

CONTROL Y SEGUIMIENTO DE PROYECTOS

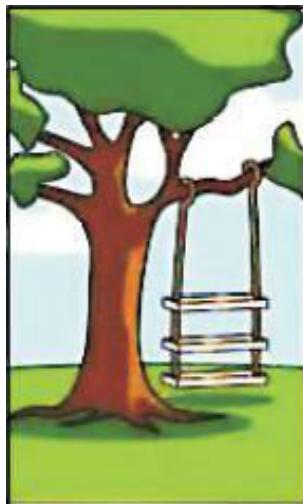
2020

Administración de Recursos

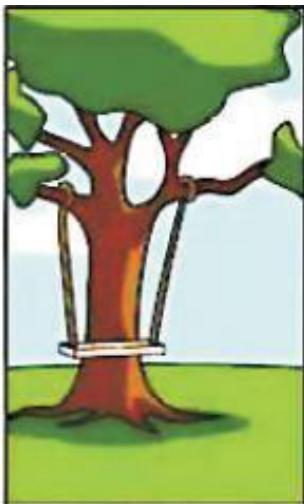
TEMARIO

- ▶ METODOLOGÍAS Y ESTÁNDARES
- ▶ DEFINICIÓN DE PROYECTO
- ▶ FASES
- ▶ RIESGOS Y PROBLEMAS
- ▶ SEGUIMIENTO Y CONTROL
- ▶ ÉXITO Y COMPLEJIDAD DE PROYECTOS

¿Por qué utilizar un metodología para gestionar?



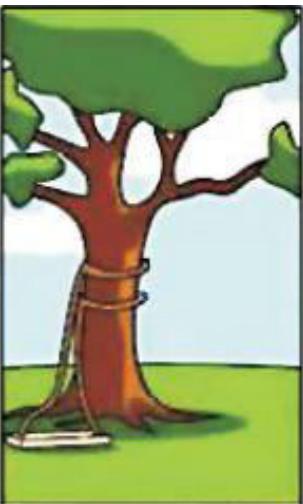
La solicitud del usuario



Lo que entendió el líder del proyecto



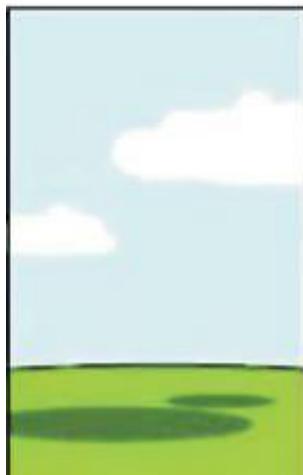
El diseño del analista de sistemas



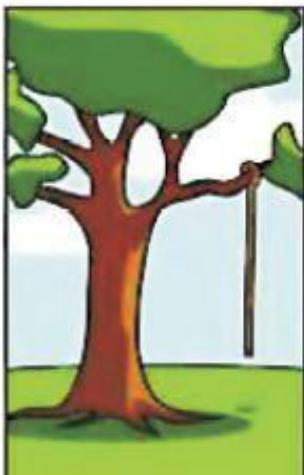
El enfoque del programador



La recomendación del consultor externo



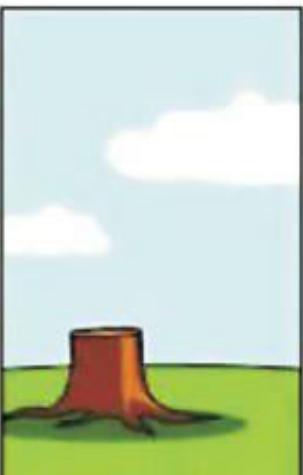
La documentación del proyecto



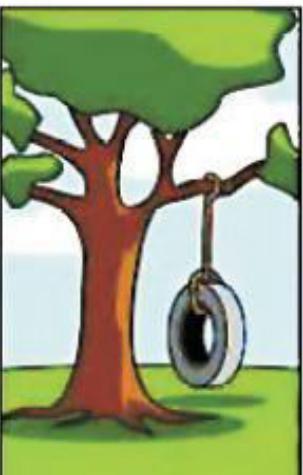
La implantación en producción



El presupuesto del proyecto



El soporte operativo



Lo que el usuario realmente necesitaba

¿Por qué utilizar un metodología para gestionar?

Fallas comunes en los proyectos que pueden evitarse o disminuirse utilizando metodologías de Gestión

- ▶ Falta de coordinación de recursos y actividades
- ▶ Productos finales que no representan lo que el cliente esperaba
- ▶ Proyectos que terminan fuera de tiempo y consumen más dinero del planificado
- ▶ Planificación inadecuada de recursos y actividades
- ▶ Falta de conocimiento del estado real del proyecto
- ▶ Entregables inutilizables o con calidad inaceptables
- ▶ Alcance del proyecto poco definido y falta de gestión de los cambios al mismo.

Estándares

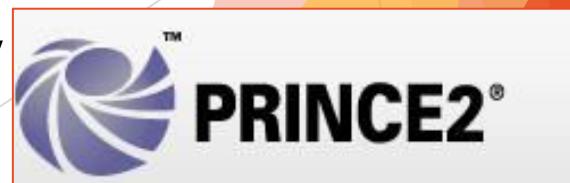
- *A Guide to Project Management Body Of Knowledge* representa el estándar para la gestión del proyectos del Project Management Institute de EE.UU., mundialmente reconocido

<http://www.pmi.org/>



- **PRINCE2** es un método de gestión de proyectos intensamente utilizado por el gobierno y el sector privado del Reino Unido y extendido internacionalmente

<http://www.prince-officialsite.com/>



Definición de Proyecto

Según el PMI

- ▶ Es un esfuerzo **temporal** comprometido con la creación de un producto o servicio de resultado **único**

Temporal significa que un proyecto tiene principio y fin definidos. El proyecto termina cuando ha alcanzado sus objetivos o resulta claro que no pueden alcanzarse o bien la necesidad del proyecto ya no existe.

Un proyecto genera productos, servicios o resultados únicos

La elaboración progresiva es una característica de los proyectos que acompaña a la temporalidad y unicidad. Esto significa avanzar en pasos e incrementos continuos

Definición de Proyecto

Según PRINCE

- ▶ Es una *organización* temporal creada con el propósito de entregar uno o más *productos* de acuerdo con un *caso de negocio* especificado

Foco en la justificación del **negocio**

Estructura organizativa definida por el **equipo de gestión** de proyecto

Enfoque de planificación basada en **producto**

Énfasis en la división del proyecto en **etapas** gestionables y controlables

Metodología flexible para su aplicación en un nivel apropiado al proyecto

ENFOQUE A MEDIDA

- ▶ Cuanto mayor sea la Incertidumbre y Complejidad del proyecto:
 - Mayor será la necesidad de una gestión flexible y adaptable.
 - Una planificación menos rígida, más dinámica, con permanente re-planificación.
- ▶ La PMO (Oficina de proyectos) debe promover el estilo de gestión del proyecto que se adapte a las particularidades del proyecto, a sus necesidades y requerimientos



“TALLE ÚNICO” VS “A MEDIDA”

PM Talle Único		PM A Medida
Objetivo	Triple Restricción	Resultados al negocio, cumpliendo múltiples criterios
Planificación	Se planifica una vez al inicio	Plan inicial + re-planificación cuando es necesario
Enfoque de Dirección	Rígido, enfocado en el plan inicial	Flexible, cambiante, adaptativo
Trabajo del proyecto	Predecible, conocido, lineal, simple	Impredecible, incierto, no lineal, complejo
Control del Proyecto	Busca desvíos respecto al plan y toma acciones para alinearlos	Identifica cambios en el entorno y ajusta el plan de acuerdo al entorno
Metodología	Todos los proyectos siguen la misma metodología	Adaptada a la complejidad e incertidumbre del proyecto

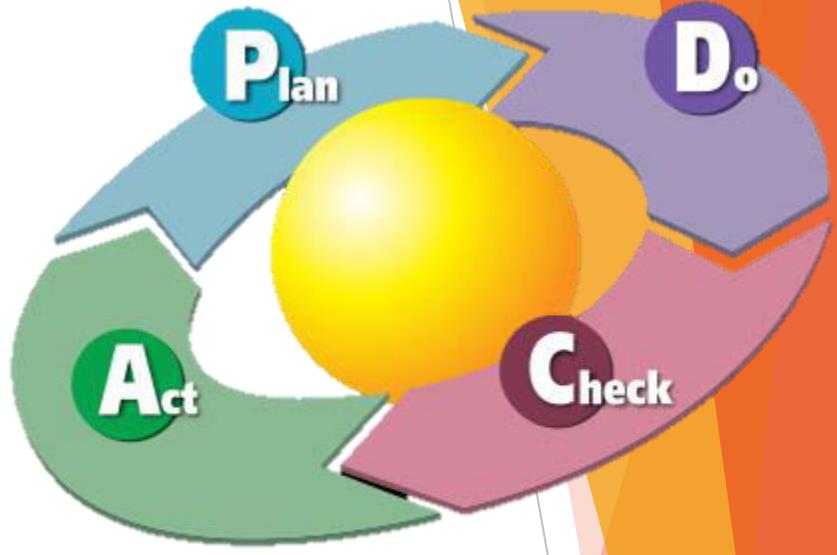
Fases

Ciclo PDCA (Plan, Do, Check, Act)

- ▶ Existe más de un modo de gestionar un proyecto, basadas en la definición de grupos de procesos, que constituyen guías para aplicar los conocimientos y habilidades apropiados durante el proyecto.
- ▶ Junto con esa variedad existe un concepto común subyacente para la interacción entre los procesos de gestión de proyecto: el ciclo Plan-Do-Check-Act.
- ▶ Creado por *Walter A. Shewhart* y perfeccionado por *William Edwards Deming* (1900-1993) para la mejora continua de la calidad.

Ciclo PDCA

- ▶ **Plan:** identificar y analizar el problema
- ▶ **Do:** elaborar e implementar una solución
- ▶ **Check:** evaluar los resultados
- ▶ **Act:** estandarizar la solución y capitalizarla en nuevas oportunidades



Fases de un proyecto

- ▶ Si bien los procesos involucrados en la gestión de un proyecto suelen extenderse a lo largo de todo su ciclo de vida, resulta conveniente dividirlos en segmentos temporales por razones técnicas o de gestión a las que suele llamárseles *fases*.
- ▶ Estas fases varían según el método de gestión utilizado y además pueden presentar particularidades en un proyecto determinado
- ▶ Una de las funciones del PM de un proyecto es saber determinar cuáles de estos procesos se utilizarán según el tipo de proyecto gestionado.
- ▶ Estos procesos a su vez se dividen por áreas de conocimiento, que varían también según la metodología.

Fases de un proyecto

Pre- Proyecto

- Se trata de revisar nuestra idea a fin de proveer la información necesaria para decidir si comenzamos el proyecto. Determinar la factibilidad técnica y económica.

Inicio

- Se trata de que todos los involucrados comprendan qué producirá el proyecto, cuándo, con qué costo y con qué calidad, que se involucren con ello y que se elabore un plan que permita realizarlo

Ejecución

- Se trata de elaborar los paquetes de trabajo del producto o servicio, controlar el flujo de trabajo de los equipos, gestionar riesgos y problemas y monitorear e informar el avance del proyecto

Cierre

- Se trata de comprobar que todo se ha realizado, determinar qué debe ser pasado a la organización e informar cómo ha finalizado el proyecto

Ejemplo PMI: Procesos de Comunicación y Costo por fases

Fase / Áreas de Conocimiento	P. De Inicio	P. de Planificación	P. de Ejecución	P. De Control y monitoreo	P. De Cierre
GESTIÓN DEL ALCANCE	-	<ul style="list-style-type: none">•Recopilar requerimientos•Definir Scope•Crear WBS	-	<ul style="list-style-type: none">•Verificar Scope•Controlar Scope	-
GESTIÓN DE LAS COMUNICACIONES	Identificar interesados	Plan de comunicaciones	<ul style="list-style-type: none">•Distribuir información•Gestionar las expectativas de los interesados	Report performance	

Riesgos y Problemas

Riesgo

- ▶ *Un riesgo es un evento o condición incierta que, si sucede, tiene un efecto en por lo menos uno de los objetivos del proyecto.* - Project Management Body Of Knowledge, PMBOK

- ▶ *Los riesgos están asociados en la incertidumbre que está presente en todos los proyectos.*

- ▶ *Existen riesgos “conocidos”, aquéllos que identificamos y analizamos y para los cuáles podemos planificar respuestas.*

- ▶ *Existen otros riesgos “desconocidos” que no podemos gestionar de manera proactiva, pero para lo cuál debemos crear un plan de contingencia.*



Atributos del Riesgo



Es la posibilidad de que el riesgo se materialice.

Es el resultado de la materialización del riesgo

Es el producto de la probabilidad por el impacto

- *Una vez que determinemos la severidad de todos los riesgos debemos determinar cuáles gestionaremos (en general los de severidad alta y media)*

Enunciado del Riesgo

- ▶ Si [evento o condición incierta expresado en presente], entonces [impacto del riesgo expresado en futuro]
- ▶ Ejemplo: Si el proveedor X no entrega el server para la base datos en la fecha acordada, entonces se incrementará significativamente el costo de las pruebas del módulo M

Gestión del Riesgo

- ▶ Si no es posible evitar la aparición de un riesgo, será posible gestionarlo
- ▶ La gestión de riesgos es un proceso sistemático que involucra:
 - ▶ **Identificación:** reconocimiento de las fuentes de riesgo y sus consecuencias potenciales
 - ▶ **Análisis:** determinación de la necesidad de tratamiento del riesgo y la prioridad de su implementación
 - ▶ **Tratamiento o respuesta:** selección de opciones para actuar sobre el riesgo y la implementación de las mismas
 - ▶ **Monitoreo y revisión:** evaluación del progreso en la implementación del tratamiento

Tratamiento o respuesta a Riesgos

- ▶ **Evitar:** implica eliminar por completo la amenaza.
Asegurar que la amenaza no podrá ocurrir o que no tendrá efecto en el proyecto.
Ej: reemplazar una tarea del proyecto por otra que no implique un riesgo.
- ▶ **Transferir:** trasladar a un tercero todo o parte del impacto negativo de una amenaza.
La transferencia de un riesgo simplemente confiere a una tercera persona la responsabilidad de su gestión; no lo elimina.
Ej: Seguros

Tratamiento o respuesta a Riesgos

- ▶ **Mitigar:** implica reducir a un umbral aceptable la probabilidad y/o el impacto de un evento adverso.
Adoptar acciones tempranas para reducir la probabilidad de ocurrencia de un riesgo y/o su impacto sobre el proyecto, a menudo es más efectivo que tratar de reparar el daño después de ocurrido el riesgo.
Ej: equipos redundantes ante posibles caídas del servicio.
- ▶ **Aceptar o asumir:** se asume que el riesgo se manifestará y se decide no tomar acción.
Esto ocurre debido a que es muy costoso tomar acción sobre el o no se pudo identificar ninguna estrategia de respuesta posible.
Ej: Terremotos en sitios donde no es habitual su ocurrencia.

Estrategias: Nivel de Riesgo/ Habilidad para Gestionarlo

Nivel de Riesgo

Alto

- ¿Tenemos claros los riesgos del proyecto?
- ¿Comprendemos donde impactan?
- ¿La medición de los riesgos es confiable?
- ¿Nuestras capacidades de gestión son reales?
- ¿Son las necesarias para esos riesgos?

¿Realmente debemos hacer este proyecto?
Medir muy bien los riesgos, donde van a impactar y sus potenciales efectos

¿Hay alguna capacidad que podamos adquirir para mejorar nuestra gestión?

Bajo

- ¿Nuestras estimaciones son reales o están sesgadas?
- ¿No habremos subestimado los riesgos o sobreestimado nuestras capacidades?

¿Nuestras mediciones de los riesgos son realistas?

¿Hay alguna capacidad que podamos adquirir para mejorar nuestra gestión?

Alto

Bajo

Capacidad
de Gestión

Problema

- ▶ *Evento o condición esperada o no que afecta negativamente los objetivos de un proyecto*
- ▶ Un problema inesperado puede ser una modificación de una regulación estatal que impida que un proveedor nos entregue un producto o servicio
- ▶ Un problema esperado puede ser un riesgo que se materializa (deja de ser una posibilidad para transformarse en un hecho)

Gestión de Problemas

- ▶ A diferencia de los riesgos -que representan incertidumbre-, los problemas son hechos sobre los que debe actuar para evitar o minimizar consecuencias negativas sobre los objetivos
- ▶ En forma similar a lo visto en riesgos, los problemas se gestionan con un proceso cuyas etapas son:
 - Registro
 - Evaluación
 - Resolución
 - Monitoreo

Seguimiento y Control

¿Porqué controlar y monitorear?

- ▶ “Cuando se puede medir aquello de lo que se está hablando y expresarlo numéricamente se sabe algo acerca de ello, pero cuando no se lo puede medir, su conocimiento es escaso e insatisfactorio” [Lord Kelvin: “Electrical Units of Measurement”, 1883].
- ▶ “No se pueden controlar lo que no se puede medir” [Tom DeMarco: “Controlling Software Projects: Management, Measurement, and Estimation”, 1982]
(*)

¿Cómo controlar y monitorear?

- ▶ Dada la evolución progresiva de los proyectos, necesitamos realizar mediciones para determinar su grado de avance
- ▶ Comparando el grado de avance medido con el esperado verificaremos si se observan desvíos significativos respecto de lo planificado. Si ese resulta el caso, tomaremos acciones correctivas
- ▶ La información de avance del proyecto (incluye alcance, cronograma, costos, calidad, equipo de trabajo, riesgos y problemas) y el análisis de la misma se vuelca en un informe de seguimiento que es tratado por la dirección del proyecto

¿Proyectos exitosos?

¿Cuándo se considera exitoso un proyecto?

- ▶ ¿Cuando terminan en Fecha?
- ▶ ¿Cuando no tiene sobrecostos?
- ▶ ¿Cuando no tiene modificaciones de alcance?
- ▶ ¿Cuando no presenta problemas?

NO!! *Un proyecto Exitoso es el que contribuye al éxito de la organización.*

Debemos redefinir el concepto de éxito y considerar....

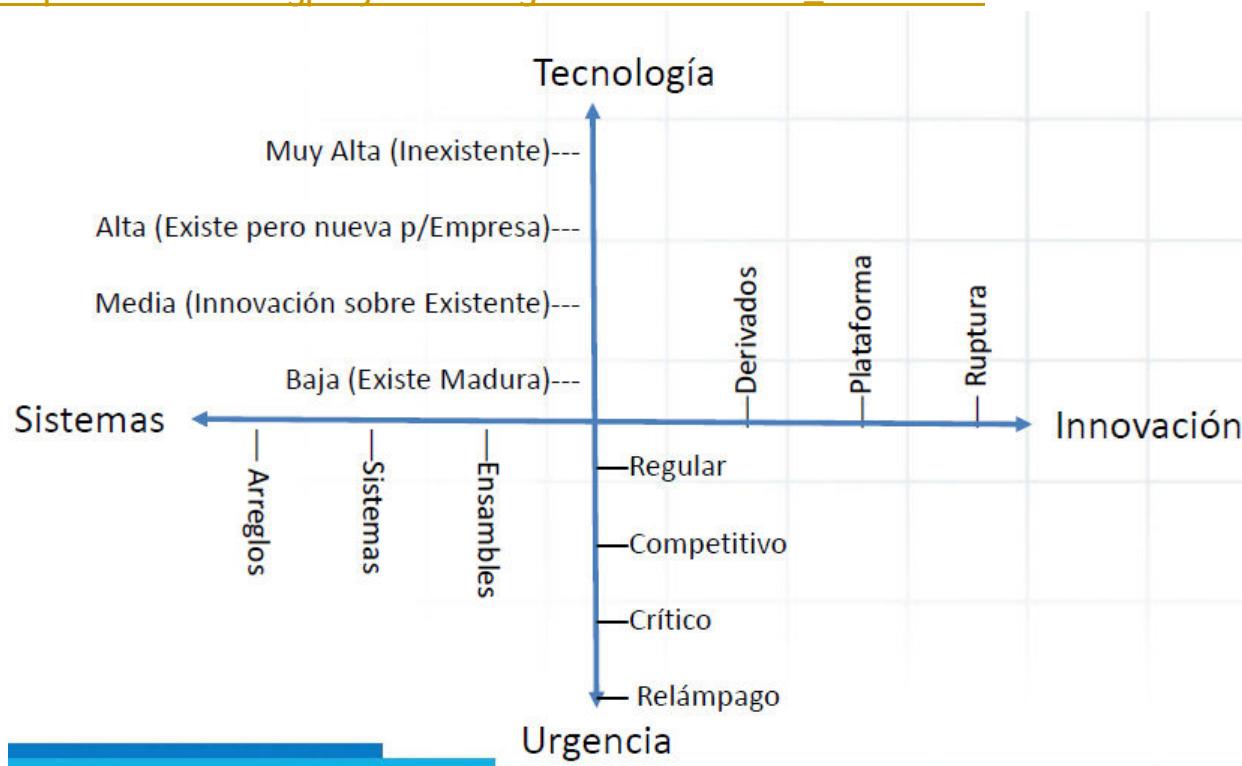
- ▶ La eficiencia del proyecto
- ▶ Impacto en el cliente
- ▶ Impacto en el equipo
- ▶ El negocio y el éxito directo
- ▶ Preparación para el futuro

Complejidad de los proyectos

Para evaluar la Complejidad de los proyectos debemos considerar que la mayoría de los proyectos modernos son inciertos, complejos y cambiantes y que están afectados fuertemente por su entorno, por la tecnología, la prisa de los mercados y la innovación.

