

Arquitectura de Aplicaciones

Guía didáctica



Unidad Académica Técnica y Tecnológica

Tecnología Superior en Transformación Digital de Empresas

Arquitectura de Aplicaciones

Guía didáctica

Carrera	PAO Nivel
▪ <i>Tecnología Superior en Transformación Digital de Empresas</i>	III

Autor:

Mora Camacho Christian Fernando



D S O F _ 2 0 4 8

Asesoría virtual
www.utpl.edu.ec

Universidad Técnica Particular de Loja

Arquitectura de Aplicaciones

Guía didáctica

Mora Camacho Christian Fernando

Diagramación y diseño digital:

Ediloja Cía. Ltda.

Telefax: 593-7-2611418.

San Cayetano Alto s/n.

www.ediloja.com.ec

edilojacialtda@ediloja.com.ec

Loja-Ecuador

ISBN digital - 978-9942-39-809-3



**Reconocimiento-NoComercial-CompartirIgual
4.0 Internacional (CC BY-NC-SA 4.0)**

Usted acepta y acuerda estar obligado por los términos y condiciones de esta Licencia, por lo que, si existe el incumplimiento de algunas de estas condiciones, no se autoriza el uso de ningún contenido.

Los contenidos de este trabajo están sujetos a una licencia internacional Creative Commons **Reconocimiento-NoComercial-CompartirIgual 4.0 (CC BY-NC-SA 4.0)**. Usted es libre de **Compartir – copiar y redistribuir el material en cualquier medio o formato. Adaptar – remezclar, transformar y construir a partir del material citando la fuente, bajo los siguientes términos: Reconocimiento- debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios.** Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante. **No Comercial-no puede hacer uso del material con propósitos comerciales. Compartir igual-Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.** No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Índice

1. Datos de información	7
1.1. Presentación de la asignatura	7
1.2. Competencias genéricas de la UTPL.....	7
1.3. Competencias específicas de la carrera	7
1.4. Problemática que aborda la asignatura	7
2. Metodología de aprendizaje	8
3. Orientaciones didácticas por resultados de aprendizaje	9
 Primer bimestre.....	 9
 Resultado de aprendizaje 1.....	 9
Contenidos, recursos y actividades de aprendizaje.....	9
 Semana 1	 9
 Unidad 1. Arquitectura de aplicaciones empresariales	 11
1.1. Introducción a la arquitectura de aplicaciones.....	11
1.2. Fases del ciclo ADM	15
Actividad de aprendizaje recomendada.....	18
 Semana 2	 19
1.3. Revisión del metamodelo de arquitectura empresarial.....	19
1.4. Business capabilities – capacidades de negocio	21
1.5. Cadena de valor	23
1.6. Servicio de aplicaciones.....	27
Actividades de aprendizaje recomendadas	28
 Semana 3	 30
1.7. Arquitectura de aplicaciones vs. Arquitectura de datos	30
1.8. Model driven architecture	34
1.9. Inputs - recolección entregables de arquitectura	36
Actividades de aprendizaje recomendadas	37
Autoevaluación 1	40
 Semana 4	 43

Unidad 2. Arquitectura de aplicaciones empresariales	43
2.1. Gestión de Portafolio de Aplicaciones (APM).....	43
2.2. Gartner's TIME model	49
Actividades de aprendizaje recomendadas	55
Semana 5	56
2.3. Definición y diseño de arquitectura de aplicaciones base.....	56
2.4. Diseño de arquitectura empresarial	65
Actividades de aprendizaje recomendadas	67
Semana 6	68
2.5. Arquitectura de integración	68
2.6. Interfaces de comunicación.....	68
2.7. Arquitectura de datos.....	75
Actividades de aprendizaje recomendadas	76
Semana 7	77
2.8. Componentes lógicos y físicos de aplicación.....	77
Actividades de aprendizaje recomendadas	85
Autoevaluación 2.....	87
Semana 8	90
Actividades finales del bimestre	90
Segundo bimestre	92
Resultado de aprendizaje 2.....	92
Contenidos, recursos y actividades de aprendizaje.....	92
Semana 9	92
Unidad 3. Arquitectura objetivo.....	93
3.1. Arquitectura de objetivo o arquitectura TO-BE	93
3.2. Estrategia Cloud 6R	96
Actividad de aprendizaje recomendada.....	99
Semana 10	100

3.3. Gap análisis.....	100
3.4. Objetivos empresariales.....	103
3.5. Diseño de alto nivel de arquitectura	104
Actividad de aprendizaje recomendada.....	105
Semana 11	106
3.6. Riesgos e impactos (gestión de riesgos).....	106
3.7. Gestión de riesgos	107
3.8. Reducir complejidad de aplicaciones.....	112
Actividades de aprendizaje recomendadas	113
Autoevaluación 3	115
Semana 12	119
Unidad 4. Roadmap de arquitectura	119
4.1. Hoja de ruta o roadmap de componentes.....	119
Actividad de aprendizaje recomendada.....	122
Semana 13	124
4.2. Arquitecturas de transición.....	124
Actividades de aprendizaje recomendadas	127
Semana 14	128
4.3. Modelo de gobierno de arquitectura	128
Actividades de aprendizaje recomendadas	135
Semana 15	136
4.4. Revisión y presentación a los stakeholders.....	136
Actividades de aprendizaje recomendadas	139
Autoevaluación 4.....	141
Semana 16	144
4. Solucionario	146
5. Glosario.....	152
6. Referencias bibliográficas	153
7. Anexos	154



1. Datos de información

1.1. Presentación de la asignatura



1.2. Competencias genéricas de la UTPL

- Comunicación oral y escrita.

1.3. Competencias específicas de la carrera

- Desarrolla aplicaciones empresariales aplicando enfoques centrados en la nube.
- Diseña modelos arquitectónicos de empresa, para gestionar el alineamiento estratégico entre negocio y TI.

1.4. Problemática que aborda la asignatura

La presente asignatura permite abordar la siguiente problemática:

Concebir la Arquitectura Digital de una empresa a través del modelado, alineación y comprensión de las interacciones significativas entre el negocio y las TIC en la era digital. La formación base en este núcleo académico girará en torno a formar los estudiantes con habilidades fuertes en la construcción de aplicaciones nativas en nube, lo que les permitirá a los estudiantes construir y entregar continuamente software escalable, resiliente y robusto en entornos de nube automatizados para un negocio digital.



2. Metodología de aprendizaje

El proceso enseñanza-aprendizaje de la asignatura utiliza la **Metodología Basada en Casos (ABC)** por su amplia ocupación y orientación al aprendizaje autónomo. En esta metodología fomenta el cumplimiento de tareas de investigación que le permitan al estudiante dar solución a los problemas planteados en los casos, respecto a la arquitectura de aplicaciones de una empresa.

Para el proceso de **enseñanza aprendizaje**, se disponen diversos recursos educativos para que como estudiante adquiera el conocimiento teórico de base y complemente su aprendizaje mediante el desarrollo de las actividades, que proporcionan el contexto práctico, donde se refuerza la teoría y se desarrolla las habilidades necesarias para proponer arquitecturas de aplicaciones.

Los casos prácticos le permitirán ubicar la forma de actuar con información real de las empresas, considerando siempre la confidencialidad y gestión de la información con base en las leyes de protección de datos, la información de las empresas a nivel de arquitectura impacta a nivel estratégico y reputacional, por eso es importante su correcta gestión y tratamiento.

Toda la información brindada en este curso le servirá para desarrollar las actividades del Prácticum, teniendo en cuenta que las 2 materias están relacionadas y trabajan coordinadas para poder lograr las competencias adecuadas en su aprendizaje.



3. Orientaciones didácticas por resultados de aprendizaje



Primer bimestre

Resultado de aprendizaje 1

- Articula las necesidades empresariales y el valor de la arquitectura de aplicaciones .

El presente resultado de aprendizaje se enfoca en la arquitectura de las aplicaciones empresariales y su integración con otros componentes de la arquitectura empresarial orientada a cubrir las necesidades empresariales.

Contenidos, recursos y actividades de aprendizaje



Semana 1

Estimado estudiante en esta asignatura analizaremos los conceptos necesarios para comprender aspectos importantes de la Arquitectura de Aplicaciones, tomando como referencia el ciclo ADM de TOGAF, de la cual se dieron nociones en la asignatura de Arquitectura de Negocio y Arquitectura de Datos. Para ello, trabajaremos con la Fase C de ADM, denominada Arquitectura de Sistemas de Información, que tiene dos componentes: datos y aplicaciones, nosotros vamos a cubrir la arquitectura de aplicaciones, que son piezas clave dentro de la Arquitectura Empresarial.

A lo largo de las unidades de estudio comprenderemos la arquitectura de aplicaciones y cada uno de sus componentes, métodos y técnicas, y como gestionar un portafolio para agregar valor a la organización. Veremos cómo los componentes arquitectónicos se integran entre los distintos dominios de la Arquitectura Empresarial. Haremos énfasis especial en los modelos de referencia para la arquitectura, puesto que estos se convertirán a futuro

en los principales insumos del proyecto de desarrollo arquitectónico. Este proceso correctamente elaborado y diseñado nos dará las pautas iniciales para una adecuada Gobernanza de Aplicaciones.

Para mejorar la comprensión utilizaremos lecturas, vídeos y actividades interactivas que guiarán en el proceso de aprendizaje. De la misma manera, se usará ejemplos prácticos que gradualmente le ayudarán a adquirir el conocimiento base para poder identificar los componentes arquitectónicos de una empresa. Es recomendable que cree una bitácora que le asista en la organización de los conceptos aprendidos, la que le permita disponer de primera mano de los temas que generan dudas e inquietudes para compartirlos con el docente en las tutorías.

En particular, la materia de Arquitectura de aplicaciones en TOGAF incluye los siguientes temas:

- Descripción de la arquitectura de aplicaciones: esta sección se enfoca en la descripción de los componentes, estructuras y relaciones de la arquitectura de aplicaciones de la organización.
- Requisitos de arquitectura de aplicaciones: esta sección se enfoca en la identificación y definición de los requisitos de la arquitectura de aplicaciones, incluyendo los requisitos de los usuarios, los requisitos de los sistemas y los requisitos de negocio.
- Diseño de la arquitectura de aplicaciones: esta sección se enfoca en la creación del diseño de la arquitectura de aplicaciones, incluyendo la identificación de las opciones de diseño, la selección de tecnologías y la creación de modelos de referencia de arquitectura.
- Implementación de la arquitectura de aplicaciones: esta sección se enfoca en la implementación de la arquitectura de aplicaciones, incluyendo la gestión de proyectos, la selección de proveedores y la creación de planes de transición.

En el marco de referencia de TOGAF, la materia de arquitectura de aplicaciones es esencial para la articulación de las necesidades empresariales y el valor de la arquitectura de aplicaciones. En particular, la arquitectura de aplicaciones se enfoca en diseñar y construir sistemas de software que cumplan con los requisitos empresariales y que agreguen valor al negocio. Por lo tanto, los profesionales de arquitectura de

aplicaciones deben comprender las necesidades y objetivos empresariales para diseñar y desarrollar una arquitectura de aplicaciones que agregue valor a la organización.

La materia de Arquitectura de aplicaciones es una disciplina clave en el contexto de la arquitectura empresarial, **en esta semana nos enfocaremos en dar una introducción** al marco de referencia de TOGAF (The Open Group Architecture Framework), la materia de Arquitectura de aplicaciones se enfoca en la creación del diseño de la arquitectura de aplicaciones, la identificación y definición de los requisitos de la arquitectura de aplicaciones, la implementación de la arquitectura de aplicaciones, y la descripción de la arquitectura de aplicaciones de la organización. Los profesionales de arquitectura de aplicaciones en el contexto de TOGAF deben comprender las necesidades y objetivos empresariales para diseñar y desarrollar una arquitectura de aplicaciones que agregue valor a la organización, **esto lo iremos descubriendo semana a semana, ánimo y esta gran travesía y nos vemos a bordo.**

Unidad 1. Arquitectura de aplicaciones empresariales

1.1. Introducción a la arquitectura de aplicaciones

The Open Group Architecture Framework (TOGAF) es un marco para la arquitectura empresarial que proporciona un enfoque integral para diseñar, planificar, implementar y gobernar la arquitectura de las Tecnologías de la Información (TI) de una empresa. TOGAF incluye distintos componentes, como un conjunto de normas, un modelo de referencia y una metodología para desarrollar y utilizar artefactos de arquitectura. La parte de arquitectura de sistemas de TOGAF se centra en el diseño de alto nivel de los sistemas de TI de una organización, incluidos el *hardware*, el *software* y las redes que soportan dichos sistemas. Se ocupa de la estructura general de los sistemas de TI, así como de las relaciones entre los diferentes sistemas y las interfaces que se utilizan para conectarlos. La arquitectura de sistemas de información proporciona un marco para diseñar e implantar sistemas de TI que estén alineados con las metas y objetivos empresariales de una organización.

La arquitectura empresarial y la arquitectura de aplicaciones son dos campos distintos que a menudo se utilizan juntos para apoyar el diseño

y desarrollo de los sistemas de Tecnología de la Información (TI) de una organización. La arquitectura empresarial se centra en la estructura y organización de alto nivel de una empresa, incluidos sus procesos de negocio, productos y servicios, información y tecnología. Proporciona un plan para alinear las estrategias de negocio y de TI de una organización, y para identificar y abordar las lagunas y oportunidades de mejora.

La arquitectura de aplicaciones, por su parte, se centra en el diseño de los sistemas informáticos de una organización, incluidos el *hardware*, el *software* y las redes que soportan dichos sistemas, esto será cubierto en arquitectura tecnológica. Se ocupa de la estructura y organización general de los sistemas informáticos, así como de las relaciones entre los distintos sistemas y las interfaces que se utilizan para conectarlos.

Para unir la arquitectura de negocio y la arquitectura de aplicaciones, primero hay que comprender las metas y objetivos empresariales de la organización, así como el estado actual de sus sistemas informáticos. Esta información puede utilizarse para desarrollar una arquitectura empresarial de alto nivel que describa los componentes clave y las relaciones de la organización, así como los procesos empresariales, los productos y servicios, la información y la tecnología que dan soporte a esos componentes.

Una vez desarrollada la arquitectura de negocio, puede utilizarse para orientar el desarrollo de la arquitectura de aplicaciones. Esto suele implicar la identificación de los sistemas de TI específicos que se requieren para apoyar la arquitectura de negocio, y el diseño de la estructura general y la organización de esos sistemas para satisfacer las necesidades de la organización. La arquitectura de aplicaciones debe diseñarse para alinearse con la arquitectura de negocio y apoyar el funcionamiento eficiente y eficaz de los sistemas de TI de la organización.

En general, el objetivo de combinar la arquitectura de negocio y la arquitectura de aplicaciones es crear un entorno informático bien diseñado, integrado y escalable que respalde las metas y objetivos empresariales de la organización. Esto puede ayudar a mejorar el rendimiento y la eficiencia de la organización, y permitirle adaptarse y responder a los cambios en su entorno empresarial.

En resumen, la arquitectura de aplicaciones se refiere a la forma en que una aplicación está estructurada y organizada. Normalmente, incluye el diseño

de alto nivel de los distintos componentes de la aplicación, las relaciones entre esos componentes y las interfaces que se utilizan para interactuar con la aplicación y acceder a ella. La arquitectura de aplicaciones es un aspecto importante del desarrollo de *software*, ya que puede afectar al rendimiento, la escalabilidad y la capacidad de mantenimiento de una aplicación.

Durante el curso iremos profundizando en estos conceptos para aplicar el conocimiento y la forma de alinear los activos tecnológicos de una organización con el negocio.

Como referencia se recopila la información de la guía de Arquitectura de Datos, para reutilizar la información expuesta del *framework* de TOGAF, a su vez poder diferenciar los puntos relevantes dentro de la arquitectura de aplicaciones.

El ciclo ADM es el método de desarrollo de la arquitectura de TOGAF, permite gestionar los requerimientos de arquitectura empresarial a través de diez fases, las mismas que pueden ser iteradas hasta conseguir el resultado arquitectónico esperado. Las iteraciones en el ciclo ADM pueden hacerse para todas las etapas, únicamente entre fases o iterar solo una fase particular, esto garantiza la optimización del tiempo de desarrollo de la arquitectura y la implementación de los recursos empresariales. En la asignatura de Arquitectura de Negocios, se hizo énfasis en las fases correspondientes al Negocio, por lo que recordaremos todos los conceptos clave previamente planteados, para tomarlos como punto de partida para el desarrollo de una Arquitectura de Aplicaciones.

El ADM, o Método de Desarrollo de Arquitecturas, es un enfoque paso a paso para diseñar e implantar una arquitectura empresarial. Consta de las 10 fases siguientes:

1. **Preliminar:** se establecen las bases para el proyecto de arquitectura empresarial. Definiendo el alcance del proyecto y de los objetivos empresariales que se quieren alcanzar, se identifican los *stakeholders* o grupos de interés involucrados en el proyecto, junto con los distintos planes de ejecución.
2. **Visión de la arquitectura:** desarrollar una visión de alto nivel para la arquitectura empresarial, incluyendo las metas y objetivos del

negocio, el alcance de la arquitectura y las partes interesadas que estarán involucradas.

3. **Arquitectura de negocio:** definir los componentes clave de negocio de la empresa, incluyendo los procesos de negocio, productos y servicios, información y tecnología.
4. **Arquitecturas de sistemas de información:** identificar los sistemas de información que son necesarios para apoyar la arquitectura de negocio, y definir la estructura general y la organización de dichos sistemas.
5. **Arquitectura tecnológica:** especificar el *hardware*, el *software* y las redes que se necesitan para apoyar las arquitecturas de los sistemas de información.
6. **Oportunidades y soluciones:** identificar las oportunidades para mejorar la arquitectura de la empresa, y desarrollar soluciones para hacer frente a esas oportunidades.
7. **Planificación de la migración:** desarrollar un plan de transición del estado actual de la arquitectura empresarial al estado objetivo.
8. **Gobernanza de la implantación:** definir las estructuras de gobierno, las políticas y los procedimientos necesarios para gestionar y mantener la arquitectura empresarial a lo largo del tiempo.
9. **Gestión de cambios en la arquitectura:** establecer procesos para supervisar, evaluar y adaptar la arquitectura empresarial según sea necesario para satisfacer las necesidades cambiantes de la empresa.
10. **Gestión de requerimientos:** se definen los requerimientos específicos para la arquitectura empresarial, se analiza los requerimientos de negocio y los requerimientos técnicos para la arquitectura empresarial, definición de los requerimientos funcionales y no funcionales, priorización de los requerimientos en función de su importancia y valor para el negocio.

En general, el método ADM proporciona un enfoque estructurado y completo para diseñar e implementar una arquitectura empresarial, y es un componente clave de TOGAF.

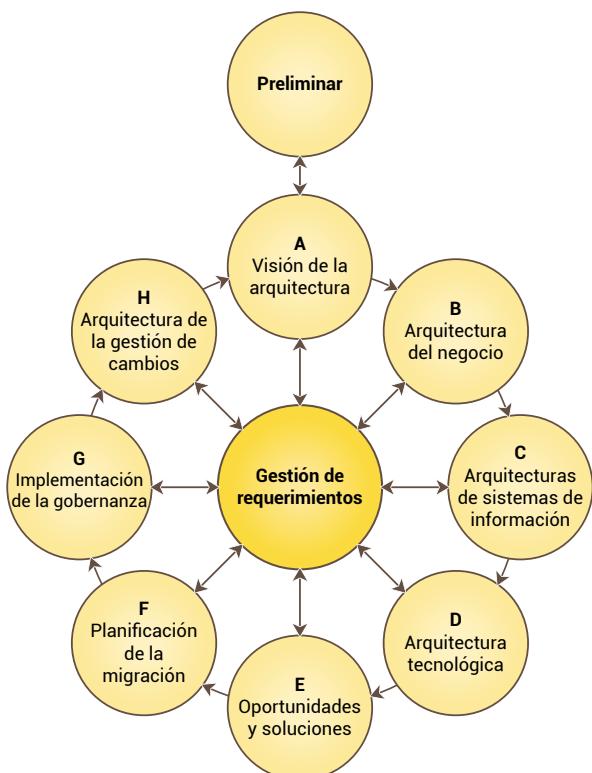
1.2. Fases del ciclo ADM

El ciclo ADM se basa en un principio de definición y desarrollo iterativo que permite desagregar la complejidad del desarrollo de la solución arquitectónica y asegurar que cada elemento preliminar sea considerado de manera apropiada (ver figura 1).

Los diez ciclos o fases se suceden el uno con el otro y pueden repetirse cuántas veces se requiera hasta que se obtenga el resultado esperado, iterando todo el ciclo entre fases o únicamente alguna de estas, de tal manera que se asegure que el resultado obtenido logre satisfacer la demanda de la empresa en términos de requisitos, herramientas, involucrados y productos esperados.

En el siguiente gráfico veremos cómo está distribuido el Ciclo de Desarrollo de la Arquitectura desde el enfoque ADM de TOGAF.

Figura 1.
Ciclo del ADM de TOGAF



Nota. Adaptado de TOGAF Standar Versión 9.2 (p. 40), por The Open Group. 2018.

1.2.1. Fase preliminar

La fase preliminar es una fase de evaluación de estado arquitectónico y se ejecuta una sola vez, puesto que en este punto se identifican niveles de madurez empresarial, además de poder valorar las capacidades empresariales de la organización desde un enfoque enteramente de transformación hacia un modelo arquitectónico. Se permite en este punto identificar y resaltar las herramientas, métodos, metodologías o *marcos de trabajo* que se pueden utilizar para identificar su integración o su futura puesta en marcha.



[Palabra clave:] Un marco de trabajo (o *framework* en inglés) es la aplicación o implementación de herramientas, técnicas, estándares, conceptos y buenas prácticas para la resolución de un problema particular.

La fase preliminar permite conocer cuál es el estado actual de la organización en términos de transformación arquitectónica y proyectarnos hacia una eficiente estrategia de implementación arquitectónica. A continuación, veremos los detalles principales de la Fase Preliminar, en donde destacaremos sus objetivos, entradas, salidas, actividades y enfoque ([ver tabla 1](#)).

1.2.2. Fase A: visión de la arquitectura

La visión arquitectónica es una de las fases más importantes dentro de la definición de la Arquitectura Empresarial, puesto que es en este punto en donde se identifican los límites y alcance de la arquitectura, así como sus involucrados y se establecen los lineamientos generales para conceptualizar el marco de la arquitectura objetivo.

La definición de visión arquitectónica centra su objetivo en el establecimiento de las directrices de la visión arquitectónica de alto nivel, a través de la selección e identificación de los procesos empresariales que serán intervenidos. (The Open Group, 2018).

Otro de los puntos focales de la fase de Visión de la Arquitectura, es la obtención de acuerdos que permitan definir quiénes serán los involucrados,

su nivel de involucramiento y su grado de participación en el trabajo arquitectónico. Se generará un documento denominado *Declaración de Trabajo de Arquitectura* y será aprobado por cada uno de los involucrados, estableciendo así el alcance del proyecto y los criterios de aceptación de este por parte de la empresa.



[Palabra clave:] El documento de Declaración de Trabajo de Arquitectura es el acuerdo o contrato firmado entre la empresa y el equipo de proveedores de la arquitectura e incluye parámetros de cumplimiento y calidad del proyecto arquitectónico a un nivel ejecutivo.

A continuación, revisaremos los detalles principales de la Fase A, en donde destacaremos sus objetivos, entradas, salidas, actividades y enfoque ([ver tabla 2](#)).

1.2.3. Fase B: arquitectura de negocio

En esta fase los esfuerzos se centran en diseñar y desarrollar la arquitectura objetivo partiendo de la base de definiciones del negocio desde sus objetivos, gestión y planificación estratégica.

En esta fase cobran especial relevancia la identificación y definición de los procesos y las herramientas de difusión e integración de la filosofía del negocio. Se identifica y evalúa el estado actual de la organización con el propósito de establecer un punto a través de los grados y niveles de madurez empresarial, así como el establecimiento de herramientas para el desarrollo y cumplimiento de los objetivos arquitectónicos.

A continuación, vamos a analizar los detalles centrales de la Fase B, para conocer sus objetivos, entradas, salidas, actividades y su enfoque ([ver tabla 3](#)).



Actividad de aprendizaje recomendada

Cada pequeño paso que das te acerca un poco más a tus objetivos. Así que, ¿por qué no dar un paso adelante esta semana y hacer algo que te acerque más a tus metas?

La actividad de aprendizaje recomendada es **leer el capítulo 10 de TOGAF** (*The Open Group Architecture Framework*) que se centra en los sistemas de información. TOGAF es un marco de arquitectura empresarial ampliamente utilizado en todo el mundo para ayudar a las empresas a diseñar, planificar y gestionar sus sistemas de información.

El capítulo 10 del estándar de TOGAF - ***Phase C: Information Systems Architectures – Application Architecture***, se puede encontrar en la biblioteca virtual en el siguiente enlace: [TOGAF – Sistemas de Información](#).

El capítulo 10 se enfoca específicamente en los sistemas de información y cómo se pueden integrar en una arquitectura empresarial coherente y efectiva. Al leer este capítulo, el estudiante obtendrá una comprensión profunda de cómo los sistemas de información pueden ser diseñados y gestionados para cumplir con los objetivos empresariales y brindar valor a la organización.

Para llevar a cabo esta actividad, el estudiante:

- Deberá dedicar un tiempo específico a leer el capítulo 10 de TOGAF de manera cuidadosa y concentrada.
- Es posible que también deba tomar notas y hacer resúmenes para asegurarse de que entiende completamente los conceptos clave.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

- Además, podría ser útil discutir con sus compañeros de clase o con el tutor cualquier pregunta o inquietud que pueda surgir durante la lectura del capítulo.



1.3. Revisión del metamodelo de arquitectura empresarial

El Metamodelo TOGAF es un modelo conceptual que proporciona un lenguaje y un marco común para describir la arquitectura de una empresa. Forma parte del TOGAF que es una metodología ampliamente utilizada para la planificación y el diseño de la arquitectura empresarial.

El Metamodelo TOGAF consta de cuatro capas: la capa de negocio, la capa de datos, la capa de aplicaciones y la capa de tecnología. Estas capas representan los diferentes componentes y elementos de una arquitectura empresarial, y están organizadas de forma jerárquica, con la capa de negocio en la parte superior y la capa de tecnología en la parte inferior.

La capa de negocio del Metamodelo TOGAF (Revisar módulo didáctico) representa las metas y objetivos estratégicos de la empresa, así como los procesos y funciones de negocio que dan soporte a dichas metas. La capa de datos representa los datos y la información que la empresa utiliza y produce, incluyendo las estructuras de datos, formatos y relaciones.

La capa de aplicación representa las aplicaciones y sistemas de software que soportan los procesos de negocio y utilizan los datos, incluyendo tanto las aplicaciones personalizadas como las comerciales (COTS – Commercial Off-the-shelf). La capa tecnológica representa el hardware, la infraestructura y los estándares técnicos que dan soporte a las aplicaciones, como servidores, redes y sistemas operativos.

En general, el Metamodelo TOGAF proporciona una forma estructurada y sistemática de describir y comprender los distintos componentes de una arquitectura empresarial, y es una herramienta valiosa para los arquitectos empresariales a la hora de planificar y diseñar su arquitectura.

Le invito a revisar la siguiente infografía sobre el Metamodelo TOGAF.

