ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

IMPLEMENTACIÓN DE UNA ARQUITECTURA TOLERANTE A FALLOS EN LOS SERVIDORES DE PRODUCCIÓN DE LOS SISTEMAS DE GESTIÓN ACADÉMICA DE LA DIRECCIÓN DE DESARROLLO ACADÉMICO (DDA) DE LA ESPOCH.

Trabajo de titulación presentado para optar el grado académico de:

INGENIERO EN SISTEMAS INFORMÁTICOS

AUTOR: JORGE OSWALDO ZARUMA TUALOMBO

TUTOR: ING. LORENA AGUIRRE

RIOBAMBA – ECUADOR

2019

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal de Trabajo de Titulación certifica que: El proyecto técnico: **"IMPLEMENTACIÓN DE UNA ARQUITECTURA TOLERANTE A FALLOS EN LOS SERVIDORES DE PRODUCCIÓN DE LOS SISTEMAS DE GESTIÓN ACADÉMICA DE LA DIRECCIÓN DE DESARROLLO ACADÉMICO (DDA) DE LA ESPOCH",** de responsabilidad del señor Jorge Oswaldo Zaruma Tualombo, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

	FIRMA	**FECHA**
Dr. Julio Santillán **VICEDECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**	_____	_____
Ing. Patricio Moreno **DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS**	_____	_____
Ing. Lorena Aguirre **DIRECTORA DE TRABAJO DE TITULACIÓN**	_____	_____
Dr. Omar Gómez **MIEMBRO DE TRABAJO DE TITULACIÓN**	_____	_____

Yo, Jorge Oswaldo Zaruma Tualombo soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la Tesis de Grado pertenece a la Escuela Superior Politécnica de Chimborazo.

Jorge Oswaldo Zaruma Tualombo

DEDICATORIA

Dedico el presente trabajo de titulación a mis padres que me enseñaron que para obtener lo que uno quiere se debe trabajar día a día y ser constante a pesar de las dificultades que se puedan presentar. También a mis hermanos y hermanas quienes siempre han estado apoyándome en cada decisión que he tomado.

Jorge

AGRADECIMIENTO

Agradezco a cada uno de mis amigos que a lo largo de la carrera me ayudaron a seguir delante de quienes siempre tendré buenos recuerdos.

Jorge

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE ANEXOS

ÍNDICE DE ABREVIATURAS

Ph.D: Philosophiae Doctor (Doctorado en investigación)

SSL: Secure Sockets Layer (Capa de sockets seguros)

DDA: Dirección de Desarrollo Académico

ESPOCH: Escuela Superior Politécnica de Chimborazo

IBM: International Business Machines Corporation

FTP: File Transfer Protocol (Protocolo de transferencia de archivos)

LVS: Linux Virtual Server

DevOps: Development and Operations (Desarrollo y Operaciones),

SQL: Structured Query Language (Lenguaje de consulta estructurada)

SGBD: Sistemas Gestor de Base de Datos

XML: Lenguaje de marcado extensible

RESUMEN

El presente trabajo de titulación tuvo como objetivo implementar una arquitectura tolerante a fallos en los servidores de producción de los sistemas de gestión académica de la Dirección de Desarrollo Académico (DDA), como lo es el Silabo, Programas Analíticos y Planificación donde se implantó módulos de gestión de usuarios y roles que fueron desarrollados bajo la metodología scrum donde se obtuvo un total de 17 historias de usuario y 14 historias técnicas. Para la arquitectura tolerante a fallos se tomó en cuenta herramientas que permitan configurar redundancia, escalabilidad y alta disponibilidad, entre las principales se puede mencionar payara como servidor de aplicaciones donde se creó un clúster, apache como servidor web que en conjunto con las herramientas de pacemaker y corosync se implementó alta disponibilidad, además se creó un archivo de configuración que funciona como balanceador de carga, por último mediante la extensión citus de postgresql se creó un clúster donde la función principal es de distribuir las tablas de las bases de datos y así lograr equilibrar la carga de las peticiones que se envíen al gestor de base de datos. El resultado final de este trabajo fue la implantación de una arquitectura tolerante a fallas donde se evaluó la fiabilidad correspondiente a la norma ISO/IEC 25010 con la herramienta Jmeter con la que se efectuó 2 escenarios con 2 casos de prueba, que permitió verificar que en condiciones normales al procesar 500 peticiones se obtiene un error del 0,36% del total de peticiones que no logran acceder al sistema mientras que al procesar 2000 peticiones el error aumenta a 10,91%. Con esto se concluye que para atender a más usuarios es necesario aumentar los recursos hardware del servidor web. Finalmente se recomienda utilizar escenarios para evaluar una arquitectura similar.

Palabras claves: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA, REDES, <TOLERANCIA A FALLAS>, <ALTA DISPONIBILIDAD>, <ESCALABILIDAD>, <REDUNDANCIA>, <SERVIDOR WEB>, <SERVIDOR DE APLICACIÓN>, <SERVIDOR DE BASE DE DATOS>, <CLÚSTER>

ABSTRACT

The goal of this current degree work was the implementation of a fault tolerant architecture of the production servers in the academic management systems at the Academic Development Direction (DDA), such us: syllabus, analytic programs and planning where user management modules and roles were implanted which were developed using Scrum methodology obtaining a total of 17 user and 14 technical stories. Different tools were taken into account at the moment of creating the fault tolerant architecture which enabled to set up redundancy, stability and high availability, the main used tools are: Payara as application server whereby a cluster was built, and Apache as web server which together implemented a high availability using Pacemaker and Corosync as tools, moreover, a configuration file was created which works as load balancer, ultimately, a cluster was built through Citus extension which belongs to PostgreSQL, it has the main function of distributing database tables and do load balance of the requests which are sent to database manager. The final result of this research was the implementation of a fault tolerant architecture where the reliability corresponding to the standard ISO/IEC 25010 was evaluated using Jmeter tool, with which 2 scenarios were made with 2 test cases which permitted to verify an error of 0,36 % of the total number of requests that cannot access to the system at the moment of processing 500 requests in normal conditions, whereas that at the moment of processing 2000 requests the error increases to 10,91%. To conclude, it is necessary to increase the hardware resources of the web server for serving more users. Finally, it is advisable to use scenarios for assessing a similar architecture.

Key words: <TECHNOLOGY AND ENGINEERING SCIENCES, NETWORK, FAULT TOLERANCE> <HIGH AVAILABILITY> <SCALABILITY> <REDUNDANCY> <WEB SERVER> <APPLICATION SERVER> <DATABASE SERVER> <CLUSTER>

INTRODUCCIÓN

Actualmente el desarrollo de aplicaciones web se ha convertido en algo esencial de nuestra vida diaria, por lo que permite la automatización de procesos en los que manualmente tardarían demasiado tiempo, pero al existir un gran crecimiento de este tipo de sistemas es necesario tener una infraestructura tecnológica que satisfaga a las necesidades de cada tipo de aplicación ya que manejan datos importantes para la empresa.

Por tal motivo en la Dirección de Desarrollo Académico (DDA) de la Escuela Superior Politécnica de Chimborazo (ESPOCH) existen varios procesos que se han automatizado como: Sílabos, Programas Analíticos y Planificación los mismos que son importantes para las tareas que se ejecutan en una carrera, en los tres sistemas mencionados anteriormente y como parte del presente trabajo se implantaran módulos de gestión de usuarios y roles.

Además, se debe tener en cuenta que se pueden presentar varios fallos a nivel de hardware y software los mimos que se deben prevenir o saber cómo reaccionar frente a las mismas para evitar pérdidas económicas a las empresas. Por tal motivo, el objetivo principal del presente trabajo de titulación se centra también en implantar una arquitectura tolerante a fallos a nivel de los servidores de producción que alojaran los sistemas de gestión académica mencionados anteriormente de la DDA y así garantizar el correcto funcionamiento de las aplicaciones.

Una vez que se haya realizado la formulación general del trabajo de titulación los capítulos que se agregan son los siguientes:

Capítulo 1, que corresponde al marco teórico es decir aquí se realizó la investigación acerca de arquitecturas relacionadas a tolerancia a fallos, características y herramientas que se van a utilizar para su implementación, teniendo en cuenta la metodológica que se va a utilizar y estándar de calidad para medir los resultados.

Capítulo 2, comprende el marco metodológico donde se detalla las fases de la metodológica scrum utilizada para el desarrollo del trabajo de titulación.

Capítulo 3, marco de resultados donde se describe el estándar de calidad ISO/IEC 25010 que se utilizó para medir el funcionamiento de la arquitectura tolerante a fallos implantada.

FORMULACIÓN GENERAL DEL TRABAJO DE TITULACIÓN

Antecedentes

La Escuela Superior Politécnica de Chimborazo, ubicada en Riobamba, según la: *"Ley 6909 del 18 de abril de 1969, expedida por el Congreso Nacional publicada por el registro Oficial Nº 173 del 7 de mayo de 1969, se crea el Instituto Superior Tecnológico de Chimborazo, iniciando sus labores académicas el 2 de mayo de 1972. El cambio de denominación a Escuela Superior Politécnica de Chimborazo ESPOCH, se produce mediante Ley No. 1223 del 29 de octubre de 1973 publicada en el Registro Oficial Nº 425 del 6 de noviembre del mismo año"* (ESPOCH, 2018). Esta institución tiene como como uno de sus fines el "*Impartir enseñanza a nivel de pregrado, postgrado y educación continua, en ciencia y tecnología, basadas en la investigación y la producción de bienes y servicios"*(ESPOCH, 2018).

Esta institución cuenta con la Dirección de Tecnologías de la Información y Comunicación que tiene como misión *"Proporcionar servicios integrales de calidad en las áreas de desarrollo organizacional y sistemas de información a la ESPOCH y entidades externas, utilizando tecnología de punta, con personal capacitado, estándares de calidad y una participación activa y eficaz del usuario"* (DTIC, 2018) el cual a su vez cumple con funciones como:

- *Desarrollar y mantener los sistemas informáticos administrativos, académicos y de la organización* (DTIC, 2018).
- *Apoyar los procesos de modernización administrativa, académica y de gestión institucionales* (DTIC, 2018).
- *Proporcionar servicios de mantenimiento de Hardware y Software a la ESPOCH y la colectividad* (DTIC, 2018).

Como la institución es parte del sector público el acceso a la información es un derecho de todos los ciudadanos como se establece en la Ley Orgánica de Transparencia y Acceso a la Información Pública en su Art. 1 Principio de Publicidad de la Información Pública en el cual establece básicamente que toda la información que emane dichas instituciones es carácter público salvo las excepciones establecidas por la ley y en caso de incumplimiento de la misma, serán sancionados de acuerdo al Art. 23 de la presente Ley, que puede ser una multa, suspensión de funciones y hasta destitución del cargo (LOTAIP, 2018). Por lo tanto, todas las instituciones deben prevenir de cualquier manera que la información este inaccesible. Existen diferentes causas para que la

información no esté disponible como: servidores públicos sin conocimiento de su área, falta de presupuesto económico para automatizar dichos procesos o porque los sistemas informáticos de las instituciones tardan demasiado tiempo en recuperarse ante fallos que se podrían presentar al momento de procesar peticiones.

Mantener un sistema disponible a pesar de los fallos que se puedan presentar es importante para que la información de las instituciones no quede inaccesible para los usuarios finales, por lo que en el presente trabajo de titulación se pretende implementar una arquitectura tolerante a fallos en los servidores de producción para los sistemas de gestión académica de la DDA de la ESPOCH.

Formulación del problema

¿Al implementar una arquitectura tolerante a fallos en los servidores de producción de la Dirección de Desarrollo Académico (DDA) de la ESPOCH evitará que los sistemas de gestión académica de la DDA experimenten algún tipo de inactividad en sus procesos?

Sistematización del problema

- ¿Cuáles son los fallos más comunes que afectan el rendimiento de un servidor?
- ¿Existen estudios sobre arquitecturas tolerante a fallos?
- ¿Es posible implementar una arquitectura tolerante a fallos en los servidores de producción de los sistemas de gestión académica de la DDA?
- ¿Cuál es la métrica de calidad de software que se puede utilizar para verificar el funcionamiento de la arquitectura tolerante a fallos implantada?

Justificación

Justificación teórica

Con el objetivo de implementar una arquitectura tolerante a fallos se tomó en cuenta las siguientes investigaciones que servirán de base para la ejecución del presente trabajo de titulación.

Josemar de Souza en su tesis acerca de la tolerancia a fallos hizo un estudio de la importancia de la utilización de clúster, da énfasis en que la tolerancia a fallos permite que un sistema continúe su funcionamiento a pesar de fallos que podría existir en su ejecución, permitiendo obtener un software más fiable, la finalidad de este trabajo fue el crear un modelo de tolerancia a fallos basado en replicación inicial de procesos y replicación dinámica. Como resultado se obtuvo un software más fiable y mediante la replicación aumento la tolerancia a fallos de los mismos. (de Souza, 2006)

Además, Calmels en su proyecto relacionado con la escalabilidad y disponibilidad para la nube manifestó la importancia de estos dos grandes factores en la actualidad. El primero que puede ser a nivel de hardware como: mayor capacidad de disco, memoria, también software dando la capacidad de aumentar nodos si es necesario, mientras que el segundo permite que un sistema en base a redundancia este fuera de servicio el menor tiempo posible lo hace posible mediante herramientas disponibles como balanceadores de carga en la nube. Como resultado el proveedor nubity el cual les garantizó que la configuración se haya realizado rápida y fácilmente, además de una solución de infraestructura que proporciona un servicio disponible en todo momento, que la respuesta ante las demandas sea rápida y eficaz, además debe presentar una rápida recuperación ante los fallos(Calmels, 2004).

A su vez, en el trabajo de Gustavo Bustos sobre clúster de alta disponibilidad llegó a la conclusión que la tecnología de clustering es importante en ambientes donde se requiere un incremento de confiabilidad y robustez ofrecido por un clúster de alta disponibilidad. Además, con el uso de heartbeat se puede constatar en tiempo real los posibles fallos que interrumpan la conexión y así evitar que los sistemas dejen de operar (Bustos, 2007).

Por último, en la tesis de maestría presentada por Rogel Miguez hizo referencia a la importancia de la redundancia en servidores, así como de la implementación del modelo activo-pasivo lo que le permitió que al fallar el nodo principal o activo el nodo redundante o pasivo pasa a asumir las funciones del nodo principal, esto a su vez ayudo a mejorar la disponibilidad, y disminuir el tiempo de recuperación ante fallos, cabe mencionar que el tiempo entre fallos se mantuvo. Pero de manera general solo con la implementación de redundancia se logra mejorar características importantes que debe cumplir los servidores como es el disponibilidad y tiempos de recuperación (Miguez, 2012).

Justificación aplicativa

Con la implementación de una arquitectura tolerante a fallos en los servidores de producción, se pretende evitar que los sistemas de gestión académica de la DDA queden fuera de servicio y con su información inaccesible para los interesados en los documentos que estos sistemas generarán, al aplicar dicha arquitectura en los sistemas de la DDA como: Sílabos, Programas Analíticos y Planificación, estos podrán trabajar sin interrupciones a pesar de la cantidad de usuarios que accedan a estos sistemas, esto ayudará que tanto los estudiantes, docentes y autoridades de la ESPOCH puedan acceder a los documentos de estos sistemas cuando los necesiten, agilizando el tiempo en la obtención de dicho documento que son necesarios para algunos trámites académicos.

Con la finalidad de que la información de los sistemas de la DDA esté actualizada se vio en la necesidad de agregar el módulo de gestión de usuarios y roles esto permitirá que cada uno de los sistemas gestione estos componentes como lo considere pertinente, además es necesario la integración del sistema de Programas Analíticos, Sílabos Institucionales y Planificación, por medio de servicios web.

Objetivos

Objetivos Generales

- Implementar una arquitectura tolerante a fallos en los servidores de producción de los sistemas de gestión académica de la Dirección de Desarrollo Académico.

Objetivos específicos

- Investigar los fallos más comunes que podrían afectar al rendimiento de los servidores de producción.
- Investigar sobre arquitecturas tolerante a fallos
- Aplicar una arquitectura tolerante a fallos en los servidores de producción.
- Evaluar la fiabilidad de la arquitectura tolerante a fallos mediante la norma ISO/IEC 25010.

CAPITULO 1

1. MARCO TEÓRICO

En este capítulo se presenta la fundamentación relacionado con el tema, para proporcionar al lector ideas acerca del tema propuesto. Se encontrarán los conceptos básicos y específicos complementarios de este estudio.

En primer lugar, se hará una breve introducción acerca de las características relacionadas con la tolerancia a fallos, posterior se detallará trabajos relacionados acerca del tema teniendo en cuenta los instrumentos que se han utilizado y la arquitectura propuesta, además se presenta información acerca de las herramientas utilizadas para la ejecución del trabajo. Por último, se describirá brevemente los criterios de valoración del estándar de calidad de software ISO/IEC 25010 para puntualizar el criterio de fiabilidad a utilizar en la presente investigación.

1.1. Tolerancia a fallos

La tolerancia a fallos hace énfasis en que una infraestructura o sistema se beneficien al máximo de sus capacidades internas para asegurar la correcta ejecución de las tareas y por su puesto brindar servicios continuamente a pesar de fallos de hardware o software que se puedan presentar (Carballeira y Mateos, 2017, p. 5).

Además uno de los objetivos fundamentales al diseñar una arquitectura tolerante a fallos es asegurar el correcto funcionamiento inclusive si se presenta alguna falla durante la ejecución de una tarea (Kumari and Kaur, 2018).

Entonces se puede argumentar que la tolerancia a fallos permite que los sistemas puedan seguir funcionando a pesar de fallos que podrían ocurrir en sus componentes, ya sea por errores durante la fase de desarrollo o por problemas en los servidores, en cualesquiera de los dos casos si se llega a la paralización de los servicios del sistema también se debe tomar en cuenta medidas que ayuden a una rápida recuperación de los componentes del sistema y así continuar con las operaciones de manera regular. En la **Figura 1-1** se puede apreciar el funcionamiento de una red tolerante a fallos.

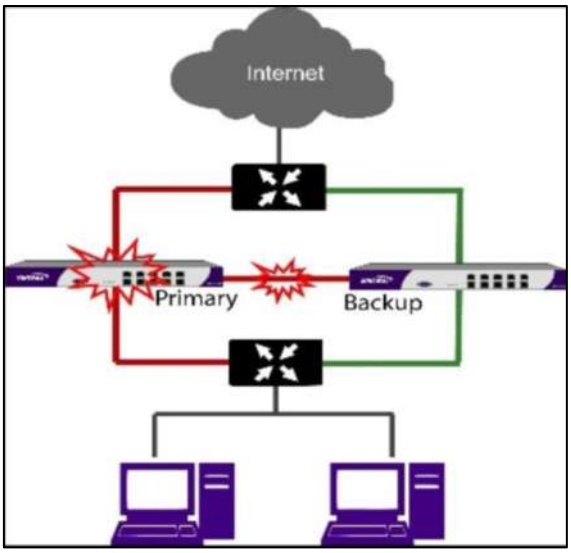


Figura 1-1: Esquema de servidores tolerante a fallos
Fuente: https://es.slideshare.net/JosMiguelBello/elementos-basicos-de-una-arquitectura-de-red

Existen varios niveles donde se pueden aplicar la tolerancia a fallos a continuación se presenta algunos de ellos de acuerdo a lo manifestado por (López, 2006, p. 9).

- **Tolerancia total frente a fallos.** A pesar de daños que se puedan presentar en algunas funcionalidades se puede seguir ejecutante las tareas normalmente, aunque por un período limitado.
- **Degradación controlada.** Las funcionalidades de un sistema operan parcialmente hasta que se reparen las demás.
- **Fallo seguro.** Para garantizar la integridad de datos de los usuarios se ve en la necesidad de una parada temporal total del sistema.

1.2. Alta Disponibilidad

Es uno de los mecanismos que se pretende aplicar para asegurar la tolerancia a fallos, al mismo que varios autores lo definen de la siguiente manera.

En una parte fundamental durante el diseño del sistema y la implementación del mismo para asegurar la continuidad de sus funcionalidades durante un tiempo determinado de medición (García, 2012, p. 3).

Otro profesional relacionado al campo define que la alta disponibilidad es un patrón de diseño de arquitecturas como: red, servicios, sistemas y gracias a esto permite durante un periodo de tiempo la continuidad operacional de las arquitecturas mencionadas, con esto se busca que el

usuario final tenga acceso a los servicios de manera ininterrumpida caso contrario si no puede acceder a dichos servicios se concluye que un sistema no está disponible (Fan et al., 2012).

Se puede contrastar que los sistemas de alta disponibilidad y tolerante a fallos se enfocan en mantener los servicios sin interrupciones. La diferencia radica en que la alta disponibilidad consiste en la replicación de sus elementos, al contrario de la tolerancia a fallos que realiza mejoras solo en un elemento (Paredes, s.f. , pp. 1–2).

Entonces el concepto de alta disponibilidad está muy ligado al objetivo fundamental de la tolerancia a fallos, es decir busca que los sistemas continúen sus operaciones normales de manera ininterrumpida a pesar de los problemas que se podrían suscitar en estos sistemas, también se dice que se considera un sistema de alta disponibilidad si se ha mantenido en funcionamiento en un 99.999% durante todo el año. En la **Figura 2-1** se puede apreciar cómo se da la respuesta a peticiones en el caso de que uno de los servidores se encuentre inactivo o con fallas, depende de las señales de hearthbeat para redirigir peticiones sin perder la comunicación entre clientes y servidores.

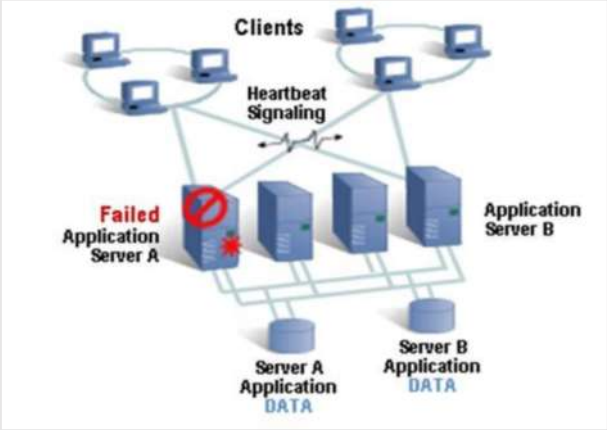


Figura 2-1: Funcionamiento de Servidores de Alta Disponibilidad
Fuente: http://pcecuador.com/web/images/stories/4-0.jpg

La importancia de implementar alta disponibilidad en los servidores se basa en que los sistemas involucrados contienen documentos que son importantes para fines educativos e institucionales, el no poder acceder a ellos durante un periodo de tiempo y más aún cuando se los requiera de manera urgente ocasionará retraso en actividades académicas, trámites, etc.

IBM que es una empresa multinacional de tecnología da a conocer algunas de las ventajas que implica la aplicación de alta disponibilidad (IBM, 2017).

- Reducir el impacto a los clientes y usuarios cuando se pretende dar mantenimientos necesarios a los sistemas.

- Brindan protección por paradas imprevistas provocadas por errores humanos o tecnológicos.
- Restauración de recursos, planes, servicios y procedimientos ante siniestros.
- Reduce el tiempo de inactividad de sistemas o servicios cuando se realizan copias de seguridad.
- Permite una distribución equilibrada de los recursos disponibles.

1.3. Escalabilidad

Es un aspecto de interés para sistemas con proyección cuando el volumen de sus usuarios vaya aumentando cada año por esta razón el conseguir que los servidores sean escalables da una mayor seguridad y confiablidad de que el sistema no llegue al punto que deje de operar. Entre algunos de los autores que define la escalabilidad están los siguientes.

Definida principalmente como la capacidad de que un sistema pueda crecer sin afectar su rendimiento y evitando agregar complejidad al mismo (Bello, 2000).

Adicionalmente se añade que la escalabilidad permite aumentar la capacidad de los sistemas ante un incremento en la concurrencia de usuarios, también pueden ser por otro factor como la cantidad adicional de tráfico en la red todo esto se puede solucionar permitiendo agregar más capacidad de almacenamiento de una manera sencilla para así pueda procesas varias transacciones sin ocasionar pérdidas de servicios. Un dato muy importante es que la aplicación debe ser pensada de una manera escalable durante su diseño y desarrollo, no solo es cuestión de aplicar la escalabilidad en los servidores que lo alojaran (Irey-Núñez, 2014, p. 13).

Tanto Núñez y Bello coinciden que el principal objetivo de la escalabilidad en si es agregar recursos cuando lo requieran sin afectar al funcionamiento de los sistemas. Por ejemplo, si se requiere más espacio en disco, memoria u otros, no se necesitaría detener la ejecución de los sistemas más bien sería un proceso transparente para los usuarios finales.

1.3.1. Tipos de Escalabilidad

Se debe tomar en cuenta el tipo de escalabilidad que se pretende utilizar, esto depende en gran medida de los recursos económicos que disponga, infraestructura y por supuesto que minimice tiempos de inactividad del sistema cuando se realice el escalamiento. A continuación, en la **Figura 3-1** se presenta los tipos que existen.

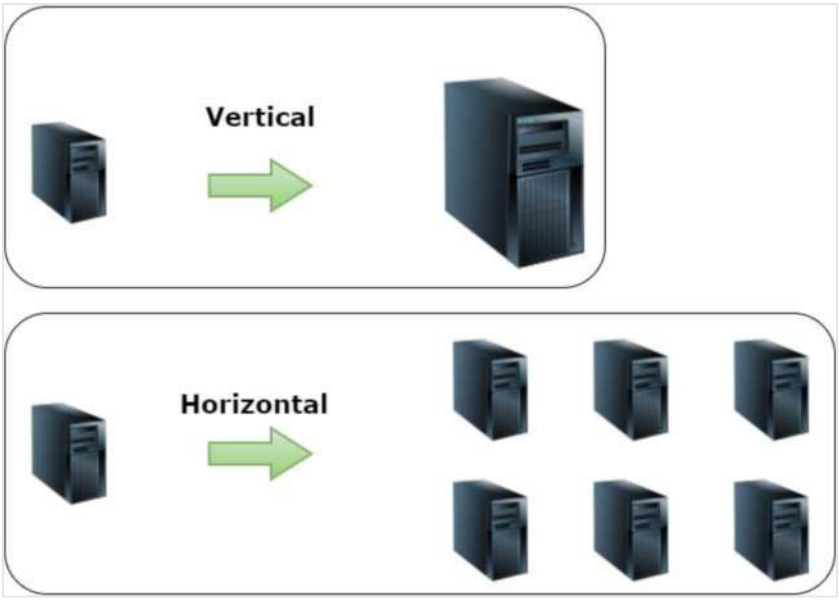


Figura 3-1: Tipos de Escalabilidad
Fuente: (Conscious IT, 2016)

- **Vertical:** La forma de escalamiento más fácil, está orientada básicamente a los componentes hardware es decir si se requiere más memoria o capacidad en disco se tendría que agregar físicamente estos componentes, pero se corre el riesgo de que se llegue al punto donde no se pueda seguir escalando más y esto ocasionaría perdidas económicas inmensas para las empresas por lo que se tendría que migrar el sistema a otros servidores o cambiar la forma de escalamiento (Krzywda et al., 2018)
- **Horizontal :** Este tipo de escalamiento es el más usado actualmente, aunque es un poco difícil vale la pena utilizarlo ya que este no cuenta con un límite y básicamente se basa en la conformación de un Clúster que son servidores denominado nodos los cuales trabajan de manera conjunto como un solo servidor administrados desde un nodo principal (Krzywda et al., 2018)

Entre las ventajas y desventajas que presentan los dos tipos de escalamiento se presenta a en la **Tabla 1-1.**

Tabla 1-1: Criterios de los tipos de Escalamiento

Criterios	Horizontal	Vertical
De implementación fácil	No	Si
Crecimiento ilimitado	Si	No
Puede combinarse con otro tipo de escalamiento.	Si	No
Fácil configuración	No	Si

Necesita Infraestructura grande	Si	No
Soporta balanceo de carga	Si	No
Fácil detección de fallos en servidores	Si	No
Conformación de Clúster	Si	No

Realizado por: Zaruma Jorge.2019

Como se puede observar en la **Tabla 1-1** los criterios que conforman el escalamiento horizontal son más deseables para implementarlo en servidores que requieran la agregación de recursos cuando sea necesario sin tomar en cuenta el límite de los equipos. Además, que cuenta con una administración central de los servidores es decir el nodo principal como se mencionó anteriormente, esto permite al administrador visualizar cuales están fallando y gracias que se puede agregar balanceador de carga hace que el sistema continúe funcionando a pesar de que uno de los nodos del clúster se encuentre inactivo.

1.4. Redundancia

Los fallos pueden ocurrir de manera imprevista por tal motivo el contar con respaldos ya sea de información o sistemas son necesarios, esto se logra por redundancia que permite replicar contenidos en diferentes servidores. Es uno de los mecanismos básicos para obtener sistemas tolerantes a fallos. Varios autores hacer referencia a este tema como, por ejemplo.

Aplicar redundancia en servicios es poseer recursos extras para la culminación de la tarea que se pretenda realizar (Nesmachnow y Iturriaga, 2011, p. 12).

Otro profesional relacionado al área manifiesta que en la redundancia se deben establecer varias réplicas del servicio para asegurar el correcto funcionamiento del sistema, las réplicas pueden ser activas o pasivas depende de cómo establezca la solución. En caso de que se presente un fallo, existe la facilidad de que por un algoritmo de selección de réplica segura, selecciona un reemplazo el cual garantiza garantice que el servicio siga ejecutándose (de Paula et al., 2019).

Las opiniones de estos dos autores concuerdan que con la redundancia se pretende garantizar que las peticiones se realicen sin ningún tipo de interrupciones, para este caso en particular la redundancia se proyectara tanto en aplicaciones y base de datos. En la **Figura 4-1** se puede observar como es un esquema redundante.

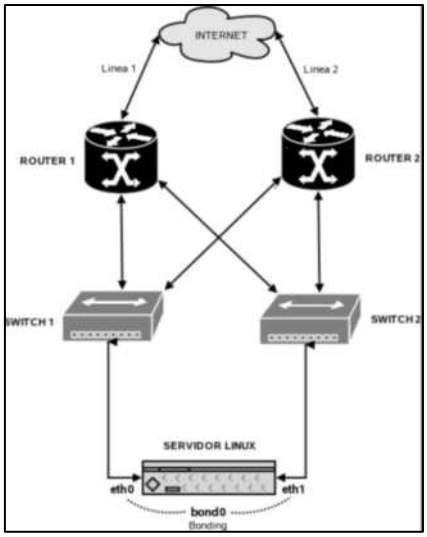


Figura 4-1: Esquema redundante
Fuente: https://e-mc2.net/sites/default/files/articles/red_redundante.png

Entre algunas ventajas y desventajas que se puede mencionar sobre la redundancia destacan los siguientes que se presentan en la **Tabla 2-1**.

Tabla 2-1: Ventajas y desventajas de la escalabilidad

Ventajas	Desventajas
Respaldos de información actualizados	El almacenamiento se reduce porque la información debe ser almacenada de manera duplicada.
Existen variedad herramientas de fácil implementación	Aumenta costos
Permite a través de las réplicas reestablecer la información y acceder a ellos.	
Trabaja en conjunto con balanceadores de carga	
Mejora los tiempos de respuesta.	

Fuente: (Nesmachnow and Iturriaga, 2011)

1.4.1. Tipos de Redundancia

Como menciona Iturriaga y Nesmachnow existen 4 tipos de redundancia principales enfocados a diferentes componentes de los sistemas y es importante conocer información básica los cuales se detalla a continuación (Nesmachnow and Iturriaga, 2011, p. 12).

- **Redundancia en el hardware:** Relacionado con los componentes físicos de un sistema de los mismos que se deben detectar y corregir errores que se presenten. Aquí se puede aplicar redundancia a los siguientes elementos como:
  - Discos
  - Tarjetas de Red
  - Fuentes de Alimentación
  - Sistemas Eléctricos
- **Redundancia en el software:** su principal objetivo es atender fallos de software. Aplicada básicamente en aplicaciones y base de datos a los mismos que se podrán acceder a través de balanceadores de cargas
- **Redundancia en la información:** Se agrega información adicional para detectar y corregir fallas como por ejemplo la inserción de código Hamming o bits de paridad.
- **Redundancia en el tiempo:** Es efectiva sobre todo en los fallos transitorios que se puedan presentar, además de la posibilidad de ser utilizada también cuando otros elementos redundantes identifican un error y ejecutan nuevamente la tarea.

Para obtener una infraestructura de servidores tolerante a fallos es necesario que los tres características mencionadas anteriormente trabajen en conjunto, es decir ser tolerante a fallos implica tener servidores de alta disponibilidad y así asegurar la confiabilidad de los sistemas para esto se requiere que tanto las aplicaciones y base de datos se encuentre distribuidos de forma redundante y como se mencionó que la redundancia implica disminuir capacidad de almacenamiento, para solucionarlo se lo complementa con la escalabilidad horizontal que permitirá seguir agregando los nodos que sean necesarios con el fin de aumentar sus capacidades.

1.5. Fallas comunes en servidores

Toda infraestructura o sistemas en algún punto de su vida útil van a presentar fallos y es de responsabilidad del administrador poder solucionarlos lo más rápido posible, o su vez implementar métodos que permita contener los fallos que se puedan presentar y así evitar inactividad de los servicios (Carballeira y Mateos, 2017).

1.5.1. Causa de fallos

Entre las posibles causas que ocasionan fallos se pueden mencionar las siguientes.

- **Especificaciones incorrectas:** Hacen referencia a la presencia de errores en el diseño tanto de hardware como de software.

- **Errores de implementación:** Conocida también como transformación incorrecta de las especificaciones.
- **Componentes defectuosos:** Este tipo de errores aparecen cuando no se realiza un estudio minucioso del dimensionamiento de la infraestructura lo que significa que los recursos del servidor estarán sobrecargados.
- **Perturbaciones externas:** Son variaciones climáticas que afectan a la infraestructura esta causa de fallos es la más destructiva ya que puede llegar a destruir los equipos.

1.5.2. Duración de fallos

Además, se toma en cuenta que las fallas pueden ser de hardware o software y pueden ser de distintos tipos de duración como:

- **Permanentes:** Su duración dependerá de varios factores como el tiempo de relación del componente o si es el caso el lapso que se demore en sustituir completamente el elemento averiado.
- **Transitorios:** Estos aparecen cada cierto periodo de tiempo, lo que permite a los administradores tener un plan de contingencia para evitar la inactividad prolongadas de sus servicios.
- **Intermitentes:** Los fallos con este tipo de duración son imperceptibles, pero es de gran importancia tratarlos a tiempo para evitar que se convierta en un problema mucho mayor posteriormente.

1.5.3. Medidas para evitar fallos

Entres algunas de las medidas a tomar en cuenta para evitar fallos se listan las siguientes:

- **Prevención:** Para lograr efectuar correctamente esta actividad se debe tomar en cuenta estudios minuciosos de los componentes de un sistema y así constatar los puntos posibles de fallos para aplicar métodos que permiten controlarlo adecuadamente.
- **Enmascaramiento:** Es la aplicación de técnicas profesionales que permitan cubrir los fallos presentes en la infraestructura para que los usuarios finales eviten interferencias en sus servicios.
- **Tolerancia:** En este punto uno de los componentes ceso sus funciones ya sea por diferentes circunstancias como, por ejemplo: capacidad de disco insuficiente, falta de memoria RAM, etc. Por tal motivo la implementación de tolerancia es importante ante los fallos por lo que permite seguir funcionando a pesar de haber ocurrido los problemas mencionados anteriormente, y esto lo puede lograr en base a la redundancia ya que es su característica principal.

1.6. Trabajos relacionados con tolerancia a fallos

1.6.1. Escalabilidad y Disponibilidad en Infraestructuras de Servicios Orientadas a la web, basadas en Cloud Computing

Para la ejecución de este proyecto Luis Nicolás Calmels utilizó las siguientes herramientas tanto para el desarrollo como para la implementación (Calmels, 2004):

- Centos 6.5 (GNU/Linux) como el sistema operativo
- Apache en su versión 2.2. con servicio web
- Varnish cache el cual permite acelerar aplicaciones web
- MySQLPercona como gestor de base de datos.
- PHP versión 5.3
- Se proveerá un servicio de FTP (File Transfer Protocol) basado en el software.

Si bien esto se ejecutara bajo la herramienta de administración Nubity que permite monitoreo, configuración y mantenimiento de infraestructuras en la nube. La misma que fue orientada a obtener una infraestructura escalable y con alta disponibilidad. Por tal motivo en la **Figura 5-1** se puede apreciar la infraestructura que se implementó en la nube como sus servicios o más conocido como IaaS que es un modelo básico que se le asigna al usuario con una capacidad de almacenamiento considerable, así como la posibilidad de acceder a los sistemas operativos virtualizados o sus servicios por medio de internet (Calmels, 2004, p. 10).

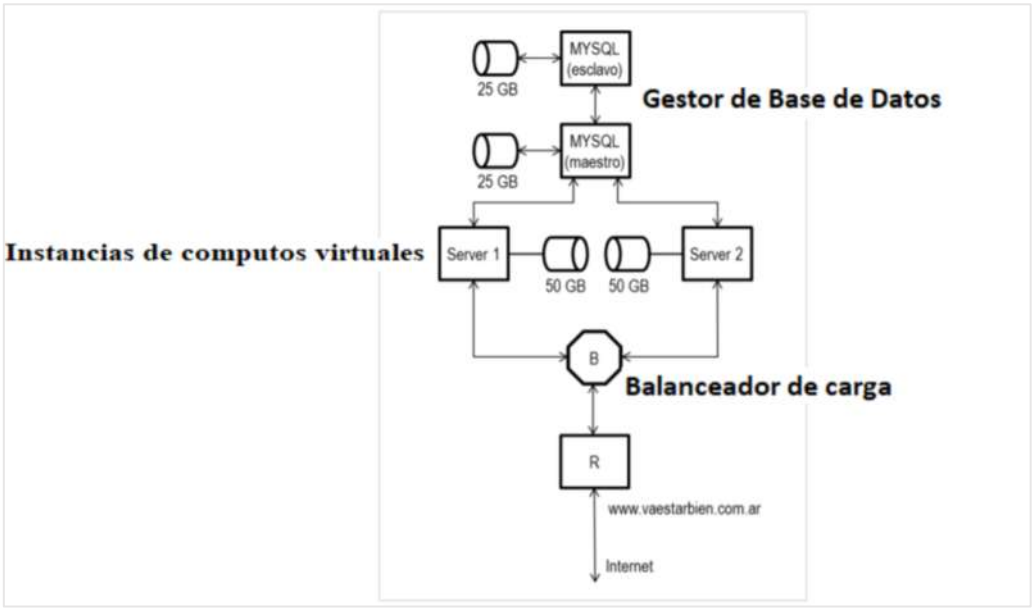


Figura 5-1: Infraestructura en la nube sobre IaaS
Fuente: (Calmels, 2004, p. 11)

Y como se puede observar en la **Figura 5-1** la arquitectura propuesta consta de los siguientes componentes:

- Balanceo de carga
- Instancias de cómputos virtuales con capacidad de 50 GB cada uno
- Gestores de Base de datos con un entorno maestro-esclavo con el fin de obtener replicación de datos cada uno con capacidad de 25 GB

Para comprobar que la infraestructura funciono correctamente ejecutaron varias pruebas como:

- **Crecimiento del tráfico:** Esto fue un problema donde plantearon dos soluciones: una es escalar horizontalmente es decir aumentar más nodos a la infraestructura la otra es escalar verticalmente lo que significa agregar más recursos a los nodos. En cualquiera de estas soluciones se logró que los sistemas no queden inactivos.
- **Caída de un servidor web:** Como se puede observar en la Figura 12-1 el servidor web se encuentra instalado en dos servidores físicos lo que permitió mediante el balanceador redirigir el tráfico hacia el nodo que se encuentra en funcionamiento hasta que se restablezca el nodo inactivo.
- **Caída del master MySQL:** En este escenario la aplicación definitivamente tendrá que estar inactivo ya que no permite en esta infraestructura no se promueve automáticamente el slave a maestro, por ende, tuvieron que realizar este proceso ingresando al servidor de base de datos.
- **Ataque DOS sobre la infraestructura:** Son ataques comúnmente usados que generan gran tráfico en la red sobre un sitio en particular ocasionando la inaccesibilidad a dicho sitio a los usuarios. Este problema se afrontó monitorizando los logs de Apache, con esto el administrador bloquea la dirección ip que ocasiona dicho ataque.
- **Caída del Balanceador:** Considerada como unos de los servicios críticos para garantizar alta disponibilidad en este caso delegaron esta responsabilidad a AWS (Servicios Web Amazon) líder en brindar soluciones en la nube, el mismo que garantizaba este servicio por medio de balanceadores sobre los balanceadores.
- **Hackeo sobre el sitio Web:** La misma que puede ser ocasionada por ataque por fuerza bruta, Mysql Injection, spoofing, o ingeniería social.

1.6.2. Configuración de un clúster de alta disponibilidad y balanceo de carga en linux para satisfacer gran demanda web y servicios de resolución de nombres

Para la elaboración de esta tesis Andrés Bustos ha utilizado las siguientes herramientas las mismas que sean definidas, para tener una idea básica del funcionamiento de cada una (Bustos, 2007b).

HA-Oscar: es una versión mejorada de Oscar que fue un software de paquetes y que ofrecía alto rendimiento, en esta versión ofrece mecanismos de alta disponibilidad, escalabilidad, seguridad, alto rendimiento, balanceo de carga entre otros, esto permite evitar fallos al acceder a los nodos del clúster.

Linux Virtual Server (LVS): le permite crear un clúster de servidores interconectados entre si administrados por el nodo director que es el que posee las mejores características del conjunto de servidores y es el que está al frente para distribuir las funciones a cada nodo, además cuenta con características de alta disponibilidad y balanceo de carga sobre el sistema operativo GNU/Linux.

Round Robin: Es le algoritmo utilizado para el balanceo de carga de LVS que consiste en redirigir el tráfico de manera cíclica, es decir, si llega una petición se lo envía al primer nodo, la segunda petición va al segundo y así sucesivamente.

Para la configuración utilizaron además los siguientes recursos:

- Cliente para acceder al servicio
- Router que fue el gateway del clúster
- Un servidor director y un standby con tarjeta de red en donde se configuro la ip virtual y la ip real
- Switch de comunicaciones para el clúster
- Servidores físicos para la conexión con el nodo director.

La infraestructura que se propuso en este trabajo se presenta en la **Figura 6-1**, donde se visualiza de manera gráfica.

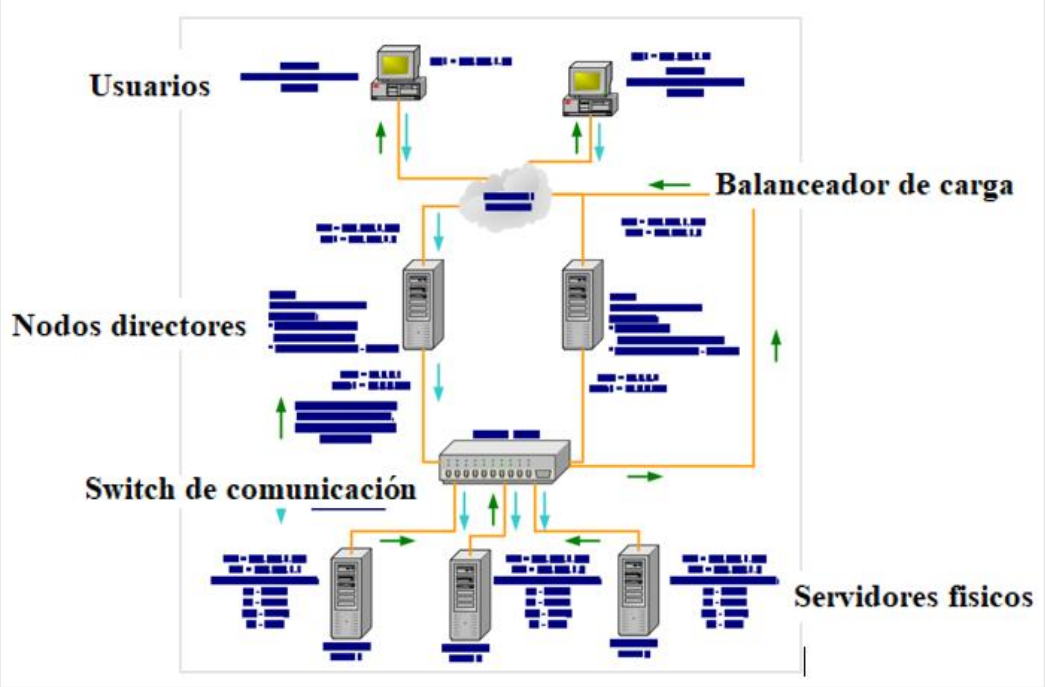


Figura 6-1: Arquitectura del clúster de balanceo de carga y alta disponibilidad.
Fuente: (Calmels, 2004)

Como se puede observar en la **Figura 6-1** la arquitectura se encuentra conformado por bloques que se explica de abajo hacia arriba para tener una idea clara de los componentes involucrados que son los siguientes:

- **Primer bloque:** utilizaron 3 servidores físicos,
- **Segundo bloque:** consiste en un switch para la comunicación entre los nodos directores y nodos reales
- **Tercer bloque:** conformado de dos nodos directores
- **Cuarto bloque:** el balanceador de carga
- **Quinto bloque:** son los usuarios que acceden a los servicios.