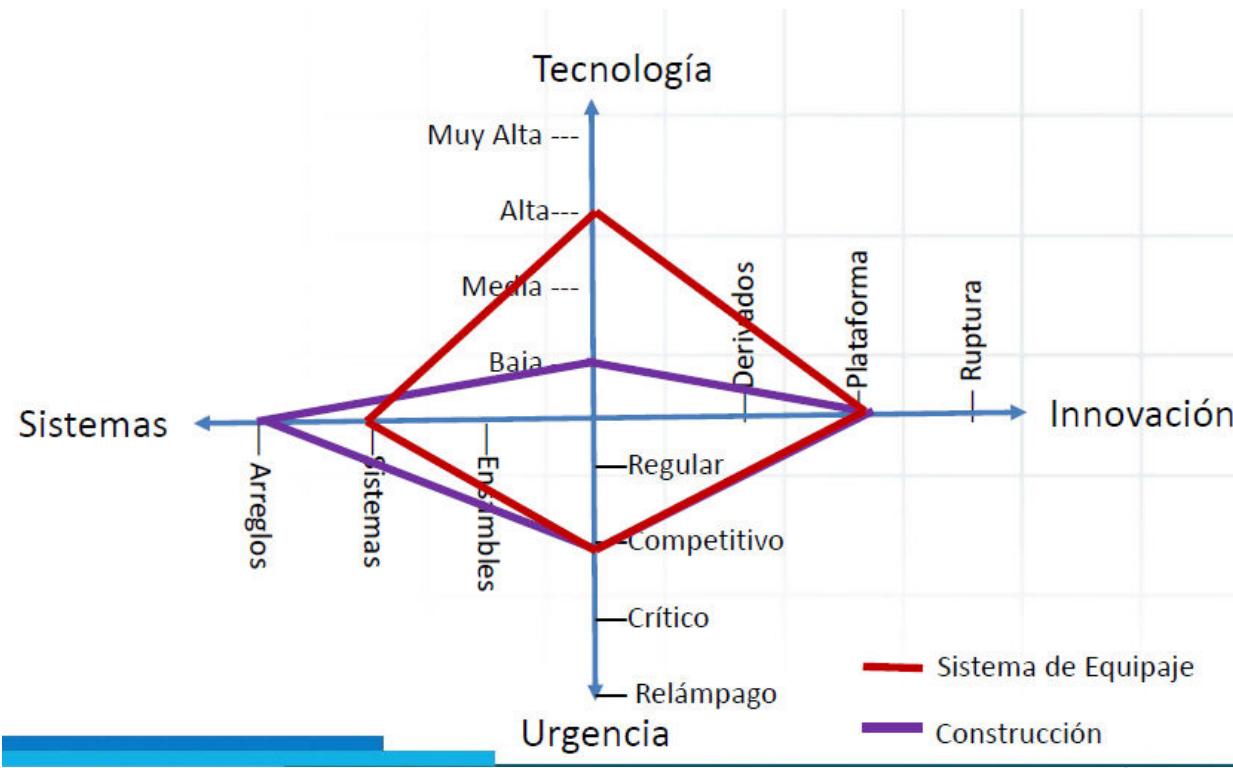
Enfoque del Diamante

http://reinventingprojectmanagement.com/010_APM.html



CASOS

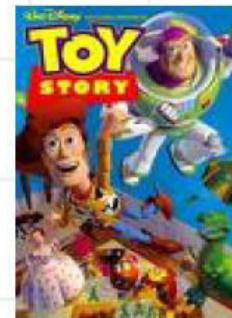
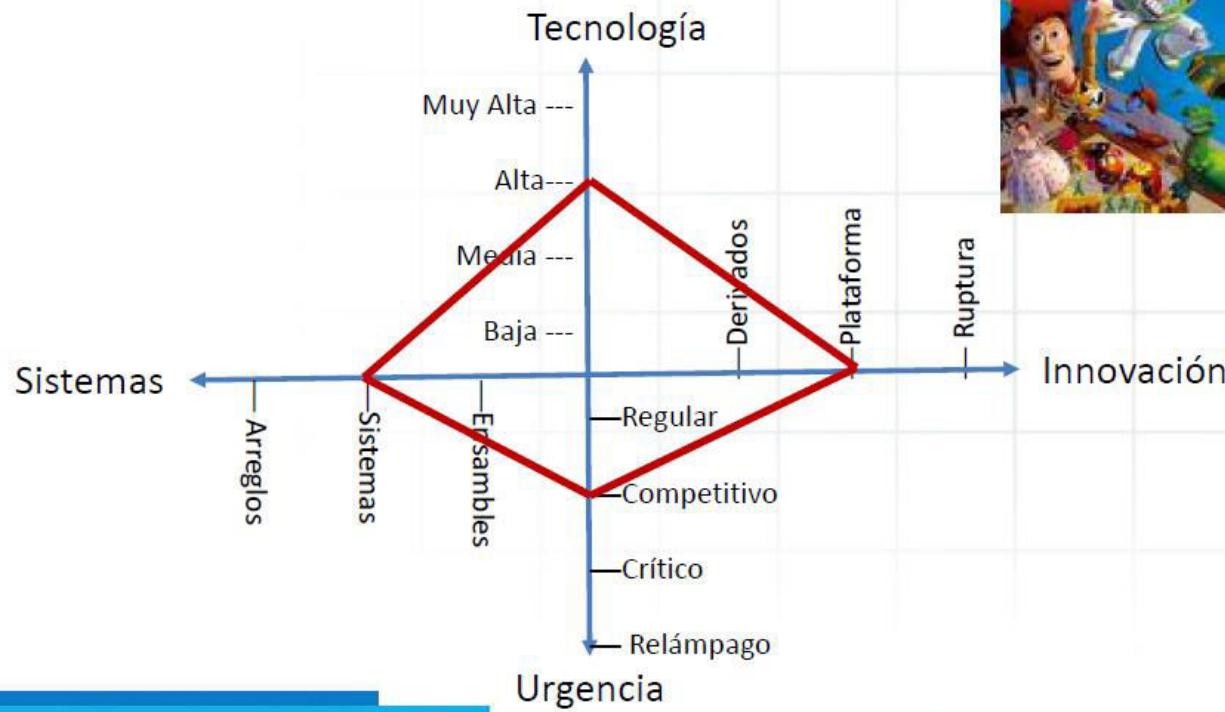
PROYECTO AEROPUERTO DENVER



CASOS

TOY STORY

Proyecto: Toy Story



ALCANCE Y PLANIFICACIÓN

2020

Administración de Recursos

ALCANCE

¿Qué es el alcance de un proyecto?

Es la definición exacta y unívoca de todo lo que estará (y lo que no) comprendido dentro del proyecto a ejecutar. Proporciona un entendimiento común entre los interesados del mismo.

Su adecuada definición garantiza que si todo se ejecuta conforme a las especificaciones, entonces los entregables y el resultado final serán completamente satisfactorios para el cliente

ALCANCE DE PRODUCTO VS ALCANCE DE PROYECTO

Utilidad del Documento de Alcance

- Guía del equipo de trabajo durante la ejecución
- Proporciona la línea base para evaluar si las solicitudes de cambio que puedan surgir se encuentran dentro o fuera de los límites establecidos
- Base para la estimación de esfuerzo y duración del proyecto
- Input para la fase de detalle de trabajo/tareas a realizar (EDT)

ALCANCE DE PRODUCTO VS ALCANCE DE PROYECTO

REQUERIMIENTOS

TIPOS DE REQUERIMIENTOS

FUNCIONALES: describen qué es lo que el sistema debe hacer. Establecen las funciones que el producto de software tiene que incluir. Surgen de la descripción de los problemas o necesidades obtenidas durante la etapa de adquisición de requerimientos del proceso de ingeniería de requerimiento. El equipo de desarrollo no genera los requerimientos funcionales, sino que su función es transformar la información obtenida en requerimientos funcionales, para que luego sean desarrollados.

NO FUNCIONALES: son las restricciones a las que está sometido el producto de software a desarrollar. Las restricciones influyen sobre el funcionamiento o sobre el proceso de desarrollo de software.

REQUERIMIENTOS

CARACTERÍSTICAS DE LOS REQUERIMIENTOS

Según la IEEE-830 un requerimiento es válido si cumple con las siguientes características:

CARACTERÍSTICA	DESCRIPCIÓN
NECESARIO	Es necesario si su omisión provoca una deficiencia en el producto.
CONCISO	Fácil de leer y entender, de redacción simple y clara.
COMPLETO	Proporciona la información suficiente para ser comprendido.
CONSISTENTE	No es contradictorio con otro requerimiento.
NO DEBE SER AMBIGUO	Tiene una sola interpretación. Su definición no causa confusiones.
VERIFICABLE	Puede ser cuantificado a través de inspección, pruebas, análisis.

COMPONENTES DEL DOCUMENTO DE ALCANCE

- Descripción clara y unívoca del objetivo de proyecto
- Justificación del proyecto
- Listado de todos los requerimientos
- Listado y descripción de los entregables del proyecto
- Definición clara de los límites del proyecto
- Descripción de los supuestos del proyecto
- Descripción de las restricciones del proyecto
- Hitos del proyecto
- Dependencias con otros proyectos

VERIFICAR EL ALCANCE

- Verificar el Alcance es el proceso que consiste en formalizar la aceptación de los entregables del proyecto que se han completado
- Verificar el alcance incluye revisar los entregables con el cliente o el sponsor para asegurarse que se han completado satisfactoriamente y para obtener de ellos su aceptación formal
- Ocurre al final de cada fase y al final del proyecto

EDT (Estructura de desglose de trabajo)

Consiste en una descomposición jerárquica del trabajo orientada a entregables.

El último nivel se denomina “Paquete de trabajo” y es que deberá figurar en nuestra planificación con su esfuerzo asociado.

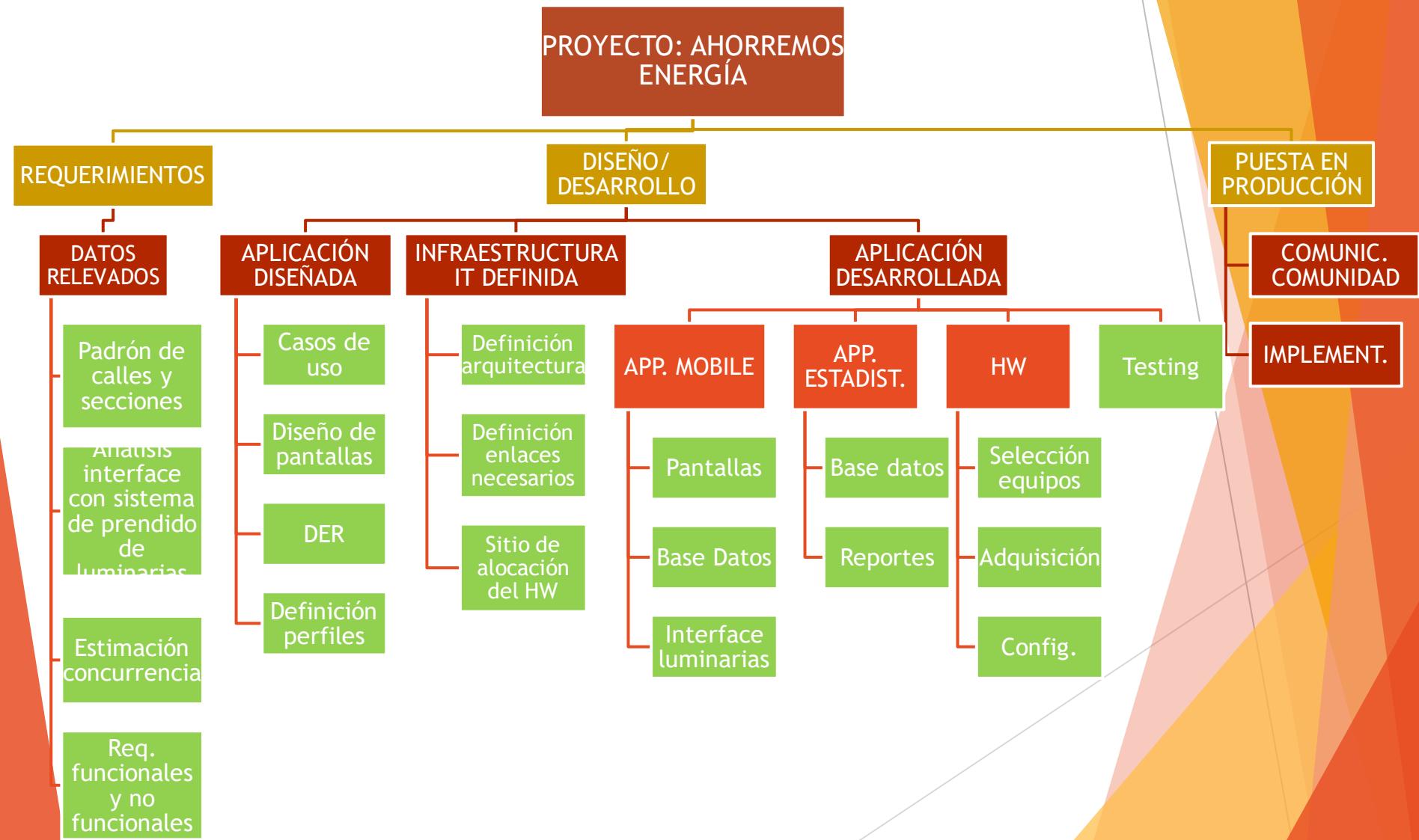
Su objetivo es organizar y definir el alcance total del proyecto.

No existe una regla en cuánto a la cantidad de niveles que debe contener, ni todos los entregables deben respetar la misma cantidad de niveles.

Se mencionarán entregables y subentregables en cada nivel hasta que las tareas que impliquen ese entregable permitan:

- Identificar el grado de avance durante la ejecución.
- Estimar su duración.
- Identificar a una persona o grupo responsable.
- Estimar su costo.

EDT (EJEMPLO)



PLANIFICACIÓN

Pasos para desarrollar una planificación

- ✓ **Definir actividades:** es el proceso de identificar las acciones específicas a realizar para producir los entregables del proyecto.
- ✓ **Secuenciar las actividades:** es el proceso de identificar y documentar las relaciones entre las actividades del proyecto
- ✓ **Estimar recursos de las actividades:** es el proceso de estimar el tipo y cantidad de materiales, personas, equipamiento o suministros requeridos para realizar cada actividad.
- ✓ **Estimar la duración de las actividades:** es el proceso de estimar el número de periodos de tiempos laborales necesarios para completar individualmente las actividades con los recursos estimados.
- ✓ **Desarrollar el cronograma:** es el proceso de analizar la secuencia de actividades, duraciones, necesidades de recursos y las restricciones de cronograma para crear el cronograma del proyecto.

PLANIFICACIÓN: Definir actividades

Herramientas y técnicas para definir actividades

Descomposición

Se utiliza de forma similar a como se lo hace en el desglose de la EDT.

El objetivo aquí es descomponer los paquetes de trabajo en Actividades/Entregables. Las actividades muestran el esfuerzo necesario para completar el paquete de trabajo.

Planificación Gradual

Es una forma de planificación mediante elaboración gradual, donde se planifica en detalle el trabajo que debe desarrollarse en el corto plazo y el trabajo futuro se planifica a un nivel superior de la EDT.

Juicio de Expertos

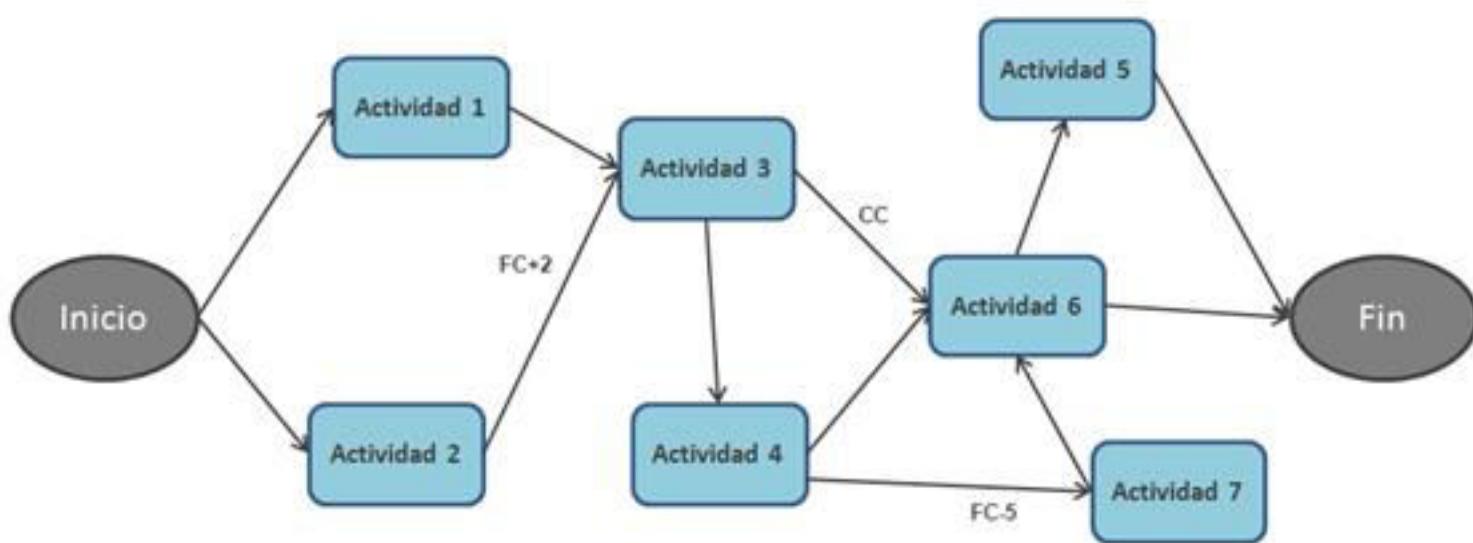
Los interesados en el proyecto pueden proveer información sobre las actividades del proyecto basados en su experiencia, habilidades y conocimientos

PLANIFICACIÓN: Secuenciar actividades

Método de Diagramación por Precedencia (PPM)

El diagrama de precedencias (PDM) utiliza círculos o rectángulos, denominados nodos, que representan las actividades, así como flechas que interconectan esos nodos y que representan las relaciones lógicas.

Esta técnica también se conoce como actividad en el nodo (AON)



PLANIFICACIÓN: Estimar recursos

Herramientas y técnicas para estimar los recursos de las actividades

Juicio de Expertos
Se utiliza para evaluar las entradas a este proceso.
Todo grupo o persona con conocimientos especializados puede aportar dicha experiencia.

Análisis de alternativas
Es común que una tarea particular pueda ser ejecutada mediante diferentes combinaciones de recursos. El análisis de alternativas consiste en encontrar la mejor forma de completar las actividades mediante la combinación de recursos

Datos de estimación publicados
Índices de producción actualizados y costos unitarios.
Extensa variedad de industrias, materiales y equipos. Diferentes países y ubicaciones geográficas.

Estimación ascendente
Esta forma de estimación consiste en descomponer una actividad o paquete de trabajo en mayor detalle para poder estimarla. Una vez que el detalle de la descomposición está disponible, se suman entre ellos de manera de obtener una estimación más confiable

PLANIFICACIÓN: Estimar duración

Herramientas y técnicas para estimar la duración de las tareas

Juicio de Expertos
Este juicio guiado por información histórica debe usarse cuando sea posible. Miembros individuales del equipo también pueden aportar. Si no se cuenta con esto, las estimaciones son inciertas y arriesgadas.

Estimación análoga (Top-Down)

Se utiliza en los comienzos del proyecto, cuando aún se cuenta con poca información. Utiliza el costo/tiempo real de un proyecto previo y similar, como base para la estimación del costo/tiempo del proyecto actual. Los proyectos deben ser similares realmente (no sólo en apariencia).

Estimación por 3 valores

Usa un promedio ponderado de estimaciones para calcular la duración de la actividad. Se basa en 3 valores (mas probable, optimista y pesimista)

PLANIFICACIÓN: Desarrollar cronograma

Una vez que se cuenta con la descripción de las tareas, su duración, sus recursos resta organizar su ejecución teniendo en cuenta sus relaciones, el orden de ejecución y los hitos que tendrá el proyecto.

Es importante saber cuál es el camino crítico de nuestro proyecto, y así poder estimar el tiempo más corto en el que es posible completarlo

Desarrollar cronograma: GANTT

El Gantt es una herramienta que nos permite modelar la planificación de tareas del proyecto.

Permite realizar una representación gráfica del progreso del proyecto.

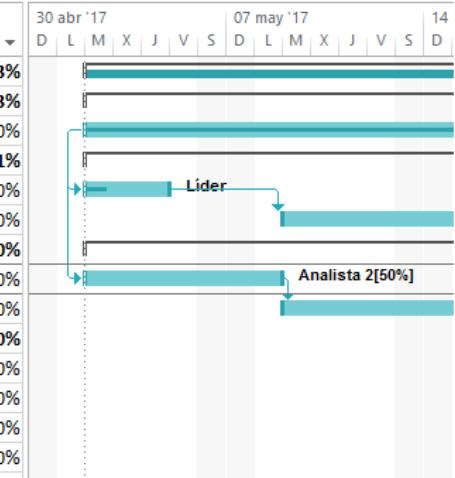


Nombres de los recursos (Responsables de las tareas):

- 1 responsable
- + de 1 responsable, separador por “,”, distribución equitativa
- + de 1 responsable, distribución desigual con [%]

Desarrollar cronograma: GANTT

	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos	% completa
1	▪ Proyecto: Ahorremos energía	40 días	mar 02/05/17	lun 26/06/17			33%
2	▪ Definición de requerimientos	22 días	mar 02/05/17	mié 31/05/17			33%
3	✓ Padrón de calles y secciones	15 días	mar 02/05/17	lun 22/05/17		Analista 1[30%],PM[70%]	100%
4	▪ Análisis interface con sistema luminarias	10 días	mar 02/05/17	lun 15/05/17			11%
5	Contactar proveedor y solicitar documentación	3 días	mar 02/05/17	jue 04/05/17	3CC	Líder	30%
6	Definir información input/output	5 días	mar 09/05/17	lun 15/05/17	5FC+2 días	Analista 2[50%]	0%
7	▪ Estimación concurrencia	10 días	mar 02/05/17	lun 15/05/17			0%
8	Analizar promedio de usuarios concurrentes	5 días	mar 02/05/17	lun 08/05/17	3CC	Analista 2[50%]	0%
9	Establecer picos y horarios	5 días	mar 09/05/17	lun 15/05/17	8	Analista 2[50%]	0%
10	▪ Requerimientos funcionales y no funcionales	7 días	mar 23/05/17	mié 31/05/17			0%
11	Definir requerimientos funcionales obligatorios	7 días	mar 23/05/17	mié 31/05/17	3,4,7	Analista 1[50%],Líder[50%]	0%
12	Definir requerimientos funcionalesopcionales	5 días	mar 23/05/17	lun 29/05/17	11CC	Analista 1[50%],Líder[50%]	0%
13	Definir requerimientos no funcionales	3 días	mar 23/05/17	jue 25/05/17	11CC	Analista 1[50%],Líder[50%]	0%
14	Requerimientos relevados	0 días	mié 31/05/17	mié 31/05/17	3,4,7,10		0%

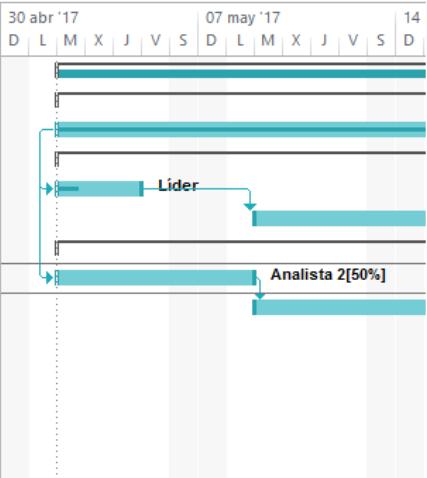


Precedencias:

- Fin a comienzo (FC): La tarea dependiente (B) no puede comenzar hasta que se haya completado la tarea de la que depende (A)
- Comienzo a comienzo (CC): La tarea dependiente (B) no puede comenzar hasta que comience la tarea de la que depende (A)
- Fin a fin (FF): La tarea dependiente (B) no se puede completar hasta que se haya completado la tarea de la que depende (A)
- Comienzo a fin (CF): La tarea dependiente (B) no se puede completar hasta que comience la tarea de la que depende (A)

Desarrollar cronograma: GANTT

	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos	% completa
1	▫ Proyecto: Ahorremos energía	40 días	mar 02/05/17	lun 26/06/17			33%
2	▫ Definición de requerimientos	22 días	mar 02/05/17	mié 31/05/17			33%
3	✓ Padrón de calles y secciones	15 días	mar 02/05/17	lun 22/05/17		Analista 1[30%],PM[70%]	100%
4	▫ Análisis interface con sistema luminarias	10 días	mar 02/05/17	lun 15/05/17			11%
5	Contactar proveedor y solicitar documentación	3 días	mar 02/05/17	jue 04/05/17	3CC	Líder	30%
6	Definir información input/output	5 días	mar 09/05/17	lun 15/05/17	5FC+2 días	Analista 2[50%]	0%
7	▫ Estimación concurrencia	10 días	mar 02/05/17	lun 15/05/17			0%
8	Analizar promedio de usuarios concurrentes	5 días	mar 02/05/17	lun 08/05/17	3CC	Analista 2[50%]	0%
9	Establecer picos y horarios	5 días	mar 09/05/17	lun 15/05/17	8	Analista 2[50%]	0%
10	▫ Requerimientos funcionales y no funcionales	7 días	mar 23/05/17	mié 31/05/17			0%
11	Definir requerimientos funcionales obligatorios	7 días	mar 23/05/17	mié 31/05/17	3,4,7	Analista 1[50%],Líder[50%]	0%
12	Definir requerimientos funcionalesopcionales	5 días	mar 23/05/17	lun 29/05/17	11CC	Analista 1[50%],Líder[50%]	0%
13	Definir requerimientos no funcionales	3 días	mar 23/05/17	jue 25/05/17	11CC	Analista 1[50%],Líder[50%]	0%
14	Requerimientos relevados	0 días	mié 31/05/17	mié 31/05/17	3,4,7,10		0%



Avance:

- % trabajo completado: avance asociado al tiempo incurrido
- % físico completado: avance asociado al trabajo real ejecutado

Desarrollar cronograma: HITOS Y ENTREGABLES

ENTREGABLE: objeto tangible producido como resultado del proyecto

HITO: Punto o evento relevante del proyecto. Simboliza el haber conseguido un logro importante en el proyecto.

Ambos tienen duración 0, pero pueden tener o no responsable asignado.

Los hitos están totalmente ligados a los entregables. La entrega a tiempo de entregables es la evidencia que indica el éxito de un hito.

Un entregable difiere de un hito del proyecto en que el hito es una medida de progreso hacia el resultado final del proyecto, mientras que el entregable es el resultado del proceso.

EJEMPLOS:

Entregables:

- * DER
- * Selección equipos

Hitos:

- * Relevamiento finalizado
- * Comunicación a la comunidad

Desarrollar cronograma: GANTT

LINEA BASE: Al final de la planificación debemos definir la línea de base del proyecto como una fotografía del cronograma.