[TOGAF Metamodelo.](#)

La capa de aplicación del Metamodelo TOGAF representa las aplicaciones y sistemas de software que soportan los procesos de negocio y utilizan los

datos de una empresa. Incluye aplicaciones tanto personalizadas como comerciales (COTS).

Ejemplos de componentes que podrían incluirse en la capa de aplicación del Metamodelo TOGAF incluyen:

- Sistemas de Planificación de Recursos Empresariales (ERP).
- Sistemas de Gestión de las Relaciones con los Clientes (CRM).
- Sistemas de Gestión de la Cadena de Suministro (SCM).
- Herramientas de Inteligencia Empresarial (BI) y análisis de datos.
- Herramientas de colaboración y comunicación.
- Sistemas de Historia Clínica Electrónica (HCE).
- Sistemas de Gestión de Recursos Humanos (RR. HH.).
- Software de gestión de proyectos.
- Sistemas de Gestión de Contenidos (CMS).
- Plataformas de comercio electrónico.

Estos son solo algunos ejemplos de los tipos de componentes que podrían incluirse en la capa de aplicación del Metamodelo TOGAF. Los componentes específicos variarán en función de las necesidades y requisitos de la empresa.

Algunos ejemplos adicionales de componentes que podrían incluirse en la capa de aplicación del Metamodelo TOGAF incluyen:

- Sistemas de gestión financiera, como software de contabilidad.
- Sistemas de planificación de la fabricación y la producción.
- Sistemas de gestión de inventarios.
- Sistemas de Punto de Venta (TPV).
- Sistemas de Gestión del Aprendizaje (LMS).
- SIG (Sistemas de Información Geográfica).
- Sistemas de gestión de documentos.
- Sistemas de firma electrónica.
- Herramientas de gestión de redes sociales.
- Sistemas de control de presencia (Raymond, 2014).

Como puede ver, hay muchos tipos diferentes de componentes que podrían incluirse en la capa de aplicación del Metamodelo TOGAF. Los componentes específicos dependerán de las necesidades y requisitos de la empresa. Es tarea del arquitecto empresarial identificar los componentes apropiados y diseñar una capa de aplicación que satisfaga las necesidades de la empresa.

Las aplicaciones comerciales disponibles en el mercado (COTS) son aplicaciones de software pre-empaquetadas que se desarrollan y venden a múltiples clientes. Están diseñadas para ser configurables y personalizables a fin de satisfacer las necesidades específicas de distintas organizaciones, pero no se desarrollan específicamente para un solo cliente.

Las aplicaciones COTS suelen ser desarrolladas por Proveedores de Software Independientes (ISV) y vendidas a través de diversos canales, como mercados en línea o revendedores. Suelen ser menos caras que el software desarrollado a medida, y pueden implantarse y desplegarse más rápidamente, ya que no se construyen desde cero.

Algunos ejemplos de aplicaciones COTS son los sistemas de Planificación de Recursos Empresariales (ERP), los Sistemas de Gestión de Relaciones con los Clientes (CRM) y los Sistemas de Gestión de la Cadena de Suministro (SCM). Este tipo de aplicaciones son utilizadas habitualmente por las organizaciones para dar soporte a sus procesos y operaciones empresariales.

Aunque las aplicaciones COTS pueden aportar muchas ventajas, también tienen algunas limitaciones. Es posible que no ofrezcan toda la funcionalidad que necesita una organización concreta y que no puedan personalizarse tanto como una aplicación desarrollada a medida. Además, las organizaciones que utilizan aplicaciones COTS pueden tener que depender del ISV para el soporte y las actualizaciones, lo que puede ser un punto potencial de vulnerabilidad.

En general, las aplicaciones COTS son aplicaciones de software pre-empaquetadas que se venden a múltiples clientes. Pueden ofrecer muchas ventajas, como el ahorro de costes y un despliegue más rápido, pero también pueden tener algunas limitaciones.

1.4. Business capabilities – capacidades de negocio

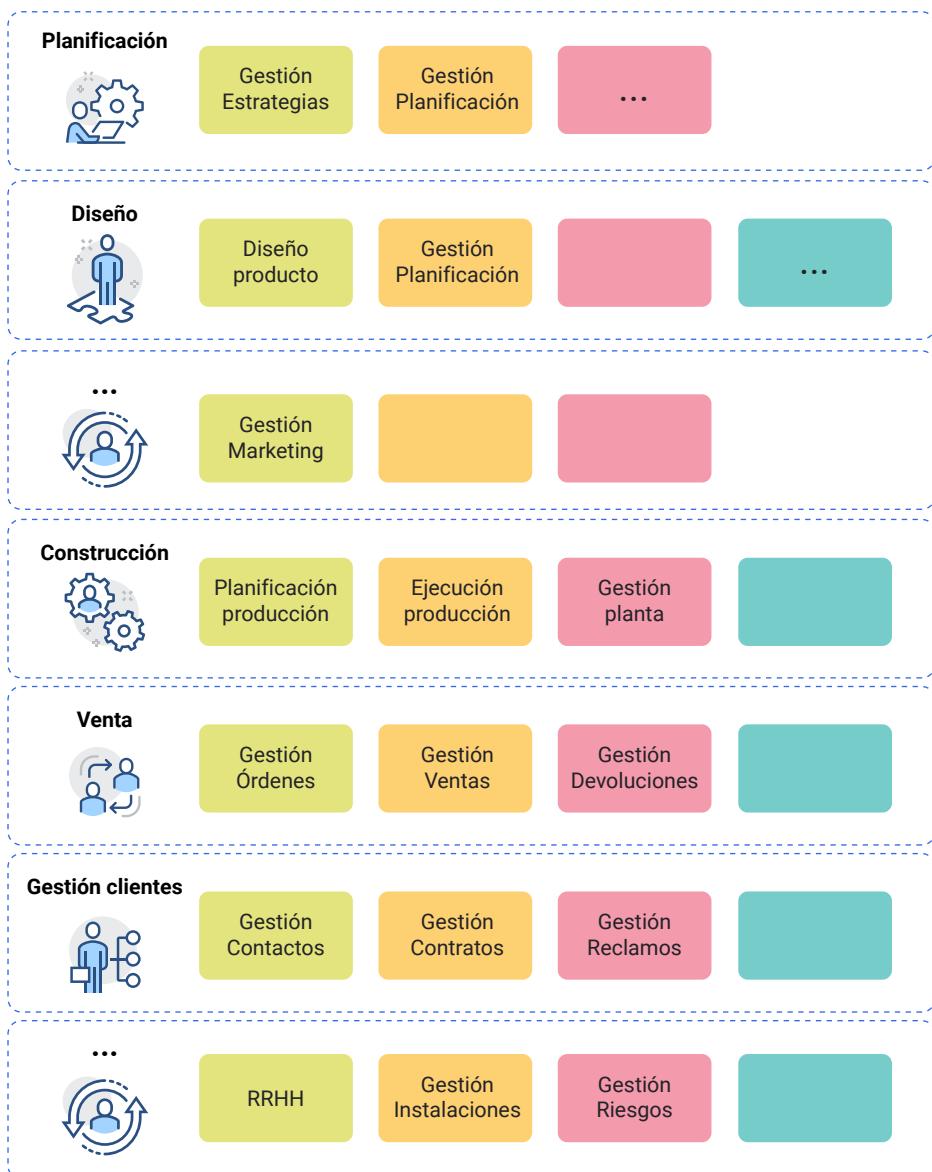
Las capacidades de negocio, son las funciones y actividades de alto nivel que realiza una empresa para alcanzar sus metas y objetivos estratégicos. Representan las competencias básicas de una empresa y permiten organizar y comprender las capacidades necesarias para que una empresa funcione eficazmente.

Las capacidades empresariales se representan a menudo como una jerarquía, con las capacidades de más alto nivel en la parte superior y capacidades más específicas en los niveles inferiores. Por ejemplo, una capacidad empresarial de alto nivel podría ser “captación de clientes”, mientras que las capacidades de nivel inferior podrían incluir “generación de clientes potenciales”, “marketing” y “ventas” . (Ver figura 2).

Las capacidades empresariales pueden considerarse los “bloques de construcción” de una empresa, ya que permiten descomponer las complejas actividades de una empresa en componentes más manejables y comprensibles. Al comprender y trazar el mapa de las capacidades empresariales de una organización, es posible identificar áreas de fortaleza y debilidad, y desarrollar estrategias para mejorar y potenciar esas capacidades.

En general, las capacidades empresariales son una herramienta valiosa para comprender y gestionar las operaciones de una empresa. Permiten desglosar las complejas actividades de una empresa en componentes más manejables y comprensibles, y pueden ayudar a las organizaciones a identificar áreas de mejora y crecimiento.

Figura 2.
Mapa de capacidades de negocio



Nota. Adaptado de Business Capabilities - Version 2 (p. 42), por The Open Group, 2022, OpenGroup.org

1.5. Cadena de valor

La cadena de valor de capacidades de negocio es un modelo que se utiliza para identificar y gestionar las actividades clave que impulsan el

éxito de una empresa. Estas actividades se dividen en dos categorías: las actividades primarias y las actividades de soporte (ver figura 3).

Las actividades primarias son las que crean valor directamente para los clientes y generan ingresos para la empresa. Estas actividades incluyen:

1. **Adquisición de clientes**: todas las actividades que realiza la empresa para atraer a nuevos clientes, como la publicidad, el *marketing* y las promociones.
2. **Desarrollo de productos**: todas las actividades relacionadas con la creación de nuevos productos y servicios, desde la investigación y el desarrollo hasta la producción y el lanzamiento al mercado.
3. **Gestión de operaciones**: todas las actividades relacionadas con la gestión y el mantenimiento de los procesos operativos de la empresa, como la gestión de la cadena de suministro, la producción, la logística y el soporte posventa .
4. **Fidelización de clientes**: todas las actividades que realiza la empresa para mantener la satisfacción y la fidelidad de los clientes, como el servicio al cliente, la gestión de quejas y la oferta de programas de lealtad.

Las actividades de soporte son aquellas que proporcionan el soporte necesario para las actividades primarias y permiten que la empresa opere de manera efectiva. Estas actividades incluyen:

1. **Gestión de recursos humanos**: todas las actividades relacionadas con la gestión de los empleados, como la selección, el entrenamiento, la compensación y el desarrollo.
2. **Gestión de tecnología de la información**: todas las actividades relacionadas con la gestión de los sistemas de información y tecnología de la empresa, como el desarrollo de software, la gestión de redes y la seguridad de la información.
3. **Gestión de infraestructura**: todas las actividades relacionadas con la gestión de los activos físicos de la empresa, como los edificios, las instalaciones y los equipos.

4. **Gestión de finanzas:** todas las actividades relacionadas con la gestión financiera de la empresa, como la contabilidad, la planificación financiera y la gestión de riesgos.

Figura 3.

Cadena de Valor Empresarial



Nota. Adaptado de Cadena de Valor Empresarial (p. 100), por Michael Porter, 1985.

Permite identificar los procesos o actividades primarias y de soporte de una empresa, para la generación de valor hacia los clientes en la fabricación de un producto o la prestación de un servicio.

Es importante mencionar la amplitud de arquitectura en el contexto de la arquitectura empresarial. La amplitud de arquitectura se refiere al alcance de la arquitectura empresarial y a los componentes que deben incluirse en ella para abarcar toda la empresa.

En este sentido, la amplitud de arquitectura se refiere a la capacidad de la arquitectura empresarial para proporcionar una visión completa y detallada de la organización, que incluye tanto los aspectos tecnológicos como los aspectos de negocio y de gestión. Esto significa que la arquitectura empresarial debe cubrir todos los aspectos relevantes de la organización, desde la estrategia y los procesos de negocio hasta la infraestructura tecnológica y los sistemas de información.

La amplitud de arquitectura es un aspecto clave de la arquitectura empresarial, ya que permite a las organizaciones tener una visión completa y coherente de su estructura y operaciones, lo que a su vez les permite tomar decisiones más informadas y eficaces. Además, la amplitud de arquitectura es esencial para asegurar que los cambios en la organización

se realicen de manera coherente y eficiente, lo que ayuda a minimizar el riesgo y los costos asociados con los cambios importantes en la empresa.

Por ejemplo, la amplitud de una arquitectura de software puede referirse al número de aplicaciones, servicios o sistemas diferentes que se incluyen en la arquitectura. Una arquitectura más amplia puede incluir más aplicaciones, servicios y sistemas, mientras que una arquitectura más limitada puede incluir menos.

La amplitud de una arquitectura puede ser un factor importante a tener en cuenta, ya que puede afectar a la complejidad y escalabilidad de la arquitectura. Una arquitectura más amplia puede ser más compleja y difícil de gestionar, pero también puede proporcionar más flexibilidad y capacidades. Una arquitectura más estrecha puede ser más sencilla y fácil de gestionar, pero también puede tener capacidades más limitadas.

En general, la amplitud de una arquitectura se refiere al número de componentes o elementos diferentes que se incluyen en la arquitectura. Es un factor importante a tener en cuenta a la hora de diseñar una arquitectura, ya que puede influir en la complejidad y las capacidades de la misma.

La complejidad de una arquitectura de aplicaciones empresariales se refiere a la cantidad y variedad de componentes y sistemas que intervienen en la arquitectura, así como a la complejidad de sus interconexiones y dependencias. Una arquitectura de aplicaciones empresariales puede ser más compleja si incluye una gran cantidad de aplicaciones y sistemas diferentes, si estos sistemas están integrados de manera compleja entre sí, o si la arquitectura incluye elementos como reglas de negocio complejas o procesos de integración de datos complejos.

La complejidad de una arquitectura de aplicaciones puede tener un impacto significativo en la forma en que se desarrollan y mantienen las aplicaciones, así como en la eficiencia y escalabilidad de la arquitectura en su conjunto. Por lo general, es importante tratar de minimizar la complejidad de la arquitectura de aplicaciones en la medida de lo posible para facilitar el desarrollo y el mantenimiento de las aplicaciones y mejorar la eficiencia del sistema en su conjunto.

Otro concepto clave dentro de la arquitectura de aplicaciones son los servicios de aplicación.

1.6. Servicio de aplicaciones

Un servicio de aplicación es un componente de software que proporciona una función empresarial o técnica específica a otras aplicaciones o sistemas. Suele estar diseñado para ser reutilizable y modular, y otras aplicaciones pueden acceder a él y utilizarlo a través de una interfaz estandarizada.

Los servicios de aplicación se utilizan a menudo para implementar funcionalidades comunes o compartidas que son necesarias para múltiples aplicaciones o sistemas.

Por ejemplo, un servicio de autenticación puede servir para que las aplicaciones autentiquen a los usuarios, mientras que un servicio de procesamiento de pagos puede servir para que las aplicaciones procesen los pagos.

Los servicios de aplicación suelen diseñarse para ser independientes y estar poco acoplados, de modo que puedan integrarse fácilmente con otras aplicaciones y sistemas. Esto permite utilizarlos en diferentes contextos y entornos, y los hace más flexibles y escalables.

En general, un servicio de aplicación es un componente de software que proporciona una función específica a otras aplicaciones o sistemas. Está diseñado para ser reutilizable y modular, y otras aplicaciones pueden acceder a él y utilizarlo a través de una interfaz estandarizada.

Caso de estudio - telecomunicaciones:

Un caso de estudio para el diseño de un metamodelo basado en TOGAF para la arquitectura de aplicaciones podría ser el de una empresa de telecomunicaciones que busca estandarizar su arquitectura de aplicaciones en todas las áreas de la empresa.

La empresa tiene diferentes áreas de negocio, como telefonía fija, telefonía móvil, Internet y televisión, y cada una de ellas tiene su propio conjunto de aplicaciones y sistemas. Además, la empresa ha crecido a través de adquisiciones y fusiones, lo que ha llevado a la adopción de diferentes tecnologías y enfoques de arquitectura.

Para abordar este desafío, se podría utilizar un enfoque basado en TOGAF para diseñar un metamodelo que defina los componentes y relaciones clave de la arquitectura de aplicaciones de la empresa. Este metamodelo proporcionaría una estructura común para la arquitectura de aplicaciones en todas las áreas de la empresa, permitiendo una mayor integración y coherencia en toda la organización.

Algunos de los componentes clave que podrían incluirse en este metamodelo son:

- Capas de la arquitectura de aplicaciones, como la capa de presentación, la capa de lógica de negocio y la capa de datos.
- Tipos de componentes, como aplicaciones, servicios, interfaces y bases de datos.
- Relaciones entre componentes, como dependencias, asociaciones y Acuerdos de Nivel de Servicio (SLA).
- Reglas y políticas para la arquitectura de aplicaciones, como estándares de codificación, políticas de seguridad y criterios de selección de tecnología.

Este metamodelo basado en TOGAF podría ayudar a la empresa de telecomunicaciones a estandarizar su enfoque de arquitectura de aplicaciones y proporcionar una base sólida para futuras iniciativas de transformación digital. Además, podría permitir una mayor colaboración y alineación entre las diferentes áreas de la empresa, lo que a su vez podría mejorar la eficiencia y reducir los costos.



Actividades de aprendizaje recomendadas

El éxito no es fácil, pero tampoco es imposible. Con perseverancia y determinación, puedes lograr cualquier cosa que se proponga.

Le invito a realizar las siguientes actividades de aprendizaje recomendadas:

- A. Leer el Metamodelo de TOGAF y, analice sus relaciones y la información que podrían gestionar en cada una de las entidades, para esta actividad puede revisar el metamodelo detallado.

[TOGAF - Metamodelo.](#)

- B. Después de la revisión del Metamodelo de TOGAF, y para afianzar los conocimientos, le planteo contestar las siguientes preguntas:
1. ¿Qué es el metamodelo de TOGAF y cuál es su propósito principal? **Respuesta:** el metamodelo de TOGAF es un modelo de datos que describe los componentes clave de la arquitectura empresarial, incluyendo los elementos de negocio, tecnología e información. Su propósito principal es proporcionar una estructura unificada y coherente para describir la arquitectura empresarial, y facilitar la comunicación y colaboración entre los diferentes grupos de interés.
 2. ¿Cuáles son los principales elementos del metamodelo de TOGAF y cuál es su función?

Respuesta: los principales elementos del metamodelo de TOGAF incluyen los componentes de arquitectura, las vistas de arquitectura y los artefactos de arquitectura. Los componentes de arquitectura son los bloques de construcción fundamentales de la arquitectura empresarial, mientras que las vistas de arquitectura son las perspectivas específicas que se utilizan para describir la arquitectura desde diferentes ángulos. Los artefactos de arquitectura son los documentos y modelos que se utilizan para describir los componentes de arquitectura y las vistas de arquitectura.

3. ¿Cómo se puede utilizar el metamodelo de TOGAF en la práctica para desarrollar una arquitectura empresarial efectiva?

Respuesta: el metamodelo de TOGAF se puede utilizar en la práctica para desarrollar una arquitectura empresarial efectiva mediante la identificación de los componentes de arquitectura relevantes, la definición de las vistas de arquitectura necesarias para describir la arquitectura empresarial desde diferentes perspectivas, y la creación de los artefactos de arquitectura necesarios para documentar y comunicar la arquitectura a los diferentes grupos de interés. Además, el metamodelo de TOGAF puede ser utilizado para guiar el proceso de desarrollo de la arquitectura empresarial, asegurando que todos los aspectos relevantes de la organización sean considerados y que se sigan las mejores prácticas para el desarrollo de la arquitectura.

Preguntas adicionales:

5. ¿Cuáles son las principales categorías de elementos del metamodelo de TOGAF, y cuál es el propósito de cada una de ellas?
6. ¿Cómo se relacionan los diferentes elementos del metamodelo de TOGAF entre sí, y cómo se pueden utilizar estas relaciones para describir la arquitectura de una organización?

¿Cómo se puede personalizar el metamodelo de TOGAF para adaptarlo a las necesidades específicas de una organización o industria en particular, y cuáles son las mejores prácticas para hacerlo de manera efectiva ?

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.



Semana 3

1.7. Arquitectura de aplicaciones vs. Arquitectura de datos

La arquitectura de datos y la arquitectura de aplicaciones son dos campos distintos, pero estrechamente relacionados en el desarrollo de software. La arquitectura de datos se ocupa del diseño y la organización de los datos de un sistema, mientras que la arquitectura de aplicaciones se centra en el diseño y la estructura de las aplicaciones de software que interactúan con esos datos. Para conectar eficazmente estas dos áreas, es importante asegurarse de que la arquitectura de datos se diseña de forma que satisfaga las necesidades de las aplicaciones que se crearán sobre ella. Para ello, hay que tener en cuenta los requisitos de las aplicaciones durante el proceso de diseño de la arquitectura de datos y asegurarse de que esta proporciona las estructuras de datos, las interfaces y los mecanismos de acceso necesarios para que las aplicaciones funcionen correctamente. Además, es importante establecer una comunicación y colaboración claras entre los arquitectos de datos y los arquitectos de aplicaciones a lo largo de todo el proceso de desarrollo para garantizar que la arquitectura de datos y la arquitectura de aplicaciones estén correctamente alineadas.

La implantación de una arquitectura de aplicaciones en una empresa suele implicar los siguientes pasos:

1. Identificar las metas y objetivos empresariales de la organización, así como el estado actual de sus sistemas informáticos. Esto ayudará a establecer una línea de base para la arquitectura de aplicaciones y a identificar cualquier laguna o área de mejora.
2. Desarrollar una arquitectura de aplicaciones detallada que especifique el *hardware*, el *software* y las redes necesarios para respaldar las metas y objetivos empresariales de la organización. Esto debe incluir la estructura general y la organización de los sistemas de TI, así como las relaciones entre los diferentes sistemas y las interfaces que se utilizan para conectarlos.
3. Desarrollar un plan completo para implantar la arquitectura de aplicaciones, que incluya plazos, presupuestos, recursos e Indicadores Clave de Rendimiento (KPI). Este plan debe incluir estrategias para integrar las nuevas aplicaciones con los sistemas existentes, migrar datos y procesos, y garantizar la seguridad y fiabilidad de la arquitectura de aplicaciones.
4. Identificar y priorizar las aplicaciones específicas que se necesitan para apoyar las metas y objetivos empresariales de la organización, y que se ajustan a los criterios establecidos en el paso 2. Esto puede incluir tanto aplicaciones desarrolladas a medida como aplicaciones de *software libre*. Esto puede incluir tanto aplicaciones desarrolladas a medida como aplicaciones comerciales disponibles en el mercado (COTS).
5. Implementar la arquitectura de aplicaciones de acuerdo con el plan desarrollado en el paso 3, incluyendo el desarrollo, prueba, despliegue y mantenimiento continuo de las aplicaciones específicas identificadas en el paso 4.
6. Supervisar y evaluar el rendimiento de la arquitectura de aplicaciones a lo largo del tiempo y realizar los ajustes o modificaciones necesarios para garantizar que sigue respaldando las metas y objetivos empresariales de la organización.

La implantación de una arquitectura de aplicaciones en una empresa requiere un plan bien diseñado y exhaustivo, así como una coordinación y ejecución cuidadosas para garantizar que los sistemas informáticos de la organización estén alineados con sus metas y objetivos empresariales.

Esto puede ayudar a mejorar el rendimiento, la eficiencia y la agilidad de la organización, y permitirle adaptarse y responder a los cambios en su entorno empresarial.

Las entradas para la fase de arquitectura de aplicaciones de un proyecto de desarrollo de software variarán en función de las necesidades y requisitos específicos del proyecto. Sin embargo, algunas de las entradas comunes para esta fase podrían incluir:

- **Requisitos empresariales:** esto puede incluir documentos o artefactos que describen las metas y objetivos del proyecto, así como los requisitos funcionales y no funcionales específicos que la arquitectura tendrá que soportar.
- **Sistema y arquitectura actuales:** puede incluir información sobre los sistemas y tecnologías existentes que se verán afectados por el proyecto, así como la arquitectura actual de la organización. Esto ayudará al arquitecto a comprender el contexto y las limitaciones del proyecto.
- **Hoja de ruta tecnológica:** puede incluir información sobre las tendencias tecnológicas y las innovaciones que la organización está interesada en seguir, así como cualquier inversión tecnológica que la organización haya planeado. Esto ayudará al arquitecto a comprender la visión y dirección a largo plazo de la organización.
- **Entrevistas con las partes interesadas:** esto puede implicar la realización de entrevistas con las principales partes interesadas, como usuarios empresariales, expertos técnicos y gestores de proyectos, para recabar sus opiniones y perspectivas sobre el proyecto. Esto ayudará al arquitecto a comprender las necesidades y expectativas de las partes interesadas y a garantizar que la arquitectura se ajusta a sus metas y objetivos.

En general, las aportaciones a la fase de arquitectura de aplicaciones de un proyecto de desarrollo de software variarán en función de las necesidades y requisitos específicos del proyecto. Sin embargo, algunos datos comunes pueden ser los requisitos empresariales, el sistema y la arquitectura actuales, la hoja de ruta tecnológica y las entrevistas con las partes interesadas.

Hay que tener claro por qué definir unos principios de arquitectura.

1.7.1. Principios de arquitectura

Los principios de arquitectura son un conjunto de directrices y buenas prácticas que ayudan a los arquitectos a diseñar y desarrollar sistemas de software de forma coherente y eficaz. Estos principios proporcionan un marco para tomar decisiones de diseño y pueden ayudar a garantizar que un sistema de software esté bien estructurado y sea mantenable y escalable. Algunos principios de arquitectura comunes son la modularidad, la separación de intereses y el acoplamiento flexible. Estos principios pueden ayudar a los arquitectos a crear sistemas fáciles de entender, modificar y ampliar con el tiempo.

Aquí le presento algunos principios de arquitectura como ejemplo, usted debería definir los que aplican mejor a su empresa:

1. **Modularidad:** los componentes de la arquitectura empresarial deben ser diseñados para ser independientes y fácilmente reutilizables, para aumentar la flexibilidad y la capacidad de adaptación de la organización.
2. **Estándares abiertos:** los componentes de la arquitectura empresarial deben estar basados en estándares abiertos, para aumentar la interoperabilidad y facilitar la integración con otros sistemas y aplicaciones.
3. **Agilidad:** la arquitectura empresarial debe ser diseñada para ser flexible y adaptable, para permitir cambios rápidos y eficientes en respuesta a las necesidades cambiantes del negocio.
4. **Orientación al servicio:** la arquitectura empresarial debe estar diseñada en torno a los servicios que la organización proporciona, en lugar de centrarse en los sistemas y tecnologías subyacentes.
5. **Seguridad:** la arquitectura empresarial debe ser diseñada para garantizar la seguridad y protección de la información y los activos de la organización.
6. **Sencillez:** la arquitectura empresarial debe ser diseñada de manera sencilla y clara, para asegurar que sea fácil de entender y utilizar para todos los grupos de interés.

7. **Interoperabilidad:** los componentes de la arquitectura empresarial deben ser diseñados para interoperar de manera efectiva con otros sistemas y aplicaciones, para facilitar la integración y la colaboración en toda la organización.
8. **Optimización de recursos:** la arquitectura empresarial debe ser diseñada para optimizar el uso de los recursos de la organización, incluyendo el tiempo, el dinero y la energía.

La arquitectura empresarial se basa en la modularidad y para esto se han creado los bloques de arquitectura.

