**IMPLEMENTACION SUPERCOMPUTADORA**

**GRUPO SUPERCOMPUTADORA**

**LIDER: EDGAR TORRES COY**

**PROFESOR: JOHN EDWARD CASTRO BERNETH**

**FUNDACION UNIVERSITARIA SAN MATEO**

**INGENIERIA DE SISTEMAS**

**COMPUTACION PARALELA Y DISTRIBUIDA**

**BOGOTA**

**2019**

Contenido

[**PROPUESTAS DEL TRABAJO A REALIZAR** 3](#_Toc21691114)

[**Propuesta 1:** 3](#_Toc21691115)

[**Propuesta 2:** 3](#_Toc21691116)

[**Propuesta 3:** 4](#_Toc21691117)

[**PROPUESTA SELECCIONADA** 5](#_Toc21691118)

[**CRONOGRAMA** 6](#_Toc21691119)

[**ROLES** 6](#_Toc21691120)

[**METODOLOGÍA A UTILIZAR (SI SE REQUIERE).** 7](#_Toc21691121)

[**ARQUITECTURA DEL DESARROLLO DEL PROYECTO.** 8](#_Toc21691122)

[**Arquitectura SOA** 8](#_Toc21691123)

[**PARTE 2 DE LA INVESTIGACIÓN DEL PROYECTO REALIZADO.** 8](#_Toc21691124)

[**PVM (Parallel virtual Machine)** 8](#_Toc21691125)

[**MOSIX** 9](#_Toc21691126)

# **PROPUESTAS DEL TRABAJO A REALIZAR**

## **Propuesta 1:**

Se ejecutará el desarrollo de una súper computadora mediante equipos físicos los cuales estarán interconectados mediante un switch y cables de red, permitiendo su comunicación, los equipos contaran con sistema operativo basado en Linux (se sugiere Debian o Pelican).

La metodología que se usara para la construcción de la súper computadora será Scrum ya que es una metodología ágil que nos permite acoplar los procesos de desarrollo que se requieren para la realización del proyecto.

Su arquitectura será la SOA (Arquitectura orientada a servicios), dado que su meta es aportar la flexibilidad mediante la automatización de infraestructuras y herramientas necesarias para ejecutar la creación de una súper computadora.

Como aporte a la materia este trabajo nos permitirá evidenciar físicamente la forma cómo se comporta una supercomputadora, mediante la estructura propuesta, y compararla con un solo clúster, con la finalidad de verificar su desempeño y sus bondades en la automatización de procesos.

## **Propuesta 2:**

Comprobar mediante el uso de máquinas virtuales el desempeño que tiene una supercomputadora, configurando un equipo como master y los demás como nodos, para brindar al master las fuerzas necesarias para ejecutar más eficientemente los procesos, los cuales usaran un sistema operativo basado en Linux.

La arquitectura bajo la cual se desarrollara será SOA (Arquitectura orientada a servicios), la cual nos brinda flexibilidad en la automatización de los procesos que ejecutan cada uno de los nodos al servicio del equipo Master.

La metodología que se pretende usar para el desarrollo de la supercomputadora es Scrum la cual nos brinda la posibilidad de llevar en mejor forma las actividades para el desarrollo del trabajo.

Como aporte a la materia, la demostración de la virtualización de las maquinas nos permitirá ver en forma más clara como se pueden compartir recursos entre varios equipos para mejorar el desempeño de una máquina.

## **Propuesta 3:**

La metodología Ágil que se pretende implementar es KANBAN ya que esta es similar a la metodología SCRUM, no poseemos historia de usuario, nos permite el trabajo en paralelo y la asignación de tareas con sus respectivos estados (To Do, Doing, Don)

Descripción del desarrollo del trabajo

* La duración de los Sprint será equivalente a 7 días
* Las tareas que se identifiquen serán asignadas dependiendo de las habilidades que tenga cada uno de los integrantes
* Los Dailys se realizaran a través del grupo de wathsapp (si se es requerido) cada uno tendrá duración de 15 minutos aprox. En el cual se expondrá las tareas en las que se está trabajando y se presentaran los posibles impedimentos

Arquitectura (backend)

Teniendo en cuenta la necesidad es viable la arquitectura SOA, que nos permite la distribución de los servicios en diferentes servidores para el trabajo en paralelo. Se pretende crear un bus de servicios, esto con el fin de centralizar y gestionar la comunicación entre los diferentes servicios, estos podrían estar alojados en dockers para poder suministrar los recursos necesarios para que cada servicio funcione de manera satisfactoria y en caso de fallar no afecte el funcionamiento de los otros.