Para comprobar que la infraestructura funciono correctamente ejecutaron varias pruebas como:

- **Pruebas de funcionamiento de LVS:** esta prueba fue realizada mediante la utilización de la herramienta telnet hacia la interfaz virtual del clúster, se esperó que responda uno de los servidores reales del clúster.

Para la realización de esta prueba en el nodo director se agregó los tres servidores reales con el puerto 23, habilitar que se acepte conexiones tcp y se asignó que el algoritmo de balaceo es Round robín luego de realizar todo este proceso. Se envió peticiones telnet al nodo director y el resultado se puede observar en la **Figura 7-1** se estableció la conexión exitosamente y el funcionamiento de round robín se aplicó con éxito.

Figura 7-1: Conexiones a servidores reales por telnet
Fuente: (Bustos, 2007b, p. 187)

- **Prueba de funcionamiento de LVS y HA-Oscar:** centrada principalmente en evidenciar la respuesta del clúster ante un alto tráfico. Para lograr esto se utilizó el generador de trafico de Ecperf que permite una carga de trabajo de rendimiento EJB TM que es real, escalable y captura la esencia de la razón por la que existen los modelos de componentes.

Este ambiente de pruebas es similar al anterior con la diferencia de que se genera tráfico hacia el puerto 80, además que se interrumpió el funcionamiento del nodo principal así comprando que HA-Oscar entre en funcionamiento y permitió que el nodo standby asuma las responsabilidades del nodo principal.

Un trabajo que demuestra el gran funcionamiento de un clúster, una de las mejores características que ofrece es sin duda la facilidad de administración y la capacidad de elegir entre varios algoritmos para el balanceo de carga.

1.7. Servidor de Aplicaciones

Sin ahondar sobre este tema se puede definir que es un programa o software que puede ser instalado en un equipo al cual pueden acceder varios usuarios a realizar peticiones en aplicaciones instaladas previamente en el servidor de aplicaciones, esta estructura hace referencia a la arquitectura cliente servidor. Para su elección se debe tener en cuenta la plataforma de desarrollo

de las aplicaciones, en este caso como las aplicaciones que se van a desplegar se han desarrollado en JAVA (Díaz, 2010).

Además se puede complementar que el servidor de aplicaciones puede establecer conexión con otros tipos de servidores como, por ejemplo el de una base de datos y archivos como se observar en la **Figura 8-1**(Groussard, 2010, p. 16).

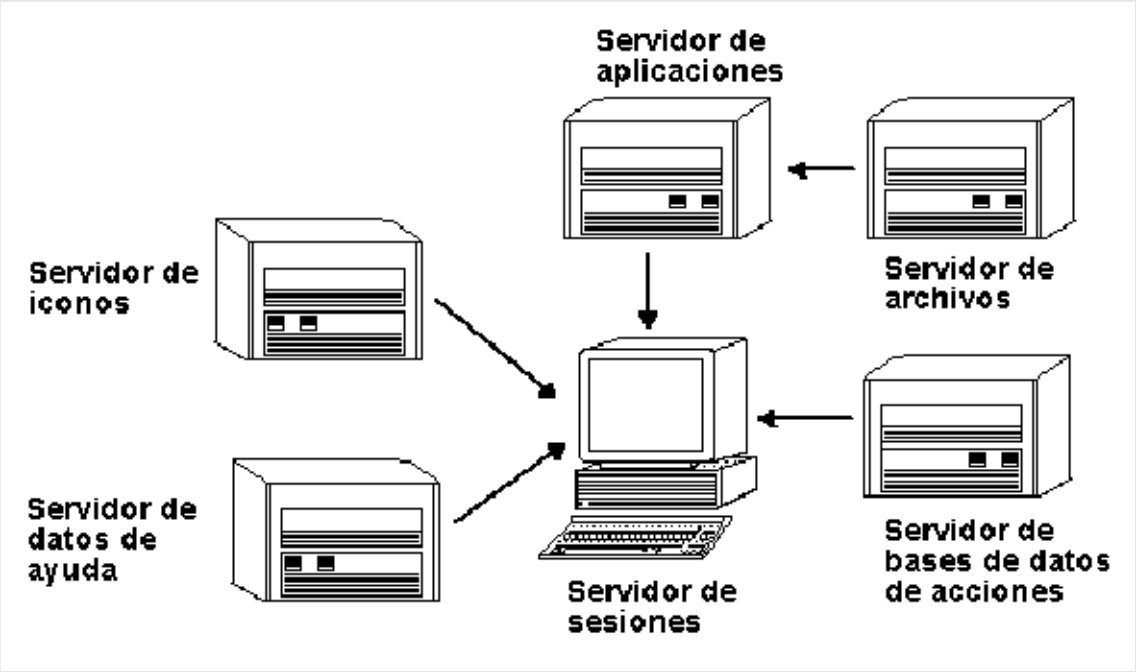


Figura 8-1: Funcionamiento de un Servidor de Aplicaciones
Fuente: https://docs.oracle.com/cd/E19683-01/816-4016/images/complexapp.tiff.gif

1.7.1. Payara

En su página oficial se menciona que es un servidor de aplicaciones de código abierto totalmente compatible y fácil de usar. La arquitectura de Payara Server es innovadora, nativa de la nube y optimizada para implementaciones de producción. El servidor de aplicaciones, compatible con Eclipse MicroProfile, está creado y respaldado por un equipo de ingenieros de DevOps dedicado al desarrollo y mantenimiento continuo del software de código abierto y comprometido con la optimización del servidor Payara como la mejor opción para la producción de aplicaciones Java EE (Payara, 2018).

1.8. Servidor Web

Básicamente es un servidor de archivos el cual recibe peticiones de los usuarios mediante el protocolo http generalmente en el puerto 8080, y solo devuelve el recurso solicitado. Dependiendo del servidor web de su elección obtendrá varios beneficios y cualidades que les permitirá obtener el máximo rendimiento, es así que puede funcionar como un balanceador de carga, re direccionar peticiones, crear reglas lo cual nos permitirá mejorar la interactividad con los usuarios como se observa en la **Figura 9-1** (Groussard, 2010, p. 16).

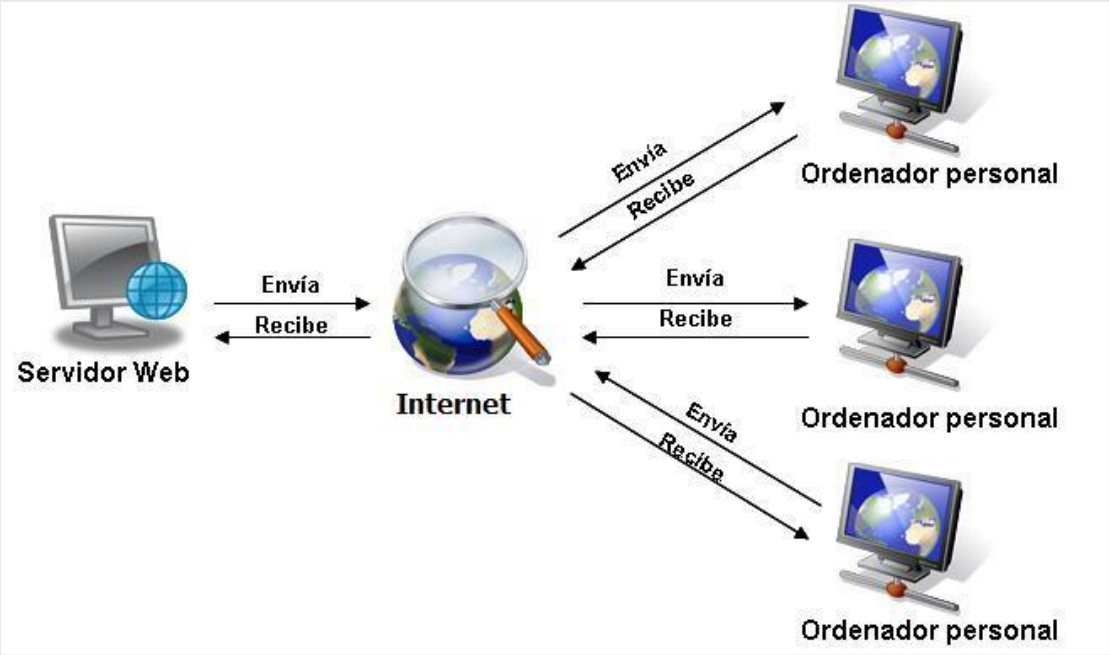


Figura 9-1: Funcionamiento de un Servidor Web
Fuente: http://s.culturacion.com/wp-content/uploads/2011/12/figura21.jpg

1.8.1. Apache

Es un servidor web usado a nivel mundial Open Source, y gracias a cualidades como ser estable, multiplataforma, modular, una comunidad activa y mediante su archivo de configuración httpd.conf se puede ir adaptando a nuestras necesidades, además permite la monitorización de sus acciones mediante su archivo de logs (Baş Seyyar et al., 2018).

Es de fácil configuración, pero así mismo contiene características de seguridad potentes que garantizan peticiones confiables. Como su comunidad está activa en todo el mundo constantemente presentan mejorías, brindan soporte, etc. Está conformado por varios módulos que se pueden desactivar o activar según las necesidades de su empresa, cuenta con la posibilidad de crear host virtuales además de realzar balanceadores de cargar y poder crear nuestras propias reglas dentro de los hosts virtuales para un manejo adecuado de las peticiones de los usuarios.

Entres algunas de las características más importantes se puede mencionar las siguientes

- Es uno de los servidores web más utilizados a nivel mundial
- Es un sistema multiplataforma
- Posee infinidad de paquetes y módulos que nos permiten trabajar con gran cantidad
- de lenguajes de programación web, así como intérpretes de SQL y otras funciones.
- Permite transacciones seguras mediante SSL lo cual permite garantizar la confiabilidad de sus transacciones
- Contiene soporte para Hosts virtuales

1.8.2. Host virtuales

Este término es usado regularmente cuando se ejecuta más de un sitio web en una sola máquina, esto es generalmente utilizado por proveedores de internet y proveedores de hosting, permitiéndoles ahorrar dinero porque los recursos se comparten entre los clientes (Hadi, 2015).
Existen dos tipos de host virtuales:

- **Basadas en ip:** Esta fue una de las primeras opciones implementadas para los hosts virtuales, su funcionamiento consistía en tener configurado el servidor para que escuche por diferentes direcciones ip como se observa en la **Figura 10-1**, también proveía la posibilidad de que sea un puerto diferente (Hadi, 2015, p. 10).

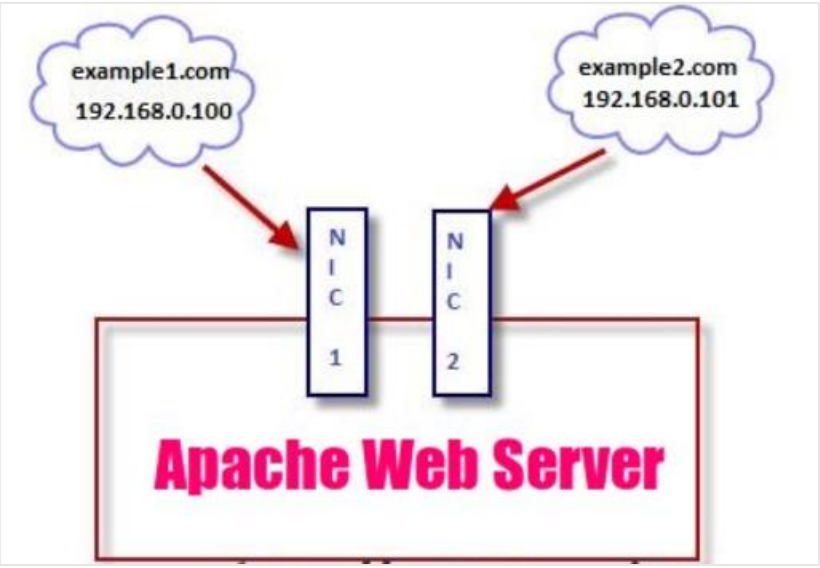


Figura 10-1: Hosts virtuales basados en ip.
Fuente: https://www.tecmint.com/wp-content/uploads/2014/01/IP-based-Virtual-Hosting.jpeg

- **Basadas en nombres:** La gran desventaja de la anterior propuesta es que se necesitaría tener varias direcciones asociadas con el inconveniente de la gran demanda de direcciones

ipv4, por tal motivo el tener asociado varios dominios a una misma dirección ip es una gran opción como se puede visualizar en la **Figura 11-1**, aunque escuchen por la misma dirección y puerto los hosts virtuales permiten identificar las peticiones de acuerdo al dominio que se esté utilizando (Hadi, 2015, p. 11).

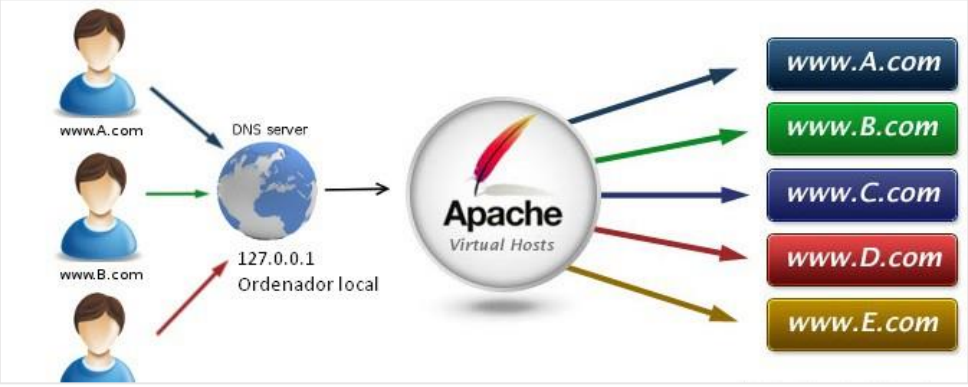


Figura 11-1: Hosts virtuales basados en nombres.
Fuente: https://medium.com/@jhordydelaguila/creando-y-configurando-virtual-host-con-apache-para-xampp-windows-f90c2b0527ac

1.9. Servidor de Base de Datos

Se puede mencionar que es un programa que se instala en un sistema y tiene como objetivo almacenar grandes cantidades de información que puede ser usado simultáneamente por varios clientes de manera eficiente, también puede ser accedido mediante herramientas de su preferencia para su administración, generalmente mantiene una estrecha relación con el servidor de aplicaciones del mismo que se puede acceder para utilizar los datos en los sistemas que lo requieran como se observa en la **Figura 12-1** (Zea Ordóñez et al., 2017)

Figura 12-1: Funcionamiento de un Servidor de Base de Datos
Realizado por: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQA99mAyu01PdCFpb1IZHzTZ--WCdAb7zb0aDnbdD5ws_Wa3m9V

Entre algunos de las características de un servidor de base de datos se puede mencionar los siguientes:

- **Consultas no predefinidas y complejas:** Los usuarios podrán realizar consultas de cualquier tipo y el sistema gestor de base de datos (SGBD) tendrá que responder inmediatamente a ellos. (Camps Paré y Universitat Oberta de Catalunya, 2005, p. 15)
- **Flexibilidad e independencia:** Esto permitirá que se puedan hacer cambios sin que afecte al funcionamiento normal de las aplicaciones que hacen uso de la base de datos (Camps y Casillas, 2005, pp. 15–16).
- **Problemas de la redundancia:** Aquí se permitirá al usuario definir qué datos serán redundantes, pero se deben establecer mecanismos que ayuden a que la actualización de estos datos será de manera automática a través del SGBD (Camps y Casillas, 2005, pp. 16–17).
- **Integridad de los datos:** Hay varios motivos por lo que se puede perder la consistencia de nuestros datos tales como: errores en programas, de operación humana, fallos en disco, solicitudes incompletas por falta de energía eléctrica, etc.

  Para que no suceda esto se establecen reglas a nivel de la aplicación y del SGBD que permite que los errores mencionados anteriormente no tengan un mayor impacto en la información que almacenamos (Camps y Casillas, 2005, p. 18).
- **Concurrencia de usuarios:** Como ya se mencionó anteriormente un servidor de base de datos debe permitir que varios clientes accedan de manera simultánea a la base de datos y realizar todas las peticiones correspondientes. (Camps y Casillas, 2005, pp. 18–19)
- **Seguridad:** Por medio de este objetivo se pretende definir autorizaciones, accesos y la confidencialidad de la información. Definidas mediante la asignación de usuarios y grupos los mismo que tienen acceso a niveles es decir puede ser de manera global a la base de datos, nivel de tablas, entidades o de atributos (Camps and Casillas, 2005, p. 21).

1.9.1. Postgresql

Como se mencionó anteriormente todas las características que tiene un servidor de base de datos se agrega que postgresql es una Base de datos objeto relacional o sus siglas en ingles ORDBMS basado en el proyecto Postgresql versión 4.2, soporta la mayoría del estándar ANSI SQL, posibilidad de almacenar grandes objetos binarios incluyendo, imágenes, sonidos, y videos, cuenta con herramientas para su administración, opciones y características de seguridad y alternativas para alta disponibilidad de servicio y por supuesto es multiplataforma (Afyouni et al., 2014).

En la **Figura 13-1**, se presenta un cuadro comparativo de las características más importantes que se deben tomar en cuenta al elegir un SGBD.

Sistema	MySQL	PostgreSQL	SAP DB
Versión	Mysql-3.23.41	PostgreSQL 7.1.3	SAP DB Version 7.3
Licencia	GPL	BSD	GPL
Cumplimiento con estándar SQL	Media	Alta	-
Velocidad	Media/Alta	Media	-
Estabilidad	Alta / Muy Alta	Alta	-
Integridad de datos	NO	Si	Si
Seguridad	Alta	Media	-
Soporte de LOCKING y CONCURRENCIA	Media	Alta	-
Soporte de Vistas	No (Planeada v4.2)	Si	Si
Soporte Subconsultas	No (Planeada v4.1)	Si	Si
Replicacion	Si	Si	-
Procedimientos almacenados	No	Si	Si
Soporte Unicode	NO	Si	-
Soporte Disparadores	No	Si	Si
Integridad referencial	No	Si	Si
Interfaces de programación	ODBC, JDBC, C/C++, OLEDB, Delphi, Perl, Python, PHP	ODBC, JDBC, C/C++, SQL embebido (en C), Tcl/Tk,Perl, Python, PHP	ODBC ,JDBC, C/C++, Precompilado(SQL Embebido),Perl, Python, PHP
Tipos de Tablas alternativas	ISAM, MYISAM, BerkeleyDB, InnoDB, HEAP, MERGE, Gemini	PostgreSQL mantiene su propio sistema de tipos de tablas	-
Transacciones	si	Si	-
Claves foráneas	NO (Planeado v4.0)	Si	-
Backups en caliente	Si	Si	-

Figura 13-1: Cuadro comparativo de Postgresql con otros SGBD
Fuente: (Patricio Denzer,2012)

1.9.2. Citus

Es una extensión de Postgresql creada para escalar masivamente la base de datos horizontalmente, al ser una extensión mantiene compatibilidad con las herramientas Postgresql existentes. Gracias a esto podemos distribuir los datos y consultas en un grupo de máquinas múltiples a la cual se denominará clúster (CitusData, 2018).

1.10. Clúster

Actualmente es un término muy utilizado en el área de la informática y se basa principalmente en agrupar un conjunto de servidores para que trabajen como si fuera un único servidor para realizar procesos en paralelo o distribuidos, estos son conectados mediante una red y son realmente escalables. Dependiendo de su naturaleza un clúster puede ser homogéneo si los servidores poseen características similares y heterogéneos cuando cada servidor posee características únicas (Cáceres, 2012, pp. 11–12).

1.10.1. Componentes

Para la creación de un clúster se debe tomar en cuenta los siguientes elementos (Cáceres, 2012, pp. 16–18).

- **Nodo:** es como se denomina a un servidor que es parte del clúster.
- **Sistema operativo:** encargado de la comunicación con el hardware y donde se instalarán todos los elementos necesarios para la creación del clúster.
- **Conexión a red:** es el medio de comunicación entre los nodos.
- **Middleware:** Es el software que se encuentra entre dos programas su función principal es el de monitorear los recursos de los nodos para saber si existen problemas y así evitar distribuir trabajo a un nodo inactivo.
- **Recursos o Servicios:** son los elementos que serán utilizados en el clúster por ejemplo un servidor web, servidor de base de datos, servidor de aplicaciones, etc.

1.10.2. Tipos

Dependiendo de la utilización de un clúster o el área para el que se ha creado se pueden utilizar uno de los siguientes:

- **Alto rendimiento:** Utilizado principalmente para el campo de investigaciones científicas donde se necesitan realizar gran cantidad de cálculos para la resolución de problemas para lo cual se necesita gran capacidad de procesamiento para lo cual este tipo de clúster está enfocado en compartir el procesamiento de cada uno de los nodos para ejecutar dichas tareas (Infosegur, 2013).
- **Alta disponibilidad:** Enfocado en mantener uno o varios servicios de manera ininterrumpida consta de un nodo activo y uno pasivo que son monitoreados por un middleware, para saber el estado de cada uno de los nodos y establecer cuál de los dos provee el servicio (Cáceres, 2012, p. 15).
- **Balanceo de Carga:** Su función principal es el de distribuir la carga de trabajo entre todos los nodos que conforman el clúster, se puede ampliar su capacidad fácilmente y ante la inactividad de uno de sus nodos los servicios pueden seguir en funcionamiento (Sinisterra et al., 2012, p. 95).

1.11. Herramientas para configurar alta disponibilidad de servicios

Existen dos herramientas principales que son las siguientes:

1.11.1. Corosync

Proporciona la comunicación entre los nodos que conforman el clúster, Para ello, utiliza direcciones multicast de forma que todos los nodos que integran un clúster utilizan la misma dirección para comunicarse entre ellos (Pons and Bonet, 2014, p. 3).

Características

Entre sus características más relevantes se puede listar las siguientes (Corosync, 2018).

- **Sincronía virtual:** Un modelo de comunicación de grupo de proceso cerrado con garantías de sincronía virtual para crear máquinas de estado replicadas.
- **Información:** Una base de datos de configuración y estadísticas en la memoria que proporciona la capacidad de configurar, recuperar y recibir notificaciones de cambios de información.
- **Disponibilidad:** Un administrador de disponibilidad simple que reinicia el proceso de aplicación cuando falla.
- **Quórum:** Un sistema de quórum que notifica a las aplicaciones cuando se alcanza o se pierde el quórum.

1.11.2. Pacemaker

Permite al administrador tener el dominio total de los recursos en cada uno de los nodos del clúster, es un proyecto basado en heartbeat el cual permite un monitoreo de todos los nodos del clúster y los servicios asociados a ellos, esto con el fin de verificar continuamente si un nodo está activo, caso contrario si pasa a un estado inactivo esta herramienta automáticamente inicia los servicios en otro nodo activo y continúa trabajando normalmente (Tejero, 2014, p. 7).

Características

Pacemarker ha mejorado las deficiencias presentadas con heartbeat 1 y 2 entre algunas de ellas se puede mencionar las siguientes (ClusterLabs, 2018).

- Una sintaxis más intuitiva.
- Umbrales de fallos (migración) y tiempos de espera
- Herramienta para hacer cambios de configuración fuera de línea.
- Una herramienta de configuración de línea de comandos unificada que oculta el xml subyacente

- Las reglas, los atributos de instancia, los atributos de meta y los conjuntos de operaciones se pueden definir una vez y hacer referencia en varios lugares
- La capacidad de conectarse desde máquinas no agrupadas
- Permitir que se activen acciones recurrentes en momentos conocidos
- Un esquema de configuración basado en RelaxNG más potente

1.11.3. Pscd

Encargado de la comunicación cifrada por credenciales entre los nodos que conforman el clúster y además es el mecanismo de control del mismo (Pons and Bonet, 2014, p. 3), en la **Tabla 3-1** se destacan los comandos básicos de esta herramienta.

Tabla 3-1: Comandos para administración del clúster

Opción	**Descripción**
Resource	Maneja los recursos del clúster
Cluster	Maneja y configura las opciones del clúster y de los nodos del clúster.
Stonith	Configura la política de desactivación y reinicio de un nodo cuando el clúster considera que funciona mal.
Property	Asigna las propiedades de Pacemaker.
Constraint	Asigna las restricciones existentes entre los recursos del clúster
Status	Muestra el estado del clúster
Config	Muestra la configuración completa del clúster.

Fuente: (Pons and Bonet, 2014, p. 2)

1.12. SCRUM

En su página oficial SCRUM se ha definido que

"Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el trabajo en productos complejos desde principios de los años 90. Scrum no es un proceso, una técnica o método definitivo. En lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las técnicas de gestión de producto y las técnicas de trabajo de modo que podamos mejorar continuamente el producto, el equipo y el entorno de trabajo", el proceso de scrum puede verse claramente reflejado todos los componentes que intervienen en la **Figura 14-1** (Scrum, 2017).

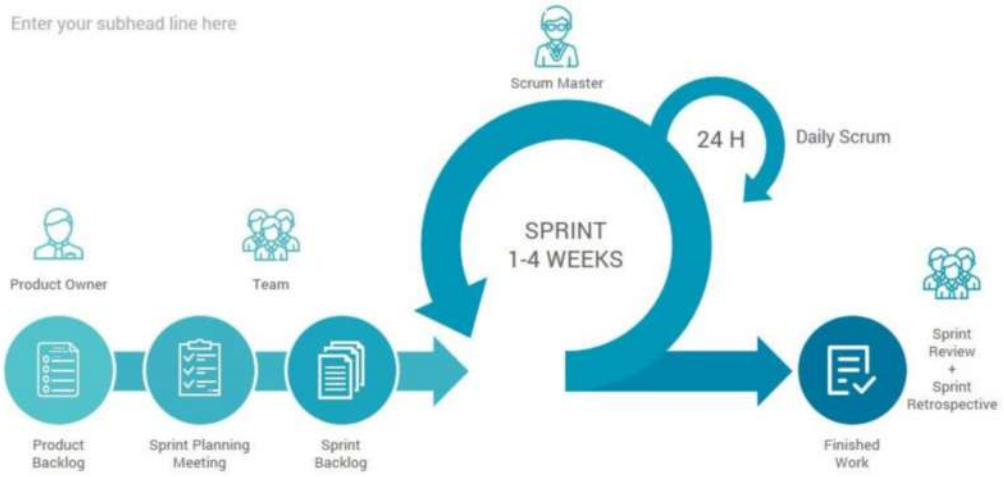


Figura 14-1: Procesos de Scrum
Fuente: https://winred.com/uploads/contenidos_usrs/originales/2566824_013_scrum_process_20180618180200.jpg

La metodología scrum consta de tres partes principales como lo son:

a) Equipo

- **Product Owner**

Responsable de maximizar el valor del producto resultante del trabajo del equipo de desarrollo, entre las principales responsabilidades se puede mencionar las siguientes (Mahalakshmi y Sundararajan, 2013).

  - Representante del cliente
  - Trabaja con el equipo.
  - Mantiene la cartera de productos.
  - Prioriza los requisitos del producto
  - Interfaz entre el equipo y la empresa.

- **Scrum Master**

Es responsable de promover y respaldar a Scrum como se define en la guía de Scrum, ayuda a los que están fuera del equipo scrum a comprender cuáles de sus interacciones son útiles y cuáles no (Scrum, 2017).

Entre las responsabilidades más notorias se pueden listar las siguientes (Mahalakshmi and Sundararajan, 2013).

  - Ayuda a los miembros del equipo y es responsable de evitar obstáculos
  - Scrum master tiene un papel de liderazgo sobre el equipo.
  - Ser responsable de gestionar el proceso.
  - Responsable de ejecutar el scrum diario, sprint y reuniones de planificación
  - Mantiene la documentación requerida.

- **Development team**

Está formado por profesionales que realizan el trabajo de entregar un Incremento potencialmente liberable de producto realizado al final de cada sprint, los equipos de desarrollo están estructurados y habilitados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y eficacia general del equipo de desarrollo (Scrum, 2017).

Entre las principales cualidades que debe cumplir un equipo son (Scrum, 2017).

- Auto organizados
- Multifuncionales
- Comprometidos

b) Eventos

Se utilizan en Scrum para crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum, se deben agregar los siguientes eventos: sprint, planificación del sprint, scrum diario, revisión del sprint y retrospectiva del sprint, para permitir la transparencia crítica y la inspección. Si no se incluye ninguno de estos eventos, se reduce la transparencia y se pierde la oportunidad de inspeccionar y adaptar (Scrum, 2017).

c) Artefactos

Está conformado por tres componentes que son:

- **Product Backlog:** Es una lista ordenada de requerimientos definidas para un producto, las mismas que son ordenadas por el propietario del producto quien a su vez las prioriza durante la finalización de cada sprint (Mahalakshmi y Sundararajan, 2013, p. 194).
- **Sprint backlog:** Es una lista de trabajo que debe ser desarrollado en el siguiente sprint, el requisito no debe cambiarse durante la ejecución del sprint (Mahalakshmi y Sundararajan, 2013, p. 194).
- **Incremento:** Es la suma de todos los elementos de Product Backlog completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores (Scrum, 2017).

1.13. Estándar ISO/IEC 25010

Perteneciente a la familia de las normas ISO/IEC 25000 conocida System and Software Quality Requirements and Evaluation o sus siglas en ingles SQuaRE el mismo que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software (Vaca, 2018).

SQuaRE maneja la calidad del producto software en tres fases que son la siguientes:

- **Calidad Interna:** producto software en desarrollo.
- **Calidad Externa:** producto software en funcionamiento.
- **Calidad en Uso:** producto software en uso.

La norma ISO/EIC 25010 está orienta a describir el modelo de calidad de un producto software y la calidad de uso teniendo en cuenta las características y subcaracterísticas para evaluar el producto software (ISO25000, 2018).

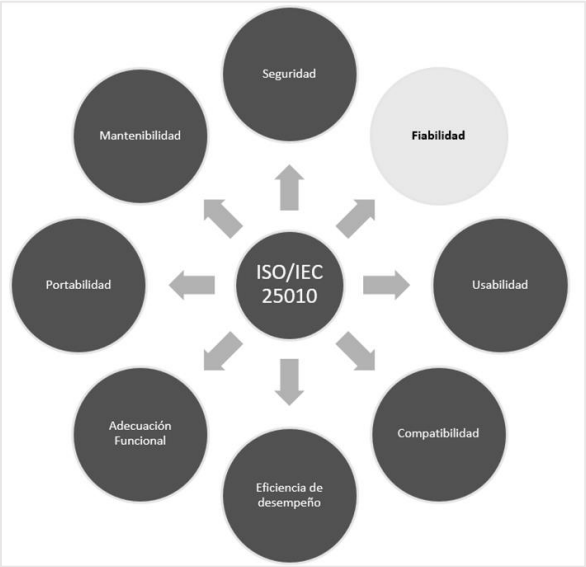


Figura 15-1: Características del estándar ISO/IEC 25010
Realizado por: Zaruma Jorge. 2019

Como se puede observar en la **Figura 15-1** la norma consta de 8 características principales para medir la calidad del producto software, pero en el presente trabajo de titulación voy a utilizar la característica de fiabilidad.

1.13.1. Fiabilidad

Es la capacidad de un sistema o un componente de desempeñar sus funciones específicas bajo distintas condiciones o periodos de tiempos determinados. Esta característica a su vez consta de 5 subcaracterísticas que se describen a continuación (Febrero et al., 2016)

- **Madurez:** Satisface las necesidades de fiabilidad bajo condiciones normales.
- **Disponibilidad:** Es la capacidad del sistema o un componente de estar disponible cuando se lo requiera.
- **Tolerancia a fallos:** Es la capacidad del sistema o un componente de seguir en funcionamiento a pesar de los fallos que puedan ocurrir, tanto de hardware o de software.
- **Capacidad de recuperación:** Es la capacidad de recuperar datos que fueron afectados y lograr restablecer el sistema a un estado deseado en caso de presentarse fallos o interrupciones que afecten al funcionamiento del sistema.

CAPÍTULO II

2. MARCO METODOLÓGICO

La finalidad de este capítulo es detallar el desarrollo de la metodología que se ha seleccionado como lo es SCRUM en el presente trabajo de titulación, esta metodología se emplea para proyectos de desarrollo ágil, además es flexible ante los cambios que se puedan presentar y permite entregas parciales del producto final que deben ser validadas por los involucrados en el proyecto para que luego sean puesto en producción, de esta manera se van integrando todos los requerimientos que se vayan a desarrollar hasta llegar a la entrega del producto final.

Cabe indicar que el presenta trabajo de titulación se integra de dos partes fundamentales que son el de investigar e implantar una arquitectura tolerante a fallos, esto será considerado como historias técnicas y desarrollar módulos en los sistemas mencionados anteriormente que serán las historias de usuario consideradas en la metodología.
La metodología antes mencionada contempla tres fases: planificación, desarrollo, finalización.

2.1. Actividades de la metodología

2.1.1. Tipo de Investigación

En el presente trabajo de titulación se utilizará la investigación aplicada, dado que su propósito encontrar información útil y aplicable que nos permitan lograr un objetivo en este caso implementar mecanismos que nos permitan brindar tolerancia a fallos. La investigación aplicada permite generar conocimiento a partir de una base existente el mismo que ayudará a mejoras a mediano plazo en la sociedad o en el sector productivo.

2.1.2. Métodos de Investigación

- **Método de Análisis – Síntesis:**

En el análisis se estudia porque algunos sistemas colapsan al recibir varias peticiones en un lapso de tiempo para así realizar una elección de las herramientas que se vayan a utilizar para mejorar estos defectos.

La síntesis se usa con el fin de recopilar información que nos permita desarrollar el marco teórico con el fin de explicar todos los instrumentos que involucran la implementación de mecanismos que brinden tolerancia a fallos.

- **Método Inductivo – Deductivo**

Con el método inductivo Se determina como integrar los mecanismos a ser implementados en los servidores, para generar un estándar que sirva a otros sistemas de similares características. Analizando el funcionamiento de los sistemas del DDA de lo particular a lo general, lo que nos permite llegar a conclusiones sobre los mismos.
En el método deductivo inicia de lo general a lo particular, es decir de las conclusiones obtenidas con el método inductivo, obtendremos las recomendaciones adecuadas.

2.1.3. Técnicas de Investigación

Para la recopilación de la información necesaria para el desarrollo de este trabajo de titulación, hemos determinado utilizar las siguientes técnicas:

- Revisión bibliográfica
- Entrevista al director de DTIC
- Entrevista al director de la dirección de desarrollo académico

2.1.4. Parámetros e indicadores

El presente trabajo de titulación va a ser evaluado en base a la norma ISO/IEC 25010 de acuerdo a la característica de fiabilidad la misma que contempla evaluar los componentes de la arquitectura bajo diferentes condiciones, las subcarterísticas que se van a tomar en cuenta son las siguientes:

- Madurez
- Disponibilidad
- Tolerancia a fallas
- Capacidad de recuperación

2.1.5. Escenarios de prueba

Se ha especificado escenarios de prueba para comprobar el funcionamiento de la arquitectura tolerante a fallas implantada la misma que está dividida en dos partes.

Escenario 1 (esperado)

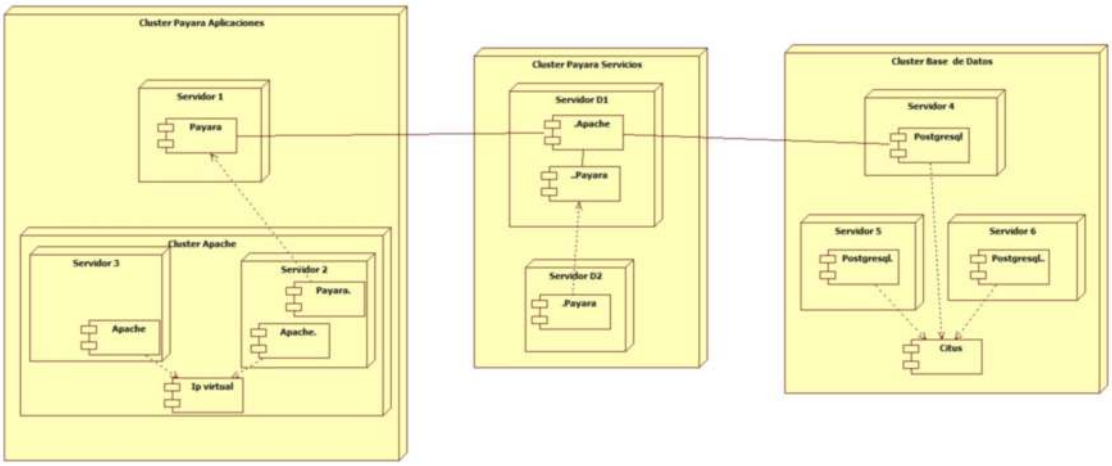


Figura 1-2: Diagrama de despliegue de la arquitectura tolerante a fallos con dos instancias
Realizado por: Zaruma Jorge. 2019

Escenario 2 (análisis)

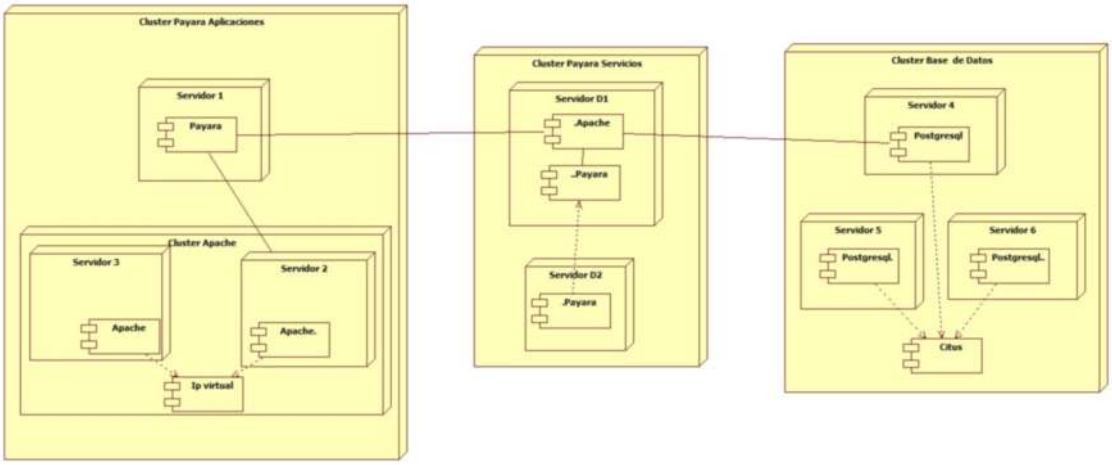


Figura 2-2: Diagrama de despliegue de la arquitectura tolerante a fallos con una instancia
Realizado por: Zaruma Jorge. 2019

Representa el número máximo de usuarios que la arquitectura es capaz de soportar sin inconvenientes, para esto se ha realizara un caso de prueba:

2.1.5.1. Casos de prueba

Se realizará dos casos de prueba con cada uno de los escenarios propuestos, a continuación, se detalla las características principales de cada uno de ellos:

Caso de prueba 1: Máximo de usuarios esperados

- 500 usuarios simultáneos
- Respuesta de 1 segundo por cada usuario
- Sin ciclos de repetición.

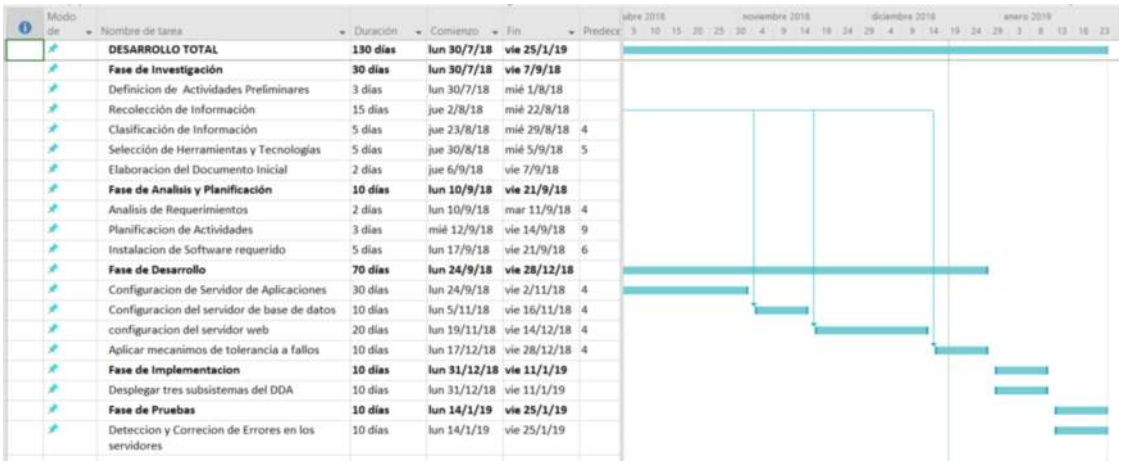
Caso de prueba 2: Número de usuarios para pruebas de estrés

- 2000 usuarios simultáneos
- Respuesta de 1 segundo por cada usuario
- Sin ciclos de repetición

2.2. Fase de planificación

En esta fase se realizará una planificación mediante un diagrama Gantt de las actividades que se pretende realizar con sus fechas estimadas, aquí se tomará en cuenta las investigaciones que se realicen en cuanto a la arquitectura y su respectiva implementación, selección de herramientas adecuadas y el desarrollo de los módulos propuestos.

El tiempo estimado se visualiza en la **Figura 3-2** que es de 130 días.



Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predece
	DESARROLLO TOTAL	**130 días**	**lun 30/7/18**	**vie 25/1/19**	
	Fase de Investigación	**30 días**	**lun 30/7/18**	**vie 7/9/18**	
	Definicion de Actividades Preliminares	3 días	lun 30/7/18	mié 1/8/18	
	Recolección de Información	15 días	jue 2/8/18	mié 22/8/18	
	Clasificación de Información	5 días	jue 23/8/18	mié 29/8/18	4
	Selección de Herramientas y Tecnologías	5 días	jue 30/8/18	mié 5/9/18	5
	Elaboracion del Documento Inicial	2 días	jue 6/9/18	vie 7/9/18	
	Fase de Analisis y Planificación	**10 días**	**lun 10/9/18**	**vie 21/9/18**	
	Analisis de Requerimientos	2 días	lun 10/9/18	mar 11/9/18	4
	Planificacion de Actividades	3 días	mié 12/9/18	vie 14/9/18	9
	Instalacion de Software requerido	5 días	lun 17/9/18	vie 21/9/18	6
	Fase de Desarrollo	**70 días**	**lun 24/9/18**	**vie 28/12/18**	
	Configuracion de Servidor de Aplicaciones	30 días	lun 24/9/18	vie 2/11/18	4
	Configuracion del servidor de base de datos	10 días	lun 5/11/18	vie 16/11/18	4
	configuracion del servidor web	20 días	lun 19/11/18	vie 14/12/18	4
	Aplicar mecanimos de tolerancia a fallos	10 días	lun 17/12/18	vie 28/12/18	4
	Fase de Implementacion	**10 días**	**lun 31/12/18**	**vie 11/1/19**	
	Desplegar tres subsistemas del DDA	10 días	lun 31/12/18	vie 11/1/19	
	Fase de Pruebas	**10 días**	**lun 14/1/19**	**vie 25/1/19**	
	Deteccion y Correcion de Errores en los servidores	10 días	lun 14/1/19	vie 25/1/19	

Figura 3-2: Diagrama Gantt
Realizado por: Zaruma Jorge. 2019

2.2.1. Personas y roles del proyecto

Existen 3 personas involucradas en el equipo SCRUM para el desarrollo del trabajo de titulación, en la siguiente en la **Tabla 1-2** se detalla su información y el rol que desempeña en el equipo.

