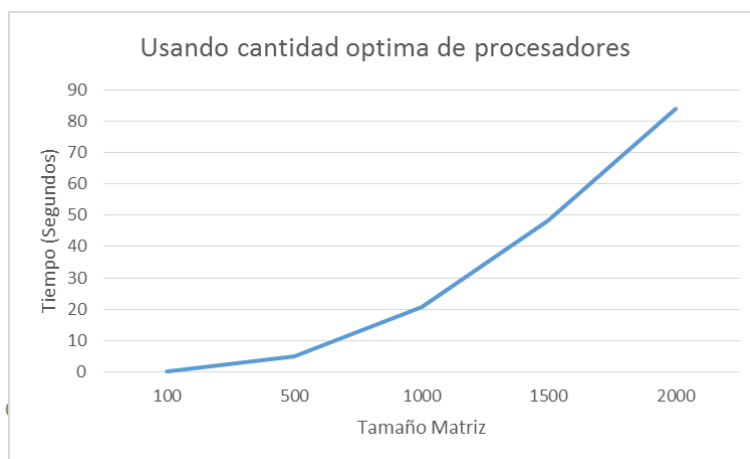
Grafico 3-4. Costo, matriz tamaño 1000. Elaboración propia.

Nuevamente el costo menor se obtiene al usar 96 procesadores de nuestra plataforma.

## Nodo optimo

Tamaño matriz	Tiempo (Segundos)
100	0,253869375
500	4,827818964
1000	20,65078497
1500	48,16490698
2000	83,95333004

Tabla 4. Tiempos ejecución paralelo usando 96 procesadores. Elaboración propia.



Realizando pruebas utilizando solo la cantidad optima de procesadores, es decir, ejecutando el programa con 96 procesadores y distintos tamaños de matrices, podemos ver que tiene un comportamiento de tipo polinómica de orden 2. Esto queda demostrado con la siguiente formula:

$$y = 5,2943X^2 - 10,692X + 5,4094$$

## Pruebas clúster heterogéneo

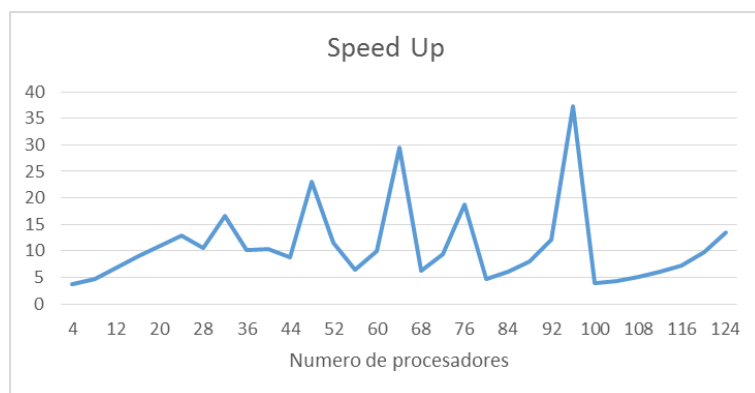
N° procesadores	Tiempo (Segundos)	Speed Up	Eficiencia	Costo
4	2,48255	3,81861	0,95465	0,62064
8	1,98546	4,77466	0,59683	0,24818
12	1,39096	6,81538	0,56795	0,11591
16	1,06168	8,92912	0,55807	0,06636
20	0,85846	11,04296	0,55215	0,04292
24	0,73434	12,90947	0,53789	0,03060
28	0,89817	10,55473	0,37695	0,03208
32	0,56962	16,64261	0,52008	0,01780
36	0,92527	10,24559	0,28460	0,02570
40	0,91091	10,40703	0,26018	0,02277
44	1,06932	8,86533	0,20148	0,02430
48	0,41041	23,09865	0,48122	0,00855
52	0,82031	11,55643	0,22224	0,01578
56	1,48513	6,38323	0,11399	0,02652
60	0,95732	9,90258	0,16504	0,01596
64	0,32262	29,38423	0,45913	0,00504
68	1,49546	6,33911	0,09322	0,02199
72	1,00669	9,41688	0,13079	0,01398
76	0,50526	18,76236	0,24687	0,00665
80	1,96712	4,81917	0,06024	0,02459
84	1,57464	6,02037	0,07167	0,01875
88	1,17697	8,05448	0,09153	0,01337
92	0,78301	12,10692	0,13160	0,00851
96	0,25387	37,34164	0,38898	0,00264
100	2,43469	3,89367	0,03894	0,02435
104	2,14301	4,42364	0,04254	0,02061
108	1,83404	5,16885	0,04786	0,01698
112	1,55108	6,11182	0,05457	0,01385
116	1,29224	7,33602	0,06324	0,01114
120	0,97516	9,72135	0,08101	0,00813
124	10,72804	13,49372	0,10882	0,08652