Una vez que el proyecto se empiece a ejecutar, usaremos esta línea de base para comparar el desempeño.

En la ejecución imprimiremos la línea de base y el plan actual, y los compararemos para descubrir los desvíos.

Es posible que ante cambios en la planificación (incorporación de requerimientos, cambios en las estimaciones, etc) sea necesario volver a establecer una nueva línea base que refleje los cambios impactados en el plan. Esto sólo debería aplicarse con gestión de cambios, analizando y aprobando los cambios que impacten en la planificación,

Desarrollar cronograma: GANTT

	Nombre del recurso	Trabajo	Detalles	S	21 may '17	D	L	M	X	J	V	S	28 may '17	D	L	M
	▪ Sin asignar	0 horas	Trabaja													
	▪ Requerimientos relevados	0 horas	Trabaja													
1	⚠️ ▪ Analista 1	101,6 horas	Trabaja			2,4h	12h	12h	12h	8h					8h	4h
	▪ Padrón de calles y secciones	41,6 horas	Trabaja			2,4h										
	▪ Definir requerimientos funcionales obligatorios	28 horas	Trabaja				4h	4h	4h	4h					4h	4h
	▪ Definir requerimientos funcionalesopcionales	20 horas	Trabaja				4h	4h	4h	4h					4h	4h
	▪ Definir requerimientos nofuncionales	12 horas	Trabaja				4h	4h	4h							
2	▪ PM	78,4 horas	Trabaja			5,6h										
	▪ Padrón de calles y secciones	78,4 horas	Trabaja			5,6h										
3	⚠️ ▪ Líder	84 horas	Trabaja				12h	12h	12h	8h					8h	4h
	▪ Contactar proveedor y solicitar documentación	24 horas	Trabaja													
	▪ Definir requerimientos funcionales obligatorios	28 horas	Trabaja				4h	4h	4h	4h					4h	4h
	▪ Definir requerimientos funcionalesopcionales	20 horas	Trabaja				4h	4h	4h	4h					4h	4h
	▪ Definir requerimientos nofuncionales	12 horas	Trabaja				4h	4h	4h							
4	▪ Analista 2	60 horas	Trabaja													
	▪ Definir información input/output	20 horas	Trabaja													
	▪ Analizar promedio de usuarios concurrentes	20 horas	Trabaja													

Alarma de recurso
sobreasignado

Periodo
sobreasignado

Horas totales
asignadas

Administración de Recursos

Glosario de Conceptos de IT

Infraestructura IT

¿Qué significa?

El término Infraestructura de TI es definido en ITIL 4 como el conjunto de hardware, software, redes e instalaciones que se requieren para desarrollar, probar, entregar, monitorear, administrar y dar soporte a los servicios de TI. Las personas asociadas, procesos y documentación no son parte de la Infraestructura de TI.

Elementos que conforman la Infraestructura IT:

- Switches
- Routers
- Firewalls
- Servers
- Planta Física
- Personal (RRHH)
- Datacenter
- Servicios
- Software
- Entre muchos otros

¿Cuáles son los beneficios de la infraestructura IT?

Actualmente las empresas requieren de utilizar la tecnología como una herramienta básica para desarrollar sus actividades, contar con una correcta infraestructura IT traerá importantes beneficios para la organización, tales como:

- Elevar la productividad empresarial
- Reducir costos operativos
- Disminuir los riesgos de fallas
- Elevar la seguridad de la información
- Tener una mejor capacidad de respuesta
- Facilidad de adaptación

Administración de Recursos

Glosario de Conceptos de IT

-Alojar una mayor cantidad de datos

-Elevar la competitividad del negocio

-Tener una mejor organización

La infraestructura IT para las empresas también puede ser otorgada por servicios externos que cuenten con todo el soporte técnico necesario para brindar un servicio de calidad.

¿Cuáles son los desafíos que enfrenta la gestión de servicios de IT en cuanto a infraestructura?

Uno de los grandes obstáculos para brindar servicios de IT que agreguen valor y en forma ágil solía ser la rigidez de las soluciones de infraestructura. Podía (y puede en algunos contextos) tomar meses el despliegue de la infraestructura necesaria para utilizar un nuevo software, lo cual volvía irrelevante la agilidad aplicada en el desarrollo para quien consumiría el servicio.

Este problema se ha resuelto, en gran medida, con la evolución de la tecnología. Virtualización, mayor ancho de banda y cloud computing han permitido que las organizaciones traten a la infraestructura de IT como un servicio o como código proveyéndole una velocidad de cambio que antes solo era posible para el software.

Fuentes:

<https://vegagestion.es/la-infraestructura-tecnologica-definicion-tipos-e-importancia/>

<http://www.icorp.com.mx/blog/infraestructura-de-ti-componentes/>

<https://www.redalyc.org/pdf/993/99330402008.pdf>

<https://prezi.com/jdivfsjx0e0t/infraestructura-de-ti-y-tecnologia-de-informacion/>

<https://www.nsit.com.co/infraestructura-ti/>

ITIL Foundation: ITIL 4 Edition, 2019, AXELOS

Administración de Recursos

Glosario de Conceptos de IT

CAPEX - OPEX - TCO

CAPEX y OPEX: ¿Qué son?

El acrónimo CAPEX se deriva de la expresión “Gastos de Capital” y, por lo tanto, se desarrolla a los gastos e inversiones asociados con bienes. En otras palabras, es todos los bienes comprados por la empresa.

OPEX, por otro lado, significa “Gasto Operativo”. Por lo tanto, se relaciona al costo relacionado con las operaciones y servicios.

En términos generales, comprar un automóvil para una empresa sería considerado como un CAPEX. Por otro lado, un gasto único por servicios de transporte(acarreo) se clasificaría como OPEX.

Una cuidadosa distinción entre estos es una buena forma de definir y analizar el KPIs de su negocio, ya que esto ofrece una visión más profunda de los gastos de la empresa, lo que ayuda al control financiero.

CAPEX y OPEX: ¿Cuáles son las diferencias?

En general, la mayoría de los costos anuales de una corporación son gastos operativos. Por lo tanto, la reducción del OPEX debe ser uno de los objetivos de la administración, siempre que esto no comprometa la calidad de los productos y/o servicios que ofrece.

Un punto que debe enfatizarse es la diferencia entre la forma en que se gravan estos dos tipos de gastos. Como la vida de un CAPEX generalmente se extiende más allá de un año fiscal, se debe usar la amortización y la depreciación para redistribuir este costo. Por el contrario, los gastos operacionales se pueden deducir de sus impuestos durante el año fiscal en que tienen lugar.

Dominar estos dos conceptos es fundamental para la planificación estratégica de una empresa, ya que la opción de invertir en un bien físico puede comprometer el flujo de caja de la misma. Los costos operacionales, por otro lado, pueden volverse excesivos en el mediano plazo sin ofrecer ningún rendimiento financiero.

Administración de Recursos

Glosario de Conceptos de IT

¿Qué es el TCO?

El Costo total de propiedad (Total Cost of Ownership, TCO) es el costo total de un producto, por ejemplo, un sistema de información) a lo largo de su ciclo de vida completo. El TCO toma en cuenta no sólo los costos directos si no también los indirectos y los recurrentes. Los costos directos pueden ser, por ejemplo, el costo de los equipos: las computadoras, las infraestructuras de red o el costo del software (los costos de las licencias). Los costos indirectos (o costos ocultos) pueden ser los costos de mantenimiento, administración, formación del usuario o del administrador, los costos de desarrollo y de soporte técnico. Por último, los costos recurrentes pueden ser, por ejemplo, los productos consumibles, la electricidad, gastos de alquiler, etc.

Esta herramienta será clave para realizar una correcta evaluación entre diferentes alternativas.

El modelo del TCO fue inventado por Gartner, en el año 1987, para poder analizar y mostrar los costos envueltos con inversiones tecnológicas, concretamente en el área desktop. Desde entonces, los modelos de TCO se han popularizado alcanzando actualmente a la contratación de activos corporativos.

Factores del TCO

Como es evidente, el precio es solo una parte de la ecuación en la negociación.

El objetivo en una negociación debería enfocarse en el llamado costo total o TCO (Total Cost of Ownership). Ese costo total está formado por varios factores, factores que es preciso identificar y medir; pero lo más importante es que es posible gestionar cada uno de los factores del TCO para reducirlos o aumentar su eficiencia.

Costos durante el ciclo de vida útil	Proveedor 1	Proveedor 2	Proveedor 3
Costo de Adquisición	\$100,000.00	\$90,000.00	\$60,000.00
Costos Operacionales	\$2,500.00	\$4,500.00	\$6,500.00
Costos de Entrenamiento	\$15,000.00	\$13,000.00	\$10,000.00
Costo de Control de Calidad	\$2,000.00	\$1,000.00	\$1,500.00
Costos de Mantenimiento	\$4,000.00	\$1,000.00	\$3,000.00
Costos de Almacenamiento	\$12,000.00	\$32,000.00	\$57,000.00
-Ingreso por Venta del Activo al final de su vida útil	\$7,000.00	\$6,000.00	\$4,000.00
TCO (Total Cost Ownership)	\$128,500.00	\$135,500.00	\$134,000.00

Administración de Recursos

Glosario de Conceptos de IT

Fuentes:

<https://www.gpmconsultoria.com/blog/introduccion-al-total-cost-of-ownership-tco.html>

<https://www.myabcm.com/es/blog-post/descubra-las-diferencias-entre-capex-y-opex/>

<https://www.pymesautonomos.com/fiscalidad-y-contabilidad/en-que-consiste-el-capex-y-por-que-es-una-magnitud-tan-importante-para-la-pyme>

Administración de Recursos

Glosario de Conceptos de IT

DevOps

¿Qué significa?

El término DevOps, que es una combinación de los términos ingleses *Development* (desarrollo) y *Operations* (operaciones), designa la unión de personas, procesos y tecnología para ofrecer valor a los clientes de forma constante.

¿Qué significa DevOps para los equipos? DevOps permite que los roles que antes estaban aislados (desarrollo, operaciones de TI, ingeniería de la calidad y seguridad) se coordinen y colaboren para producir productos mejores, más confiables y en menos tiempo. Al adoptar una cultura de DevOps junto con prácticas y herramientas de DevOps, los equipos adquieren la capacidad de responder mejor a las necesidades de los clientes, aumentar la confianza en las aplicaciones que crean y alcanzar los objetivos empresariales en menos tiempo.



Administración de Recursos

Glosario de Conceptos de IT

DevOps establece una “intersección” entre Desarrollo, Operaciones y Calidad, pero no se rige por un marco estándar de prácticas, sino que permite una interpretación mucho más flexible en la medida en que cada organización quiera llevarlo a la práctica, según su estructura y circunstancias.

"El objetivo final de DevOps es minimizar el riesgo de los cambios que se producen en las entregas y dar así un mayor valor tanto a los clientes como al propio negocio"

Ahora que tenemos algo más clara la definición de DevOps, veamos sus connotaciones prácticas, tanto a nivel cultural como tecnológico.

Por qué es tan importante DevOps

Además de los esfuerzos por romper las barreras de comunicación y fomentar la colaboración entre los equipos de desarrollo y operaciones tecnológicas, uno de los principales valores de DevOps es lograr la satisfacción del cliente y la prestación de servicios más rápidamente. DevOps también se ha creado para impulsar la innovación empresarial y ser el motor de continuas mejoras en los procesos.

La práctica de DevOps propicia que cada empresa se ponga como objetivo ofrecer un mejor servicio, en menos tiempo, de mejor calidad y con mayor seguridad a sus clientes finales; por ejemplo, con actualizaciones, funciones o versiones de producto más frecuentes. Puede reflejarse en la rapidez con la que llega al cliente una nueva versión del producto o una nueva función manteniendo los mismos niveles de calidad y seguridad, o en el poco tiempo que se necesita para identificar un problema o un error y, a continuación, solucionarlo y volver a publicar una versión corregida.

Sin duda, todo este trabajo de DevOps se sustenta en una infraestructura subyacente con un rendimiento, una disponibilidad y una fiabilidad fluida y sin interrupciones del software, que primero se desarrolla y se prueba y luego se lanza a la fase de producción.

Administración de Recursos

Glosario de Conceptos de IT

El fin de la “pared de la confusión” y el modelo unidireccional

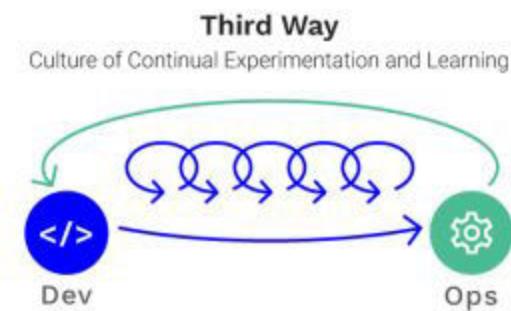
Una de las problemáticas que pretende zanjar el movimiento DevOps es la llamada “pared de confusión”. Es el conflicto que surge cuando el equipo de Desarrollo hace el traspaso de código al equipo de Operaciones, y éste trabaja en el testing y en los scripts para el pase a producción. Es entonces cuando surgen incidencias que no se sabe bien a quién atribuir, si a Desarrollo por entregar código desarrollado con fallos o a Operaciones por realizar una mala configuración. Y mientras ambos equipos se ponen de acuerdo, la fecha de lanzamiento está cada vez más cerca y no hay tiempo para hacer más pruebas-error.

En esta situación cada departamento vela por sus intereses (el primero busca un sistema fluido y dinámico en el que poder añadir nuevas funcionalidades, el segundo la estabilidad por encima de todo) y tienen sus propias prioridades y forma de trabajar, provocando en más de una ocasión fricciones y falta de confianza entre los equipos. Precisamente DevOps entra en escena para aunar los intereses de ambos departamentos y crear un ambiente de confianza y cooperación mediante prácticas que fomentan una **actitud positiva respecto al fallo y una cultura sin culpas (*blameless culture*)**, **colaborativa, en la que la información fluye, el talento se intensifica y el equipo se siente comprometido**. Se contribuye a la **comunicación y trabajo bidireccional** mediante actividades en la que desarrolladores y Operaciones trabajan estrechamente, por ejemplo en las tareas de pruebas y calidad.

Además de las implicaciones en las Personas, DevOps supone un cambio en los Procesos. DevOps va más allá de las metodologías ágiles y se fundamenta en la llamada metodología *Lean*. La aplicación de principios Lean al ciclo de vida DevOps ayuda a lograr los objetivos y minimizar el riesgo de los cambios que se producen en las entregas de una manera más precisa y rigurosa, para dar aún más valor a clientes y al propio negocio.

Administración de Recursos

Glosario de Conceptos de IT



¿Qué hace un DevOps?

A continuación, se listan objetivos básicos y principales que debe tener un DevOps:

- **Lograr ciclos de desarrollo más cortos.** Facilita a los desarrolladores una integración continua en sus códigos, CI (Continuous integration).
- **Hacer despliegue continuo**, es decir pasar la versión de software a entorno de producción mucho más frecuentemente. Esto se llama CD (Continuous Deployment).
- **Mantener una plataforma estable con un 99% o más de disponibilidad.**
- **Eficiencia y automatización.** Eliminar tareas humanas y manuales.
- **Monitorear el rendimiento de una aplicación** (performance) recopilar datos en determinado tiempo (métricas), y en función del análisis de los mismos tomar decisiones en el equipo para mejorar tiempos de respuesta.

¿Cómo implementar DevOps en la empresa?

No es solo contratar un DevOps y listo, que haga maravillas, no funciona así. **Una empresa tiene que empezar por adoptar bases de metodologías ágiles de software en su equipo.** Al tener estos cimientos sólidos, se crea la necesidad de implementar metodologías DevOps para acelerar la entrega del software; esto genera la importancia de capacitar al equipo en herramientas de automatización y tecnologías con el fin de que NO exista la famosa muralla entre operaciones y desarrollo. Ahora todos son un solo equipo que va hacia un mismo objetivo, entregar software de calidad y más rápido.

Administración de Recursos

Glosario de Conceptos de IT

Herramientas DevOps

Existen muchísimas herramientas con la cual un DevOps puede trabajar, **entre las más importantes están:**

- **Manejar nuestra infraestructura como nube**, ya sea On Premise o en proveedores de nube pública como Amazon Web Service (AWS) y Google Cloud Computing (las más populares).
- **Herramientas de integración y despliegue continuo** como Jenkins, GitLabCI, CircleCI o Travis.
- **Lenguajes que apoyaran en la construcción de infraestructura como código**, por ejemplo Terraform y Chef.
- **Herramientas de monitoreo**, contamos con Prometheus, Nagios o Zabbix.
- **Lenguajes que apoyaran a las tareas comunes de un DevOps**: Bash scripting, Python, YAML entre las básicas.

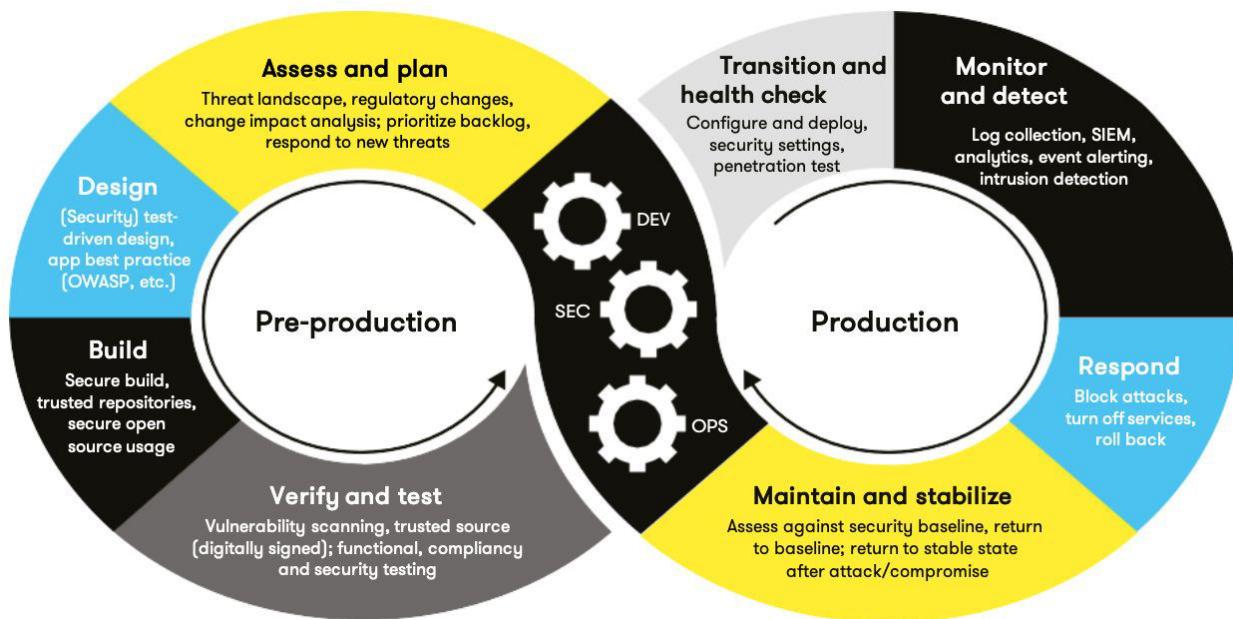
El paso siguiente DevSecOps

Cuando incorporamos a DevOps las consideraciones de seguridad de la información e integramos a los equipos de seguridad dentro de la ecuación, realmente podremos pensar en aplicaciones robustas.

En la imagen siguiente podemos observar cómo se integran las actividades de seguridad dentro del ciclo de DevOps(Tanto en el desarrollo como en la operación de los sistemas), convirtiendo al mismo en DevSecOps.

Administración de Recursos

Glosario de Conceptos de IT



Fuentes:

<https://azure.microsoft.com/es-es/overview/what-is-devops/>

<https://www.xeridia.com/blog/sabes-realmente-que-es-devops>

<https://www.netapp.com/es/info/what-is-devops.aspx>

<https://positivedevops.com/pero-que-es-devops/>

<https://openwebinars.net/blog/que-es-devops/>

Administración de Recursos

Glosario de Conceptos de IT

Concepto de On - Premise

¿Qué significa Software On Premise?

El término *on-premise* o en local se refiere al **tipo de instalación** de una solución de software. Esta instalación se lleva a cabo dentro del servidor y la infraestructura (TIC) de la empresa. Es el modelo tradicional de aplicaciones empresariales.

Con el modelo on-premise, la **empresa es la responsable** de la seguridad, disponibilidad y gestión del software. Por lo que la empresa debe tener un **departamento de sistemas** que dedique parte de sus recursos a la gestión de la infraestructura *in situ*. Sin embargo, el proveedor también suele proporcionar servicios de integración y soporte post-venta. Las soluciones TIC que pueden encontrarse en este tipo de implementación son aquellas que ya llevan un tiempo en el mercado como gestión documental, sistemas ERP o gestión de relaciones con el cliente (CRM), las más nuevas suelen ofrecerse solamente en la nube, que es otro tipo de instalación.

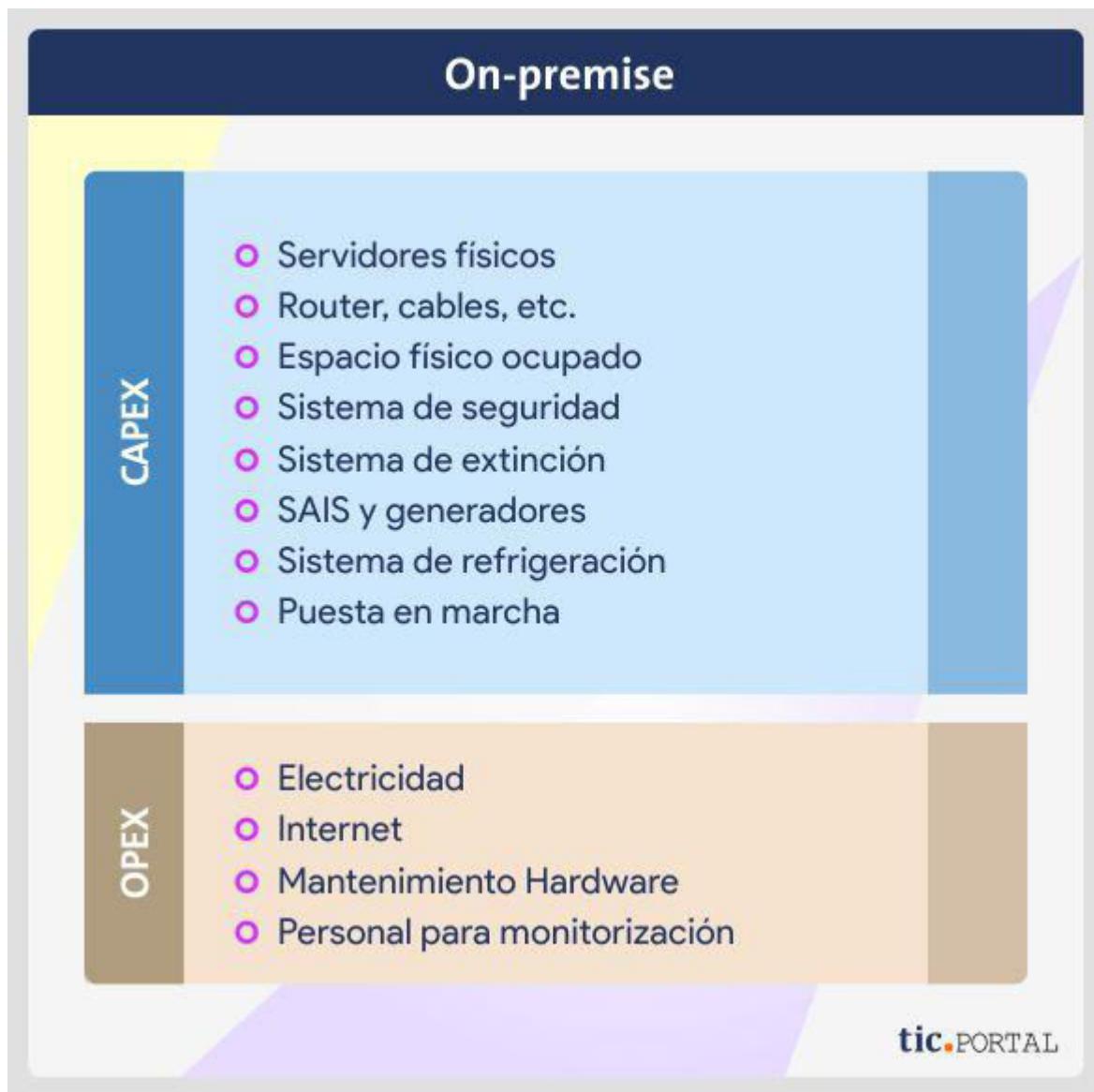
La instalación on-premise ofrece ventajas como un mayor control, pero requiere de una inversión inicial elevada y su crecimiento suele ser escalonado (por ciclos posteriores de nuevas inversiones).