Arquitectura (Infraestructura)

Balanceador, para la distribución de peticiones entre las máquinas que se vayan a implementar.

Si es posible, X cantidad de máquinas con las mismas características. En caso contrario, X cantidad de máquinas virtualizadas.

Tecnologías y Herramientas

• Servicios REST con dotNet Core para garantizar la interoperabilidad entre sistemas (Windows y Linux).

• Dockers, la implementación de estos es viable en Windows y Linux.

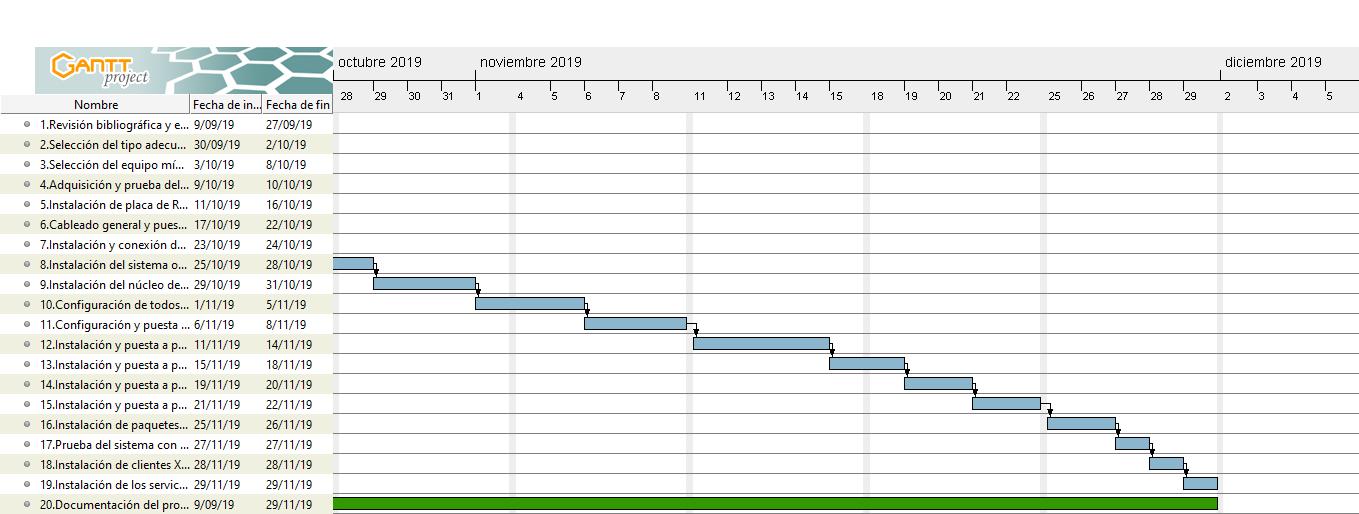
El aporte que pretendemos dar a la clase es el de realizar por medio de una comprobación física o virtual las bondades de la creación de supercomputadoras y como al realizar cualquiera de las tres propuestas estamos realizando una comprobación de los procesos educativos y los temas vistos durante el desarrollo de la materia.