Tabla 1-2: Personas y Roles

Persona	Contacto	Rol
Ing. Milton Jaramillo	milton.jaramillo@espoch.edu.ec	Product Owner
Ing. Jorge Ariel Menéndez Verdecia	jorge.menendez@espoch.edu.ec	Scrum Master

Jorge Oswaldo Zaruma Tualombo	jorge.zaruma@espoch.edu.ec	Team Development

Realizado por: Zaruma Jorge. 2019

2.2.2. Tipos y roles de usuario

Al desarrollar los módulos para los sistemas de gestión académica de la DDA se identificó los siguientes roles los mismos que están descritos en la **Tabla 2-2**.

Tabla 2-2: Tipos y función de usuarios

Tipo de Usuario	Función
Administrador (DDA)	Perona que se encargara de administrar los servidores y las aplicaciones de la DDA.

Realizado por: Zaruma Jorge. 2019

2.2.3. Pila del producto

La pila del producto son la lista de funcionalidades que el cliente determina en este caso también se va a tener en cuenta los requerimientos que como desarrollador se proponga para la arquitectura tolerante a fallos, además se debe tener en cuenta que la tarjeta de trabajo de una pila de producto va a ser las historias de usuario e historias técnicas, para la estimación de cada una de las funcionalidades se va a utilizar el método T-shirt que se detalla en la **Tabla 3-2.**

Tabla 3-2: Método T-shirt

Talla	Puntos estimados	Iteración
XS	24	1/3
S	40	1/2
M	**80**	**1**
L	120	2
XL	240	3
XLL	>240	>3

Realizado por: Zaruma Jorge. 2019

Un punto estimado corresponde a una hora de trabajo real por lo que durante este proceso se va a trabajar 8 horas diarias, y se determinó que cada iteración es de dos semanas por lo que en cada iteración se deben cumplir 80 horas de trabajo.

La lista de historias técnicas e historias de usuario planteadas para el presente trabajo de titulación se listan en la **Tabla 4-2.**

Tabla 4-2: Product backlog

ID	Descripción	Puntos Estimados
HT_01	Como desarrollador se desea investigar sobre técnicas de tolerancia a fallos	80
HT_02	Como desarrollador se desea investigar sobre herramientas para la técnicas de tolerancia a fallos	40
HT_03	Como desarrollador se diseñar la arquitectura tolerante a fallos.	40
HT_04	Como desarrollador se desea instalar el servidor de aplicaciones	40
HT_05	Como desarrollador se desear configurar un clúster en el servidor de aplicaciones	40
HT_06	Como desarrollador se requiere instalar un servidor de base de datos	40
HT_07	Como desarrollador se requiere configurar un clúster de base de datos	40
HT_08	Como desarrollador se desea instalar un servidor web	24
HT_09	Como desarrollador se desea configurar alta disponibilidad en el servidor web	40
HT_10	Como desarrollador se desea crear un balanceador de carga en el servidor web	16
HU_01	Como administrador se requiere listar los roles del Sistema de Planificación	16
HU_02	Como administrador se requiere listar los usuarios del Sistema de Planificación	16
HU_03	Como administrador se requiere agregar Usuarios a Rol del Sistema de Planificación	64
HU_04	Como administrador se requiere agregar roles a usuarios del Sistema de Planificación	40
HU_05	Como administrador se requiere listar los roles del Sistema de Sílabos	16
HU_06	Como administrador se requiere listar los usuarios del Sistema de Sílabos	16

HU_07	Como administrador se requiere agregar usuarios a rol del Sistema de Sílabos	64
HU_08	Como administrador se requiere agregar roles a usuarios del Sistema de Sílabos	40
HU_09	Como administrador se requiere listar los roles del Sistema de Programas Analíticos	16
HU_10	Como administrador se requiere listar los usuarios del Sistema de Programas Analíticos	16
HU_11	Como administrador se requiere agregar usuarios a rol del Sistema de Programas Analíticos	64
HU_12	Como administrador se requiere agregar roles a usuarios del Sistema de Programas Analíticos	40
HU_13	Como desarrollador se desea desplegar las aplicaciones en los servidores	24
HU_14	Como desarrollador se desea realizar la detección y corrección de errores en los servidores configurados	24
HU_15	Como administrador se requiere buscar un usuario por su cedula en el sistema de sílabos	24
HU_16	Como administrador se requiere buscar un usuario por su cedula en el sistema de programas analíticos	24
HU_17	Como administrador se requiere buscar un usuario por su cedula en el sistema de planificación	24
HT_11	Como desarrollador se desea analizar el funcionamiento de la arquitectura tolerante a fallos.	32
HT_12	Como desarrollador se desea determinar las conclusiones y recomendaciones.	16
HT_13	Como desarrollador se desea realizar el manual de instalación y configuración de los servidores	32
HT_14	Como desarrollador se desea realizar el documento del trabajo de titulación	32

Realizado por: Zaruma Jorge. 2019

En total se han identificado 17 historias de usuario y 14 historias técnicas que serán distribuidas en los sprints posteriores para la ejecución del trabajo.

2.2.4. Análisis Económico

Como es un trabajo destinado para la Dirección de Desarrollo Académico de la ESPOCH los distintos tipos de servidores configurados corresponden a los tipos de software libre, así como los servidores físicos son proporcionados por el departamento mencionado anteriormente. Por lo tanto, en la Tabla 5-2 se listan algunos gastos necesarios para complementar el desarrollo del trabajo.

Tabla 5-2: Presupuesto del Proyecto

Descripción	**Cantidad**	**Valor Unitario**	**Total**
Hardware			
Laptop Dell Inspiron 15 5567 Intel(R) Core(TM) i7-7500U CPU @2.90 GHZ	1	1200	1200
Otros			
Arriendo (Internet y Servicios Básicos)	5 meses	70	350
Total			

Realizado por: Zaruma Jorge. 2019

2.3. Fase de diseño

En esta fase se detalla las actividades que se han realizado para elaborar el trabajo de titulación, así como la definición de requerimientos, el diseño de la arquitectura tolerante a fallos, rediseño de las bases de datos y diagramas uml necesarios para comprender mejor el enfoque del proyecto.

2.3.1. Diagrama de casos de uso

Este tipo de diagramas permite demostrar las funcionalidades de los sistemas en los que se van agregar módulos, esto permite tener una visión de cómo va a interactuar el usuario con las funcionalidades.

En la **Figura 4-2** se visualiza las funcionalidades que puede ejecutar el administrador del DDA una vez concluido con la presente arquitectura tolerante a fallos ya que podrá realizar la misma configuración en otros servidores con el fin de verificar el correcto funcionamiento del mismo

para que posteriormente pueda desplegar las aplicaciones que considere pertinente. Así como las bases de datos correspondientes a las aplicaciones.

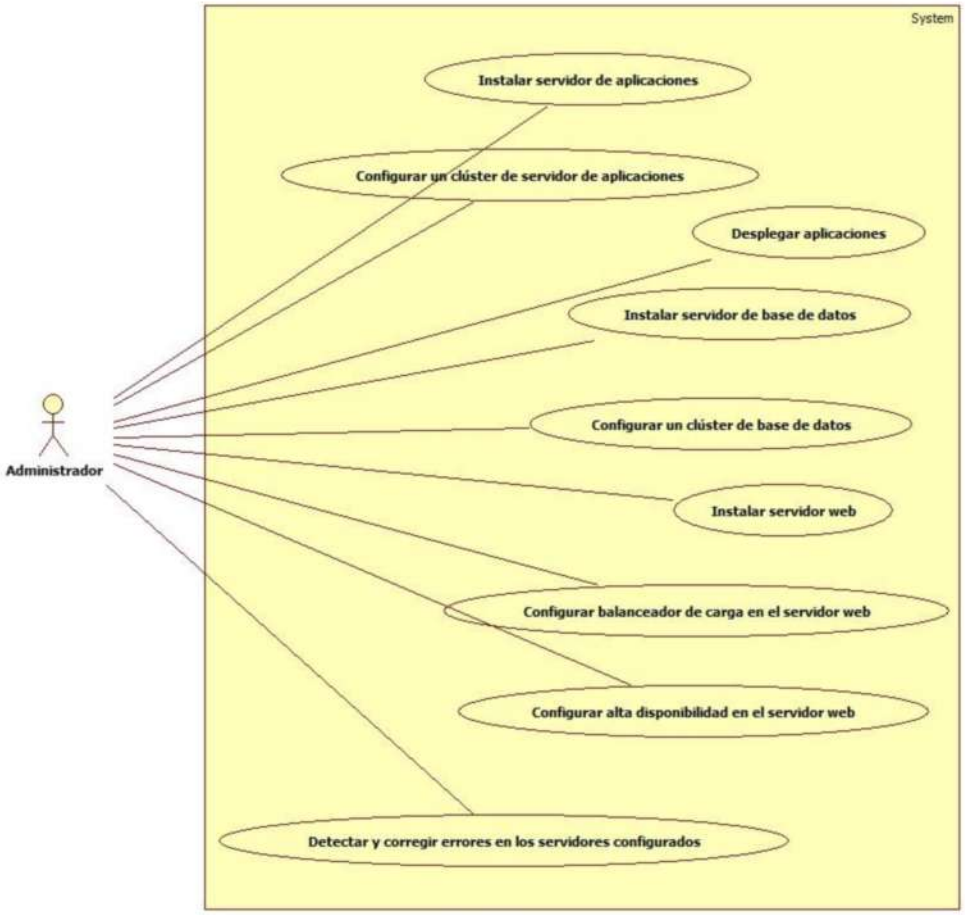


Figura 4-2: Diagrama de caso de uso para el administrador de la DDA
Realizado por: Zaruma Jorge; 2019

El diagrama de casos de uso de los sistemas de gestión académica de la DDA se encuentra especificados en el **Anexo A.**

En la **Tabla 6-2** se describe el actor, descripción, las precondiciones, la secuencia normal, las post condiciones y las excepciones del caso de uso instalar un servidor de aplicaciones.

Tabla 6-2: Documentación de caso de uso

Caso de Uso	**Instalar servidor de aplicaciones**	
Actores:	Administrador	
Precondición	El administrador debe poseer las credenciales que le faculte ingresar a la consola de administración remota del servidor físico.	
Secuencia normal	**Paso**	**Acción**

	1	Ingresar a la herramienta de conexión remota ssh en este caso putty.
	2	Ingresar sus credenciales
	3	Descargar el zip de payara como servidor de aplicaciones con la herramienta wget
	4	Descomprimir el zip
	5	Agregar un usuario para administrar payara
	6	Habilitar payara para que se ejecuta una vez iniciado el servidor
	7	Habilitar el puerto 4848 para administración
	8	Habilitar la forma de autenticación remota agregando un usuario y contraseña de payara para poder administrarlo desde la consola grafica
	9	Inicializar el servidor de aplicaciones
Post condición	Al ingresar las credenciales de administración remota se pueda acceder a la consola de administración gráfica.	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

Realizado por: Zaruma Jorge. 2019

Las tablas de documentación de caso de uso están disponibles en el **Anexo A.2**

2.3.2. Diagrama de Secuencia

Este tipo de diagrama muestra la interacción entre los principales objetos del sistema, se realiza por cada caso de uso. En la **Figura 5-2** se visualiza el proceso que el administrador sigue para instalar el servidor de aplicaciones en el servidor físico cabe señalar que la conexión al mismo se lo realizo a través del cliente ssh putty pero, para una mejor apreciación del procedo se denomina servidor (DAS).

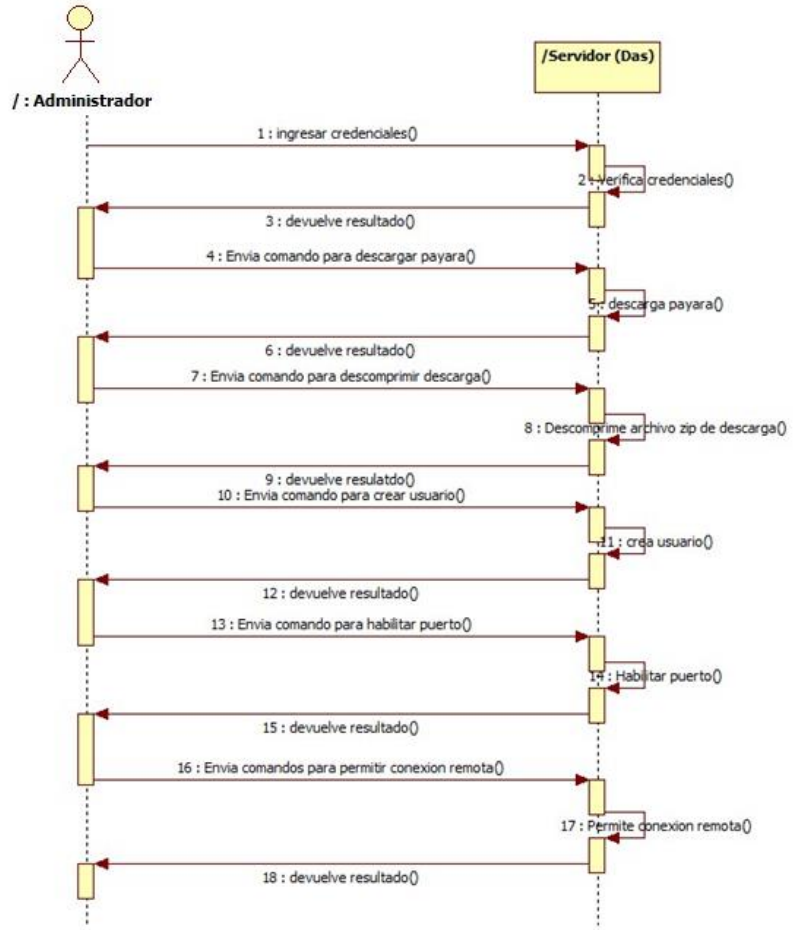


Figura 5-2: Diagrama de secuencia para instalar el servidor de aplicaciones
Realizado por: Zaruma Jorge. 2019

La totalidad de los diagramas de secuencia se puede encontrar en el **Anexo B,** además **se** adjuntan los diagramas de colaboración se pueden encontrar en el **Anexo C**

2.3.3. Diagrama de clases

En la siguiente figura se puede observar el diagrama de clases del paquete comunes el mismo que será utilizado en los sistemas de gestión académica de la DDA, se representa con sus respetivos atributos, la relación que existe entre los mismos y los métodos disponibles. Las clases definidas en la **Figura 7-2** permitió agregar los módulos de usuarios y roles, que son importantes para desplegar las aplicaciones en los servidores y realizar las pruebas pertinentes.

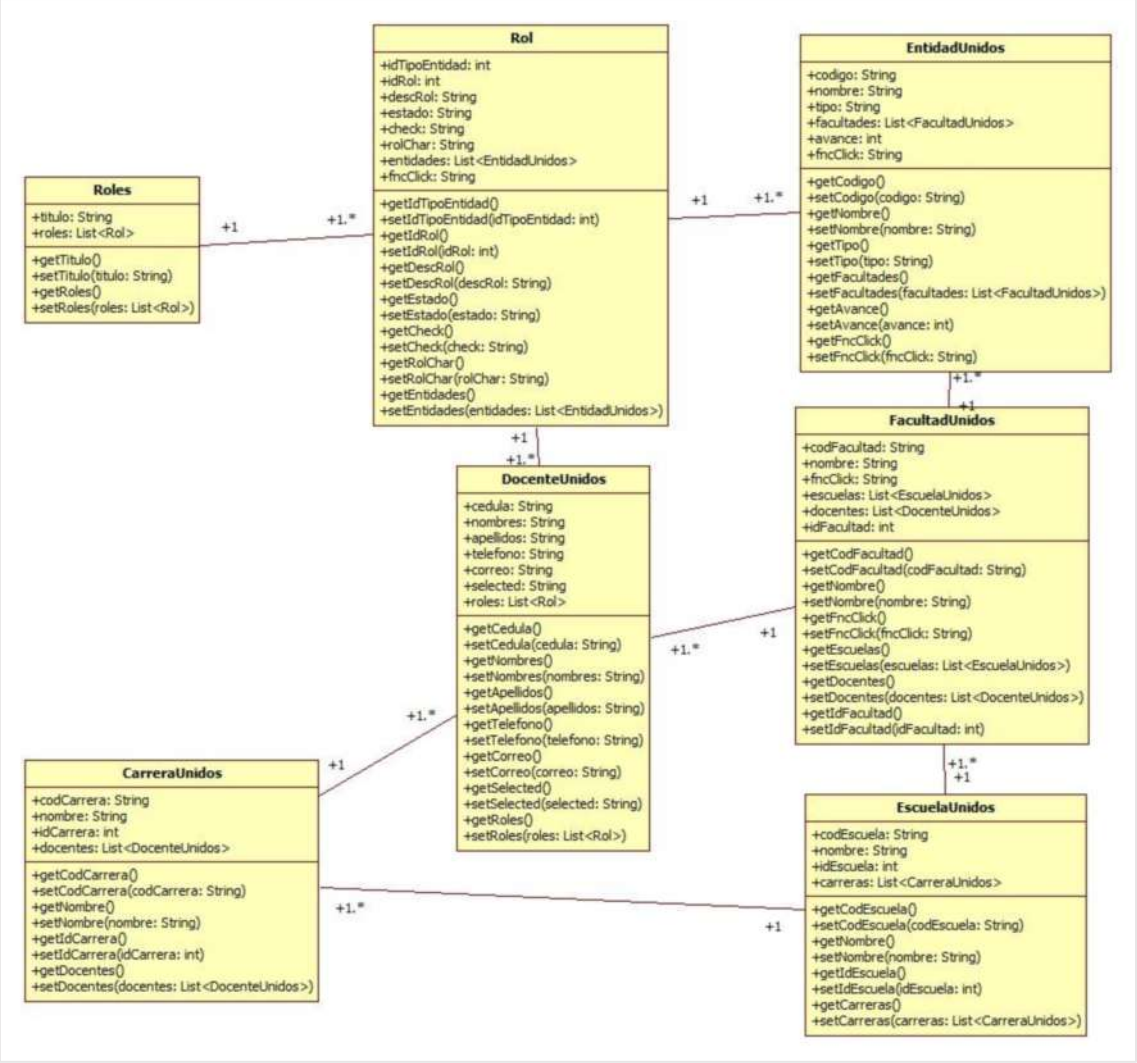


Figura 6-2: Diagrama de clases
Realizado por: Zaruma Jorge. 2019

2.3.4. Diagrama de componentes

Este tipo de diagramas nos permite tener una visión clara de los componentes software que contiene cada sistema y a su vez muestra las dependencias que existen entre los mismo. En la **Figura 8-2** Se visualiza el diagrama de componentes del sistema de sílabos el mismo que interactúa con el diagrama de componentes del Programa Analítico y Planificación, pero para una mejor apreciación se lo ha dividido por cada sistema.

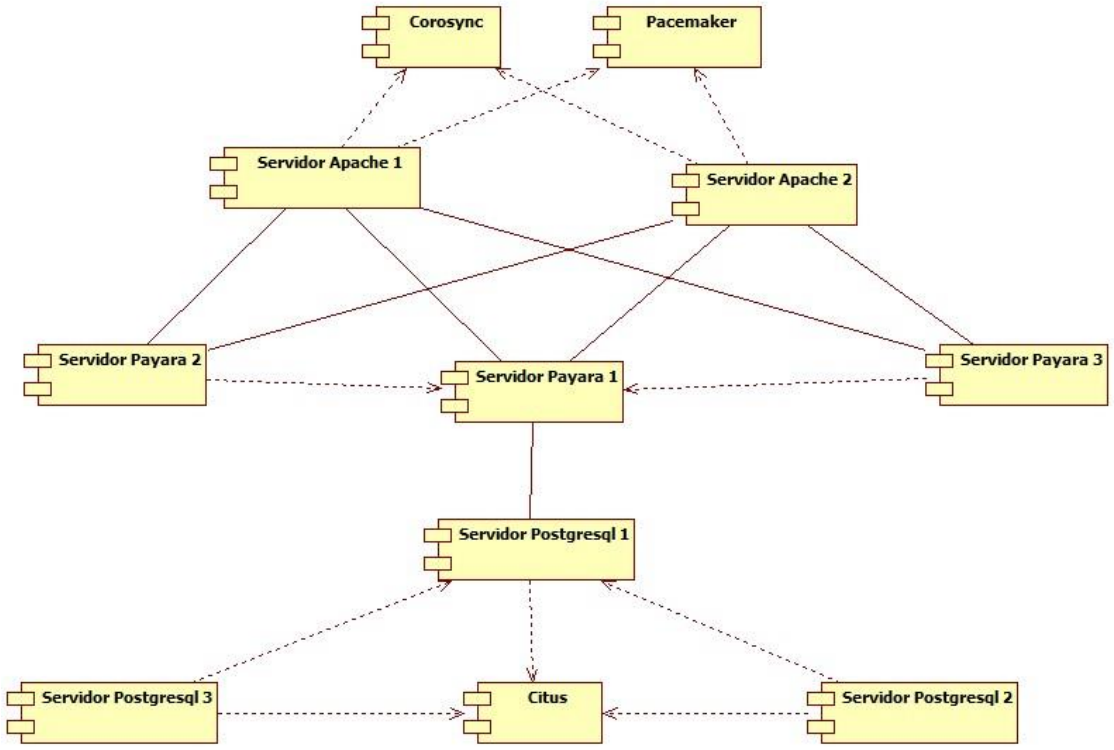


Figura 7-2: Diagrama de componentes de los servidores
Realizado por: Zaruma Jorge. 2019

El diagrama de componentes de los sistemas de gestión de gestión académica de la DDA se encuentra en el **Anexo D**

2.3.5. Recursos Necesarios

Para la implementación de los módulos en los sistemas como la configuración de los servidores se utilizaron recursos hardware y software los cuales se detallan a continuación.

Recurso Hardware

Este tipo de recurso fue básicamente usado en el desarrollo de los módulos de gestión de roles y usuarios de los sistemas, el mismo que se detalla en la **Tabla 7-2**

Tabla 7-2: Recursos Hardware

Descripción	Características	Estado
Laptop	**Fabricante:** Dell **Modelo:** Inspiron 155567	Disponible

	Procesador: Core(TM) i7-7500U CPU @2.90 GHZ **Memoria RAM:** 16 GB **Sistema Operativo:** Windows 10 de 64 bits.	

Realizado por: Zaruma Jorge. 2019

Recursos Software

Este tipo de recursos fue necesario tanto para la parte de desarrollo de los módulos y para la configuración de los servidores, en la **Tabla 8-2** se listan recursos y la función que tuvieron en el desarrollo del trabajo.

Tabla 8-2: Recursos Software

Descripción	Tipo	Función
NetBeans IDE 8.2	Software de desarrollo	Desarrollo de los módulos de gestión
Microsoft Office	Ofimática	Gestión de documentos
Payara	Software	Servidor de aplicaciones
Apache	Software	Servidor web
Pacemaker	Software	Configuración de alta disponibilidad de apache
Corosync	Software	Configuración de alta disponibilidad de apache
Bootstrap	Framework	Diseño de interfaces
PostgreSQL 10	Gestor de base de datos	Servidor de base de datos
Centos 7	Sistema operativo	Alojara las distintas herramienta utilizadas para el presente trabajo
pgAdmin IV	Software	Herramienta para el manejo de base de datos

Realizado por: Zaruma Jorge. 2019

2.3.6. Diseño de interfaz de usuario

Mediante la interfaz el usuario puede comunicar con la aplicación por tal motivo es de vital importancia proveer una interfaz amigable con un diseño intuitivo y así mejorar la experiencia del usuario al utilizar el sistema.

Cabe indicar que las pantallas que se van a visualizar son estándar tanto para el silabo, programa analítico y planificación.

- **Pantalla para la gestión de roles**

En la **Figura 9-2** se puede observar que se listan los roles del sistema dependiendo del usuario que se encuentra administrando dicho sistema, existen dos funciones principales que es el de agregar usuarios al rol y permite también agregar opciones al rol.

Figura 8-2: Pantalla para la gestión de roles
Realizado por: Zaruma Jorge. 2019

- **Asignación de roles a usuarios:** En la **Figura 10-2** se puede visualizar que para agregar un rol al usuario se toma en cuenta el nivel del rol en este caso es un administrador de carrera por lo que obligatoriamente debería elegir en qué carrera va a desempeñar ese rol.

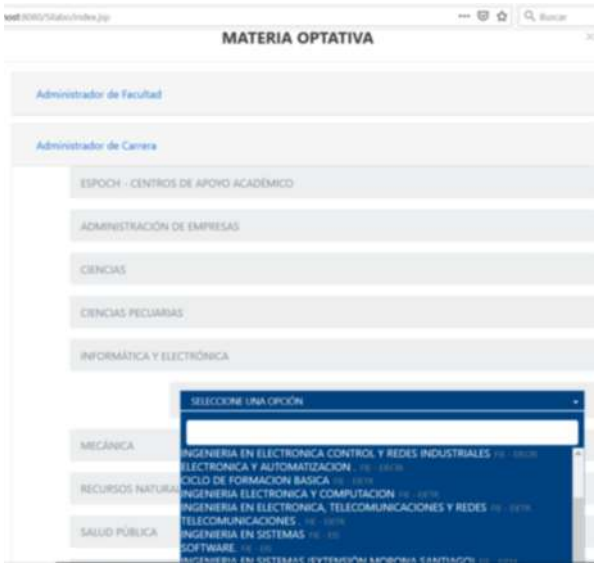


Figura 9-2: Pantalla para la gestión de roles
Realizado por: Zaruma Jorge. 2019

- **Gestión de usuarios:** En la **Figura 11-2** al igual que en la de roles mantiene un estándar de como visualizar cada registro de usuario a su vez cuenta también con la opción de editar información, eliminar el usuario agregado, asignar roles al usuario y existe la función de agregar un nuevo usuario.

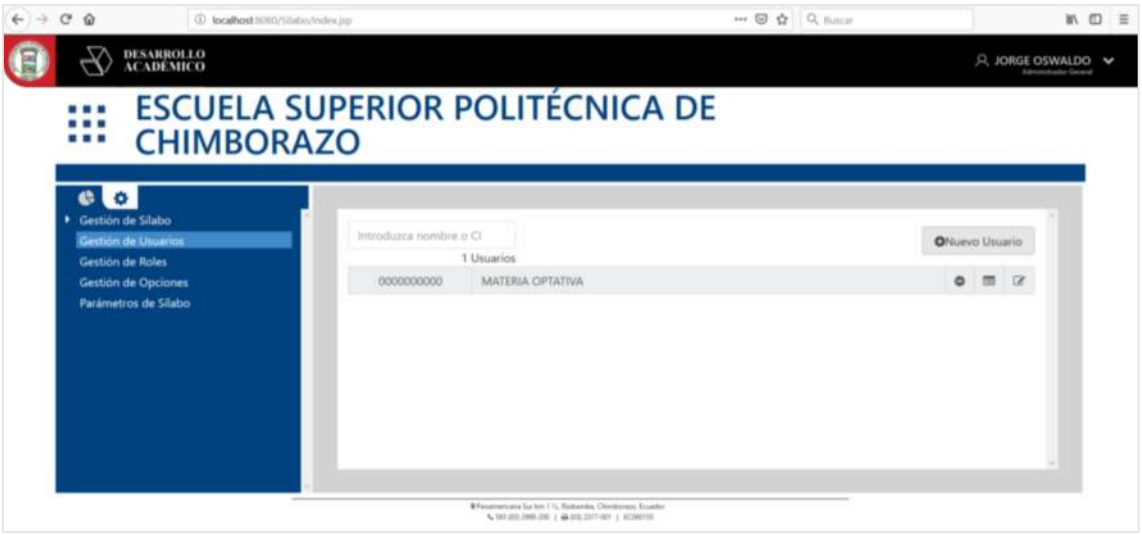


Figura 10-2: Pantalla para la gestión de usuarios
Realizado por: Zaruma Jorge. 2019

- **Buscar usuario:** Como ya se menciona en la **Figura 12-2** el poder agregar un usuario se lo realiza en esta pantalla para lo cual tiene que buscarlo por cedula y si existe, nos muestra la información con la opción de agregar si es un usuario externo.

Figura 11-2: Pantalla para agregar un nuevo usuario
Realizado por: Zaruma Jorge. 2019

- **Asignación de usuarios a roles:** Por ultimo en esta pantalla se mantiene el estándar de la **Figura 13-2** con la diferencia de que aquí se selecciona un usuario para el rol y la entidad seleccionada.

Figura 12-2: Pantalla para agregar usuarios a un rol
Realizado por: Zaruma Jorge. 2019

2.4. Fase de desarrollo e implementación

2.4.1. Sprint backlog

El sprint backlog es el conjunto de historias de usuario e historias técnicas que se van a realizar para el desarrollo del siguiente trabajo. El mismo consta de 13 sprint, cada sprint tiene la duración de dos semanas, es decir de 80 horas que dan 80 puntos estimados. El plan de entrega de cada sprint se detalla en la **Tabla 9-2**.

Tabla 9-2: Sprint backlog

N. Sprint	ID	Requerimiento	Puntos estimados	Puntos totales
1	HT_01	Como desarrollador se desea investigar sobre técnicas de tolerancia a fallos	80	80
2	HT_02	Como desarrollador se desea investigar sobre herramientas para las técnicas de tolerancia a fallos	40	80
	HT_03	Como desarrollador se diseñar la arquitectura tolerante a fallos.	40	
3	HT_04	Como administrador se desea instalar el servidor de aplicaciones	40	80

	HT_05	Como administrador se desear configurar un clúster en el servidor de aplicaciones	40	
4	HT_06	Como administrador se requiere instalar un servidor de base de datos	40	80
	HT_07	Como administrador se requiere configurar un clúster de base de datos	40	
5	HT_08	Como desarrollador se desea instalar un servidor web	24	80
	HT_09	Como administrador se desea configurar alta disponibilidad en el servidor web	40	
	HT_10	Como administrador se desea crear un balanceador de carga en el servidor web	16	
6	HU_05	Como administrador se requiere listar los roles del Sistema de Sílabos	16	80
	HU_07	Como administrador se requiere agregar usuarios a rol del Sistema de Sílabos	64	
7	HU_06	Como administrador se requiere listar los usuarios del Sistema de Sílabos	16	80
	HU_08	Como administrador se requiere agregar roles a usuarios del Sistema de Sílabos	40	
	HU_15	Como administrador se requiere buscar un usuario por su cedula en el sistema de sílabos	24	
8	HU_01	Como administrador se requiere listar los roles del Sistema de Planificación	16	80
	HU_03	Como administrador se requiere agregar usuarios a rol del Sistema de Planificación	64	
9	HU_02	Como administrador se requiere listar los usuarios del Sistema de Planificación	16	80
	HU_04	Como administrador se requiere agregar roles a usuarios del Sistema de Planificación	40	
	HU_17	Como administrador se requiere buscar un usuario por su cedula en el sistema de Planificación	24	
10	HU_09	Como administrador se requiere listar los roles del Sistema de Programas Analíticos	16	80

	HU_11	Como administrador se requiere agregar usuarios a rol del Sistema de Programas Analíticos	64	
11	HU_10	Como administrador se requiere listar los usuarios del Sistema de Programas Analíticos	16	80
	HU_12	Como administrador se requiere agregar roles a usuarios del Sistema de Programas Analíticos	40	
	HU_16	Como administrador se requiere buscar un usuario por su cedula en el sistema de Programas Analíticos	24	
12	HU_13	Como desarrollador se desea desplegar las aplicaciones en los servidores	24	80
	HU_14	Como desarrollador se desea realizar la detección y corrección de errores en los servidores configurados	24	
	HT_11	Como desarrollador se desea analizar el funcionamiento de la arquitectura tolerante a fallos.	32	
13	HT_12	Como desarrollador se desea determinar las conclusiones y recomendaciones.	16	80
	HT_13	Como desarrollador se desea realizar el manual de instalación y configuración de los servidores	32	
	HT_14	Como desarrollador se desea realizar el documento del trabajo de titulación	32	

Realizado por: Zaruma Jorge. 2019

Sprint 1: Durante este periodo se realizó una investigación principalmente sobre los principios de tolerancia a fallos, trabajos realizados acerca del tema, gracias a esto se logró identificar tres técnicas que combinadas permitían obtener una infraestructura a nivel de servidores óptima para la producción. Entre las técnicas encontradas están: escalabilidad, redundancia y alta disponibilidad.

Tabla 10-2: Detalle pila sprint 1

Sprint 1			
Inicio: 30/07/2018	**Fin:** 10/08/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HT_01	Como desarrollador se desea investigar sobre técnicas de tolerancia a fallos	80	Análisis	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 2: Habiendo obtenido las técnicas que se van a implementar en la arquitectura fue necesario también investigar sobre que herramientas de software bajo el Sistema operativo linux permitan implementar estas técnicas, entre las herramientas que se van a utilizar destacan: payara, apache, pacemaker, Corosync y citus. Con las herramientas obtenidas y la disponibilidad de 8 servidores se realizó el diseño de la arquitectura tolerante a fallos teniendo en cuenta la distribución de estos recursos.

Tabla 11-2: Detalle pila sprint 2

Sprint 2			
Inicio: 13/08/2018	**Fin:** 24/08/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HT_02	Como desarrollador se desea investigar sobre herramientas para la técnicas de tolerancia a fallos	40	Análisis	Jorge Zaruma
HT_03	Como desarrollador se diseñar la arquitectura tolerante a fallos.	40	Diseño	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 3: en este sprint se empezó con la configuración del servidor de aplicaciones como un clúster, el mismo que está conformado por 3 servidores a los que se denominó: nodo 1 y nodo 2 como nodos secundarios, Das como el nodo principal por medio del cual se realiza la monitorización de los nodos que conforman el clúster. Esto permitirá obtener escalabilidad y redundancia de la información.

Tabla 12-2: Detalle pila sprint 3

Sprint 3			
Inicio: 27/08/2018	**Fin:** 07/09/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HT_04	Como administrador se desea instalar el servidor de aplicaciones	40	Análisis	Jorge Zaruma
HT_05	Como administrador se desear configurar un clúster en el servidor de aplicaciones	40	Configuración	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 4: También uno de los puntos más vulnerables de cualquier Sistema es el gestor de base de datos, ya que aquí se van a ejecutar varias transacciones con datos importantes. Por tal motivo en este sprint se realizó la configuración de clúster mediante la extensión de postgresql que es citus el mimo que permite la manipulación de los nodos para escalar horizontalmente, cabe mencionar que citus permite obtener una base de datos distribuida, esto mejora las consultas, pero no permite redundancia de información.

Tabla 13-2: Detalle pila sprint 4

Sprint 4			
Inicio: 10/09/2018	**Fin:** 21/09/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HT_06	Como administrador se requiere instalar un servidor de base de datos	40	Análisis	Jorge Zaruma
HT_07	Como administrador se requiere configurar un clúster de base de datos	40	Configuración	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 5: Al crear un clúster del servidor de aplicaciones es necesario tener un balanceador de carga que permita redirigir las peticiones de los usuarios a nodos disponibles en el clúster, evitando el balanceo de peticiones en nodos que estén fuera de servicio esto se logró gracias a la

facilidad de configurar dicho balanceador en apache. Además, fue importante también configurar alta disponibilidad en el servidor apache para evitar que los sistemas queden fuera de servicio.

Tabla 14-2: Detalle pila sprint 5

Sprint 5			
Inicio: 24/09/2018	**Fin:** 05/10/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HT_08	Como administrador se desea instalar un servidor web	24	Análisis	Jorge Zaruma
HT_09	Como administrador se desea configurar alta disponibilidad en el servidor web	40	Configuración	Jorge Zaruma
HT_10	Como administrador se desea crear un balanceador de carga en el servidor web	16	Configuración	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 6: Luego de haber realizado todas las configuraciones en los servidores se procedió a desarrollar el módulo de gestión de roles del sistema de sílabos, esto permite que el Sistema pueda operar con usuarios de varios roles los mismos que están en el proceso para la creación del silabo.

Tabla 15-2: Detalle pila sprint 6

Sprint 6			
Inicio: 08/10/2018	**Fin:** 19/10/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HU_05	Como administrador se requiere listar los roles del Sistema de Sílabos	16	Codificación	Jorge Zaruma
HU_07	Como administrador se requiere agregar usuarios a rol del Sistema de Sílabos	64	Codificación	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 7: Durante este sprint se agregó al Sistema de sílabos el módulo de gestión de usuarios, esto permite que se puedan agregar un usuario externo, siempre y cuando se encuentre registrado en el Sistema de hoja de vida y tener credenciales institucionales para que puedan ingresar en el Sistema a trabajar con el rol que también se le puede agregar.

Tabla 16-2: Detalle pila sprint 7

Sprint 7			
Inicio: 22/10/2018	**Fin:** 01/11/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HU_06	Como administrador se requiere listar los usuarios del Sistema de Sílabos	16	Codificación	Jorge Zaruma
HU_08	Como administrador se requiere agregar roles a usuarios del Sistema de Sílabos	40	Codificación	Jorge Zaruma
HU_15	Como administrador se requiere buscar un usuario por su cedula en el sistema de sílabos	24	Codificación	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 8: Al igual que lo realizado en el Sistema de sílabos durante este sprint se agregó el módulo de gestión de roles al Sistema de planificación, cabe indicar que el tiempo de desarrollo es el mismo que en el silabo a pesar de tener que migrar la codificación realizada al Sistema actual, pero durante el proceso aparecieron errores los mismos que ocasionaron que no se pueda desarrollar en el menor tiempo posible.

Tabla 17-2: Detalle pila sprint 8

Sprint 8			
Inicio: 05/11/2018	**Fin:** 16/11/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HU_01	Como administrador se requiere listar los roles del Sistema de Planificación	16	Codificación	Jorge Zaruma

HU_03	Como administrador se requiere agregar usuarios a rol del Sistema de Planificación	64	Codificación	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 9: Como ya se menciona los problemas al migrar la codificación del sistema de sílabos, durante este sprint sucedió lo mismo, pero al final se logró agregar el módulo de gestión de usuarios para el sistema de planificación

Tabla 18-2: Detalle pila sprint 9

Sprint 9				
Inicio: 19/11/2018	**Fin:** 30/11/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80	
Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HU_02	Como administrador se requiere listar los usuarios del Sistema de Planificación	16	Codificación	Jorge Zaruma
HU_04	Como administrador se requiere agregar roles a usuarios del Sistema de Planificación	40	Codificación	Jorge Zaruma
HU_17	Como administrador se requiere buscar un usuario por su cedula en el sistema de Planificación	24	Codificación	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 10: Durante este sprint se agregó el módulo de gestión de roles al Sistema de programas analíticos en el cual se va a poder agregar usuarios a dicho rol. Este proceso es el mismo detallado en los dos sistemas anteriores como es el de planificación y sílabos.

Tabla 19-2: Detalle pila sprint 10

Sprint 10				
Inicio: 03/12/2018	**Fin:** 14/12/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80	
Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**

HU_09	Como administrador se requiere listar los roles del Sistema de Programas Analíticos	16	Codificación	**Jorge Zaruma**
HU_11	Como administrador se requiere agregar usuarios a rol del Sistema de Programas Analíticos	64	Codificación	

Realizado por: Zaruma Jorge. 2019

Sprint 11: También en el sprint 11 se agrega el módulo de gestión de usuarios al Sistema de programas analíticos, permitiendo guardar un usuario externo a la institución en caso de que la dirección de desarrollo académico así lo considere, en este módulo se va a poder agregar roles al usuario para que pueda ingresar al Sistema y desempeñe las funciones que le corresponden.

Tabla 20-2: Detalle pila sprint 11

Sprint 11			
Inicio: 17/12/2018	**Fin:** 28/12/2018	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HU_10	Como administrador se requiere listar los usuarios del Sistema de Programas Analíticos	16	Codificación	Jorge Zaruma
HU_12	Como administrador se requiere agregar roles a usuarios del Sistema de Programas Analíticos	40	Codificación	Jorge Zaruma
HU_16	Como administrador se requiere buscar un usuario por su cedula en el sistema de Programas Analíticos	24	Codificación	Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 12: Luego de haber finalizado con las últimas modificaciones en los sistemas de gestión académica de la dda e procedió a desplegar los sistemas en los Servidores ya configurados previamente, al mismo tiempo que se analizó el desempeño de los servidores durante la utilización de estos sistemas.

Tabla 21-2: Detalle pila sprint 12

Sprint 12			
Inicio: 01/01/2019	**Fin:** 11/01/2019	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo estimado**	**Tipo**	**Responsable**
HU_13	Como desarrollador se desea desplegar las aplicaciones en los servidores	24		Jorge Zaruma
HU_14	Como desarrollador se desea realizar la detección y corrección de errores en los servidores configurados	24		Jorge Zaruma
HT_11	Como desarrollador se desea analizar el funcionamiento de la arquitectura tolerante a fallos.	32		Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

Sprint 13: Luego de haber finalizado todas las tareas de desarrollo y configuración, este sprint está dedicado principalmente a realizar los manuales correspondientes a todas las configuraciones realizadas, determinar las conclusiones y recomendación del trabajo realizado para elaborar el documento del trabajo de titulación.

Tabla 22-2: Detalle pila sprint 13

Sprint 13			
Inicio: 14/01/2019	**Fin:** 25/01/2019	**Esfuerzo estimado:** 80	**Esfuerzo Real:** 80

Pila del Sprint				
Product backlog ID	**Descripción**	**Esfuerzo Estimado**	**Tipo**	**Responsable**
HT_12	Como desarrollador se desea determinar las conclusiones y recomendaciones.	16	Análisis	Jorge Zaruma
HT_13	Como desarrollador se desea realizar el manual de instalación y configuración de los servidores	32		Jorge Zaruma

HT_14	Como desarrollador se desea realizar el documento del trabajo de titulación	32		Jorge Zaruma

Realizado por: Zaruma Jorge. 2019

2.4.2. Historia técnica

Para constatar que los requerimientos se estén desarrollado de acuerdo al plazo establecido, scrum requiere que se documente las historias de usuario e historias técnicas, las mismas que conlleva a la realización de tareas de ingeniería y pruebas de aceptación, mediante esta última permite saber si se cumplieron o no los objetivos del sprint. A continuación, se presenta el esquema como se debe documentar las historias de cada sprint. Como ejemplo se tomó la historia técnica 4 correspondiente al sprint 3.

Sprint 3

Tabla 23-2: Historia técnica 4

HISTORIA TÉCNICA	
Número: HT_04	**Nombre de la historia:** instalar el servidor de aplicaciones
Modificación de historia de usuario:	
Usuario: Administrador	**Sprint Asignada:** 3
Prioridad en el Negocio: Alta	**Puntos Estimados:** 40
Riesgo en el Desarrollo: Bajo	**Puntos Reales:** 40
Descripción: como administrador quiero instalar el servidor de aplicaciones para desplegar los sistemas de gestión académica de la DDA.	
Observaciones: • Se debe tener en cuenta los requerimientos necesarios para instalarlo	

Realizado por: Zaruma Jorge. 2019

Además, cada historia técnica o de usuario está ligada a tareas de ingeniería que son los procesos de fondo que se realizan para cumplir con lo propuesto, en las tareas de ingeniería se establecen fechas de culminación, así como el responsable de la misma, y teniendo en cuenta las pruebas de aceptación que se deben cumplir al finalizar cada tarea. En la **Tabla 24-2** se visualiza una tarea de ingeniería de la HT_04.

Tabla 24-2: Tarea de ingeniería 1 de la HT_04

TAREA DE INGENIERÍA	
Historia Técnica: instalar el servidor de aplicaciones	
Número de Tarea: TI_01	**Nombre de Tarea:** realizar el manual para instalar el servidor de aplicaciones
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 40
Fecha Inicio: 15/10/2018	**Fecha Fin:** 15/10/2018
Programador Responsable: Jorge Zaruma	
Descripción: Se debe tener en cuenta que todos los pasos de configuración se documenten, el documento estará disponible en la dirección: /Manuales/Manuales de Configuración/ Servidor Payara en la sección A del mismo	
Pruebas de Aceptación • Verificar que se pueda acceder al servidor de aplicaciones payara	

Realizado por: Zaruma Jorge. 2019

Por último, en la **Tabla 25-2** se representa la tarjeta para documentar la prueba de aceptación de la tarea de ingeniería 1 correspondiente a la historia técnica 4.