Costos de una instalación on-premise

A la hora de sopesar la inversión en una implementación de software hay que tener en cuenta qué tipo de inversión se requiere. Hay inversiones de muchos tipos, pero para las inversiones de software se deben considerar las de tipo **CAPEX** (CAPital EXpenditures o inversiones de bienes de capital) y las de **OPEX** (OPerating EXPense o costo de funcionamiento). En la categoría de CAPEX entran todos aquellos activos que compra la empresa, mientras que en la de OPEX entran todos aquellos gastos corrientes operativos para el funcionamiento de la empresa. En la nube, solo hay costos de tipo operativos **OPEX**.

Administración de Recursos

Glosario de Conceptos de IT



Administración de Recursos

Glosario de Conceptos de IT

En la instalación en la nube ha surgido una **nueva corriente** en la que, en vez de comprar una solución, se alquila. A esto se le conoce como software como servicio (*Software as a Service, SaaS*). La ventaja que presenta SaaS, es que se paga una **cantidad mensual** determinada al proveedor, pagando por los servicios que le presta.

La prolongación en el tiempo de un sistema de alquiler de una solución de nube con un volumen considerable y estable de consumo de servicios, puede generar que el pago acumulado de SaaS fuera mayor que la inversión de la compra de una solución on-premise. Para empresas de gran tamaño con un departamento de sistemas propio con experiencia, la compra de una solución podría ser **más económica a largo plazo**.

Cloud vs. On Premise: Puntos de Análisis

1. ¿Con qué capital inicial se cuenta?

Si se decide por un modelo on-premise, se requerirá de una inversión inicial, ya que toda la infraestructura correrá por tu cuenta.

Con un modelo cloud-based, es el proveedor de la solución quien se hace cargo de los costos de infraestructura(inversiones) y mantenimiento(gastos).

Es necesario para ello evaluar el perfil de la empresa. ¿Se trata de una empresa ya establecida, que cuenta con la capacidad de hacer la inversión inicial necesaria?, o bien, ¿la empresa es nueva, con un rápido crecimiento y requiere de un acceso inmediato a la solución?.

Se debe tener en cuenta que existen soluciones que no requieren de hardware propietario ni servidores muy costosos para su implementación on-premise, además de que permiten el escalamiento, por lo tanto la inversión inicial en esos casos no sería muy grande de todos modos.

2. ¿Existe personal IT especializado para que se haga cargo del mantenimiento del software y el hardware?

Si ya se cuenta con recursos que destinados a esta tarea, tal vez la opción de cloud-based no haga la diferencia y la modalidad de instalación en propia, seguiría siendo la mejor opción.

Pero si adquirir la solución implica contratar más personal IT para que se encargue de toda la administración de la infraestructura, entonces la solución en la nube puede ser lo indicado.

Administración de Recursos

Glosario de Conceptos de IT

3. ¿El negocio requiere alta disponibilidad y continuidad de servicio?

En ese caso, bajo una modalidad on-premise, la propia empresa debe hacerse cargo de mantener una arquitectura redundante, realizar las copias de seguridad y tomar todas las medidas, inclusive las edilicias con respecto a tu centro de datos, para garantizar la disponibilidad del servicio.

En cambio, con una modalidad cloud-based, el proveedor es quien da garantía sobre la continuidad operativa y la disponibilidad del servicio.

4. ¿Preocupa la seguridad y privacidad de los datos?

En general, cuando de seguridad y privacidad de datos se trata, la opción on-premise pareciera ser la más adecuada.

Sin embargo, es posible obtener garantías de integridad, confiabilidad y privacidad de la información, también con soluciones alojadas en la nube.

Para ello no solo se debe contemplar la cuestión técnica sino también la perspectiva de adherencia de la organización a políticas y normativas internas, regulaciones y leyes. Esto último se lo conoce como *compliance* e incluye todas las actividades que mantienen y proveen una prueba sistemática del cumplimiento de las obligaciones mencionadas. Se trata de identificar y clasificar los riesgos operativos y legales a los que se enfrentan y establecer mecanismos internos de prevención, gestión, control y reacción frente a los mismos.

5. ¿Disponibilidad de una conexión a internet confiable?

Si no se dispone de una conexión a internet garantizada, confiable y suficiente, o resulta costosa, entonces se debe descartar la opción cloud-based.

Para on-premise la conectividad a internet, no sería un requisito, en la medida que las aplicaciones sean de consumo local.

Administración de Recursos

Glosario de Conceptos de IT

Fuentes:

<https://blog.inconcertcc.com/cloud-contact-center-vs-on-premises-5-puntos-de-analisis/>

<https://www.iberiza.es/universo-iberiza/cloud-almacenamiento-hibrida-inversion-it/>

<https://www.ticportal.es/glosario-tic/on-premise>

<https://www.sctrade.es/software-on-premise-cloud/>

<http://www.worldcomplianceassociation.com/que-es-compliance.php>

Administración de Recursos

Glosario de Conceptos de IT

CLOUD

¿Qué es cloud computing?

El término **cloud computing** hace referencia a una concepción tecnológica y a un modelo de negocio que reúne ideas tan diversas como el almacenamiento de información, las comunicaciones entre computadoras, la provisión de servicios o las metodologías de desarrollo de aplicaciones, todo ello bajo el mismo concepto: todo ocurre en la nube.

Tratando de engranar todas estas cuestiones con vistas a la definición de qué es el *cloud computing*, quizá debiéramos detenernos un instante y definir el concepto de nube o, para ser más exactos, de internet. **Internet, definida de una manera deliberadamente simple, es un conjunto de computadoras, distribuidas por el mundo y unidos por una tupida malla de comunicaciones, que ofrece espacios de información a todo el que tenga acceso.** El acceso a la información que nos ofrecen los computadoras que componen Internet es “transparente”, es decir, no es relevante para el usuario el lugar en el que está alojada físicamente la información. De ahí que Internet se represente de una manera universal, como una nube a la que se accede en busca de información y servicios.

No todo lo que ocurre en Internet es *cloud computing*; Internet es un universo que, básicamente, ofrece dos cosas: publicación de información y oferta de servicios. Se puede afirmar que la mera publicación de información no forma parte del modelo de *cloud computing*, así que, con esto, obtenemos una primera frontera que separa lo que está dentro de nuestro ejercicio de definición de aquello que no lo está. Centrémonos por lo tanto en los servicios.

No todo lo que ocurre en Internet es ‘cloud computing’

Internet es también un gran mercado de servicios de diversa naturaleza y formato que podríamos dividir en dos grandes grupos, en base al uso que se le da en la red: los servicios que utilizan la red como canal y los que se encuentran en la red y le ofrecen recursos propios. Respecto a los primeros, pensemos en un banco que ofrece sus servicios transaccionales, oficinas virtuales de atención al cliente, canales de venta o subasta. En realidad la utilidad de internet en estos procesos no es sino un mero canal de comunicación. Estos servicios no se consideran *cloud computing*.

Administración de Recursos

Glosario de Conceptos de IT

En cuanto a los servicios que se encuentran en la red y le ofrecen recursos propios, destacan los servicios de *hosting* que nos permiten guardar información fuera de nuestros sistemas personales o empresariales, es decir, en servidores que están en la nube y a los que podemos acceder a través una red de comunicaciones. Otro ejemplo sería el servicio de correo electrónico, en este caso todo, tanto la aplicación que utilizamos como los datos que intercambiamos con nuestros destinatarios, están almacenados en la nube. Estos servicios sí pueden considerarse *cloud computing*.

Por tanto, y resumiendo todo lo tratado anteriormente, **podríamos definir *cloud computing* como una concepción tecnológica y un modelo de negocio en el que se prestan servicios de almacenamiento, acceso y uso de recursos informáticos esencialmente radicados en la red**, en los que el concepto de canal es un mero instrumento.

Recursos que la red ofrece como servicio

¿Y cuáles son los recursos que la red ofrece como servicio? Podemos considerar tres tipologías básicas de servicios que constituyen el modelo de negocio del *cloud computing* y que, dentro del vocabulario informático han venido en llamarse la generación “As a service” (IaaS, PaaS, SaaS).

Para la definición de estos términos seguiremos el camino que siguen los datos de nuestras computadoras hacia la nube, comenzando por los servicios más vinculados a las máquinas (hardware) hasta llegar a los de naturaleza más lógica (software) pasando por los que hacen posible estos últimos (herramientas de desarrollo).

Infraestructura como servicio

Ofrecer al cliente espacio de almacenamiento o capacidad de procesamiento en sus servidores. Así el usuario tendrá a su disposición “un disco duro de aparente capacidad ilimitada” y un procesador de rendimiento aparente casi infinito, solo restringido a la limitación impuesta por el proveedor y a la capacidad económica de contratación del cliente del servicio. Este servicio se basa en el acceso al uso de hardware radicado en la nube.

Administración de Recursos

Glosario de Conceptos de IT

Plataforma como servicio

El servicio de Plataforma pone a disposición de los usuarios herramientas para la realización de desarrollos informáticos, de manera que aquellos pueden construir sus aplicaciones o piezas de software sin necesidad de adquirir e implantar dichas herramientas en sus computadoras locales. Este servicio tiene dos claras ventajas para el desarrollador de aplicaciones: no tiene que adquirir las costosas licencias para desarrollo de las herramientas de mercado y, por otra parte, el proveedor de servicios se encarga de que dichas herramientas estén en óptima situación de mantenimiento.

Software como servicio

En el punto más alto de las habituales clasificaciones de componentes del mundo informático se encuentran las aplicaciones finales; productos terminados que ofrecen servicios concretos para los que fueron desarrollados. Estas aplicaciones son infinitas en sus distintas naturalezas y usos. El servicio ofrecido como SaaS consiste directamente en la utilización por parte del usuario final de los servicios ofrecidos por dichas aplicaciones, situadas en los servidores del proveedor *cloud*, con un mecanismo de facturación (en caso de no ser un servicio gratuito) más o menos simple de pago por uso.

Todo como servicio

La evolución de las clásicas soluciones cloud as-a-Service es *Everything as a Service* conocido como **XaaS**. No solo IaaS, PaaS y SaaS sino almacenamiento, Disaster Recovery, cyber-security, inteligencia artificial, video, BBDD, integración de servicios cloud (múltiples proveedores) y la lista no para de crecer.

Modelos de nube según la privacidad

El hecho de que la información manejada resida temporal o definitivamente en servidores en la nube lleva a que dichos servicios ofrezcan distintos formatos de privacidad que pueden elegir los usuarios. De ahí que se planteen varios modelos de nubes como espacios de desarrollo de los servicios ofertados. Serían:

Nubes públicas

Los usuarios acceden a los servicios mediante un canal de comunicación compartido sin que exista un exhaustivo control sobre la ubicación de la información que reside en los servidores del proveedor. El hecho de sean públicas no es un sinónimo de sean inseguras.

Administración de Recursos

Glosario de Conceptos de IT

Nubes privadas

Para los clientes que necesiten, por la criticidad de la información que manejen una infraestructura, plataforma y aplicaciones de su uso exclusivo. Accedidas por un canal de comunicación dedicado.

Nubes híbridas

Combinan características de las dos anteriores, de manera que parte del servicio se puede ofrecer de manera privada (onPremise) y otra parte de manera compartida y flexible (por ejemplo entornos de desarrollo, desborde para escalabilidad de capacidades onPremise).

Cinco características esenciales que debe cumplir un servicio para considerarse ‘cloud’

Hasta aquí la definición de *cloud computing* desde un punto de vista informal. Para dar formalidad a lo definido apelaremos a un organismo reputado internacionalmente, el NIST (National Institute of Standards and Technology) del departamento de Comercio de los Estados Unidos de América para citar las cinco características esenciales que debe cumplir un servicio para considerarse *cloud*.

Autoservicio bajo demanda

Un usuario debe poder, de forma unilateral, proveerse de recursos informáticos tales como tiempo de proceso o capacidad de almacenamiento en la medida de sus necesidades sin que sea necesaria la intervención humana del proveedor del servicio.

Acceso amplio a la red

Los servicios proporcionados deben poder ser accesibles a través de mecanismos estándares y desde plataformas heterogéneas (por ejemplo: computadoras, teléfonos móviles o tablets).

Administración de Recursos

Glosario de Conceptos de IT

Asignación común de recursos

Los recursos son puestos a disposición de los consumidores siguiendo un modelo de multipropiedad, asignándose y reasignándose dispositivos físicos o lógicos atendiendo a la demanda de dichos consumidores. En este sentido el usuario no tiene un estricto control del lugar exacto en el que se encuentra su información, aunque sí debe poder especificar un ámbito mínimo de actuación (por ejemplo: un país, un estado o un centro de proceso de datos concreto).

Rápida elasticidad

Las capacidades en los recursos proporcionados a los usuarios deben poder crecer o decrecer bajo demanda de los mismos con celeridad, incluso mediante procesos automáticos.

Servicio medible

Los sistemas *cloud* deben controlar y optimizar sus recursos dotándose de capacidades para medir su rendimiento en un nivel de abstracción suficiente para la naturaleza del servicio proporcionado. Además dicho control debe permitir ser reportado de manera transparente tanto al proveedor del servicio como al consumidor del mismo.

Fuentes:

<https://www.redhat.com/es/topics/cloud>

<https://www.ticbeat.com/cloud/que-es-cloud-computing-definicion-concepto-para-neofitos/>

<https://www.salesforce.com/mx/cloud-computing/>

<https://www.salesforce.com/mx/cloud-computing/>

<https://azure.microsoft.com/es-es/overview/what-is-the-cloud/>

<https://www.zdnet.com/article/xaaS-why-everything-is-now-a-service/>

Administración de Recursos

Glosario de Conceptos de IT

Interoperabilidad

¿Qué es la interoperabilidad?

Interoperabilidad es la capacidad que tienen los sistemas y/o equipos no sólo de intercambiar información si no de interpretarla y procesarla en un formato amigable a su usuario.

Nivel de Operabilidad



¿Cómo se implementa?

La implementación práctica de la interoperabilidad se basa en el hecho fundamental de compartir interfaces standard, es decir, “Hacer que los sistemas se comuniquen entre sí a través de un lenguaje común”, esto no es más que la implementación de un **“Protocolo”** de comunicación.

Hablar de interoperabilidad refiere a cómo facilitar el intercambio de información entre entidades o sistemas, a la creación de modelos estandarizados con los que puedan representarse e intercambiarse datos acerca de las estructuras más comúnmente

Administración de Recursos

Glosario de Conceptos de IT

encontradas y repetidas, las cuales posiblemente faciliten que paulatinamente se vaya creando, una nube de información con un mínimo de duplicación de información.

¿Y cómo elegir entre tantos estándares disponibles? ¿Qué formatos, protocolos y lenguajes son los más adecuados? ¿Cómo podemos determinar cuál es un estándar abierto?

Los estándares abiertos deberán tener, como mínimo, las características siguientes:

- Disponibilidad.
- Que los derechos de autor estén disponibles, libres de regalías y condiciones.
- Madurez.
- Internacionalmente aceptados.
- De fácil distribución.
- Con amplio soporte en el mercado.

El sólo hecho de reducir la cantidad de datos redundantes ya por sí sólo lo valdría. Agreguemos a esto que reducirá fuertemente la dependencia de proveedores únicos y de la permanencia en el mercado de productos específicos, fomentando el crecimiento de una verdadera industria de desarrollo de software.

¿Por qué la interoperabilidad es tan importante?

Principalmente porque facilita en gran medida la gestión conjunta y ordenada de toda la información.

Además entre sus ventajas se puede destacar:

- La adaptabilidad: los distintos sistemas que absorben la información, se conectan y se encargan de distribuir dicha información de forma automática y flexible.
- Cohesión de datos garantizada: con la interoperabilidad, la información es gestionada de forma eficaz y controlada por todas las partes.

Administración de Recursos

Glosario de Conceptos de IT

- Aumento de la productividad: de la anterior ventaja se puede desprender la idea de que gracias a esta herramienta, se puede asociar toda la información disponible a la cadena de valor. De tal forma, se puede trabajar y operar de forma concordante, a lo largo de todo el proceso productivo, asegurando que la información está disponible y es accesible para todas las partes, sistemas y personas.

Fuentes:

<https://www.tcass.com/menu-notas-de-interes/62-salud/204-que-es-la-interoperabilidad>

<https://dej.rae.es/lema/interoperabilidad>

<http://interoperability-definition.info/es/>

<https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/432/1/M-CD4336.pdf>

<https://sg.com.mx/revista/33/programar-es-un-estilo-vida-interoperabilidad>

<https://nexusintegra.io/es/blog/que-es-la-interoperabilidad-y-como-puede-lograrla-mi-empres/>

<https://www.espaciobim.com/interoperabilidad>

<https://www.chakray.com/es/interoperabilidad-definicion-e-importancia/>

Administración de Recursos

Glosario de Conceptos de IT

SLA (Service Level Agreement)

Definición

Según ITIL 4 un acuerdo de nivel de servicio (SLA) **es un acuerdo documentado entre un proveedor de servicios y el cliente que identifica tanto los requerimientos como el nivel esperado de servicio** (una o más métricas que definen la calidad de servicio esperada o alcanzada).

Los SLA se basan en “lo entregado”, dado que su propósito es específicamente definir lo que el cliente recibirá. Los SLA no definen cómo se proporciona o entrega el servicio en sí.

Un ejemplo...

El SLA que un proveedor de servicios de Internet (ISP) proporcionará a sus clientes es un ejemplo básico de un SLA de un proveedor de servicios. Las métricas que definen los niveles de servicio para un ISP deben apuntar a garantizar:

- **Una descripción del servicio que se brinda:** mantenimiento de áreas tales como conectividad de red, servidores de nombres de dominio, servidores de protocolo de configuración dinámica de host
- **Confiabilidad:** cuando el servicio está disponible (tiempo de actividad porcentual) y se puede esperar que los límites de interrupciones permanezcan dentro de
- **Capacidad de respuesta:** la puntualidad de los servicios que se realizarán en respuesta a las solicitudes y las fechas de servicio programadas
- **Procedimiento para reportar problemas:** con quién se puede contactar, cómo se reportarán los problemas, el procedimiento para la escalada y qué otros pasos se toman para resolver el problema de manera eficiente
- **Nivel de servicio de supervisión e informe:** quién supervisará el rendimiento, qué datos se recopilarán y con qué frecuencia y cuánto acceso tiene el cliente a las estadísticas de rendimiento
- **Consecuencias por no cumplir con las obligaciones de servicio:** puede incluir crédito o reembolso a los clientes, o permitir que el cliente termine la relación.
- **Cláusulas o restricciones de escape:** circunstancias en las que no se aplica el nivel de servicio prometido. Un ejemplo podría ser una exención de cumplir con los requisitos de tiempo de actividad en caso de que las inundaciones, incendios u otras situaciones peligrosas dañen el equipo del ISP.

Administración de Recursos

Glosario de Conceptos de IT

Si bien las métricas específicas para cada SLA varían según el proveedor del servicio, las áreas cubiertas son más o menos uniformes: **volumen y calidad del trabajo, velocidad, capacidad de respuesta y eficiencia**. Al cubrir estas áreas, el documento pretende establecer un entendimiento mutuo de los servicios, las áreas priorizadas, las responsabilidades y las garantías proporcionadas por el proveedor del servicio.

El nivel de definiciones de servicio debe ser específico y medible en cada área. Esto permite que la calidad del servicio sea un punto de referencia y, si así lo estipula el acuerdo, se recompense o se penalice en consecuencia. Un SLA usará comúnmente definiciones técnicas que cuantifiquen el nivel de servicio, como el tiempo medio entre fallas (MTBF) o el tiempo medio de recuperación, respuesta o resolución (MTTR), que especifica un valor "objetivo" (promedio) o "mínimo" para el rendimiento de nivel de servicio.

El uso de SLA también es común en la subcontratación, la computación en la nube y otras áreas donde la responsabilidad de una organización se transfiere a otro proveedor.

SLAs hoy

Una evaluación honesta de los SLA actuales por parte de los proveedores de servicios sugeriría que un gran porcentaje simplemente no tiene sentido. Lo más importante, cuando se les pregunta, es lo difícil que resulta articular qué es la disponibilidad y cómo se mide.

Veamos un ejemplo de una aerolínea de la vida real: la función común de imprimir etiquetas de equipaje. El sistema imprimía información incorrecta en las etiquetas del equipaje, lo que provocaba problemas importantes, como retrasos en los vuelos y una mala experiencia del cliente. Sin embargo, cuando el proveedor informó el rendimiento del servicio durante el mes, cumplió con todos los SLA. Vamos a aclarar esto: el sistema estaba disponible, estaba imprimiendo las etiquetas del equipaje, por lo que cumplió con el objetivo de disponibilidad acordado.

Administración de Recursos

Glosario de Conceptos de IT

SLAs en el futuro

Hay una gran oportunidad para evolucionar y diseñar SLA innovadores utilizando un enfoque centrado en la experiencia del cliente. Se trata de elaborar niveles de servicio que se basen en resultados de negocio.

En nuestro ejemplo, podría ser:

- Una lista precisa y optimizada está disponible para gestionar las operaciones de la tripulación
- Mínimo trabajo manual empleado en el ingreso de datos regulatorios en aeropuertos
- Cambios de aeronaves realizados para cumplir con los horarios de vuelo y el desempeño a tiempo

Dicho esto, sería raro que un solo proveedor acepte este tipo de acuerdos, tanto desde el punto de vista financiero como de rendición de cuentas; típicamente, ningún proveedor individual tiene la propiedad completa del resultado.

Una solución para enfrentar este desafío es diseñar "acuerdos basados en resultados". El equipaje perdido constituye la categoría más alta de quejas para las aerolíneas. La gestión del equipaje involucra a muchos proveedores de servicios y componentes: proporcionar los datos de destino, imprimir las etiquetas de equipaje, escanear y rastrear equipaje. Colectivamente, todos los proveedores contribuyen al resultado comercial de una aerolínea: el equipaje de los clientes llega a tiempo, al destino correcto con la prioridad correcta.

El diseño de acuerdos de nivel de servicio basados en resultados, acordados con cada proveedor, determinaría cómo cada uno de ellos contribuye a la gestión de equipaje para la aerolínea y permitiría a la organización informar una visión integral a sus partes interesadas.

ITIL SVS (Service Value System) representa cómo los diversos componentes y actividades de la organización trabajan juntos para facilitar la creación de valor a través de servicios habilitados por TI.

Para garantizar un enfoque holístico de la gestión de servicios, se describen cuatro dimensiones de la gestión de servicios, a partir de las cuales se debe considerar cada componente de la SVS. Las cuatro dimensiones son:

- organizaciones y personas
- información y tecnología
- socios y proveedores
- cadenas de valor y procesos

Administración de Recursos

Glosario de Conceptos de IT

ITIL (Information Technology Infrastructure Library)

ITIL, acrónimo de “Information Technology Infrastructure Library”, es un conjunto de prácticas detalladas para la gestión de servicios de TI que se centra en alinear los servicios de TI con las necesidades de las empresas.

ITIL ha liderado la industria de ITSM con programas de orientación, capacitación y certificación durante más de 30 años. ITIL 4 actualiza ITIL mediante la remodelación de gran parte de las prácticas establecidas de gestión de servicios de IT (ITSM) en el contexto más amplio de la experiencia del cliente, el flujo de valor y la transformación digital, así como la adopción de nuevas formas de trabajo, como Lean, Agile y DevOps.

Los componentes centrales de ITIL 4 son el sistema de valor de servicio (Service Value System, SVS) y las cuatro dimensiones del modelo.

ITIL describe procesos, procedimientos, tareas y listas de verificación que no son específicos de la organización ni de la tecnología, pero que pueden ser aplicados por una organización para establecer la integración con la estrategia de la organización, entregar valor y mantener un nivel mínimo de competencia. Le permite a la organización establecer una línea de base desde la cual puede planificar, implementar y medir. Se utiliza para demostrar el cumplimiento y para medir la mejora. No hay una evaluación formal e independiente de cumplimiento de terceros disponible para el cumplimiento de ITIL en una organización. La certificación en ITIL solo está disponible para individuos.