1.7.2. Bloques de arquitectura

Los bloques de arquitectura reutilizables, también conocidos como componentes arquitectónicos o *building blocks*, son elementos predefinidos y probados previamente que pueden utilizarse como base para crear nuevos sistemas o aplicaciones de software. Estos bloques suelen estar diseñados para ser modulares y flexibles, lo que permite combinarlos y personalizarlos fácilmente para satisfacer las necesidades específicas de distintos proyectos. Los bloques de arquitectura reutilizables pueden ayudar a acelerar el proceso de desarrollo y reducir el coste global de la creación de sistemas de software, ya que proporcionan un conjunto de componentes bien definidos, probados y contrastados que pueden utilizarse como base para nuevos sistemas. Algunos ejemplos de bloques de arquitectura reutilizables son los componentes de interfaz de usuario, los componentes de almacenamiento y acceso a datos y los componentes de seguridad y autenticación.

Los modelos son una pieza importante en las arquitecturas, su origen es el siguiente.

1.8. Model driven architecture

La OMG (Object Management Group) es un consorcio internacional sin fines de lucro que se dedica a la estandarización de la arquitectura y tecnología de software. Fue fundada en 1989 y cuenta con miembros de diversas industrias, incluyendo tecnología, servicios financieros, gobierno y salud.

La OMG se centra en la estandarización de los lenguajes de modelado, las metodologías de desarrollo de software y las arquitecturas de aplicaciones. Sus estándares incluyen UML (Unified Modeling Language) para modelado de sistemas, BPMN (Business Process Model and Notation) para modelado de procesos empresariales, y SysML (Systems Modeling Language) para modelado de sistemas de ingeniería. También promueve la arquitectura de aplicaciones empresariales (SOA, por sus siglas en inglés) y el modelado basado en componentes (MDA, por sus siglas en inglés).

La OMG también trabaja en estandarizar la interconexión de diferentes sistemas, mediante el uso de protocolos y tecnologías como CORBA (Common Object Request Broker Architecture), DDS (Data Distribution Service) y UAF (Unified Architecture Framework) (Group O. M., s.f.).

La Arquitectura Dirigida por Modelos (MDA) es un enfoque de desarrollo de software basado en el uso de modelos como medio principal para especificar, diseñar e implementar sistemas de software. En MDA, los modelos se utilizan para capturar los aspectos esenciales de un sistema de forma abstracta, independiente de la plataforma y fácil de entender. A continuación, estos modelos se transforman y traducen a diferentes formas, como código ejecutable o artefactos de despliegue, para crear una implementación específica del sistema.

El enfoque OMG MDA se basa en tres principios fundamentales: independencia de la plataforma, separación de preocupaciones y automatización. Independencia de la plataforma significa que los modelos utilizados en MDA no están vinculados a ningún lenguaje de programación, sistema operativo o plataforma de hardware específicos. Esto permite utilizar los mismos modelos para generar diferentes implementaciones de un sistema para distintas plataformas. Separación de intereses significa que los distintos aspectos de un sistema, como su funcionalidad, comportamiento y estructura, se modelan y gestionan por separado en MDA. Esto ayuda a reducir la complejidad y mejorar la capacidad de mantenimiento del sistema. Automatización significa que la transformación y traducción de los modelos a diferentes formas se realiza automáticamente mediante herramientas, en lugar de hacerlo manualmente los desarrolladores.

El enfoque OMG MDA pretende proporcionar una forma estandarizada y coherente de desarrollar sistemas de software que sea independiente de las tecnologías y plataformas específicas utilizadas. Se basa en el

uso de lenguajes de modelado y transformación bien definidos, así como de herramientas y técnicas para gestionar el proceso de desarrollo.

El objetivo de MDA es permitir que los desarrolladores de software se centren en los aspectos esenciales de un sistema, como su funcionalidad y comportamiento, en lugar de estancarse en los detalles de una implementación específica.

Los documentos de entrada de la etapa C (Diseño de la Arquitectura) de TOGAF son aquellos que se utilizan para guiar el proceso de diseño de la arquitectura empresarial. Estos documentos se utilizan como base para la creación de los artefactos de arquitectura que describen la arquitectura empresarial y sus componentes. Algunos de los documentos de entrada comunes para la etapa C incluyen:

1.9. *Inputs - recolección entregables de arquitectura*

Documentos de entrada – etapa C

- Modelo organizativo para la arquitectura empresarial incluyendo:
 - Alcance de las organizaciones afectadas.
 - Evaluación de la madurez, lagunas y enfoque de resolución.
 - Funciones y responsabilidades del equipo o equipos de arquitectura.
 - Restricciones en el trabajo de arquitectura.
 - Requisitos presupuestarios.
 - Gobierno y estrategia de apoyo.
- Marco de arquitectura a medida (véase Parte IV, Sección 32.2.21), incluyendo:
 - Método de arquitectura a medida.
 - Contenido de la arquitectura adaptada (entregables y artefactos).
 - Herramientas configuradas y desplegadas.
- Principios de aplicación (ver Parte III, Sección 20.6.3), si existen.
- Declaración de trabajo de arquitectura (ver Parte IV, Sección 32.2.20).
- Visión de la arquitectura (ver Parte IV, Sección 32.2.8).
- Repositorio de arquitectura (ver Parte IV, Sección 32.2.5), incluyendo:

- Bloques de construcción reutilizables.
 - Modelos de referencia disponibles públicamente.
 - Modelos de referencia específicos de la organización.
 - Estándares de la organización.
- Borrador del documento de definición de la arquitectura (ver Parte IV, sección 32.2.3), incluyendo:
- Arquitectura empresarial de referencia, versión 1.0 (detallada), si procede.
 - Arquitectura de negocio objetivo, versión 1.0 (detallada).
 - Arquitectura de datos de referencia, versión 1.0 (detallada), o versión 0.1 (visión).
 - Arquitectura de datos de destino, versión 1.0 (detallada), o versión 0.1 (visión).
 - Arquitectura de aplicaciones de referencia, versión 0.1, si procede y está disponible.
 - Arquitectura de la aplicación de destino, versión 0.1, si está disponible.
 - Arquitectura tecnológica de referencia, versión 0.1 (visión).
 - Arquitectura tecnológica de destino, versión 0.1 (visión).
- Borrador de especificación de requisitos de arquitectura (ver Parte IV, sección 32.2.6), incluyendo:
- Resultados del análisis de carencias (de la Arquitectura Empresarial y la Arquitectura de Datos, si están disponibles).
 - Requisitos técnicos relevantes que se aplicarán a esta fase.
- Componentes de Arquitectura de Negocio y de Datos de una Hoja de Ruta de Arquitectura, si están disponibles (The Open Group, 2018).

Ahora le invito a desarrollar las siguientes actividades recomendadas.



Actividades de aprendizaje recomendadas

- A. Realizar los documentos de entrada de TOGAF del módulo 10, que se centra en los sistemas de información. Los documentos de entrada

son una parte importante del proceso de arquitectura empresarial, ya que proporcionan la información necesaria para planificar, diseñar y gestionar los sistemas de información de una organización.

Para llevar a cabo esta actividad, el estudiante deberá seguir los siguientes pasos:

1. **Identificar los documentos de entrada requeridos** para el módulo 10 de TOGAF. Estos pueden incluir documentos como políticas de seguridad, requisitos del negocio, diagramas de flujo de datos, entre otros.
2. **Recopilar los documentos de entrada necesarios.** El estudiante deberá obtener acceso a los documentos relevantes de la organización o solicitarlos a los responsables correspondientes.
3. **Revisar y analizar los documentos de entrada.** El estudiante deberá leer cuidadosamente cada documento y analizar su contenido para identificar los requisitos clave, las restricciones, las dependencias y otros factores importantes que puedan afectar el diseño y la gestión de los sistemas de información.
4. **Elaborar los documentos de entrada.** Utilizando la información recopilada y analizada, el estudiante deberá elaborar los documentos de entrada requeridos para el módulo 10 de TOGAF. Esto puede incluir la creación de diagramas de flujo, descripciones de procesos, listas de requisitos, entre otros.
5. **Revisar y validar los documentos de entrada.** Una vez que se hayan elaborado los documentos de entrada, es importante revisarlos cuidadosamente y validar que contengan la información correcta y sean coherentes entre sí.

En general, esta actividad requiere de una dedicación importante de tiempo y esfuerzo, ya que los documentos de entrada son una parte crítica del proceso de arquitectura empresarial y pueden afectar significativamente el éxito de los sistemas de información de una organización.

- B. Con base a los documentos recomendados por TOGAF como entrada para la definición de la arquitectura de aplicaciones, levante los siguientes documentos:

1. Visión de la arquitectura.
2. Componentes de arquitectura de Negocio y Datos.
3. Modelo organizativo.
4. Especificación de requisitos de arquitectura.
5. Declaración de trabajo de arquitectura.

Esta información la puede encontrar en el siguiente enlace: [TOGAF – Sistemas de Información](#).

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

C. Estimado estudiante, hemos culminado la unidad 1 en la cual se ha realizado una introducción a la arquitectura de aplicaciones desde la visión de arquitectura empresarial, para poder consolidar los conocimientos, repase el contenido con la siguiente autoevaluación.



Autoevaluación 1

1. ¿Cuál de las siguientes opciones describe mejor el propósito del metamodelo de TOGAF?
 - a. Proporcionar un lenguaje común y una estructura para el desarrollo de la arquitectura empresarial.
 - b. Definir los principios y objetivos de la arquitectura empresarial.
 - c. Proporcionar una guía paso a paso para la implementación de la arquitectura empresarial.
2. ¿Qué son los principios de arquitectura en TOGAF?
 - a. Directrices fundamentales que orientan y limitan el diseño de la arquitectura empresarial.
 - b. Elementos clave de la arquitectura empresarial que deben ser implementados de inmediato.
 - c. Documentos detallados que describen todos los aspectos de la arquitectura empresarial.
3. ¿Qué son las capacidades de negocio en TOGAF?
 - a. Elementos de la arquitectura empresarial que representan las funciones y procesos críticos de la organización.
 - b. Documentos que describen los requisitos de negocio y los objetivos de la organización.
 - c. Estructuras de la arquitectura empresarial que describen las relaciones entre los componentes del sistema.
4. ¿Qué son los *building blocks* en TOGAF?
 - a. Componentes de la arquitectura empresarial que se pueden reutilizar para construir soluciones de arquitectura.
 - b. Modelos detallados de los procesos de negocio de la organización.
 - c. Documentos que describen los riesgos y oportunidades asociados con la implementación de la arquitectura empresarial.

5. ¿Cuál es uno de los insumos clave para la fase de arquitectura de aplicaciones en TOGAF?
 - a. El modelo de cadena de valor de la organización.
 - b. La visión y estrategia empresarial.
 - c. El análisis de requerimientos de negocio.
6. ¿Qué son los modelos de arquitectura en TOGAF?
 - a. Descripciones detalladas de los componentes de la arquitectura empresarial y cómo se relacionan entre sí.
 - b. Estructuras de la arquitectura empresarial que describen los procesos de negocio críticos de la organización.
 - c. Documentos que describen los riesgos y oportunidades asociados con la implementación de la arquitectura empresarial.
7. ¿Qué es la cadena de valor en TOGAF?
 - a. Un modelo de la estructura organizativa de la empresa.
 - b. Una herramienta de análisis de riesgos empresariales.
 - c. Un modelo que describe cómo la empresa crea valor para sus clientes.
8. ¿Cuál es el propósito de los insumos de la fase de arquitectura de aplicaciones en TOGAF?
 - a. Proporcionar información sobre los requerimientos de negocio y los objetivos de la organización.
 - b. Ayudar a definir la estructura de la arquitectura empresarial y a identificar las soluciones de arquitectura necesarias.
 - c. Proporcionar un plan detallado para la implementación de la arquitectura empresarial.
9. ¿Cuál es el propósito de la fase de arquitectura de aplicaciones en TOGAF?
 - a. Definir la estructura de la arquitectura empresarial y establecer los objetivos y requisitos de la organización.
 - b. Identificar y seleccionar soluciones de arquitectura que satisfagan los requerimientos de negocio.
 - c. Diseñar e implementar soluciones de arquitectura que satisfagan los requerimientos de negocio.

10. ¿Cuál es el beneficio principal de utilizar TOGAF para el desarrollo de la arquitectura empresarial?
- a. Proporcionar un lenguaje común y una estructura para el desarrollo de la arquitectura empresarial.
 - b. Asegurar que los requerimientos de negocio se reflejen adecuadamente en la arquitectura empresarial.
 - c. Reducir el tiempo y los costos asociados con la implementación de soluciones de arquitectura.



Nota: se pueden encontrar las respuestas a estas preguntas en la sección de solucionario, en la parte de autoevaluación 1.

[Ir a solucionario](#)



Unidad 2. Arquitectura de aplicaciones empresariales

2.1. Gestión de Portafolio de Aplicaciones (APM)

Es una metodología para gestionar la obsolescencia de las aplicaciones, se conoce como “Gestión de la Cartera de Aplicaciones” (APM). Esta metodología consiste en identificar y clasificar las aplicaciones de la cartera de una organización y, a continuación, utilizar esa información para tomar decisiones sobre qué aplicaciones conservar, retirar o sustituir.

Para aplicar el enfoque de Gestión de la Cartera de Aplicaciones (APM) en una empresa, puede seguir estos pasos:

1. **Identifique las aplicaciones de la cartera:** empiece por catalogar todas las aplicaciones que utiliza actualmente la empresa, incluidas las aplicaciones desarrolladas a medida y las comerciales disponibles en el mercado (COTS). Esto proporcionará una visión completa de las aplicaciones que se utilizan actualmente.
2. **Clasificar las aplicaciones:** a continuación, clasifique las aplicaciones según varios criterios, como su valor empresarial, complejidad técnica y nivel de obsolescencia. Esto le ayudará a priorizar las aplicaciones e identificar cuáles son las más críticas para la empresa.
3. **Evalúe las aplicaciones:** realice una evaluación detallada de las aplicaciones para determinar su estado actual y cualquier riesgo o problema potencial. Esto puede implicar la realización de entrevistas con las partes interesadas, la revisión de la documentación o la realización de evaluaciones técnicas.
4. **Elaborar un plan:** basándose en la evaluación, desarrolle un plan para gestionar las aplicaciones de la cartera. Esto puede implicar la retirada o sustitución de algunas aplicaciones, mientras se continúa dando soporte y mantenimiento a otras. Asegúrese de incluir en el plan plazos, presupuestos y recursos específicos.

5. **Ejecutar el plan:** ponga en marcha el plan, proporcionando los recursos y el apoyo necesarios para garantizar su éxito. Esto puede implicar la formación del personal, la actualización de sistemas y procesos y la implantación de nuevas herramientas y tecnologías.

Siguiendo estos pasos, puede aplicar el enfoque APM en su empresa para gestionar eficazmente la cartera de aplicaciones y mitigar los riesgos asociados a la obsolescencia de las aplicaciones. Esto le ayudará a garantizar que las aplicaciones de su empresa están alineadas con sus metas y objetivos empresariales, y que son capaces de respaldar las operaciones y el crecimiento de la empresa.

El diseño de una cartera de aplicaciones implica varios pasos, entre los que se incluyen los siguientes:

1. Identificar las metas y objetivos empresariales de la organización, así como el estado actual de sus sistemas informáticos. Esto ayudará a establecer una línea de base para la cartera de aplicaciones y a identificar cualquier carencia o área de mejora.
2. Realizar un inventario exhaustivo de las aplicaciones existentes en la organización, que incluya la finalidad, la funcionalidad y el uso de cada aplicación. Esto proporcionará una visión completa del panorama actual de aplicaciones de la organización.
3. Evaluar la cartera de aplicaciones actual para determinar qué aplicaciones siguen siendo relevantes y útiles, y cuáles pueden ser redundantes, obsoletas o infroutilizadas. Esto puede ayudar a identificar oportunidades de consolidación, racionalización o retirada de aplicaciones.
4. Desarrollar un conjunto de criterios para evaluar y priorizar las nuevas aplicaciones, basándose en factores como la alineación con las metas y objetivos empresariales, la integración con los sistemas existentes, el coste y los beneficios potenciales.
5. Identificar y priorizar las nuevas aplicaciones necesarias para respaldar las metas y objetivos empresariales de la organización, y que se ajusten a los criterios establecidos en el paso 4. Esto puede incluir tanto aplicaciones desarrolladas a medida como aplicaciones de software libre. Esto puede incluir tanto aplicaciones desarrolladas

a medida como aplicaciones comerciales disponibles en el mercado (COTS).

6. Desarrollar un plan completo para implantar y gestionar la cartera de aplicaciones, que incluya plazos, presupuestos, recursos e indicadores clave de rendimiento (KPI). Este plan debe incluir estrategias para integrar las nuevas aplicaciones con los sistemas existentes, migrar datos y procesos, y garantizar la seguridad y fiabilidad de la cartera de aplicaciones.

En general, el objetivo de diseñar una cartera de aplicaciones es crear un conjunto bien diseñado e integrado de aplicaciones que respalte las metas y objetivos empresariales de la organización, y que pueda gestionarse y mantenerse eficazmente a lo largo del tiempo. Esto puede ayudar a mejorar el rendimiento, la eficiencia y la agilidad de la organización, y permitirle adaptarse y responder a los cambios en su entorno empresarial.

Un portafolio de aplicaciones es una colección de aplicaciones de software que son utilizadas por una organización para apoyar sus operaciones y procesos de negocio. Para crear una cartera de aplicaciones, se pueden seguir los siguientes pasos:

1. **Identificar los objetivos y necesidades empresariales de la organización.** Esto puede implicar la realización de un análisis exhaustivo de los procesos, actividades y requisitos empresariales, con el fin de comprender las funciones y capacidades clave que deben soportar las aplicaciones de la cartera.
2. **Identificar las aplicaciones actuales que utiliza la organización.** Esto puede implicar la realización de un inventario de las aplicaciones existentes, incluidas sus funciones, capacidades y dependencias.
3. **Evaluar las aplicaciones existentes para determinar su alineación con los objetivos y necesidades de la empresa.** Esto puede implicar evaluar las capacidades y el rendimiento de las aplicaciones existentes, y determinar si cumplen los requisitos de la organización.
4. **Desarrollar un plan para la cartera de aplicaciones.** Puede consistir en identificar las carencias o deficiencias de las aplicaciones existentes y elaborar una hoja de ruta para subsanarlas. El plan también debe

incluir estrategias para mantener y actualizar las aplicaciones de la cartera a lo largo del tiempo.

5. **Implementar el plan para la cartera de aplicaciones.** Esto puede implicar la implementación de nuevas aplicaciones o la realización de cambios en las aplicaciones existentes, así como el establecimiento de procesos y procedimientos para gestionar y mantener la cartera.
6. **Supervisar y revisar la cartera de aplicaciones de forma continua.** Esto puede implicar la evaluación periódica del rendimiento y la eficacia de las aplicaciones de la cartera y la realización de los ajustes necesarios para garantizar que sigan respaldando los objetivos y las necesidades empresariales de la organización.

La gestión del ciclo de vida de una aplicación implica la gestión de la aplicación a lo largo de todo su ciclo de vida, desde su desarrollo y despliegue iniciales hasta su mantenimiento, evolución y eventual retirada. Esto incluye normalmente los siguientes pasos.

1. **Desarrollo:** consiste en diseñar y desarrollar la aplicación de acuerdo con los requisitos y especificaciones definidos. Puede incluir actividades como codificación, pruebas, depuración y documentación de la aplicación.
2. **Despliegue:** consiste en desplegar la aplicación en un entorno de producción, donde pueda ser utilizada por los usuarios de la organización. Esto puede incluir actividades como la instalación, configuración e integración de la aplicación con otros sistemas y aplicaciones.
3. **Mantenimiento:** consiste en mantener la aplicación a lo largo del tiempo para garantizar su funcionamiento y fiabilidad. Puede incluir actividades como la supervisión de la aplicación, la identificación y corrección de errores o defectos, y la aplicación de actualizaciones y parches.
4. **Evolución:** se trata de hacer evolucionar la aplicación a lo largo del tiempo, para mantenerla alineada con las necesidades y requisitos cambiantes de la organización. Esto puede incluir actividades como añadir nuevas características y capacidades a la aplicación, modificar

o sustituir componentes existentes e integrar la aplicación con nuevos sistemas y tecnologías.

5. **Retirada:** se trata de retirar la aplicación cuando ya no es necesaria o cuando ya no es factible mantenerla. Esto puede incluir actividades como el desmantelamiento de la aplicación, la migración de datos y procesos a otros sistemas y la eliminación del *hardware* y el *software* de la aplicación.

En general, la gestión del ciclo de vida de una aplicación implica la gestión de la aplicación a lo largo de todo su ciclo de vida, con el fin de garantizar su funcionamiento continuo y su alineación con las metas y objetivos empresariales de la organización. Esto puede ayudar a garantizar la eficacia y eficiencia de la aplicación, y permitir a la organización adaptarse y responder a los cambios en su entorno empresarial.

El portafolio de aplicaciones empresariales (también conocida como arquitectura de aplicaciones en plural) es una disciplina que suele crearse al principio como parte de una práctica de arquitectura empresarial. A veces, la arquitectura del portafolio de aplicaciones se combina con la disciplina de TPM (Trusted Platform Module). Tal vez esto ocurrió como resultado de la confusión sobre la distinción entre una aplicación y una tecnología, o tal vez ocurrió debido a los vínculos potencialmente fuertes que las tecnologías y las aplicaciones pueden tener entre sí.

Para recapitular brevemente la distinción entre una tecnología y una aplicación, una tecnología no contiene reglas de negocio que soporten una capacidad de negocio, mientras que una aplicación sí. Hay numerosos productos de software que son tecnologías, como motores de reglas, hojas de cálculo y herramientas de desarrollo (por ejemplo, MS Access) que clasificamos simplemente como tecnologías. Sin embargo, una vez que las reglas de negocio se colocan dentro de cualquier instancia de dicha tecnología, entonces esa instancia específica de un motor de reglas, hoja de cálculo o archivo MS Access es una aplicación, que debe ser mantenida por un equipo de desarrollo de aplicaciones.

Así qué para aclarar, una vez que una hoja de cálculo contiene fórmulas complejas que se utilizan para apoyar una capacidad de negocio, esa instancia de esa hoja de cálculo debe pasar formalmente a convertirse en una aplicación.

Como aplicación, debe probarse, su código fuente, debe controlarse en un repositorio de producción, debe hacerse una copia de seguridad con fines de recuperación, etcétera. Sin embargo, si la hoja de cálculo es simplemente un documento o informe, de manera similar a cualquier documento de procesamiento de textos como un archivo de MS Word o Google Docs , que no contiene reglas de negocio, entonces esas instancias son simplemente “documentos electrónicos” que no deben clasificarse como una aplicación. Como mucho, dependiendo del papel que desempeñen, deberían clasificarse como informes de producción.

La arquitectura de la cartera de aplicaciones tiene una perspectiva de Gestión de la Cartera de Aplicaciones (APM) “macro” y “micro”, y ambas son importantes. La más común de las dos es la perspectiva “micro”, que implica el mantenimiento de un inventario de aplicaciones con toda su información relevante. En la perspectiva “micro”, cada aplicación es independiente. En un estado más maduro, se vincula a la cartera tecnológica para ilustrar de qué tecnologías depende una aplicación determinada.

La perspectiva menos común es la vista “macro” de las aplicaciones. En este punto de vista, las aplicaciones se asocian a las capacidades empresariales que soportan en su estado actual, así como a una serie de alternativas entre las que se puede elegir un estado futuro. En su estado más maduro, el estado actual de la cartera de aplicaciones y alternativas por capacidad de negocio se ha evaluado para su análisis de riesgo relativo de coste-beneficio a lo largo del tiempo.

Como muchos han observado en las empresas, los costes de las aplicaciones tienden a aumentar con el tiempo a medida que el mantenimiento del software se hace más complejo, y las licencias de los paquetes más antiguos y sus tecnologías de apoyo aumentan correspondientemente de coste a medida que disminuye la base de clientes del proveedor.

A menos que una aplicación pueda retirarse finalmente como parte de su fin de vida normal, las opciones son mantener el creciente coste de mantenimiento, o sustituir la aplicación, o potencialmente consolidar la aplicación con otras que proporcionen un coste de vida útil más bajo.

Cuando se gestiona adecuadamente, esta disciplina arquitectónica proporciona a cualquier empresa una forma de proyectar con precisión los costes de automatización de una determinada línea de negocio para facilitar

la planificación avanzada de la dirección empresarial y la elaboración de presupuestos. (Luisi, 2014).

2.2. Gartner's TIME model

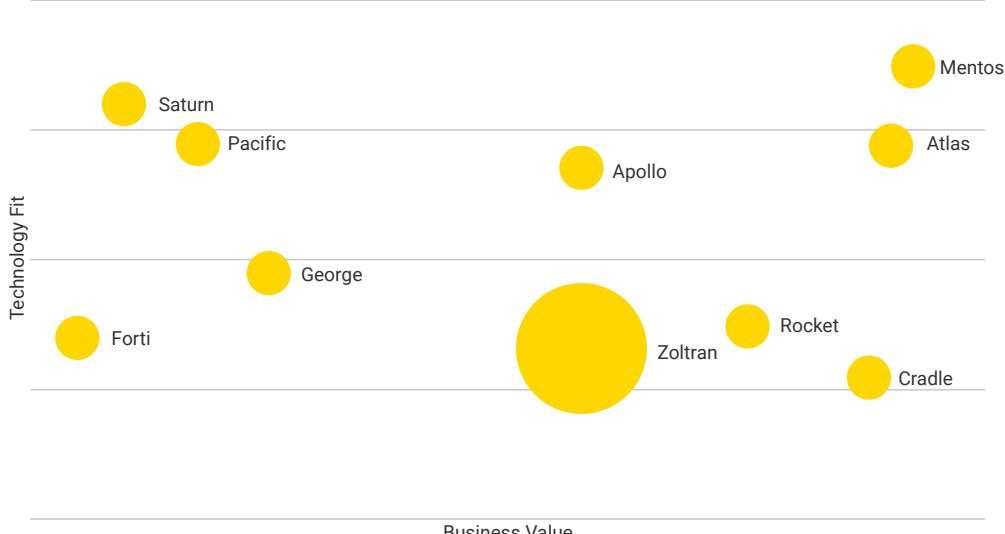
Gartner ha llevado a cabo una investigación estructurada en este ámbito para ayudar a evaluar la cartera de aplicaciones de una organización. Esto ayuda a comprender el estado de referencia de la cartera de aplicaciones y a identificar las aplicaciones candidatas para reducir los residuos y optimizar la cartera. Se trata de una práctica estándar que puede tener distintos nombres en su organización, como modernización de aplicaciones, estandarización de aplicaciones, racionalización de aplicaciones y simplificación de la cartera. Son bastante similares, pero tienen objetivos y resultados ligeramente diferentes.

Idealmente, una organización debería tener una estrategia de gestión y gobernanza para controlar la afluencia de nuevas aplicaciones. En un departamento de TI sobrecargado de trabajo, es difícil controlar la introducción de nuevas aplicaciones. Debido a la falta de recursos informáticos cualificados y a la ausencia de una gobernanza reforzada, las unidades de negocio empiezan rápidamente a buscar en otra parte. Con muchas aplicaciones disponibles bajo el modelo SaaS, es bastante fácil para una unidad de negocio adquirir una nueva aplicación y añadirla a la cartera de aplicaciones de la organización. Esto empeora cuando se añaden a la cartera aplicaciones que realizan funciones iguales o similares (redundantes).