Tabla 5. Tiempos ejecución paralelo matriz tamaño 100, clúster heterogéneo. Elaboración propia.



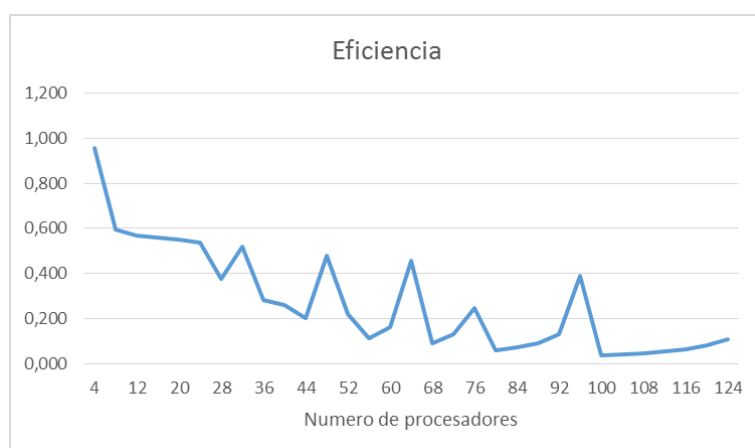
Gráfico 5-1. Tiempo ejecución paralelo clúster heterogéneo, tamaño matriz 100. Elaboración propia.

Al cambiar un nodo del clúster, podemos ver que cuando se utiliza esos procesador, los tiempos de ejecución son un casi 16 veces más lentos que el clúster homogéneo



El mejor Speed up sigue siendo con 96 procesadores, pero disminuyo al usar los computadores con menores capacidades, los cuales corresponden a los procesadores 22 - 24

Grafico 5-2. Speed Up clúster heterogéneo, tamaño matriz 100 Elaboración propia.



La eficiencia se mantiene de la misma forma, por qué estos computadores se usan al 100%, son más lentos pero se usan en su totalidad.

Grafico 5-3. Eficiencia clúster heterogéneo, tamaño matriz 100 Elaboración propia.



El costo de usar estos procesadores sube considerablemente. Es por esto que solo se usó para realizar pruebas.

Grafico 5-4. Costo clúster heterogéneo, tamaño matriz 100 Elaboración propia.

## 9. Valor agregado

Usando la misma plataforma, podemos realizar simulación de propagación de enfermedades, las cuales contemplan el mismo método de simulación que la propagación de incendios forestales, solo se ajustan los parámetros de entradas y elimina la creación de imágenes. El documento de respuesta de PDF solo contiene texto.