La certificación, es reconocida profesionalmente, proporciona una guía completa para establecer un sistema de gestión de servicios, y proporciona un glosario común de términos para empresas que utilizan servicios TI.

¿Qué es un OLA (ITIL 4)?

Un acuerdo entre un proveedor de servicios de IT y otra parte de la misma organización. Apoya la provisión de servicios a los clientes por parte del proveedor y define los bienes o servicios a proveer junto con las responsabilidades de ambas partes.

Por ejemplo, podría establecerse un OLA:

- entre el proveedor de servicios y el departamento de compras para obtener hardware en plazos acordados
- entre la mesa de ayuda y un grupo de soporte para proveer resolución de incidentes en plazos acordados

Administración de Recursos

Glosario de Conceptos de IT

Referencias:

<https://www.axelos.com/best-practice-solutions/itil>

<https://www.axelos.com/news/blogs/february-2019/slas-of-future-measuring-outcomes-not-it-availability>

<https://www.axelos.com/news/blogs/august-2016/of-airlines-technology-and-human-error>

Administración de Recursos

Glosario de Conceptos de IT

RPO Y RTO

¿Qué significa RPO?

El término RPO (Recovery Point Objetive o Punto de recuperación objetivo) se refiere al tiempo que transcurre entre el momento del desastre y el último punto de restauración de nuestros datos, es decir, la cantidad de datos que nuestra empresa va a perder en caso de que se produzca un fallo del sistema.

Pongamos un ejemplo sencillo, si nuestra política de backup establece que hacemos una copia diaria a las 22:00 y tenemos un incidente con pérdida de datos a las 11:00, habremos perdido los datos que hayamos introducido en el sistema entre las 22:00 y las 11:00 del día siguiente, es decir, hemos perdido 13h de trabajo. Si únicamente hacemos una copia diaria, nuestro RPO será de 24h.

Evidentemente, lo ideal es disponer de un RPO cuanto más bajo mejor, de manera que si perdemos datos, se cae el sistema, o sucede un desastre, minimicemos el impacto que puede tener en nuestra compañía. Para ello podemos lanzar más puntos de restauración a lo largo del día, si nuestro entorno está virtualizado realizar **réplicas de las máquinas virtuales** en otro servidor o incluso disponer de una **replicación** entre dos centros de datos.

¿Qué significa RTO?

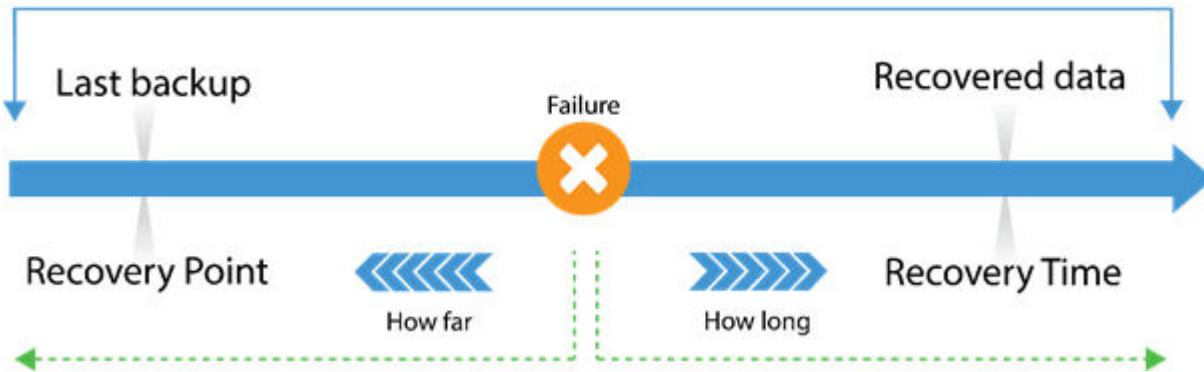
El término RTO (Recovery Time Objetive) tiene que ver con el tiempo de recuperación. Es decir, el tiempo que estamos dispuestos a asumir entre la caída del sistema o la pérdida de datos, hasta que dicho sistema ha sido levantado de nuevo o bien, se han recuperado los datos eliminados.

Al igual que el RPO, como es lógico, se busca que este tiempo sea lo más bajo posible de manera de dicha incidencia tenga un impacto lo más leve posible en el funcionamiento de nuestra empresa.

Para reducir el RTO o tiempo de recuperación, hay que contar con un sistema de resguardo de datos (backup) ágil y moderno que nos permita volver al estado normal en el menor tiempo posible. En este sentido, son varios los factores que pueden influir en el RTO, como puede ser el tipo de destino de copia (cinta y/o disco), la red y los sistemas redundantes.

Administración de Recursos

Glosario de Conceptos de IT



Si bien RPO y RTO pueden parecer similares, sirven para diferentes propósitos y, en un mundo ideal, sus valores serían tan cercanos a cero como sea posible. Sin embargo, de vuelta en nuestro mundo, el “cero” para RPO y RTO sería extremadamente costoso.

El RPO es la cantidad de datos que puede perder antes de que afecte las operaciones. Por ejemplo, para un sistema bancario, una hora de pérdida de datos puede ser catastrófica, ya que operan transacciones en vivo. A nivel personal, también puede pensar en RPO como el momento en que guardó por última vez un documento en el que está trabajando. En caso de que su sistema falle y pierda su progreso, ¿cuánto de su trabajo está dispuesto a perder antes de que le afecte?

Por otro lado, RTO es el marco de tiempo dentro del cual la aplicación y los sistemas deben ser restaurados después de una interrupción. Es una buena práctica medir el RTO a partir del momento en que ocurre la interrupción, en lugar del momento en que el equipo de TI comienza a solucionar el problema. Este es un enfoque más realista, ya que representa el punto exacto en el que los usuarios comienzan a verse afectados.

Cómo definir los valores de RTO y RPO para sus aplicaciones

No existe una solución única para un plan de continuidad del negocio y sus métricas. Las organizaciones son diferentes, tienen distintas necesidades y, por lo tanto, tienen diferentes requisitos para sus objetivos de recuperación. Sin embargo, una práctica común es dividir aplicaciones y servicios en diferentes niveles, así como establecer los valores de tiempo de recuperación y objetivo (RTPO, por sus siglas en inglés) de acuerdo con los **acuerdos de niveles de servicio (SLA, por sus siglas en inglés)**, con los que se compromete la organización.

Administración de Recursos

Glosario de Conceptos de IT

La clasificación de protección de datos es importante para determinar cómo almacenar, acceder, proteger, recuperar y actualizar datos e información de manera más eficiente en función de sus criterios específicos. Es de suma importancia analizar sus aplicaciones y determinar cuáles de ellas están impulsando su negocio, generando ingresos y siendo imprescindibles para mantenerse operativo. Este proceso, que es esencial para un buen plan de continuidad comercial, se denomina **análisis de impacto del negocio (BIA, por sus siglas en inglés)**, y establece protocolos y acciones para enfrentar un desastre.

Por ejemplo, puede usar un modelo de tres niveles para diseñar su plan de continuidad del negocio:

- **Nivel 1:** Aplicaciones esenciales que requieren un RTO de menos de 15 minutos.
- **Nivel 2:** Aplicaciones esenciales para un negocio que requieren RTO de 2 horas y RPO de 4 horas.
- **Nivel 3:** Aplicaciones no esenciales que requieren RTO de 4 horas y RPO de 24 horas.

Es importante tener en cuenta que las aplicaciones esenciales, para un negocio y las no esenciales, varían de una industria a otra, y cada organización define estos niveles en función de sus operaciones y requisitos.

Fuentes:

<https://www.2ksystems.com/blog/14-blog-sistemas-servidores/199-diferencia-entre-rpo-y-rto>

<https://www.itsafer.com/que-es-el-rto-y-el-rpo-en-un-plan-de-recuperacion-de-desastres-drp/>

<https://www.veeam.com/blog/es-lat/rto-rpo-definitions-values-common-practice.html>

<https://datalogic.com.uy/rpo-y-rto-en-la-continuidad-del-negocio/>

<https://blog.globaltrustassociation.org/es/diferencias-entre-rpo-y-rto/>

Administración de Recursos

Glosario de Conceptos de IT

Escalabilidad

¿Qué es la escalabilidad?

La escalabilidad es la capacidad que posee un sistema o proceso de poder expandirse para satisfacer las necesidades de mayor capacidad en alguno de los aspectos de su arquitectura. Ésto puede suceder por necesidades transitorias o permanentes que van más allá de las capacidades habituales del sistema hasta un momento dado.

En general, hablamos de escalabilidad, haciendo referencia a las posibilidades de dicho proceso o sistema tiene de aumentar sus capacidades (por ejemplo, capacidad de atención, requerimientos por segundo, entre muchos otros), sin perder o degradar la calidad del servicio.

¿De qué depende que un sistema sea escalable?

Que un sistema sea escalable o no, depende de innumerables factores, tales como la arquitectura de la solución, la tecnología usada, la complejidad de la solución, el costo, entre muchos otros. Uno de los factores de mayor peso, suele ser el diseño propio de la solución. Un mal diseño inicial, difícilmente será escalable, salvo que se tomen acciones específicas para que ello suceda.

Lo que sí podemos afirmar es que en esta característica de los sistemas, se aplica el siguiente concepto:

“Una cadena es tan fuerte como su eslabón más débil”.

(Frase del filósofo escocés Thomas Reid, escrita en el siglo XVII)

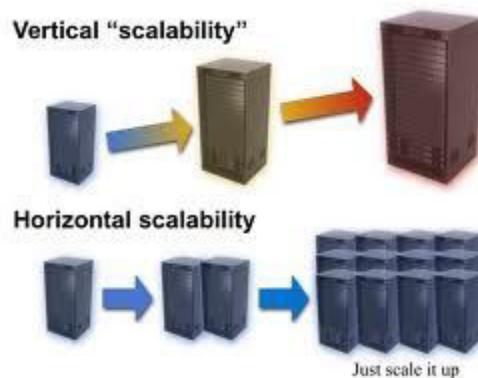
Ya que, en general un componente del sistema que no permita escalabilidad (o se vea reducido en esta capacidad), afecta o limita la capacidad de escalar del sistema entero.

Administración de Recursos

Glosario de Conceptos de IT

¿Cómo puede escalar nuestro sistema?

De manera general siempre hablaremos de escalado vertical, horizontal o una combinación de ambas.



El **escalado vertical**, consiste básicamente en aumentar la capacidad de uno o varios elementos concreto de nuestra arquitectura, por ejemplo ampliar la memoria de nuestro servidor, o sustituir las CPUs por otras de mayor velocidad.

El **escalado horizontal**, se basa en aumentar el número de elementos que desempeñan una determinada tarea, por ejemplo si tenemos un servidor web saturado, añadimos otro para que se haga cargo de parte de las tareas (cargas).

Referencias:

<https://docs.microsoft.com/es-es/biztalk/core/what-is-scalability>

<http://blog.aodbc.es/?p=1231>

<https://www.auditool.org/blog/control-interno/3338-tan-fuerte-como-el-eslabon-mas-debil>

Administración de Recursos

Glosario de Conceptos de IT

Concurrencia

¿Qué es la concurrencia?

Concurrencia es la tendencia de las cosas a producirse al mismo tiempo en un sistema. La concurrencia es un fenómeno natural, por supuesto. En el mundo real, en un momento dado, suceden muchas cosas de forma simultánea. Cuando se diseña software para supervisar y controlar los sistemas del mundo real, hay que enfrentarse a esta concurrencia natural.

En lo que se refiere a temas de concurrencia en sistemas de software, hay dos aspectos importantes en general: poder detectar y responder a los sucesos externos que se producen en un orden aleatorio y garantizar que estos sucesos se responden en un intervalo requerido mínimo.

- Dos o más procesos decimos que son concurrentes, paralelos, o que se ejecutan concurrentemente, cuando son procesados al mismo tiempo, es decir, que para ejecutar uno de ellos, no hace falta que se haya ejecutado otro.
- En sistemas multiprocesador, esta ejecución simultánea podría conseguirse completamente, puesto que podremos asignarle, por ejemplo, un proceso A al procesador Nro 1 y un proceso B al procesador Nro 2 y cada procesador realizará la ejecución de su proceso.
- Cuando tenemos un solo procesador se producirá un intercalado de las instrucciones de ambos procesos, de tal forma que tendremos la sensación de que hay un paralelismo en el sistema (concurrencia, ejecución simultánea de más de un proceso).

Problema de la Alta Concurrencia

En general, durante el proceso de puesta en marcha de un sistema, nos topamos con el problema de resolver de qué forma puede el mismo sostener un nivel de servicio ante una alta demanda por parte de los usuarios.

Administración de Recursos

Glosario de Conceptos de IT

Es aquí donde debemos realizar un análisis del tráfico y/o peticiones que deberá soportar el sistema para poder plantear una solución acorde a la demanda. ¿Cómo podemos plantear soluciones? Utilizando una serie de herramientas, tecnologías y patrones de diseño que nos permitan desarrollar soluciones robustas, escalables y paralelizables.

¿Cómo garantizar la concurrencia?

La forma de validar la capacidad de concurrencia viene de la mano de las pruebas de concurrencia o stress. Para realizar las mismas se necesita de los casos funcionales del sistema, la secuencia de uso, la alternancia y cuantificación de cada una de las acciones sobre el sistema.

Por otra parte, se requiere de alguna herramienta que sea capaz de automatizar el proceso de entradas al sistema. La herramienta se configura para simular los casos en la cantidad y secuencias definidas. Luego de correr los casos de prueba se evalúan los resultados sobre los cuales deberá existir previamente una métrica de rangos de tiempo y respuesta esperados.

De la comparación de los resultados obtenidos con el esperado se podrá determinar si el nivel de concurrencia probado está validado o no. En caso de no ser aceptable/esperado el resultado de las pruebas, se deberán analizar las causas del desvío y proponer y aplicar las mejoras correspondientes para que los resultados de las nuevas pruebas generen respuestas dentro del rango admisible.

Fuentes:

<https://www.monografias.com/docs114/conceptos-concurrencia/conceptos-concurrencia.shtml>

<https://webprogramacion.com/43/sistemas-operativos/concurrencia-de-procesos.aspx>

https://cgrw01.cgr.go.cr/rup/RUP.es/SmallProjects/core.base_rup/guidances/concepts/concurrency_EE2E011A.html

Administración de Recursos

Glosario de Conceptos de IT

UPS (uninterruptible power supply)

Definición

Es un dispositivo que permite mantener el suministro de energía eléctrica por un tiempo limitado a todos los dispositivos que tenga conectados con independencia de la continuidad de la tensión de la red eléctrica.

Sus componentes básicos consisten en un controlador de transferencia y el banco de baterías. El banco de baterías almacena energía. A mayor cantidad de baterías, mayor “autonomía”, entendido como tiempo de continuidad energética ante la ausencia de corriente de entrada al sistema.

Adicionalmente, las UPS funcionan como estabilizador de corriente, filtrando subidas y bajadas de tensión y eliminando armónicos en AC. Ésto genera una mejoría de la calidad de la energía eléctrica que llega a las cargas.

Objetivo

El objetivo de la instalación de un sistema de UPS es mantener la continuidad energética de una instalación ante la ausencia de energía desde la fuente de entrada primaria y hasta que se realice la transferencia a la fuente de energía secundaria (generalmente un generador de corriente).

Tipos

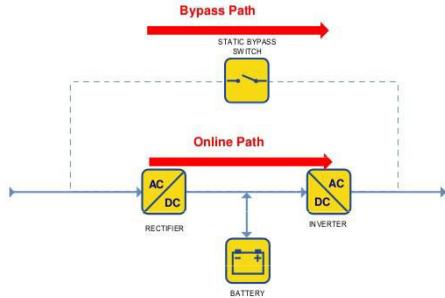
Clasificación por “modo de operación”:

Los sistemas on-line/doble conversión, se caracterizan por estar permanentemente transformando la energía y alimentando las cargas desde la energía almacenada en las baterías. A diferencia de los sistemas off-line, que están alimentando las cargas desde la entrada y manteniendo la carga de energía en el banco de baterías y al momento de detectar una falla en la entrada ya sea la ausencia de suministro o la salida de rangos de tensión transfiere automáticamente la alimentación de sus cargas al propio sistema.

Administración de Recursos

Glosario de Conceptos de IT

Simple diagram of power flow of a double-conversion, online UPS...



© 2014 Schneider Electric. All Rights Reserved - Eco mode, Benefits and Risks of Energy-Saving Modes of UPS Operation - June 2014

Off Line UPS

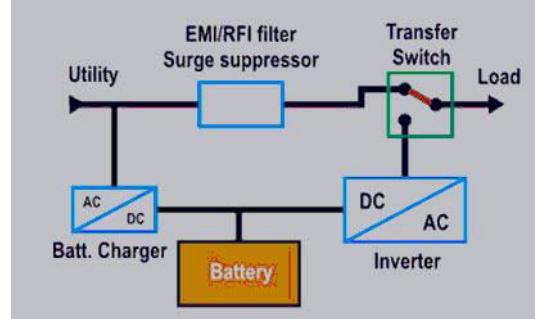


Imagen izquierda: diagrama de operación de UPS “on-line”. Siempre alimentando sus cargas transformando la energía almacenada en el banco de baterías. En operación normal, las baterías se mantienen permanentemente “a flote” cargándose con la energía proveniente de la red de suministro. Al entregar energía “transformada”, la calidad de la salida es garantizada.

Imagen derecha: diagrama de operación de UPS “off-line”. Siempre alimentando sus cargas desde el suministro de la red y simultáneamente se mantienen “a flote” el nivel de carga del banco de baterías. En caso de ausencia de suministro desde la red, se realiza automáticamente la transferencia de cargas para ser alimentada por la propia energía almacenada en el banco de baterías.

Administración de Recursos

Glosario de Conceptos de IT

Tolerancia a Fallos (Fault Tolerance)

¿Qué es la tolerancia a fallos?

La tolerancia a fallos es la propiedad que le permite a un sistema seguir funcionando correctamente en caso de fallo de uno o varios de sus componentes.

La tolerancia a fallos, tal y como la conocemos hoy en día, se basa fundamentalmente en un concepto: redundancia. La mejor forma de asegurar la disponibilidad de nuestros equipos, y de los datos y los servicios que ellos suministran de manera altamente *confiable* y sin interrupción las 24 horas del día durante siete días a la semana, es la duplicación de todos sus elementos críticos y la disposición de los elementos software y hardware necesarios para que los elementos redundantes actúen cooperativamente, bien sea de forma activa-activa o activa-pasiva, pero siempre de forma transparente para el usuario.

La confianza que se puede poner en un sistema de computación, está dada por los siguientes atributos: **Fiabilidad, Disponibilidad, Resguardo y Seguridad**.

- La fiabilidad trata de la continuidad de servicio.
- La disponibilidad trata del porcentaje de tiempo en que el servicio está disponible para su uso.
- El resguardo trata de evitar la pérdida como consecuencias del incidente.
- La seguridad trata de la protección contra intrusos, aunque de manera genérica puede incluir a todos los anteriores.

Las métricas comúnmente utilizadas para medir la disponibilidad y la fiabilidad de un sistema son el tiempo medio entre fallos o MTTF (*mean time to failure*) que mide el tiempo medio transcurrido hasta que un dispositivo falla, y el tiempo medio de recuperación o MTTR (*mean time to recover*) que mide el tiempo medio tomado en restablecerse la situación normal una vez que se ha producido el fallo.

Lógicamente, el principal objetivo es aumentar el MTTF y reducir el MTTR de forma que minimicemos ese tiempo.

Administración de Recursos

Glosario de Conceptos de IT

Prevención de Fallos de los Sistemas

Una disfunción ocurre cuando el servicio de un sistema no es consistente con su especificación. Esto sucede por ejemplo cuando falla alguno de sus componentes.

La prevención de Fallos nos permite analizar las características del sistema para impedir que se presenten fallos en los componentes. Los métodos pueden ser: de diseño, de prueba y/o de validación.

Por ejemplo para validar la capacidad de respuesta de un sistema simulando su carga operativa esperada.

Fuentes:

<http://armlaz.tripod.com/armandolazarte/stf.htm>

[https://es.ryte.com/wiki/Tolerancia frente a Fallos](https://es.ryte.com/wiki/Tolerancia_frente_a_Fallos)

<https://es.scribd.com/doc/128867681/Tesis-Tolerancia-a-fallos>

Administración de Recursos

Glosario de Conceptos de IT

Plan de Contingencia

Dentro de la [seguridad informática](#) se denomina **plan de contingencia** (también de recuperación de desastres o de continuación de negocios), a la definición de acciones a realizar, recursos a utilizar y personal a emplear en caso de producirse un acontecimiento intencionado o accidental que inutilice o degrade los recursos informáticos o de transmisión de datos de una organización. Es decir, es la determinación precisa del quién, qué, cómo, cuándo y dónde realizar acciones en caso de producirse una anomalía en el sistema de información.

El plan de contingencia debe considerar todos los componentes del sistema: Datos críticos, equipo lógico de base, aplicaciones, equipos físicos y de comunicaciones, documentación y personal. Además, debe contemplar también todos los recursos auxiliares, sin los cuales el funcionamiento de los sistemas podría verse seriamente comprometido: suministro de potencia; sistemas de climatización; instalaciones; etc. Finalmente, debe prever también la carencia de personal cualificado (por ejemplo, por una huelga que impida el acceso del mismo) para el correcto funcionamiento del sistema. Se debe destacar, que previo al comienzo del trabajo, se debe obtener el pleno compromiso de los máximos responsables de la organización. Sin su apoyo decidido y constante, el fracaso del plan está garantizado.

El plan de contingencias debe ser comprobado de forma periódica para detectar y eliminar problemas. La manera más efectiva de comprobar si funciona correctamente, es programar simulaciones de desastres. Los resultados obtenidos deben ser cuidadosamente revisados, y son la clave para identificar posibles defectos en el plan de contingencia. Además el plan de contingencia debe contemplar los planes de emergencia, resguardo, recuperación, comprobación mediante simulaciones y mantenimiento del mismo.

Un plan de contingencia adecuado debe ayudar a las empresas a recobrar rápidamente el control y capacidades para procesar la información y restablecer la marcha normal del negocio, minimizando el impacto o pérdida a causa del incidente.

El plan de contingencias comprende tres subplanes. Cada plan determina las contramedidas necesarias en cada momento del tiempo respecto a la materialización de cualquier amenaza:

- El **plan de prevención**. Contempla las contramedidas preventivas **antes** de que se materialice una amenaza. Su finalidad es contar con las medidas necesarias para que en caso de materialización del riesgo se pueda restituir el servicio. Ej: contar con repuestos, equipos sustitutos, datos de resguardo.
- El **plan de emergencia**.¹ Contempla las contramedidas necesarias **durante** la materialización de una amenaza, o inmediatamente después. Su finalidad es **paliar** los efectos adversos de la amenaza.
- El **plan de recuperación**. Contempla las medidas necesarias **después** de materializada y controlada la amenaza. Su finalidad es **restaurar** el estado de las cosas tal y como se encontraban antes de la materialización de la amenaza.

Administración de Recursos

Glosario de Conceptos de IT

Por otra parte, el plan de contingencias no debe limitarse a estas medidas organizativas. También debe expresar claramente:

- Qué recursos materiales son necesarios.
- Qué personas están implicadas en el cumplimiento del plan.
- Cuáles son las responsabilidades concretas de esas personas y su rol dentro del plan.
- Qué protocolos de actuación deben seguir y cómo son.

Fuentes:

https://www.ecured.cu/Plan_de_contingencia_en_seguridad_Inform%C3%A1tica

https://es.wikipedia.org/wiki/Plan_de_contingencias

<http://www.forodeseguridad.com/artic/discipl/4132.htm>

Administración de Recursos

Glosario de Conceptos de IT

Alta Disponibilidad (HA - High Availability)

¿Qué es la disponibilidad?

La disponibilidad es una de las características de las arquitecturas empresariales que mide el grado con el que los recursos del sistema están disponibles para su utilización por el usuario final a lo largo de un período dado. Ésta no sólo se relaciona con la prevención de caídas del sistema (también llamadas tiempos *fuerza de línea, downtime* u *offline*), sino incluso con la percepción de “caída” desde el punto de vista del usuario: cualquier circunstancia que nos impida trabajar productivamente con el sistema – desde tiempos de respuesta prolongados, escasa asistencia técnica o falta de estaciones de trabajo disponibles – es considerada como un factor de baja disponibilidad.