Un arquitecto empresarial, actuando como gestor de la cartera de aplicaciones, necesita examinar lo que tiene, cuál es la magnitud del problema y cuál es el camino a seguir. La gestión de la cartera de aplicaciones mide el valor empresarial de una aplicación en función de su adecuación tecnológica. Al cuantificar (puntuar) el valor empresarial y el ajuste tecnológico, una aplicación puede medirse y evaluarse en comparación con otras aplicaciones. La comparación puede verse trazando las puntuaciones de las aplicaciones en un gráfico X-Y, como se muestra a continuación (ver figura 4).

Figura 4.

Mapa de Aplicaciones clasificado en base al ajuste funcionalidad



Nota. Tomado de Application Portfolio Management using TIME [Ilustración], por Bhardwaj, M., 2018, [Linkedin](#). CC BY 2.0

A continuación le invito a revisar la gráfica TIME y un caso de estudio de una empresa financiera.

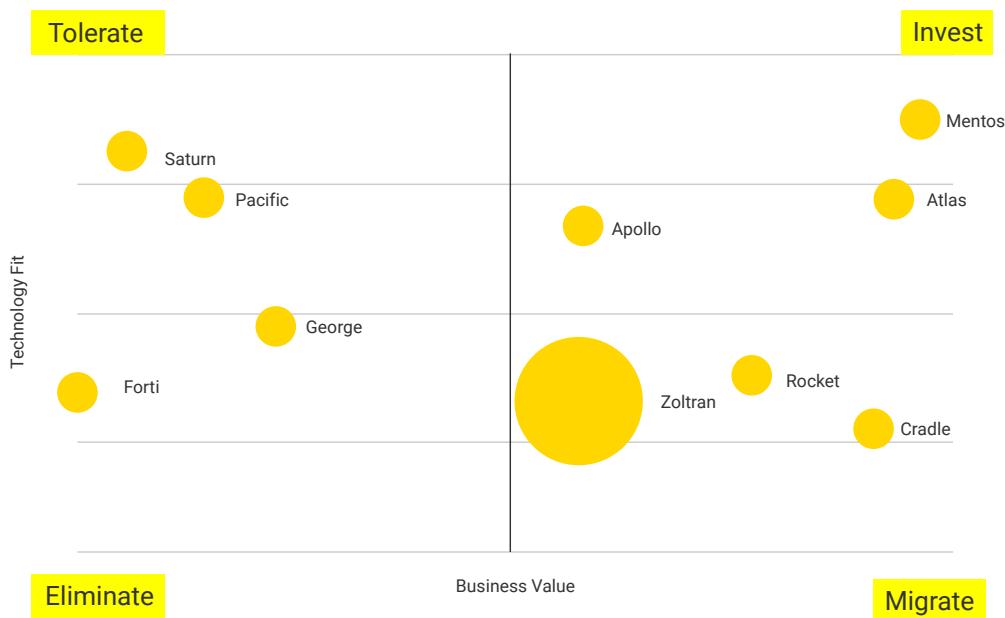
2.2.1. Gráfica TIME

A continuación, el gráfico se divide en cuatro categorías (cuadrantes), los cuadrantes están etiquetados como “TIME”, donde las siglas equivalen a:

- T = Tolerar.
- I = Invertir.
- M = Migrar.
- E = Eliminar.

El gráfico se representa de la siguiente manera (ver figura 5):

Figura 5.
Reporte TIME del Portafolio de Aplicaciones



Nota. Tomado de Application Portfolio Management using TIME [Ilustración], por Bhardwaj, M., 2018, [Linkedin](#). CC BY 2.0

Las aplicaciones del cuadrante **Tolerar** (Tolerate) aportan cierto valor empresarial y se ajustan bien a la tecnología de la organización. Tienen bajo riesgo y buenos niveles de servicio. No son aplicaciones críticas para el negocio, y su buen encaje dentro de la organización las convierte en candidatas a ser toleradas. Idealmente, las aplicaciones de esta categoría son candidatas a ser eliminadas, pero sus interfaces con otras aplicaciones críticas pueden hacerlas tolerables por el momento. Algunas de estas aplicaciones pueden estar en plataformas heredadas o basadas en arquitecturas antiguas, pero siguen siendo lo “suficientemente buenas” como para ser toleradas por TI.

Las aplicaciones del cuadrante **Invertir** (Invest) aportan un buen valor empresarial y se adaptan bien a la tecnología. Las aplicaciones de este cuadrante son las “mejores de su clase”. Debe invertirse más esfuerzo en conservar y reutilizar estas aplicaciones de formas diferentes e innovadoras (como reutilizarlas en procesos empresariales existentes y nuevos) para extraer más beneficios de ellas.

El tercer cuadrante, **Migrar** (Migrate), contiene aplicaciones que aportan un buen valor empresarial, pero que no encajan bien en la tecnología de la organización. Son fundamentales para el funcionamiento de la empresa. Pero no se adaptan bien a la tecnología, ya que pueden ejecutarse en una plataforma de *hardware* y *software* no compatible. Otra posibilidad es que no se adapten bien a la tecnología por falta de personal con las capacidades de desarrollo y mantenimiento adecuadas. Puede haber otros problemas, como información redundante o inexacta, o que su mantenimiento e integración con otras aplicaciones resulte difícil (caro). Sean cuales sean los retos, estas aplicaciones deben migrar a una plataforma adecuada. Algunas aplicaciones críticas para el negocio, basadas en texto y de pantalla verde, suelen entrar en este cuadrante.

En el cuarto cuadrante, **Eliminar** (Eliminate), se sitúan las aplicaciones que tienen cierto valor empresarial y no se adaptan bien a la tecnología. Dado que no son críticas para el negocio y se adaptan mal a la tecnología, lo mejor es eliminarlas. Las aplicaciones o fuentes de información redundantes, junto con las plataformas de *hardware* y *software* no compatibles y la falta de recursos humanos cualificados, hacen que estas aplicaciones sean caras de mantener y operar.

La gestión de la cartera de aplicaciones es una tarea importante en la práctica de la arquitectura empresarial. Gartner proporciona un enfoque estructurado para evaluar su panorama de aplicaciones basado en su enfoque del cuadrante TIME. Proporciona una visión clara en forma de un gráfico de burbujas familiar en el que las aplicaciones se pueden clasificar en función del TIEMPO. Esto ayuda a un gestor de cartera a evaluar qué aplicaciones son candidatas a ser eliminadas y migradas y cuáles deben promoverse activamente para una mayor reutilización. Aunque es subjetivo asignar valores numéricos a atributos de “calidad de servicio” como la fiabilidad y la seguridad, es necesario hacer un esfuerzo de suposición para asignarlos a cada aplicación, de modo que cada una de ellas pueda medirse en función de una escala definida. La creación de gráficos TIME no es el objetivo, sino un medio para alcanzarlo: el objetivo es apoyar a las empresas con necesidades cambiantes, ser ágiles y proporcionar innovación de la manera más rentable. (Bhardwaj, 2018).

Caso de estudio – empresa financiera

Para este caso de estudio, se presentará la arquitectura de una aplicación para la gestión de un portafolio de aplicaciones de una empresa financiera utilizando la metodología de TOGAF.

1. **Descripción del problema:** la empresa financiera XYZ cuenta con un gran número de aplicaciones que son utilizadas para llevar a cabo sus operaciones diarias. Debido a la falta de una gestión centralizada del portafolio de aplicaciones, la empresa tiene dificultades para evaluar y controlar el uso de sus aplicaciones, lo que resulta en costos innecesarios y una baja eficiencia en la gestión de sus operaciones.
2. **Objetivos de la solución:** la empresa financiera XYZ busca implementar una solución que permita una gestión centralizada del portafolio de aplicaciones. La solución debe proporcionar una visión completa de todas las aplicaciones en uso, sus dependencias y relaciones, y permitir a la empresa tomar decisiones informadas sobre cómo optimizar su uso.
3. **Enfoque de la solución:** se propone la implementación de una arquitectura de aplicación que se base en la metodología de TOGAF. La arquitectura constará de las siguientes capas:
 - **Capa de negocio:** esta capa contendrá los procesos de negocio de la empresa y los servicios que ofrecen. Esta capa estará vinculada a la capa de aplicaciones y proporcionará la base para la gestión de aplicaciones.
 - **Capa de aplicación:** esta capa contendrá las aplicaciones utilizadas por la empresa financiera. Cada aplicación será evaluada y catalogada según su importancia y valor para la empresa. Se crearán perfiles de aplicaciones y se identificarán sus dependencias y relaciones con otras aplicaciones y servicios.
 - **Capa de tecnología:** esta capa contendrá la infraestructura tecnológica utilizada por la empresa financiera. Se evaluarán las necesidades de la infraestructura y se asegurará que las aplicaciones estén adecuadamente alojadas y disponibles para su uso.
 - **Capa de datos:** esta capa contendrá los datos necesarios para el funcionamiento de las aplicaciones y procesos de negocio. Se

desarrollará una estrategia de gestión de datos y se establecerán políticas para el acceso y seguridad de los datos.

4. **Implementación:** la arquitectura de la aplicación será implementada utilizando herramientas y tecnologías de la industria financiera, como software de gestión de aplicaciones, software de gestión de bases de datos, soluciones de virtualización y herramientas de gestión de proyectos. También se llevará a cabo una capacitación para los empleados de la empresa financiera para asegurarse de que puedan utilizar eficazmente la nueva solución.
5. **Resultados esperados:** la implementación de la solución de gestión de portafolio de aplicaciones permitirá a la empresa financiera XYZ tomar decisiones informadas sobre cómo optimizar su uso de aplicaciones. Se espera que la empresa pueda reducir costos innecesarios, mejorar la eficiencia en la gestión de operaciones y tener una visión completa y actualizada de su portafolio de aplicaciones en todo momento.
6. **Evaluación y seguimiento:** una vez implementada la arquitectura de la aplicación, se llevará a cabo una evaluación y seguimiento continuo para asegurarse de que se están logrando los objetivos de la solución. Se realizarán informes regulares sobre la gestión del portafolio de aplicaciones, se medirá el rendimiento y se evaluarán las mejoras que se pueden realizar. Se llevarán a cabo revisiones regulares de la arquitectura de la aplicación para asegurarse de que sigue siendo relevante y eficaz.
7. **Conclusiones:** la implementación de la arquitectura de la aplicación basada en TOGAF para la gestión del portafolio de aplicaciones ha permitido a la empresa financiera XYZ tener una visión completa y actualizada de sus aplicaciones. Se han mejorado los procesos de gestión de aplicaciones y se han reducido costos innecesarios. Además, se ha establecido una base sólida para la gestión de aplicaciones y la toma de decisiones informadas en el futuro.



Actividades de aprendizaje recomendadas

Le invito a realizar las siguientes actividades de aprendizaje recomendadas para esta semana:

- A. Crear su propio portafolio de aplicaciones empresariales, junto con el inventario. El portafolio de aplicaciones empresariales es una colección de aplicaciones y software que se utilizan en una organización para apoyar los procesos empresariales. El inventario es una lista detallada de todas las aplicaciones y software utilizados por la organización.

Para llevar a cabo esta actividad, el estudiante deberá seguir los siguientes pasos:

1. **Identificar las aplicaciones y software utilizados por la organización.** El estudiante deberá realizar una investigación exhaustiva y recopilar información sobre todas las aplicaciones y software que se utilizan en la organización, incluyendo su propósito, funcionalidad, arquitectura y cualquier otra información relevante.
2. **Clasificar las aplicaciones y software.** Una vez que se hayan identificado todas las aplicaciones y software, el estudiante deberá clasificarlos en diferentes categorías, como aplicaciones de gestión de recursos humanos, aplicaciones de contabilidad, software de diseño gráfico, entre otros.
3. **Crear el portafolio de aplicaciones empresariales.** Utilizando la información recopilada y clasificada, el estudiante deberá crear un portafolio de aplicaciones empresariales que muestre todas las aplicaciones y software utilizados en la organización, organizados por categoría y con una breve descripción de cada uno.
4. **Crear el inventario.** El estudiante deberá crear un inventario detallado de todas las aplicaciones y software utilizados en la organización, incluyendo información como el nombre de la aplicación, el proveedor, la versión, la fecha de instalación, el costo y cualquier otra información relevante.

5. **Revisar y validar el portafolio y el inventario.** Una vez que se hayan creado el portafolio y el inventario, es importante revisarlos cuidadosamente y validar que contengan la información correcta y sean coherentes entre sí.

En general, esta actividad requiere de una dedicación importante de tiempo y esfuerzo, ya que el portafolio de aplicaciones empresariales y el inventario son elementos críticos para la gestión efectiva de los sistemas de información en una organización.

- Para complementar su estudio investigar sistemas que permiten la gestión de arquitecturas de aplicaciones empresariales o herramientas de arquitectura empresarial que permitan la gestión de un portafolio de aplicaciones.
 - Realizar una matriz comparativa de características de estos sistemas.
 - Levantar los sistemas que componen su arquitectura dentro de la empresa o el caso de estudio.
 - Clasificar las aplicaciones usando el gráfico TIME.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.



Semana 5

2.3. Definición y diseño de arquitectura de aplicaciones base

El diseño de aplicaciones es un proceso de ingeniería que debe considerar muchos aspectos de diseño y alineación al negocio que permitan que las soluciones sean eficientes y escalables, agregando valor al negocio y rentabilidad.

Para diseñar la arquitectura de una aplicación, es necesario comprender los requisitos y objetivos empresariales: empiece por comprender los requisitos y objetivos empresariales que la arquitectura tendrá que respaldar. Esto te ayudará a asegurarte de que la arquitectura está alineada con las necesidades del negocio, y que será capaz de soportar los resultados deseados.

Es importante identificar los componentes clave y las partes interesadas: identifique los componentes clave y las partes interesadas que participarán en la arquitectura. Esto puede incluir aplicaciones, servicios, sistemas y personas, y le ayudará a comprender el alcance y la complejidad de la arquitectura.

A partir de los requisitos empresariales y los componentes clave, desarrolle un diseño de alto nivel para la arquitectura. Esto debería incluir una visión descendente de la arquitectura, mostrando cómo encajaran e interactuarán los diferentes componentes.

Defina los componentes individuales de la arquitectura con más detalle. Esto puede implicar la creación de especificaciones y diagramas detallados para cada componente, así como la definición de las interfaces y las interacciones entre los componentes.

Es importante considere cualquier restricción técnica u operativa que pueda afectar al diseño de la arquitectura. Esto puede incluir sistemas y tecnologías existentes, requisitos de rendimiento y escalabilidad, y cuestiones normativas o de cumplimiento.

Validar el diseño de la arquitectura es un paso clave para garantizar que cumple los requisitos de la empresa y es factible de implantar. Esto puede implicar la realización de entrevistas con las partes interesadas, la realización de evaluaciones técnicas o la creación de prototipos o pruebas de concepto.

Finalmente, basándose en los comentarios y la validación, refine y finalice el diseño de la arquitectura. Esto puede implicar realizar ajustes en el diseño o identificar requisitos o dependencias adicionales.

Con estas recomendaciones puede diseñar una arquitectura de aplicación que satisfaga las necesidades de la empresa y sea factible de implantar. Esto ayudará a garantizar que la arquitectura sea eficaz y eficiente, y que pueda respaldar las operaciones y el crecimiento de la empresa.

El diseño de una arquitectura de aplicaciones implica la creación de un proyecto detallado para la estructura y organización de las aplicaciones de una organización, con el fin de apoyar las metas y objetivos de negocio de la organización. Normalmente incluye los siguientes pasos

- 1. Identificar las metas y objetivos empresariales de la organización:** esto implica la identificación de las metas y objetivos empresariales específicos que la arquitectura de la aplicación está destinada a apoyar. Incluyendo la identificación de las partes interesadas clave y sus necesidades, los procesos empresariales clave y sus requisitos, y los impulsores empresariales clave y su impacto en la organización.
- 2. Definir el marco de la arquitectura de la aplicación:** se trata de definir la estructura y organización generales de la arquitectura de la aplicación, incluidos los componentes clave y sus relaciones, y los principios y normas que guiarán el desarrollo y la evolución de la arquitectura. Esto puede incluir la definición de los dominios de la arquitectura de la aplicación, como la interfaz de usuario, la gestión de datos y la integración, así como los puntos de vista y perspectivas de la arquitectura de la aplicación que se utilizarán para analizar y describir la arquitectura.
- 3. Desarrollar la arquitectura de la aplicación:** se trata de desarrollar un plan detallado para la arquitectura de la aplicación, basado en las metas y objetivos empresariales identificados en el paso 1 y en el marco definido en el paso 2. Este plan debe incluir una descripción detallada de la arquitectura de la aplicación, así como una descripción de la arquitectura de la aplicación. Este anteproyecto debe incluir una descripción detallada del estado actual de la organización, su estado objetivo y los pasos necesarios para pasar del estado actual al estado objetivo. También debe incluir una descripción detallada de los componentes y atributos clave de la arquitectura de la aplicación, y cómo encajan entre sí para apoyar las metas y objetivos empresariales de la organización.
- 4. Implementar la arquitectura de la aplicación:** se trata de poner en práctica la arquitectura de la aplicación, de acuerdo con el plan desarrollado en el paso 3. Con las actividades como el diseño y la implementación de la arquitectura de la aplicación, considerando el diseño y el desarrollo de aplicaciones, el despliegue, la configuración y la integración de aplicaciones con otros sistemas y tecnologías.

En general, el diseño de una arquitectura de aplicaciones implica la creación de un plan detallado para la estructura y organización de las aplicaciones de la organización.

Considerando los puntos mencionados anteriormente se debe profundizar en las arquitecturas y como se relacionan los componentes de software en una solución.

2.3.1. Arquitectura de aplicaciones

La arquitectura de aplicaciones puede respaldar el despliegue y el funcionamiento de aplicaciones en diferentes entornos, como locales, en la nube o híbridos, proporcionando una estructura clara y coherente para el diseño y la implementación de aplicaciones.

Puede definir un conjunto de componentes, patrones y estándares comunes que pueden utilizarse en diferentes aplicaciones y entornos, lo que puede ayudar a garantizar que las aplicaciones se diseñen e implementen de forma coherente e interoperable. Esto puede facilitar el despliegue y el funcionamiento de las aplicaciones en distintos entornos y contribuir a reducir la complejidad y la sobrecarga asociadas a la gestión y el mantenimiento de las aplicaciones en distintos entornos.

También puede proporcionar orientación y apoyo para diseñar e implementar aplicaciones de una manera que sea escalable, fiable y segura, lo que puede ayudar a garantizar que las aplicaciones puedan funcionar eficazmente en diferentes entornos. Por ejemplo, una arquitectura de aplicaciones puede orientar sobre cómo diseñar aplicaciones de alta disponibilidad, cómo implementar componentes redundantes y resistentes y cómo proteger las aplicaciones frente a las amenazas y vulnerabilidades más comunes.

En general, una arquitectura de aplicaciones puede apoyar el despliegue y funcionamiento de aplicaciones en diferentes entornos, como en las instalaciones, en la nube o híbridos, proporcionando una estructura clara y coherente para el diseño y la implementación de aplicaciones, y proporcionando orientación y apoyo para el diseño e implementación de aplicaciones que sean escalables, fiables y seguras.

2.3.2. Clean code

El diseño de aplicaciones basado en Clean Code se enfoca en escribir código legible, mantenable y fácil de probar. Algunas de las características clave de un diseño basado en Clean Code incluyen:

1. **Nombres significativos:** los nombres de variables, métodos y clases deben ser específicos y significativos, de modo que sea fácil entender su propósito sin tener que leer el código en sí.
2. **Funciones pequeñas:** las funciones deben tener una sola responsabilidad y ser lo más pequeñas posible. Esto ayuda a asegurar que el código sea fácil de entender y probar.
3. **Comentarios útiles:** los comentarios deben ser útiles y explicar el propósito del código, no solo repetir lo que el código ya está haciendo.
4. **Buenas prácticas de programación:** el código debe seguir buenas prácticas de programación, como usar patrones de diseño, evitar el código duplicado y seguir estándares de codificación.
5. **Pruebas unitarias:** el código debe ser fácil de probar mediante pruebas unitarias, lo que ayuda a asegurar que el código funcione correctamente y sea fácil de mantener.
6. **Simplicidad:** el código debe ser lo más simple posible, evitando la complejidad innecesaria.
7. **Consistencia:** el código debe ser consistente en cuanto a estilo, nomenclatura y estructura.

Siguiendo estas pautas, se espera que el código sea fácil de entender, modificar y mantener a largo plazo, lo que ayudará a reducir el tiempo y los costos de desarrollo de una aplicación. (Martin, 2008).

2.3.3. Arquitectura SOA

La Arquitectura Orientada a Servicios (SOA) es un estilo arquitectónico para construir sistemas y aplicaciones distribuidos. Se basa en la idea de organizar las aplicaciones como un conjunto de servicios independientes y autónomos que se comunican entre sí a través de una red.

En SOA, los servicios se definen como unidades autónomas de funcionalidad a las que pueden acceder e invocar otros servicios o aplicaciones cliente. Los servicios se implementan normalmente como componentes o módulos de software diseñados para realizar una tarea específica o para proporcionar una capacidad específica. Los servicios suelen estar expuestos a la red mediante un protocolo de comunicación

común, como HTTP o SOAP, y se accede a ellos mediante una interfaz bien definida que especifica las operaciones y los datos que el servicio puede proporcionar.

SOA permite componer e integrar diferentes aplicaciones y servicios de forma flexible y modular, sin necesidad de que estén estrechamente acoplados o integrados. Esto permite que las aplicaciones se construyan y mantengan de forma más ágil y escalable, y que se adapten más fácilmente a los cambiantes requisitos empresariales y entornos tecnológicos.

En general, SOA es un estilo arquitectónico basado en la idea de organizar las aplicaciones como un conjunto de servicios independientes y autónomos que se comunican entre sí a través de una red. Permite componer e integrar aplicaciones de forma flexible y modular, y puede ayudar a mejorar su agilidad, escalabilidad y adaptabilidad.

2.3.4. Arquitectura de microservicios

La arquitectura de microservicios es un estilo arquitectónico para construir sistemas y aplicaciones distribuidos. Se basa en la idea de descomponer una aplicación compleja en un conjunto de servicios pequeños e independientes, cada uno responsable de una tarea o capacidad específica.

En la arquitectura de microservicios, cada servicio es una unidad pequeña y autónoma de funcionalidad que puede desarrollarse, desplegarse y mantenerse independientemente de los demás servicios. Esto permite que los servicios se desarrollem y evolucionen de forma más ágil e incremental, y que se adapten más fácilmente a los cambiantes requisitos empresariales y entornos tecnológicos.

Los microservicios suelen implementarse como componentes o módulos de software diseñados para realizar una tarea específica o proporcionar una capacidad concreta. Los microservicios suelen estar expuestos a la red mediante un protocolo de comunicación común, como HTTP o gRPC, y se accede a ellos mediante una interfaz bien definida que especifica las operaciones y los datos que el servicio puede proporcionar.

La arquitectura de microservicios permite componer e integrar diferentes aplicaciones y servicios de forma flexible y modular, sin necesidad de que estén estrechamente acoplados o integrados. Esto permite que las aplicaciones se construyan y mantengan de forma más ágil y escalable, y

que se adapten más fácilmente a los cambiantes requisitos empresariales y entornos tecnológicos.

En general, la arquitectura de microservicios es un estilo arquitectónico que se basa en la idea de descomponer una aplicación compleja en un conjunto de servicios pequeños e independientes, cada uno responsable de una tarea o capacidad específica. Permite componer e integrar aplicaciones de forma flexible y modular, y puede ayudar a mejorar su agilidad, escalabilidad y adaptabilidad.



Para una mejor comprensión del tema, le invito a observar el vídeo: [Arquitecturas de microservicios: intentando no morir en el intento, por Alberto Grande](#).

2.3.5. Event driven architecture

La arquitectura basada en eventos es un estilo arquitectónico para construir sistemas y aplicaciones distribuidos. Se basa en la idea de desencadenar acciones o respuestas en un sistema a partir de eventos que se producen en el sistema o en el entorno.

En la arquitectura basada en eventos, los eventos se definen como sucesos o cambios en el sistema o en el entorno que son de interés para la aplicación. Los eventos pueden ser generados por una amplia variedad de fuentes, como acciones del usuario, operaciones del sistema o estímulos externos. Los eventos suelen representarse como mensajes o notificaciones que se transmiten a las partes interesadas, como otras aplicaciones o servicios, o a los componentes que gestionan los eventos y se encargan de reaccionar ante ellos.

La arquitectura orientada a eventos permite desacoplar aplicaciones y servicios e interactuar entre sí de forma asíncrona, utilizando los eventos como principal medio de comunicación. Esto permite que las aplicaciones y los servicios se desarrollen y evolucionen de forma más independiente y desacoplada, y que se adapten más fácilmente a los cambiantes requisitos empresariales y entornos tecnológicos.

En general, la arquitectura orientada a eventos es un estilo arquitectónico que se basa en la idea de desencadenar acciones o respuestas en un sistema a partir de eventos que se producen en el sistema o en el entorno. Permite que las aplicaciones y los servicios se desacoplen unos de otros e interactúen entre sí de forma asíncrona, utilizando los eventos como principal medio de comunicación. Puede ayudar a mejorar la agilidad, escalabilidad y adaptabilidad de un sistema.

2.3.6. DevSecOps

Hoy en día la arquitectura de aplicaciones se debe pensar con procesos automatizados de integración y despliegue, siempre pensando en la seguridad.

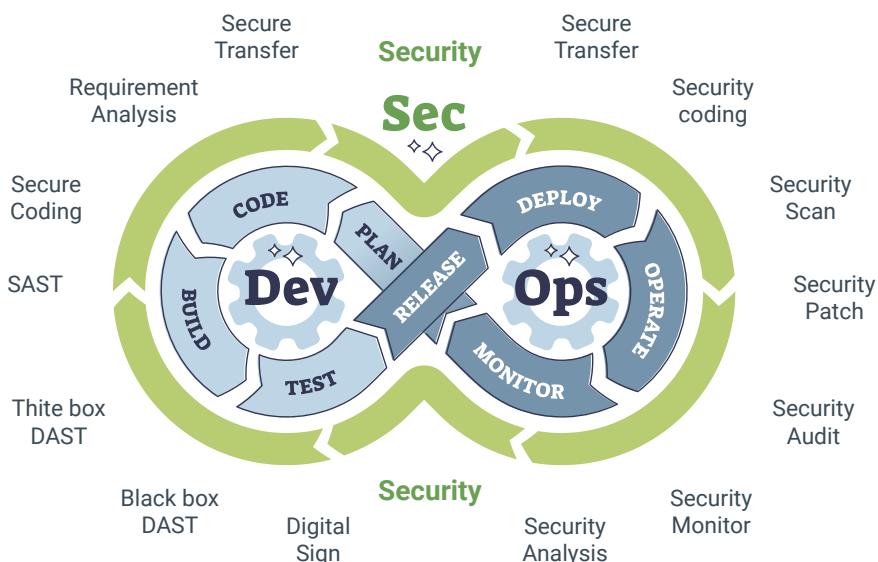
DevSecOps es un enfoque de desarrollo de software que hace hincapié en la colaboración, automatización e integración entre los equipos de desarrollo, seguridad y operaciones (ver figura 6). El objetivo de DevSecOps es permitir a las organizaciones ofrecer software de alta calidad de forma más rápida y fiable, integrando procesos y prácticas de seguridad en el ciclo de vida del desarrollo de software.