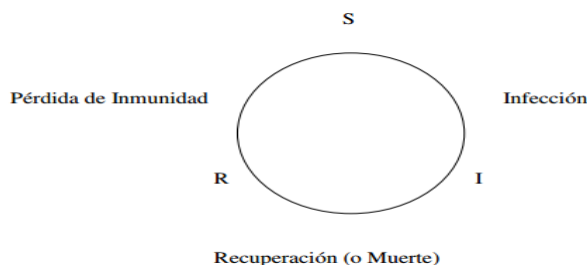
### Percolación y propagación de enfermedades

Al margen del origen netamente inorgánico de los procesos que estudió inicialmente la teoría percoladora, la analogía que ellos muestran con la transmisión de enfermedades no resulta difícil de establecer a primera vista. Del mismo modo en que la "gelación" se dispara a partir de cierto aumento de vínculos y que una malla puede pasar de un estado de aislamiento y desconexión de sus componentes a otro de conexión genérica, una enfermedad se puede difundir dramáticamente a partir de que cierta masa crítica de contagios (el umbral epidemiológico) es alcanzada dentro de una población.

Prescindiendo de este primer contexto de aplicación de tales ideas, completamente extraño a la medicina y a las ciencias sociales, podemos decir que la mayoría de las aproximaciones actuales a la difusión del contagio en las ciencias médicas fueron desarrolladas originalmente por el químico escocés William Ogilvy Kermack y el matemático A.G. McKendrick en dos artículos de los años 1927 y 1932, pero fueron recuperados y valorados en una fecha tan tardía como 1979, cuando un conocido artículo de Anderson y May en la revista "Nature" los transformó en el punto de partida de casi todas las modelizaciones contemporáneas.

¿Qué sostiene básicamente este modelo epidemiológico? Los componentes de este modelo original, conocido como SIR, son bastante sencillos en su formulación más típica. Por un lado hay una categoría poblacional que es la de los susceptibles de ser infectados (S o "susceptible" en inglés), por otro lado están los infectados en sí (I o "infectious") y por últimos los recuperados de un proceso infeccioso (R o "recovered"). Una vez que la infectada toma contacto con un grupo de individuos susceptibles, el porcentaje de individuos afectados aumenta dramáticamente (fase de crecimiento logístico) en función de medidas que son inherentes a la enfermedad en cuestión: la tasa de recuperación y la probabilidad de que un contacto entre un S y un I devenga en contagio (infecciosidad). Al desarrollar las ecuaciones que corresponden a este modelo encontramos que en cada una de las fases el número de contagiados será función del volumen de cada una de las tres poblaciones: pequeño inicialmente, a partir de un punto de inflexión crecerá con rapidez y luego, cuando el número de infectados sea menor que el número de recuperados, se estabilizará y caerá de nuevo. El modelo SIR produce entonces 3 fases típicas aplicables a cada enfermedad: arranque, explosión y remisión.

Los estados del SIR clásico





Es fácil probar que estos modelos funcionan bastante ajustadamente en medios abiertos y en donde la posibilidad de contagio es relativamente aleatoria. Si pensamos en la transmisión de virus informáticos, vigente en un medio en donde todos los usuarios están potencialmente vinculados entre sí, o en el mismo aire, para el caso de una epidemia de gripe, veremos que ambos contextos satisfacen los requisitos de contacto al azar que resultan tan imprescindibles para su validación. Sin embargo una serie de preguntas surgen apenas tomamos contacto con estas explicaciones y sus consecuencias teóricas más generales. ¿Cuál es su capacidad proyectiva? ¿Son aplicables a todas las situaciones de contagio?

El modelo SIR asume que aproximadamente toda la población está homogéneamente distribuida, es decir que todos los individuos (nodos) tienen aproximadamente el mismo número de contactos en el mismo tiempo, y que todos los contactos transmiten la enfermedad con la misma probabilidad. Consideremos un par de individuos que están conectados, y que  $i$  está infectado y que  $j$  es susceptible. Supongamos que la tasa promedio para causar la enfermedad es  $r_{ij}$ , y que el sujeto infectado permanece infectado por un tiempo  $\tau_i$ . Entonces la probabilidad  $1 - T_{ij}$  de que la enfermedad no sea transmitida de  $i$  a  $j$  es:

$$1 - T_{ij} = \lim_{\delta t \rightarrow 0} (1 - r_{ij} \delta t)^{\tau_i / \delta t} = e^{-r_{ij} \tau_i}$$

Y la probabilidad de transmisión será:

$$T_{ij} = 1 - e^{-r_{ij} \tau_i}$$

Para efectos de cálculo

$$T_{ij} = 1 - (1 - r_{ij})^{\tau_i}$$

En general  $r_{ij}$  y  $\tau_i$  variaran entre individuos, dado que la probabilidad de transmisión también varía.