# **PROPUESTA SELECCIONADA**

La propuesta seleccionada para realizar el trabajo es la # 1, teniendo en cuenta lo expuesto en la misma se realizaran las siguientes actividades para poder cumplir con la propuesta:

1. Revisión bibliográfica y estudio de las bases de funcionamiento de los clusters.
2. Selección del tipo adecuado para las necesidades del grupo (MOSIX).
3. Selección del equipo mínimo necesario para implementar el cluster elegido.
4. Adquisición y prueba del hardware seleccionado.
5. Instalación de placa de Red adicional en el nodo maestro.
6. Cableado general y puesta a punto de la conexión del servidor a la red local de la FUS.
7. Instalación y conexión de los nodos mediante un Switch.
8. Instalación del sistema operativo Linux en todas las PCs
9. Instalación del núcleo del sistema MOSIX en ambas PCs
10. Configuración de todos los servicios y protocolos de red necesarios.
11. Configuración y puesta a punto del MOSIX
12. Instalación y puesta a punto del MPI
13. Instalación y puesta a punto del PVM
14. Instalación y puesta a punto del sistema de colas de procesos (PBS).
15. Instalación y puesta a punto del sistema de administración remota (Webmin).
16. Instalación de paquetes, lenguajes y herramientas específicas para cálculo numérico.
17. Prueba del sistema con diferentes aplicaciones (software de redes neuronales, programas de cálculo pesado, etc.)
18. Instalación de clientes X (entorno gráfico) y SSH para acceso desde las terminales Windows.
19. Documentación del proceso (versiones preliminares de este informe)

# **CRONOGRAMA**



# **ROLES**

|  |  |  |
| --- | --- | --- |
| **Rol** | **Responsabilidades:** | **Entregable:** |
| Tester (Jhon Jairo López Sáez) | Encargado de realizar diferentes pruebas desde el primer prototipo - modelo hasta el final en lineamiento con la arquitectura y metodología colaborativa propuesta; asi commo el registro de los cambios y mejoras en la evolución en el tiempo. | Diagramas y resultados de pruebas obtenidas. |
| Testing (Brayam Estiven Duran Cardona) | Encargado de realizar pruebas funcionales y no funcionales con el fin de detectar defectos en el uso de la plataforma | Pruebas documentadas realizadas (test configuration, set de pruebas) |
| Implementador (David Gerardo Urrego Camargo) | Encargado de realizar y verificar los componentes del proyecto de acuerdo con la metodología y la arquitectura establecida al inicio del proyecto, al igual será el responsable de implementar un prototipo y generar un reporte respecto. | Documentación Técnica |
| Desarrollador (Jhonatan Stiv Cuadros Cárdenas) | Encargado de implementar todos los requerimientos relacionados en el documento ya sea con un software o con la implementación y/o configuración de la supercomputadora establecida en el documento, siguiendo los parámetros de la metodología. | Implementación |
| Diseñador (Anderson Rubio) | Delegado de la implementación de la metodología, relaciona el diseño con la implementación incluyendo el nivel de detalle esperado en el diseño antes de que proceda la implementación.  Así como la verificación de los requerimientos para el desarrollo del proyecto.  creación de un concepto de sistema que ayude a cumplir los objetivos del proyecto | Diagramas de estructura  Requerimientos de Vista metodología de proyecto |
| Analista (Tatiana Pinzon) | Encargada de analizar la información para el levantamiento de los requerimientos y las necesidades para la ejecución del proyecto. | Matriz de requerimientos |
| Edwin Alexander Chavez Barreto | Encargado del soporte técnico sobre los equipos de cómputo a intervenir para la implementación de la supercomputadora, instalación de sistemas operativos necesarios y mantenimiento funcional de los aplicativos y conexión y configuración necesaria para la conexión de la red de datos | Informe técnico, usuarios y claves de acceso, informe de configuración de los sistemas operativos en cada equipo de cómputo. |

# **METODOLOGÍA A UTILIZAR (SI SE REQUIERE).**

**Metodología Scrum:** Esta nos permite planificar la realización de la ejecución del proyecto en pequeños springs, que se ejecutaran por semana según el cronograma

El desarrollo de producto tiene un ciclo de vida en la metodología Scrum. Estas son fases en las que se divide un proceso Scrum:

¿Qué y quién? El producto que queremos conseguir una vez terminemos el Sprint, y los roles de equipo con sus tareas asignadas, según cada semana programada que se debe intervenir sobre el proyecto

Se efectuará una reunión para la planificación del Sprint de cada semana. En ella, se divide el tiempo de duración del Sprint, así como el objetivo y entregable del mismo. Además, el equipo de desarrollo deberá saber cómo realizarlo, el cual ya se encuentra constatado en el cronograma establecido, para algunas actividades será un Scrum diario. Para ejecutar actividades para elaborar el plan del día, adicionalmente se verificará si lo objetivos del proyecto se están cumpliendo

Se tendrá también una reunión del sprint del líder del proyecto con el docente encargado para verificar si se encuentra en cumplimiento los requisitos o si se requiere ejecutar algún tipo de modificación, la cual se ejecutará al después de acabar el sprint correspondiente.