Tradicionalmente, los equipos de desarrollo, seguridad y operaciones han trabajado en silos, con escasa colaboración o coordinación entre ellos. Esto ha dado lugar a menudo a procesos de desarrollo de software lentos, ineficaces y propensos a errores, así como a vulnerabilidades de la seguridad que se descubren tarde en el proceso de desarrollo, cuando son más difíciles y caras de solucionar.

DevSecOps pretende superar estos retos fomentando una estrecha colaboración y comunicación entre los equipos de desarrollo, seguridad y operaciones a lo largo de todo el proceso de desarrollo de software. Esto incluye la Integración de Procesos y Prácticas de Seguridad en el proceso de Integración y Entrega Continuas (CI/CD), de modo que las pruebas y análisis de seguridad puedan realizarse de forma automática y continua como parte del proceso de desarrollo de software.

En general, DevSecOps es un factor clave para el desarrollo de software ágil, eficiente y seguro, y es cada vez más importante a medida que las organizaciones se esfuerzan por ofrecer software de alta calidad de forma más rápida y fiable.

Figura 6.
Proceso DevSecOps



Nota. Adaptado de Proceso de DevSecOps Quantitative Evaluation of Software Quality, por Barry W Boehm, John R Brown, y Myron Lipow, 1976

Este recurso electrónico grafica el proceso de mejora continua en cada uno de los pasos del proceso de Desarrollo, Seguridad y Operaciones, con las pruebas de seguridad que se realizan en cada una de las etapas.

2.3.7. Aplicaciones de diseño

Hay muchas herramientas que pueden utilizarse para diseñar la arquitectura de una aplicación. Algunas herramientas comunes son:

- **Herramientas de diagramación**, como Microsoft Visio o Lucidchart, que permiten crear diagramas y modelos de la arquitectura de la aplicación.
- **Lenguajes de modelado**, como el Lenguaje Unificado de Modelado (UML), que proporcionan una forma estandarizada de representar los componentes y las interacciones de una aplicación.
- **Patrones de diseño**, que son soluciones probadas a problemas de diseño comunes que pueden aplicarse a la arquitectura de una aplicación.

- **Arquitecturas de referencia**, que son arquitecturas predefinidas que pueden utilizarse como punto de partida para diseñar una nueva aplicación.
- **Herramientas de creación de prototipos**, como *wireframing* o *mockups*, que permiten crear rápidamente una representación visual de la interfaz de usuario de la aplicación.
- **Herramientas de desarrollo**, como Entornos de Desarrollo Integrados (IDE) y sistemas de control de versiones, que se utilizan para escribir y gestionar el código de la aplicación.

El uso de estas herramientas permite diseñar una arquitectura de aplicación bien estructurada, eficaz y fácil de mantener.

2.4. Diseño de arquitectura empresarial

El diseño de la arquitectura empresarial es el proceso de creación de un plan completo y estructurado para la organización, el desarrollo y la gestión de los sistemas de Tecnología de la Información (TI) de una empresa. El diseño de la arquitectura empresarial suele implicar los siguientes pasos.

1. Identificar las metas y objetivos de negocio de la empresa, así como el estado actual de sus sistemas de TI. Esto ayudará a establecer una línea de base para la arquitectura empresarial y a identificar cualquier laguna o área de mejora.
2. Desarrollar una arquitectura empresarial de alto nivel que describa los componentes clave y las relaciones de la empresa, así como los procesos empresariales, los productos y servicios, la información y la tecnología que soportan dichos componentes.
3. Desarrollar una arquitectura de aplicación detallada que especifique el *hardware*, el *software* y las redes necesarios para respaldar la arquitectura empresarial. Esto debe incluir la estructura y organización general de los sistemas de TI, así como las relaciones entre los diferentes sistemas y las interfaces que se utilizan para conectarlos.

4. Desarrollar una arquitectura tecnológica que defina el *hardware*, el *software* y las redes necesarios para respaldar la arquitectura de la aplicación. Esto debe incluir tanto la infraestructura como las plataformas necesarias para ejecutar los sistemas informáticos de la organización.
5. Desarrollar una arquitectura de datos que defina las estructuras, formatos y estándares de datos que se utilizan en toda la empresa. Esto debe incluir los modelos de datos, diccionarios de datos y diagramas de flujo de datos que se necesitan para apoyar los procesos de negocio de la organización y los sistemas de TI.
6. Desarrollar una arquitectura de seguridad que especifique las medidas y controles necesarios para proteger los activos de información de la empresa. Esto debe incluir políticas, procedimientos y tecnologías para garantizar la confidencialidad, integridad y disponibilidad de los datos y sistemas de la organización.
7. Desarrollar un plan de gobierno para gestionar y mantener la arquitectura empresarial a lo largo del tiempo. Esto debería incluir estrategias para garantizar la alineación con las metas y objetivos de negocio de la empresa, así como para supervisar, evaluar y adaptar la arquitectura empresarial según sea necesario.

En general, el diseño de la arquitectura empresarial proporciona un enfoque completo y estructurado para diseñar, desarrollar y gestionar los sistemas de TI de una empresa. Puede ayudar a mejorar el rendimiento, la eficiencia y la agilidad de la empresa, y permitirle adaptarse y responder a los cambios en su entorno empresarial.



Para una mejor comprensión del tema, le invito a observar el vídeo: [SGVirtual - Archimate para arquitectos empresariales](#).



Actividades de aprendizaje recomendadas

Le invito a realizar las siguientes actividades de aprendizaje recomendadas para esta semana:

- A. Analizar diagramas de arquitectura en sus distintos tipos como microservicios, arquitectura SOA y EDA (Event-Driven Architecture). El análisis de diagramas de arquitectura es una habilidad fundamental en el campo de la arquitectura empresarial y puede ayudar al estudiante a comprender cómo los sistemas de información están diseñados y cómo funcionan.

Para llevar a cabo esta actividad, el estudiante deberá seguir los siguientes pasos:

1. **Identificar los diagramas de arquitectura relevantes.** El estudiante deberá recopilar una variedad de diagramas de arquitectura que representen diferentes tipos de arquitectura, como microservicios, SOA y EDA.
2. **Analizar los diagramas de arquitectura.** El estudiante deberá examinar cuidadosamente cada diagrama de arquitectura y analizar su contenido para identificar los componentes clave, las interacciones y las dependencias entre los diferentes elementos de la arquitectura.
3. **Comparar y contrastar los diferentes tipos de arquitectura.** El estudiante deberá comparar y contrastar los diferentes tipos de arquitectura que se representan en los diagramas analizados, identificando las similitudes y diferencias en términos de diseño, funcionalidad y beneficios empresariales.
4. **Reflexionar y evaluar el análisis realizado.** Una vez que se hayan analizado los diagramas de arquitectura, el estudiante deberá reflexionar sobre el proceso de análisis y evaluar su propia comprensión y habilidades para analizar diagramas de arquitectura.
5. **Investigar y aprender más sobre los temas identificados.** Si se identifican áreas de interés o necesidades de mejora durante el

análisis, el estudiante deberá investigar y aprender más sobre esos temas para mejorar su comprensión y habilidades en el futuro.

En general, esta actividad requerirá que el estudiante dedique una cantidad significativa de tiempo y esfuerzo para analizar cuidadosamente los diagramas de arquitectura y comprender los conceptos y principios clave de los diferentes tipos de arquitectura.

Seleccione una herramienta que permita el diagramado de arquitecturas, ubicar los componentes levantados del caso de estudio o de la empresa seleccionada.

- B. Revisar el vídeo: [Using the ArchiMate® Modeling Language in Real Life Examples](#).



Semana 6

2.5. Arquitectura de integración

En general, la elección de la arquitectura para la integración de software depende de los requisitos y limitaciones específicos del proyecto. Existen muchos estilos y enfoques arquitectónicos diferentes que pueden utilizarse para la integración de software, como la Arquitectura Orientada a Servicios (SOA), la arquitectura de microservicios y la arquitectura basada en eventos, entre otros. La mejor elección de arquitectura dependerá de las necesidades y objetivos específicos del proyecto, y de las compensaciones aceptables en términos de complejidad, rendimiento y funcionalidad. Es importante evaluar cuidadosamente las distintas opciones y seleccionar la arquitectura que mejor se adapte a los requisitos y limitaciones específicos del proyecto.

2.6. Interfaces de comunicación

Una interfaz de comunicación en arquitectura de aplicaciones es un mecanismo que permite que dos o más aplicaciones o sistemas se comuniquen entre sí y compartan información. Las interfaces de comunicación pueden ser de diferentes tipos y pueden utilizarse para diferentes propósitos en una arquitectura de aplicaciones. Algunos ejemplos

comunes de interfaces de comunicación en arquitectura de aplicaciones incluyen:

1. **Interfaces de aplicación a aplicación:** estas interfaces permiten que dos aplicaciones se comuniquen directamente entre sí y comparten información. Por ejemplo, una aplicación de gestión de proyectos puede utilizar una interfaz de aplicación a aplicación para obtener información de una aplicación de seguimiento de horas.
2. **Interfaces de aplicación a base de datos:** estas interfaces permiten que las aplicaciones accedan a una base de datos y realicen operaciones de lectura y escritura de datos.
3. **Interfaces de servicio web:** estas interfaces utilizan el protocolo HTTP para permitir que las aplicaciones se comuniquen a través de la red utilizando mensajes estructurados en formato XML o JSON.
4. **Interfaces de mensajería:** estas interfaces utilizan colas de mensajes y otros mecanismos para permitir que las aplicaciones se comuniquen de forma asincrónica y no dependiente del tiempo.

En general, las interfaces de comunicación son un componente importante de la arquitectura de aplicaciones, ya que permiten que las aplicaciones se integren y comparten información de manera eficiente.

2.6.1. Protocolos

Existen muchos protocolos diferentes que se utilizan para comunicarse entre aplicaciones de software, dependiendo de los requisitos y características, específicos de las aplicaciones y del entorno de red en el que se ejecutan. Algunos de los protocolos más comunes que se utilizan para la comunicación entre aplicaciones son los siguientes:

1. **Protocolo de Control de Transmisión (TCP):** TCP es un protocolo de transporte ampliamente utilizado que proporciona una comunicación fiable y orientada a la conexión entre aplicaciones. Garantiza que los datos se entreguen de forma fiable y en el orden correcto, estableciendo una conexión virtual entre las aplicaciones y retransmitiendo cualquier dato perdido o dañado.
2. **Protocolo de Datagramas de Usuario (UDP):** UDP es un protocolo de transporte simple y sin conexión que proporciona una entrega de

datos entre aplicaciones al mejor esfuerzo. No establece una conexión virtual entre las aplicaciones y no garantiza la entrega ni la integridad de los datos.

3. **Protocolo de Transferencia de Hipertexto (HTTP):** HTTP es un protocolo de capa de aplicación ampliamente utilizado para comunicarse entre aplicaciones *web* y servidores *web*. Define un conjunto de reglas y convenciones para formatear y transmitir datos, y es la base de la *World Wide Web*.
4. **Protocolo de Transferencia de Archivos (FTP):** FTP es un protocolo de capa de aplicación de uso común que se utiliza para transferir archivos entre ordenadores. Permite a los usuarios cargar y descargar archivos desde un servidor remoto, y gestionar los archivos en el servidor.
5. **Protocolo Simple de Transferencia de Correo (SMTP):** SMTP es un protocolo de capa de aplicación ampliamente utilizado para transmitir mensajes de correo electrónico entre servidores de correo y clientes. Define un conjunto de reglas y convenciones para formatear y transmitir mensajes de correo electrónico, y es la base del sistema de correo electrónico.

En general, hay muchos protocolos diferentes que se utilizan para comunicarse entre aplicaciones de *software*, dependiendo de los requisitos y características, específicos de las aplicaciones y del entorno de red en el que se ejecutan. Estos protocolos proporcionan la base para la comunicación y colaboración entre aplicaciones, y son esenciales para el funcionamiento de los sistemas informáticos modernos.

2.6.2. SOAP y REST

REST y SOAP son dos estilos arquitectónicos diferentes para construir servicios *web*, que son aplicaciones que proporcionan API (Interfaces de Programación de Aplicaciones) basadas en la *web* que pueden ser utilizadas por otras aplicaciones para acceder a datos y servicios a través de *Internet*.

- **REST (Representational State Transfer)** es un estilo arquitectónico basado en los principios de la *World Wide Web*. Define un conjunto de restricciones y directrices para crear servicios *web* escalables, modulares y fáciles de usar. Los servicios *web* RESTful suelen

construirse mediante HTTP (Hypertext Transfer Protocol) y utilizan los métodos HTTP (GET, POST, PUT, DELETE) para realizar operaciones sobre los recursos.

- **SOAP** (Simple Object Access Protocol) es un protocolo basado en XML que se utiliza para intercambiar información estructurada entre aplicaciones a través de una red. Define un conjunto de reglas y convenciones para formatear y transmitir datos, y suele utilizarse junto con otros protocolos, como HTTP o SMTP (Simple Mail Transfer Protocol), para proporcionar una comunicación segura y fiable entre aplicaciones.

Es difícil decir qué estilo arquitectónico es mejor, ya que en última instancia depende de los requisitos y limitaciones específicos de las aplicaciones y del entorno en el que se ejecutan. En general, REST se considera más sencillo y ligero que SOAP, y es más adecuado para API basadas en *web* a las que necesita acceder una amplia gama de clientes. SOAP, por su parte, es más adecuado para aplicaciones complejas de nivel empresarial que requieren características y funcionalidades más avanzadas, como seguridad, fiabilidad e interoperabilidad. En última instancia, la elección del estilo arquitectónico debe basarse en los requisitos y objetivos específicos de las aplicaciones y la organización.

En general, tanto REST como SOAP son enfoques válidos para construir servicios *web*, y la elección entre ellos depende de los requisitos y limitaciones específicos del proyecto. A menudo se considera que REST es más sencillo y ligero que SOAP, lo que lo hace idóneo para crear servicios *web* destinados a ser consumidos por una amplia variedad de clientes, incluidos navegadores, *web*, dispositivos móviles y dispositivos IoT. SOAP, por otro lado, suele considerarse más potente y flexible que REST, lo que lo hace idóneo para crear servicios *web* que requieren un alto grado de interoperabilidad y seguridad. En última instancia, la elección entre REST y SOAP depende de las necesidades y objetivos específicos del proyecto, y de las compensaciones aceptables en términos de complejidad, rendimiento y funcionalidad.

2.6.3. REST y GraphQL

REST y GraphQL son dos estilos arquitectónicos diferentes para construir servicios *web*, que son sistemas que permiten a diferentes aplicaciones comunicarse e intercambiar datos a través de *Internet*.

- **REST** (Representational State Transfer) es un estilo arquitectónico basado en un conjunto de principios para diseñar servicios *web* sencillos, ligeros y escalables. Los servicios *web* RESTful se construyen en torno al concepto de recursos, que son entidades lógicas a las que se puede acceder y manipular utilizando un conjunto uniforme de operaciones, como GET, PUT, POST y DELETE. Los servicios *web* RESTful suelen estar diseñados para ser apátridas, lo que significa que no mantienen ninguna información sobre el estado del cliente o del servidor.
- **GraphQL** es un lenguaje de consulta y un sistema en tiempo de ejecución que se utiliza para construir servicios *web*. Se basa en la idea de permitir a los clientes solicitar datos específicos al servidor, utilizando un lenguaje de consulta declarativo que describe los datos que se necesitan. GraphQL permite a los clientes especificar exactamente los datos que necesitan, y el servidor es responsable de devolver los datos en el formato especificado. Este enfoque permite a los clientes recuperar solo los datos que necesitan, lo que puede mejorar el rendimiento y la eficiencia del servicio *web*.

En general, tanto REST como GraphQL son enfoques válidos para construir servicios *web*, y la elección entre ellos depende de los requisitos y limitaciones específicos del proyecto. A menudo se considera que REST es más sencillo y está más extendido que GraphQL, lo que lo hace más adecuado para crear servicios *web* destinados a ser consumidos por una amplia variedad de clientes, incluidos navegadores, web, dispositivos móviles y dispositivos IoT. Por otro lado, GraphQL suele considerarse más potente y flexible que REST, lo que lo hace idóneo para crear servicios *web* que requieren un alto grado de rendimiento y flexibilidad.

2.6.4. gRPC

gRPC es un *framework* RPC (Remote Procedure Call Framework) universal de código abierto y alto rendimiento desarrollado y mantenido por Google. Se basa en los principios de la Arquitectura Orientada a Servicios (SOA) y está diseñado para permitir una comunicación eficiente entre microservicios, servicios en la nube y otros sistemas distribuidos.

gRPC utiliza HTTP/2 como protocolo de transporte y Protocol Buffers como formato de serialización. Esto permite a gRPC proporcionar una plataforma

de comunicación de alto rendimiento, baja latencia e independiente del lenguaje para microservicios y otros sistemas distribuidos.

gRPC soporta una amplia gama de lenguajes de programación, incluyendo C++, Java, Python, Go, Ruby, C#, y muchos otros. Proporciona un rico conjunto de características y funcionalidades, incluyendo soporte para *streaming* bidireccional, control de flujo, autenticación y cancelación.

2.6.5. Cloud VPN

Una VPN en la nube es una Red Privada Virtual (VPN) que se ofrece como un servicio en la nube. Una VPN es una tecnología de red que crea una conexión segura y cifrada entre el dispositivo de un usuario y una red remota, permitiendo al usuario acceder a la red y a sus recursos como si estuviera conectado directamente a la red.

Una VPN en la nube suele ser suministrada por un proveedor de servicios en la nube, que aloja los servidores VPN y gestiona la infraestructura VPN en nombre del usuario. El usuario puede entonces conectarse al servicio VPN desde su dispositivo, utilizando un cliente VPN o una aplicación proporcionada por el proveedor de servicios en la nube. Esto permite al usuario acceder a los recursos de la red remota de forma segura, sin tener que conectarse directamente a la red o gestionar la infraestructura VPN por sí mismo.

Las VPN en la nube se utilizan normalmente para permitir el acceso remoto a redes corporativas o para proporcionar conectividad segura entre diferentes redes o ubicaciones. También pueden utilizarse para conectarse de forma segura a servicios de nube pública, o para permitir escenarios de nube híbrida en los que los recursos se comparten entre una red privada local y una nube pública.

En general, una VPN en la nube es una forma segura y cómoda de acceder a redes y recursos remotos, sin tener que conectarse directamente a la red ni gestionar la infraestructura VPN. Es un componente importante de las soluciones de red y conectividad basadas en la nube, y puede ayudar a mejorar la seguridad, el rendimiento y la flexibilidad de la red de una organización.

2.6.6. Server-Side Rendering – SSR

La Renderización del Lado del Servidor, o SSR, es una técnica para renderizar una aplicación del lado del cliente o basada en JavaScript en el servidor y luego enviar la página completamente renderizada al cliente. Esto permite al cliente recibir una página HTML completa, que puede mostrarse inmediatamente sin necesidad de procesamiento adicional por parte del cliente. La SSR puede mejorar el rendimiento y la experiencia de usuario de una aplicación web, sobre todo en el caso de usuarios con conexiones a Internet o dispositivos más lentos. Se utiliza habitualmente en aplicaciones que dependen en gran medida de la renderización del lado del cliente, como las aplicaciones de una sola página o las creadas con marcos como React o Angular.

2.6.7. Client-Side Rendering – CSR

La Renderización del Lado del Cliente, o CSR, es una técnica para renderizar una aplicación web en el lado del cliente, normalmente utilizando JavaScript. En una aplicación basada en CSR, el HTML inicial enviado desde el servidor suele ser mínimo, y la mayor parte de la renderización se realiza en el lado del cliente mediante JavaScript. Esto permite una experiencia de usuario más interactiva y dinámica, pero también puede dar lugar a un rendimiento más lento, sobre todo en dispositivos o conexiones a Internet más lentos. El CSR se utiliza habitualmente en aplicaciones de una sola página y en aquellas creadas con marcos como React o Angular.

2.6.8. WebSockets

Los WebSockets son un protocolo que permite la comunicación en tiempo real entre un servidor web y un cliente. Proporcionan un canal de comunicación bidireccional *full-duplex* a través de una única conexión TCP, lo que permite una comunicación de baja latencia y alta frecuencia entre el servidor y el cliente. Esto los hace idóneos para aplicaciones que requieren actualizaciones frecuentes o interacciones en tiempo real, como los juegos en línea, el chat o la edición colaborativa de documentos. WebSockets es diferente de HTTP, que es un protocolo de petición-respuesta diseñado para la comunicación unidireccional. Los WebSockets suelen utilizarse junto con otras tecnologías web, como HTML y JavaScript, para crear aplicaciones web interactivas y dinámicas.

2.6.9. ETL – Extraer, Transformar, Cargar

ETL significa E xtraer, T transformar y C argar. Se trata de un proceso de almacenamiento de datos que consiste en extraer datos de diversas fuentes, transformarlos en un formato adecuado para el análisis y cargarlos en un almacén de datos u otro destino. El objetivo de ETL es permitir a las empresas acceder a datos de múltiples fuentes y analizarlos de forma coherente y eficaz. El proceso suele implicar la extracción de datos de diversas fuentes, como bases de datos, archivos planos u otros sistemas, y su posterior transformación a un formato común. Esto puede implicar limpiar o transformar los datos para eliminar incoherencias, errores o duplicados, así como aplicar reglas o lógica para combinar o manipular los datos. A continuación, los datos transformados se cargan en un destino, como un almacén de datos, para su posterior análisis y elaboración de informes. La ETL es un componente fundamental del almacenamiento de datos y la inteligencia empresarial, ya que permite a las empresas obtener información de sus datos y tomar decisiones basadas en ellos.

2.7. Arquitectura de datos

La arquitectura de aplicaciones y la arquitectura de datos están estrechamente relacionadas ya que la arquitectura de aplicaciones define cómo se desarrollan y utilizan las aplicaciones que acceden y procesan los datos, mientras que la arquitectura de datos define cómo se almacenan, protegen y utilizan los datos en una organización.

En la práctica, esto significa que la arquitectura de aplicaciones debe tener en cuenta la arquitectura de datos existente al diseñar las aplicaciones y sus interconexiones. Por ejemplo, si la arquitectura de datos incluye una base de datos centralizada que se utiliza por varias aplicaciones, la arquitectura de aplicaciones debe diseñarse de manera que estas aplicaciones puedan acceder y utilizar los datos de manera eficiente y coherente.

Además, la arquitectura de aplicaciones y la arquitectura de datos deben trabajar juntas para garantizar la integridad y la seguridad de los datos, y deben ser diseñadas de manera que se puedan adaptar y escalar eficientemente a medida que cambien las necesidades de la organización.

En resumen, la arquitectura de aplicaciones y la arquitectura de datos son dos aspectos clave de la arquitectura empresarial y deben trabajar juntas

para proporcionar una solución eficiente y sostenible para el procesamiento y el almacenamiento de datos en una organización.



Actividades de aprendizaje recomendadas

La actividad de aprendizaje recomendada para el estudiante es analizar los protocolos que tienen todos los sistemas o aplicaciones que se utilizan en su arquitectura, diagramar las interacciones en su diseño de arquitectura completo, determinar si existen protocolos distintos a los estudiados en la semana, y representar las transacciones que pueden fluir desde el cliente en su ecosistema de arquitectura.

Para llevar a cabo esta actividad, el estudiante deberá seguir los siguientes pasos:

- 1. Identificar los sistemas y aplicaciones que se utilizan en su arquitectura.** El estudiante deberá realizar una investigación exhaustiva y recopilar información sobre los diferentes sistemas y aplicaciones que se utilizan en su arquitectura.
- 2. Analizar los protocolos utilizados por cada sistema o aplicación.** El estudiante deberá examinar cuidadosamente los protocolos utilizados por cada sistema o aplicación en su arquitectura para comprender cómo interactúan entre sí.
- 3. Diagramar las interacciones en su diseño de arquitectura completo.** Utilizando la información recopilada sobre los sistemas y aplicaciones y los protocolos utilizados, el estudiante deberá diagramar las interacciones en su diseño de arquitectura completo para comprender mejor cómo se comunican los diferentes componentes.
- 4. Determinar si existen protocolos distintos a los estudiados en la semana.** El estudiante deberá comparar los protocolos utilizados en su arquitectura con los que se han estudiado en la semana, identificando cualquier protocolo que sea diferente y comprender cómo afecta la interacción de los sistemas y aplicaciones.
- 5. Representar las transacciones que pueden fluir desde el cliente en su ecosistema de arquitectura.** El estudiante deberá representar las transacciones que pueden fluir desde el cliente a través de

su ecosistema de arquitectura, para comprender mejor cómo los diferentes sistemas y aplicaciones interactúan con los clientes y entre sí.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

En general, esta actividad requerirá que el estudiante dedique una cantidad significativa de tiempo y esfuerzo para recopilar y analizar información sobre los diferentes sistemas y aplicaciones utilizados en su arquitectura, así como los protocolos que se utilizan para comunicarse entre sí. Además, deberá tener habilidades para diagramar y representar interacciones y transacciones para comprender mejor el diseño de su arquitectura.



Semana 7

2.8. Componentes lógicos y físicos de aplicación

Los componentes físicos de las aplicaciones son las entidades tangibles y físicas que conforman una arquitectura de aplicaciones. Estos componentes pueden incluir *hardware*, como servidores, almacenamiento y equipos de red; *software*, como sistemas operativos, *middleware* y aplicaciones, e infraestructura, como centros de datos, conexiones de red y fuentes de alimentación.

Los componentes físicos específicos de las aplicaciones que se incluyen en una arquitectura de aplicaciones dependerán de las necesidades y requisitos de la organización. Por ejemplo, una organización pequeña puede tener una arquitectura relativamente simple que incluya unos pocos servidores y un puñado de aplicaciones, mientras que una organización más grande puede tener una arquitectura más compleja que incluya múltiples centros de datos, cientos de servidores y miles de aplicaciones.

Los componentes físicos de las aplicaciones son una parte importante de una arquitectura de aplicaciones, ya que proporcionan la infraestructura física y los recursos necesarios para soportar la arquitectura. Son los elementos tangibles y concretos que hacen que la arquitectura sea real y operativa, y son esenciales para que las aplicaciones y los sistemas de la arquitectura funcionen correctamente.

En general, los componentes físicos de la aplicación son las entidades tangibles y físicas que conforman una arquitectura de aplicaciones.

Estos componentes pueden incluir *hardware*, *software* e infraestructura, y proporcionan la infraestructura física y los recursos necesarios para soportar la arquitectura.