*Para el problema de propagación de enfermedades, se usa los siguientes parámetros:*

- *Tipo de enfermedad.*
- *Población.*
- *Distribución de los individuos.*

*Internamente, se complementa con los siguientes parámetros:*

- *Nivel de contagio que tiene la enfermedad.*

*Dónde:*

***Tipo de enfermedad:*** *Es la enfermedad que se simulara, para este problema se trabajó con 6 enfermedades de origen infecciosa como lo son:*

- *Gripe.*
- *Sarampión.*
- *Meningitis.*
- *Pediculosis.*
- *Paperas.*
- *AH1N1*

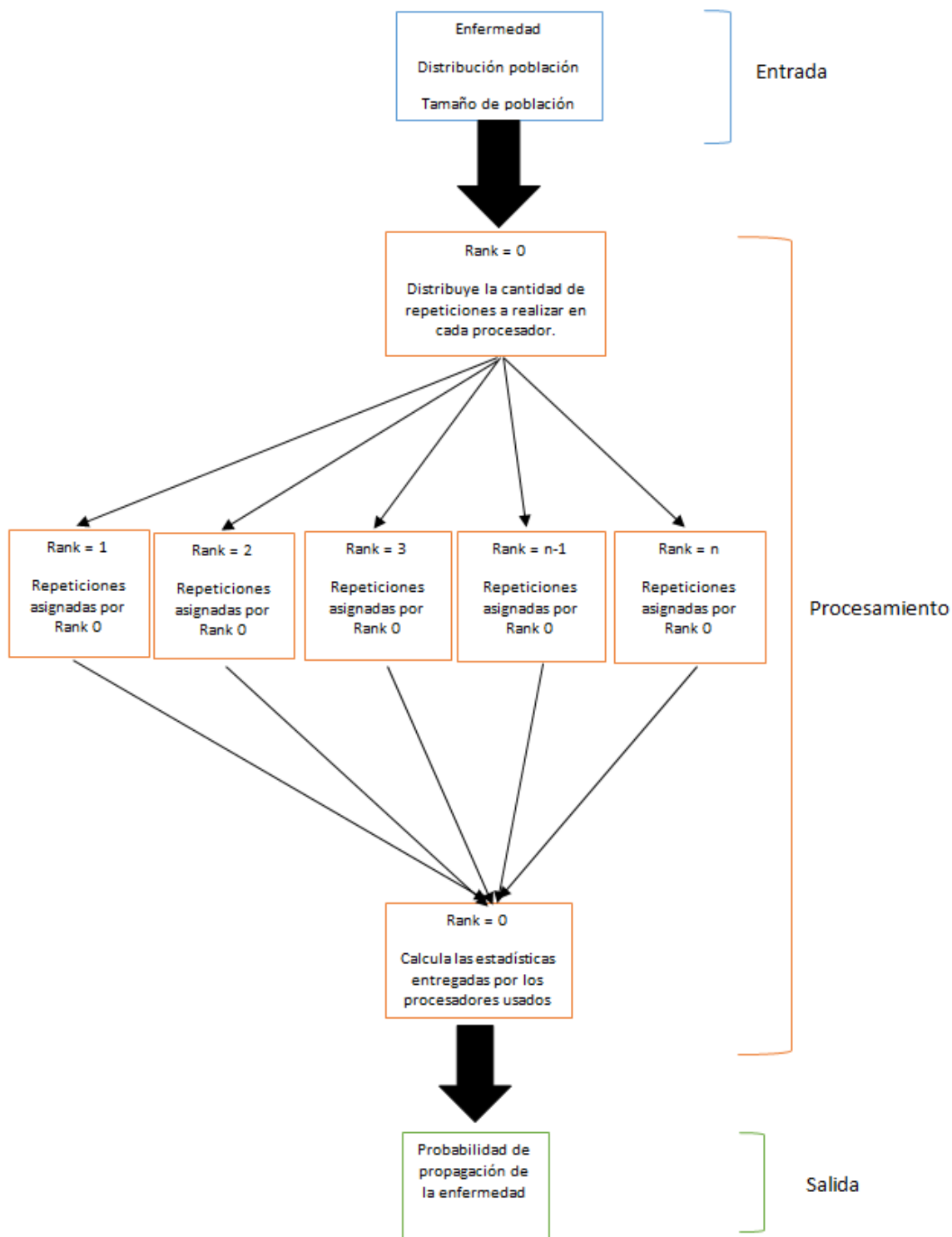
*Donde cada una tiene una probabilidad de contagio distinta, teniendo las más altas de contagio como la gripe y la AH1N1.*