# **ARQUITECTURA DEL DESARROLLO DEL PROYECTO.**

## **Arquitectura SOA**

La meta de esta arquitectura es aportar la flexibilidad de activos existentes que permite la automatización de infraestructuras y herramientas necesarias para ejecutar la creación de una súper computadora.

Por su versatilidad es posible que los servicios puedan ser consumidos por los clientes en aplicaciones o procesos de negocios distintos ya que tiene una ventaja que es la interoperabilidad entre ellas

# **PARTE 2 DE LA INVESTIGACIÓN DEL PROYECTO REALIZADO.**

## **PVM (Parallel virtual Machine)**

En su traducción al español Máquina Virtual Paralela, es un sistema o conjunto de bibliotecas que se usa principalmente dentro de la programación paralela, la cual me permite realizar una conexión entre dos o más equipos de cómputo, con el objetivo integrar los recursos de estas computadoras, para finalmente obtener el esquema de una sola máquina.

Este software opera en diferentes plataformas como lo pueden ser (Unix y Windows), aunque para la creación del proyecto se usara o se implementara dentro de la plataforma Linux.

Algunas de las funciones del uso del PVM consiste en la mensajería asincrónica y para el control de procesos entre los diferentes nodos presentes dentro del sistema, además que permite la escalabilidad del mismo.

**Ventajas:**

Entre las diferentes librerías existentes para el envío de mensajes en el paralelismo es uno de los más fáciles de usar.

Control arbitrario de dependencias de estructura, esto permite que la aplicación determine: (Donde y cuando Iniciar o terminar tareas, que maquinas se añaden o se eliminan desde la máquina virtual y finalmente que tareas se puede comunicar y/o sincronizar con otras.)