¿Cómo medimos la disponibilidad?

De primera instancia, todo sistema debe tener establecido un Acuerdo de Nivel de Servicio (Service Level Agreement – SLA) que defina cuánto tiempo y en qué horarios debe estar en línea. En el caso de aplicaciones de baja criticidad, dicho SLA puede ser de 8×5 horas a la semana excluyendo días festivos; para sistemas con mayor criticidad como una red de cajeros automáticos se tienen niveles de servicio que alcanzan las 24 horas al día, los 365 días del año. Así entonces, suponiendo un sistema con un SLA de 24×365 podríamos calcular su disponibilidad de la siguiente manera:

$$\text{Disponibilidad} = ((A - B)/A) \times 100 \text{ por ciento}$$

Donde:

A = Horas comprometidas de disponibilidad: $24 \times 365 = 8,760$ Horas/año.

B = Número de horas fuera de línea (Horas de “caída del sistema” durante el tiempo de disponibilidad comprometido). Por ejemplo: 15 horas por falla en un disco; 9 horas por mantenimiento preventivo no planeado.

así entonces:

$$\text{Disponibilidad} = ((8,760 - 24)/8,760) \times 100 \text{ por ciento} = 99.726\%$$

Administración de Recursos

Glosario de Conceptos de IT

Cuando se realicen negociaciones para definir objetivos de disponibilidad con los usuarios, es necesario hacerlos conscientes de las implicaciones técnicas y económicas, como se muestra en la siguiente tabla:

Disponibilidad (%)	Tiempo offline/año	Tiempo offline/mes	Tiempo offline/día
90%	36.5 días	73 hrs	2.4 hrs
95%	18.3 días	36.5 hrs	1.2 hrs
98%	7.3 días	14.6 hrs	28.8 min
99%	3.7 días	7.3 hrs	14.4 min
99.5%	1.8 días	3.66 hrs	7.22 min
99.9%	8.8 hrs	43.8 min	1.46 min
99.95%	4.4 hrs	21.9 min	43.8 s
99.99%	52.6 min	4.4 min	8.6 s
99.999%	5.26 min	26.3 s	0.86 s
99.9999%	31.5 s	2.62 s	0.08 s

¿Qué es Alta Disponibilidad?

La disponibilidad de un sistema se suele expresar como un “100% menos el tiempo que no está disponible”. Por tanto, cuando hablamos de alta disponibilidad, estamos hablando de un porcentaje muy alto, cercano a 100, que suele medirse en “nueves”, por ejemplo 99%, 99.9%, 99.99%, etc.

Administración de Recursos

Glosario de Conceptos de IT

Con esto, lo que queremos decir, es que el sistema estará funcionando sin problemas, por ejemplo, el 99% del tiempo, es decir, en un año, estaría disponible todas las horas salvo 3 días y 15 horas aproximadamente. Y con una disponibilidad de 99.99% solo estaría parado unos 53 minutos.

Este tiempo que el sistema no está disponible puede deberse a fallos imprevistos, pero también a tareas necesarias de mantenimiento como reinicios, parches de software, etc.

Referencias:

<https://www.nearcrumbs.com/es/que-es-la-alta-disponibilidad/>

https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_72/rzahg/rzahgavailability.htm

<https://www.hostingred.com/servidores/alta-disponibilidad-informacion/>

<https://everac99.wordpress.com/2008/08/19/alta-disponibilidad-que-es-y-como-se-logra/>

Arquitectura de Software

Expositor: Ing. Mariano Gecik
ADR - UTN - FRBA

Arquitectura de Software

¿Qué es la arquitectura de software?

Arquitectura de Software

► Definición:

“Define de manera abstracta, el conjunto de estructuras que la componen. Son elementos de **tecnología, relaciones y propiedades** entre ellas”.

► Objetivos:

“Los sistemas de software son construidos para satisfacer los objetivos del negocio”.

► En que consiste:

Estructuras => Elementos => Relaciones

“La Arq. omite ciertos detalles internos de cada elemento, se abstrae de su dificultad, y se ocupa de lo exterior”.

Interfaces □ Dividen lo privado de lo público, se centra en la complejidad de la interacción de los elementos.

Arquitectura de Software

“NO TODAS LAS ARQUITECTURAS SON BUENAS”

¿Por qué la arquitectura de Software es importante?

- ▶ Usuario dependiente de la rapidez, disponibilidad y confiabilidad de los sistemas.
- ▶ Cliente preocupado por que se implemente una Arq, bajo calendario y presupuesto seleccionado.
- ▶ Project Manager, preocupado porque los equipos trabajen en forma independiente interactuando con disciplina.
- ▶ El Arquitecto se ocupa que los 3 puntos antes mencionados funcionen correctamente y en forma sincronizada

Arquitectura de Software

“¿Cuáles son los interesados de una Arquitectura?”

- ▶ Clientes.
- ▶ Usuarios.
- ▶ Project Manager.
- ▶ Arquitecto.
- ▶ Desarrolladores.
- ▶ Testers.
- ▶ Y más...

Arquitectura de Software

DECISIONES DE DISEÑO A TENER EN CUENTA

1. ¿Procesamiento distribuído o no?.
2. ¿Software dividido en capas?. ¿Cuántas?.
3. ¿Comunicación sincrónica o asincrónica?".
4. ¿Se depende del sisop?
5. ¿Se depende del Hard?.

Arquitectura de Software

CONTEXTO



Arquitectura de Software

Atributos de Calidad

Arquitectura de Software

Atributos de Calidad

Es una propiedad de medida o de testeo que permite indicar que tan bien funciona un sistema y como satisfacer las necesidades de los interesados.

- ▶ Requerimientos Funcionales
- ▶ Requerimientos de calidad del sistema
- ▶ Restricciones

Arquitectura de Software

Atributos de Calidad - Disponibilidad

1. Minimizar las interrupciones del servicio
2. Mitigar posibles fallas que puedan ocurrir

Tácticas:

- ▶ Detectar Fallas
- ▶ Recuperación de Fallas
- ▶ Prevención de Fallas

Arquitectura de Software

Atributos de Calidad - Interoperabilidad

1. 2 o más sistemas pueden intercambiar información vía interfaces y hasta comprender dicha información.
2. Si conocemos las interfaces de los sistemas externos, donde nuestros sistemas operan, podemos diseñar este conocimiento.

Tácticas:

- ▶ Locate - Los sistemas que operan deben ser descubiertos en tiempo de ejecución.
- ▶ Manage Interfaces - Agrega o elimina capacidades de una interface.

Arquitectura de Software

Atributos de Calidad - Adaptabilidad

- ▶ Cambio
- ▶ Costo
- ▶ Riesgo

Arquitectura de Software

Atributos de Calidad - Performance

- ▶ Tiempo
- ▶ Habilidad

Arquitectura de Software

Atributos de Calidad - Seguridad

► Detectar Ataques

- ▶ Detección de Intrusos
- ▶ Denegación de un servicio
- ▶ Verificación de integridad de msj
- ▶ Atraso en los mensajes

► Resistir Ataques

- ▶ Autenticación de actores
- ▶ Límite de acceso
- ▶ Encriptación de Datos

Arquitectura de Software

Atributos de Calidad - Usabilidad

- ▶ “Cuán fácil es para el usuario ejecutar una tarea deseada”
- ▶ “Es una de las formas más fáciles de mejorar la calidad de un sistema”

Arquitectura de Software

Atributos de Calidad - Capacidad de Prueba y Testeo

“Entre el 30 y el 50% del costo de una buena ingeniería en el desarrollo de los sistemas es absorbida por las pruebas”

Arquitectura de Software

OTROS ATRIBUTOS DE CALIDAD

- ▶ **Variabilidad** = **Adaptación al contexto.**
- ▶ **Portabilidad** = **Cambios de plataforma.**
- ▶ **Desarrollo Distribuído** = **Diseño del Soft.**
- ▶ **Escalabilidad** = **Agregar mas recursos, como un server.**
- ▶ **Monitoreo** = **Investigar el sistema mientras trabaja.**
- ▶ **Comerciabilidad** = **No siempre se adapta a lo que necesitamos.**

Arquitectura de Software

Tácticas de Arquitecturas y Patrones

Arquitectura de Software

Tácticas de Arquitectura y Patrones

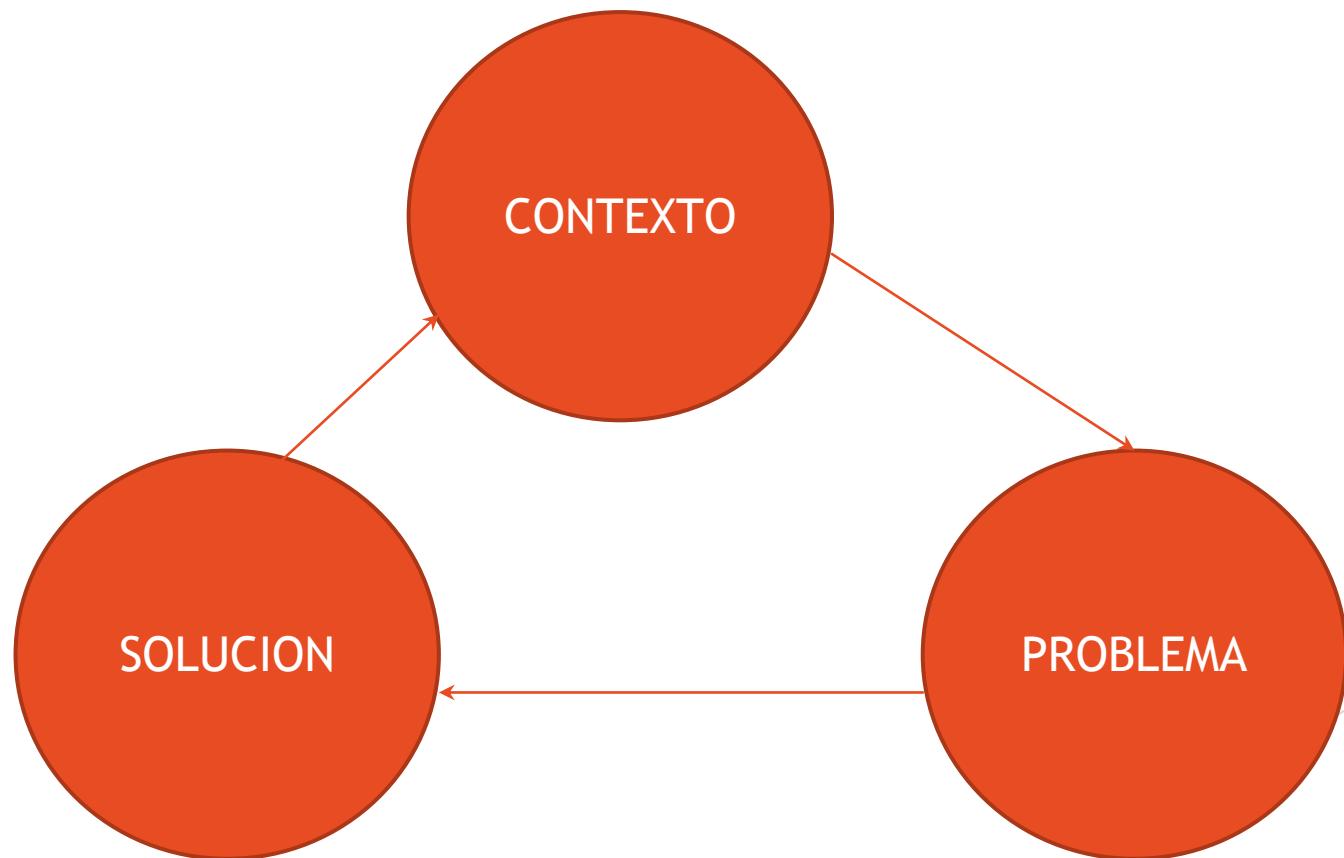
“Tener éxito en el diseño de la arquitectura es complejo y cambiante, por eso los diseñadores buscaron las mejores formas de reutilizar el conocimiento arquitectónico”

Un patrón de Arquitectura:

- ▶ Es un paquete de decisiones de diseño que se encuentra en la práctica.
- ▶ Conoce propiedades que permiten reutilización.
- ▶ Describe un class de arquitecturas.

Arquitectura de Software

PATRONES DE ARQUITECTURA: Relación entre...



Arquitectura de Software

ASR (Requerimientos de Arquitectura
Significativos) en los Ciclos de Vida

Arquitectura de Software

Los ASR en la Arquitectura...

- ▶ Reunir los ASRs de los documentos de requerimientos
- ▶ Reunir los ASRs entrevistando a los interesados
- ▶ Reunir los ASRs entendiendo los objetivos del Negocio

Arquitectura de Software

La Arquitectura en los Proyectos ágiles

“Los métodos y procesos se han agilizado y los proyectos han tenido que cambiar”

Puntos importantes:

1. Alta satisfacción del cliente cuando se entrega un versión.
2. Si cambian los requerimientos, aunque sea tarde, se adapta bien.
3. Entregas de software: Entre semanas y meses, tiempos en general cortos.
4. Hay gran interacción entre la gente del negocio, y la gente de IT. Interacción cara a cara.
5. Motivación del grupo de trabajo.

Arquitectura de Software

“En muchas organizaciones existe una **combinación** de Arquitecturas que se basan en tipos de **proyectos ágiles** y Arquitecturas de paradigmas **estructurados**, no siempre hay que caer en lo que ofrece el mercado como solución.”

Arquitectura de Software

Gestión y Gobierno

Arquitectura de Software

Evaluación de una Arquitectura

1. El Arquitecto debe interesarse por la gestión de proyectos...
2. El PM es la persona que junto al Arquitecto, deben trabajar en conjunto, por la perspectiva de la organización...
3. A Mayor complejidad de proyectos mas útil es la implementación de una arquitectura...

Planificación

1. La planificación de un proyecto sucede constantemente, pero existe un plan inicial para convencer a la dirección de construir el sistema y dar una idea de costo y agenda...
2. El PM debe educar a otros managers para poder corregir desvíos en el desarrollo del software...

Arquitectura de Software

Organización

- ▶ Team Leader - Gestiona las tareas del equipo
- ▶ Developer - Diseñan e implementan los subsistemas de código.
- ▶ Configuration Manager - Ejecutan y construyen test de integración.
- ▶ System Test Manager - Testeo de sistema y testing de aceptación.
- ▶ Product Manager - Representan el marketing.

Arquitectura de Software

Arquitectura Cloud

Arquitectura de Software

Arquitectura Cloud

- ▶ Servicios a Demanda
- ▶ Acceso único de red
- ▶ Pool de recursos
- ▶ Independencia de ubicación
- ▶ Elasticidad rápida
- ▶ Servicios medidos

Arquitectura de Software

Arquitectura Cloud

“Los modelos de desarrollo del cloud se diferencian por quienes son dueños y quienes lo operan”.

Existen 2 modelos básicos que tienen 2 variantes:

- ▶ Cloud Privado
- ▶ Cloud Publico

Arquitectura de Software

Arquitectura en un entorno “Cloud”

“El Arquitecto necesita prestar atención en la adaptabilidad, la usabilidad, la interoperabilidad y el testeo, como haría en otra plataforma”.

“Los atributos de calidad que tienen diferencias, son:

- ▶ Seguridad
- ▶ Performance
- ▶ Disponibilidad

Bibliografía

Software Architecture in Practice

Third Edition:

Capítulos: 1,2,3,4,5,
6,7,8,9,10,
11,12,13,14,15,
16,21,22,23,26

GRACIAS !

Introducción a la Arquitectura de Microservicios (MSA)

Administración de Recursos 2020

Temario

1. Introducción.
2. Enfoque monolítico vs enfoque microservicios.
3. Características de la arquitectura de microservicios.
4. Ventajas de la arquitectura de microservicios.
5. Desventajas de la arquitectura de microservicios.
6. ¿Cuándo utilizar una arquitectura de microservicios?
7. Patrones relacionados.

Microservicios

Estilo de arquitectura en el que una aplicación se desarrolla como un conjunto de servicios que:

- Ejecutan en su propio proceso y se comunican con mecanismos ligeros.
- Se construyen en torno a capacidades de negocio.
- Pueden desplegarse de forma independiente mediante procesos automatizados.
- Poseen una mínima gestión centralizada.
- Pueden escribirse en diferentes lenguajes de programación y utilizar diferentes tecnologías de almacenamiento de datos.

Monolítico vs Microservicios

Estilo de arquitectura monolítico:

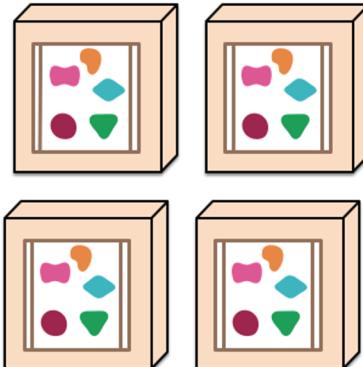
- La aplicación se construye como una unidad (*monolito*).
- Se utilizan las herramientas del lenguaje de implementación para modularizar la aplicación (clases, funciones, namespaces).
- Todos los módulos ejecutan dentro de un mismo proceso y sobre un mismo hardware.
- Cualquier cambio implica construir y desplegar una nueva versión de toda la aplicación.
- El escalado requiere que escale toda la aplicación.

Monolítico vs Microservicios

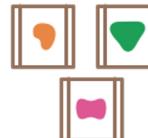
A monolithic application puts all its functionality into a single process...



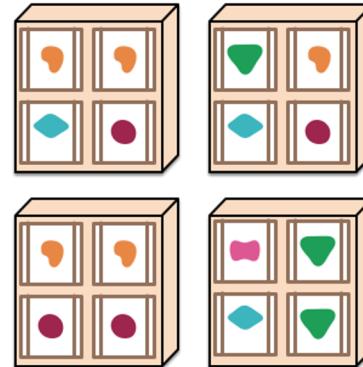
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



... and scales by distributing these services across servers, replicating as needed.



Monolítico vs Microservicios

Ventajas del enfoque monolítico:

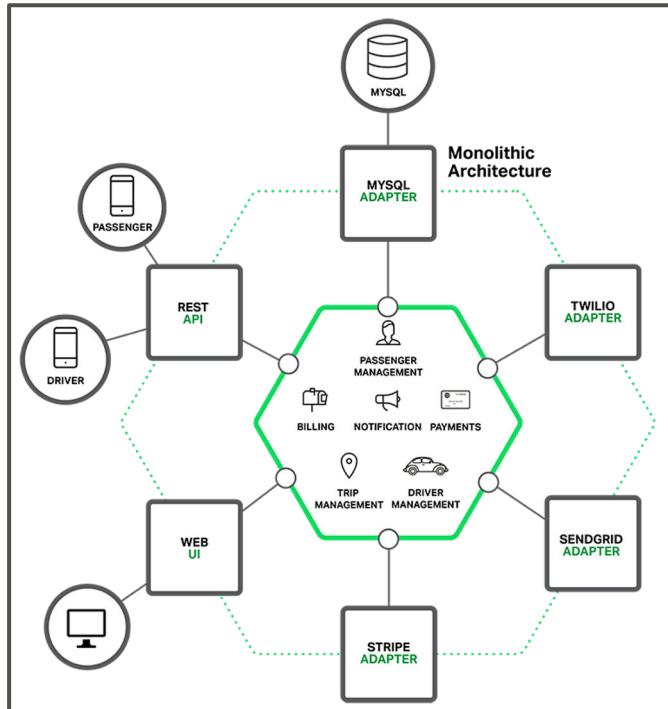
- Simplicidad.
- Facilidad de desarrollo.
- Facilidad de testeo.
- Facilidad de despliegue.
- Facilidad de operación.
- Facilidad de escalado.

Monolítico vs Microservicios

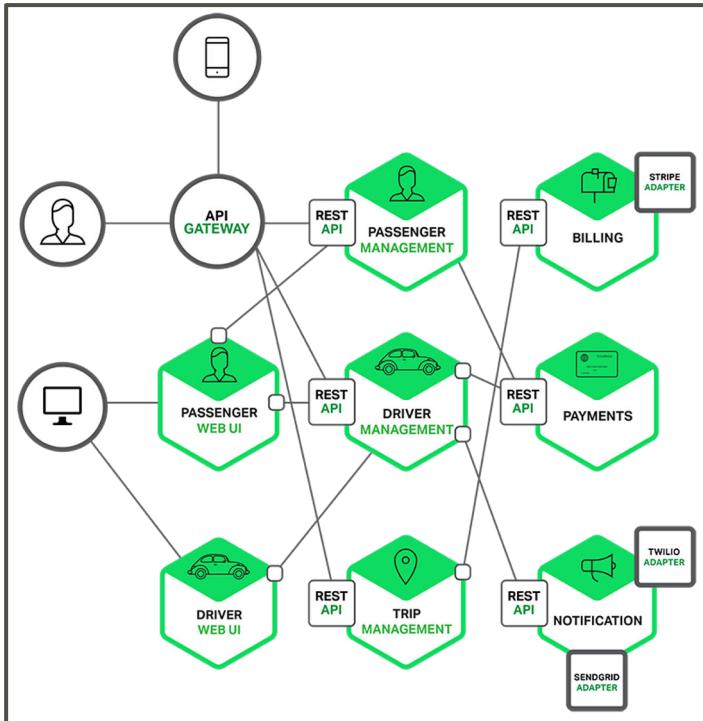
Inconvenientes del enfoque monolítico:

- Ciclos de cambios acoplados.
- A medida que la aplicación crece:
 - Aumenta la complejidad del monolito.
 - Aumenta la dificultad de mantener la modularidad.
 - Se ralentiza el tiempo de inicio de la aplicación.
 - Se dificulta la incorporación de nuevas tecnologías.
 - Disminuye la fiabilidad; un error en cualquier módulo puede potencialmente afectar la disponibilidad de toda la aplicación.
- Escalado ineficiente.
 - Escala toda la aplicación.
 - Todos los módulos ejecutan sobre el mismo hardware.

Monolítico vs Microservicios



Monolítico vs Microservicios



MSA - Características

Principales características de una arquitectura de microservicios:

- Componentización a través de servicios.
- Organizada en torno a funcionalidades de negocio.
- Productos, no proyectos.
- Smart endpoints and dumb pipes.
- Gobierno descentralizado.
- Gestión de datos descentralizada.
- Automatización de infraestructura.
- Diseño tolerante a fallos.
- Diseño evolutivo.

MSA - Características

Componentización a través de servicios.

Las MSA utilizan bibliotecas pero su forma principal de crear componentes de su propio software es segmentarlo en servicios.

- Componente: unidad de software que se puede reemplazar y actualizar independientemente.
- Bibliotecas: componentes vinculados a un programa que se invocan mediante llamadas a funciones en memoria.
- Servicios: componentes fuera del proceso que se comunican mediante mecanismos como peticiones a web services o RPC.

MSA - Características

Componentización a través de servicios (cont).

Utilización de servicios como componentes.

- Se basa en que los servicios son independientemente desplegables.
- Resulta en interfaces de componentes más explícitas.
- Tiene algunas desventajas:
 - Las llamadas remotas son más costosas que las invocaciones en memoria.
 - Se dificulta el cambio de asignación de responsabilidades entre componentes cuando los movimientos de comportamientos cruzan los límites del proceso.

MSA - Características

Organizada en torno a funcionalidades de negocio.

Ley de Conway:

“Cualquier organización que diseñe un sistema (definido ampliamente) producirá un diseño cuya estructura es una copia de la estructura de comunicación de la organización.”

- Melvyn Conway, 1967

MSA - Características

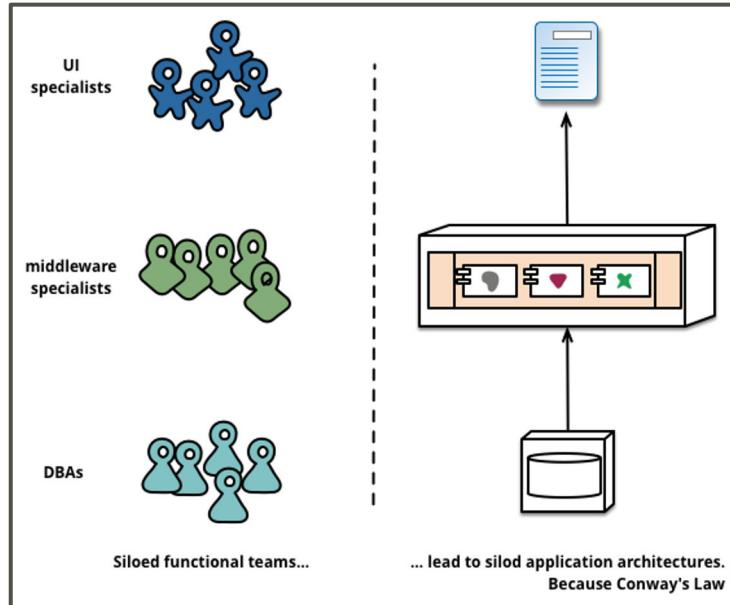
Organizada en torno a funcionalidades de negocio (cont).