Los componentes físicos de aplicación específicos que se consideren “más importantes” dependerán de las necesidades y requisitos de la organización. Sin embargo, aquí hay 13 ejemplos de componentes físicos de aplicación que podrían considerarse importantes en muchas situaciones:

1. **Servidores:** son las máquinas físicas que alojan las aplicaciones y los datos de la arquitectura. Pueden estar dedicados a una aplicación o propósito específico, o pueden estar virtualizados y ser compartidos por múltiples aplicaciones.
2. **Almacenamiento:** es el *hardware* físico que se utiliza para almacenar los datos de la arquitectura. Puede incluir almacenamiento local en los servidores, así como sistemas de almacenamiento externo, como SAN (redes de área de almacenamiento) o almacenamiento en la nube.
3. **Equipo de red:** incluye el *hardware* físico, como conmutadores, enruteadores y cortafuegos, que se utiliza para conectar los servidores y el almacenamiento de la arquitectura.
4. **Sistemas operativos:** son las plataformas de *software* que se instalan en los servidores, y que proporcionan la infraestructura subyacente para las aplicaciones y servicios de la arquitectura.
5. **Middleware:** es el *software* que proporciona la infraestructura y los servicios que soportan las aplicaciones de la arquitectura. Algunos ejemplos son los servidores de aplicaciones, las bases de datos y los sistemas de mensajería.
6. **Aplicaciones:** son los programas de *software* que proporcionan la funcionalidad empresarial o técnica específica de la arquitectura. Pueden ser aplicaciones desarrolladas a medida o aplicaciones comerciales disponibles en el mercado (COTS).
7. **Centros de datos:** son las instalaciones físicas que albergan los servidores, el almacenamiento y los equipos de red de la arquitectura.

Pueden ser propiedad de la organización y estar gestionados por ella, o pueden estar alojados por un proveedor externo.

8. **Conexiones de red:** son las conexiones físicas, como líneas alquiladas, banda ancha o VPN, que proporcionan la conectividad de red entre los centros de datos, servidores y otros componentes de la arquitectura.
9. **Fuentes de alimentación:** se trata de la infraestructura física, como generadores, Sistemas de Alimentación Interrumpida (SAI) y Unidades de Distribución de Energía (PDU), que proporcionan la energía a los servidores, el almacenamiento y los equipos de red de la arquitectura.
10. **Dispositivos de seguridad:** son el hardware físico, como cortafuegos, Sistemas de Detección de Intrusiones (IDS) y sistemas de control de acceso, que se utilizan para proteger la arquitectura.
11. **Herramientas de supervisión y gestión:** son las herramientas de software y hardware que se utilizan para supervisar y gestionar el rendimiento, la disponibilidad y la seguridad de la arquitectura.
12. **Balanceadores de carga:** son los dispositivos físicos que distribuyen la carga de trabajo entre múltiples servidores o sistemas, para mejorar el rendimiento y la fiabilidad.
13. **Sistemas de recuperación de desastres:** son los sistemas físicos, como servidores de backup y almacenamiento.

Los componentes que deben incluirse en una arquitectura empresarial dependerán de las necesidades y requisitos específicos de la organización. Sin embargo, algunos componentes comunes que suelen incluirse en una arquitectura empresarial podrían incluir:

1. **Arquitectura de negocio:** describe el modelo de negocio, los procesos y los objetivos de la organización, y proporciona una visión de alto nivel del negocio.
2. **Arquitectura de datos:** describe los activos de datos de la organización, sus relaciones e interacciones, y las reglas y normas que rigen su uso.

3. **Arquitectura de aplicaciones:** describe las aplicaciones y sistemas que utiliza la organización, y muestra cómo encajan entre sí e interactúan.
4. **Arquitectura tecnológica:** describe la infraestructura tecnológica que soporta las operaciones de la organización, como servidores, almacenamiento y equipos de red.
5. **Arquitectura de seguridad:** describe los controles y políticas de seguridad que se utilizan para proteger los activos y datos de la organización.
6. **Arquitectura de integración:** describe los mecanismos y estándares que se utilizan para permitir que los diferentes componentes de la arquitectura se comuniquen e intercambien datos.

En general, una arquitectura empresarial suele incluir una serie de componentes diferentes que juntos proporcionan una visión global de la organización y sus operaciones.

2.8.1. Servidor de aplicaciones

Un servidor de aplicaciones se suele clasificar como parte de la capa de *middleware* de una arquitectura de aplicaciones. La capa de *middleware* es la que se sitúa entre las aplicaciones y la infraestructura tecnológica subyacente, y proporciona la infraestructura y los servicios necesarios para dar soporte a las aplicaciones.

Un servidor de aplicaciones es un tipo de *middleware* que proporciona un entorno de ejecución para las aplicaciones, así como varios servicios de infraestructura, como gestión de transacciones, seguridad y mensajería. Actúa como anfitrión de las aplicaciones y les permite interactuar con la infraestructura tecnológica subyacente, como bases de datos y otros servicios.

Al formar parte de la capa de *middleware*, un servidor de aplicaciones se sitúa entre las aplicaciones y la infraestructura tecnológica, y proporciona un puente entre ambas. Esto permite a las aplicaciones aprovechar los servicios y capacidades de la capa de *middleware*, sin tener que interactuar directamente con la infraestructura tecnológica subyacente.

En general, un servidor de aplicaciones se suele clasificar como parte de la capa de *middleware* de una arquitectura de aplicaciones. Esta capa proporciona la infraestructura y los servicios necesarios para dar soporte a las aplicaciones, y un servidor de aplicaciones es un tipo de *middleware* que proporciona un entorno de ejecución para las aplicaciones, así como diversos servicios de infraestructura.

2.8.2. Componentes de una arquitectura de microservicios

Una arquitectura de microservicios es una arquitectura de *software* que se basa en la idea de construir aplicaciones como una colección de servicios pequeños e independientes. Cada servicio está diseñado para ser autónomo y proporcionar una función empresarial o técnica específica.

Hay varios componentes que se incluyen típicamente en una arquitectura de microservicios, incluyendo:

1. **Servicios**: son los componentes individuales que conforman la arquitectura. Cada servicio está diseñado para ser autónomo y proporcionar una función empresarial o técnica específica.
2. **API Gateway**: es un componente que se sitúa entre los servicios y los clientes de la arquitectura. Actúa como un punto de entrada para las solicitudes entrantes, y enruta las solicitudes al servicio apropiado.
3. **Registro de servicios**: este es un componente que mantiene una lista de los servicios que están disponibles en la arquitectura, y permite a los servicios descubrir y comunicarse entre sí.
4. **Balanceador de carga**: es un componente que distribuye la carga de trabajo entre múltiples instancias de un servicio, para mejorar el rendimiento y la fiabilidad.
5. **Herramientas de supervisión y gestión**: son las herramientas que se utilizan para monitorizar y gestionar el rendimiento, la disponibilidad y la seguridad de la arquitectura.

En general, una arquitectura de microservicios suele incluir varios componentes, como servicios, una pasarela API, un registro de servicios, un equilibrador de carga y herramientas de supervisión y gestión. Estos componentes trabajan juntos para permitir que la arquitectura funcione con

eficacia y eficiencia, y para apoyar los objetivos empresariales o técnicos de la organización.

2.8.3. API management

La gestión de API es un conjunto de prácticas y tecnologías que se utilizan para gestionar las API (Interfaces de Programación de Aplicaciones) de una organización. Incluye actividades como diseñar, publicar, documentar y proteger las API, así como gestionar su ciclo de vida y su uso.

Para habilitar ecosistemas con gestión de API, se pueden seguir los siguientes pasos:

- 1. Identificar a las partes interesadas del ecosistema:** empiece por identificar a las partes interesadas del ecosistema, como socios, desarrolladores y clientes, y comprenda sus necesidades y objetivos. Esto le ayudará a determinar qué API deben estar disponibles y cómo deben diseñarse y gestionarse.
- 2. Diseñar y publicar API:** utilice los estándares de diseño de API y las mejores prácticas para diseñar las API que estarán disponibles en el ecosistema. Publique las API en un catálogo o portal central de API para que las partes interesadas puedan descubrirlas y acceder a ellas.
- 3. Documentar y probar las API:** proporcione una documentación exhaustiva de las API que incluya información sobre su funcionalidad, entradas, salidas y condiciones de error. Además, proporcione herramientas y procesos para probar las API, con el fin de garantizar que funcionan como se espera y que pueden ser utilizadas por las partes interesadas.
- 4. Proteger las API:** implantar controles y políticas de seguridad adecuados para proteger las API y los datos a los que acceden. Esto puede incluir mecanismos de autenticación y autorización, cifrado y otras medidas de seguridad.
- 5. Supervisar y gestionar el uso de las API:** utilice herramientas de supervisión y análisis para realizar un seguimiento del uso de las API e identificar posibles problemas o tendencias. Utilice herramientas de gestión de API para gestionar el ciclo de vida de las API, incluidas las versiones, la eliminación y la retirada.

Siguiendo estos pasos, puede utilizar la gestión de API para habilitar ecosistemas y apoyar la colaboración y la innovación. Esto le ayudará a garantizar que las API estén bien diseñadas, documentadas y sean seguras, y que las partes interesadas del ecosistema las utilicen de forma eficaz y eficiente.

2.8.4. Monetización de API

Una empresa puede monetizar sus activos de datos de varias formas, entre las que se incluyen las siguientes:

1. **Vender el acceso a los datos:** una forma de monetizar los datos es vender el acceso a ellos. Esto podría implicar la concesión de licencias de los datos a otras organizaciones, o proporcionarles un acceso a los datos basado en una suscripción.
2. **Utilizar los datos para generar ideas:** otra forma de monetizar los datos es utilizarlos para generar conocimientos e inteligencia que puedan venderse o utilizarse para mejorar las operaciones de la empresa. Por ejemplo, un minorista podría utilizar los datos de los clientes para identificar tendencias y patrones de compra, y utilizar esta información para mejorar su gestión de inventario y sus estrategias de *marketing*.
3. **Utilizar los datos para desarrollar nuevos productos o servicios:** una empresa también puede utilizar sus datos para desarrollar nuevos productos o servicios que puedan venderse a clientes o socios. Por ejemplo, un proveedor de servicios sanitarios podría utilizar los datos de los pacientes para desarrollar algoritmos de análisis predictivo que puedan utilizarse para identificar posibles riesgos para la salud y recomendar medidas preventivas.
4. **Utilizar los datos para mejorar los productos o servicios existentes:** los datos también pueden utilizarse para mejorar la calidad o el valor de los productos o servicios existentes. Por ejemplo, una compañía de seguros puede utilizar datos sobre el comportamiento y las preferencias de sus clientes para adaptar sus pólizas y ofertas a cada uno de ellos.

En general, una empresa puede monetizar sus activos de datos de muchas maneras. Por ejemplo, vendiendo el acceso a los datos, utilizándolos

para generar información, desarrollando nuevos productos o servicios y mejorando los ya existentes. Si aprovechan sus datos de forma eficaz, las empresas pueden generar nuevas fuentes de ingresos y obtener una ventaja competitiva en el mercado.

La arquitectura de aplicaciones puede desempeñar un papel crucial en el apoyo a la transformación digital de una empresa. La transformación digital es el proceso de utilizar tecnologías y datos digitales para mejorar y transformar las operaciones empresariales, y la arquitectura de aplicaciones es la base sobre la que se construyen y despliegan estas tecnologías y datos.

La arquitectura de aplicaciones proporciona la base para construir y desplegar tecnologías y servicios digitales. Al definir los componentes, las relaciones y los estándares que se utilizan en la arquitectura, permiten a la organización desarrollar y desplegar tecnologías y servicios digitales de forma coherente y escalable.

Permite a la organización integrar y alinear sus tecnologías y servicios digitales con sus procesos y objetivos empresariales. Al proporcionar una visión integral de las aplicaciones y sistemas que utiliza la organización, optimizan, alinean las tecnologías y servicios con los procesos y objetivos de negocio, para mejorar el rendimiento y la eficiencia.

La arquitectura de aplicaciones proporciona el marco para gestionar y proteger las tecnologías digitales y los datos de la organización. Al definir las políticas y los controles que se utilizan para gestionar y proteger las aplicaciones y los datos de la arquitectura, ayuda a la organización a proteger sus activos y a garantizar el cumplimiento de los reglamentos y normas pertinentes.

En general, la arquitectura de aplicaciones puede aportar valor a la transformación digital de una empresa, proporcionando la base para construir y desplegar tecnologías y servicios digitales, permitiendo la integración y alineación de estas tecnologías con el negocio, y proporcionando un marco para gestionar y asegurar los activos digitales de la organización.



Actividades de aprendizaje recomendadas

Le invito a realizar las siguientes actividades de aprendizaje recomendadas para esta semana:

- A. Crear un inventario de componentes de TI y relacionarlo con las aplicaciones utilizadas en la organización. Este proceso permitirá al estudiante comprender cómo los diferentes componentes de TI se utilizan en conjunto para respaldar la operación de los sistemas y aplicaciones empresariales.

Con base a lo estudiado analice los componentes de TI que tiene su arquitectura, genere un inventario y relacionelo con las aplicaciones, estos componentes permiten la operación de los sistemas como plataformas de *middleware*.

Relacionar los componentes de TI con las aplicaciones implica identificar todos los componentes de TI utilizados por las aplicaciones de la organización, así como las dependencias y relaciones entre ellos, se pueden seguir estos pasos:

- **Identificar los componentes de TI:** La primera tarea consiste en identificar todos los componentes de TI utilizados por las aplicaciones de la organización. Esto incluye tanto los componentes de *hardware* como los de *software*, como servidores, bases de datos, sistemas operativos, *middleware*, aplicaciones y servicios *web*.
- **Documentar las dependencias y relaciones:** Una vez que se han identificado los componentes de TI, es necesario documentar las dependencias y relaciones entre ellos. Por ejemplo, un servidor de aplicaciones puede depender de una base de datos para su funcionamiento, o una aplicación *web* puede utilizar un servicio *web* externo para acceder a datos.
- **Mapear los componentes con las aplicaciones:** El siguiente paso es mapear los componentes de TI identificados con las aplicaciones que los utilizan. Para ello, es necesario revisar el

inventario de aplicaciones de la organización y determinar qué componentes de TI son necesarios para cada aplicación.

- **Identificar posibles redundancias:** durante el proceso de mapeo de componentes con aplicaciones, es posible identificar redundancias o componentes que no son necesarios. Es importante identificar estas redundancias para reducir costos y simplificar la arquitectura de aplicaciones.
- **Actualizar el inventario de aplicaciones:** finalmente, es necesario actualizar el inventario de aplicaciones de la organización para reflejar los componentes de TI identificados y sus dependencias y relaciones. Esto ayuda a tener una visión completa de la arquitectura de aplicaciones de la organización y facilita la toma de decisiones en cuanto a la gestión de los componentes de TI.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

Esta actividad requerirá que el estudiante dedique una cantidad significativa de tiempo y esfuerzo para recopilar y analizar información sobre las aplicaciones empresariales y los componentes de TI utilizados en la organización, así como para documentar las dependencias y relaciones entre ellos. Sin embargo, este proceso permitirá al estudiante comprender mejor cómo se utilizan los diferentes componentes de TI en conjunto para respaldar las operaciones empresariales y ayudará a la organización a mejorar la gestión de sus sistemas de información.

- B. Adicionalmente, ¿cómo se deberían diagramar los componentes de TI para la exposición de API de manera segura? Elabore un diagrama con los componentes necesarios.
- C. Estimado estudiante, ahora para consolidar el conocimiento realice la siguiente autoevaluación:



Autoevaluación 2

Seleccione la respuesta que considere correcta, se debe considerar la mejor opción con base en la justificación.

1. ¿Qué es la arquitectura de microservicios?
 - a. Una arquitectura que utiliza grandes bloques de código monolíticos.
 - b. Una arquitectura que utiliza pequeños servicios independientes y conectados entre sí.
 - c. Una arquitectura que no utiliza ningún servicio y se enfoca solo en la funcionalidad de una sola aplicación.
2. ¿Qué es REST?
 - a. Un protocolo de comunicación basado en mensajes XML.
 - b. Un protocolo de comunicación basado en mensajes JSON.
 - c. Un protocolo de comunicación que utiliza una arquitectura cliente-servidor y opera a través de peticiones HTTP.
3. ¿Qué es ETL?
 - a. Una arquitectura de microservicios.
 - b. Un protocolo de comunicación utilizado para transferir datos entre sistemas.
 - c. Un proceso utilizado para extraer, transformar y cargar datos desde una fuente de datos a un destino.
4. ¿Qué es Clean Code?
 - a. Un conjunto de buenas prácticas para escribir código limpio y legible.
 - b. Un protocolo de comunicación utilizado para intercambiar datos entre sistemas.
 - c. Una arquitectura de aplicaciones basada en microservicios.

5. ¿Qué es la arquitectura de datos?
 - a. Una arquitectura que se enfoca solo en la capa de presentación de una aplicación.
 - b. Una arquitectura que se enfoca solo en la capa de lógica de una aplicación.
 - c. Una arquitectura que se enfoca en la gestión y organización de los datos de una aplicación.
6. ¿Qué protocolo se utiliza comúnmente en arquitecturas de microservicios para la comunicación entre servicios?
 - a. SOAP.
 - b. REST.
 - c. EDA.
7. ¿Cuál de los siguientes es un enfoque arquitectónico para el desarrollo de sistemas empresariales que se centra en el uso de servicios como componentes fundamentales?
 - a. EDA.
 - b. SOA.
 - c. Clean Code.
8. ¿Qué enfoque de arquitectura de software se centra en el diseño de interfaces de comunicación claramente definidas y contratos bien definidos entre servicios?
 - a. Websockets.
 - b. REST.
 - c. gRPC.
9. ¿Qué tecnología se utiliza comúnmente para la integración de datos en sistemas empresariales?
 - a. ETL.
 - b. SSR.
 - c. CSR.

10. ¿Cuál de los siguientes enfoques se utiliza comúnmente en arquitecturas de aplicaciones para mejorar el rendimiento y la experiencia del usuario?
- a. SSR (Server-Side Rendering).
 - b. CSR (Client-Side Rendering).
 - c. REST (Representational State Transfer).



Nota: se pueden encontrar las respuestas a estas preguntas en la sección de solucionario, en la parte de autoevaluación 2.

[Ir a solucionario](#)



Actividades finales del bimestre

¡Excelente! Prepararse adecuadamente para el primer parcial del curso de arquitectura de aplicaciones es fundamental para asegurar un buen rendimiento. Estos son algunos motivadores para ayudar al estudiante a prepararse en los siguientes temas:

1. **TOGAF - Sistemas de información:** TOGAF es un marco de trabajo ampliamente utilizado en la industria para el diseño y gestión de la arquitectura empresarial. Prepararse en este tema permitirá al estudiante comprender los procesos, herramientas y técnicas necesarios para diseñar y gestionar sistemas de información empresariales de manera eficiente y efectiva.
2. **Metamodelo - TOGAF:** el metamodelo es una parte fundamental de TOGAF y ayuda a describir y definir la arquitectura empresarial. Prepararse en este tema permitirá al estudiante comprender cómo se estructura y se define la arquitectura empresarial en TOGAF.
3. **Arquitectura de aplicaciones empresariales:** la arquitectura de aplicaciones empresariales es una parte importante de la arquitectura empresarial y se enfoca en el diseño y gestión de las aplicaciones empresariales. Prepararse en este tema permitirá al estudiante comprender cómo se diseñan y gestionan las aplicaciones empresariales para lograr los objetivos empresariales.
4. **Entregables de arquitectura de aplicaciones:** los entregables de la arquitectura de aplicaciones son los productos y artefactos que se producen durante el proceso de diseño y gestión de las aplicaciones empresariales. Prepararse en este tema permitirá al estudiante comprender qué entregables son necesarios y cómo se producen para diseñar y gestionar las aplicaciones empresariales de manera efectiva.
5. **Diseño de arquitectura de aplicaciones:** el diseño de la arquitectura de aplicaciones se enfoca en la creación de soluciones de aplicaciones que cumplan con los requisitos empresariales. Prepararse en este

tema permitirá al estudiante comprender cómo se diseña y desarrolla la arquitectura de aplicaciones para garantizar que cumpla con los requisitos empresariales y de los usuarios.

6. **Arquitecturas de integración:** las arquitecturas de integración son una parte importante de la arquitectura empresarial y se enfocan en la integración de sistemas y aplicaciones empresariales. Prepararse en este tema permitirá al estudiante comprender cómo se integran los sistemas y aplicaciones empresariales para lograr los objetivos empresariales.
7. **Componentes lógicos y físicos:** los componentes lógicos y físicos son los elementos de *hardware* y *software* que se utilizan para diseñar y gestionar sistemas y aplicaciones empresariales. Prepararse en este tema permitirá al estudiante comprender cómo se diseñan y gestionan los componentes lógicos y físicos para lograr los objetivos empresariales.

Prepararse adecuadamente en estos temas permitirá al estudiante comprender los procesos, herramientas y técnicas necesarios para diseñar y gestionar sistemas y aplicaciones empresariales de manera eficiente y efectiva, lo que será fundamental para el éxito en el primer parcial del curso de arquitectura de aplicaciones.

La presente semana tiene como actividad calificada la evaluación bimestral, que consiste en un examen con preguntas objetivas que abarcan los contenidos de todo el bimestre. Prepare con anticipación su examen que consistirá en la aplicación tanto de sus habilidades de comprensión teórica como de la aplicación de ejemplos y casos de razonamiento.

Le sugiero (Insider, 2020), que siga esta estrategia para su preparación:

1. Revise los contenidos de cada uno de los temas tratados en el bimestre.
2. Revise las tablas, ejemplos prácticos y casos propuestos.
3. Realice la autoevaluación al finalizar la unidad para comprender cuáles son los conceptos que debe reforzar en su preparación.
4. Revise los apuntes de su bitácora para validar sus resúmenes y los puntos de refuerzo que ha considerado para cada semana.

¡Qué tenga mucho éxito en el desarrollo de su evaluación bimestral!



Segundo bimestre

Resultado de aprendizaje 2

- Comprende el proceso de definición de la hoja de ruta para implementación de la arquitectura de aplicaciones integradas a las arquitecturas de negocio y datos.

El presente resultado de aprendizaje es fundamental para la definición de una hoja de ruta, la implementación de la arquitectura de aplicaciones integradas a las arquitecturas de negocio y datos. La hoja de ruta es un plan detallado que describe los pasos necesarios para implementar la arquitectura de aplicaciones de la organización, y que tiene en cuenta factores como los requisitos de negocio, las restricciones tecnológicas y los recursos disponibles.

Contenidos, recursos y actividades de aprendizaje



Semana 9

En la presente unidad analizaremos los documentos arquitectónicos que nos permitirán desarrollar nuestra arquitectura de aplicaciones, evaluaremos qué conceptos, componentes, o elementos forman parte de cada documento arquitectónico y cómo estos se convierten en elementos arquitectónicos, que posteriormente serán difundidos en toda la organización.

Para desarrollar nuestra arquitectura, es importante que comprendamos con claridad todas las herramientas requeridas y las técnicas para poder utilizarlas y desarrollarlas, puesto que como se ha venido aclarando, la arquitectura es un proceso que requiere el trabajo conjunto tanto de los arquitectos como de los equipos de la empresa, por lo tanto, manejar un estándar común de diseño de documentación arquitectónica será de mucha ayuda. Los invitamos a revisar todos los temas de la unidad en la que empezaremos a generar y formalizar nuestra arquitectura de aplicaciones a medida.

Unidad 3. Arquitectura objetivo

3.1. Arquitectura de objetivo o arquitectura TO-BE

Una arquitectura objetivo o arquitectura TO-BE es una representación de cómo se desea que sea la arquitectura de una organización en el futuro. Esta arquitectura se utiliza como una guía para el diseño y la implementación de la arquitectura actual (llamada arquitectura AS-IS) con el objetivo de lograr la arquitectura deseada.

La arquitectura objetivo se puede utilizar para abordar una variedad de objetivos, como mejorar la eficiencia de la arquitectura, hacerla más escalable, reducir la complejidad o mejorar la seguridad.

En general, la arquitectura objetivo se establece como parte de un proceso de planificación de la arquitectura que incluye la evaluación de la arquitectura actual, la definición de objetivos y requisitos y el diseño de la arquitectura futura. Una vez que se ha definido la arquitectura objetivo, se pueden establecer planes de implementación y medir el progreso hacia la arquitectura deseada.

Una arquitectura objetivo es una descripción detallada del estado futuro deseado de una organización, en términos de sus metas y objetivos empresariales, sus componentes y atributos clave, y las relaciones entre esos componentes. Una arquitectura objetivo suele incluir una visión o declaración de misión de alto nivel que esboza la dirección y el propósito generales de la organización, así como un plan detallado que describe los componentes y atributos específicos de la organización y cómo encajan entre sí para apoyar las metas y objetivos empresariales de la organización.

El equipo de arquitectura empresarial suele desarrollar una arquitectura objetivo, en colaboración con otras partes interesadas clave, como los líderes empresariales, los directores de TI y los directores de proyecto. La arquitectura objetivo suele basarse en la arquitectura empresarial, que proporciona un modelo para alinear las estrategias de negocio y de TI de la organización.

El objetivo principal de una arquitectura objetivo es proporcionar una visión clara y concisa del estado futuro deseado de la organización y

servir de modelo para guiar las iniciativas, proyectos y actividades de la organización. Una arquitectura objetivo también puede ayudar a coordinar y alinear los esfuerzos de los distintos equipos y partes interesadas, y a garantizar que los recursos de la organización se utilizan de forma eficaz y eficiente.

En general, una arquitectura de objetivos es una herramienta importante para planificar, coordinar y aplicar los cambios necesarios para respaldar las metas y objetivos empresariales de la organización y permitirle adaptarse y responder a los cambios en su entorno empresarial.

1. **Identifica tus objetivos:** antes de comenzar a diseñar la arquitectura objetivo, es importante tener una buena comprensión de los objetivos que quieras alcanzar con la arquitectura. Estos objetivos pueden incluir mejorar la eficiencia, reducir la complejidad, mejorar la seguridad o hacer la arquitectura más escalable.
2. **Evalúa la arquitectura actual:** un paso clave en el diseño de la arquitectura objetivo es evaluar la arquitectura actual (llamada arquitectura AS-IS) para entender sus fortalezas y debilidades. Esto puede incluir el análisis de la estructura y los componentes de la arquitectura, así como la evaluación de cómo se utilizan y cómo se integran entre sí.
3. **Define tus requisitos:** una vez que tienes una buena comprensión de tus objetivos y de la arquitectura actual, es importante definir los requisitos que deben cumplirse para alcanzar la arquitectura objetivo. Estos requisitos deben estar basados en tus objetivos y deben tener en cuenta las limitaciones y restricciones existentes.
4. **Diseña la arquitectura objetivo:** con base en tus objetivos y requisitos, puedes comenzar a diseñar la arquitectura objetivo. Esto incluye la definición de la estructura y los componentes de la arquitectura, así como la forma en que estos componentes interactúan entre sí.
5. **Establece un plan de implementación:** una vez que tienes un diseño de la arquitectura objetivo, es importante establecer un plan de implementación detallado para llevar a cabo la transición de la arquitectura actual a la arquitectura objetivo. Este plan debe incluir pasos específicos para implementar cada componente de

la arquitectura objetivo, así como una estimación del tiempo y los recursos necesarios para llevar a cabo la implementación.

6. También es importante tener en cuenta que la implementación de la arquitectura objetivo puede llevar tiempo y puede requerir la realización de cambios en la arquitectura actual. Por lo tanto, es importante ser *flexible* y estar *preparado para ajustar el plan de implementación según sea necesario*.
7. Por último, es importante *medir el progreso hacia la arquitectura objetivo y ajustar el plan de implementación según sea necesario* para asegurar que se alcanzan los objetivos deseados.