Tabla 25-2: Prueba de aceptación 1 de la tarjeta de ingeniería 1

PRUEBA DE ACEPTACIÓN	
Código: PA_01	**Historia Técnica:** instalar el servidor de aplicaciones
Nombre: TI_01 Verificar que se pueda acceder al servidor de aplicaciones payara	
Responsable: Jorge Zaruma	**Fecha:** 15/10/2018
Descripción: Una vez realizado la instalación se debe acceder remotamente al servidor de aplicaciones payara	
Condiciones de Ejecución: Haber habilitado el puerto 4848 y la autenticación remota.	
Pasos de ejecución: ● Acceder a su navegador de preferencia ● Escribir en la url https://172.17.103.117:4848 ● Ingresar credenciales	
Resultado esperado: Se visualice la página de inicio de payara	
Evaluación de la prueba: Exitosa.	

Realizado por: Zaruma Jorge. 2019

2.4.3. Guía para implementar una arquitectura tolerante a fallos

Con el fin de realizar un instrumento de guía para la replicación de la arquitectura tolerante a fallas implantada en el presente trabajo de titulación se ha identificado las siguientes fases:

2.4.3.1. Fase 1: Estudio Preliminar

Esta fase está orientada a obtener información relevante para determinar los recursos que se van a utilizar para el proceso.

- **Identificar institución**

Nombre de la Institución: ESPOCH específicamente la Dirección de Desarrollo Académico

- **Identificar recursos hardware**

Para el presente trabajo la DDA dispuso de 8 servidores los mismos que poseen las características presentes en la **Tabla 26-2.**

Tabla 26-2: Características principales de servidores

Características	6 Servidores	1 Servidor	1 Servidor
Disco Duro	20 GB	100GB	300 GB
Memoria ram	1.8 GB	3.7 GB	1.8 GB
Swap	2 GB	3.9 GB	2 GB
Sistema operativo	CentOS Linux 7 (Core)	CentOS Linux 7 (Core)	CentOS Linux 7 (Core)

Realizado por: Zaruma Jorge. 2019

- **Identificar características de una arquitectura tolerante a fallas**

Las características que van a ser utilizadas en la arquitectura son:

  - Escalabilidad
  - Redundancia
  - Alta disponibilidad

Para obtener una información referente a las características antes listada debe dirigirse al capítulo 1 del presente documento.

- **Identificar herramientas software adecuadas**

Finalmente, al conocer los requerimientos, recursos hardware disponibles y las técnicas tolerantes a fallas, es importante que las herramientas software a utilizar estén acorde a las necesidades para evitar problemas durante su configuración, las herramientas a utilizar son las siguientes:

- Payara como servidor de aplicaciones
- Apache como servidor web, además tiene la función de balanceador de carga
- Postgresql como servidor de base de datos
- Corosync, pacemaker y pcsd como herramientas de configuración de alta disponibilidad

Para conocer más acerca de estas herramientas debe revisar el capítulo 1 del presente documento.

2.4.3.2. Fase 2: Diseño de Arquitectura

De acuerdo a lo obtenido en la fase 1 se debe proceder a realizar un diagrama de componentes con los recursos hardware y software disponible esto permitirá realizar una distribución adecuada cada uno de los recursos.

Diagrama conceptual

El siguiente diseño permitirá tener una idea más clara de cómo estará distribuido la configuración de los servidores, es necesario aclarar que en la **Figura 14-2** se visualiza un esquema ampliamente detallado, cabe recordar que se cuenta solamente con 8 servidores.

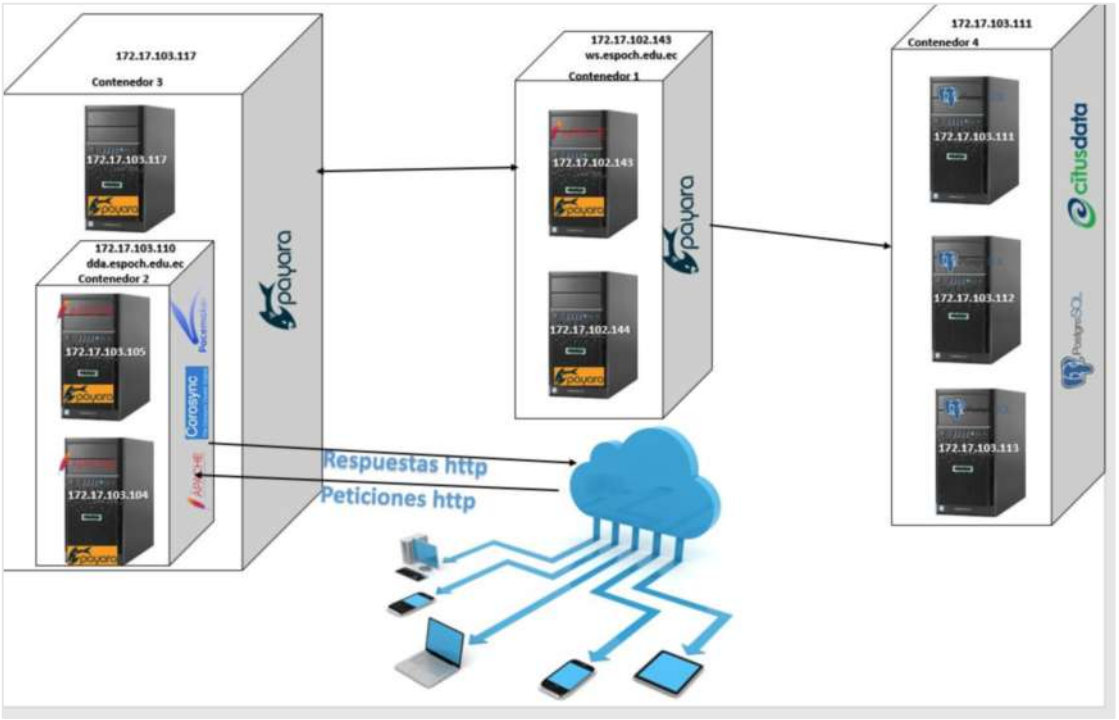


Figura 13-2: Diagrama conceptual de la arquitectura tolerante a fallas a implantar
Realizado por: Zaruma Jorge. 2019

Diagrama de despliegue

Este diagrama representa los componentes hardware que interactúan en tiempo de ejecución de un sistema, además se puede agregar los elementos de software que interviene en cada nodo del diagrama. Como se puede apreciar en la **Figura 15-2** solo representa la configuración de la arquitectura tolerante a fallos a su vez se distribuyen los componentes utilizados en los servidores, así como: payara, apache, Citus, etc.

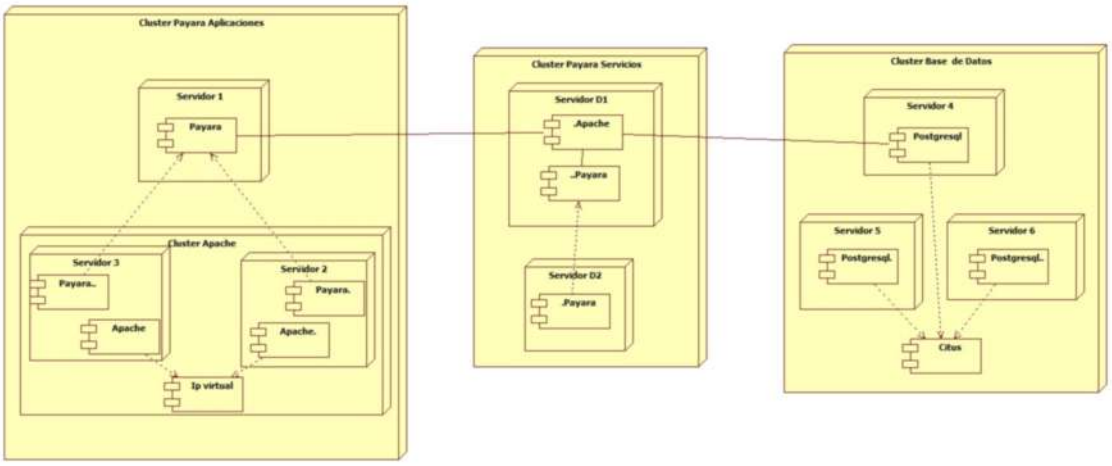


Figura 14-2: Diagrama de componentes de la arquitectura tolerante a fallas a implantar
Realizado por: Zaruma Jorge. 2019

2.4.3.3. Fase 3: Configuración de Componentes

- **Servidor de Aplicaciones**

Este servidor va ser configurado en dos escenarios uno para interfaz como se muestra en la **Figura 16-2** y otro para la lógica de negocios como se visualiza en la **Figura 17-2.**

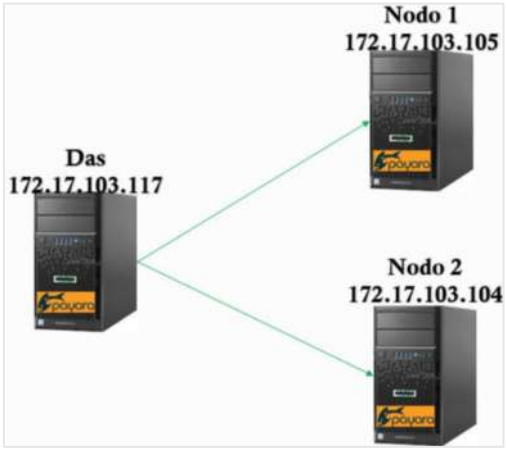


Figura 15-2: Escenarios para servidor de aplicaciones de interfaz de usuario
Realizado por: Zaruma Jorge. 2019

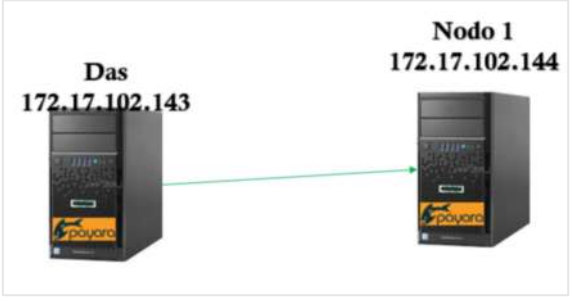


Figura 16-2: Escenario para servidor de aplicaciones de la lógica de negocios.
Realizado por: Zaruma Jorge. 2019

En la **Tabla 27-2** se visualiza los pasos a seguir para la instalación y configuración del clúster de servidor de aplicaciones con payara, además se agrega una descripción de cada uno de los pasos de manera general y por último se especifica en que servidor se debe ejecutar, para una mejor comprensión DAS hace referencia al servidor principal y NODO al servidor o los servidores secundarios que integraran el clúster, el manual de configuración con los pasos que se explican de una manera más detallada se encuentran en el **Anexo G**.

Tabla 27-2: Pasos de Instalación y configuración de un clúster con payara.

	COMANDO O ACCIÓN	**DESCRIPCIÓN**	**SERVIDOR**
1	# yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel	Instalar java	DAS, NODO
2	# vi /etc/hosts	Agregar las direcciones ip con su respectivo nombre de los servidores involucrados en el cluster	
3			DAS
4	# wget https://s3-eu-west-1.amazonaws.com/payara.fish/Payara+Downloads/Payara+4.1.2.173/payara-4.1.2.173.zip	Descargar payara	
5	# adduser payara	Crear usuario	
6	# unzip payara-4.1.2.173.zip	Descomprimir archivo .zip	
7	# chown -R payara:payara payara41	Asignar privilegios de la carpeta descomprimida al usuario payara	
8	# vi /etc/systemd/system/payara.service	Crear archivo de arranque para manejo de comando básicos de payara el contenido se detalla en el **Anexo E**	
9	# systemctl start payara # systemctl disable payara # systemctl restart payara # systemctl stop payara # systemctl status payara	Acciones para control de payara	
10	# su payara #cd /home/payara # vi .bashrc	Edición del archivo y agregar al final: **export PATH=$PATH:/opt/payara41/glassfish/bin** para para agregar la ruta de instalación de payara	
11	# cd /opt/payara41/glassfish/bin # ./asadmin change-admin-password #./asadmin enable-secure-admin # systemctl restart payara	Permitir administración remota de payara, debe asignar adecuadamente la información de configuración.	
12	# firewall-cmd --zone=public --add-port=4848/tcp –permanent #firewall-cmd –reload	Permitir acceso al servidor mediante el puerto 4848	
13	En el navegador escribir: **http://ip-servidor:4848**	Acceder a la administración remota de payara	
14	En la sección de **Nodes** (nodos)	Crear un nodo que haga referencia a los nodos remotos	
15	En la sección de **Configurations (**configuraciones**)**	Crear un nuevo archivo de configuración en base al archivo **default-config** e ingresar a **Hazelcast** para habilitarlo	
16	En la sección Domain (dominio)	Ingresar a la subsección Hazelcast y habilitarlo.	
17	# cd /opt/payara41/glassfish/bin # ./asadmin list-domains # ./asadmin create-domain --adminport 4848 domain1 # ./asadmin start-domain	Verificar si existen dominio para los nodos, caso contrario crearlos.	NODO
18	En la sección **Instances** (instancias)	Crear una instancia por cada nodo creado y hacer referencia al archivo de configuración creado luego iniciarlo.	DAS

Realizado por: Zaruma Jorge. 2019

- **Servidor de Base de datos**

Para la base de datos se creó un clúster mediante citus el cual permite distribución de tablas lo que genera que también que las peticiones puedan procesarse entre los nodos que lo conforman, su esquema se visualiza en la **Figura 16-2**, y se puede observar que el nodo coordinador es quien recibe las peticiones, pero este solo procesara las de escritura como INSERT, DELETE o UPDATE y tanto el nodo 1 como el nodo 2 procesaran peticiones de lectura es decir los SELECT.

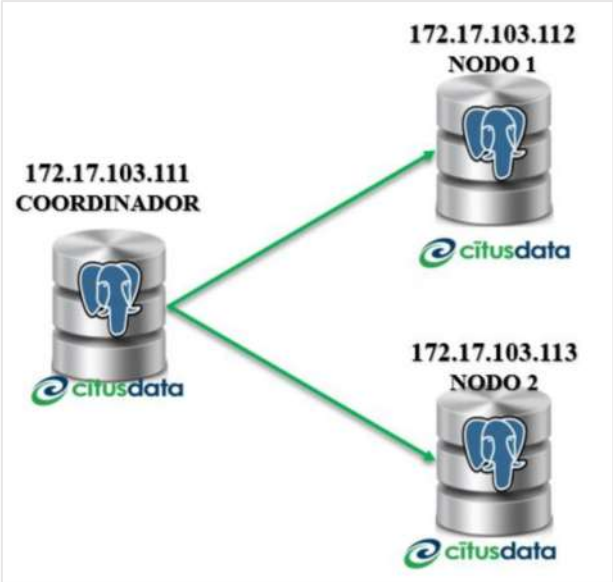


Figura 17-2: Escenario para base de datos
Realizado por: Zaruma Jorge. 2019

En la **Tabla 28-2** se visualiza los pasos a seguir para la instalación y configuración del clúster de servidor de base de datos con postgresql y su extensión citus, además se agrega una descripción de cada uno de los pasos de manera general y por último se especifica en que servidor se debe ejecutar, para una mejor comprensión COORDINADOR hace referencia al servidor principal y NODO al servidor o los servidores secundarios que integraran el clúster, el manual de configuración con los pasos que se explican de una manera más detallada se encuentran en el **Anexo H**.

Tabla 28-2: Instalar postgresql y configurar clúster mediante citus

	COMANDOS	DESCRIPCIÓN	SERVIDOR
1	# curl https://install.citusdata.com/community/rpm.sh | sudo bash	Agregar repositorio de citus	COORDINADOR, NODO
2	# vi /etc/hosts	Agregar las direcciones ip con su respectivo nombre de los servidores involucrados en el clúster	
3	# yum install -y citus74_10	Instalar postgresql con su extensión citus	
4	# /usr/pgsql-10/bin/postgresql-10-setup initdb	Inicializar servidor postgresql	
5	# echo "shared_preload_libraries = 'citus'"	Agregar extensión citus al archivo de configuración de postgresql	
6	# vi /var/lib/pgsql/11/data/postgresql.conf	En el archivo descomentar listen_addresses (direcciones a	

		escuchar) y port (puerto) de la sección Connections and Authentication (conexiones y autenticación).	
7	# vi /var/lib/pgsql/11/data/pg_hba.conf	Archivo para agregar redes que van a acceder al servidor, por ejemplo: host all all 172.0.0.1/8 trust Es importante destacar que el método de encriptado de contraseñas siempre debe ser trust	
8	# service postgresql-11 restart # chkconfig postgresql-10 on	Reiniciar y habilitar encendido automático de postgresql	
9	# sudo -i -u postgres psql -c "CREATE EXTENSION citus;"	Ejemplo de agregar citus a la base de datos postgres que se crear por defecto	
10	# firewall-cmd --zone=public --add-port=5432/tcp --permanent # firewall-cmd –reload	Permitir conexión remota mediante el puerto que se asigna para postgresql en este caso 5432	
11	# sudo -i -u postgres psql -c "SELECT * from master_add_node('nombre del nodo', Puerto del nodo);"	Agregar un nodo fuera del coordinador para el clúster	COORDINADOR
12	# sudo -i -u postgres psql -c "SELECT * FROM master_get_active_worker_nodes();"	Verificar nodos activos del clúster	

Realizado por: Zaruma Jorge. 2019

- **Servidor web**

Como se tiene un clúster de servidor de aplicaciones los sistemas desplegados serán redundantes, pero es necesario tener un punto de entrada por lo que la implementación de un servidor web como balanceador de carga permitirá distribuir peticiones hacia distintos servidores, pero también este punto de entrada podría ser objeto de ataques dos lo que inhabilitaría el acceso a las aplicaciones, para corregir este problema se agregó alta disponibilidad para el servidor web como se observa en la **Figura 17-2** que ahora recibe peticiones por una ip virtual que monitoriza el estado de ambos nodos y utilizar el que se encuentre activo, el manual de configuración con los pasos que se explican de una manera más detallada se encuentran en el **Anexo I**.

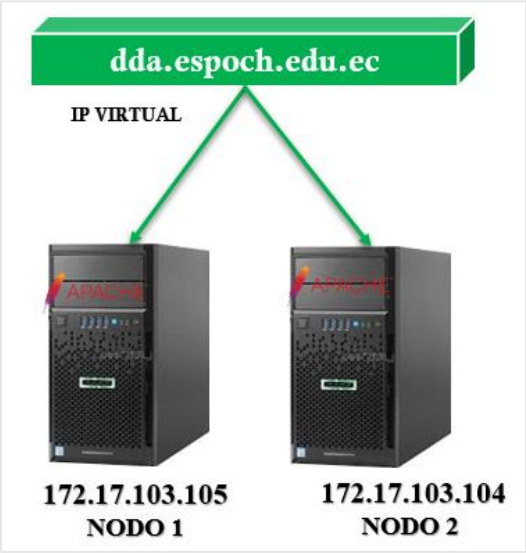


Figura 18-2: Escenario para configurar servidor web
Realizado por: Zaruma Jorge. 2019

En la **Tabla 29-2** se visualiza los pasos a seguir para la instalación y configuración del clúster de servidor web con apache, además se agrega una descripción de cada uno de los pasos de manera general y por último se especifica en que servidor se debe ejecutar, para una mejor comprensión NODO 1 hace referencia al servidor principal y NODO al servidor o los servidores secundarios que integraran el clúster.

Tabla 29-2: Instalación de apache y configuración de clúster de alta disponibilidad

	COMANDOS	**DESCRIPCIÓN**	**SERVIDOR**
1	# yum install httpd –y	Instalar apache	NODO 1, NODO
2	# vi /etc/hosts	Agregar las direcciones ip con su respectivo nombre de los servidores involucrados en el clúster	
3	# firewall-cmd --zone=public --add-port=80/tcp --permanent # firewall-cmd --reload	Permitir conexión remota mediante el puerto que se asigna para apache en este caso es el 80	
4	# sed -i 's/Listen/#Listen/' /etc/httpd/conf/httpd.conf	Comentar todos los puertos o direcciones ip que estén escuchando apache	
5	# systemctl restart httpd	Reiniciar apache	
6	# yum install corosync pcs pacemaker –y	Instalar herramientas para configurar alta disponibilidad	
7	#passwd hacluster	Asignar misma contraseña para el usuario hacluster	
8	#systemctl start pcsd	Iniciar la herramienta pcsd para manipular los recursos del clúster	
9	# firewall-cmd --zone=public --add-port=2224/tcp --permanent # firewall-cmd --permanent --add-service=high-availability # firewall-cmd --add-service=high-availability # firewall-cmd –reload	Permitir conexiones por puerto y por servicio en la tabla de firewall y por ultimo reiniciar.	
10	# pcs cluster auth nodo1 nodo2	Verificar autenticación de los nodos que conformaran el clúster	NODO 1
11	pcs cluster setup --name cluster_httpd nodo1 nodo2	Crear un clúster y asignar un nombre en este caso se denomina cluster_httpd	
12	pcs cluster start –all	Iniciar clúster	
13	# pcs property set stonith-enabled=false	Evitar cercado de nodos	
14	# pcs property set no-quorum-policy=ignore	Ignorar quorum, esta propiedad debe habilitarse si existe 4 nodos en el clúster	
15	# pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=172.17.103.110 cidr_netmask=24 op monitor interval=30s	Crear recursos de ip virtual que se monitorea cada 30 segundos	
16	# echo "Listen 172.17.103.110:80"|sudo tee --append /etc/httpd/conf/httpd.conf	Asignar al archive de configuración de apache que escuche mediante la ip virtual	NODO 1, NODO
17	pcs resource create webserver ocf:heartbeat:apache configfile=/etc/httpd/conf/httpd.conf statusurl="http://localhost/server-status" op monitor interval=1min	Crear recurso y asignar un nombre en este caso es webserver porque hace referencia al servidor web apache con un monitoreo cada minuto	NODO 1
18	# pcs constraint colocation add webserver virtual_ip INFINITY	Restricción para que los recursos siempre se inicien en un nodo	
19	# pcs constraint order virtual_ip then webserver	Restricción para que el recurso de ip virtual se inicie antes que webserver	

20	# pcs constraint location webserver prefers nodo1=50	Restricción para que el recursos webserver se inicie en el nodo 1 si se encuentra activo	
21	#vi /etc/httpd/conf.d/ddaespoch.conf	Crear archivo de configuración adicional para el balanceador de carga y agregarle un nombre en este caso se denomina ddaespoch.conf al mismo que se debe agregar el contenido del Anexo F	
22	# pcs cluster stop --all && sudo pcs cluster start –all	Detener e iniciar el clúster	

Realizado por: Zaruma Jorge. 2019

2.4.3.4. Fase 4: Escenarios de evaluación

En esta fase se debe investigar sobre herramientas que permitan verificar si la arquitectura que se implanta cumple con lo establecido.

- **Evaluación por componente**

Herramientas utilizadas

httpd-tools: es un paquete que tiene herramientas incluidas que se pueden utilizar en conjunto con apache server para estas pruebas se utilizara las herramientas de evaluación comparativa o más conocida con su nombre en inglés benchmarking tool.

lsof: herramienta de monitorización para los sistemas linux en este caso se utilizará sobre un puerto específico.

Navegador: puede ser el de su preferencia, el cual ayudara a ingresar a la administración remota de payara.

PgAdmin: herramienta de administración del sistema gestor de base de datos Postgresql, se utiliza la versión 4.3

Escenario 1: Clúster de servidor de aplicaciones

En la **Figura 19-2** se visualiza de izquierda a derecha un servidor encargado del envío masivo de peticiones y el balanceador de cargar y por último el bloque de servidores de aplicaciones que conforman un clúster denominados: DAS, nodo 1 y nodo 2.

Nota: La tipografía empleada en las tablas del presente capitulo es Courier New por ser recomendada para el texto de consola.

Figura 19-2: Esquema de balanceador de carga y servidores de aplicaciones
Realizado por: Zaruma Jorge. 2019

Objetivo

Constatar que el balanceador redirija las peticiones a las instancias activas del clúster de payara

Procedimiento

Para este escenario se utilizará dos instancias activas y una inactiva del clúster, así como se puede apreciar en la **Figura 20-2** para así verificar si el balanceador de carga no presenta ningún inconveniente al realizar las peticiones.

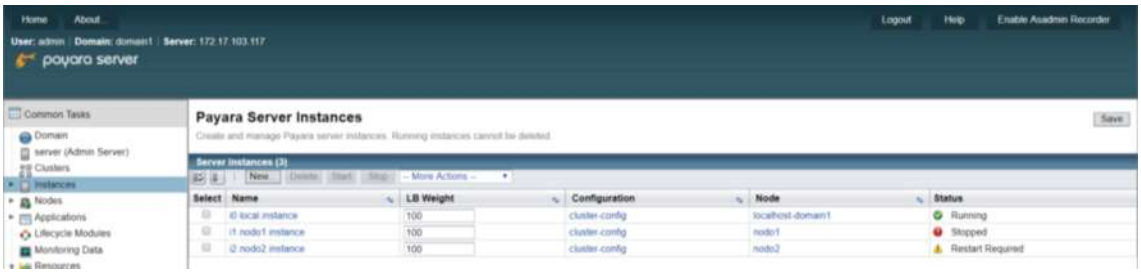


Figura 20-2: Servidor de aplicaciones principal
Realizado por: Zaruma Jorge. 2019

También es importante mencionar que el tráfico hacia el servidor web va ser simulado mediante httpd-tools y para visualizar el tráfico en los servidores se utiliza lsof.

En la **Tabla 30-2** se procede a ejecutar en el nodo coordinador un comando que permite enviar varias peticiones simultaneas de la siguiente manera:

 # ab -n 100000 -c 10 http://dda.espoch.edu.ec/Menu

ab: es un comando que tiene integrado el servicio de las herramientas apache.

-n 100000: representa el número de solicitudes a realizar

-c 250: representa el número de solicitudes simultaneas a realizar

http://dda.espoch.edu.ec/Menu: es la dirección del servicio a donde se van a enviar las solicitudes

Tabla 30-2: Comando a ejecutar en el nodo coordinador del escenario 1 de disponibilidad

COORDINADOR
`[root@localhost ~]# ab -n 100000 -c 250 http://dda.espoch.edu.ec/Menu/`

Realizado por: Zaruma Jorge. 2019

Además, en la **Tabla 31-2**, **Tabla 32-2** y **Tabla 33-2** correspondientes al nodo DAS, 1 y 2, se ejecuta el comando **# netstat -an | grep :28080 | grep ESTABLISHED |wc -l** el mismo que permite contabilizar el número de conexiones establecidas en el puerto 28080 en un tiempo determinado.

Tabla 31-2: Comando a ejecutar en el nodo 2 del escenario 1 de disponibilidad

NODO 2
`[root@nodo2 ~]# netstat -an | grep :28080 | grep ESTABLISHED |wc -l`

Realizado por: Zaruma Jorge. 2019

Tabla 32-2: Comando a ejecutar en el nodo das del escenario 1 de disponibilidad

DAS
`[root@das ~]# netstat -an | grep :28080 | grep ESTABLISHED |wc -l`

Realizado por: Zaruma Jorge. 2019

Tabla 33-2: Estado en el nodo 1 luego de ejecutar el comando lsof

NODO 1
`[root@nodo1 ~]# netstat -an | grep :28080 | grep ESTABLISHED |wc -l`

Realizado por: Zaruma Jorge. 2019

Como se observó anteriormente los servidores tienen que ejecutar el comando **netstat** a un puerto especifico en este caso el 28080 la razón, es porque ahí donde se encuentran desplegadas las aplicaciones como se puede visualizar en la **Figura 21-2.**

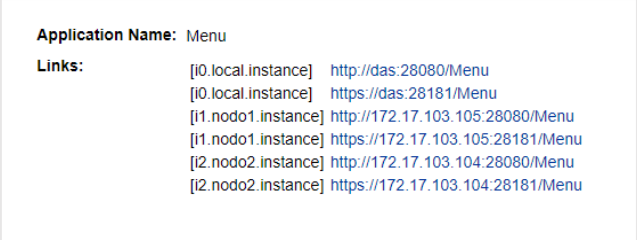


Figura 21-2: Despliegue de la aplicación menú en el clúster payara
Realizado por: Zaruma Jorge. 2019

Una vez definida la estructura y los elementos que se van a utilizar se procede a ejecutar él envió de peticiones con el comando de la **Tabla 31-2**. Al realizar este proceso se puede apreciar en la **Tabla 34-2** el resultado que brinda la herramienta httpd-tools.

Tabla 34-2: Estado en el nodo coordinador luego de ejecutar el comando ab

COORDINADOR
`[root@localhost ~]# ab -n 100000 -c 400 http://dda.espoch.edu.ec/Menu/`

`Benchmarking dda.espoch.edu.ec (be patient)`
`Completed 10000 requests`
`…….`
`Completed 90000 requests`
`Completed 100000 requests`
`Finished 100000 requests`
`Server Software: Payara`
`Server Hostname: dda.espoch.edu.ec`
`Server Port: 80`
 	
`Document Path: /Menu`
`Document Length: 180 bytes`
 	
`Concurrency Level: 250`
`Time taken for tests: 61.348 seconds`
`Complete requests: 100000`
`Failed requests: 0`
`Write errors: 0`
`Non-2xx responses: 100000`
`Total transferred: 55000000 bytes`
`HTML transferred: 18000000 bytes`
`Requests per second: 1630.05 [#/sec] (mean)`
`Time per request: 153.369 [ms] (mean)`
`Time per request: 0.613 [ms] (mean, across all concurrent requests)`
`Transfer rate: 875.52 [Kbytes/sec] received`

Realizado por: Zaruma Jorge. 2019

Y finalmente en la **Figura 22-2** se visualiza información acerca del número de conexiones que se ha establecido en los nodos das, 1 y 2 involucrados en el siguiente escenario en el puerto 28080 a través de la herramienta netstat.

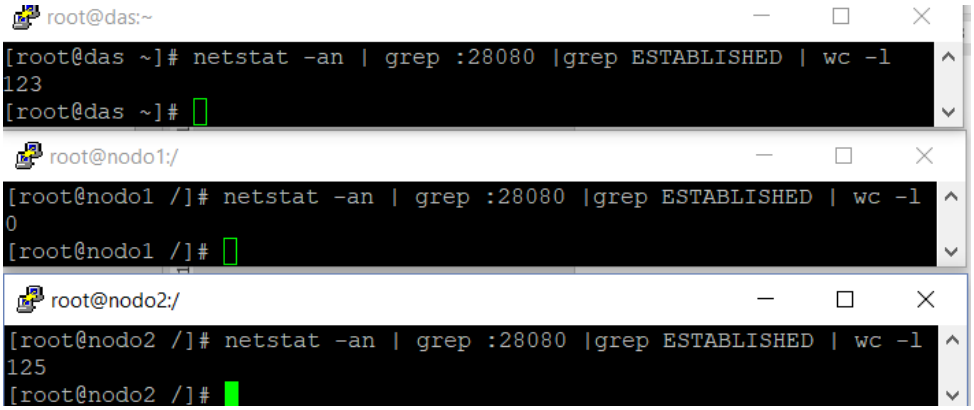


Figura 22-2: Número de conexiones en los nodos del clúster de payara
Realizado por: Zaruma Jorge. 2019

Resultado de escenario:

Tabla 35-2: Datos luego de haber realizado el escenario 1 de disponibilidad

Peticiones	Concurrencia	Solicitudes	Duración de Escenario	Solicitudes por Segundo
100,000	250(peticiones)	100,000	61.348 (segundos)	1,630.05 (media)

Realizado por: Zaruma Jorge. 2019

Mediante la utilización de lsof como herramienta de monitorización se observó que las peticiones únicamente fueron dirigidas a las instancias activas del clúster de payara, y con la ayuda de netstat se contabilizo el número de conexiones en cada instancia. Por lo tanto, en la **Tabla 35-2** se observa que las 100,000 peticiones realizadas con una concurrencia de 250 solicitudes fueron distribuidas para las 2 instancias activas, por ende, cada instancia ejecuto 125 solicitudes como se observa en la **Figura 22-2**, tardando 61 segundos en completar todas las solicitudes con una media de 1630 peticiones por segundo.

Escenario 2: Clúster del servidor web

En la **Figura 23-2** se puede apreciar de izquierda a derecha un servidor quien envía peticiones masivas hacia el dominio dda.espoch.edu.ec el mismo que es el encargado de monitorizar el servidor web que se encuentre activo, además se dispone de un bloque de dos servidores web u no activo y otro pasivo y por ultimo una instancia del clúster de servidor de aplicaciones denominado das.

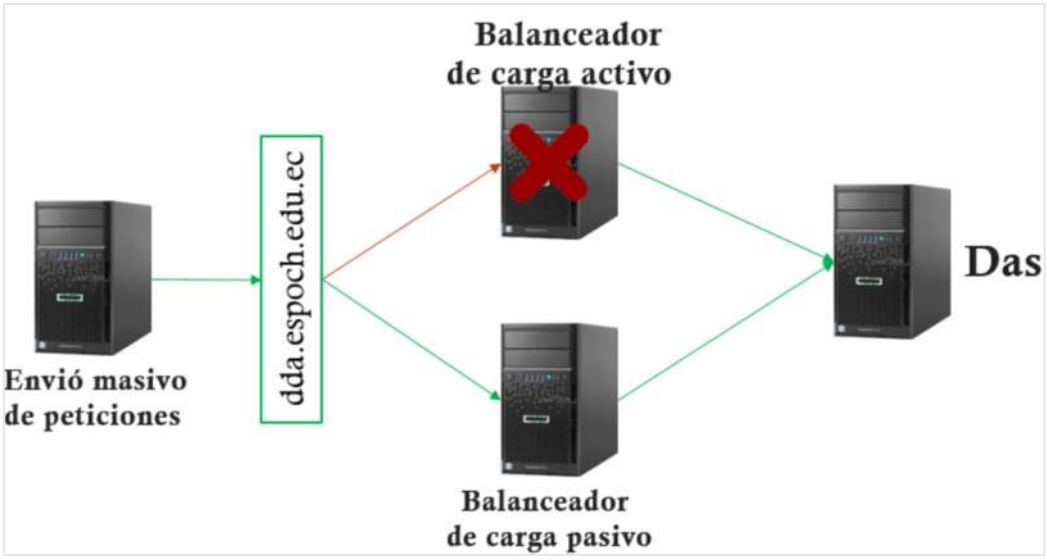


Figura 23-2: Esquema de balanceadores de carga
Realizado por: Zaruma Jorge. 2019

Objetivo

Constatar que el clúster del servidor web apache funcione correctamente cuando el nodo activo cese sus funciones.

Procedimiento

Tanto en el nodo 1 y 2 se ejecuta **# pcs status** para verificar el estado del clúster del servidor web, al ejecutar el comando se obtiene como resultado que el recurso apache y la ip virtual se encuentra ejecutando en el nodo 1 como se observa en la **Tabla 36-2** y **Tabla 37-2**

Tabla 36-2: Estado del clúster de apache en el nodo 1

NODO 1
`[root@nodo1 ~]# pcs status`

`Cluster name: cluster_httpd`
`Stack: corosync`
`Current DC: nodo2 (version 1.1.18-11.el7_5.3-2b07d5c5a9) - partition with quorum`
`Last updated: Tue Jan 15 19:26:46 2019`
`Last change: Wed Jan 9 15:26:34 2019 by root via cibadmin on nodo1`
`2 nodes configured`
`2 resources configured`
`Online: [ nodo1 nodo2 ]`
`Full list of resources:`
` webserver (ocf::heartbeat:apache): Started nodo1`
` virtual_ip (ocf::heartbeat:IPaddr2): Started nodo1`
`Daemon Status:`

```
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Realizado por: Zaruma Jorge. 2019

Tabla 37-2: Estado del clúster en el nodo 2

NODO 2
`[root@nodo2 ~]# pcs status`

`Cluster name: cluster_httpd`
`Stack: corosync`
`Current DC: nodo2 (version 1.1.18-11.el7_5.3-2b07d5c5a9) - partition with quorum`
`2 nodes configured`
`2 resources configured`
`Online: [ nodo1 nodo2 ]`
`Full list of resources:`
` webserver (ocf::heartbeat:apache): Started nodo1`
` virtual_ip (ocf::heartbeat:IPaddr2): Started nodo1`
`Daemon Status:`
`  corosync: active/enabled`
`  pacemaker: active/enabled`
`  pcsd: active/enabled`

Realizado por: Zaruma Jorge. 2019

Posterior en la **Figura 24-2** se representa la información de las instancias del clúster del servidor de aplicaciones para este escenario, solo la instancia local va a estar activa, la razón es porque en la figura anterior se menciona que el clúster de apache está en los nodos 1 y 2 por lo tanto es necesario monitorear solo las peticiones realizadas para el servidor web.

Figura 24-2: Estado de los servidores luego durante él envió de peticiones.
Realizado por: Zaruma Jorge. 2019

Una vez establecido las condiciones de ejecución se procede a ejecutar en los 4 servidores involucrados representados en tablas los siguientes comandos:

- En la **Tabla 38-2** correspondiente al nodo das se ejecuta el envió de peticiones mediante la herramienta httpd-tools
- En la **Tabla 39-2** que corresponde al nodo DAS se procede a monitorizar en tiempo real el puerto 28080 con la herramienta lsof
- En la **Tabla 40-2** se procede a detener el clúster en el nodo 1 con pcs.
- Por ultimo en la **Tabla 41-2** se verifica el estado del clúster luego de ejecutar el comando de la **Tabla 40-2.**

Tabla 38-2: Comando a ejecutar en el nodo coordinador

COORDINADOR
`[root@localhost ~]# ab -n 100000 -c 10 http://dda.espoch.edu.ec/clusterjsp`

Realizado por: Zaruma Jorge. 2019

Tabla 39-2: Comando a ejecutar en el nodo das

DAS
`[root@das ~]# watch lsof -i :28080`

Realizado por: Zaruma Jorge. 2019

Tabla 40-2: Comando a ejecutar en el nodo 1

NODO 1
`[root@nodo1 ~]# pcs cluster stop nodo1`

Realizado por: Zaruma Jorge. 2019

Tabla 41-2: Comando a ejecutar en el nodo 2

NODO 2
`[root@nodo2 ~]# pcs status`

Realizado por: Zaruma Jorge. 2019

El resultado de haber ejecutado los comandos anteriores se puede observar en la **Tabla 42-2**, **Tabla 43-2**, **Tabla 44-2** y la **Tabla 45-2**.

Tabla 42-2: Estado del nodo coordinador luego de haber ejecutado el comando ab

COORDINADOR
`[root@localhost ~]# ab -n 100000 -c 10 http://dda.espoch.edu.ec/clusterjsp`

`This is ApacheBench, Version 2.3 <$Revision: 1430300 $>`
`Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/`
`Licensed to The Apache Software Foundation, http://www.apache.org/`
`Benchmarking dda.espoch.edu.ec (be patient)`

```
Completed 10000 requests
……
Finished 100000 requests
Server Software:        Payara
Server Hostname:        dda.espoch.edu.ec
Server Port:            80
Document Path:          /clusterjsp
Document Length:        186 bytes
Concurrency Level:      10
Time taken for tests:   64.667 seconds
Complete requests:      100000
Failed requests:        0
Write errors:           0
Non-2xx responses:      100000
Total transferred:      56200000 bytes
HTML transferred:       18600000 bytes
Requests per second:    1546.38 [#/sec] (mean)
Time per request:       6.467 [ms] (mean)
Time per request:       0.647 [ms] (mean, across all concurrent requests)
Transfer rate:          848.70 [Kbytes/sec] received
```

Realizado por: Zaruma Jorge. 2019

Tabla 43-2: Conexiones establecidas en el nodo DAS

DAS