Equipos funcionales aislados:

- Incluso los cambios simples pueden llevar a que un proyecto entre equipos tome tiempo y requiera aprobación presupuestaria.
- Los equipos suelen intentar optimizar su rendimiento introduciendo lógica en la parte sobre la que tienen control directo.

MSA - Características

Organizada en torno a funcionalidades de negocio (cont.).



MSA - Características

Organizada en torno a funcionalidades de negocio (cont).

El enfoque en microservicios es diferente:

- Los servicios son definidos en torno a funcionalidades de negocio.
- Cada servicio requiere una implementación amplia de software para el área de negocio a la que pertenece por lo que los equipos son **multifuncionales** y poseen la gama completa de habilidades requeridas.

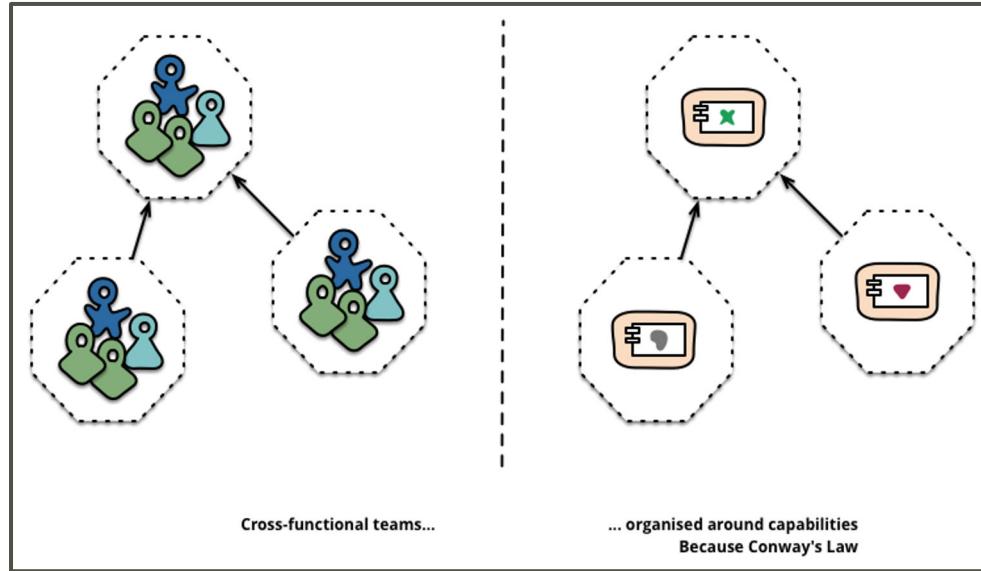
MSA - Características

Organizada en torno a funcionalidades de negocio (cont.).

- Grandes aplicaciones monolíticas también se pueden modularizar en torno a funciones de negocio.
 - Las líneas modulares requieren una gran disciplina para cumplirse.
 - En arquitecturas de microservicios la separación necesariamente más explícita entre servicios componentes facilita la delimitación de los equipos.

MSA - Características

Organizada en torno a funcionalidades de negocio (cont.).



MSA - Características

Productos, no proyectos.

Enfoque de proyecto:

- El objetivo es entregar una pieza de software que en algún momento se considerará terminada.
- Al finalizar, el software se asignará a una organización de mantenimiento y el equipo que lo creó se disuelve.

MSA - Características

Productos, no proyectos.

Mentalidad de producto:

- Un equipo debe poseer un producto durante toda su vida útil.
 - “*You build it, you run it*” (Amazon).
 - El equipo de desarrollo asume toda la responsabilidad del software en producción.
- Se establece una mejor relación entre el producto y la capacidad de negocio a la que pertenece.
- Se concibe al software en una relación continua en la que se busca como éste puede ayudar a sus usuarios a mejorar sus capacidades en el negocio.

MSA - Características

Smart endpoints and dumb pipes.

- La lógica de la aplicación se encuentra en los servicios y no en los mecanismos de comunicación.
- Los mensajes son coreografiados utilizando protocolos simples en lugar de protocolos complejos como WS-Choreography o BPEL u alguna herramienta de orquestación central.
- Los protocolos más utilizados son:
 - HTTP request / response con resource APIs.
 - Lightweight messaging.

MSA - Características

Gobierno descentralizado.

- Un gobierno centralizado tiende a estandarizar una única plataforma tecnológica.
- La segmentación de una aplicación en servicios permite elegir para cada uno de ellos cuál es el stack tecnológico que mejor se adapta a sus necesidades.
- Se suelen implementar patrones de diseño específicos para gestionar los contratos de los servicios buscando reducir su acoplamiento, limitando la necesidad de un gobierno central de los contratos.
- El concepto “*You build it, you run it*” es quizás la mayor expresión de gobierno descentralizado.

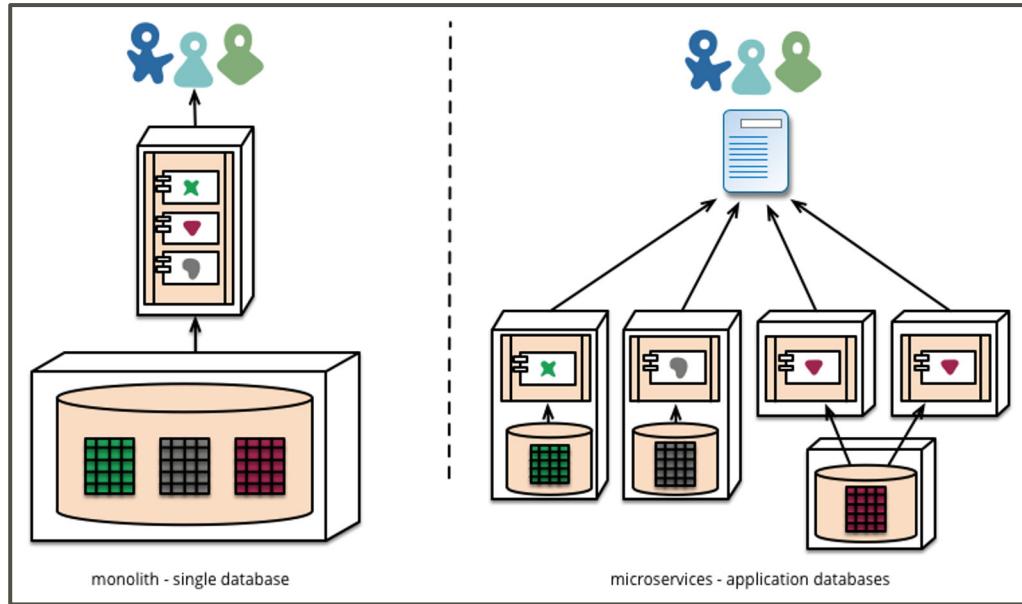
MSA - Características

Gestión de datos descentralizada.

- Descentralización del modelo conceptual.
 - Bounded Context (DDD): Segmenta un dominio complejo en múltiples dominios delimitados y mapea las relaciones entre ellos.
- Descentralización de las decisiones de persistencia de datos.
 - Cada servicio administra su propia base de datos.
 - Diferentes instancias de una misma tecnología.
 - Persistencia políglota.

MSA - Características

Gestión de datos descentralizada (cont.).



MSA - Características

Gestión de datos descentralizada (cont).

Teorema de CAP: es imposible para un sistema distribuido de persistencia de datos proveer simultáneamente:

- Consistencia: Cada lectura recibe la escritura más reciente o un error.
- Disponibilidad: Cada request recibe una respuesta no errónea, sin garantizar que contenga la escritura más reciente.
- Tolerancia al particionamiento: el sistema sigue funcionando incluso si un número arbitrario de mensajes son descartados (o retrasados) entre nodos de la red.

MSA - Características

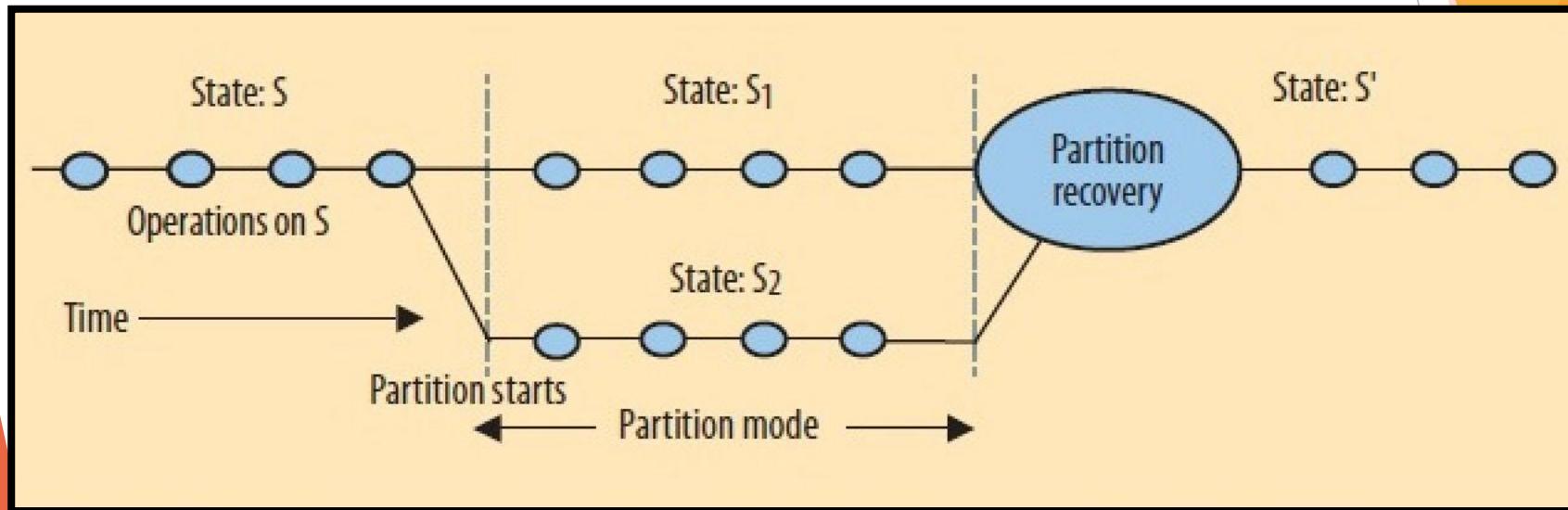
Gestión de datos descentralizada (cont.).

Manejo de actualizaciones.

- Enfoque tradicional de transacciones.
 - Alto nivel de consistencia.
 - Impone un fuerte acoplamiento temporal.
- Transacciones distribuidas.
 - Alto costo de implementación.
- Coordinación de servicios sin transacciones.
 - Reconocimiento explícito de que la consistencia puede ser sólo **consistencia eventual**.
 - Los problemas se resuelven con operaciones de compensación.

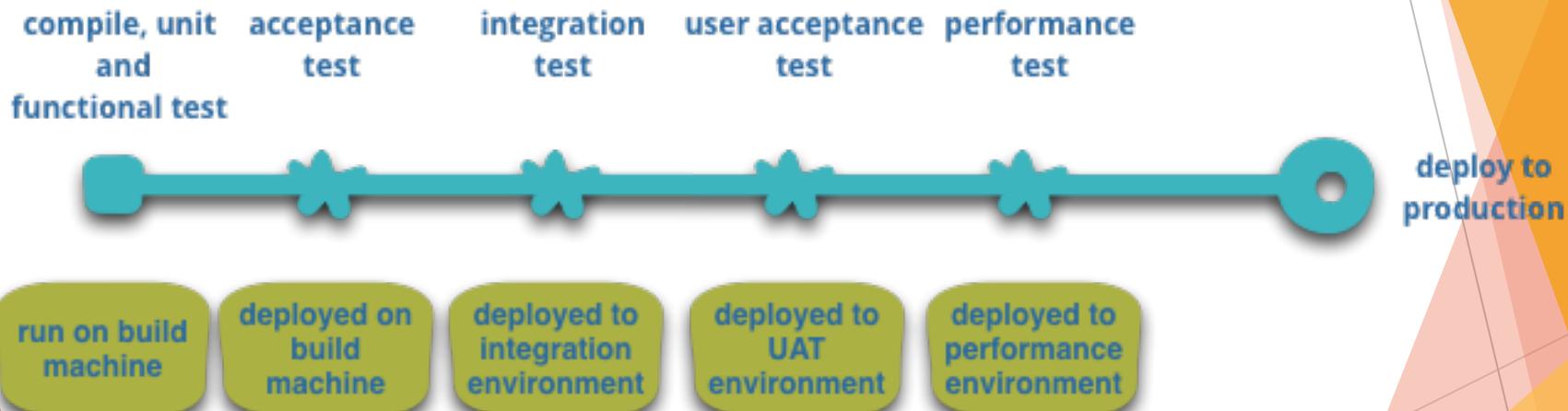
MSA - Características

Gestión de datos descentralizada (cont).



MSA - Características

Automatización de infraestructura



MSA - Características

Automatización de infraestructura (cont.)

- Pipeline de entrega continua (Continuous Delivery)
 - Integración continua del software desarrollado, construyendo ejecutables y ejecutando pruebas automatizadas sobre éstos.
 - Despliegue de componentes en entornos cada vez más similares a producción.
- Automatización en la gestión de microservicios en producción.
 - Orquestación de contenedores.

MSA - Características

Diseño tolerante a fallos

- Las comunicaciones sobre redes son inherentemente poco confiables.
- Las aplicaciones se diseñan para ser resilientes y manejar errores, no sólo para prevenirlos.
 - Se aplican patrones de diseño específicos.
- Fuerte énfasis en el monitoreo en tiempo real de la aplicación.
 - Monitoreo de elementos de arquitectura.
 - Monitoreo de métricas relevantes del negocio.

MSA - Características

Diseño evolutivo

- El diseño modular busca mantener las cosas que cambian al mismo tiempo dentro de un mismo módulo.
 - La propiedad clave de un componente es la noción de su independencia de cambio y actualización.
- Los servicios evolucionan buscando reducir al mínimo el impacto de los cambios en sus consumidores.
- Los servicios se diseñan buscando el mínimo acoplamiento posible al contrato de sus proveedores.
- La utilización de servicios como componentes posibilita planeamientos de despliegues más granulares.

Métricas



Desarrollo de software

Plazo de ejecución de cambios



Despliegue de software

Tasa de falla en cambios



Operación del servicio

Disponibilidad

Frecuencia de despliegue

Tiempo de restauración del servicio

← → *Cuatro métricas clave*

Métricas

Aspecto de la performance de la entrega de software	Élite	Alto	Medio	Bajo
Frecuencia de despliegue	Bajo demanda (varios despliegues por día)	Entre una vez por día y una vez por semana	Entre una vez por semana y una vez por mes	Entre una vez por mes y una vez por semestre
Plazo de ejecución de cambios	Menos de un día	Entre un día y una semana	Entre una semana y un mes	Entre uno y seis meses
Tiempo de restauración de servicio	Menos de una hora	Menos de un día	Menos de un día	Entre una semana y un mes
Tasa de falla en cambios	0-15%	0-15%	0-15%	46-60%

MSA - Ventajas

- Facilita el *continuous delivery* de aplicaciones grandes y complejas.
 - Mejora la mantenibilidad.
 - Descomposición modular de la complejidad.
 - Ciclos de evolución de componentes más desacoplados.
 - Límites y contratos de servicios más explícitos.
 - Facilita la incorporación de nuevas tecnologías.
 - Permite el despliegue y escalado independiente de servicios.
 - Permite organizar los esfuerzos de desarrollo en torno a múltiples equipos autónomos.

MSA - Ventajas

- Los servicios son relativamente pequeños:
 - Son más fáciles de entender para un desarrollador.
 - La aplicación inicia en menor tiempo.
 - Aumento de productividad de desarrollo.
 - Mejora los tiempos de implementación.
- Mejora el aislamiento de fallos.
- Elimina compromisos a largo plazo con un stack tecnológico.

MSA - Inconvenientes

- Aumento significativo de la complejidad.
 - Complejidad adicional propia de un sistema distribuido.
 - Requiere la implementación de mecanismos de comunicación entre servicios y el manejo de fallos parciales.
 - Se dificulta el testing de interacción entre servicios.
 - Aumenta la complejidad de implementación, gestión y monitoreo.
 - Se dificulta la detección de errores en tiempo de ejecución.
 - Las herramientas/IDEs están orientados al desarrollo de aplicaciones monolíticas.

MSA - Inconvenientes

- Complejidad de la arquitectura de persistencia de datos particionada.
 - Son muy comunes las transacciones de negocio que requieren actualizaciones en repositorios pertenecientes a múltiples servicios.
 - Las transacciones distribuidas no siempre son una opción.
 - El enfoque de persistencia eventual es más complejo que el enfoque tradicional transaccional.
- Las soluciones que abarcan múltiples servicios son más complejas y requieren una ajustada coordinación entre equipos.
- Aumenta el consumo de memoria de la aplicación.

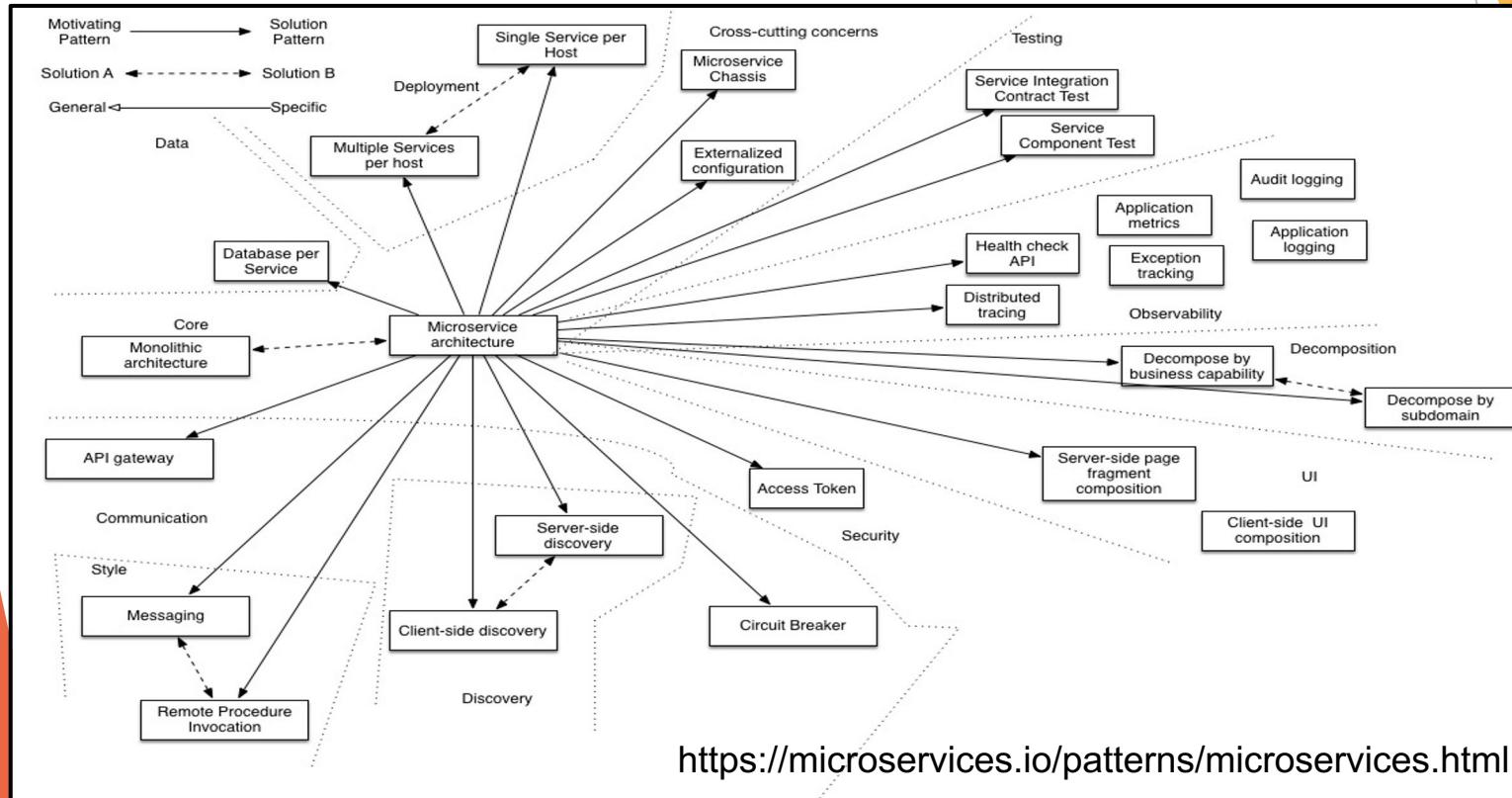
¿Cuando utilizar una MSA?

- Como toda arquitectura, tiene su compensación.
- En diversas situaciones una arquitectura monolítica es la mejor opción.
- Generalmente, las primeras versiones de una aplicación no tienen los problemas que resuelve este enfoque.
- Si se requiere de una aplicación en el menor tiempo posible se debe considerar que el uso de una arquitectura elaborada y distribuida ralentizará el desarrollo.
- A medida que una aplicación va creciendo y surgen los desafíos de escalado y segmentación funcional, se complejiza la descomposición de un monolito en un conjunto de servicios.

¿Cuando utilizar una MSA?

- Las MSA tienen serias consecuencias en la operación de la aplicación por lo que existen un conjunto de capacidades a considerar al evaluar su factibilidad de implementación:
 - Rápido aprovisionamiento.
 - Monitoreo básico.
 - Técnico (latencia, disponibilidad del servicio, etc).
 - De negocio (cantidades de pedidos)
 - Rápida implementación.
 - Generalmente, mediante un pipeline.

Patrones relacionados



Referencias

- <https://martinfowler.com/articles/microservices.html>
- <https://microservices.io/patterns/microservices.html>
- <https://www.nginx.com/blog/introduction-to-microservices/>
- <https://developers.redhat.com/promotions/microservices-for-java-developers/>
- <https://martinfowler.com/bliki/ContinuousDelivery.html>
- <https://martinfowler.com/articles/microservice-trade-offs.html>
- <https://martinfowler.com/bliki/MicroservicePrerequisites.html>
- <https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed/>
- <https://services.google.com/fh/files/misc/state-of-devops-2019.pdf>

Benchmark

Proceso sistemático y continuo que permite evaluar comparativamente los productos, servicios y procesos de trabajo en organizaciones. Estas mediciones son las que van a definirnos cuál es el rendimiento de lo que estamos evaluando y así permitir la toma de decisión correspondiente.

Permite:

- Comparar elementos a través de características claves para la solución
- Obtener resultados objetivos
- Obtener la mejor relación costo/beneficio
- Comprobar si los elementos estudiados se adecuan a las necesidades

Etapas

- Determinar el elemento de estudio: Elegir los factores y variables claves. Seleccionar las opciones disponibles en el mercado.
- Preparar el entorno de prueba: Recopilar requerimientos del tipo de Benchmark a realizar. Realización de tareas previas a la etapa de ejecución. Preparación del ambiente en donde se realizará la prueba.
- Realizar benchmark: Someter elementos a las pruebas. Medir las respuestas para cada variable analizada y determinar la diferencia respecto a lo que se compara.
- Analizar resultados: Descartar elementos estudiados que en base a los resultados, no cumplen con las necesidades. Realizar informes para mostrar los resultados obtenidos. Determinar si se requiere recalibrar el benchmark y reiniciar. Desarrollar planes de acción.