Existen varios componentes que son importantes a considerar al diseñar una arquitectura TO-BE:

1. **Componentes de la arquitectura actual:** es importante tener una buena comprensión de los componentes de la arquitectura actual y cómo se integran entre sí para poder diseñar una arquitectura objetivo que tenga en cuenta estos componentes existentes.
2. **Requisitos de la organización:** los requisitos de la organización deben ser una guía para el diseño de la arquitectura objetivo. Estos requisitos pueden incluir consideraciones como la escalabilidad, la seguridad, la disponibilidad y la eficiencia.
3. **Integración con sistemas externos:** es importante considerar cómo se integrará la arquitectura objetivo con sistemas externos, como proveedores de servicios o sistemas de clientes.
4. **Redundancia y alta disponibilidad:** es importante considerar cómo se puede diseñar la arquitectura objetivo para garantizar la redundancia y la alta disponibilidad de los componentes críticos.
5. **Seguridad:** la seguridad debe ser una consideración importante en el diseño de la arquitectura objetivo, ya que se deben tomar medidas para proteger los datos y evitar ataques externos.
6. **Escalabilidad:** la escalabilidad es una consideración importante al diseñar una arquitectura objetivo ya que se refiere a la capacidad de la arquitectura para adaptarse y crecer a medida que cambian las

necesidades de la organización. Algunas formas en que se puede abordar la escalabilidad en la arquitectura objetivo incluyen:

- a. Diseñar la arquitectura de manera modular y desacoplada para facilitar la adición o eliminación de componentes según sea necesario.
 - b. Utilizar técnicas de distribución y replicación de datos para aumentar la capacidad de procesamiento y almacenamiento.
 - c. Utilizar tecnologías que permitan la escalabilidad horizontal, como el uso de máquinas virtuales o contenedores.
 - d. Implementar medidas de seguridad y alta disponibilidad para garantizar la disponibilidad de la arquitectura incluso durante el crecimiento.
 - e. Realizar pruebas de rendimiento y monitoreo continuo para identificar cuellos de botella y tomar medidas para solucionarlos.
7. **Flexibilidad:** la arquitectura objetivo debe ser flexible y adaptable para poder evolucionar con el tiempo y satisfacer las necesidades cambiantes de la organización.
8. **Mantenibilidad:** la arquitectura objetivo debe ser fácil de mantener y actualizar, ya que esto puede reducir los costos y mejorar la eficiencia a largo plazo.
9. **Eficiencia:** la arquitectura objetivo debe ser eficiente en términos de tiempo de procesamiento y uso de recursos, ya que esto puede mejorar la eficiencia y reducir los costos.
10. **Experiencia del usuario:** la experiencia del usuario debe ser una consideración importante al diseñar la arquitectura objetivo, ya que esto puede afectar la satisfacción del usuario y la lealtad a la marca.

3.2. Estrategia Cloud 6R

La estrategia 6R *Cloud Strategy* es un enfoque que se utiliza para planificar la implementación de servicios *cloud* en una organización. La estrategia 6R se refiere a seis aspectos clave que se deben considerar al planificar la

adopción de la nube, en la siguiente infografía se detalla cada uno de estos aspectos.

Estrategia Cloud 6R.

La estrategia 6R *Cloud Strategy* se utiliza para ayudar a las organizaciones a planificar la implementación de la nube de manera efectiva y a elegir la opción más adecuada en función de sus necesidades y objetivos.

En la estrategia 6R de la arquitectura *cloud*, la complejidad de aplicar cada R puede variar dependiendo de la situación particular de cada empresa (ver figura 7). Sin embargo, en términos generales, se podría decir que la opción “*Rehosting*” (mover) es la más fácil de aplicar, mientras que la opción “*Refactoring*” (reconstrucción) es la más compleja.

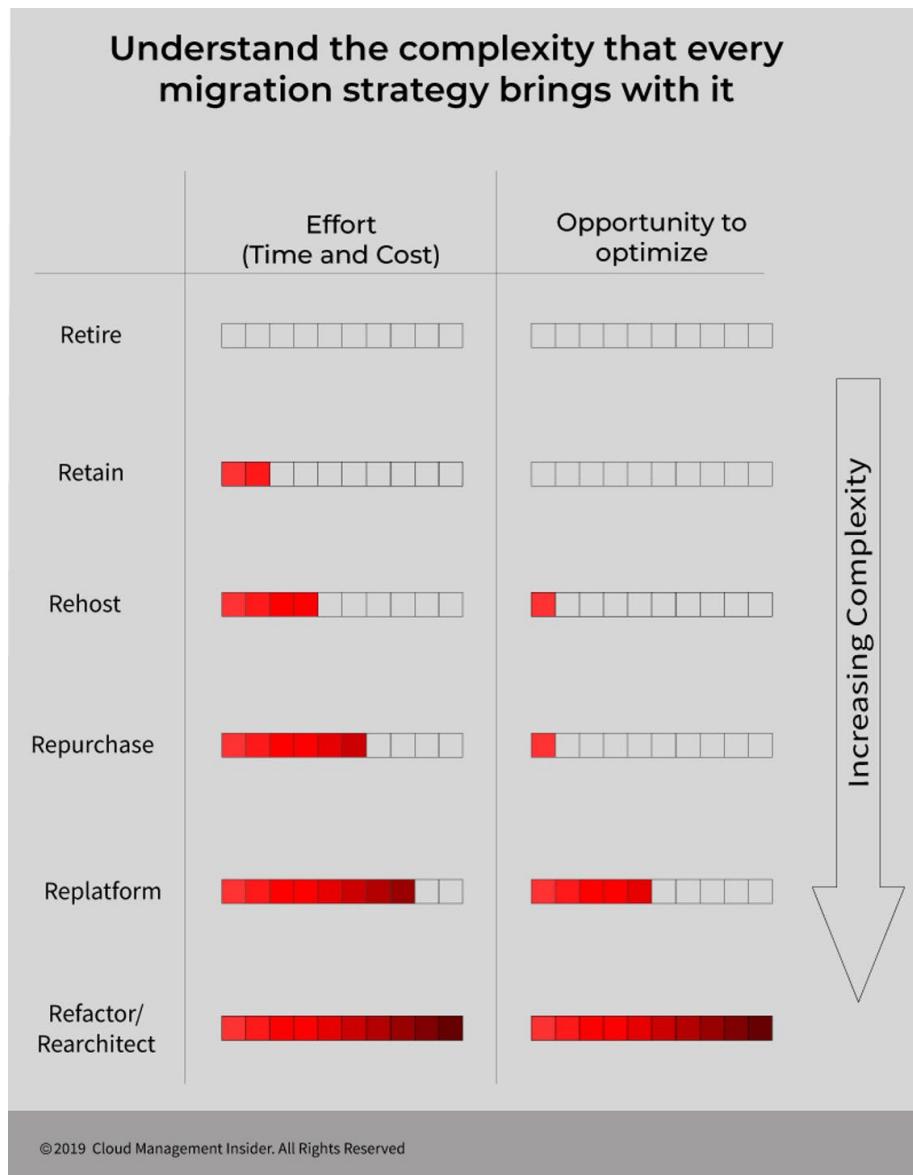
La opción de “*Rehosting*” implica la migración de aplicaciones y sistemas existentes a la nube, sin realizar ningún cambio significativo en la arquitectura. Esta opción es relativamente fácil de aplicar, ya que no requiere grandes cambios en la arquitectura existente, y puede ser una buena opción para empresas que buscan una migración rápida y simple.

Por otro lado, la opción de “*Refactoring*” implica la reconstrucción de las aplicaciones y sistemas existentes para aprovechar las capacidades de la nube y mejorar la escalabilidad, rendimiento y disponibilidad. Esta opción puede ser la más compleja de aplicar, ya que requiere cambios significativos en la arquitectura existente y puede requerir una reescritura significativa del código fuente. Sin embargo, a largo plazo, esta opción puede ser la más beneficiosa para las empresas que buscan aprovechar plenamente las capacidades de la nube y mejorar la eficiencia y el rendimiento de sus aplicaciones.

En resumen, la complejidad de aplicar cada opción de la estrategia 6R en la arquitectura *cloud* puede variar según la situación particular de cada empresa. Sin embargo, en general, la opción de “*Rehosting*” puede ser la más fácil de aplicar, mientras que la opción de “*Refactoring*” puede ser la más compleja debido a los cambios significativos que requiere en la arquitectura existente.

Figura 7.

Complejidad de la Estrategia 6R



Nota. Adaptado de “The 6R Strategy” to Successfully Migrate Applications on Cloud, por Cloud Evangelist, 2019, Cloud Management Insider



Actividad de aprendizaje recomendada

La actividad de aprendizaje recomendada para esta semana es diseñar la arquitectura Objetivo o TO-BE a la que quiere llegar la empresa a la cual está apoyando, para ello se recomienda seguir los siguientes pasos:

1. **Comprender los objetivos empresariales:** el estudiante debe comprender los objetivos empresariales y estrategias de la organización a la que está apoyando. Esto permitirá al estudiante alinear la arquitectura con los objetivos de negocio.
2. **Identificar los requisitos y necesidades empresariales:** el estudiante debe identificar los requisitos y necesidades empresariales actuales y futuras de la organización. Esto incluye los requisitos funcionales y no funcionales, los riesgos y las limitaciones.
3. **Definir los componentes lógicos y físicos de la arquitectura:** el estudiante debe definir los componentes lógicos y físicos que soportarán la arquitectura Objetivo o TO-BE. Esto incluye los servidores, aplicaciones, bases de datos, redes, seguridad, etc.
4. **Seleccionar los recursos de Internet adecuados:** el estudiante debe utilizar recursos de Internet relevantes y actualizados para apoyar en el diseño de la arquitectura. Por ejemplo, puede utilizar sitios web de proveedores de soluciones *cloud*, foros de discusión, *blogs* de expertos en arquitectura empresarial, etc.
5. **Crear un modelo visual de la arquitectura:** el estudiante debe crear un modelo visual de la arquitectura utilizando herramientas de diseño adecuadas. Esto permitirá visualizar la arquitectura Objetivo o TO-BE y hacer ajustes según sea necesario.
6. **Validar y refinar el diseño:** una vez que se ha creado el modelo visual de la arquitectura, el estudiante debe validar y refinar el diseño. Esto incluye revisar la arquitectura con los *stakeholders*, identificar y solucionar cualquier problema o limitación, y ajustar la arquitectura según sea necesario.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

La actividad de aprendizaje recomendada para el estudiante que debe diseñar la arquitectura Objetivo o TO-BE es comprender los objetivos empresariales, identificar los requisitos y necesidades empresariales, definir los componentes lógicos y físicos de la arquitectura, seleccionar los recursos de *Internet* adecuados, crear un modelo visual de la arquitectura, y validar y refinar el diseño. Al seguir estos pasos, el estudiante podrá diseñar una arquitectura empresarial que sea efectiva y cumpla con los objetivos y requisitos de la organización.



Semana 10

3.3. Gap análisis

El análisis de carencias o de faltantes es una técnica que se utiliza para comparar el estado actual de una organización, sistema o proceso con su estado objetivo, con el fin de identificar las carencias o deficiencias que deben subsanarse. El análisis de carencias suele constar de los siguientes pasos:

1. **Identificar el estado actual:** se trata de identificar las características, componentes y atributos clave de la organización, el sistema o el proceso en su estado actual. Puede incluir un análisis detallado de la estructura, los procesos, los productos, los servicios, la información y la tecnología de la organización.
2. **Identificar el estado objetivo:** se trata de identificar las características, componentes y atributos deseados de la organización, sistema o proceso en su estado objetivo. Esto puede incluir una descripción detallada del estado futuro deseado de la organización, así como las metas y objetivos específicos que la organización está tratando de alcanzar.
3. **Comparar los estados actual y objetivo:** se trata de comparar los estados actual y objetivo para identificar las carencias o deficiencias que deben abordarse. Esto puede incluir la identificación de cualquier diferencia entre los estados actual y objetivo, así como cualquier componente o atributo que falte o sea inadecuado en el estado actual.

4. **Desarrollar un plan para colmar las lagunas:** se trata de desarrollar un plan para colmar las lagunas o deficiencias detectadas, con el fin de que la organización, el sistema o el proceso pasen de su estado actual a su estado objetivo. Este plan puede incluir iniciativas, proyectos y actividades específicos necesarios para subsanar las deficiencias detectadas, así como plazos, presupuestos, recursos e Indicadores Clave de Rendimiento (KPI) para medir los avances.

En general, el análisis de carencias es una técnica útil para identificar áreas de mejora en una organización, sistema o proceso, y para desarrollar un plan para cerrar esas carencias con el fin de alcanzar el estado futuro deseado de la organización. Esto puede ayudar a garantizar que la organización, el sistema o el proceso estén alineados con las metas y objetivos empresariales de la organización, y a permitirle adaptarse y responder a los cambios de su entorno.

Un análisis de carencias es un proceso que se utiliza para identificar las diferencias entre el estado actual de la arquitectura de una aplicación y su estado deseado. Esto puede ayudar a las organizaciones a identificar áreas de mejora y a desarrollar un plan para abordar cualquier laguna o deficiencia en la arquitectura.

Para llevar a cabo un análisis de carencias de la arquitectura de una aplicación, se pueden seguir los siguientes pasos:

1. **Definir el alcance del análisis:** comience por definir el alcance del análisis, incluidos los componentes y aspectos específicos de la arquitectura que se evaluarán. Esto le ayudará a centrar el análisis en las áreas más importantes y evitará perder tiempo y recursos en aspectos menos relevantes de la arquitectura.
2. **Identifique el estado actual de la arquitectura:** a continuación, identifique el estado actual de la arquitectura, incluidos sus componentes, relaciones y métricas de rendimiento. Esto puede hacerse revisando la documentación y las especificaciones de la arquitectura, así como realizando entrevistas y encuestas a las partes interesadas que utilizan o gestionan la arquitectura.
3. **Identificar el estado deseado de la arquitectura:** tras identificar el estado actual de la arquitectura, determine el estado deseado de la

misma. Esto debe incluir las metas y objetivos de la organización, así como los requisitos específicos y las expectativas de la arquitectura.

4. **Compare los estados actual y deseado:** una vez identificados los estados actual y deseado de la arquitectura, compárelos para identificar las lagunas y deficiencias. Esto puede hacerse analizando las diferencias entre los dos estados y evaluando el impacto de las lagunas en el rendimiento y la eficacia de la arquitectura.
5. **Elabore un plan para subsanar las deficiencias:** por último, elabore un plan para subsanar las carencias y deficiencias detectadas en el análisis. En él deben figurar las medidas y pasos concretos que se adoptarán para colmar las lagunas y mejorar la arquitectura, así como los recursos y plazos necesarios para aplicar el plan.

En general, un análisis de carencias es una herramienta útil para identificar las diferencias entre el estado actual y el deseado de la arquitectura de una aplicación, y para desarrollar un plan que aborde cualquier carencia o deficiencia. Al realizar un análisis de carencias, las organizaciones pueden identificar áreas de mejora y desarrollar una hoja de ruta para mejorar y optimizar su arquitectura de aplicaciones.

Para diseñar el estado actual de una arquitectura, se pueden seguir los siguientes pasos:

1. **Definir el alcance de la arquitectura:** comience por definir el alcance de la arquitectura, incluyendo los componentes y aspectos específicos de la organización que se incluirán en el diseño. Esto le ayudará a centrar el diseño en los aspectos más relevantes e importantes de la organización, y a evitar perder tiempo y recursos en áreas menos relevantes.
2. **Identifique las metas y objetivos empresariales:** a continuación, identifique las metas y objetivos empresariales de la organización y utilícelos para guiar el diseño de la arquitectura. Esto ayudará a garantizar que la arquitectura esté alineada con las metas y objetivos de la organización, y que respalte las operaciones empresariales y la dirección estratégica de la organización.
3. **Identificar a las partes interesadas y sus requisitos:** identifique a las partes interesadas que se verán afectadas por la arquitectura

y comprenda sus requisitos y expectativas. Para ello, se pueden realizar entrevistas, encuestas o grupos de discusión con las partes interesadas para recabar sus opiniones y comentarios.

4. **Desarrollar un diseño de alto nivel:** basándose en las metas, objetivos y requisitos de las partes interesadas, desarrolle un diseño de alto nivel para la arquitectura. Debe incluir los principales componentes de la arquitectura, así como sus relaciones e interacciones.
5. **Refinar el diseño:** una vez desarrollado el diseño de alto nivel, hay que perfeccionarlo y ampliarlo para incluir especificaciones y requisitos más detallados. Esto puede implicar la realización de investigaciones y análisis adicionales, o la colaboración con otros equipos o expertos para desarrollar un diseño más completo y detallado.

En general, diseñar el estado actual de una arquitectura implica definir el alcance de la arquitectura, identificar las metas y objetivos empresariales, recabar información de las partes interesadas y desarrollar y perfeccionar el diseño. Siguiendo estos pasos, las organizaciones pueden desarrollar un diseño eficaz y completo para su arquitectura que se ajuste a las metas y objetivos de la organización y satisfaga las necesidades y expectativas de las partes interesadas.

3.4. Objetivos empresariales

Las metas y los objetivos empresariales son las metas que una empresa se propone alcanzar. Estas metas y objetivos deben ser específicos, medibles, alcanzables, pertinentes y de duración determinada, lo que se conoce comúnmente como metas SMART. Para definir las metas y los objetivos empresariales de una empresa, empiece por identificar la misión y la visión de la empresa. Esto le ayudará a comprender la dirección general y el propósito de la empresa. A partir de ahí, puede hacer una lluvia de ideas sobre posibles metas y objetivos que estén en consonancia con la misión y la visión de la empresa. Estas metas y objetivos deben ser específicos, es decir, deben describir claramente lo que la empresa espera conseguir. También deben ser mensurables, es decir, debe haber una forma de seguir el progreso hacia su consecución. Además, las metas y objetivos deben ser alcanzables, es decir, deben ser realistas teniendo en cuenta los recursos y capacidades actuales de la empresa. También deben ser pertinentes, es decir, deben estar en consonancia con la dirección y el propósito generales

de la empresa. Por último, deben tener una duración determinada, es decir, un plazo concreto para su consecución.

3.5. Diseño de alto nivel de arquitectura

Para desarrollar un diseño de alto nivel de la arquitectura de una aplicación, siga estos pasos:

1. Identifique las principales metas y objetivos de la aplicación, así como las principales partes interesadas que la utilizarán.
2. Definir los principales requisitos funcionales y no funcionales de la aplicación, como el rendimiento, la escalabilidad, la seguridad y la usabilidad.
3. Identificar los principales componentes de la aplicación, como la interfaz de usuario, la lógica de negocio, el almacenamiento de datos y cualquier sistema o servicio externo con el que deba interactuar.
4. Desarrollar un diagrama o modelo de alto nivel de la arquitectura de la aplicación, que muestre los componentes principales y cómo interactuarán entre sí.
5. Definir las principales tecnologías y marcos de trabajo que se utilizarán para implementar la aplicación, como lenguajes de programación, bases de datos y *middleware*.
6. Desarrollar un plan de alto nivel para el proceso de desarrollo de la aplicación, incluyendo los principales hitos y entregables.
7. Revisar el diseño de alto nivel con las partes interesadas y realizar las revisiones necesarias en función de sus comentarios.
8. Utilice el diseño de alto nivel como guía para desarrollar el diseño detallado y la implementación de la aplicación.

Siguiendo estos pasos, puede desarrollar un diseño de alto nivel de la arquitectura de una aplicación que defina claramente los componentes principales, las interacciones y las tecnologías utilizadas en la aplicación, así como el proceso de desarrollo general. Esto puede ayudar a garantizar

que la aplicación esté bien diseñada, cumpla los requisitos de las partes interesadas y pueda desarrollarse y mantenerse de forma eficiente.



Actividad de aprendizaje recomendada

Como actividad de aprendizaje recomendada para esta semana debe realizar el Gap Análisis de los proyectos necesarios para implementar la arquitectura TO-BE, para lograr este objetivo cubrir lo siguiente:

1. **Comprender los objetivos empresariales:** el estudiante debe comprender los objetivos empresariales de la organización y la estrategia que debería seguir para lograrlos. Esto permitirá al estudiante identificar las áreas de enfoque para el Gap Análisis.
2. **Identificar los proyectos necesarios:** el estudiante debe identificar los proyectos necesarios para implementar la arquitectura TO-BE. Esto puede incluir la adopción de tecnologías *cloud*, la actualización de sistemas existentes, la integración de aplicaciones, la mejora de la seguridad, entre otros.
3. **Evaluar los proyectos actuales:** el estudiante debe evaluar los proyectos actuales de la organización y determinar si estos proyectos son suficientes para alcanzar los objetivos empresariales. Si no es así, se deben identificar los proyectos adicionales que se necesitan.
4. **Identificar las brechas:** el estudiante debe identificar las brechas entre los proyectos actuales y los proyectos necesarios para implementar la arquitectura TO-BE. Esto permitirá al estudiante determinar cuáles son los proyectos críticos y cuáles pueden ser postergados.
5. **Priorizar los proyectos:** el estudiante debe priorizar los proyectos identificados en función de su importancia para lograr los objetivos empresariales y la estrategia de la organización. Esto ayudará a la organización a enfocar sus recursos en los proyectos más críticos.
6. **Crear un plan de acción:** el estudiante debe crear un plan de acción detallado para cada proyecto identificado. El plan de acción debe

incluir los objetivos del proyecto, los recursos necesarios, el calendario y los indicadores clave de desempeño.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

En resumen, la actividad de aprendizaje recomendada es comprender los objetivos empresariales y la estrategia de la organización, identificar los proyectos necesarios, evaluar los proyectos actuales, identificar las brechas, priorizar los proyectos, y crear un plan de acción detallado para cada proyecto identificado. Al seguir estos pasos, el estudiante podrá identificar los proyectos críticos para implementar la arquitectura TO-BE y crear un plan de acción efectivo para lograr los objetivos empresariales de la organización.



Semana 11

3.6. Riesgos e impactos (gestión de riesgos)

Los riesgos en una arquitectura empresarial pueden gestionarse mediante el uso de un proceso formal de gestión de riesgos que identifique, evalúe y mitigue los riesgos para la arquitectura empresarial. Este proceso suele incluir los siguientes pasos.

- Identificar los riesgos:** esto implica identificar los riesgos específicos que están asociados con la arquitectura empresarial, incluidos los riesgos para la disponibilidad, integridad y confidencialidad de los activos de información de la empresa, así como los riesgos para el rendimiento, la fiabilidad y la escalabilidad de los sistemas de TI de la empresa.
- Evaluar los riesgos:** esto implica evaluar el impacto potencial y la probabilidad de cada riesgo identificado, y determinar el nivel general de riesgo para cada uno. Esto puede ayudar a priorizar los riesgos y a centrar los esfuerzos de gestión de riesgos en los riesgos que son más significativos.
- Mitigar los riesgos:** se trata de aplicar estrategias y controles para reducir la probabilidad o el impacto de cada riesgo identificado. Esto

puede incluir la aplicación de controles técnicos, como medidas de seguridad y sistemas de supervisión, así como controles operativos, como políticas, procedimientos y formación.

4. **Supervisar y revisar los riesgos:** esto implica supervisar la arquitectura de la empresa y el proceso de gestión de riesgos de forma continua, y revisar y actualizar el plan de gestión de riesgos según sea necesario para tener en cuenta los cambios en la arquitectura de la empresa o en el entorno empresarial.

En general, la gestión de riesgos en una arquitectura empresarial implica identificar, evaluar y mitigar los riesgos para los activos de información y los sistemas de TI de la empresa, y aplicar procesos de supervisión y revisión continuos para garantizar que el plan de gestión de riesgos siga siendo eficaz a lo largo del tiempo. Esto puede ayudar a proteger la arquitectura de la empresa de posibles amenazas y vulnerabilidades, y garantizar que siga apoyando el negocio de la empresa.

3.7. Gestión de riesgos

Un proceso formal de gestión de riesgos es un enfoque estructurado y sistemático para identificar, evaluar y mitigar los riesgos para los activos de información y los sistemas de TI de una organización. Un proceso formal de gestión de riesgos suele incluir los siguientes pasos:

1. **Identificar los riesgos:** esto implica identificar los riesgos específicos que están asociados con los activos de información y los sistemas de TI de la organización, incluidos los riesgos para la disponibilidad, integridad y confidencialidad de los activos de información, así como los riesgos para el rendimiento, la fiabilidad y la escalabilidad de los sistemas de TI.
2. **Evaluar los riesgos:** esto implica evaluar el impacto potencial y la probabilidad de cada riesgo identificado, y determinar el nivel general de riesgo para cada uno. Esto puede ayudar a priorizar los riesgos y a centrar los esfuerzos de gestión de riesgos en los más significativos.
3. **Mitigar los riesgos:** se trata de aplicar estrategias y controles para reducir la probabilidad o el impacto de cada riesgo identificado. Esto puede incluir la aplicación de controles técnicos, como medidas de

seguridad y sistemas de supervisión, así como controles operativos, como políticas, procedimientos y formación.

4. **Supervisar y revisar los riesgos:** esto implica supervisar los activos de información y los sistemas de TI de la organización, así como el proceso de gestión de riesgos de forma continua, y revisar y actualizar el plan de gestión de riesgos según sea necesario para tener en cuenta los cambios en la organización o en el entorno empresarial.

En general, un proceso formal de gestión de riesgos proporciona un enfoque estructurado y sistemático para identificar, evaluar y mitigar los riesgos para los activos de información y los sistemas de TI de una organización. Esto puede ayudar a proteger a la organización de posibles amenazas y vulnerabilidades, y a garantizar que sus activos de información y sistemas de TI sean seguros, fiables y estén alineados con las metas y objetivos empresariales de la organización.

La gestión de riesgos es un aspecto importante de la arquitectura empresarial, ya que implica identificar, evaluar y mitigar los riesgos y vulnerabilidades potenciales que pueden afectar a los sistemas y la infraestructura de TI de una empresa. Para hacer de la gestión de riesgos una parte clave de la arquitectura empresarial, se pueden seguir los siguientes pasos:

1. Identificar los riesgos y vulnerabilidades potenciales que pueden afectar a la arquitectura empresarial. Esto puede hacerse realizando un análisis exhaustivo de los sistemas y la infraestructura de TI actuales, así como de los procesos y actividades empresariales que dependen de esos sistemas.
2. Evaluar la probabilidad y el impacto de cada riesgo identificado. Esto puede hacerse considerando factores como la probabilidad de que el riesgo ocurra, las consecuencias potenciales del riesgo si ocurriera, y los costes e impactos potenciales del riesgo en el negocio.
3. Priorizar los riesgos en función de su probabilidad e impacto. Esto puede ayudar a centrar los esfuerzos de gestión de riesgos en los más importantes y a asignar los recursos en consecuencia.

4. Desarrollar y aplicar estrategias para mitigar o eliminar los riesgos identificados. Esto puede implicar la implementación de controles técnicos, como cortafuegos y sistemas de detección de intrusos, así como controles organizativos, como políticas y procedimientos, para prevenir o minimizar el impacto potencial de los riesgos.
5. Supervisar y revisar el proceso de gestión de riesgos de forma continua. Esto puede implicar la evaluación periódica de la eficacia de las estrategias de mitigación de riesgos y la realización de los ajustes necesarios para garantizar que la arquitectura de la empresa siga siendo segura y resistente.