*Cabe destacar que al momento de realizar las simulaciones, no se toma en cuenta las medidas de prevención, es decir, se asume que son poblaciones donde no existen vacunas ni medicamentos para este tipo de enfermedades.*

***Población:*** *Se refiere a la cantidad de individuos que está en la muestra, para el modelo de simulación no se toma en cuenta la separación por géneros, edad, país, etc. Esa clasificación se realiza de ante mano, esta simulaciones asumen que todas las personas son iguales.*

***Distribución de los individuos:*** *Se refiere ha como son las condiciones de estos individuos, si viven juntos, separados, se conglomeran en algunos puntos, o simplemente están muy distanciados.*

## Workflow propagación de enfermedades modelo paralelo



Al igual que en el código paralelo de propagación de incendios, acá el procesador maestro es quien distribuye y reparte la información que debe procesar cada procesador esclavo, pero acá no se entregan imágenes de los focos más importantes por qué el nuevo contagio es común, por lo cual este método permite realizar cual es la probabilidad que afecte a toda la población esta enfermedad.

## 10. Análisis y conclusiones

Lo primero es verificar que se cumplieron los objetivos de este módulo, es decir,

- Implementar una solución al problema utilizando el umbral de percolación.
- Implementar una solución secuencial y una paralela del problema de percolación.
- Analizar algoritmo secuencial y paralelo, analizando su comportamiento y obtener métricas de desempeño.
- Dar respuestas a los alcances de un incendio forestal.

Utilizando la teoría de la percolación y la simulación a través de la plataforma clúster, fue posible calcular el umbral de percolación con muchas repeticiones, para así obtener el valor más cercano al real.

Al utilizar el algoritmo secuencial, vemos que al realizar pruebas con distintos tamaños, los tiempos de ejecución tenían un comportamiento exponencial y se logró encontrar la función que represente el comportamiento, el cual es:

$$y = 0,1124e^{1,4459x}$$

Al comparar los resultados del algoritmo secuencial, con los tiempos de ejecución del algoritmo paralelo, podemos ver que este último es casi 38 veces mejor. Esto sucede siempre y cuando se utilicen los 96 procesadores que es la cantidad óptima de procesadores determinada.

Luego de realizar pruebas con distintos tamaños de matrices en el clúster homogéneo, se puede apreciar que usar 96 procesadores es lo correcto, esto brinda menor tiempo de ejecución, mayor Speed Up, menor costo y mayor eficiencia. Es por esto que se recomienda utilizar esta cantidad de procesadores en la plataforma clúster, en caso contrario, por motivos de disponibilidad u otros, se aconseja utilizar una cantidad menor a 96 procesadores, pero nunca mayor a esta cantidad. Luego de este umbral, no es posible encontrar mejoras de ningún tipo, ya sea, tiempo de ejecución, eficiencia, Speed Up o costo.

Por otro lado, al usar la plataforma clúster heterogénea, se cambió un nodo, disminuyendo sus características de hardware haciendo que su rendimiento fuera muy bajo, en tiempo de ejecución por ejemplo, un proceso se demora casi 16 veces más al usar estos procesadores de menores características.