```
[root@das ~]# watch lsof -i :28080
```

```
Every 2,0s: lsof -i :28080

COMMAND  PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
java    7662 payara  427u  IPv6 594599      0t0  TCP *:28080 (LISTEN)
java       7662  payara    499u    IPv6  752675          0t0    TCP  das:28080-
>172.17.103.104:44936 (ESTABLISHED)
java       7662  payara    564u    IPv6  752676          0t0    TCP  das:28080-
>172.17.103.104:44942 (ESTABLISHED)
java       7662  payara    565u    IPv6  752677          0t0    TCP  das:28080-
>172.17.103.104:44950 (ESTABLISHED)
java       7662  payara    566u    IPv6  752678          0t0    TCP  das:28080-
>172.17.103.104:44954 (ESTABLISHED)
java       7662  payara    567u    IPv6  752679          0t0    TCP  das:28080-
>172.17.103.104:44960 (ESTABLISHED)
java       7662  payara    568u    IPv6  752680          0t0    TCP  das:28080-
>172.17.103.104:44966 (ESTABLISHED)
java       7662  payara    570u    IPv6  752681          0t0    TCP  das:28080-
>172.17.103.104:44972 (ESTABLISHED)
……..
```

Realizado por: Zaruma Jorge. 2019

Tabla 44-2: Clúster de servidor web en el nodo 1 detenido.

NODO 1
`[root@nodo1 ~]# pcs cluster stop nodo1`
`nodo1: Stopping Cluster (pacemaker)...` `nodo1: Stopping Cluster (corosync)...`

Realizado por: Zaruma Jorge. 2019

Tabla 45-2: Clúster del servidor web iniciados en el nodo 2

NODO2
`[root@nodo2 ~]# pcs status`

`[root@nodo2 ~]# pcs status`
`Cluster name: cluster_httpd`
`Stack: corosync`
`Current DC: nodo2 (version 1.1.18-11.el7_5.3-2b07d5c5a9) - partition with quorum`
`Last updated: Tue Jan 15 19:42:02 2019`
`Last change: Wed Jan 9 15:26:34 2019 by root via cibadmin on nodo1`
`2 nodes configured`
`2 resources configured`
`Online: [ nodo2 ]`
`OFFLINE: [ nodo1 ]`
`Full list of resources:`
` webserver (ocf::heartbeat:apache): Started nodo2`
` virtual_ip (ocf::heartbeat:IPaddr2): Started nodo2`
`Daemon Status:`
`  corosync: active/enabled`
`  pacemaker: active/enabled`
`  pcsd: active/enabled`
`[root@nodo2 ~]#`

Realizado por: Zaruma Jorge. 2019

Resultado de escenario:

Uno de los puntos críticos es el servidor web porque responde a todas las solicitudes, ocasionando que llegue a su capacidad máxima con lo cual quedara inhabilitado, para esto se realizó alta disponibilidad y quedó demostrado en este escenario que si el nodo principal del servidor web se detiene toma el control el nodo secundario, con este método los usuarios accederán al sistema solo con unos momentos de inactividad hasta que el nodo secundario tome el control total como

servidor web principal concluyendo así que el sistema seguirá disponible si se presentara dicha condición de cese de función.

Escenario 3: Clúster de base de datos

En la **Figura 25-2** se representa de izquierda a derecha los siguientes componentes para el escenario, el primero corresponde al nodo coordinador administrado por pgadmin, luego el mismo nodo coordinador, pero accedido remotamente encargado de la escritura de las peticiones y por último el bloque de servidores denominados nodo 1 y nodo 2 que serán los que realizarán el proceso de lectura de las peticiones.

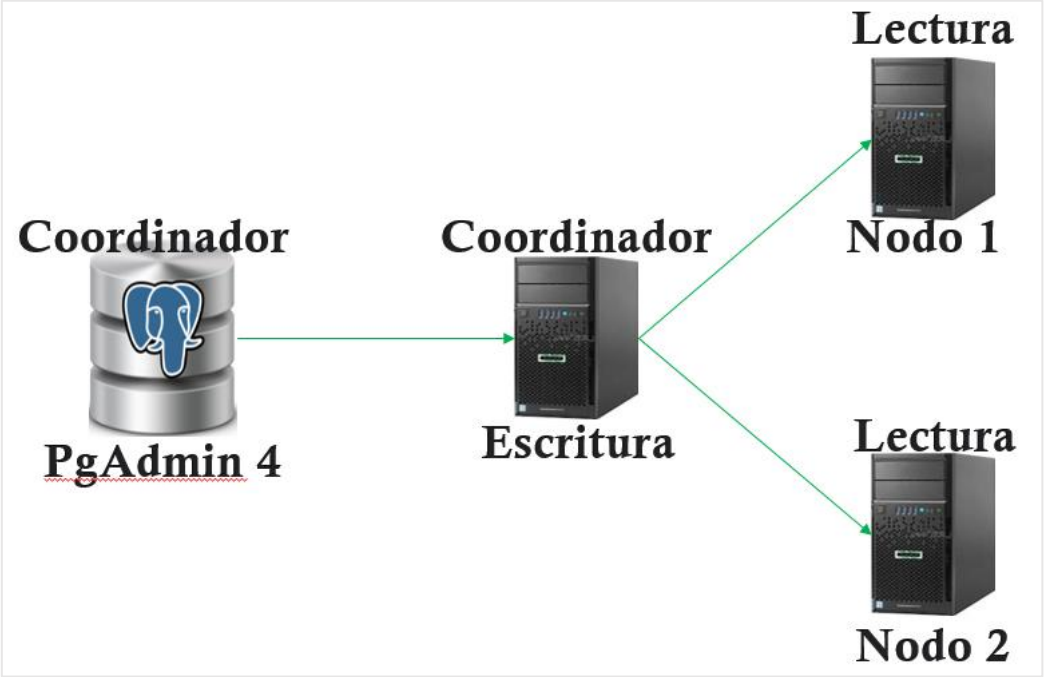


Figura 25-2: Esquema de los servidores de base de datos
Realizado por: Zaruma Jorge. 2019

Objetivo

Verificar que las tablas se distribuyan y las peticiones al servidor sean dirigidas a los nodos que deben realizar dichas peticiones

Procedimiento

Base de datos

Algo muy importante es no sobrecargar de transacciones al servidor de base de datos para lo cual mediante citus se logró distribuir las dichas transacciones entre los nodos que conforma el clúster.

Al igual que en el clúster creado en el servidor de aplicaciones los servidores están denominados de la siguiente manera para una mejor asimilación de los procesos que se van a llevar acabo, en la **Figura 26-2** se visualiza la administración remota de los servidores de base datos mediante pgadmin que están numeradas de la siguiente manera.

- La imagen superior izquierda representa al nodo 1 de la base de datos donde se muestra las tablas tbsilabo con su identificador
- La imagen inferior izquierda representa al nodo 2 de la base de datos donde se muestra las tablas tbsilabo con su identificador
- La imagen de la derecha representa al nodo coordinador de la base datos donde se ha creado una tabla denominada tbsilabo y ejecutado la sentencia para que distribuya en los nodos

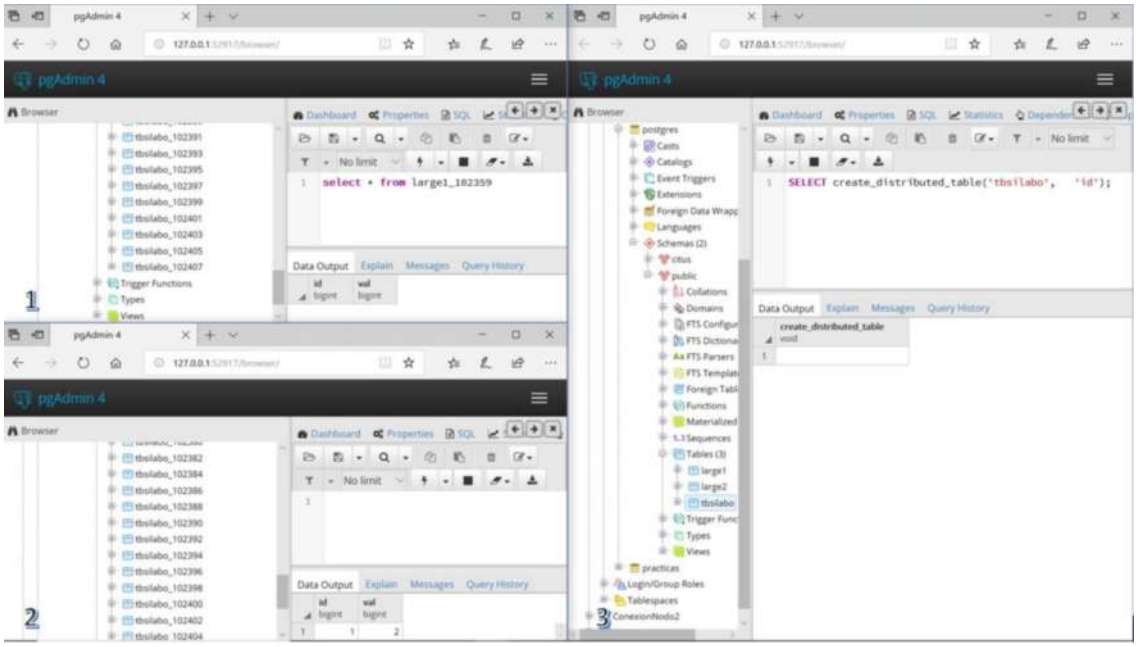


Figura 26-2: Distribución del clúster de base de datos
Realizado por: Zaruma Jorge. 2019

Citus ejecuta las sentencias INSERT, UPDATE Y DELETE en el nodo coordinador y las sentencias SELECT en los nodos que conforman el clúster es así que como se muestra en la **Figura 27-2** se ha ingresado tres registros y que han sido distribuidos equitativamente en la tabla distribuida anteriormente.

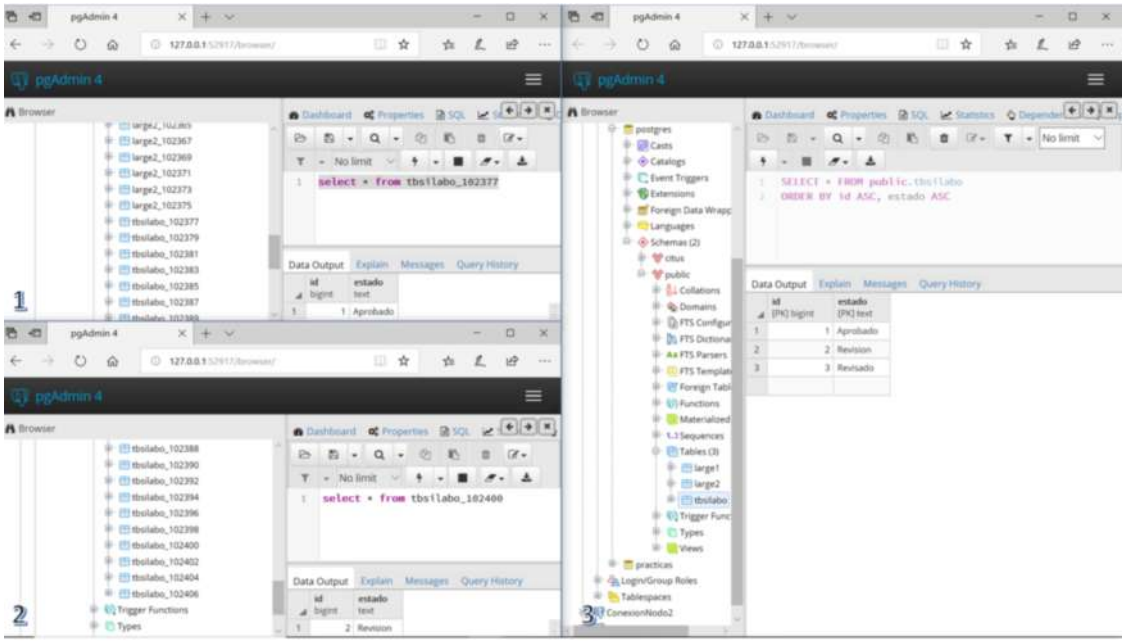


Figura 27-2: Distribución de registros ingresados
Realizado por: Zaruma Jorge. 2019

Resultado de escenario:

En este escenario se puedo constatar que el clúster de base de datos mediante citus funciona adecuadamente es decir se puede distribuir las tablas mediante su id, además las sentencias realizadas al servidor principal son distribuidas, esto ayuda a que no se sobrecargue de trabajo solo el nodo coordinador y así evitar perdida de información incluso inactividad de los sistemas.

- **Evaluación por solución.**

Una vez concluido todo el proceso de configuración se procede a realizar un escenario para verificar el comportamiento de la arquitectura implantada la misma que está disponible en el capítulo 3 como parte de la evaluación de fiabilidad según la norma ISO/IEC 25010

2.4.4. Gestión del Proyecto

BurnDown Chart

Para la gestión del proyecto se utilizó el diagrama BurnDown Chart, mediante el cual las personas involucradas en el proyecto visualizan el tiempo de ejecución y durante cada sprint visualizar si los puntos estimados correspondiente a la línea de color azul son iguales a los puntos reales que es la línea de color naranja, gracias a esto permite conocer cuáles fueron los sprints que no se cumplieron dentro del tiempo estimado. Y como se puede observar en el **Grafico 1-2** el eje x corresponde al número de sprints realizados, y el eje y corresponde al esfuerzo total estimado.

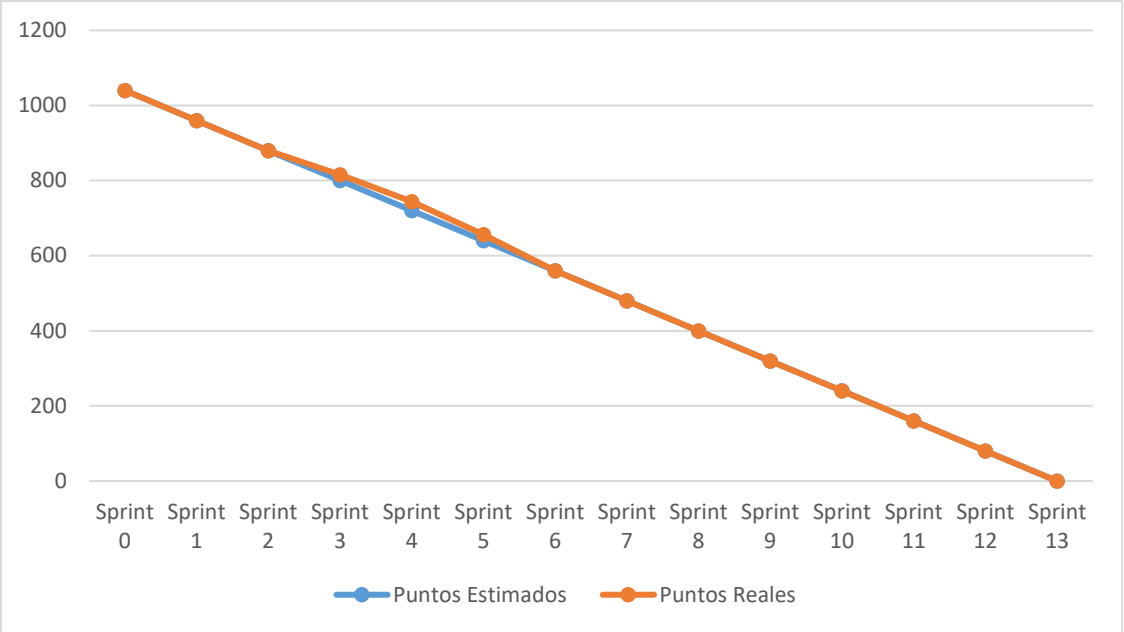


Gráfico 1-2: Gestión del Proyecto
Realizado por: Zaruma Jorge. 2019

Como se puede observar en el **Gráfico 1-2** anterior existe una variación en los sprints 3, 4, 5 es aquí donde se realizó la configuración de servidores donde se presentaron errores que se tuvieron que corregir por ende el tiempo real sobrepaso al estimado, en total hubo una variación de 7 días al tiempo estimado, los mismos que fueron como un trabajo extra para que el tiempo de entrega del proyecto final no se vea afectado.

CAPÍTULO III

3. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS DE RESULTADOS

Una vez concluido las configuraciones de la arquitectura tolerante a fallos realizado en el capítulo 2, es necesario comprobar cada uno de sus componentes para así verificar su correcto funcionamiento y constatar que se cumplan las subcaracterísticas de la fiabilidad correspondiente a la ISO/IEC 25010 que son la madurez, disponibilidad, tolerancia a fallos y capacidad de recuperación.

3.1. PRUEBA DE FIABILIDAD

Herramientas utilizadas

Jmeter: se utiliza para simular una carga pesada en un servidor

BadBoy: Herramienta que permite capturar el proceso de ejecución de un sistema.

3.1.1. Comportamiento de la arquitectura tolerante a fallas

En la **Figura 1-3** se aprecia la arquitectura tolerante a fallos, la misma que se someterá a evaluación en base a 3 casos mediante jmeter en conjunto con badboy.

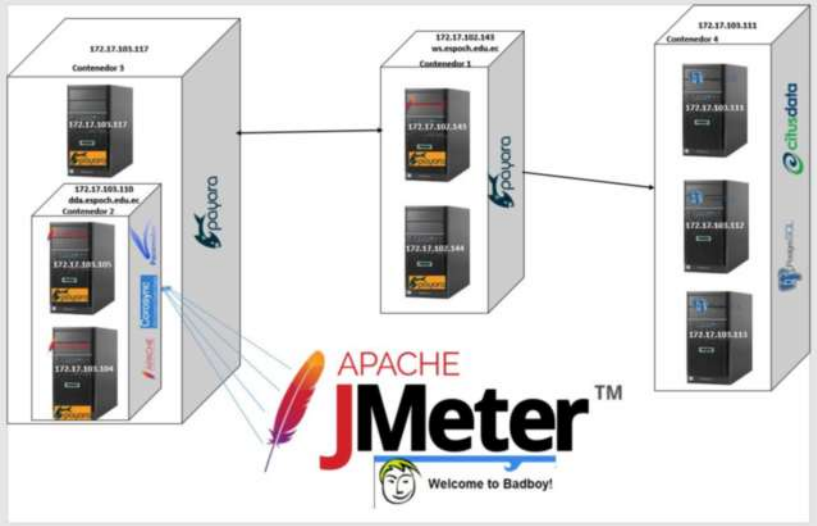


Figura 1-3: Esquema de escenario con jmeter y badboy
Realizado por: Zaruma Jorge. 2019

Objetivo

Verificar el comportamiento de la arquitectura tolerante a fallos mediante la utilización de jmeter.

Procedimiento

Mediante el programa badboy se grabó la ejecución de 4 módulos del sistema de programas analíticos para ejecutar el plan de pruebas como se observa en la **Figura 2-3**.

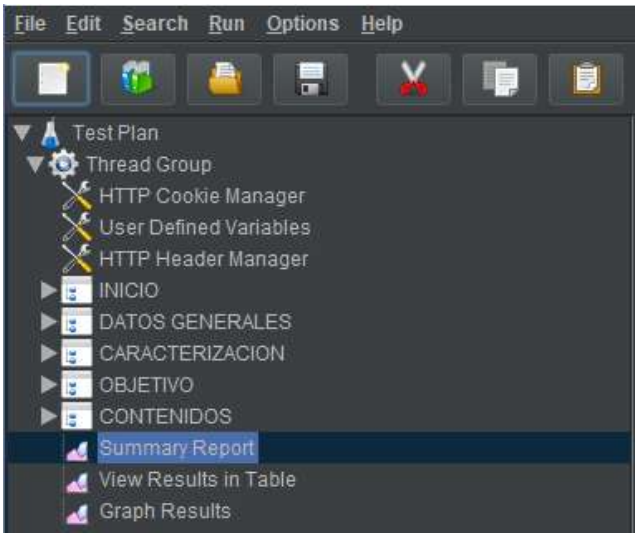
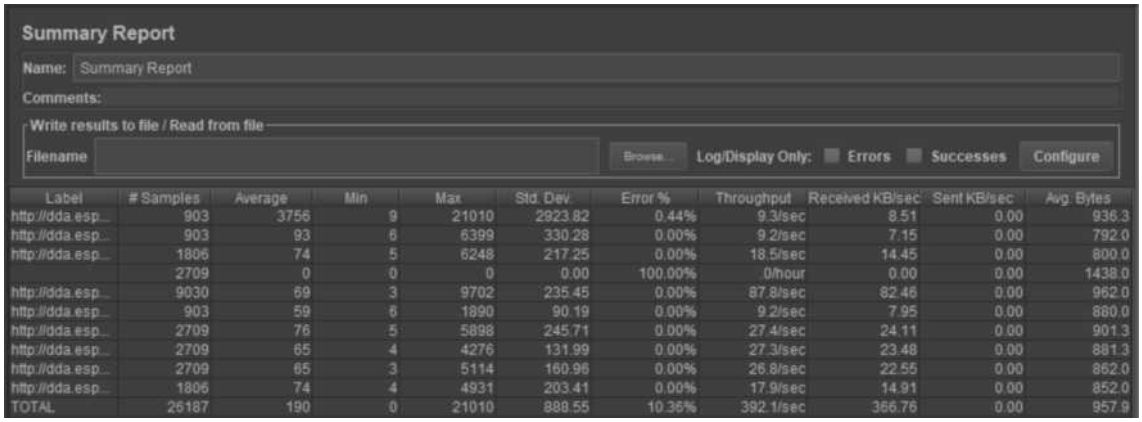


Figura 2-3: Test del sistema de programas analíticos
Realizado por: Zaruma Jorge. 2019

Escenario 1:

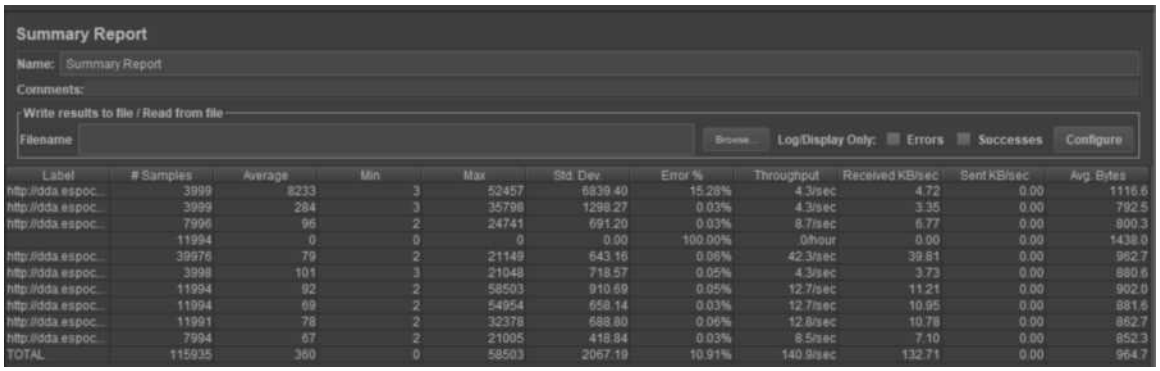
Caso 1: En la **Figura 3-3** se evaluó 500 usuarios con respuesta de 1 segundo por cada usuario, sin ciclos de repetición y con la particularidad de dos instancias activas.



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://dda.esp...	903	3756	9	21010	2923.82	0.44%	9.3/sec	8.51	0.00	936.3
http://dda.esp...	903	93	6	6399	330.28	0.00%	9.2/sec	7.15	0.00	792.0
http://dda.esp...	1806	74	5	6248	217.25	0.00%	18.5/sec	14.45	0.00	800.0
	2709	0	0	0	0.00	100.00%	0/hour	0.00	0.00	1438.0
http://dda.esp...	9030	69	3	9702	235.45	0.00%	87.8/sec	82.46	0.00	962.0
http://dda.esp...	903	59	6	1890	90.19	0.00%	9.2/sec	7.95	0.00	880.0
http://dda.esp...	2709	76	5	5898	245.71	0.00%	27.4/sec	24.11	0.00	901.3
http://dda.esp...	2709	65	4	4276	131.99	0.00%	27.3/sec	23.48	0.00	881.3
http://dda.esp...	2709	65	3	5114	160.96	0.00%	26.8/sec	22.55	0.00	862.0
http://dda.esp...	1806	74	4	4931	203.41	0.00%	17.9/sec	14.91	0.00	852.0
TOTAL	26187	190	0	21010	888.55	10.36%	392.1/sec	366.76	0.00	957.9

Figura 3-3: Escenario de 500 usuarios y dos instancias activas
Realizado por: Zaruma Jorge. 2019

Caso 2: En la **Figura 4-3** se evaluó 2000 usuarios con respuesta de 1 segundo por cada usuario, en dos instancias activas

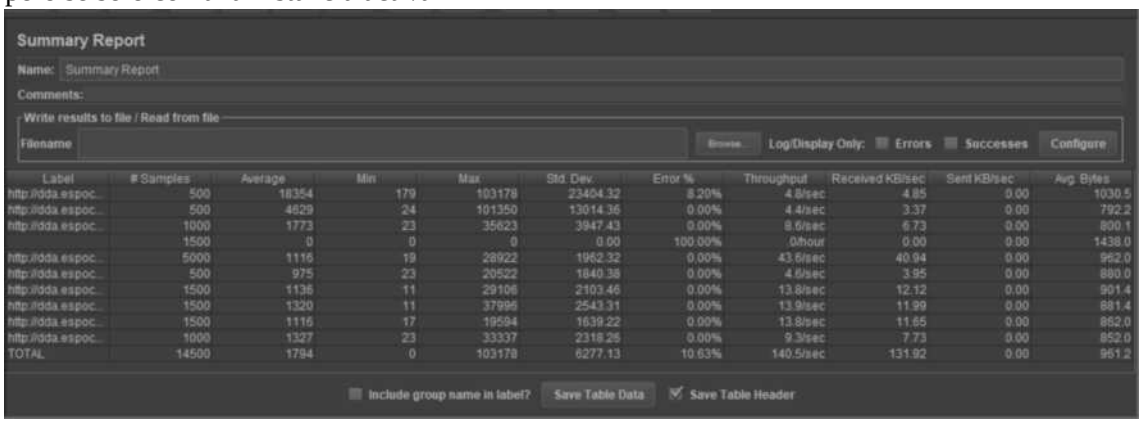


Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://dda.espoc...	3999	8233	3	52457	6839.40	15.28%	4.3/sec	4.72	0.00	1116.6
http://dda.espoc...	3999	284	3	35798	1298.27	0.03%	4.3/sec	3.35	0.00	792.5
http://dda.espoc...	7996	96	2	24741	691.20	0.03%	8.7/sec	6.77	0.00	800.3
	11994	0	0	0	0.00	100.00%	0/hour	0.00	0.00	1438.0
http://dda.espoc...	39976	79	2	21149	643.16	0.06%	42.3/sec	39.81	0.00	962.7
http://dda.espoc...	3998	101	3	21048	718.57	0.05%	4.3/sec	3.73	0.00	880.6
http://dda.espoc...	11994	92	2	58503	910.69	0.05%	12.7/sec	11.21	0.00	902.0
http://dda.espoc...	11994	69	2	54954	658.14	0.03%	12.7/sec	10.95	0.00	881.6
http://dda.espoc...	11991	78	2	32378	688.80	0.06%	12.8/sec	10.78	0.00	862.7
http://dda.espoc...	7994	67	2	21005	418.84	0.03%	8.5/sec	7.10	0.00	852.3
TOTAL	115935	360	0	58503	2067.19	10.91%	140.9/sec	132.71	0.00	964.7

Figura 4-3: Escenario de 2000 usuarios y dos instancias activas
Realizado por: Zaruma Jorge. 2019

Escenario 2:

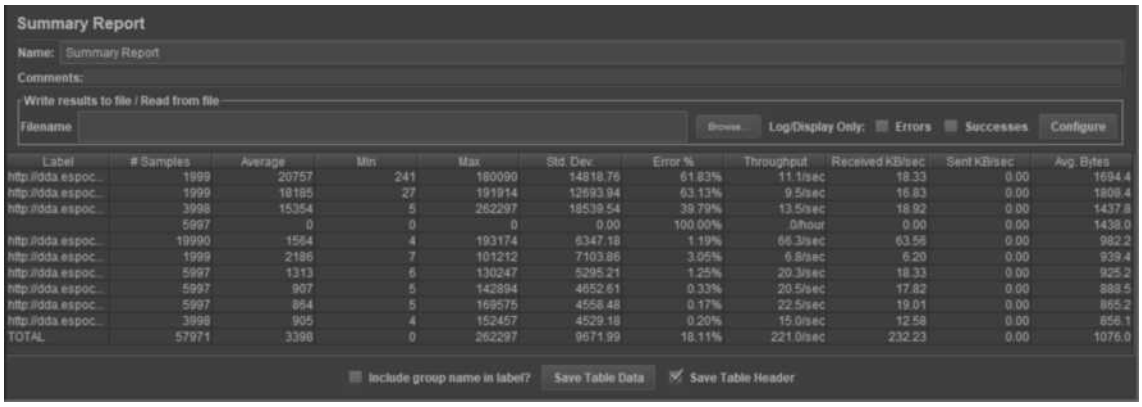
Caso 1: En la **Figura 5-3** se evaluó 500 usuarios con respuesta de 1 segundo por cada usuario, pero se solo con una instancia activa



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://dda.espoc...	500	18354	179	103178	23404.32	8.20%	4.8/sec	4.85	0.00	1030.5
http://dda.espoc...	500	4629	24	101350	13014.36	0.00%	4.4/sec	3.37	0.00	792.2
http://dda.espoc...	1000	1773	23	35623	3947.43	0.00%	8.6/sec	6.73	0.00	800.1
	1500	0	0	0	0.00	100.00%	0/hour	0.00	0.00	1438.0
http://dda.espoc...	5000	1116	19	28922	1962.32	0.00%	43.6/sec	40.94	0.00	962.0
http://dda.espoc...	500	975	23	20522	1840.38	0.00%	4.6/sec	3.95	0.00	880.0
http://dda.espoc...	1500	1136	11	29106	2103.46	0.00%	13.8/sec	12.12	0.00	901.4
http://dda.espoc...	1500	1320	11	37996	2543.31	0.00%	13.9/sec	11.99	0.00	881.4
http://dda.espoc...	1500	1116	17	19594	1639.22	0.00%	13.8/sec	11.65	0.00	862.0
http://dda.espoc...	1000	1327	23	33337	2318.26	0.00%	9.3/sec	7.73	0.00	852.0
TOTAL	14500	1794	0	103178	6277.13	10.63%	140.5/sec	131.92	0.00	961.2

Figura 5-3: Escenario de 500 usuarios y ejecución de 2 a 1 instancia
Realizado por: Zaruma Jorge. 2019

Caso 2: En la **Figura 6-3** se evaluó 2000 usuarios con respuesta de 1 segundo por cada usuario, en una instancia activa



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
http://dda.espoc...	1999	20757	241	180099	14818.76	61.83%	11.1/sec	18.33	0.00	1694.4
http://dda.espoc...	1999	18185	27	191914	12693.94	63.13%	9.5/sec	16.83	0.00	1808.4
http://dda.espoc...	3998	15354	5	262297	18539.54	39.79%	13.5/sec	18.92	0.00	1437.8
	5997	0	0	0	0.00	100.00%	0/hour	0.00	0.00	1438.0
http://dda.espoc...	19990	1564	4	193174	6347.18	1.19%	66.3/sec	63.56	0.00	982.2
http://dda.espoc...	1999	2186	7	101212	7103.86	3.05%	6.8/sec	6.20	0.00	939.4
http://dda.espoc...	5997	1313	6	130247	5295.21	1.25%	20.3/sec	18.33	0.00	925.2
http://dda.espoc...	5997	907	5	142894	4652.61	0.33%	20.5/sec	17.82	0.00	888.5
http://dda.espoc...	5997	864	5	169575	4558.48	0.17%	22.5/sec	19.01	0.00	865.2
http://dda.espoc...	3998	905	4	152457	4529.18	0.20%	15.0/sec	12.58	0.00	856.1
TOTAL	57971	3398	0	262297	9671.99	18.11%	221.0/sec	232.23	0.00	1076.9

Figura 6-3: Escenario de 2000 usuarios y una instancia activa
Realizado por: Zaruma Jorge. 2019

Resultado de Escenarios

A continuación, en la **Tabla 1-3** se presenta un resumen de datos resultantes luego de haber ejecutado los escenarios de prueba.

Tabla 1-3: Resultados de escenarios ejecutados con jmeter

	Escenario 1 (2 instancias)		**Escenario 2 (1 instancia)**	
	Error %	**Media (solicitudes)**	**Error %**	**Media (solicitudes)**
Caso 1 (500)	0,36	190	0,63	1794
Caso 2 (2000)	10.91	360	18,11	3398

Realizado por: Zaruma Jorge. 2019

Una vez concluido las pruebas con la herramienta Jmeter se procede a interpretar los resultados presentes en la Tabla 1-3 en base a gráficos como se aprecia a continuación.

Métrica de madurez:

Para satisfacer la subcaracterística de madurez se efectuó la prueba en base a que cada uno de los componentes de la arquitectura tenga tareas específicas y el número máximo de usuarios sea el permitido en este caso 500, dando como resultado que un 0.36% de las peticiones no fueron procesadas como se observa en el **Grafico 1-3**.

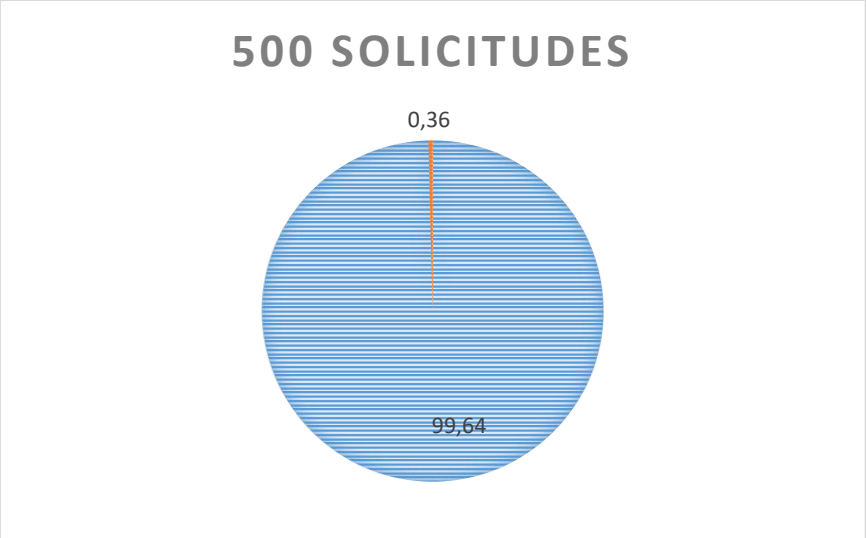


Gráfico 1-3: Diagrama del caso 1 correspondiente al escenario 1
Realizado por: Zaruma Jorge. 2019

Métrica de disponibilidad:

En condiciones normales las solicitudes se procesan con un porcentaje mínimo de error como se observa en el **Grafico 1-3**, sin embargo para el análisis de rendimiento se realizó 2 casos de prueba con 2000 usuarios, con una y dos instancias activas, dando como resultado que mientras más instancias se encuentran trabajando en conjunto en el servidor de aplicaciones el porcentaje de error disminuirá, además para que se pueda procesar más usuarios de forma simultanea es necesario aumentar más recursos hardware en el servidor web para así mantener un sistema disponible indiferente del número de usuarios que accedan al mismo tiempo.

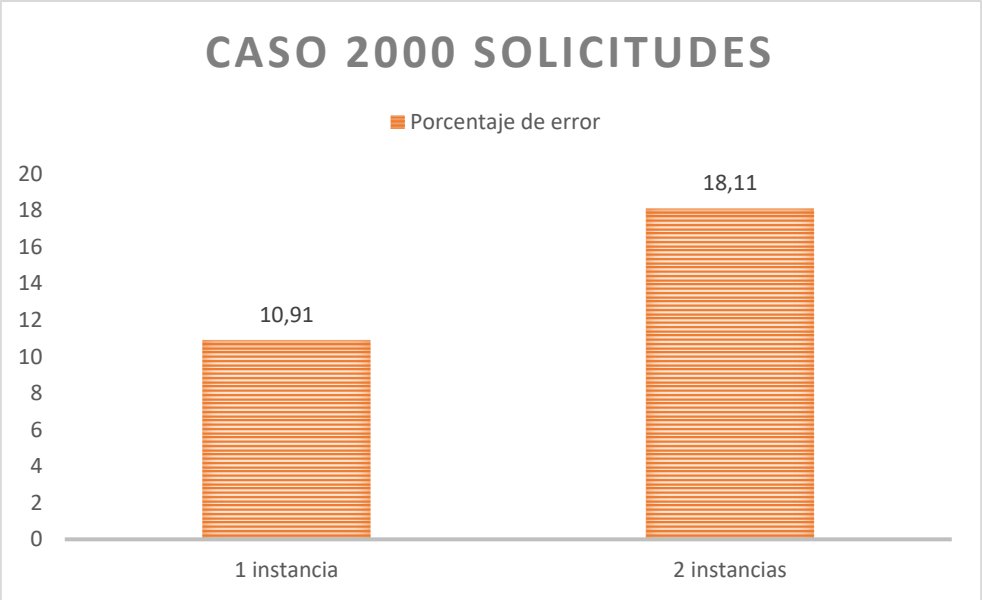


Gráfico 2-3: Diagrama de los casos de pruebas con 200 solicitudes
Realizado por: Zaruma Jorge. 2019

Métrica de tolerancia a fallas y capacidad de recuperación:

Por último en el **Grafico 3-3** se observa que al procesar peticiones solo con el número máximo de usuarios permitidos el error no varía en gran medida al utilizar una o dos instancias del servidor de aplicaciones verificando así que la arquitectura implantada es tolerante a fallos y gracias a la administración remota y centralizada de las instancias se puede reestablecer las operaciones de una instancia inactiva con esto se puede decir que la arquitectura posee capacidad de recuperación de una manera rápida.

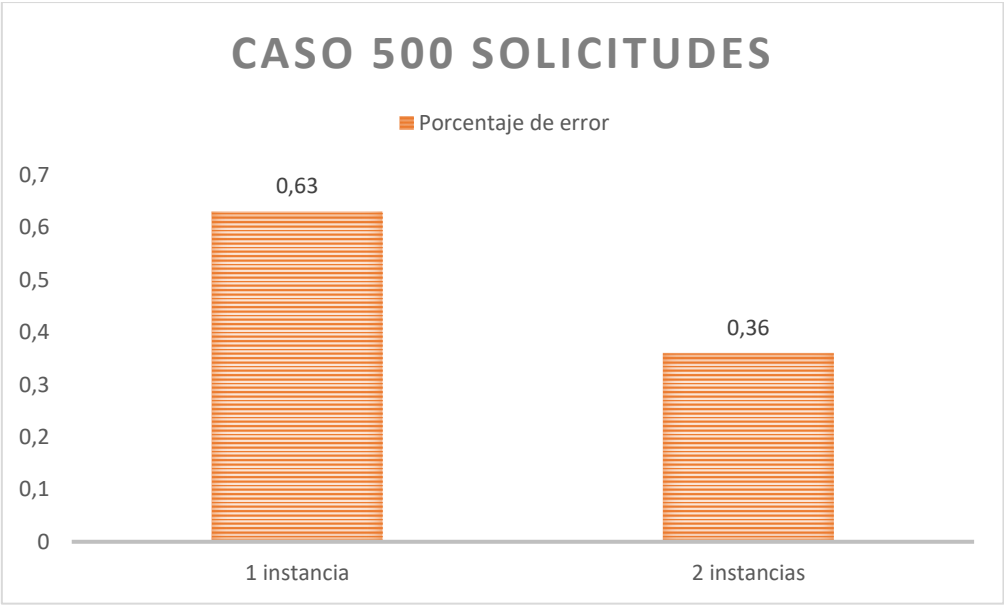


Gráfico 3-3: Diagrama para caso 1 del escenario 1 y del caso 1 del escenario 2
Realizado por: Zaruma Jorge. 2019

CONCLUSIONES

- Las fallas comunes que pueden afectar al rendimiento de servidores son las siguientes: especificaciones incorrectas, errores de implementación, componentes defectuosos y perturbaciones externas. Por tal motivo es necesario realizar estudios minuciosos de todos los componentes tanto de hardware como software que permitan verificar su correcto funcionamiento.

- Se tomó en cuenta dos trabajos relacionados con la tolerancia a fallos, una implantada para servicios en la nube y la otra orientada a servidores físicos, las dos con el único fin de evitar que sus sistemas queden inactivos para los usuarios finales, para lograr tal fin utilizaron elementos como clúster de servidores, balanceadores de carga, replicación de datos, etc. Lo que les permitió obtener arquitecturas escalables y de alta disponibilidad, las misma que fueron evaluadas en base a escenarios para validar su correcto funcionamiento.

- Teniendo en cuenta los trabajos relacionados al tema de tolerancia a fallos se aplicó una arquitectura para los servidores de la DDA, que consta de un clúster del servidor web apache en modo activo-pasivo en dos servidores, donde tienen el mismo archivo de configuración del balanceador de carga, esto se logró mediante las herramientas pacemarker y corosync. Además, se generó un clúster de payara conformado por tres servidores donde se desplegarán las aplicaciones web, Por ultimo gracias a la extensión citus de Postgresql se implantó un clúster para base de datos con el fin de distribuir el trabajo de lectura y escritura.

- Con el fin de evaluar la fiabilidad de la arquitectura implantada según la norma ISO 25010, se realizó 2 escenarios gracias a la herramienta jmeter y badboy con lo cual se procesó la solicitud de 500 usuarios con respuesta de 1 segundo por cada uno, verificando así que en condiciones del máximo de usuarios esperados la arquitectura funciona correctamente ya que solo hubo un 0,36% de error, sin embargo para el análisis de rendimiento se realizó 2 casos de prueba con 2000 usuarios, con una y dos instancias activas, dando como resultado que mientras más instancias se encuentran trabajando en conjunto en el servidor de aplicaciones el porcentaje de error disminuirá, además para que se pueda procesar más usuarios de forma simultanea es necesario aumentar más

recursos hardware en el servidor web para así mantener un sistema disponible indiferente del número de usuarios que accedan al mismo tiempo.

- En condiciones normales la arquitectura responde adecuadamente a las peticiones solo con un mínimo de error de 0,36% de 500 solicitudes, pero al procesar peticiones solo con el número máximo de usuarios permitidos el error no varía en gran medida al utilizar una o dos instancias del servidor de aplicaciones verificando así que la arquitectura implantada es tolerante a fallos y gracias a la administración remota y centralizada de las instancias se puede reestablecer las operaciones de una instancia inactiva con esto se puede decir que la arquitectura posee capacidad de recuperación de una manera rápida.

RECOMENDACIONES

- Se sugiere investigar sobre herramientas que permitan obtener alta disponibilidad en base de datos para contribuir a que la arquitectura propuesta no presente perdida de información de los usuarios en caso de que el nodo coordinador sufra algún tipo de inactividad.

- Para trabajos relacionados con este tema se recomienda realizar diferentes escenarios de pruebas para determinar si cada componente de un clúster desempeña la función que se le asigna.

- Es importante tener en cuenta que los componentes software que vaya a utilizar para replicar esta arquitectura tolerante a fallos debe ser de una versión estable mas no utilizar la última versión ya que puede presentar fallas y afectaría al rendimiento de su arquitectura y también deben poseer características como: documentación variada y soporte técnico.

- Para obtener más información acerca de la configuración realizada, de las características de cada servidor y los componentes software que se utilizaron a nivel de servidor de aplicaciones, web y base de datos se recomienda revisar los manuales de cada uno de ellos.

BIBLIOGRAFÍA

Afyouni, I., Ray, C., Ilarri, S., Claramunt, C. "A PostgreSQL extension for continuous path and range queries in indoor mobile environments". *Pervasive Mob. Comput* (2014).

Alejandro Frechina. *Proceso Scrum* [en línea]. [Consulta: 09 de julio de 2018]. Disponible en:https://winred.com/uploads/contenidos_usrs/originales/2566824_013_scrum_process_20180618180200.jpg

Baş Seyyar, M., Çatak, F.Ö., Gül, E. "Detection of attack-targeted scans from the Apache HTTP Server access logs". *Appl. Comput. Inform*. (2018)

Bello, C.L. "Sistemas Distribuidos Algunos esquemas de esta presentación I.T.I. Sistemas". *Instructor's Guide for Coulouris*. (2000)

Blancarte, O. *Escalabilidad Horizontal y Vertical* [blog]. [Consulta: 7 septiembre 2018]. Disponible en: https://www.oscarblancarteblog.com/2017/03/07/escalabilidad-horizontal-y-vertical/

Bustos, A. "Configuración de un clúster de alta disponibilidad y balanceo de carga en linux para satisfacer gran demanda web y servicios de resolución de nombres" [En línea] (Tesis). Escuela Politécnica Nacional, Quito, Ecuador. 2007. [Consulta:2018-10-01]. Disponible en: http://bibdigital.epn.edu.ec/bitstream/15000/276/1/CD-0691.pdf

Cáceres, G. "Estrategia de Implementación de un Clúster de Alta Disponibilidad de N nodos sobre Linux usando Software Libre" [En línea] (Tesis). Universidad San Francisco de Quito, Quito, Ecuador. 2012. [Consulta: 2019-01-30]. Disponible en: http://repositorio.usfq.edu.ec/bitstream/23000/1943/1/104087.pdf

Calmels, L. "Escalabilidad y Disponibilidad en Infraestructuras de Servicios Orientadas a la Web, basadas en Cloud Computing" [En línea] (Tesis). Universidad Nacional de La Pampa, La Pampa, Argentina. 2004. [Consulta: 2018-09-16]. Disponible en: http://www.biblioteca.unlpam.edu.ar/rdata/tesis/i_calesc252.pdf

Camps, R., **Casillas, L.** 2005. Bases de datos [En línea]. Barcelona, España: 2005. [Consulta: 12 septiembre 2018]. Disponible en: https://www.uoc.edu/masters/oficiales/img/913.pdf

Carballeira, F.G., Mateos, A.C. "Diseño de Sistemas Distribuidos Tolerancia a fallos". *Grupo de Arquitectura de Computadores.* (2017)

What is Citus? [en línea]. Las aplicaciones multiusuario de rápido crecimiento desean agregar nuevos clientes. [Consulta: 12 de septiembre de 2018]. Disponible en: https://docs.citusdata.com/en/v7.5/get_started/what_is_citus.html

Pacemaker [en línea]. Gestor de recursos de código abierto y alta disponibilidad adecuado para clústeres pequeños y grandes. [Consulta: 12 de septiembre de 2018]. Disponible en: https://clusterlabs.org/pacemaker/

Conscious IT [en línea]. Escalamiento vertical [Consulta: 7 de septiembre de 2018]. Disponible en: https://ralphavalon.files.wordpress.com/2016/03/scaling.png

