

MAQUINAS VIRTUALES VS CONTENEDORES

Huichi Contreras, Franklin Carlos (2016056193), Gonzales Cave, Angel Gabriel (2017057861), Condori Quispe, Yhónn Joel (2016056358), Pastor Mendoza, José Edilberto (2016055237)

Escuela Profesional de Ingeniería de Sistemas

Universidad Privada de Tacna

Tacna, Perú

Abstract

Lorem Ipsum

1. Resumen

Lorem Ipsum

2. Introducción

Lorem Ipsum

3. Marco teorico

3.1. ¿Que es la virtualización?

La virtualizacion es la creación de una versión virtual (no física) de algo. Esta basada en software, se puede aplicar a sistemas operativos, almacenamiento, servidores, aplicaciones, redes, etc. y es una manera de reducir gastos y aumentar eficiencia y agilidad en las empresas.

3.1.1. Virtualizacion de servidores

La virtualizacion de servidores ayuda a evitar ineficiencias ya que permite ejecutar varios sistemas operativos en una maquina fisica con maquinas virtuales con acceso a los recursos de todos. Tambien permite generar un cluster de servidores en un unico recurso para así mejorar mucho mas la eficiencia y la reduccion de costes. Tambien permite el aumento de rendimiento de las aplicaciones y la disponibilidad al aumentar la velocidad en la carga de trabajo.

3.1.2. Virtualizacion de escritorios

La implementacion de escritorios virtualizados permite ofrecer a las sucursales o empleados externos de forma rapida y sencilla un entorno de trabajo y una reduccion de la inversion a la hora de gestionar cambios en estos.

3.1.3. Virtualizacion de red

Se trata de reproducir una red fisica completa mediante software para poder ejecutar los mismos servicios que una red convencional y sus dispositivos. Cuentan con las mismas características y garantías que las redes físicas con las ventajas que nos ofrece la virtualizacion ademas de la liberacion del hardware.

3.1.4. Almacenamiento definido por software

La virtualizacion del almacenamiento permite prescindir de los discos de los servidores. Los combina en depositos de almacenamiento de alto rendimiento y los distribuye como software. Este nuevo modelo permite aumentar la eficiencia en el guardado de datos.

3.2. Ventajas de la virtualizacion

Como se ha podido apreciar en los tipos de virtualizacion presentados anteriormente, esta conlleva una mejora considerable tanto en el rendimiento, agilidad, flexibilidad, escalabilidad, etc. como en una reduccion considerable de los costes economicos y de tiempo y una simplificacion en la gestion de la infraestructura.

- Reduce los costes de capital y los gastos operativos.
- Minimiza o elimina los tiempos de inactividad
- Aumenta la productividad, la eficiencia, la agilidad y la capacidad de respuesta

- Implementa aplicaciones y recursos con mas rapidez.
- Garantiza la continuidad del negocio y la recuperacion ante desastres.
- Simplifica la gestion del centro de datos.

4. Autores

Roles	Integrantes
Product owner	Gonzales Cave, Angel Gabriel
Scrum master	Huichi Contreras, Franklin Carlos
Equipo scrum	Condori Quispe, Yhónn Joel — Pastor Mendoza, José Edilberto

Cuadro 1: Integrantes del equipo

5. Planteamiento del problema

Lorem Ipsus

5.1. *Problema*

Lorem Ipsus

5.2. *Justificacion*

Lorem Ipsus

5.3. *Alcance*

Lorem Ipsus

5.4. *General*

Lorem Ipsus

5.5. *Especificos*

- Analizar cuáles son las peticiones de soporte técnico mas solicitadas por los usuarios, para obtener la informacion necesaria y definir los requerimientos para el correcto desarrollo del sistema
- Mejorar la productividad de la empresa informando a los usuarios sobre como solucionar los problemas comunes.

6. Referentes Teoricos

Lorem Ipsum

A continuación se describe la conformación actual de ITIL v3 :

- Libro 1 Estrategia del Servicio Propone tratar la gestión de servicios no sólo como una capacidad sino como un activo estratégico
- Libro 2 Diseño del Servicio Cubre los principios y métodos necesarios para transformar los objetivos estratégicos en portafolios de servicios y activos.
- Libro 3 Transición del Servicio Cubre el proceso de transición para la implementación de nuevos servicios o su mejora
- Libro 4 Operación del Servicio Cubre las mejores prácticas para la gestión del día a día en la operación del servicio.
- Libro 5 Mejora Continua del Servicio Proporciona una guía para la creación y mantenimiento del valor ofrecido a los clientes a través de un diseño, transición y operación del servicio optimizado.