Tipos

- Sintéticos: Mide el rendimiento de un componente individual del hardware, normalmente llevando el componente escogido a su máxima capacidad.
 - Whetstone: Provee una medida muy razonable de rendimiento de monoprocesadores de aritmética flotante. (Unidad = MWIPS)
 - Dhrystone: Para medir la velocidad del sistema en cuanto a rendimiento no numérico. Mide la capacidad del procesador. (Unidad = DPS)
- Aplicaciones: Simulan una carga de trabajo para medir el comportamiento global del equipo. Tienen gran cantidad de código y de requerimientos de almacenamiento de datos.
- Test de Bajo nivel: Miden directamente el rendimiento de los componentes. Ej: reloj d CPU
- Test de Alto nivel: Están más enfocados a medir el rendimiento de la combinación componente/controlador/SO de un aspecto específico del sistema. Ej: E/S con archivos

Más conocidos (faltan varios)

- HINT: Resolver un problema matemático cuya solución se puede ir mejorando. (QUIPS)
- TPC: (TPC-C) Mide los resultados en transacciones por minuto. Se simula la carga de trabajo y solicitan la ejecución de una mezcla de cinco tipos de transacciones: New-Order, Payment, Order-Status, Delivery, Stock-Level. Las transacciones simuladas deben cumplir:
 - Atomicidad: Las transacciones se confirmen o se aborten. Si falla, no a medias.
 - Consistencia: Solo se ejecutan aquellas operaciones que no van a romper la reglas.
 - Aislamiento: una operación no puede afectar a otras.
 - Durabilidad: Una vez realizada la operación, no se deshace aunque falle el sistema.

Almacenamiento

- Centralizado: Un único sistema central.
 - Sistemas de almacenamiento accedidos de manera directa (DAS): Accesible mediante controladoras e interfaces conectadas en forma directa a los servidores implementando protocolos de comunicación específicos.
 - Red de almacenamiento de datos (SAN): “ ” ” ” ” ”
 - Red IP de almacenamiento (NAS): Almacenamiento accesible por medio de redes IP de transferencia de datos.
 - Servicios de almacenamiento en la nube

Disponibilidad (Availability): Es la posibilidad de poder continuar dando servicios de storage en el evento de una falla de hardware o software.

Rendimiento (Performance): Es la métrica usada para definir la velocidad de un sistema de almacenamiento. Existen 3 métodos de medición: I/OxSeg, MBxSeg, response time (ms).

Soluciones de resguardo offline

- Almacenamiento de datos en cintas magnéticas: Tipos Drives manuales, semi-automaticos, automáticos, virtuales.
 - RAID
 - RAID 0: Los datos se dividen en pequeños segmentos y se distribuyen entre varios discos. Lectura y escritura rápida. No tolerancia a fallos.
 - RAID 1: Discos adicionales sobre los que se realiza una copia bloque a bloque. Lectura rápida. Tolerancia a fallos. Costoso.
 - RAID 3: Dedica un disco al almacenamiento de información de paridad. Altas tasas de transferencia y confiabilidad. Rendimiento de transacción pobre.
 - RAID 5: Utiliza bloques distribuidos para el almacenamiento de información de paridad que le permite ofrecer tolerancia a fallas. Escrituras simultáneas.
 - RAID 6: Implementa doble paridad usando dos discos. Permite la falla de dos discos de manera simultánea. Costoso.

Plan de contingencia y recuperación

Recovery Time Objective (RTO): Tiempo que pasará una infraestructura antes de estar disponible.

Recovery Point Objective (RPO): Cuantos datos está dispuesta a perder la organización antes de estar disponible.

(unos cuadros que no se entienden una chota)

Estrategias de protección de datos

- Copias de resguardo en discos locales y externos: Rápido, integración con apps y BDD. Costoso. No transportable a otro datacenter. Si la falla se produce en el dc no recuperable.
 - Copias de resguardo periódicas en cinta: Bajo costo. Transportable a otro dc. Demora más.
 - Replicación de datos en sitio externo: Resguardo fuera de dc. No continuar operación.
 - Replicación de datos en dc externo implementado como sitio de contingencia: Continuar operación. Rapidez. Costo alto.

Medidas preventivas: evitar ocurrencia. de detección: para detección. correctivas: recuperar la op.

Tipos de computadores

- Mainframe: Computadora central destinada a procesos críticos a gran escala con gran confiabilidad y capacidad de procesamiento. Es una potente herramienta para dar soporte en línea a miles de usuarios. Tiene la capacidad de ser reparado sin detener los servicios. Muy costosa. Incompatible con otras plataformas. Decenas de procesadores.
- Supercomputadora: Es una computadora con capacidad de cálculo muy superiores a una PC común. Se usan para aplicaciones especializadas que requieren enormes cantidades de cálculos matemáticos. Costosa, consume mucha energía. Miles de procesadores.

Paralelismo Funcional: Procesos, hilos, redundancia. De datos: Vectores, matrices.

Métricas de rendimiento

- MIPS: Millones de instrucciones por segundo.
- MFLOPS: Millones de operaciones en punto flotante por segundo.

Servidores

- Tipo torre: Su infraestructura es una PC básica. Empresas pequeñas y medianas
- En rack: Requieren infraestructura específica para su instalación y funcionamiento. Para dc
- Blade: Ideales para reducir espacio y disminuir el consumo de energía.

Cluster

Grupo de computadoras individuales (denominadas “nodo”) trabajando en forma conjunto bajo una solución de software y conectividad que se ponen al servicio del procesamiento de una determinada tarea.

Componentes: Nodos, almacenamiento, SO, conexiones red, protocolos de comunicación, soft.

Características: Rendimiento (procesa grandes volúmenes), Disponibilidad (si falla un nodo no impide operatividad), Eficiencia (aprovecha recursos en su totalidad), Escalabilidad (nuevos nodos)

Tipos de Cluster

- LB-C (de balanceo de carga): Rutea la totalidad de la carga de trabajo por medio de un servidor, el cual la distribuye entre los nodos de procesamiento activo.
- HP-C (de alta performance): Explota el potencial del procesamiento en paralelo entre múltiples computadoras. Es el más indicado para funciones complejas.
- HA-C (de alta disponibilidad): Diseñado para mantener computadoras redundantes que pueden convertirse en sistemas de backup en caso de una falla del sistema.

Grid computing: Permite la organización de un “pool” de múltiples recursos. Los recursos pueden ser accedidos por una gran cantidad de usuarios. No es un cluster, pero puede serlo o contenerlo.

Cloud Computing: Permite disponibilidad de implementación y crecimiento inmediata.

Dependencia de los proveedores. Exposición de datos sensibles ante terceros.

- IaaS (Infrastructure as a Service): Procesamiento, Cache, Almacenamiento, Streaming.
- PaaS (Platform as a Service): Desarrollo de aplicaciones.
- SaaS (Software as a Service): Hosting de aplicaciones en la nube.
- MSP (Managed Service Provider): Backup/Restore, Monitoreo.

Software base

Sistema Operativo: Es un conjunto de software de sistema que controla como las aplicaciones acceden al hardware y la manera en que los usuarios acceden a recursos.

Compuesto por: Gestor de procesos, gestor de MP, gestor de Memoria Secundaria, gestor de E/S. Sistema de archivos, sistema de protección, sistema de comunicaciones. Programas de sistema.

Clasificación

- Administración de tareas:
 - *Monotarea*: Solamente puede ejecutar un proceso en un momento dado.
 - *Multitarea*: Es capaz de ejecutar varios procesos de forma alternada.
- Administración de usuarios:
 - *Monousuario*: Si sólo permite ejecutar los programas de un usuario a la vez.
 - *Multiusuario*: Permite que varios usuarios ejecuten simultáneamente, accediendo a la vez a los recursos. Utilizan métodos de protección de datos.
- Manejo de Recursos:
 - *Centralizado*: Permite utilizar los recursos de una sola computadora.
 - *Distribuido*: Permite utilizar los recursos de más de una computadora a la vez.
- Manejo de eventos:
 - *Tiempo Real*: multitarea, con intercambio de procesos basado en prioridades y eventos externos.
 - *Tiempo compartido*: intercambian tareas en función de interrupciones de reloj.

Criterios de selección

- *Aplicaciones*: Software que se requiere utilizar. Disponibilidad de otras aplicaciones. Disponibilidad de herramientas de gestión y su facilidad de extensión.
- *Estabilidad*: Evolución del núcleo y las interfaces. BUG, Instalación y actualización.
- *Confiabilidad*: Consumo del sistema base. Optimización del uso de recursos (Deadlocks).
- *Seguridad*: Reporte y resolución de BUG. Administración de credenciales y permisos.
- *Escalabilidad*: Capacidad de clusterización. Modularidad y personalización.
- *Usabilidad*: Facilidad de uso, de aprendizaje y de automatización de tareas. Interfaz GUI.
- *TCO*
 - *Disponibilidad de RRHH capacitados*:
 - SO privativos: Disponibilidad Guru → costoso, Disp. novatos → económico
 - SO no privativos: Aprendizaje más lento. Riqueza técnica superior.
 - *Licenciamiento y soporte*:
 - SO privativos: Licenciamiento pago (WINDOWS, UNIX)
 - SO no privativos: Licenciamiento gratuito (BSD, LINUX)
 - *Soporte*: Comercial (pago) o de la comunidad (voluntario).

Tipos de licenciamiento

- Software propietario: Utiliza una licencia de código cerrado. Código fuente inaccesible.
- Software libre: (COPYLEFT) El trabajo realizado se mantiene disponible de forma libre.
L0) Ejecutar el programa, para cualquier propósito. L1) cambiarlo para que haga lo que se requiera (OPEN SOURCE). 2L) Redistribuir copias. L3) Mejorar y publicar sus mejoras.

Open source: Software que se pone a disposición con el código fuente, y algunos otros derechos.

Freeware: Software propietario que puede ser utilizado sin costo de licencia.

Bases de datos

Que se busca con los SGBD: Abstracción, Independencia, Redundancia mínima, Consistencia, Seguridad, Integridad, Respaldo y Recuperación, Control de concurrencia, Tiempo de Respuesta.

Aspectos funcionales

- Distribución y replicas: La replicación es un conjunto de herramientas para copiar y distribuir datos y objetos de una BDD a otra y sincronizarlas para mantener la coherencia.
- Concurrencia y niveles de aislamiento: (MVCC) Provee de cada usuario conectado con la base de datos una “foto” de la BDD para esa persona. Ningún cambio realizado será visualizado por otros usuarios hasta que la transacción haya sido confirmada.
- Tipos de Índices:
 - Índices Hash
 - Árbol B: Utilizado en la mayoría de los índices de BDD.
 - Índices bitmap: Consumen menor almacenamiento que Árbol B. En un índice bitmap se crea un mapa de bits para cada valor clave del índice.
 - Árbol R-/R+: Similar al Árbol B pero optimizado para acceder a información multidimensional. Generalmente utilizado para coordenadas geográficas.
 - Índice parcial: Se aplica solo a aquellas filas que cumplen una determinada condición, permitiendo generar un índice para un propósito más específico que un índice común, evitando indexar filas innecesarias.
 - Índices de expresión: Se basan en el indexado de una función o expresión aplicada sobre las columnas del registro, y es eso lo que se indexa. Ej.: por upper(col1).
 - Fulltext: Permite la búsqueda de n palabras combinadas.
- Backups: (no te explica una mierda, solo dice que tipos de backups tiene c/ BDD)
- Vistas materializadas: Una vista es una consulta guardada en la BDD, que permite reutilizarse como si reutilizamos una consulta ya hecha. En este caso lo que se guarda no es el resultado de la consulta, sino las relaciones utilizadas para volver a ejecutarla.
- Optimización de consultas: Proceso de selección del plan de evaluación de las consultas más eficiente de entre las muchas estrategias generalmente disponibles para el procesamiento de una consulta dada.
- Particionamiento: Técnica para dividir tablas grandes en tablas pequeñas más manejables, y permite que se manipulen desde el punto de vista de la aplicación de una manera uniforme. Mejoran la performance. Los hay por rango y por hash.

Comparación de rendimientos: **TPC-H** es un punto de referencia apoyo a las decisiones que consiste en una cadena de consultas ad-hoc...

Flashback: Característica para recuperar de errores humanos. Permite a los usuarios ver el estado del dato, trasladado a un instante anterior sin requerir de cambios estructurales a la BDD.

Bases de datos no SQL

- Ventajas: Resuelven problemas de escalabilidad horizontal, Manejo de enorme cantidades de datos, Fácil de escalar, Excelentes tiempos de respuesta en almacenamiento y recuperación.

- Desventajas: No cumplen ACID, Utilizan mayormente memoria en vez de disco como la principal ubicación de escritura, Falta de madurez, Falta de estándares, Aplicación para BI.

Clasificación

- Clave-Valor: Los datos se almacenan en pares del tipo clave-valor.
- Column Family: Permiten almacenar claves mapeadas a valores y esos valores agrupados en múltiples familias de columnas siendo cada columna un mapa de datos.
- Basadas en documentos: La base de datos almacena y recupera documentos que pueden estar en XML, JSON o BSON.
- Basadas en grafos: Permiten almacenar entidades y relaciones entre esas entidades. Tanto los nodos como las relaciones tienen sus propiedades asociadas.

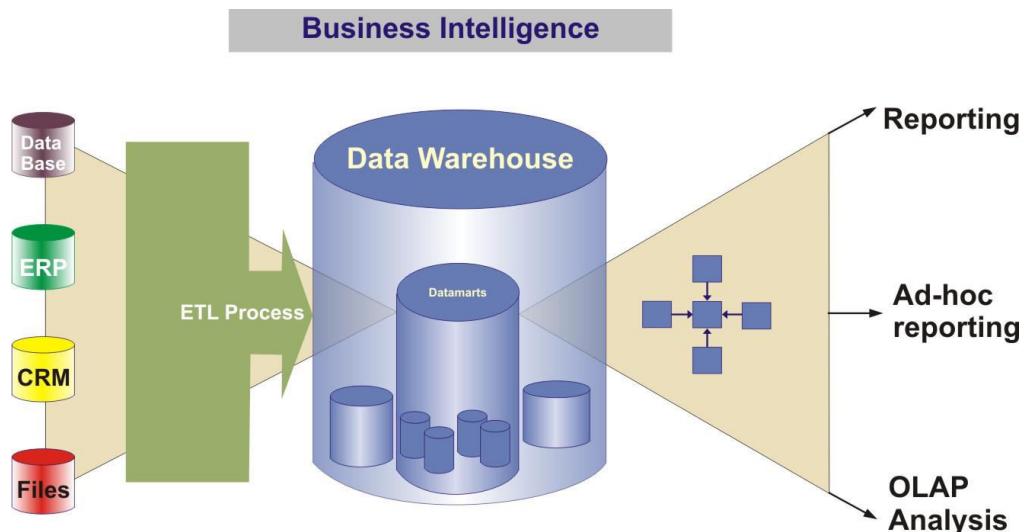
Business Intelligence (BI)

Abarca las tecnologías de datawarehousing los procesos en el ‘back end’, consultas, informes, análisis y herramientas para mostrar información y procesos en el ‘front end’.

Objetivo: asistir en la toma de decisiones y, posteriormente descubrir conocimiento desconocido para la organización hasta el momento.

Beneficios:

- *Tangibles*: Reducción de costes y tiempo, generación de ingresos.
- *Intangibles*: Tener disponible la información para la toma de decisiones.
- *Estratégicos*: Todos aquellos que nos facilitan la formulación de la estrategia.



Virtualización de sistemas

Es un modo de abstraer aplicaciones y sus componentes subyacentes del hardware que lo soporta y presentar una vista lógica o virtual de esos recursos.

Objetivos: Mayor nivel de performance, escalabilidad, confiabilidad/disponibilidad, agilidad.

Creación de un dominio unificado de seguridad y gestión.

Vistas: La virtualización puede crear una vista artificial donde varias computadoras individuales se muestran como una sola o donde una computadora se muestra como varias individuales.

Capas

Virtualización de acceso: Hardware y software que permite que prácticamente cualquier dispositivo acceda a cualquier aplicación y viceversa, sin que ninguno tenga mucho conocimiento del otro. Se mete en la comunicación entre dispositivos.

Virtualización de aplicaciones: Software que permite que las aplicaciones corran en diferentes sistemas operativos y plataformas de hardware. Existen 2 formas:

- Del lado del cliente: Se crea un ambiente protegido que posibilita que las aplicaciones estén aisladas de otras y del SO base.
- Del lado del servidor: La colocación de aplicaciones del lado del servidor en un ambiente virtual. Aplicaciones que son incompatibles pueden funcionar juntas.

Virtualización de Procesamiento: Hardware y software que oculta la configuración de hardware a los SO o aplicaciones. Tipos:

- Un sistema único se muestra como varios: El software de máquina virtual permite que todo el software que constituye un sistema se encapsule en un archivo de máquina virtual. La virtualización del SO permite que muchas aplicaciones corran bajo un solo SO y darle a cada una un ambiente aislado y protegido.
- Varios sistemas se muestran como uno solo: Los monitores de procesamiento paralelo permiten que varias máquinas ejecuten las mismas aplicaciones o componentes con el objetivo de reducir el tiempo de procesamiento. Los monitores de carga de trabajo permiten que varias instancias de una misma aplicación puedan correr simultáneamente en varias máquinas.
- Varios sistemas se muestran como uno solo: Los monitores de alta disponibilidad/tolerancia a fallas/recuperación de desastres permiten que las personas utilicen servicios de protección contra fallas. La virtualización de memoria o memoria cache distribuida permite que varios sistemas compartan sus memorias internas.

Virtualización de Red: Hardware y software que presenta una vista la red que difiere de la vista física. O hacer que varias redes se presenten como una sola. Funciones:

- Ruteo: El tráfico dirigido a clientes o servidores remotos se envía de una red a otra hasta alcanzar el destino.
- Network Address Translation (NAT): Una dirección de red puede ser realmente usada por un número de clientes o servidores. Osea, IP única para un sector.

- Network Isolation: los clientes y servidores solamente pueden ver (y por ello comunicarse con) sistemas específicos.

Virtualización de almacenamiento: Hardware y software que oculta dónde están los sistemas de almacenamiento y qué tipo de dispositivos realmente almacenan aplicaciones y datos. Funciones:

- Viabilidad de sistemas distribuidos de archivos: Los dispositivos de almacenamiento remoto se arman para que parezcan directamente conectados al sistema.
- Creación de volúmenes artificiales de almacenamiento: Varios dispositivos de almacenamiento se unen para crear la imagen de un dispositivo único mucho mayor.
- Creación de arreglos de volúmenes de almacenamiento: Los datos y las aplicaciones se pueden distribuir sobre un número de dispositivos y servidores de almacenamiento para mejorar la performance global.
- Viabilidad de mayor control sobre el espacio de almacenamiento: los dispositivos se pueden segmentar en varios “file systems”, lo que permite una utilización más eficiente del almacenamiento.
- Viabilidad de compartir dispositivos de almacenamiento entre sistemas incompatibles: SOs usan mecanismos diferentes para almacenar y recuperar aplicaciones y datos. La virtualización permite que todos ellos comparten los dispositivos de almacenamiento y los archivos que contienen.

Seguridad de ambientes virtuales: Se trata de tecnología de software que controla el acceso a diferentes elementos en un ambiente virtual y previene usos no autorizados o maliciosos

Gestión de ambientes virtuales: Software que permite que varios sistemas sean provistos y gestionados como si fueran un recurso individual. Si los ambientes de vuelven más complejos, esta capa se vuelve más importante. Funciones:

- Creación y provisión ambientes virtuales o componentes de los mismos
- Monitoreo y control de su ejecución
- Análisis del registro de ejecución para encontrar problemas
- Optimización del uso de ambientes o componentes

Big data: Combinación de virtualizaciones de procesamiento y almacenamiento. Permiten gestionar conjunto muy grandes de datos y dispositivos de almacenamiento.

Virtualización de servidores: Se utiliza y aplica al uso de tecnología de máquina virtual o virtualización de sistema operativo para permitir que un solo sistema físico soporte múltiples cargas.

(FALTAN)

Clusters

Virtualización de escritorio

Sistemas de alta performance

Procesamiento extremo de transacciones

Arquitectura de Software

Es el conjunto de estructuras que componen al software. En sí, son elementos de tecnología, relaciones y propiedades entre ellas. Las estructuras contienen elementos y sus relaciones. La arquitectura trata de omitir ciertos detalles internos de cada uno de estos elementos, para abstraerse de su dificultad. Es importante para:

- El usuario: que la rapidez, de la disponibilidad y de que sea confiable.
- El cliente: que sea implementada conforme al calendario y al presupuesto.
- El Project Manager: que permita que los equipos trabajen independientemente.
- El arquitecto: que todos los puntos anteriores funcionen correcta y sincronizadamente.

Requerimientos

- Requerimientos Funcionales: Estos definen, qué tiene que hacer el sistema, y cómo debe actuar en tiempo de ejecución.
- Requerimientos de calidad del sistema: Estos definen, las características de los requerimientos funcionales, a lo largo de todo el producto. Ej: rapidez para hacer un ReqF.

Restricciones: La restricción es una decisión de diseño que debe tomarse. Ej.: elección de lenguaje.

Atributos de calidad

- Disponibilidad: Minimizar las interrupciones de servicio y mitigar las posibles fallas. Hay 3 tipos de tácticas: detectar, recuperar y prevenir fallas.
- Interoperabilidad: 2 o más sistemas pueden intercambiar y comprender información importante vía interfaces.
- Adaptabilidad: Posibilidad de cambio al menor costo y riesgo.
- Performance: Habilidad que tienen los sistemas para contestar a los eventos que ocurren en tiempo y forma. Hay 2 tipos de tácticas: disminuir demanda o aumentar gestión de rec.
- Seguridad: Habilidad que tienen los sistemas, para proteger datos e información de un acceso no autorizado. Hay 2 tipos de tácticas: Detectar o resistir ataques (revocar o lock).
- Capacidad de prueba y testeo: Facilidad con la que el software puede hacer para demostrar sus fallos a través de pruebas.
- Usabilidad: Cuán fácil es para el usuario, ejecutar una tarea deseada y cuál es el tipo de usuario que soporta. Comprende el aprendizaje, eficiencia, confianza, etc.

Tácticas y patrones de arquitecturas

Son los caminos para realizar un buen diseño de estructuras que pueden ser reutilizadas.

Los patrones de arquitectura establecen relaciones entre: Contexto, problema y solución.

Las definiciones de patrones son estrictas pero toleran pequeñas modificaciones.

Tipos de patrones

Requerimientos de arquitectura significativos (ASR): son aquellos que tienen un efecto profundo sobre la arquitectura. La arquitectura bien podría ser diferente en ausencia de tal requisito

Métricas

Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo. Se usan para: Evaluar el proyecto, Seguir riesgos potenciales, Predecir problemas críticos, Ajustar el flujo y las tareas, Evaluar la habilidad del equipo en controlar la calidad.

Indicador: Métrica o combinación de métricas que da una visión profunda del proceso y proyecto de software o el producto. Se recopilan medidas y desarrollan métricas para obtener indicadores.

Métricas de software

- Medidas directas: Líneas de código producidas (LDC), Velocidad de ejecución, Memoria utilizada, Defectos en un periodo de tiempo.
- Medidas indirectas: Funcionalidad, Calidad, Complejidad, Facilidad de mantenimiento

Métricas orientadas al tamaño: (Errores, defectos, \$, páginas documentadas) por KLDC (miles LDC)

Métricas orientadas a la función (Errores, defectos, \$, páginas documentadas) por PF

Puntos de Función

Se basa en el diseño lógico del sistema y en características externas.

Objetivos: Medir independientemente de la tecnología, Proporcionar factor de normalización para comparar software.

Elementos:

- Entradas (External Input): Entran desde fuera de la aplicación.
- Salidas (External Output): Datos que salen de la aplicación.
- Consultas (External Query): Combinación de E/S que se obtiene de una búsqueda.
- Archivos internos (Internal Logic File): Grupo de datos relacionados, mantenidos y utilizados dentro de la aplicación
- Archivo de interface externa (External Interface File): Grupo de datos relacionados, mantenido por otra aplicación.

Para cada parámetro externo se indica su complejidad (baja, media o alta) en base al número de campos, cantidad de ficheros que referencia y el número de registros (en ficheros).

Medida de calidad del software

- Corrección: Grado en que lleva a cabo su función (defectos por KLDC).
- Facilidad de mantenimiento: Facilidad para corregir un error o realizar un cambio.
- Integridad: Habilidad para resistir ataques (accidentales e intencionados).
- Facilidad de uso: Que tan amigable es con el usuario.

Herramientas de estimación: Juicio experto, datos históricos de proyectos similares. Estimaciones individuales. Paramétrica (Definir una fórmula en función del esfuerzo por unidad de trabajo).

PERT (Por 3 valores): Calcula la duración de cada actividad con promedios ponderados de estimaciones. La duración del proyecto es la suma de la duración de las tareas del camino crítico.