Corosync [en línea]. Sistema de comunicación grupal con características adicionales para implementar una alta disponibilidad dentro de las aplicaciones. [Consulta: 03 de diciembre de 2018]. Disponible en: https://corosync.github.io/corosync/

de Paula, C.P., Visnadi, L.B., de Castro, H.F. "Multi-objective optimization in redundant system considering load sharing ". *ScienceDirect* (2019).

de Souza, J.R. "FTDR: Tolerancia a fallos, en clúster de computadoras geográficamente distribuidos, basado en replicación de datos". [En línea] (Tesis). Universidad Autónoma de Barcelona, Barcelona, España. 2006. [Consulta: 2018-09-16]. Disponible en: https://www.tesisenred.net/bitstream/handle/10803/5759/jrs1de1.pdf?sequence=1&isAllowed=y

Díaz, J. "Servidores de Aplicaciones en Linux". *Escuela de Administración Publica*. (2010).

DTIC, *Dirección de Tecnologías de la Información y Comunicación* [en línea]. Proporcionar servicios integrales de calidad en las áreas de desarrollo organizacional y sistemas de información a la ESPOCH y entidades externas. [Consulta: 20 septiembre 2018]. Disponible en: https://www.espoch.edu.ec/index.php/direccion-de-tecnologias-de-la-informacion-y-comunicacion.html

ESPOCH. *Escuela Superior Politécnica Chimborazo* [en línea]. Institución de educación superior. [Consulta: 13 septiembre 2018]. Disponible en: https://www.espoch.edu.ec/index.php/espoch.html

Fan, L., Tiejun, W., Liuyi, W. "Research and Implementation on Model for High Availability of Enterprise Information System". *IERI Procedia* [en línea], 2012, (China), pp 181–185. [Consulta: 10 octubre 2018]. Disponible en: https://www.sciencedirect.com/science/article/pii/S2212667812002390?via%3Dihub

Febrero, F., Calero, C., Ángeles Moraga, M. "Software reliability modeling based on ISO/IEC SQuaRE". *ScienceDirect* [en línea], 2016, (Spain). [Consulta: 10 octubre 2018]. Disponible en: https://www.sciencedirect.com/science/article/abs/pii/S0950584915001652

García, F. *Seguridad y Alta Disponibilidad* [en línea]. Implantación de soluciones de Alta Disponibilidad. [Consulta: 07 septiembre 2018]. Disponible en: https://mgarciafelipe.files.wordpress.com/2012/03/ud-6-implantacic3b3n-de-soluciones-de-alta-disponibilidad-miguelangelgarcia.pdf

Groussard, T., 2010. *Java Enterprise Edition: Desarrollo de aplicaciones web con JEE 6*. Barcelona-España: Ediciones ENI, 2010, pp 20-21.

Hadi, M. *Virtual Host (Web Server)* [en linea]. Virtual Networking implementation. [Consulta: 12 diciembre 2018]. Disponible en: http://zenhadi.lecturer.pens.ac.id/kuliah/Jarkom2/Modul%203%20Virtual%20Host.pdf

IBM. *IBM Knowledge Center* [en linea]. Ventajas de la alta disponibilidad. [Consulta: 07 septiembre 2018]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_72/rzarj/rzarjbenefitsha.htm

Infosegur. *Seguridad Informática* [en línea]. Clúster de servidores. [Consulta: 07 septiembre 2018]. Disponible en: https://infosegur.wordpress.com/tag/cluster/

Irey-Núñez, J. 2014. "Alternativas para la escalabilidad de aplicaciones en plataformas web de alta concurrencia". *Ulima* (2014)

ISO25000. *ISO 25010* [en línea]. ISO 25000 calidad de productos software. [Consulta: 24 septiembre 2018]. Disponible en: https://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&start=3

Krzywda, J., Ali-Eldin, A., Carlson, T.E., Östberg, P.-O., Elmroth, E. "Power-performance tradeoffs in data center servers: DVFS, CPU pinning, horizontal, and vertical scaling". *Future Gener. Comput* [en linea], 2018, (Singapore). [Consulta: 10 octubre 2018]. Disponible en: http://www.sciencedirect.com/science/article/pii/S0167739X17304910

López, M. "Fiabilidad y Tolerancia a Fallos en Sistemas en Tiempo Real". (2006)

LOTAIP. *Ley Orgánica de Transparencia y Acceso a la Información Pública* [en línea]. Entidad técnica de supervisión y control de las organizaciones. [Consulta: 01 octubre 2018]. Disponible en: http://www.seps.gob.ec/ley-de-transparencia?lotaip

Mahalakshmi, M., Sundararajan, D.M. *Traditional SDLC Vs Scrum Methodology – A Comparative Study*. Certified Journal, 2013 pp 192-193

Miguez, R. "Desarrollo de una infraestructura de redundancia para servidores Proxy GNU/LINUX en la intranet de la Facultad de Ciencias". [En línea](Tesis). Escuela Superior

Politécnica de Chimborazo, Riobamba, Ecuador. 2012. [Consulta: 2018-10-01]. Disponible en: http://dspace.espoch.edu.ec/bitstream/123456789/4053/1/20T00444.pdf

Nesmachnow, S., Iturriaga, S. *Computación de Alta Performance*. [en línea]. Tolerancia a Fallos [Consulta: 06 septiembre 2018]. Disponible en: https://www.fing.edu.uy/inco/cursos/hpc/material/clases/Tema12-2011.pdf

Paredes, J.P. *Alta disponibilidad para Linux* [en línea]. Principios básicos de la alta disponibilidad. [Consulta: 16 septiembre 2018]. Disponible en: http://www.othlo.com/htecnologia/documentacion/hispalinux04/11disponibilidad.pdf

Payara. *Servidor Payara: robusto. De confianza. Soportado*. [en línea]. Servidor de Aplicaciones para proyectos java. [Consulta: 16 septiembre 2018]. Disponible en: https://www.payara.fish/about

Pons, S., Bonet, E. *Servicios de alta disponibilidad*. [en línea]. Capacidad de ofrecer un conjunto de servicios sobre un sistema informático que es capaz de recuperarse ante fallos. [Consulta: 03 diciembre 2018]. Disponible en: http://informatica.uv.es/it3guia/AGR/apuntes/teoria/documentos/Corosync.pdf

Scrum. *Scrum Guide*. [en línea]. Utilizado para desarrollar, entregar y mantener productos complejos. [Consulta: 03 diciembre 2018]. Disponible en: https://www.scrumguides.org/scrum-guide.html

Sinisterra, M.M., Díaz Henao, T.M., Ruiz López, E.G. "Clúster de balanceo de carga y alta disponibilidad para servicios web y mail". *Revista Informador Técnico* (2012).

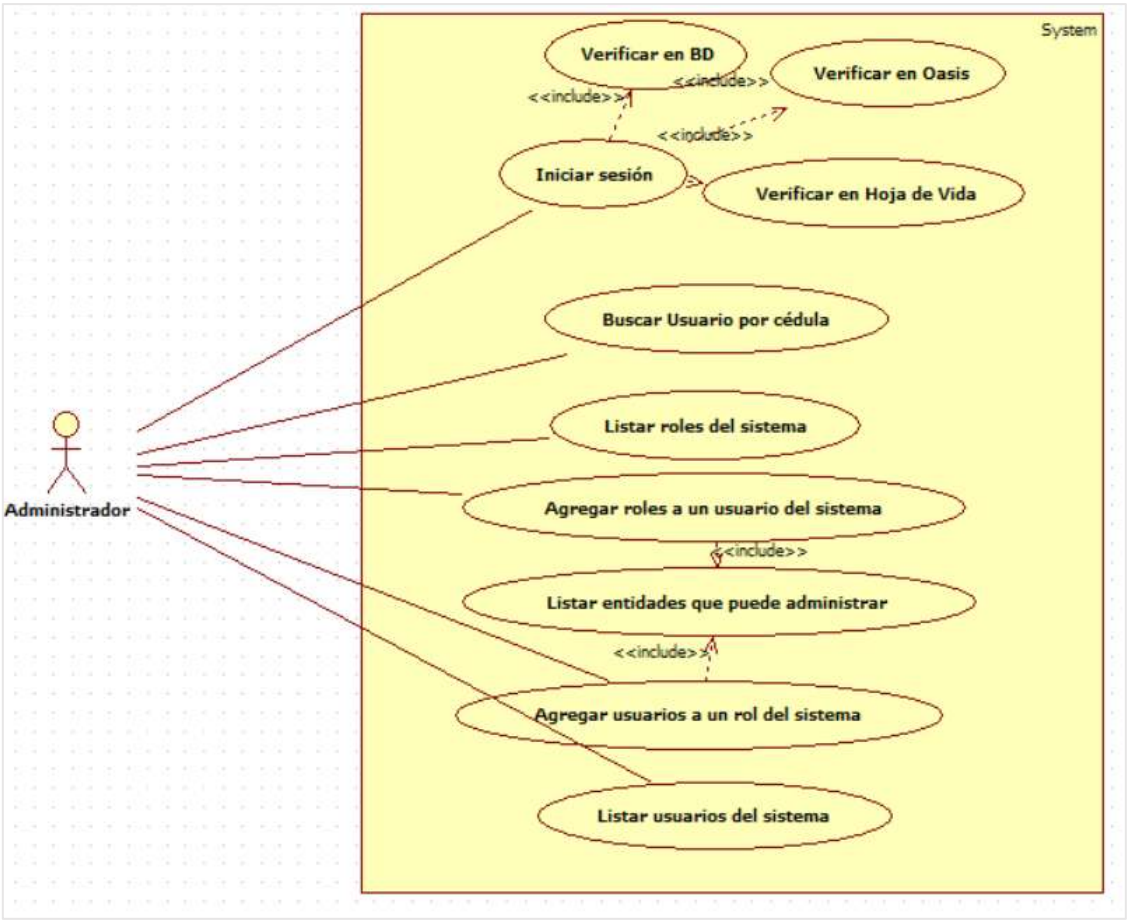
Vaca, T. *Calidad de software del módulo de talento humano del sistema informático de la Universidad Técnica del Norte bajo la norma ISO/IEC 25000*. [en línea]. Calidad de software [Consulta: 12 diciembre 2018]. Disponible en: https://www.researchgate.net/publication/325022337_Calidad_de_software_del_modulo_de_talento_humano_del_sistema_informatico_de_la_Universidad_Tecnica_del_Norte_bajo_la_norma_ISOIEC_25000

Zea Ordóñez, M.P., Molina Ríos, J.R., Redrován Castillo, F.F. *ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL*. Alicante, España: Editorial Científica 3Ciencias, 2017, pp 9-12

ANEXOS

ANEXO A: Diagramas de casos de uso

En la siguiente figura se visualiza las funcionalidades que se van agregar a los sistemas del DDA tal como: Sílabos, Programas Analíticos y Planificación con lo que permitirá al administrador la gestión de roles y usuarios en este sistema, estos módulos son de vital importancia porque permitirá agregar roles a un usuario o viceversa, que se encuentre o no en la base de datos, ya que por ejemplo los roles por defecto que se pueden extraer de los servicios web del oasis son: docente y director. Pero existen otros roles que se necesita que interactúe con el sistema como: coordinador de campo de formación, vicedecano, decano, director de la dirección de desarrollo académico y este último puede ser el administrador de los sistemas o a su vez delegar a otra persona para que desempeñe esta función. Es importante aclaras que el diagrama de caso de uso para los tres sistemas mencionados anteriormente es igual.



Realizado por: Zaruma Jorge; 2019

ANEXO A.2: Documentación de Casos de Uso

- **Configurar clúster de servidor de aplicaciones.**

Caso de Uso	**Configurar clúster de servidor de aplicaciones**	
Actores:	Administrador	
Precondición	Haber instalado el servidor de aplicaciones De poseer credenciales root de los nodos remotos.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la dirección http://172.17.103.117:4848
	2	Ingresar sus credenciales
	3	Ingresar a la ventana de nodos
	4	Agregar nodos haciendo referencia a los servidores remotos.
	5	Crear un nuevo archivo de configuración haciendo referencia al de configuración por defecto.
	6	Habilitar Hazeltcast
	7	Crear instancias de los nodos remotos y el nodo local creados en el paso 4
	8	Inicializar instancias
Post condición	Si da clic en agregar el usuario se le agregar a la base de datos	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Instalar servidor de base de datos.**

Caso de Uso	**Instalar servidor de base de datos**	
Actores:	Administrador	
Precondición	El administrador debe contar con las credenciales del servidor que se va a utilizar	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la herramienta de conexión remota ssh en este caso putty.
	2	Ingresar sus credenciales
	3	Agregar repositorios de citus
	4	Instalar postgresql con su extensión de citus

	5	Habilitamos en el archivo de configuración escuche todas las redes.
	6	Permitimos que redes se pueden conectar al servidor de base de datos
	7	Habilitar el puerto que se va a utilizar y agregar a la tabla del firewall del servidor.
	8	Se inicializa el servidor de base de datos
Post condición	Si al ejecutar el comando para verificar el estado del servidor este debe mostrar que este activo.	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Configurar clúster de base de datos.**

Caso de Uso	**Configurar clúster de base de datos**	
Actores:	Administrador	
Precondición	El servidor de base de datos se haya instalado en los nodos correspondientes y habilitado el correspondiente puerto de administración.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar al nodo principal mediante putty
	2	Ingresar sus credenciales
	3	Ejecutar el comando que permita la agregación de los nodos secundarios.
	4	Inicializar citus
	5	Verificar nodos activos en el clúster
Post condición	Al verificar los nodos activos se deben visualizar los mismos que se agregaron en el paso 3	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Instalar servidor web.**

Caso de Uso	**Instalar servidor web**
Actores:	Administrador

Precondición	El administrador debe contar con las credenciales que le permita ingresar a la consola de administración remota del servidor	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la herramienta de conexión remota ssh para el nodo en este caso putty.
	2	Ingresar sus credenciales
	3	Ejecutar el comando de instalación yum de apache
	4	Habilitar en el archivo de configuración mediante que puerto recibirá las peticiones por lo general es el 80
	5	Habilitar en la tabla de firewall el puerto de apache
	6	Habilitar para que apache se inicie cuando el servidor se encienda.
	7	Inicializar apache
Post condición	Al ingresar en el navegador la dirección del nodo con el puerto debe aparecer por defecto la página de apache.	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Configurar balanceador de carga en el servidor web.**

Caso de Uso	**Configurar balanceador de carga en el servidor web**	
Actores:	Administrador	
Precondición	El administrador debe poseer las credenciales del servidor y debe haberse configurado el clúster del servidor de aplicaciones.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar al nodo que se instaló el servidor web apache mediante putty.
	2	Ingresar sus credenciales
	3	Crear un archivo de configuración para agregar un host virtual
	4	Le especificamos el nombre del dominio que corresponde el host virtual mediante el serverName
	5	Especificamos que va a ser un balanceador
	6	Agregamos los nodos de los servidores donde se encuentran las aplicaciones desplegadas

	7	Agregamos la opción de route que específica que cada sesión de usuario va a estar ligada a una instancia del servidor de aplicaciones.
	8	Incluimos en el archivo de configuración principal la dirección donde se encuentre el archivo de configuración creado.
	9	Guardamos y reiniciamos el servidor web
Post condición	Al reiniciar el servidor web no debe dar problemas	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Configurar alta disponibilidad en el servidor web.**

Caso de Uso	**Configurar alta disponibilidad en el servidor web**	
Actores:	Administrador	
Precondición	El administrador poseer las credenciales de los nodos que se van a utilizar para este caso de uso El servidor de aplicaciones se encuentre instalado Poseer una ip apartada	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la administración remota de los servidores mediante putty
	2	Ingresar sus credenciales
	3	En ambos nodos involucrados se agrega en su archivo host la información de los nodos es decir su dirección ip y el nombre que considere pertinente.
	4	Instalar mediante yum las herramientas Corosync y pacemarker
	5	Autenticamos los nodos involucrados para agregarlos al clúster
	6	Creamos un recurso de ip virtual y le asignamos la dirección apartada.
	7	Creamos un recurso de servidor web y le asignamos el servidor web instalado que es apache
	8	En el nodo principal se le agrega que va a escuchar las peticiones por la ip virtual asignada.

	9	Verificamos el estado del clúster.
Post condición	Los nodos deben estar activos y los recursos agregados deben estar ejecutándose en alguno de los nodos del clúster.	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Desplegar aplicaciones.**

Caso de Uso	**Desplegar aplicaciones**	
Actores:	Administrador	
Precondición	Los servidores se encuentren configurados correctamente Poseer credenciales hacia los servidores de base de datos y de aplicaciones	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la consola de administración remota del servidor de aplicaciones
	2	Ingresar sus credenciales
	3	Dar clic en la ventana de aplicaciones
	4	Dar clic en desplegar
	5	Seleccionar la aplicación
	6	Agregar las instancias donde estará la aplicación
	7	Guardar
	8	Ingresar a Pgadmin para administrar las base de datos
	9	Ingresar credenciales
	10	Crear una nueva base de datos
	11	Crear la base de datos o restaurar la misma.
	12	Guardar
Post condición	Al ejecutar los sistemas deben ingresar a la pantalla principal	
Excepciones	**Paso**	**Acción**
	2 - 9	Si las credenciales no son correctas se termina el caso de uso

- **Buscar usuarios por cedula**

Caso de Uso	**Buscar usuario por cedula**
Actores:	Administrador
Precondición	El administrador debe haberse autenticado correctamente.

	La cedula debe corresponder a un usuario que se encuentre registrado en el OASIS, base de datos o el sistema de hoja de vida.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la página http://dda.espoch.edu.ec/[Aplicacion]
	2	Ingresar sus credenciales
	3	Dar clic en el icono de administración
	4	Dar clic en usuarios
	5	Dar clic en el botón nuevo usuario
	6	Ingresar la cedula en el cuadro de texto
	7	Se visualiza la información del usuario que corresponde la cedula ingresada
Post condición	Si da clic en agregar el usuario se le agregar a la base de datos	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Listar roles del sistema**

Caso de Uso	**Listar roles del sistema**	
Actores:	Administrador	
Precondición	El administrador debe haberse autenticado correctamente. La cedula debe corresponder a un usuario que se encuentre registrado en el OASIS, base de datos o el sistema de hoja de vida.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la página http://dda.espoch.edu.ec/[Aplicacion]
	2	Ingresar sus credenciales
	3	Dar clic en el icono de administración
	4	Dar clic en roles
Post condición	Se deben visualizar los roles definidos en el sistema	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Agregar roles a un usuario del sistema**

Caso de Uso	**Agregar roles a un usuario del sistema**	
Actores:	Administrador	
Precondición	El administrador debe haberse autenticado correctamente. La cedula debe corresponder a un usuario que se encuentre registrado en el OASIS, base de datos o el sistema de hoja de vida.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la página http://dda.espoch.edu.ec/[Aplicacion]
	2	Ingresar sus credenciales
	3	Dar clic en el icono de administración
	4	Dar clic en usuarios
	5	Dar clic en el icono de asignar roles
	6	Seleccionar el rol y la entidad donde quiere agregar al usuario
Post condición	El mensaje que se debe visualizar es el de haberse guardado la asignación correctamente	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

- **Listar usuarios del sistema**

Caso de Uso	**Listar usuarios del sistema**	
Actores:	Administrador	
Precondición	El administrador debe haberse autenticado correctamente. La cedula debe corresponder a un usuario que se encuentre registrado en el OASIS, base de datos o el sistema de hoja de vida.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la página http://dda.espoch.edu.ec/[Aplicacion]
	2	Ingresar sus credenciales
	3	Dar clic en el icono de administración
	4	Dar clic en usuarios
Post condición	Se deben listar los usuarios externos asignados en cada sistema	

Excepciones	Paso	Acción
	2	Si las credenciales no son correctas se termina el caso de uso

- **Agregar usuarios a un rol del sistema**

Caso de Uso	Agregar usuarios a un rol del sistema	
Actores:	Administrador	
Precondición	El administrador debe haberse autenticado correctamente. La cedula debe corresponder a un usuario que se encuentre registrado en el OASIS, base de datos o el sistema de hoja de vida.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la página http://dda.espoch.edu.ec/Silabo
	2	Ingresar sus credenciales
	3	Dar clic en el icono de administración
	4	Dar clic en roles
	5	Dar clic en el icono de agregar usuarios del rol
	6	Seleccionar un usuario de la entidad que corresponda.
Post condición	El mensaje que se debe visualizar es el de haberse guardado la asignación correctamente	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

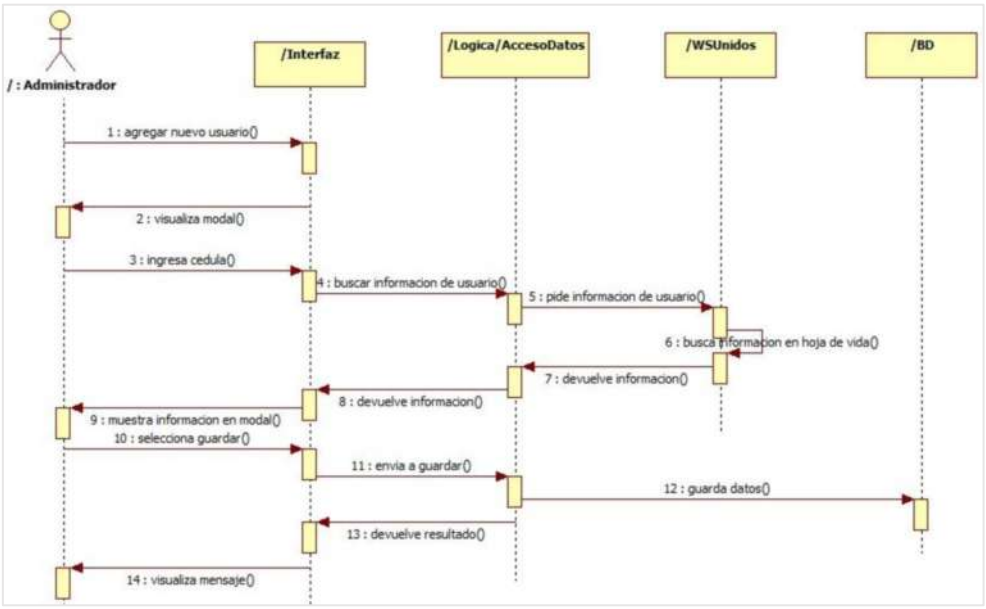
- **Listar entidades que puede administrar**

Caso de Uso	Listar entidades que puede administrar	
Actores:	Administrador	
Precondición	El administrador debe haberse autenticado correctamente. La cedula debe corresponder a un usuario que se encuentre registrado en el OASIS, base de datos o el sistema de hoja de vida.	
Secuencia normal	**Paso**	**Acción**
	1	Ingresar a la página http://dda.espoch.edu.ec/Silabo
	2	Ingresar sus credenciales
	3	Dar clic en el icono de administración
	4	Dar clic en usuarios o roles
	5	Dar clic en agregar usuarios o agregar roles

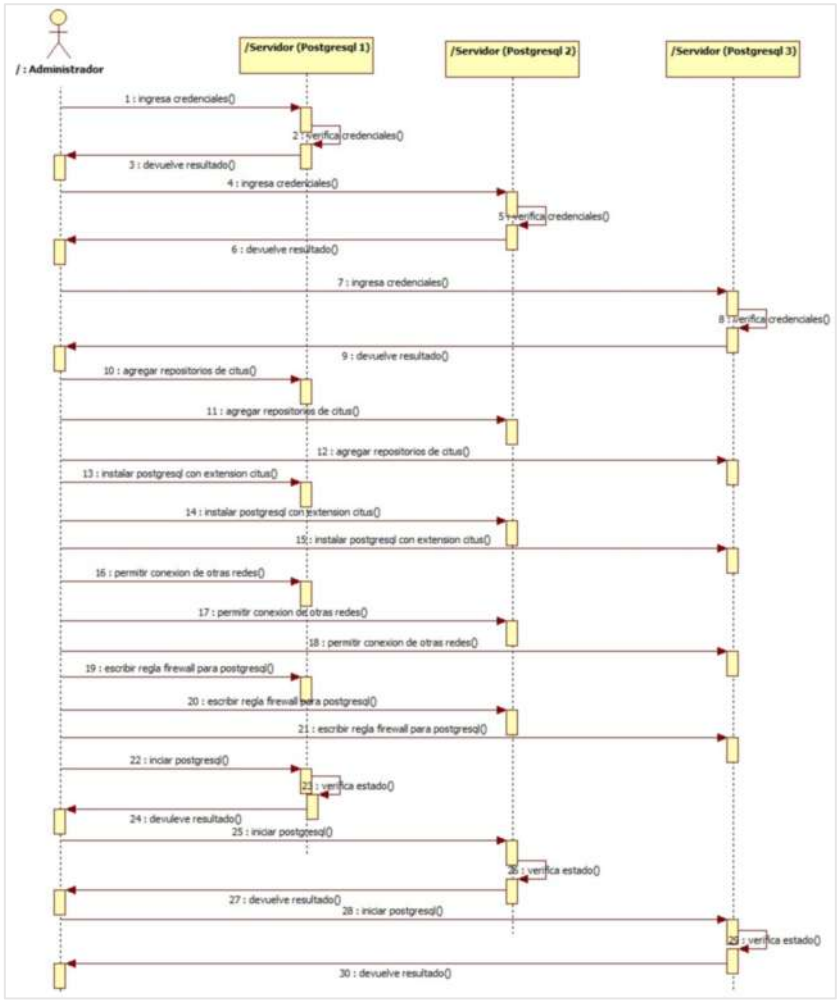
Post condición	Si deben listar las entidades que puede administrar.	
Excepciones	**Paso**	**Acción**
	2	Si las credenciales no son correctas se termina el caso de uso

ANEXO B: Diagramas de Secuencia

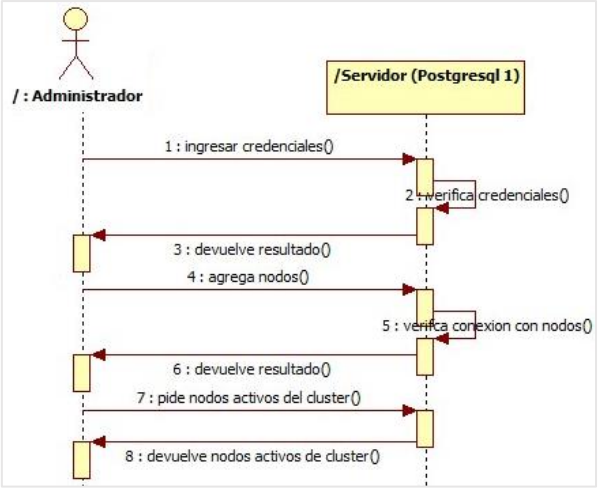
- **Buscar usuarios por cedula**



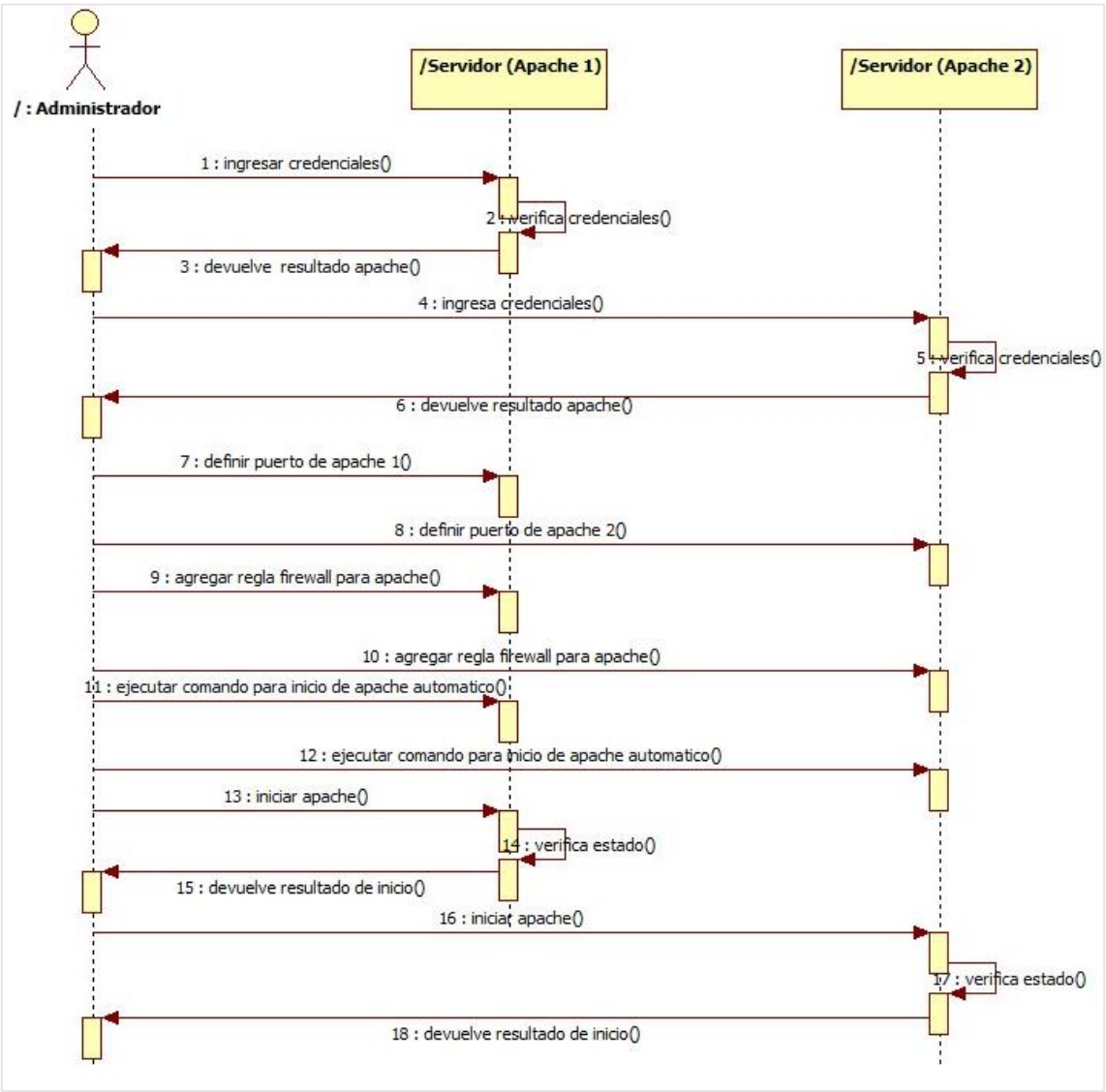
- **Instalar servidor de base de datos.**



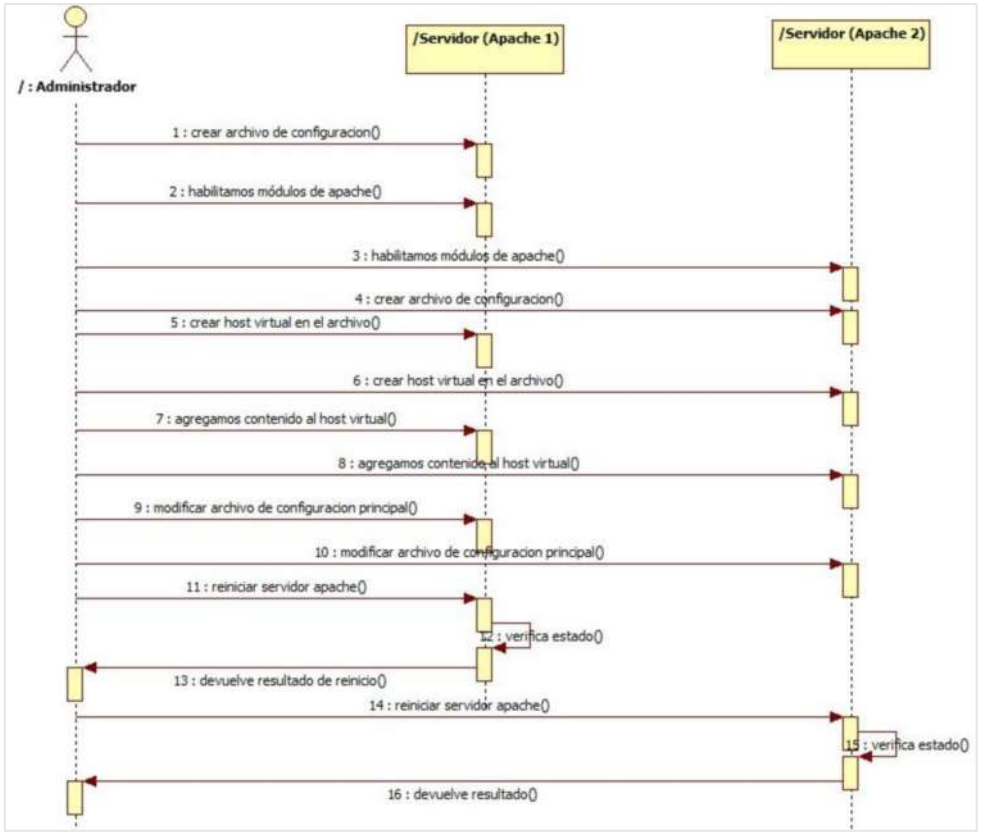
- **Configurar clúster de base de datos.**



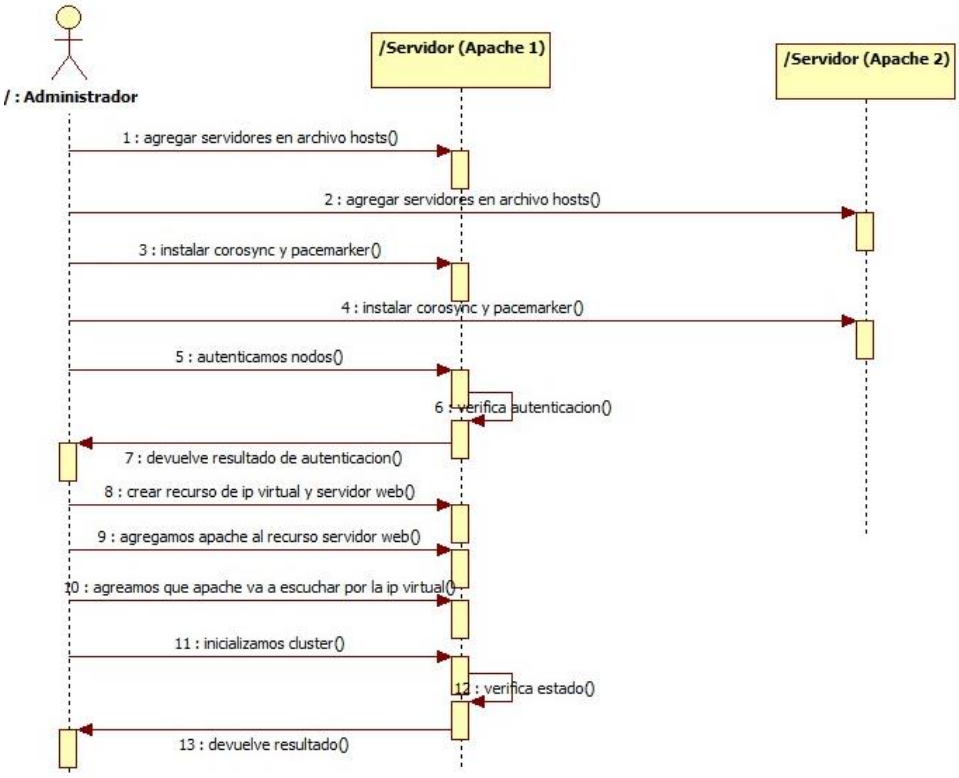
- **Instalar servidor web.**



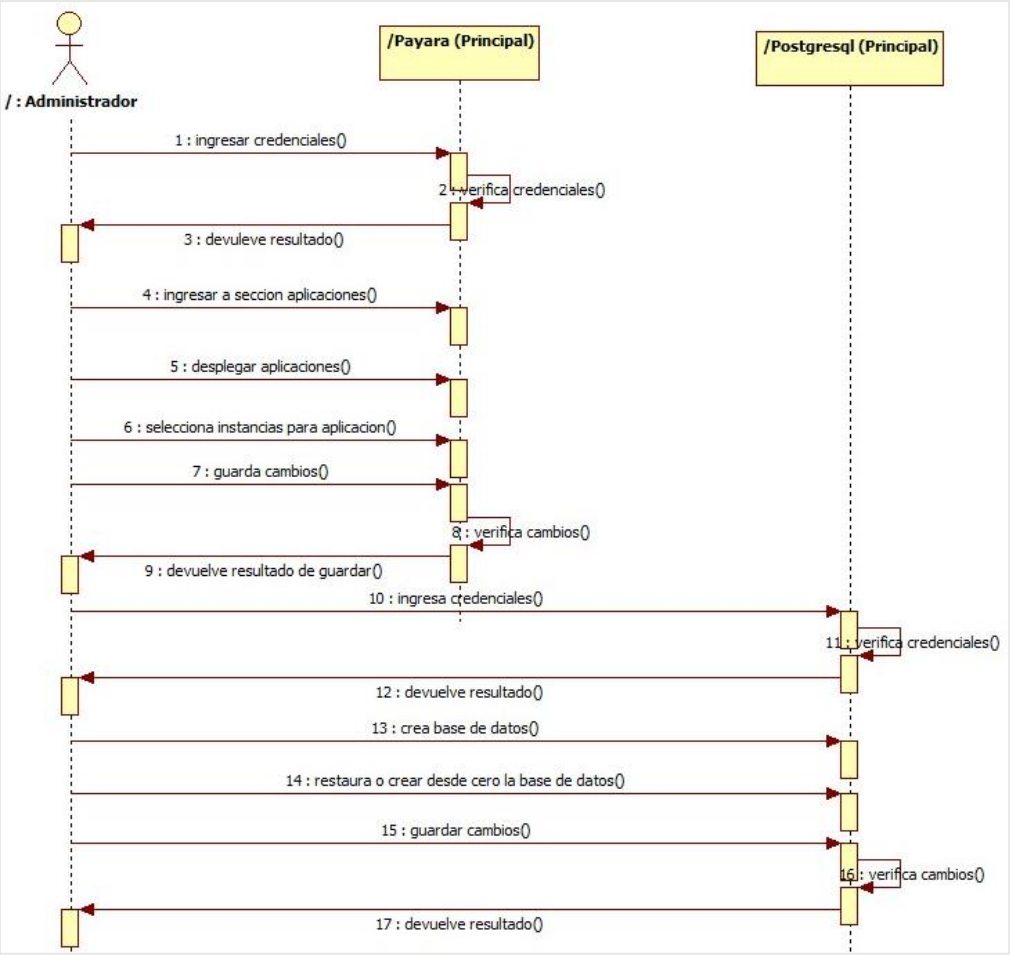
- **Configurar balanceador de carga en el servidor web.**



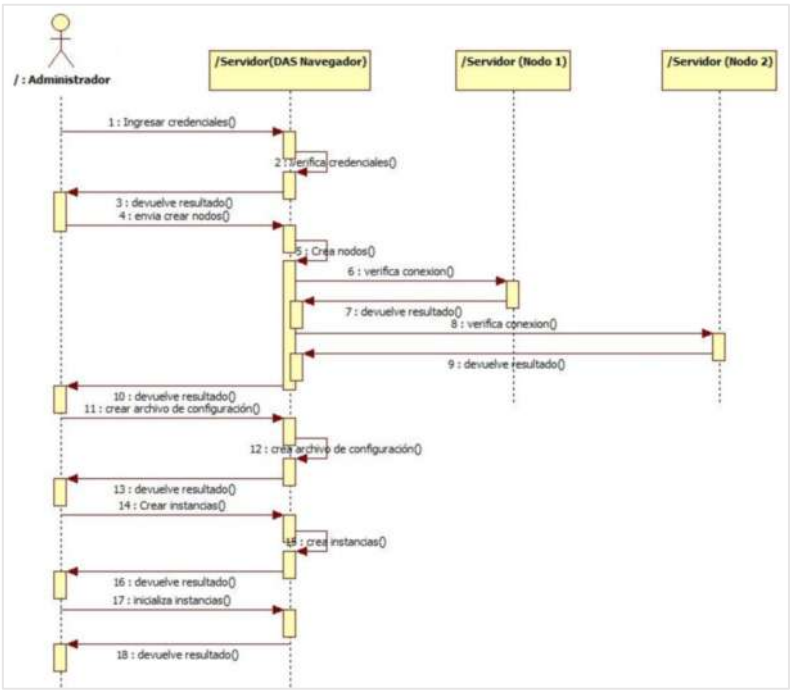
- **Configurar alta disponibilidad en el servidor web.**



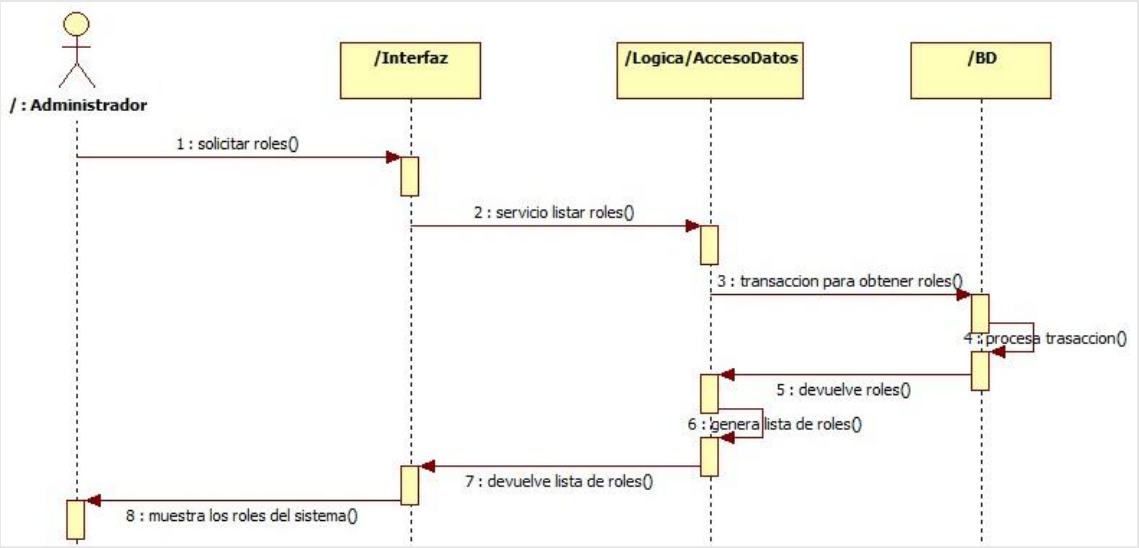
- **Desplegar aplicaciones.**



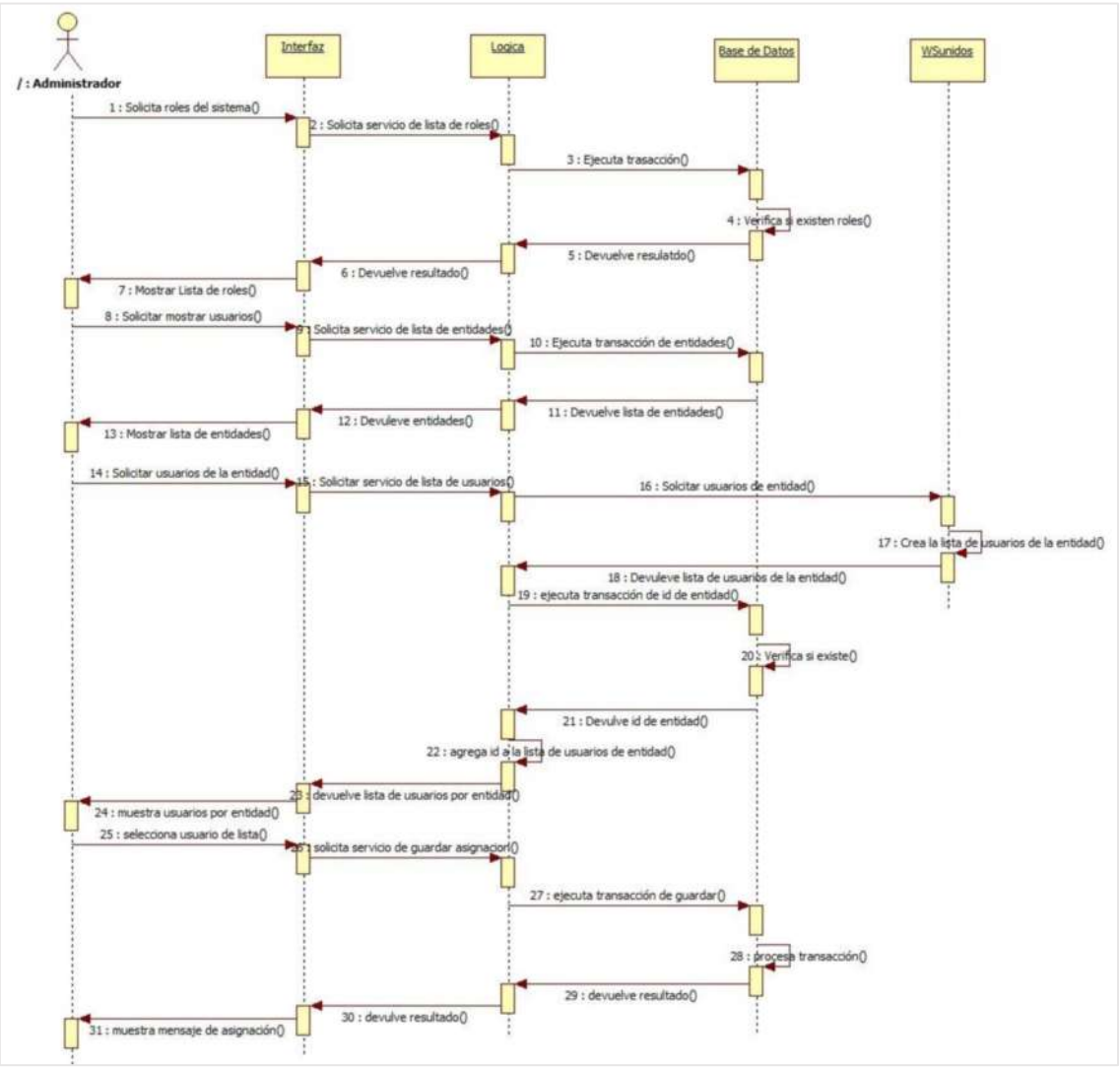
- **Configurar clúster de servidor de aplicaciones.**



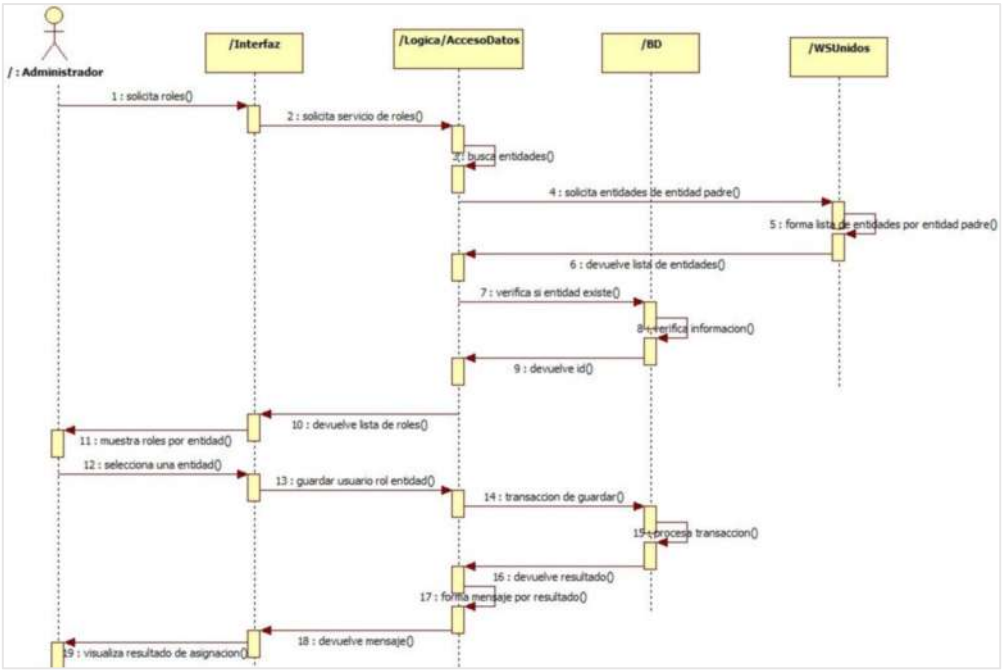
- **Listar roles del sistema**



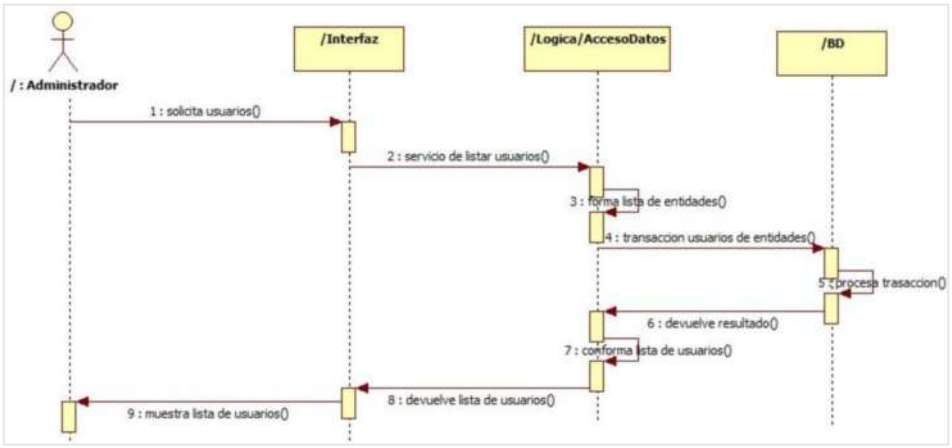
- **Agregar usuarios a un rol del sistema**



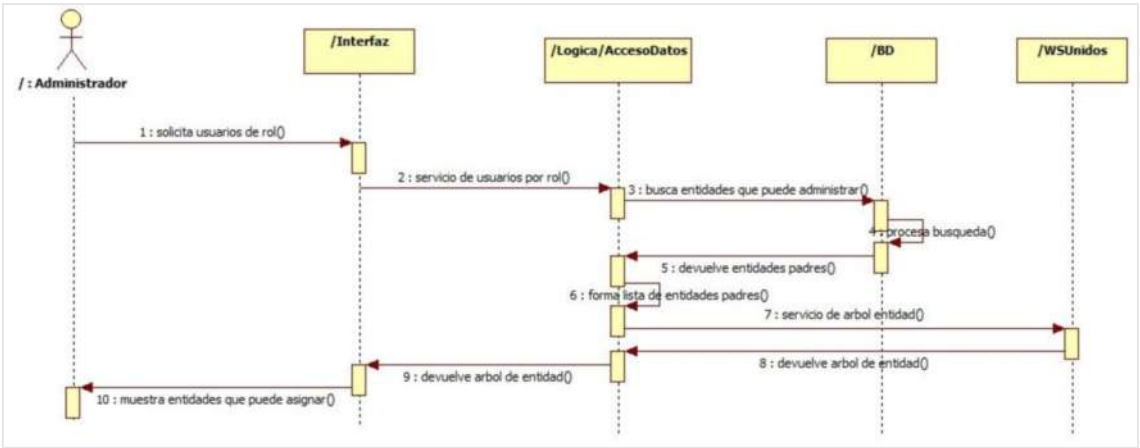
- **Agregar roles a un usuario del sistema**



- **Listar usuarios del sistema**

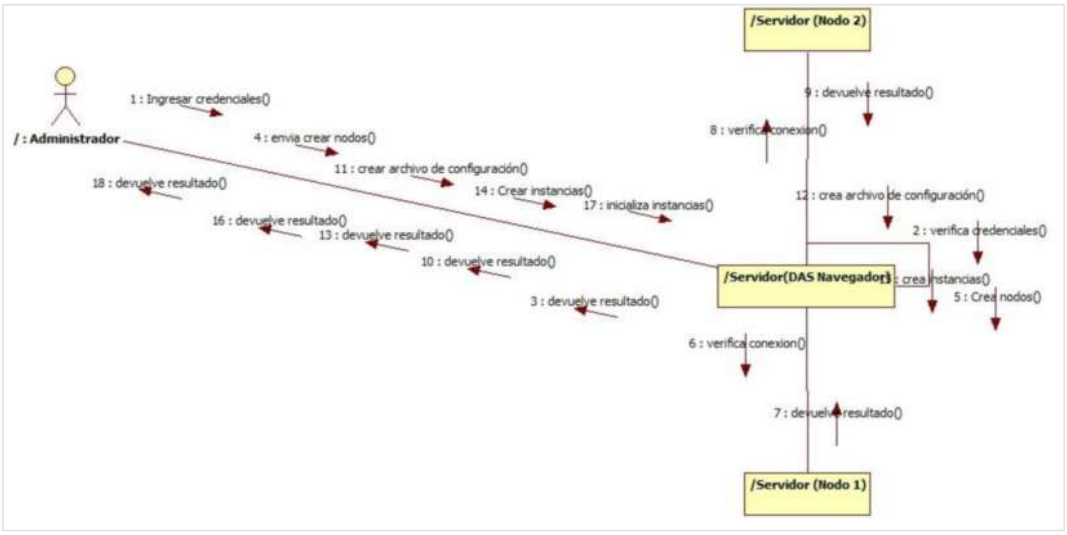


- **Listar entidades que puede administrar**

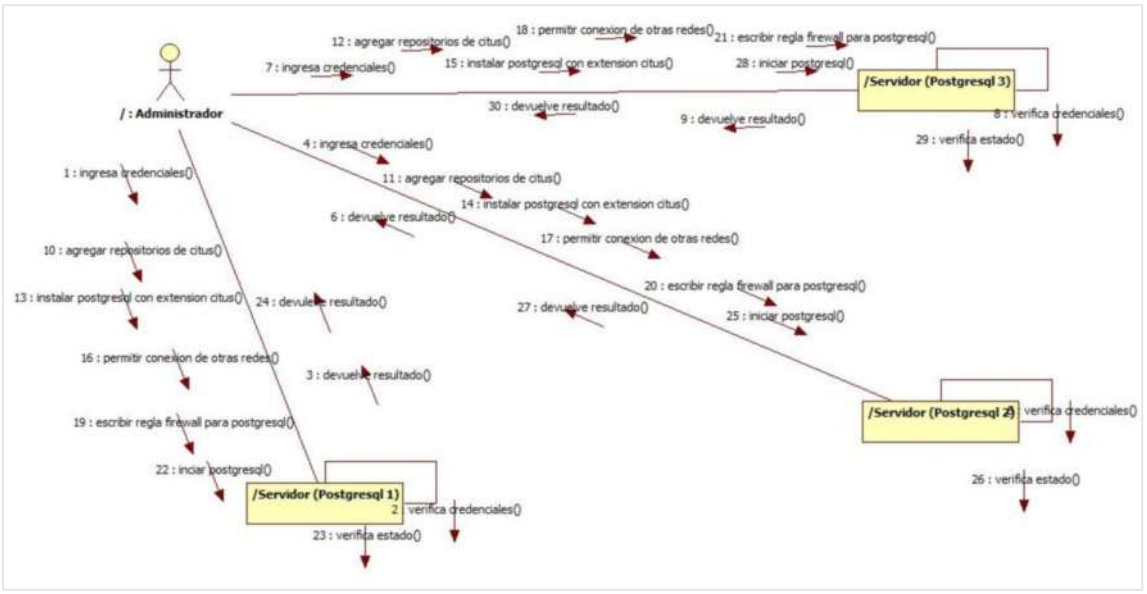


ANEXO C: Diagramas de Colaboración

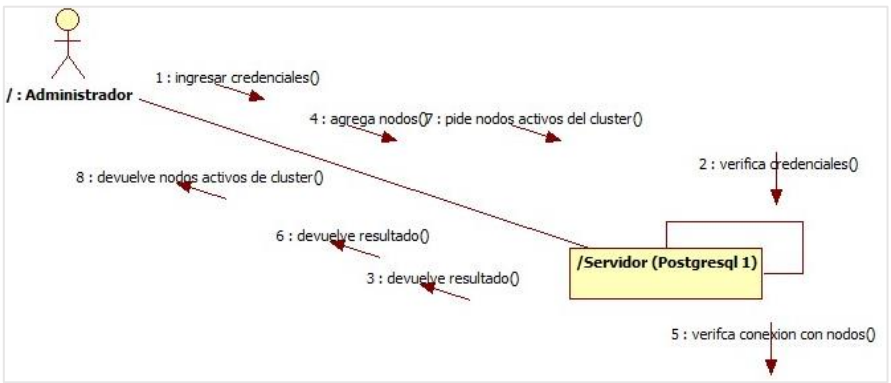
- **Configurar clúster de servidor de aplicaciones.**



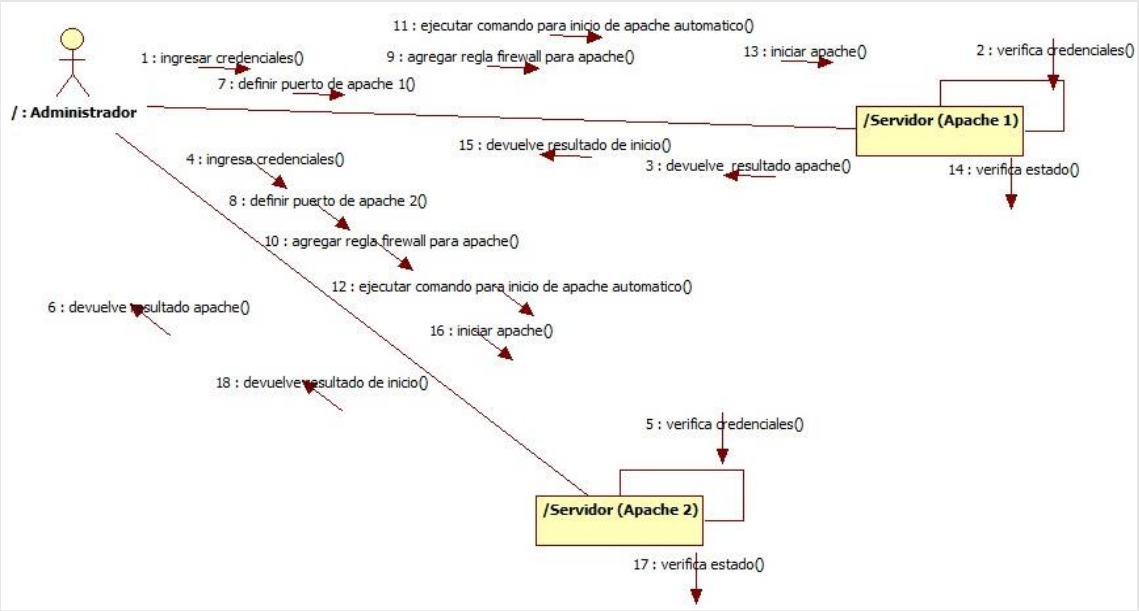
- **Instalar servidor de base de datos.**



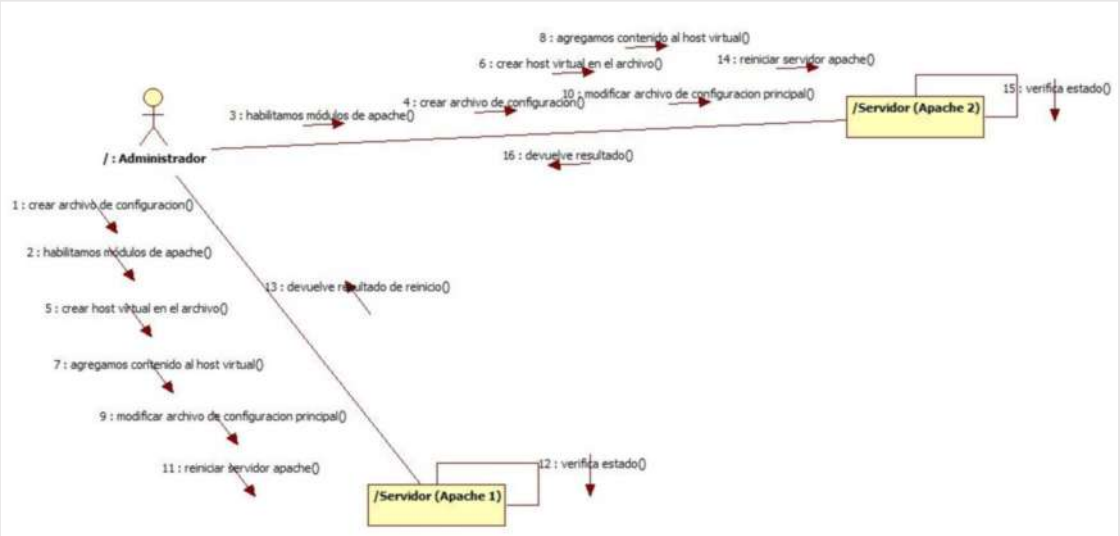
- **Configurar clúster de base de datos.**



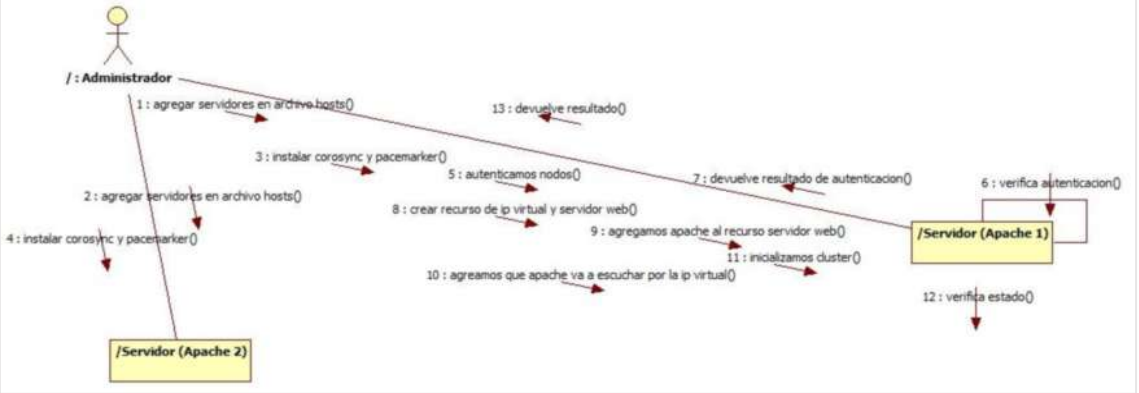
- **Instalar servidor web.**



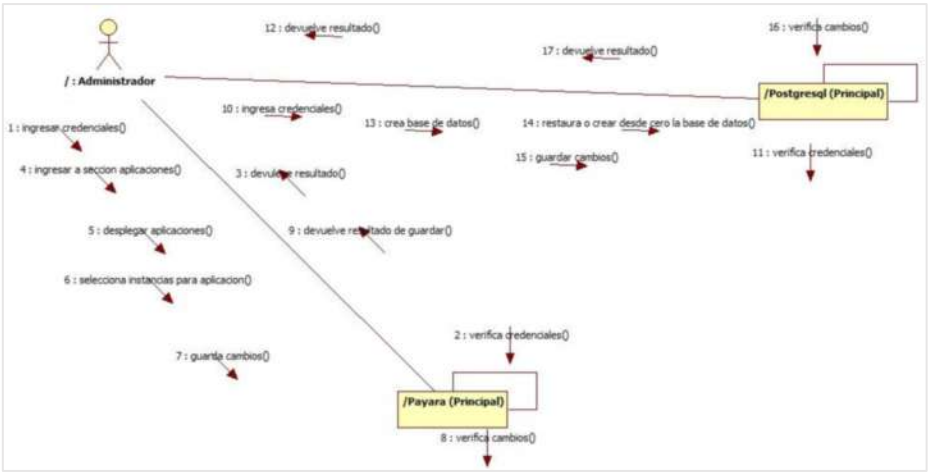
- **Configurar balanceador de carga en el servidor web.**



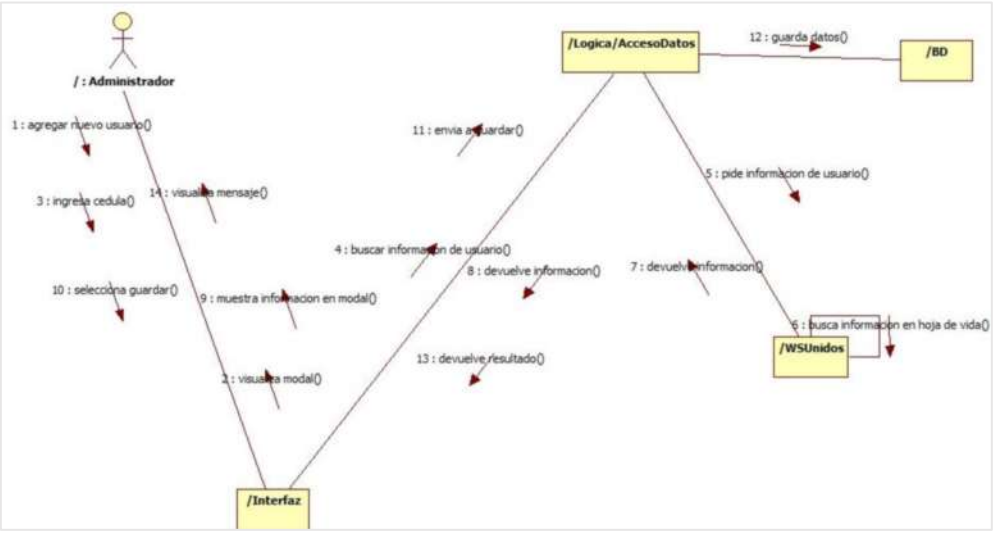
- **Configurar alta disponibilidad en el servidor web.**



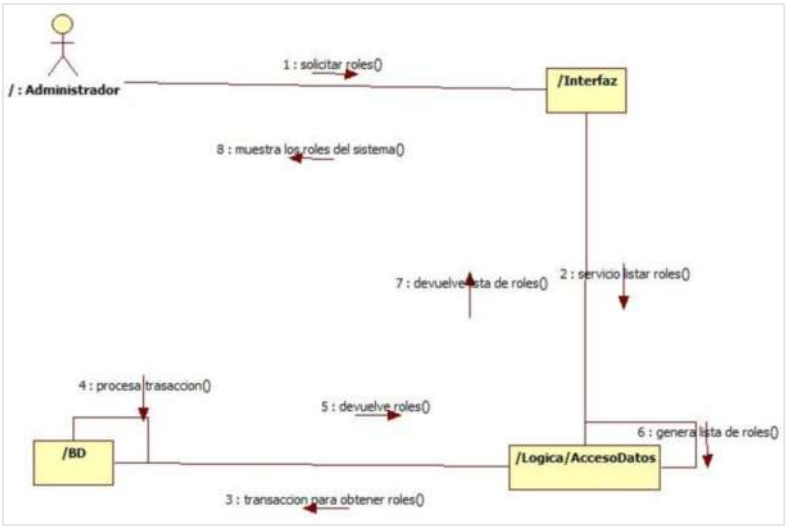
- **Desplegar aplicaciones.**



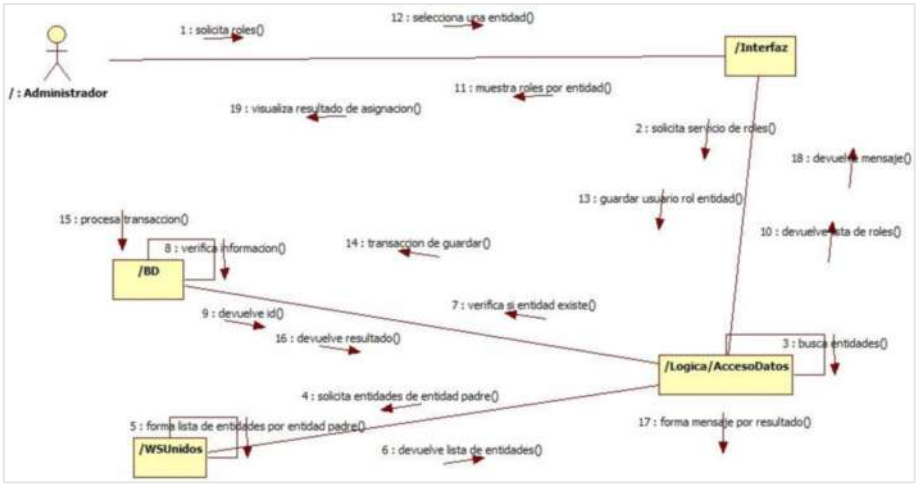
- **Buscar usuarios por cedula**



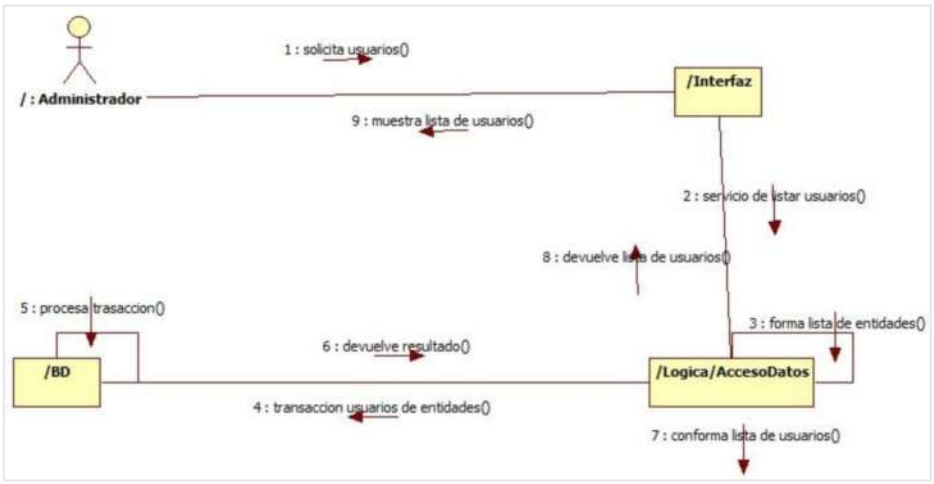
- **Listar roles del sistema**



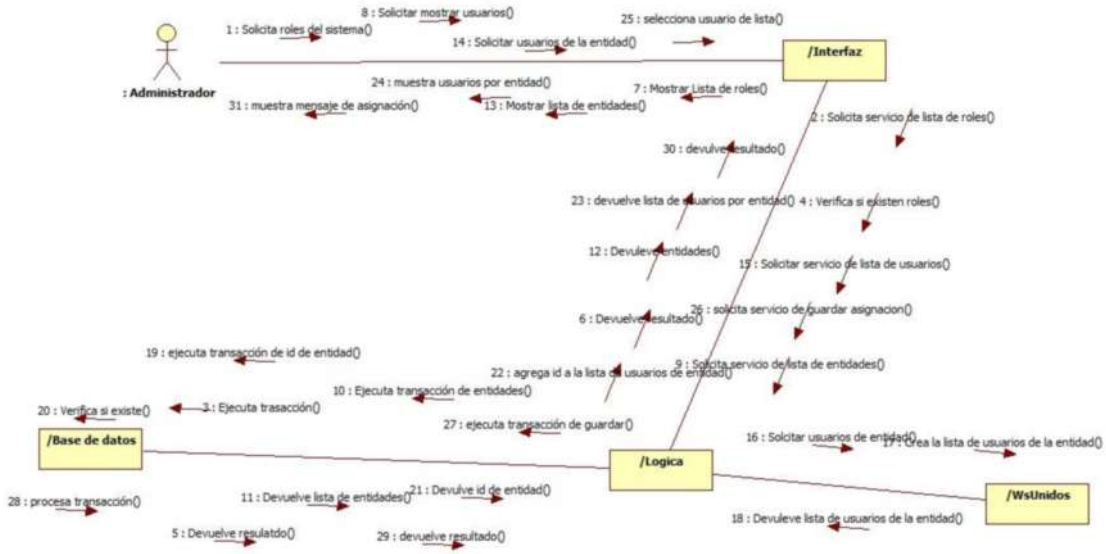
- **Agregar roles a un usuario del sistema**



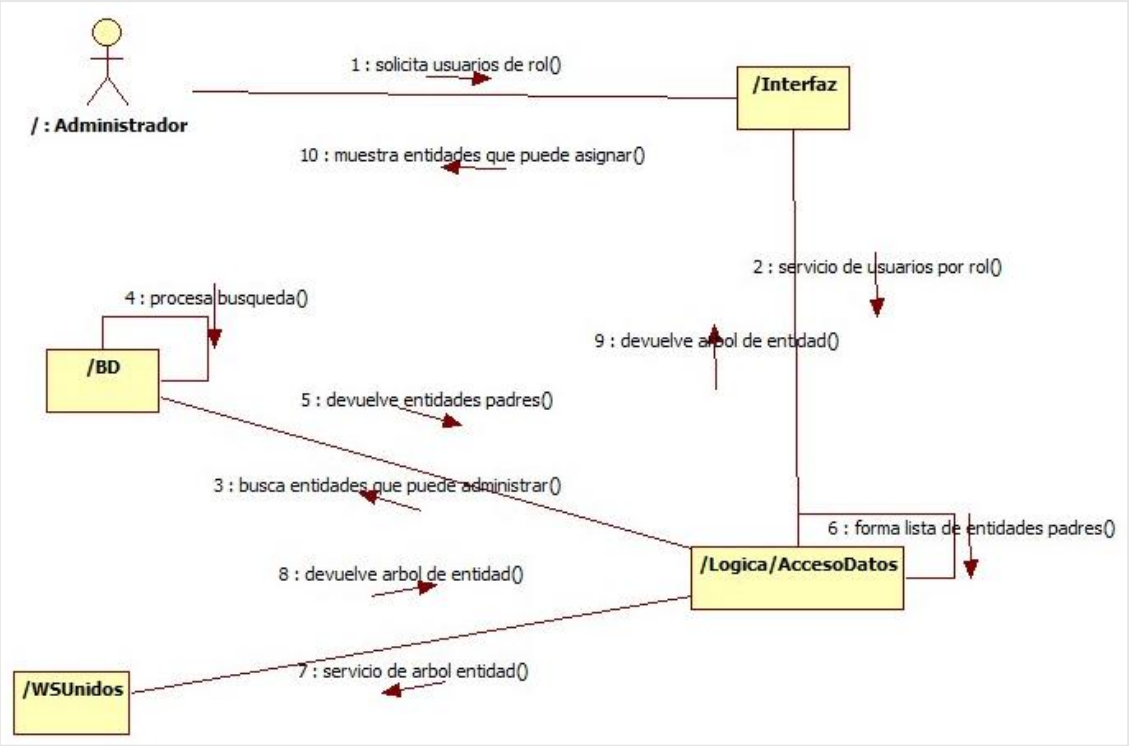
- **Listar usuarios del sistema**



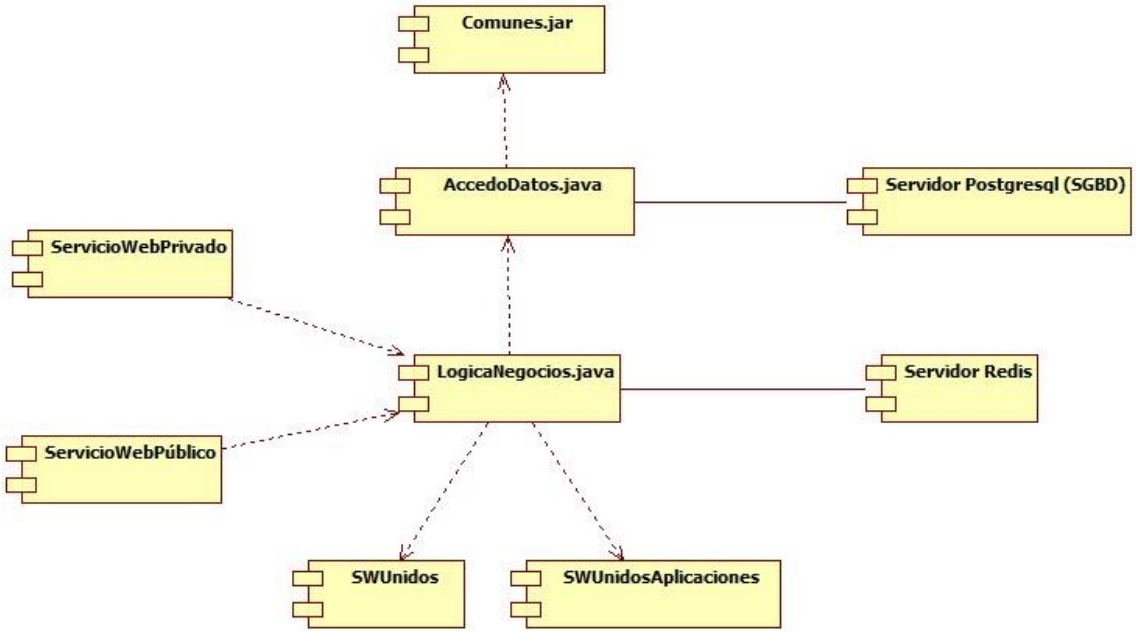
- **Agregar usuarios a un rol del sistema**



- **Listar entidades que puede administrar**

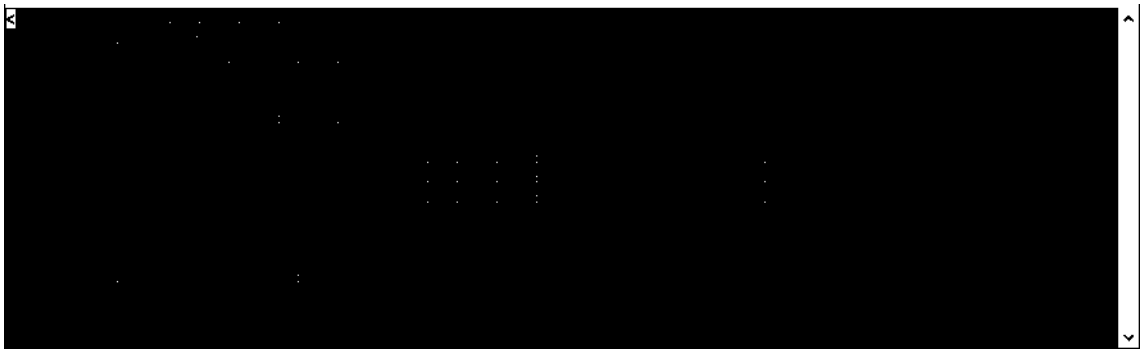


ANEXO D: Diagramas de Componentes



ANEXO E: Archivo de Arranque de Payara

ANEXO F: Balanceador de Carga



ANEXO G: PROCEDIMIENTO DE SERVIDOR DE APLICACIONES

1. Verificar si en el nodo **DAS** se encuentra instalado una versión de java jdk 8 con el siguiente comando:

   # update-alternatives --config java
2. Si en el paso anterior no se encontró una versión de java se procede a instalarlo con el siguiente comando:

 # yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel

3. Crear un nuevo usuario para configurar el servidor Payara con el siguiente comando:

 # adduser payara

Nota: el usuario puede ser cualquier nombre que considere pertinente

En la Figura 13 se visualiza la creación del usuario payara

4. Luego dirigirse al siguiente directorio:

 # cd /opt y se descarga él .zip de payara de la siguiente dirección mediante wget:

 # wget https://s3-eu-west-1.amazonaws.com/payara.fish/Payara+Downloads/Payara+4.1.2.173/payara-4.1.2.173.zip

Nota: antes de descargar el archivo se debe instalar dos paquetes con los siguientes comandos:

 # yum install wget

 # yum install unzip

Wget permite acceder a la descarga de archivos mediante la dirección donde se encuentra el archivo a descargar mientras.

Unzip permite descomprimir archivos rar, zip

Con lo descargado anteriormente procedemos a descomprimir el archivo con el comando

 # unzip payara-4.1.2.173.zip

5. Con el comando: # **chown -R payara:payara payara41** permite que el directorio payara41 pertenezca al usuario payara.
6. Además, se debe crear el archivo de arranque para ejecutar las acciones básicas de payara de la siguiente manera:

 # vi /etc/systemd/system/payara.service

Con el comando anterior se crea un archivo de texto y se escribe lo siguiente