[3]

7. Desarrollo de la propuesta

7.1. *Tecnología de información*

- NoSQL el término fue inicialmente utilizado en el año 1998, y fue para denominar una base de datos relacional que no utilizaba el lenguaje SQL para funcionar. Características principales :

Evitar la complejidad innecesaria Las bases de datos relacionales proporcionan gran cantidad de funcionalidades y restricciones para mantener la consistencia de los datos, en ciertos casos, mucho más de lo necesario.

Alto rendimiento Muchas bases de datos NoSQL proporcionan un rendimiento superior al que ofrecen los sistemas RDBS convencionales.

[2]

- Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.
- Flask es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código.

7.2. *Metodología, técnicas usadas*

SCRUM es una metodología para gestión, mejora y mantenimiento de un sistema nuevo o existente. SCRUM se concentra en cómo los miembros del equipo deberían funcionar a fin de producir un sistema flexible en un entorno que cambia constantemente. [1]

7.2.1. *Inicio*

- Planeamiento: Consiste en establecer la visión, el presupuesto, forma de financiamiento y el backlog del producto. En esta fase se selecciona que funcionalidad es la más apropiada para desarrollo inmediato. También se establece el equipo de trabajo, se evalúan las herramientas de desarrollo y se define la fecha de entrega (es una fecha aproximada). [1]
- Arquitectura : Esta fase consiste en la conceptualización y análisis. Si el proyecto se trata de la mejora de un nuevo sistema, sólo se realiza un análisis limitado. Se realiza un diseño de alto nivel para actualizar los modelos del dominio y reflejar el contexto del nuevo sistema y los requerimientos y las modificaciones necesarias de la arquitectura del sistema. Los diseñadores y arquitectos dividen el proyecto en paquetes basándose en los ítems del backlog. En la jerga de SCRUM se llaman "paquetes" a los objetos o componentes que necesitan cambiarse en cada iteración. [1]

7.2.2. *Desarrollo*

En esta etapa se realiza el desarrollo propiamente dicho. También se la conoce como "Ingeniería concurrente". La misma se divide en iteraciones que proveen como resultado funcionalidades incrementales al fin de cada una de ellas. Dichas iteraciones se llaman sprints.

Un sprint dura aproximadamente entre una semana y 30 días. Cada sprint incluye las fases tradicionales del desarrollo de software: requerimientos, análisis, diseño, desarrollo y entrega. Durante un sprint no se utilizan diagramas de gantt para seguimiento de tareas (estos parten del supuesto que las tareas de un proyecto se pueden identificar y ordenar), debido a que el desarrollo es semi-caótico y cambiante como para que se le aplique un proceso definido. Durante un sprint no se pueden cambiar los miembros del equipo scrum. Tampoco pueden introducirse cambios durante un sprint (si surge algún cambio se incluya en el backlog del producto, pero no en el del sprint). El scrum master mantiene el sprint backlog. Actualiza las tareas finalizadas y para las que no lo están, el tiempo que el equipo piensa que tomará para terminarlas. [1]

7.2.3. *Reuniones Scrum*

Durante un sprint, todos los días se realizan reuniones llamadas «SCRUM». El objetivo de las mismas es quitar los impedimentos que le surgen a los miembros del equipo scrum. Cada una de ellas dura aproximadamente 15 minutos. A cada miembro del equipo scrum se le pregunta:

- ¿Qué hizo durante las últimas 24 horas?
- ¿Qué planea hacer las próximas 24 horas?
- ¿Qué obstáculos se le han presentado en las últimas 24 horas ?

Estas reuniones deben realizarse obligatoriamente.[1]

7.2.4. *Cierre*

Esta etapa comienza cuando el equipo de management decide que las variables de entorno, tales como los requerimientos se han completado. En esta etapa se genera la documentación. Un final, se realiza el testing pre-lanzamiento y el lanzamiento propiamente dicho. [1]