De todas formas, al seguir la recomendación de usar una cantidad menor de 96 procesadores, no se ve afectado el rendimiento por qué, en ningún caso alcanza a usar los procesadores de menores características.

Es necesario señalar que estos tiempos corresponden al tiempo total de ejecución realizado por el algoritmo de percolación, dejando de lado los tiempos de creación de PDF, ingreso de requerimientos desde la plataforma web y envíos de respuestas a través del servidor.

Una gran ventaja que brinda este módulo, es que bajo la misma forma de simulación permite integrar más problemas como se realizó con propagación de enfermedades. Esto permite tener una gran escalabilidad y mayor versatilidad para diferentes problemas.

## Anexo L: Asana



Asana es un software de gestión de tareas vía web diseñada para mejorar la colaboración y comunicación de un equipo. Esta herramienta da la posibilidad de organizar los hitos de las actividades en diferentes proyectos independientes clasificándolos con etiquetas. La versión gratuita de Asana permite usar la mayoría de funcionalidades hasta un máximo de 15 usuarios en una organización, para lograr acceder a más se debe adquirir la versión de pago.

Esta aplicación comienza a ser creada en el año 2007 dentro de Facebook por los desarrolladores Justin Rosenstein y Dustin Moskovitz. Después de un tiempo se dieron cuenta que Facebook los limitaba, por lo que la transformaron en una buenísima herramienta para emprendedores, compañías y demases para mejorar su productividad y eliminar todos esos correos con muchísimas respuestas junto con elaborar un diseño amigable, con características útiles que sean fácil de usar y que perduren en tu proceso de organización y proyecto en general.

### Hitos Importantes

#### Propuesta

- Recolectar información
- Fijación de objetivos
- Analizar información
- Confección de documento de la propuesta

#### Plataforma

- Instalación de Ubuntu
- Agregar nodos Heterogéneos
- Configuración MPI
- Configuración LAMP
- Instalación Editor de texto
- Configuración DNS
- Configuración DHCP
- Servidor de correos
- Túnel ssh

## Página Web

- Modelo
- SubProcesos
- Front End
- Back End

## Código Percolación

- Recolectar Información
- Analizar Información
- Fijar Objetivos
- Entrega 1
- Modelo Algoritmo Secuencial
- Implementación Algoritmo Secuencial
- Entrega 2
- Modelo Algoritmo Paralelo
- Implementación Algoritmo Paralelo
- Entrega 3
- Documentación Algoritmo

## Código Fasta

- Recolectar Información
- Analizar Información
- Fijar Objetivos
- Entrega 1
- Modelo Algoritmo Secuencial
- Implementación Algoritmo Secuencial
- Entrega 2
- Modelo Algoritmo Paralelo
- Implementación Algoritmo Paralelo
- Entrega 3
- Documentación Algoritmo

### **Código de la Biblia**

- Recolectar Información
- Analizar Información
- Fijar Objetivos
- Entrega 1
- Modelo Algoritmo Secuencial
- Implementación Algoritmo Secuencial
- Entrega 2
- Modelo Algoritmo Paralelo
- Implementación Algoritmo Paralelo
- Entrega 3
- Documentación Algoritmo

### **Tareas Administrativas**

- Estructurar Asana
- Investigación Asana

### **Confección Avance I**

- Calculo de Esfuerzo
- Carta Gantt
- Valor Ganado
- Armado Documento
- Entrega Avance 2

### **Confección Avance II**

- Calculo de Esfuerzo
- Carta Gantt
- Valor Ganado
- Armado Documento
- Entrega Avance 2

### **Confección Avance III**

- Calculo de Esfuerzo
- Carta Gantt
- Valor Ganado
- Armado Documento
- Entrega Avance 3



## Confección Informe Final

- Calculo de Esfuerzo
- Carta Gantt
- Valor Ganado
- 1era Parte del Informe Final
- 2da Parte del Informe Final
- 3era Parte del Informe Final
- Entrega Informe Final