```
[Unit]
Description = Payara Server v4.1
After = syslog.target network.target
[Service]
User=payara
ExecStart = /usr/bin/java -jar /opt/payara41/glassfish/lib/client/appserver-cli.jar start-domain
```

ExecStop = /usr/bin/java -jar /opt/payara41/glassfish/lib/client/appserver-cli.jar stop-domain
ExecReload = /usr/bin/java -jar /opt/payara41/glassfish/lib/client/appserver-cli.jar restart-domain
Type = forking
[Install]
WantedBy = multi-user.target

7. Al ejecutar el paso anterior se puede utilizar los comandos systemctl para ejecutar las acciones básicas del servidor lo cual permite ejecutar le siguiente comando:

 # systemctl enable payara permite habilitar que el servidor payara se inicie automáticamente cuando el servidor se inicie en la Figura 18

Los comandos principales que se pueden ejecutar para el servidor payara son los siguientes:

- systemctl start payara: Para iniciar el servidor
- systemctl disable payara: Para deshabilitar el inicio automático de payara
- systemctl restart payara: Para reiniciar payara
- systemctl stop payara: Para detener el servidor
- systemctl status payara: Para verificar el estado de nuestro servidor

8. Ingresar al usuario payara creado anteriormente con el comando:

 # su payara

Luego dirigirse al directorio del usuario payara

#cd /home/payara

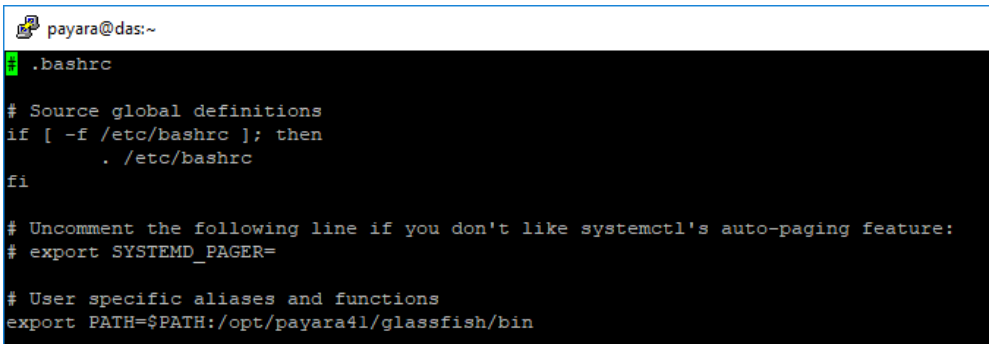
Una vez ahí se edita el siguiente archivo:

 # vi .bashrc

Al final del archivo agregar la siguiente línea de código:

export PATH=$PATH:/opt/payara41/glassfish/bin

El archivo editado quedara de la siguiente manera como se visualiza en la siguiente figura.



9. Una vez ejecutado los primeros se procede a habilitar la conexión remota para lo cual se debe dirigir al directorio bin del servidor payara con el siguiente comando:

cd /opt/payara41/glassfish/bin

Una vez ahí se ejecuta mediante asadmin los siguiente

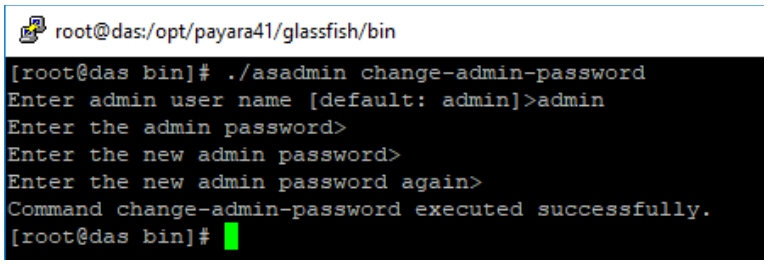
./asadmin change-admin-password

El comando anterior nos permite actualizar o donde nos solicita lo siguiente

admin: este el usuario por defecto que se crea para la conexión remota

espochdas: es la contraseña que se estableció para acceder al servidor de payara de manera remota.

En la siguiente figura al ejecutar el comando para asignar credenciales de payara, se solicita ingresar la información correspondiente lo que permitirá autenticarse de manera remota al servidor.



10. Ahora se habilita la conexión remota con el siguiente comando:

./asadmin enable-secure-admin el cual solicitara que ingresemos las credenciales creadas en el paso anterior si todo sale bien por ultimo solo se reinicia el servidor payara con el comando:

systemctl restart payara

Nota: Se debe habilitar el puerto 4848 en la tabla del firewall del servidor con los siguientes comandos:

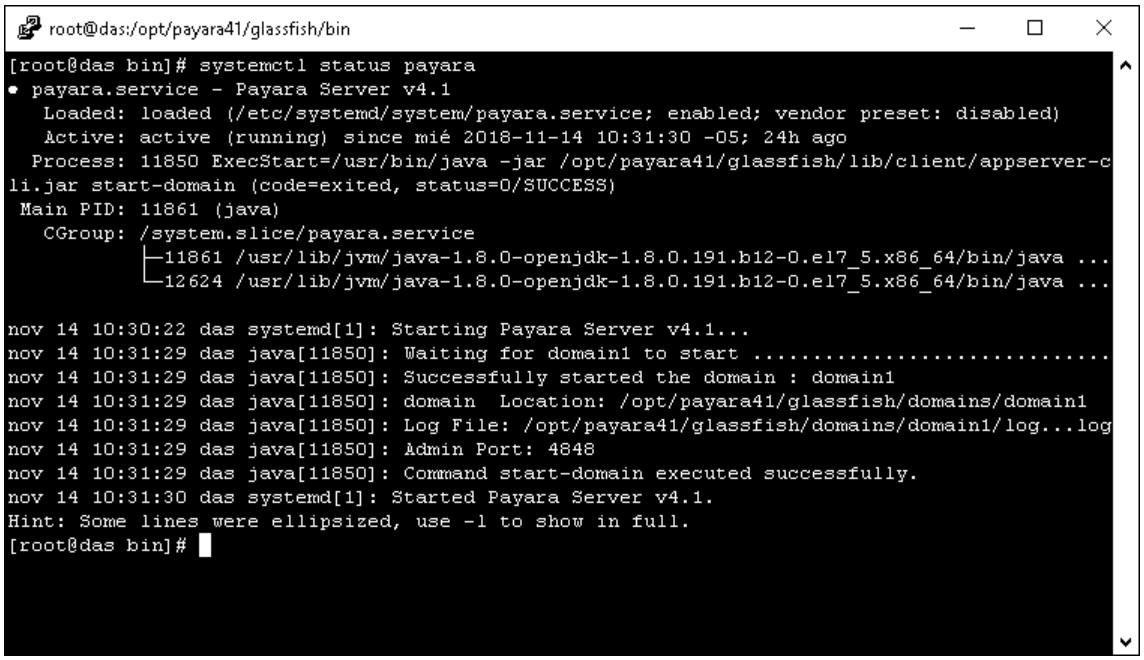
 # firewall-cmd --zone=public --add-port=4848/tcp --permanent permite conexión por el puerto 4848 que pertenece al servidor payara

#firewall-cmd --reload permite reiniciar el firewall

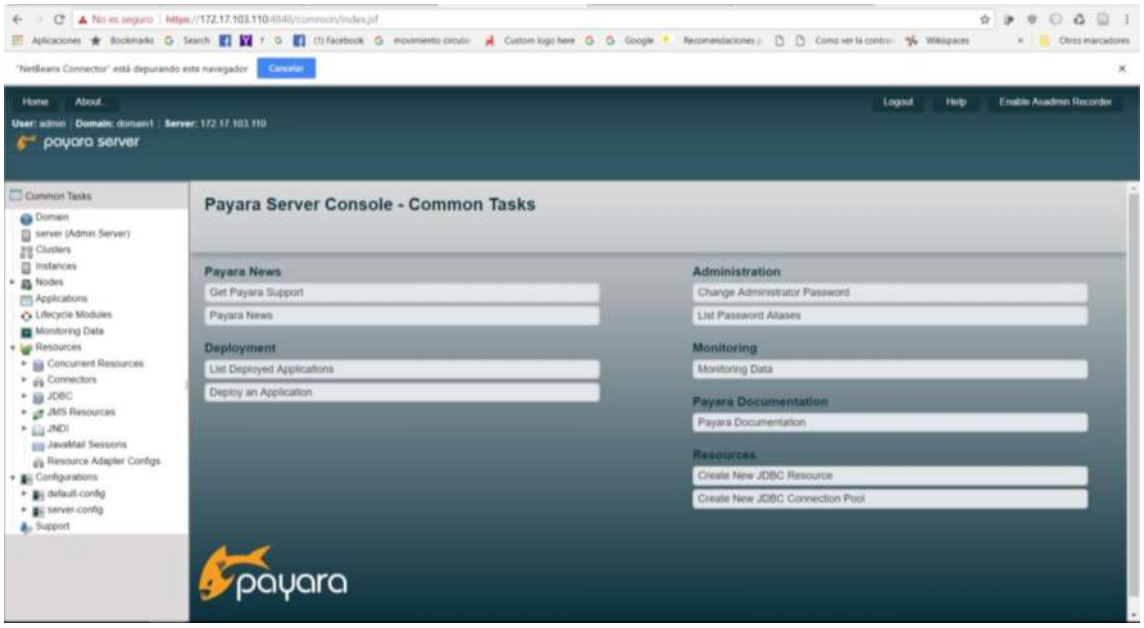
Por último, se verifica el estado del servidor payara con el siguiente comando:

 # systemctl status payara

La información luego de haber ejecutado el comando anterior es el que se visualiza en la siguiente figura.



Una vez realizado todos los pasos anteriores desde un navegador se puede ingresar a la consola de administración de payara como se observa en la siguiente figura



CONFIGURACIÓN DEL CLÚSTER

1. Se debe verificar que en los dos nodos este habilitado la conexión remota mediante ssh y esto se logra con el siguiente comando.

 # netstat -putna |grep ':22'

En el caso de que no esté habilitado instalarlo y habilitarlo para conexión remota

2. En este paso se debe verificar que los nodos permitan autenticación remota del usuario root en el siguiente archivo

 # vi /etc/ssh/sshd_config

Los archivos en ambos nodos deben guardarse según la siguiente figura

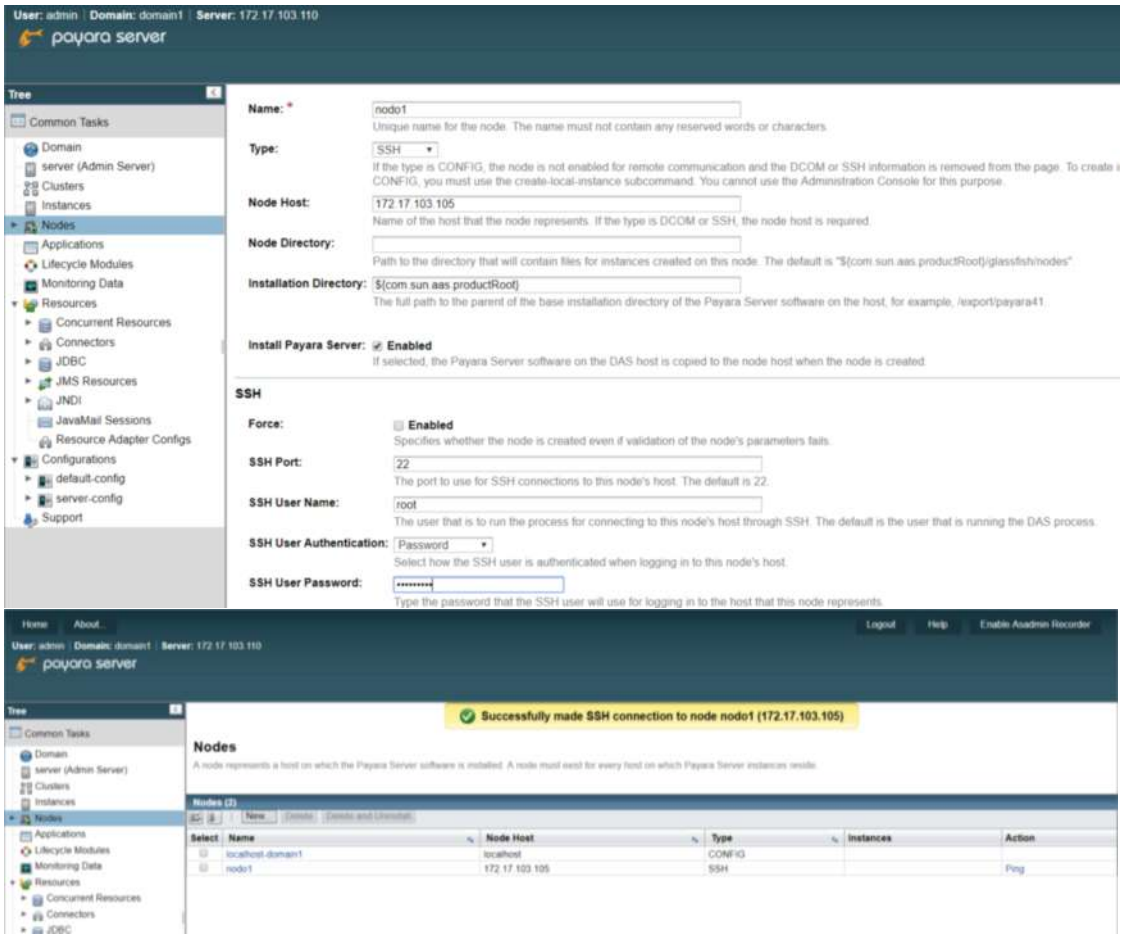
```
# Authentication:

#LoginGraceTime 2m
#PermitRootLogin yes
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
```

Además, se debe instalar los complementos de java en ambos nodos, para evitar inconvenientes debe ser el mismo instalado en el servidor principal (DAS).

yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel

3. Una vez configurado lo anterior se procede a crear los nodos haciendo referencia a cada una de las direcciones ips de los servidores disponibles para el clúster. Para lo cual se procede a ejecutar los pasos como se muestra en las dos figuras siguientes.



4. Una vez instalado remotamente payara en los servidores que estarán dispuesto como nodos se ejecuta los siguientes pasos para crear el dominio en caso de que no esté disponible.

Verificar si en el servidor se ha creado un dominio para esto dirigirse al siguiente directorio:

cd /opt/payara41/glassfish/bin

./asadmin list-domains nos permite listar los dominios existentes en el servidor.

Nota: en caso de que no existan dominios activos se visualizara un mensaje de error

Si no existe ningún dominio se procede a crearlo con el siguiente comando:

./asadmin create-domain --adminport 4848 domain1

[domain1**]=** nombre del dominio que se pretende crear puede ser cualquier nombre

[4848**]=** es el puerto del servidor que se configuro al instalar payara en el servidor principal (DAS)

Inicializar el dominio creado con el comando:

./asadmin start-domain

Habilitar la administración remota con el comando:

./asadmin change-admin-password: donde se especifica el nombre del administrador que por defecto es **admin** y se asigna una contraseña que en este caso es: **espochdas**

Verificar que las credenciales asignadas en el punto anterior con el siguiente comando:

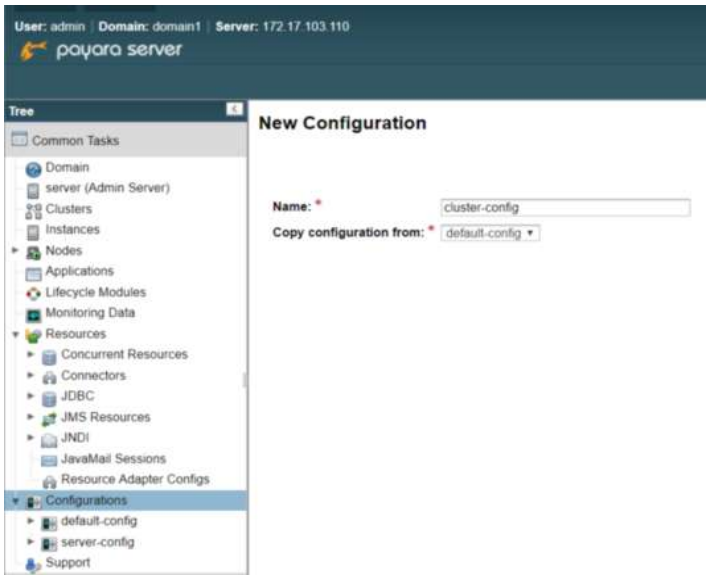
./asadmin enable-secure-admin

Por ultimo reiniciar el dominio con los siguientes comandos:

#./asadmin stop-domain : Detiene la ejecución del dominio

#./asadmin start-domain : Inicia la ejecución del dominio

5. Crear un nuevo archivo de configuración basado en default-config como se observa en la siguiente figura.



6. Configurar las instancias del servidor, se sigue el mismo proceso por cada nodo creado los parámetros para crear una instancia son los siguientes.

Instance Name: es el nombre de la instancia que se va a crear.

Node: especifica en que nodo se creara la instancia

Configuration: es el archivo de configuración con que se va a crear la instancia se debe elegir la nueva configuración que se creó anteriormente en este caso [**cluster-config**]

Por ultimo seleccionar la opción **Reference the Selected Configuration** [Referencia a la configuración seleccionada]

En la siguiente figura se observa la creación de una instancia

7. Una vez creado las instancias debe acceder a los nodos y modificar el siguiente archivo.

 # vi /opt/payara41/glassfish/nodes/nodo2/agent/config/das.properties

En el cual se debe asignar al atributo das.host la dirección ip del servidor principal en este caso la 172.17.103.117 como se visualiza en la siguiente figura, si no se realiza este paso probablemente no podrá inicializar las instancias creadas.

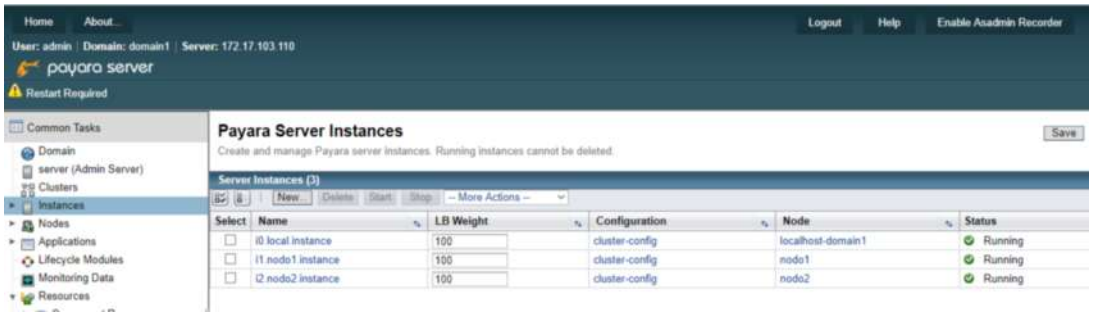
```
#Domain Administration Server Connection Properties
#Wed Nov 07 16:45:08 ECT 2018
agent.das.protocol=http
agent.das.port=4848
agent.das.host=172.17.103.110
agent.das.isSecure=false
```

8. Antes de ejecutar este paso se debe habilitar el puerto 24848 en ambos nodos de la siguiente manera

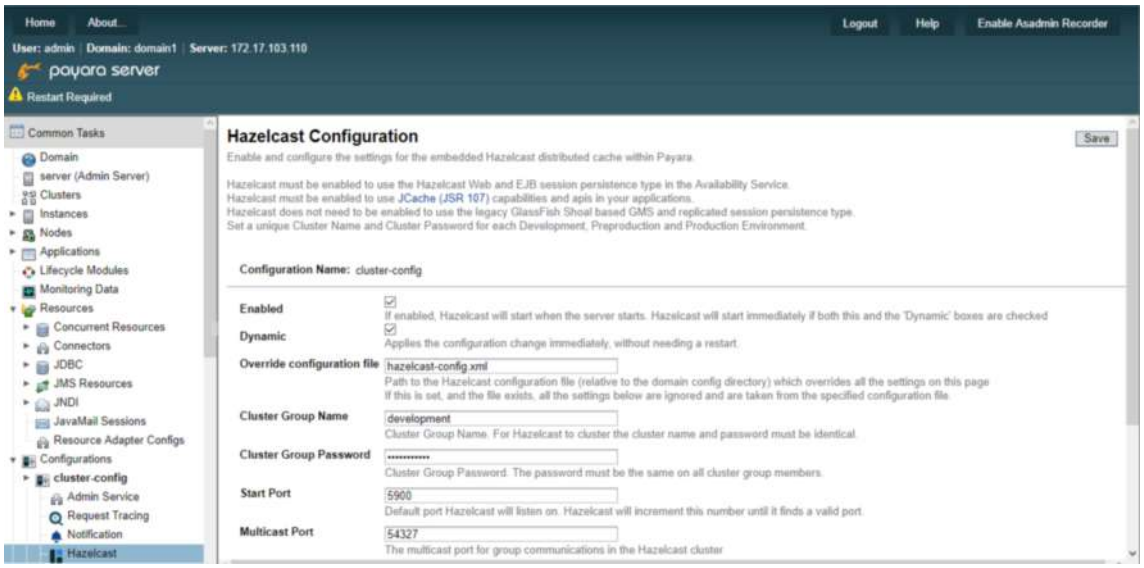
 # firewall-cmd --zone=public --add-port=24848/tcp --permanent

 # firewall-cmd --reload

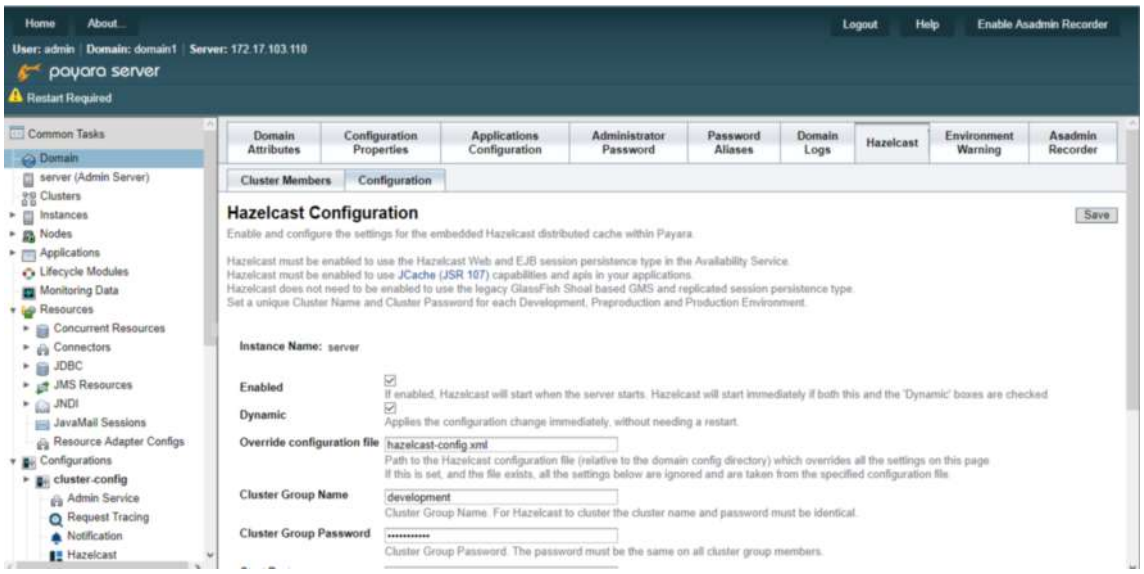
Ahora solo se debe iniciar las instancias como se observa en la siguiente figura.



9. Para crear el clúster basta con habilitar la opción Hazeltcast en el archivo cluster-config como se visualiza en la siguiente figura.



10. También se necesita que en **Domain** se habilite **hazelcast** como se observa en la siguiente figura, mediante esto se logra tener un clúster listo para desplegar aplicaciones.



ANEXO H: PROCEDIMIENTO DE SERVIDOR DE BASE DE DATOS

1. Agregar el repositorio de citus en los nodos mediante el siguiente comando

 # curl https://install.citusdata.com/community/rpm.sh | sudo bash

Luego de haber ejecutado el comando anterior se debe visualizar en los tres nodos el siguiente mensaje que se presenta en la siguiente figura.

```
The repository is set up! You can now install packages.
```

2. Instalar la base de datos postgresql con su extensión de citus en todos los nodos con el siguiente comando:

 # yum install -y citus74_10

3. Inicializar la base datos en todos los nodos con el siguiente comando: # **/usr/pgsql-10/bin/postgresql-10-setup initdb**

El resultado esperado es el que se visualiza en la siguiente figura.

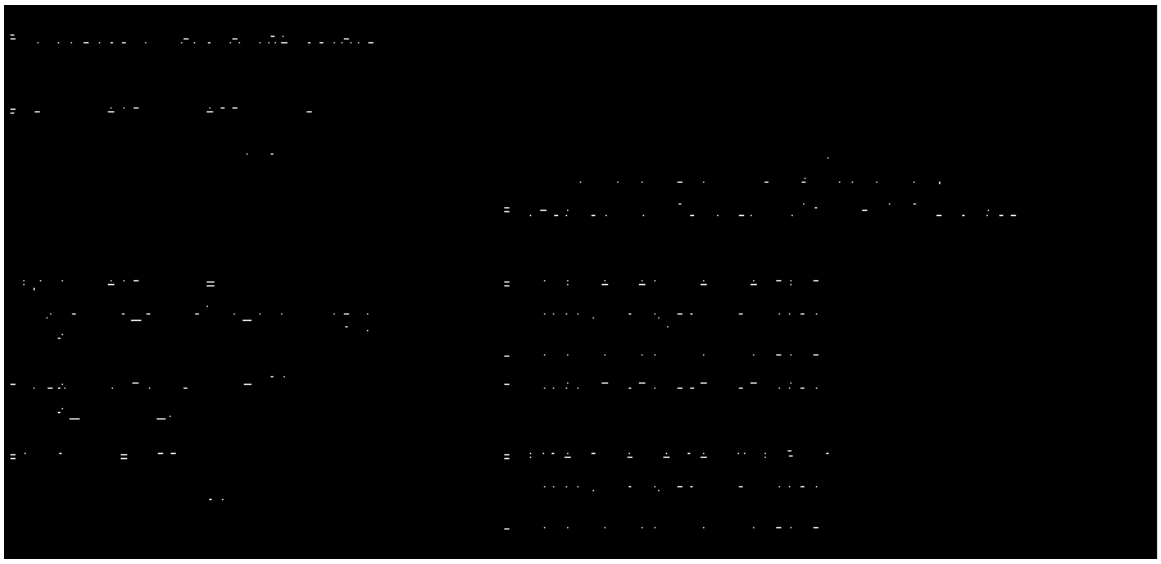
```
Initializing database ... OK
```

4. Cargar la extensión de citus en todos los nodos con el siguiente comando:
   # **echo "shared_preload_libraries = 'citus'"**
5. Ahora es necesario editar el archivo **postgresql.conf** para permitir conexiones externas y a su vez elegir el puerto que se va a utilizar, el archivo se encuentra en la siguiente ruta.

 # vi /var/lib/pgsql/11/data/postgresql.conf

Y los cambios deben quedar como se muestra en la siguiente figura

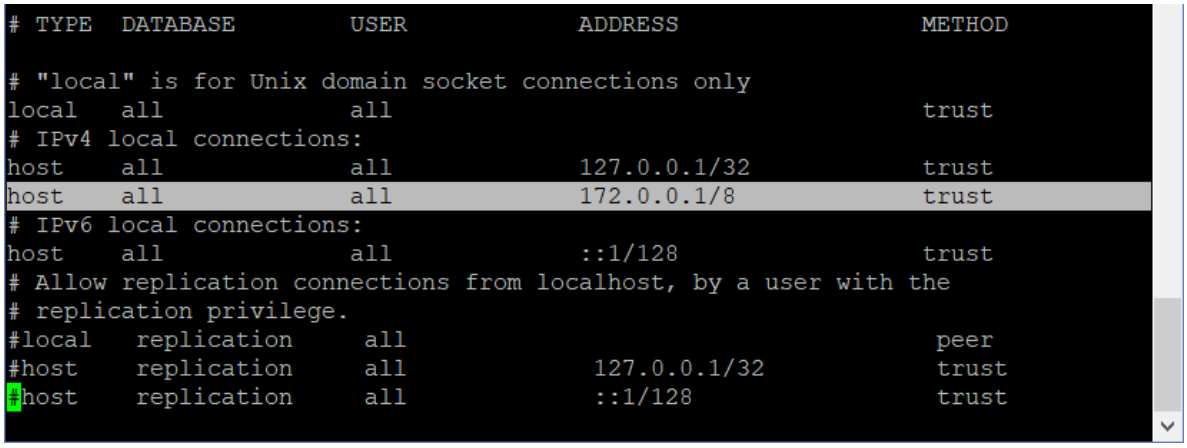
NODO COORDINADOR



6. Lo importante para la posterior creación del clúster es configurar el siguiente archivo **pg_hba.conf** lo que permitirá comunicarse desde el nodo coordinador hacia los nodos secundarios en este caso el 1 y el 2, el archivo se encuentra en la siguiente ruta.

 # vi /var/lib/pgsql/11/data/pg_hba.conf

La edición del archivo **pg_hba.conf** se lo realiza en los tres nodos y debe guardarse con los cambios que se presentan en la siguiente figura.

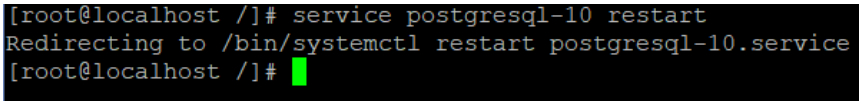


Es importante mencionar que el método de autenticación entre los nodos debe ser trust para configurar el clúster mediante citus sin problemas

7. Reiniciar el servidor de base de datos en los nodos ejecutando el siguiente comando

 # service postgresql-11 restart

El resultado del comando anterior se visualiza en la siguiente figura.



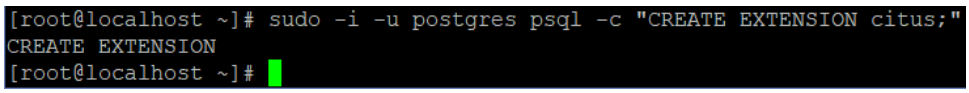
8. Habilitar el encendido automático de postgresql una vez que se encienda el servidor en los tres nodos en base al siguiente comando

 # chkconfig postgresql-10 on

9. Se debe crear la extensión citus en los nodos con el siguiente comando

 # sudo -i -u postgres psql -c "CREATE EXTENSION citus;"

Al ejecutar dicho comando se debe visualizar el resultado de la siguiente figura



10. Por último, se debe incluir en la tabla de firewall que el puerto de la base de datos esté disponible para conexiones externas, se debe incluir en los nodos y esto se logra con los siguientes comandos:

 # firewall-cmd --zone=public --add-port=5432/tcp --permanent

 # firewall-cmd –reload

CONFIGURACIÓN DEL CLÚSTER MEDIANTE CITUS

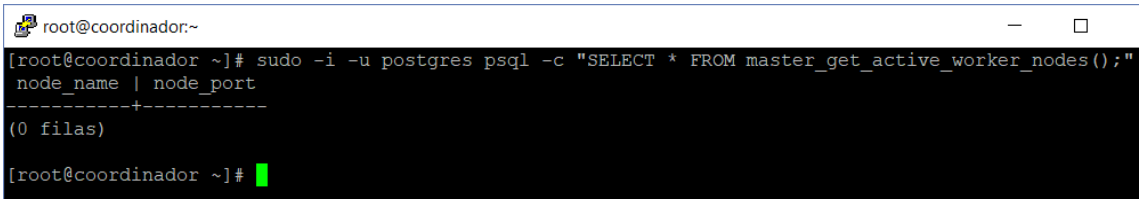
Para la configuración del clúster es tan sencillo mediante citus cabe indicar que este clúster solo permite distribuir las tablas de una base de datos lo que permite la distribución equilibrada de peticiones es decir el coordinador es el encargado de la escritura y los nodos 1 y 2 son encargados de la lectura.
Ahora se procede a crear el clúster con los siguientes pasos.

1. Primero verificar los nodos que pueda contener el nodo coordinador con el siguiente comando

sudo -i -u postgres psql -c "SELECT * FROM master_get_active_worker_nodes();"

Actualmente la salida en el nodo coordinador es la que se presenta en la siguiente figura.

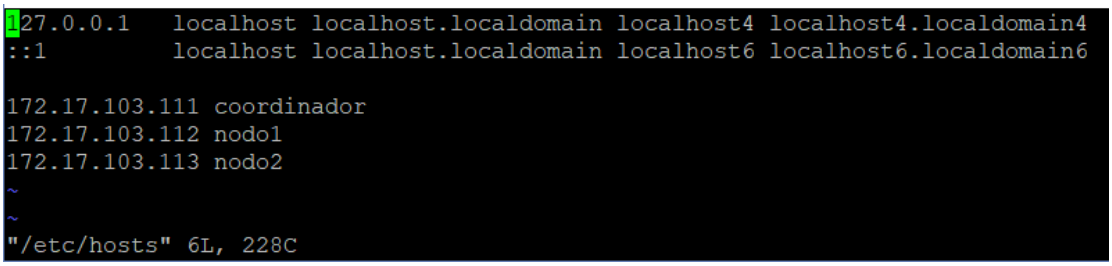
NODO COORDINADOR



2. Antes de agregar los nodos se debe modificar el archivo de la ruta

vi/etc/hosts para que reconozca todas las direcciones de los nodos involucrados.

Debe realizarse en cada uno de los nodos con la información que se presenta en la siguiente figura.



Con esto se asegura poder trabajar solo con los nombres que se designa a cada dirección.

1. Ahora se procede a agregar nodos únicamente en el nodo coordinador con el siguiente comando

 # sudo -i -u postgres psql -c "SELECT * from master_add_node('nombre del nodo', Puerto del nodo);"

El resultado de ejecutar el comando anterior se presenta en la siguiente figura

2. Ahora debe ejecutar nuevamente el comando del paso 1 y se debe obtener el resultado como se presenta la siguiente figura.

```
root@coordinador:~
[root@coordinador ~]# sudo -i -u postgres psql -c "SELECT * FROM master_get_active_worker_nodes();"
 node_name | node_port
-----------+-----------
 nodo1     |      5432
 nodo2     |      5432
(2 filas)

[root@coordinador ~]#
```

Como se puede observar en la Figura anterior existen dos nodos agregados al clúster de citus

ANEXO I: PROCEDIMIENTO DE SERVIDOR WEB

INSTALACIÓN DE APACHE

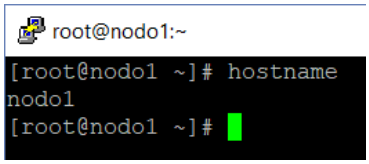
1. Asignar los siguientes nombres a los servidores como se observa en la siguiente tabla.

SERVIDORES	
172.17.103.105	nodo1
172.17.103.104	nodo2

Para asignar los nombres se debe ejecutar el siguiente comando:

#hostnamectl set-hostname [nombre de nodo]

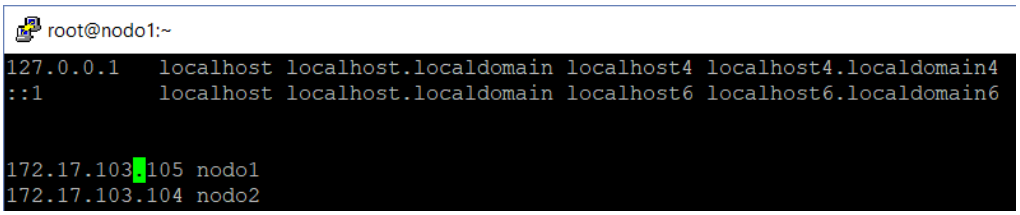
Luego de esto se ejecuta el comando hostname para verificar que los cambios se hayan realizado como se observa en la siguiente figura caso contrario debe salir de sesión y volver a ingresar.



2. Configurar el archivo hosts para que reconozca las direcciones ip de los dos nodos editando el siguiente archivo.

 # vi /etc/hosts

En la siguiente figura se visualiza como debe guardar el archivo hosts.



3. Instalar el servidor Apache en los nodos ejecutando el siguiente comando.

 # yum install httpd -y

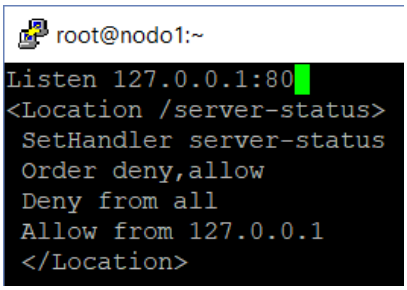
4. Una vez instalado apache se habilita el puerto que está asignado para el servicio de apache, en este caso es el puerto 80 para lo cual se procede a ejecutar en ambos nodos los siguientes comandos.

 # firewall-cmd --zone=public --add-port=80/tcp --permanent

 # firewall-cmd --reload

5. Crear un archivo de configuración, para verificar el estado de los nodos tal y como se presenta en la siguiente figura, esto permite saber que nodo está disponible.

 # vi /etc/httpd/conf.d/serverstatus.conf



6. Desactivar de la configuración de apache el puerto por el que este escuchando actualmente para evitar problemas posteriores durante la creación del clúster, se realiza en los nodos con el siguiente comando.

 # sed -i 's/Listen/#Listen/' /etc/httpd/conf/httpd.conf

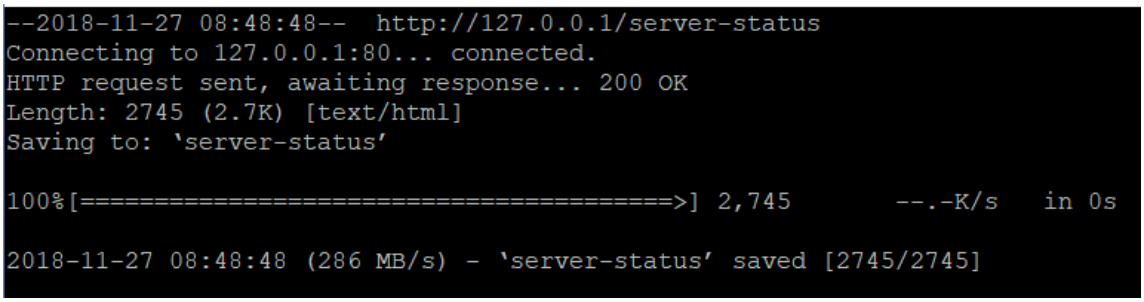
7. Reiniciar los servidores con el siguiente comando:

 # systemctl restart httpd

Luego verificar el estado de los nodos ejecutando el siguiente comando

 # wget http://127.0.0.1/server-status

El resultado esperado es el que se presenta en la siguiente figura



CONFIGURACIÓN DEL CLÚSTER

1. Instalar las herramientas corosync y pacemarker en los nodos con el siguiente comando

 # yum install corosync pcs pacemaker –y

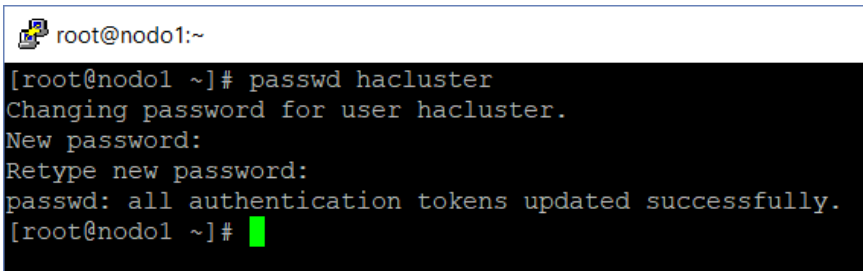
2. Para gestionar los nodos del clúster se utiliza pacemarker, esto permite tener un nodo quien controlara los demás nodos del clúster para ello se debe asignar una contraseña al usuario que se crea por defecto que es hacluster, en ambos nodos se debe asignar la misma contraseña que servirá para la autenticación entre los dos nodos tal y como se puede observar en la figura.

#passwd hacluster

New password: [contraseña]

Retype new password: [contraseña]

Contraseña asignada: **espochhttpd**

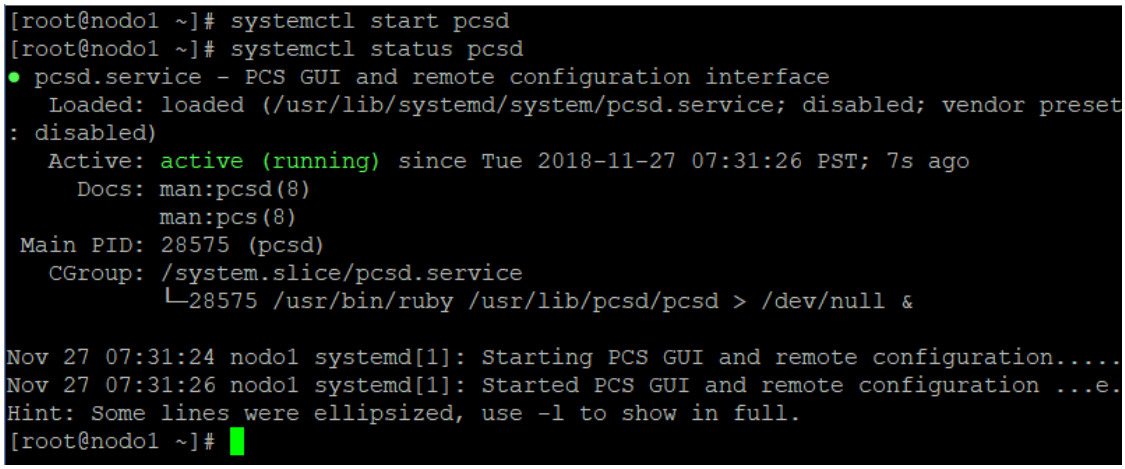


3. Inicializar el servicio pcsd para manipular los recursos que se instaló y posterior verificar su estado ejecutando los siguientes comandos.

#systemctl start pcsd

#systemctl status pcsd

El resultado esperado se observa en la siguiente figura.



4. Permitir el uso de las herramientas instaladas en los nodos se debe habilitar en puerto 2224 con el siguiente comando

 # firewall-cmd --zone=public --add-port=2224/tcp --permanent

Y a su vez también habilitar el servicio de alta disponibilidad en los nodos con los siguientes comandos

 # firewall-cmd --permanent --add-service=high-availability

 # firewall-cmd --add-service=high-availability

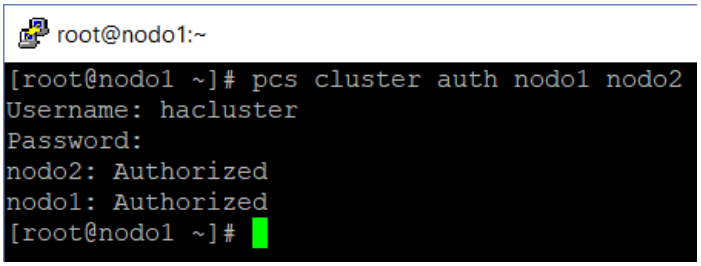
Y por último reiniciar el firewall para que se apliquen los cambios.

 # firewall-cmd --reload

5. Una vez verificado que el servicio este activo y al haber asignado la contraseña en el usuario hacluster, se procede a establecer comunicación entre los nodos mediante el siguiente comando.

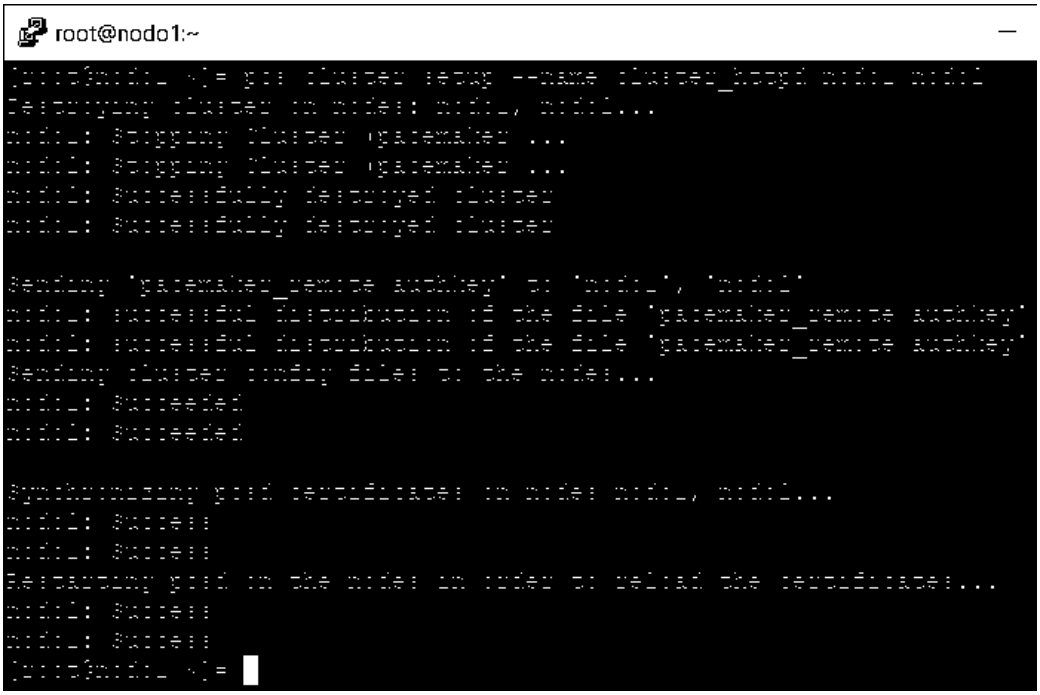
 # pcs cluster auth nodo1 nodo2

El resultado esperado se aprecia en la siguiente figura



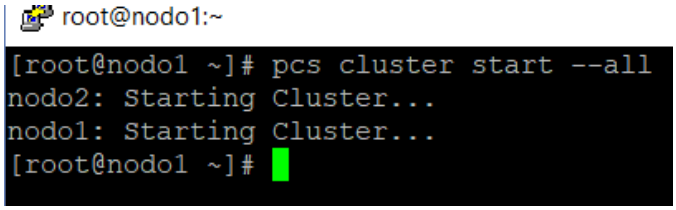
6. Con la herramienta pcs crear un clúster donde va a estar asignados los nodos, en este caso el nombre del clúster es **cluster_httpd,** el resultado debe ser el que se visualiza en la siguiente figura posterior a ejecutar el siguiente comando.

 # pcs cluster setup --name cluster_httpd nodo1 nodo2



7. Iniciar el clúster que se creó, el resultado debe ser el que se visualiza en la siguiente figura, posterior a ejecutar el siguiente comando.

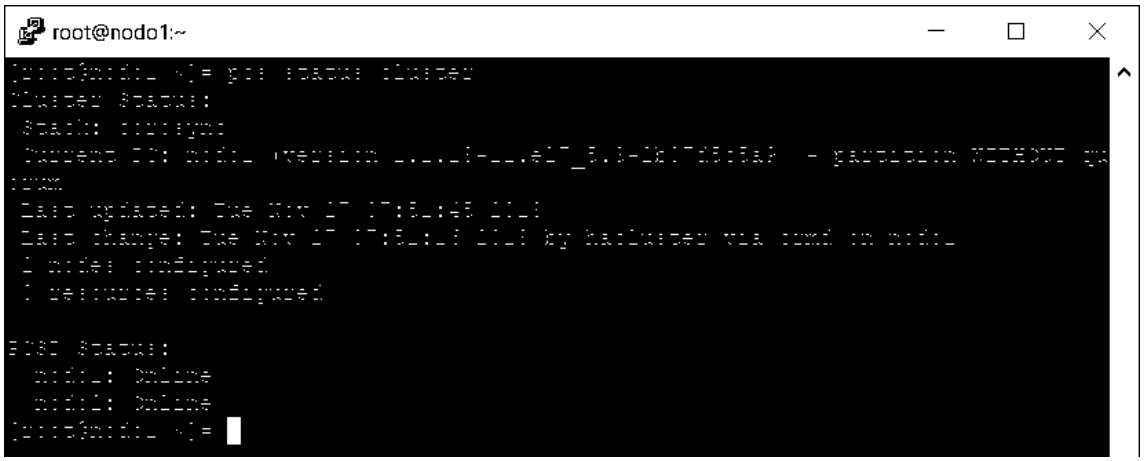
 # pcs cluster start --all



8. Con el siguiente comando se verifica el estado de los nodos

 # pcs status cluster

El comando anterior debe mostrar como resultado que existen 2 nodos configurados en estado activo como se presenta en la siguiente figura.



9. Deshabilitar stonith, que es el encargado del cercado en caso de que un nodo falle, para esto se ejecuta el siguiente comando

 # pcs property set stonith-enabled=false

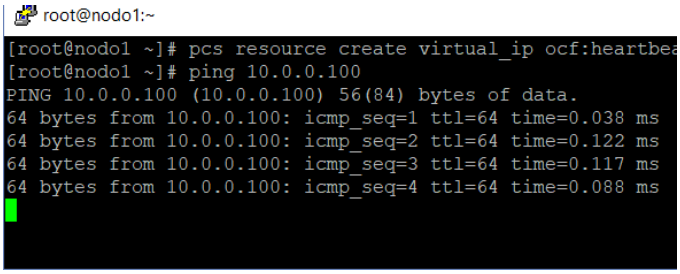
10. Ignorar el quorum ya que es un escenario solo de dos nodos, debería estar habilitado en el caso que se configure un total de 4 nodos, esto se logra bajo con el siguiente comando

 # pcs property set no-quorum-policy=ignore

11. Crear una ip virtual con el servicio pcs, la ip creada permitirá recibir todas las peticiones del servicio de apache ya instalado en los nodos.

 # pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=172.17.103.110 cidr_netmask=24 op monitor interval=30s

12. Luego de haber creado la ip virtual hacer ping a la dirección creada, desde los nodos como se observa en la siguiente figura para ver si tienen conexión entre sí.



13. Asignar la ip virtual como la dirección por la que va a escuchar nuestro servidor apache, en este caso fue 172.17.103.110 y el puerto que es el 80 esto debe realizarse en los nodos al ejecutar el siguiente comando.

 # echo "Listen 172.17.103.110:80"|sudo tee --append /etc/httpd/conf/httpd.conf

14. Se debe crear un recurso en este caso se denominado webserver al cual se le asignará el servicio de apache, el mimo que será objeto de monitoreo cada minuto y así verificar las fallas que se pueden presentar. Para esto debe ejecutar el siguiente comando.

 # pcs resource create webserver ocf:heartbeat:apache configfile=/etc/httpd/conf/httpd.conf statusurl="http://localhost/server-status" op monitor interval=1min

15. Al ejecutar el siguiente comando permitirá que tanto el recurso de ip virtual creado y el servidor web se inicialicen en el mismo nodo para evitar inconvenientes al procesar las peticiones del servidor web.

 # pcs constraint colocation add webserver virtual_ip INFINITY

16. Se necesita una ip para utilizar el servidor web por eso es necesario agregar la restricción de que primero debe estar disponible el recurso de ip virtual para que el recurso servidor web funcione, esto se logra con el siguiente comando

 # pcs constraint order virtual_ip then webserver

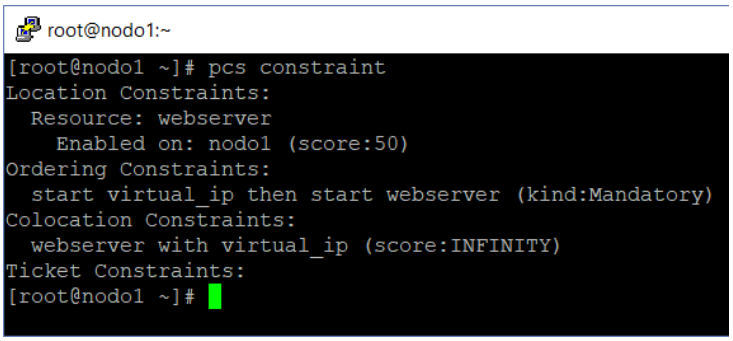
17. Para la creación de un clúster a veces pueden ser con máquinas que tengan diferentes características en los recursos de las mismas. Razón por la cual el siguiente comando nos permitirá ubicar en el nodo que nos parezca conveniente en este caso se utiliza el nodo 1

 # pcs constraint location webserver prefers nodo1=50

18. El siguiente comando se listan todas las restricciones que se ha configurado.

 # pcs constraint

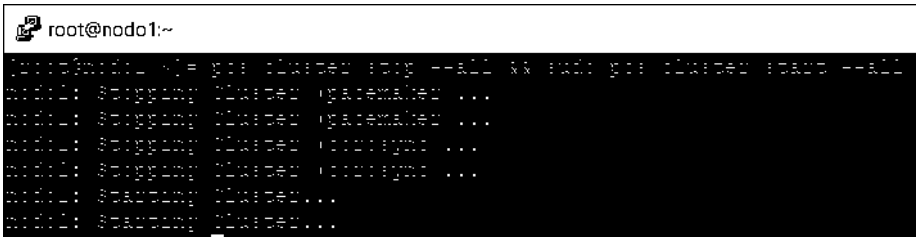
Que luego de ejecutar el comando anterior la información se debe presentar como en la siguiente figura.



19. Detener e iniciar el clúster configurado con el siguiente comando

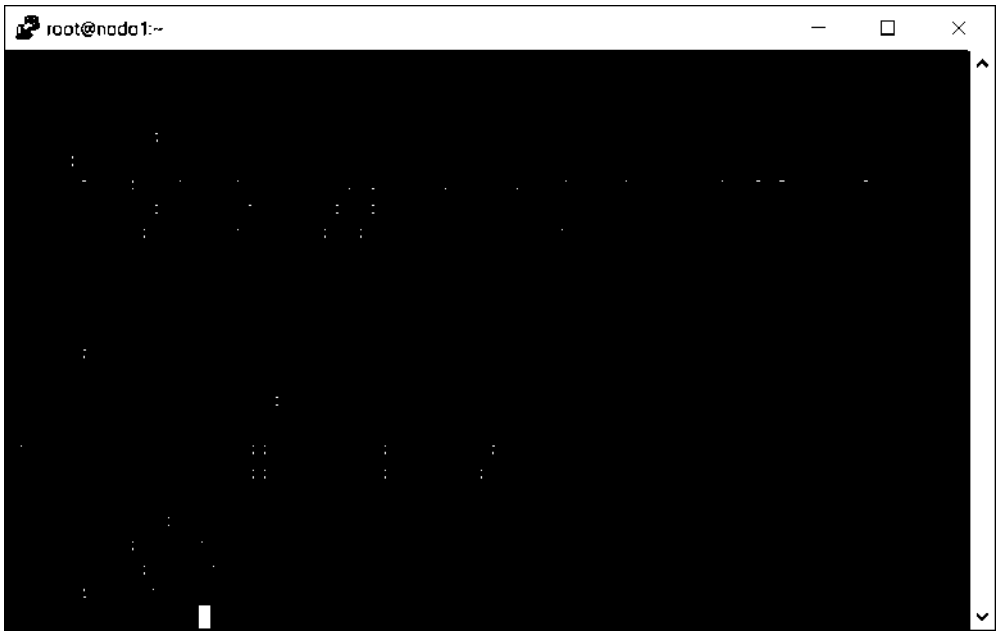
 # pcs cluster stop --all && sudo pcs cluster start –all

El resultado esperado es el que se presenta en la siguiente figura.



20. Es necesario verificar el estado del clúster, con el siguiente comando permite visualizar la lista de recursos configurados, por defecto se inicializan en el nodo 1 como se estableció en la restricción. Si la salida no se presenta como en la siguiente figura asegúrese de haber seguido los pasos correctamente y como último recurso reiniciar los servidores.

 # pcs status


root@nodo1:~