**Desventajas:**

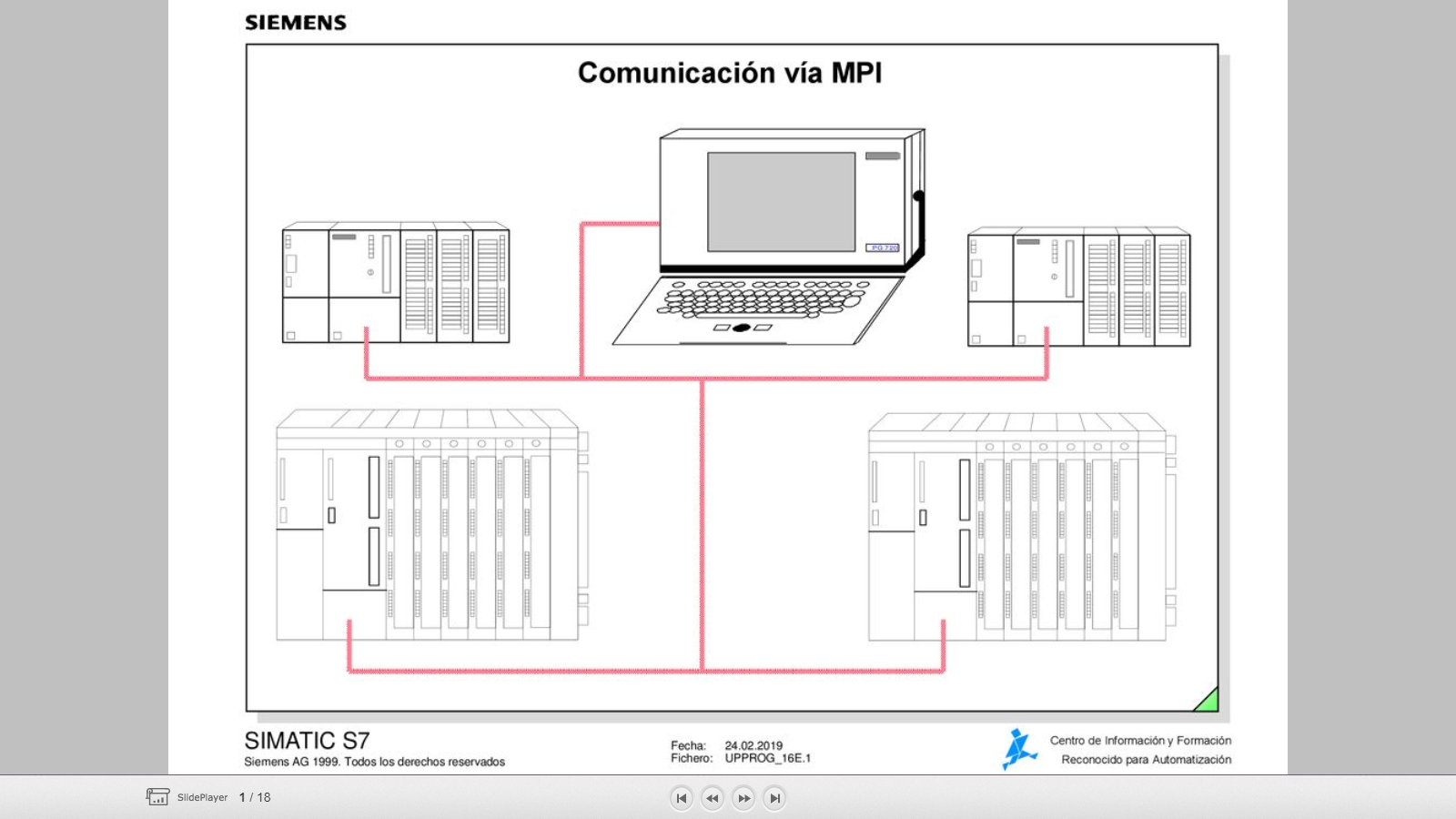
En gran medida el rendimiento del esquema dependerá de la capacidad de procesamiento de los ordenadores o nodos que se encuentran dentro del esquema.

Lo que es una ventaja también se puede convertir en una desventaja, ya que, aunque es muy fácil de usar, esto conlleva a que sea algo deficiente en cuanto al paso de mensaje entre nodos.

## **MPI (Message Passing Interface)**

Es una librería que permite el paso de mensajes, ya sea de forma estándar y/o portable, se utilizan principalmente a la hora de realizar cálculos dentro de un clúster, se usa a través de un programa que permita la migración de procesos entre múltiples procesadores.

Él envío de mensaje es una técnica que se usa en programación concurrente que permite la sincronización de diversos procesos que se desean ejecutar entre nodos.



## **MOSIX**

Es un software con capacidades de computación de clústers, específicamente permite una mejora en el Kernel (Intermediario entre software y hardware) de Linux, esta mejora permite que cualquier clúster de estaciones de trabajo y servidores trabajen coordinadamente como parte de un solo sistema, permitirá ejecutar aplicaciones no paralelizados en un clúster, en otras palabras el software permite ejecutar programas convencionales destinados para funcionar en una maquita, en un clúster a través del envío de mensajes o la migración de procesos entre los nodos que se encuentran dentro del sistema.

Una característica muy eficiente que tiene MOSIX dentro del funcionamiento de un clúster, es que realizara un monitoreo constante en los procesos que encuentre o que se están ejecutando y es de esta forma que dividir de forma correcta los procesos entre los nodos y generar un equilibrio, cabe destacar que este proceso antes mencionado lo realizara de forma automática.

Debemos tener en cuenta que solo algunas aplicaciones se beneficiaran bajo este sistema de clúster que permita MOSIX, algunos de estos son:

* Procesos que necesiten para su funcionamiento el uso de una gran cantidad de CPU.
* Procesos que tengan tiempos de ejecución impredecibles.
* Un gran número de usuarios simultáneos (Concurrencia).
* Servidores Web predispuestos a escalabilidad.