8. Cronograma

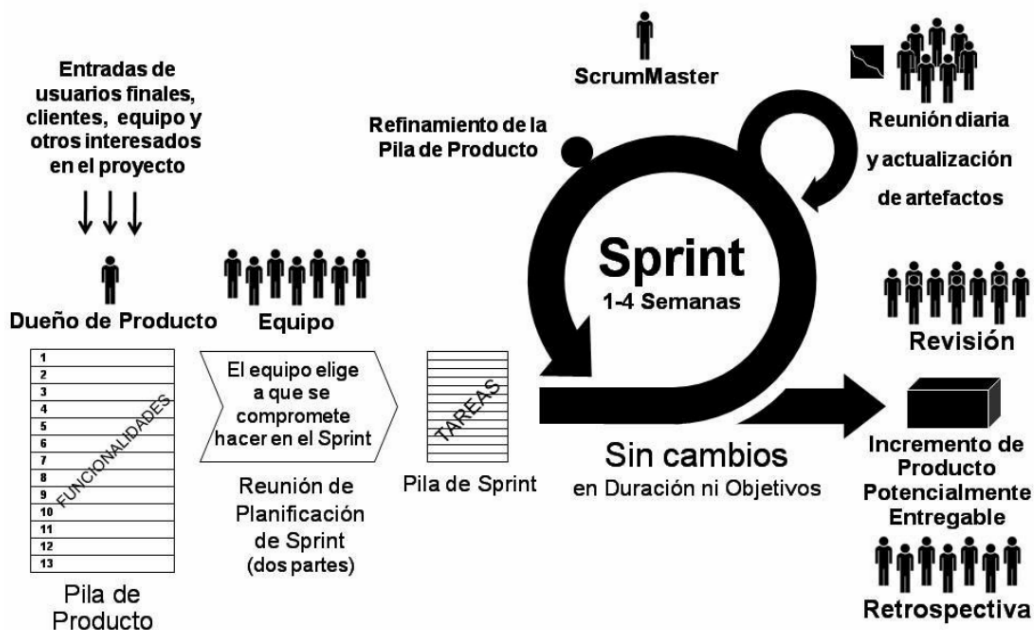


Figura 1: Cronograma del proyecto

Referencias

- [1] Caso, N. (2004). Scrum development process. Recuperado de <https://bit.ly/2ouTR57>. Accedido el 09-10-2019.
- [2] Herranz, R. (2014). Bases de datos nosql : Arquitectura y ejemplos de aplicacion. Recuperado de <https://core.ac.uk/download/pdf/44310803.pdf>. Accedido el 10-09-2019.
- [3] Ríos, S. (2007). *ITIL v3*. Biagle Management.

Cronograma del proyecto

Powered by  **monday.com**
Click here to start your free trial

Especificar requisitos y evaluar proyecto

Name	Owner	Status	Timeline - Start	Timeline - End	Priority
Recolección de requisitos	Huichi		2019-10-09	2019-10-12	High
Incorporar mejoras en requisitos	Huichi, Gonzalez		2019-10-13	2019-10-14	Medium
Elaborar el cronograma	Huichi		2019-10-15	2019-10-16	High
Conseguir aprobación para continuar	Huichi		2019-10-17	2019-10-17	High
Asegurar los recursos necesarios	Condori		2019-10-18	2019-10-20	Medium

Análisis y Diseño

Name	Owner	Status	Timeline - Start	Timeline - End	Priority
Análisis de requisitos	Huichi, Gonzalez		2019-10-21	2019-10-24	High
Diseño de arquitectura	Huichi, Gonzalez		2019-10-25	2019-10-27	Medium
Desarrollo de prototipos	Huichi, Gonzalez		2019-10-27	2019-11-01	Medium
Revisión de diseño	Condori		2019-10-27	2019-11-01	Medium
Incorporar mejoras en el diseño	Huichi, Gonzalez		2019-11-02	2019-11-05	Low
Conseguir aprobación para aprobar	Huichi		2019-11-06	2019-11-06	High

Programación

Name	Owner	Status	Timeline - Start	Timeline - End	Priority
Revisión en especificación de diseño	Condori, Pastor		2019-11-03	2019-11-08	High
Diseño detallado de módulos	Angel		2019-11-07	2019-11-10	High
Codificación	Condori, Pastor, Gonzalez		2019-11-10	2019-11-15	Medium
Pruebas de programación	Condori		2019-11-10	2019-11-15	Medium

Pruebas

Name	Owner	Status	Timeline - Start	Timeline - End	Priority
Probar módulos y sus componentes	Condori		2019-11-16	2019-11-19	Medium
Detectar fallos	Condori		2019-11-16	2019-11-22	High
Corregir fallos	Condori		2019-11-16	2019-11-22	High
Volver a probar código modificado	Condori		2019-11-16	2019-11-22	High

Despliegue

Name	Owner	Status	Timeline - Start	Timeline - End	Priority
Establecer estrategia de despliegue	Gonzalez, Pastor		2019-10-27	2019-11-09	High
Desarrollar metodología	Huichi, Gonzalez		2019-10-27	2019-11-22	High
Desplegar el software	Condori, Pastor		2019-11-23	2019-11-26	Medium

Figura 2: Cronograma del proyecto