## Gráficos de avance

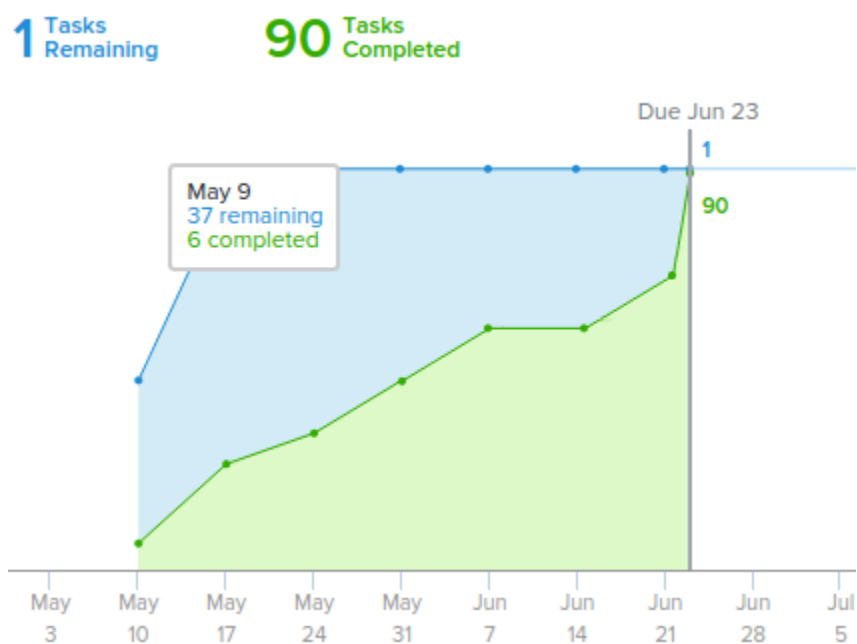


Gráfico de progreso de los hitos generales

## Conclusiones

Teniendo en cuenta el gráfico de progreso de hitos generales podemos hacer las siguientes conclusiones:

- Las actividades en azul son las que han sido establecidas.
- Las actividades en verde son las completadas.
- En la última etapa se puede ver mayor esfuerzo debido a que se debían cumplir los atrasos antes de que llegara el término del proyecto.

### Anexo M: Carta de amonestación

En el desarrollo de la plataforma, un integrante del proyecto se ausentó durante la primera etapa y parte de la segunda. Esto lo hizo merecedor de una carta de amonestación. Dicha carta se le entregó y fue recibida sin ningún problema.

La persona en cuestión, explicó su problemática, estando al tanto de esto se le brindó apoyo y se le asignaron nuevas tareas, las que cumplió sin presentar mayores dificultades.

#### Carta de Amonestación N° 1

Señor (a) **Rodrigo Andrés Reyes Silva**.

Presente.

Santiago, 19 de mayo de 2015

No referimos a usted para informarle que la conducta observada, en relación al cumplimiento de sus deberes estipulados en un principio en el laboratorio dos de Computación Paralela, no ha sido satisfactoria, dejando en evidencia su incumplimiento en **la fase uno** de la plataforma de servicios. Esto constituye una falta a las obligaciones al laboratorio, situación que es de su conocimiento y que además se encuentra contemplada por su encargado **Manuel Alejandro Venegas Solís**

Por la presente carta amonestación se requiere el estricto cumplimiento de las instrucciones dadas por su gerencia en cuanto a la forma correcta de llevar el registro de asistencia y cumplimiento de deberes, de lo contrario, se adoptarán las medidas pertinentes, inclusive la terminación de su participación en el laboratorio dos de computación paralela.

Saluda a usted.

Diego Hernández García

**Nombre y Firma Gerencia General**

NOTA:

1. Deberá entregarse personalmente al alumno, debiendo firmar en señal de recepción. En el caso que se niegue a recibir o a firmar, se informara al profesor correspondiente.