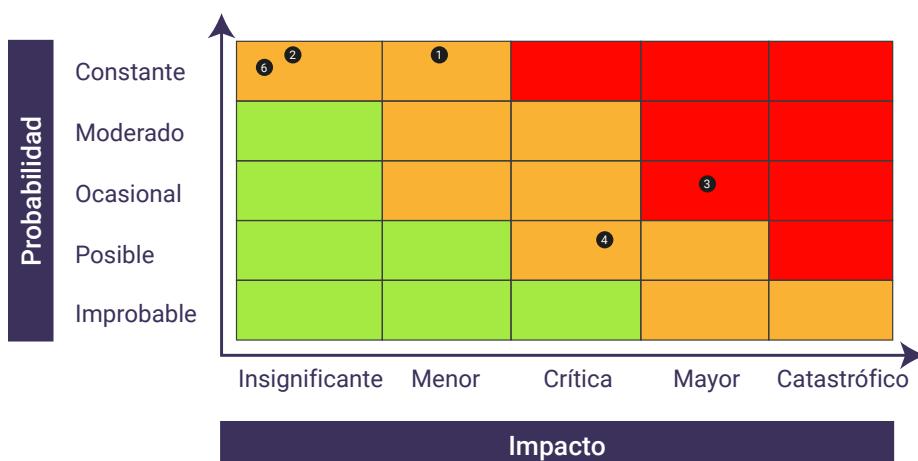
3.7.1. Matriz de riesgos

Una matriz de riesgos es una herramienta utilizada en la gestión de riesgos que proporciona una visión general de los riesgos que pueden afectar un proyecto, proceso o actividad empresarial. La matriz de riesgos es una tabla que muestra la probabilidad de que un riesgo ocurra y su impacto en el proyecto o actividad si llega a ocurrir.

La matriz de riesgos suele tener dos ejes (ver figura 8): uno que indica la probabilidad de que un riesgo ocurra, que puede ser alta, media o baja; y otro que indica el impacto del riesgo, que también puede ser alto, medio o bajo. Con esta información, los riesgos se pueden clasificar en cuatro categorías: riesgos de alto impacto y alta probabilidad, riesgos de alto impacto y baja probabilidad, riesgos de bajo impacto y alta probabilidad, y riesgos de bajo impacto y baja probabilidad.

La matriz de riesgos ayuda a la gestión de riesgos al permitir que se identifiquen y prioricen los riesgos de manera sistemática, de manera que los esfuerzos de gestión de riesgos puedan centrarse en los riesgos más críticos y se puedan tomar medidas para mitigarlos. Al evaluar los riesgos en términos de probabilidad e impacto, la matriz de riesgos también puede ayudar a la toma de decisiones al permitir que se identifiquen los riesgos más críticos y se asignen recursos para mitigarlos de manera efectiva.

Figura 8.
Matriz de Riesgos



Nota. Adaptado de ¿Qué es una matriz de riesgos?, por Gonzal89, 2020, Colombia Fácil

Para generar una matriz de riesgos, sigue estos pasos:

- 1. Identifica los riesgos:** la primera etapa es identificar todos los riesgos potenciales que pueden afectar el proyecto o la organización. Esto puede incluir riesgos de negocio, riesgos técnicos, riesgos de recursos humanos y cualquier otro tipo de riesgo relevante.
- 2. Evalúa el impacto y la probabilidad de cada riesgo:** una vez que has identificado todos los riesgos, es importante evaluar el impacto y la probabilidad de cada uno. El impacto puede medirse en términos de tiempo, costo, calidad o cualquier otro factor relevante para el proyecto. La probabilidad se refiere a la probabilidad de que el riesgo se materialice.
- 3. Asigna una puntuación a cada riesgo:** una vez que has evaluado el impacto y la probabilidad de cada riesgo, puedes asignarle una puntuación combinando estos dos factores. Por ejemplo, puedes asignar una puntuación de alto, medio o bajo para cada factor y luego combinarlos para obtener una puntuación final.
- 4. Clasifica los riesgos:** con base en las puntuaciones asignadas, puedes clasificar los riesgos en categorías como altos, medianos

o bajos. Esto te ayudará a priorizar los riesgos y a determinar qué medidas de mitigación debes implementar primero.

5. **Crea una matriz de riesgos:** con base en la información recopilada, puedes crear una matriz de riesgos que muestre los riesgos identificados, sus puntuaciones y su categoría de prioridad. Esta matriz te ayudará a visualizar y monitorear los riesgos de manera más efectiva.
6. **Establece medidas de mitigación para cada riesgo:** una vez que has clasificado los riesgos, es importante establecer medidas de mitigación para reducir el impacto o la probabilidad de cada uno de ellos. Estas medidas pueden incluir la implementación de planes de contingencia, la adquisición de seguros o la realización de pruebas de validación para minimizar el riesgo.
7. **Monitorea y actualiza la matriz de riesgos:** una vez que has establecido medidas de mitigación para cada riesgo, es importante monitorear y actualizar la matriz de riesgos de manera regular para asegurarte de que los riesgos estén siendo adecuadamente gestionados. Si se producen cambios en el proyecto o en la organización, es importante actualizar la matriz de riesgos para reflejar estos cambios.



Es importante tener en cuenta que la generación de una matriz de riesgos es un proceso continuo y es necesario actualizarla de manera regular para asegurar que los riesgos estén siendo adecuadamente gestionados. Además, es importante involucrar a todas las partes interesadas en el proceso de identificación y evaluación de riesgos para asegurar una visión completa y precisa de los riesgos potenciales.

3.8. Reducir complejidad de aplicaciones

¿Cómo reducir la complejidad de las aplicaciones informáticas?

Hay varias formas de reducir la complejidad de las aplicaciones informáticas, entre ellas las siguientes:

1. **Adoptar un diseño modular y basado en componentes:** al descomponer una aplicación compleja en componentes más pequeños, independientes y reutilizables, es posible reducir la complejidad y mejorar la mantenibilidad de la aplicación.
2. **Utilizar patrones de diseño y estilos arquitectónicos:** aplicando patrones de diseño y estilos arquitectónicos comunes, como la arquitectura de microservicios o la arquitectura dirigida por eventos, es posible reducir la complejidad y mejorar la escalabilidad y fiabilidad de la aplicación.
3. **Utilizar la abstracción y la encapsulación:** mediante el uso de la abstracción y la encapsulación, es posible ocultar la complejidad de la aplicación detrás de una interfaz simple y bien definida, lo que puede hacer que la aplicación sea más fácil de entender y usar.
4. **Utilizar pruebas automatizadas e integración continua:** mediante el uso de pruebas automatizadas e integración continua, es posible reducir la complejidad y mejorar la calidad y fiabilidad de la aplicación mediante la detección y corrección de errores y otros problemas en las primeras etapas del proceso de desarrollo.
5. **Utilizar métodos de desarrollo ágiles:** mediante el uso de métodos de desarrollo ágiles, como Scrum o Kanban, es posible reducir la complejidad y mejorar la adaptabilidad y capacidad de respuesta de la aplicación al permitir que el equipo de desarrollo responda rápidamente a los requisitos cambiantes y a la retroalimentación.

En general, hay muchas formas de reducir la complejidad de las aplicaciones informáticas, y el mejor enfoque dependerá de los requisitos y limitaciones específicos del proyecto. Si se adopta un diseño modular y basado en componentes, se utilizan patrones de diseño y estilos arquitectónicos, se recurre a la abstracción y la encapsulación, se utilizan pruebas automatizadas e integración continua y se emplean métodos de

desarrollo ágiles, es posible reducir la complejidad de las aplicaciones de software y mejorar su mantenimiento, escalabilidad, fiabilidad y adaptabilidad.



Actividades de aprendizaje recomendadas

Le invito a realizar las siguientes actividades de aprendizaje recomendadas en esta semana:

- A. Investigar qué campos puede tener una plantilla para el análisis y la gestión de riesgos de arquitectura, para ello se recomienda:
 1. **Identificar la fuente de información:** el estudiante debe identificar fuentes confiables de información sobre el análisis y la gestión de riesgos de arquitectura. Esto puede incluir libros, artículos, revistas especializadas, sitios web de expertos en gestión de riesgos, entre otros.
 2. **Comprender los conceptos básicos:** el estudiante debe comprender los conceptos básicos de la gestión de riesgos y cómo se aplican en la arquitectura empresarial. Esto incluye la identificación de riesgos, evaluación de riesgos, mitigación de riesgos, y seguimiento de los riesgos.
 3. **Identificar los campos de una plantilla de análisis de riesgos de arquitectura:** el estudiante debe investigar los diferentes campos que pueden tener una plantilla de análisis de riesgos de arquitectura, que pueden incluir:
 - *Identificación del riesgo:* descripción del riesgo y su posible impacto.
 - *Causas del riesgo:* factores que contribuyen al riesgo.
 - *Probabilidad:* la probabilidad de que el riesgo ocurra.
 - *Impacto:* el impacto del riesgo en el proyecto o actividad empresarial.
 - *Severidad:* la gravedad del impacto del riesgo.
 - *Acciones de mitigación:* las medidas que se pueden tomar para mitigar el riesgo.

- *Responsabilidad*: la persona o equipo responsable de implementar las medidas de mitigación.
 - *Estado del riesgo*: el estado actual del riesgo, si se ha mitigado o no.
- 4. Levantar una matriz de riesgos:** una vez que se han identificado los campos de una plantilla de análisis de riesgos de arquitectura, el estudiante debe levantar una matriz de riesgos que identifique los principales riesgos que pueden afectar los proyectos de gap análisis para llegar a la arquitectura objetivo. La matriz debe incluir los campos identificados en la investigación, y debe ser lo suficientemente completa para permitir una evaluación efectiva de los riesgos.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

En resumen, la actividad de aprendizaje recomendada para el estudiante que debe investigar los campos de una plantilla de análisis y gestión de riesgos de arquitectura es identificar fuentes de información confiables, comprender los conceptos básicos de la gestión de riesgos, investigar los campos de una plantilla de análisis de riesgos de arquitectura, y levantar una matriz de riesgos para identificar los principales riesgos que pueden afectar los proyectos de gap análisis que permitirán llegar a la arquitectura objetivo. Al seguir estos pasos, el estudiante podrá diseñar y aplicar una plantilla de análisis de riesgos de arquitectura efectiva para la gestión de proyectos de arquitectura empresarial.

B. Desarrollar la siguiente autoevaluación:



Autoevaluación 3

¡Hola, estudiante!

Quiero motivar a realizar la autoevaluación para reforzar tu conocimiento en el curso que estás llevando. La autoevaluación es una herramienta muy efectiva para identificar tus fortalezas y áreas de oportunidad en el aprendizaje. Además, te permite prepararte mejor para las evaluaciones formales.

Le recomiendo que tome el tiempo necesario para realizar la autoevaluación y que no se preocupe si encuentras algunas áreas en las que necesitas mejorar, ya que esto es parte del proceso de aprendizaje. Utiliza los resultados de la autoevaluación como una guía para enfocar tus esfuerzos de estudio y mejorar tu comprensión del curso.

¡Ánimo! Realizar la autoevaluación te ayudará a consolidar tu aprendizaje y estarás mejor preparado para las evaluaciones formales.

1. ¿Qué son los componentes de TI?

- a. Componentes físicos de una computadora.
- b. Componentes que conforman la infraestructura tecnológica de una organización.
- c. Componentes de software que se utilizan en el desarrollo de aplicaciones.

2. ¿Qué son los contenedores?

- a. Un tipo de servidor de aplicaciones.
- b. Una tecnología de virtualización que permite ejecutar aplicaciones de forma aislada.
- c. Una plataforma para la gestión de API .

3. ¿Qué es la arquitectura objetivo?
 - a. Una arquitectura que describe el estado actual de la organización.
 - b. Una arquitectura que describe el estado futuro deseado de la organización.
 - c. Una arquitectura que describe la evolución de la organización a lo largo del tiempo.
4. ¿Qué es la monetización de API ?
 - a. La práctica de hacer públicas las API de una organización.
 - b. La práctica de generar ingresos a través de las API de una organización.
 - c. La práctica de restringir el acceso a las API de una organización.
5. ¿Qué es el gap análisis y gestión de riesgos en arquitectura de aplicaciones?
 - a. Un proceso para identificar y analizar las brechas entre la arquitectura actual y la arquitectura objetivo.
 - b. Un proceso para identificar y evaluar los riesgos asociados con la implementación de soluciones de arquitectura.
 - c. Un proceso para definir los requisitos de seguridad de una aplicación.
6. ¿Cuál es la mejor estrategia para garantizar la disponibilidad y escalabilidad de una aplicación en la nube?
 - a. Implementar un balanceador de carga en el servidor de aplicaciones.
 - b. Utilizar contenedores para empaquetar y desplegar la aplicación.
 - c. Monitorear constantemente el tráfico y capacidad de la infraestructura en la nube.

7. ¿Cuál es la mejor opción para diseñar una arquitectura de microservicios eficiente y escalable?
 - a. Utilizar un componente monolítico para todas las funciones de la aplicación.
 - b. Dividir la aplicación en pequeños componentes independientes.
 - c. Utilizar una arquitectura SOA para integrar los diferentes componentes de la aplicación.
8. ¿Cuál es la mejor forma de monetizar una API?
 - a. Ofrecerla de forma gratuita a todos los usuarios.
 - b. Cobrar una tarifa fija por el uso de la API.
 - c. Cobrar una tarifa variable en función del uso y la demanda de la API.
9. ¿Cuál es la mejor estrategia para diseñar una arquitectura de componentes de TI escalable y eficiente?
 - a. Utilizar un enfoque monolítico para todos los componentes de la aplicación.
 - b. Dividir la aplicación en pequeños componentes independientes y escalables.
 - c. Utilizar una arquitectura EDA para integrar los diferentes componentes de la aplicación.

10. ¿Cuál es la mejor forma de gestionar los riesgos asociados al diseño de una arquitectura de aplicaciones?
- a. Realizar un análisis GAP para identificar las brechas en la arquitectura actual y la arquitectura objetivo.
 - b. Utilizar una metodología Agile para el desarrollo y despliegue de la arquitectura.
 - c. Utilizar una estrategia de *roll-out* gradual para minimizar los riesgos de la implementación.



Nota: la respuesta a las preguntas se puede encontrar en la sección de solucionario en la autoevaluación 3.

[Ir a solucionario](#)



Unidad 4. *Roadmap de arquitectura*

4.1. Hoja de ruta o *roadmap* de componentes

Una hoja de ruta empresarial es un plan de alto nivel que describe las iniciativas, proyectos y actividades clave necesarios para alcanzar las metas y objetivos de negocio de una empresa. Una hoja de ruta empresarial suele incluir un calendario que muestra cuándo se espera completar cada iniciativa, proyecto o actividad, así como las dependencias clave y los hitos asociados a cada uno.

El proceso de definición de la hoja de ruta para la implementación de la arquitectura de aplicaciones integradas a las arquitecturas de negocio y datos se lleva a cabo en varias fases dentro del marco de referencia de TOGAF:

- **Fase de preparación:** en esta fase, se establecen los objetivos y requisitos de la arquitectura de aplicaciones, se identifican los interesados y se definen las métricas para evaluar el éxito de la implementación de la arquitectura de aplicaciones.
- **Fase de definición de la arquitectura:** en esta fase, se crea una visión de la arquitectura de aplicaciones, se establece un modelo de referencia de arquitectura de aplicaciones, se identifican las opciones de diseño, se definen las interacciones entre los componentes y se desarrollan los modelos de datos.
- **Fase de planificación de la transición:** en esta fase, se define la hoja de ruta para la implementación de la arquitectura de aplicaciones, se identifican las iniciativas necesarias para la transición y se establece el plan de implementación.
- **Fase de implementación:** en esta fase, se lleva a cabo la implementación de la arquitectura de aplicaciones, se integran los sistemas de software y se prueban las soluciones.

- **Fase de seguimiento y evaluación:** en esta fase, se evalúa la implementación de la arquitectura de aplicaciones y se realiza un seguimiento continuo para garantizar que la arquitectura de aplicaciones cumpla con los requisitos de negocio y esté alineada con la estrategia empresarial.

Los profesionales de arquitectura de aplicaciones en el contexto de TOGAF deben comprender este proceso para garantizar que la arquitectura de aplicaciones agregue valor a la organización y esté alineada con la estrategia empresarial.

La hoja de ruta de la empresa suele ser elaborada por el equipo de arquitectura empresarial, en colaboración con otras partes interesadas clave, como los líderes empresariales, los directores de TI y los directores de proyecto. La hoja de ruta suele basarse en la arquitectura de la empresa, que proporciona un modelo para alinear las estrategias de negocio y de TI de la empresa.

El objetivo principal de una hoja de ruta empresarial es proporcionar una visión clara y concisa de las iniciativas, proyectos y actividades que se necesitan para apoyar las metas y objetivos de negocio de la empresa, y para permitir a la empresa adaptarse y responder a los cambios en su entorno empresarial. Una hoja de ruta empresarial también puede ayudar a coordinar y alinear los esfuerzos de los distintos equipos y partes interesadas, y a garantizar que los recursos de la empresa se utilizan de forma eficaz y eficiente.

En general, una hoja de ruta empresarial es una herramienta importante para planificar, coordinar y ejecutar las iniciativas, proyectos y actividades necesarios para apoyar las metas y objetivos de la empresa.

Una empresa necesita una hoja de ruta para orientar su estrategia, sus operaciones y su toma de decisiones. Una hoja de ruta ofrece una representación clara y visual de los planes y objetivos de la empresa, y puede utilizarse para comunicar estos planes a las partes interesadas, como empleados, inversores o clientes. Una hoja de ruta también puede ayudar a una empresa a priorizar sus actividades y asignar recursos de forma eficaz, ya que proporciona una visión clara de los pasos e hitos que deben alcanzarse para lograr sus objetivos. Además, una hoja de ruta puede utilizarse para hacer un seguimiento de los progresos y supervisar el rendimiento de la empresa, lo que permite a la organización hacer ajustes

y correcciones del rumbo según sea necesario. En general, una hoja de ruta puede proporcionar un marco valioso para alinear los esfuerzos de la empresa hacia los resultados deseados.

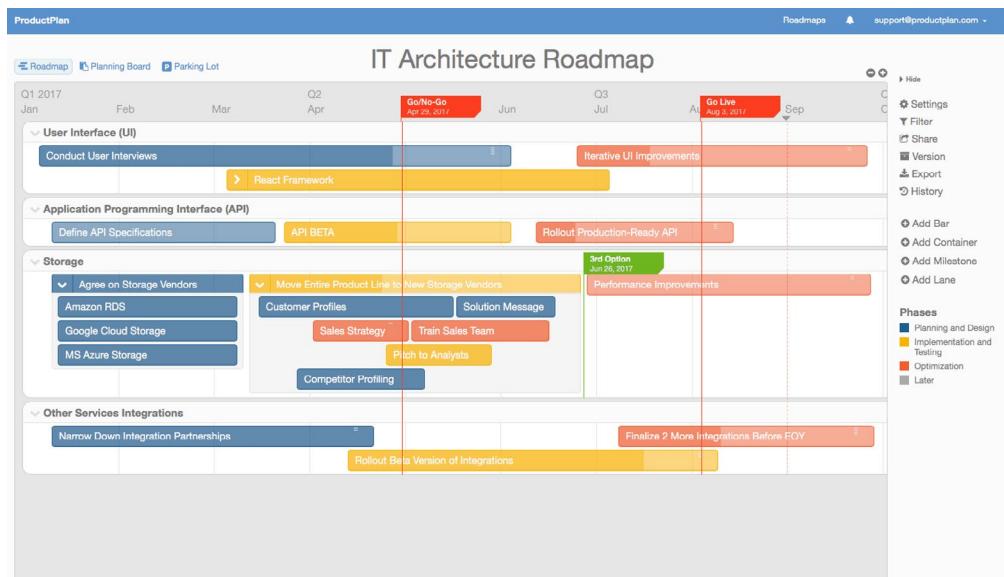
He aquí algunos pasos que puede seguir para planificar una estrategia de mejora de la arquitectura de una empresa:

- 1. Identificar el estado actual de la arquitectura de la empresa:** comience por obtener una comprensión profunda del estado actual de la arquitectura de la empresa, incluidas las tecnologías y los sistemas que están en su lugar, los desafíos y puntos de dolor a los que se enfrenta la arquitectura, y las metas y objetivos del negocio.
- 2. Definir el estado deseado de la arquitectura de la empresa:** basándose en el estado actual de la arquitectura y en las metas y objetivos de la empresa, defina el estado deseado de la arquitectura. Esto puede implicar la identificación de nuevas tecnologías o sistemas que podrían implementarse, o cambios en la arquitectura existente que mejorarían el rendimiento, la escalabilidad o la fiabilidad.
- 3. Desarrollar una hoja de ruta para la transición al estado deseado:** Elabore una hoja de ruta que describa los pasos e hitos que deben alcanzarse para pasar del estado actual de la arquitectura al estado deseado. Debe incluir un calendario, las principales partes interesadas y los recursos que se necesitarán para completar la transición.
- 4. Comunicar el plan a las partes interesadas:** comparta el plan y la hoja de ruta con las principales partes interesadas, como empleados, inversores o clientes, para asegurarse de que son conscientes de los cambios que se están planificando y tienen la oportunidad de proporcionar comentarios o aportaciones.
- 5. Aplicar el plan:** una vez finalizados el plan y la hoja de ruta, empiece a aplicar los cambios necesarios para mejorar la arquitectura de la empresa. Esto puede implicar implementar nuevas tecnologías o sistemas, o realizar cambios en la arquitectura existente.
- 6. Supervise y ajuste el plan según sea necesario:** a medida que se implementa el plan, supervise el progreso y el rendimiento de la

arquitectura, y realice los ajustes necesarios para garantizar que se alcanza el estado deseado.

Siguiendo estos pasos, se puede desarrollar una estrategia integral para mejorar la arquitectura de una empresa, y garantizar que el negocio esté bien posicionado para alcanzar sus metas y objetivos.

Figura 9.
Roadmap de IT



Nota. Adaptado de 4 Example Business Roadmaps, por ProductPlan, 2017, Product Plan.



Actividad de aprendizaje recomendada

La actividad de aprendizaje recomendada para esta semana es investigar y hacer una presentación de la hoja de ruta que debe seguir su arquitectura para llegar a la arquitectura objetivo, y definir la visión y los hitos importantes que se deben cumplir para lograr el objetivo, a través de los siguientes pasos:

- 1. Identificar la visión:** el estudiante debe identificar la visión de la arquitectura objetivo, es decir, la descripción clara y concisa de lo que se espera lograr con la arquitectura empresarial. La visión debe ser

un objetivo alcanzable y realista que pueda guiar el desarrollo de la hoja de ruta.

2. **Identificar los hitos importantes:** el estudiante debe identificar los hitos importantes que se deben cumplir para lograr la visión de la arquitectura objetivo. Estos hitos son metas específicas y concretas que se deben alcanzar para avanzar en la dirección correcta. Los hitos deben ser alcanzables y realistas, y deben estar alineados con la visión de la arquitectura.
3. **Investigar la hoja de ruta:** el estudiante debe investigar las mejores prácticas en la definición de hojas de ruta para la arquitectura empresarial. Esto puede incluir el uso de herramientas y marcos de trabajo como TOGAF y ArchiMate, entre otros. La hoja de ruta debe ser clara, detallada y fácil de seguir.
4. **Diseñar la hoja de ruta:** el estudiante debe diseñar la hoja de ruta para la arquitectura empresarial, basándose en los hitos identificados y la visión de la arquitectura objetivo. La hoja de ruta debe ser un documento detallado que describa los pasos específicos que se deben seguir para alcanzar la visión, y debe incluir plazos y recursos necesarios para cada paso.
5. **Hacer una presentación:** el estudiante debe hacer una presentación que describa la hoja de ruta y los hitos importantes que se deben cumplir para alcanzar la visión de la arquitectura objetivo. La presentación debe ser clara, concisa y debe incluir gráficos y diagramas que faciliten la comprensión de la hoja de ruta.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

En resumen, la actividad de aprendizaje recomendada es identificar la visión, identificar los hitos importantes, investigar la hoja de ruta, diseñar la hoja de ruta y hacer una presentación. Al seguir estos pasos, el estudiante podrá crear una hoja de ruta detallada que guíe el desarrollo de la arquitectura empresarial y facilite su comprensión y seguimiento.



4.2. Arquitecturas de transición

Una arquitectura de transición es un conjunto de planes y estrategias que describen los pasos, actividades y recursos necesarios para que una empresa pase de su estado actual a su estado objetivo. Una arquitectura de transición suele incluir una hoja de ruta detallada que muestra la secuencia de pasos y actividades necesarios para pasar del estado actual al estado objetivo, así como los recursos, dependencias y riesgos asociados a cada paso.

Una arquitectura de transición suele ser desarrollada por el equipo de arquitectura empresarial, en colaboración con otras partes interesadas clave, como líderes empresariales, directores de TI y directores de proyecto. La arquitectura de transición suele basarse en la arquitectura empresarial, que proporciona un plan para alinear las estrategias de negocio y de TI de la empresa.

El objetivo principal de una arquitectura de transición es proporcionar un plan claro y completo para pasar del estado actual al estado objetivo, de forma eficiente, eficaz y alineada con las metas y objetivos de negocio de la empresa. Una arquitectura de transición también puede ayudar a coordinar y alinear los esfuerzos de los distintos equipos y partes interesadas, y a garantizar que los recursos de la empresa se utilizan de forma eficaz y eficiente.

En general, una arquitectura de transición es una herramienta importante para planificar, coordinar y aplicar los cambios necesarios para apoyar las metas y objetivos de la empresa y permitirle adaptarse y responder a los cambios en su entorno empresarial.

Una matriz de transición de arquitectura es una herramienta utilizada para planificar y gestionar la transición de una arquitectura de aplicación a otra. Se trata de una tabla que asigna los componentes de la arquitectura actual a los componentes correspondientes de la arquitectura de destino, mostrando cómo se migrarán, sustituirán o retirarán. Para definir una matriz de transición, siga estos pasos:

1. Identifique las arquitecturas de aplicación actual y de destino, y enumere los componentes principales de cada una.
2. Cree una tabla con dos columnas, una para la arquitectura actual y otra para la arquitectura de destino.
3. En la primera fila de la tabla, enumere los componentes de la arquitectura actual.
4. En la primera columna de la tabla, enumera los componentes de la arquitectura de destino.
5. Para cada componente de la arquitectura actual, identifique el componente correspondiente en la arquitectura de destino e introduzcalo en la celda adecuada de la tabla.
6. Para cada componente de la arquitectura de destino que no tenga un componente correspondiente en la arquitectura actual, introduzca "nuevo" en la celda adecuada de la tabla.
7. Para cada componente de la arquitectura actual que no tenga un componente correspondiente en la arquitectura de destino, introduzca "retirado" en la celda adecuada de la tabla.
8. Revise la matriz de transición y realice las modificaciones necesarias.

La matriz de transición proporcionará una visión clara y concisa de los cambios que deben realizarse en la arquitectura de la aplicación y puede utilizarse como hoja de ruta para llevar a cabo la transición. Al definir y utilizar una matriz de transición, puede asegurarse de que la transición está bien planificada y se ejecuta sin problemas.

Una matriz de arquitecturas de transición podría incluir las siguientes columnas:

1. **Nombre de la arquitectura:** esta columna debe incluir el nombre de cada arquitectura que se está considerando para la transición.
2. **Descripción de la arquitectura:** esta columna debe proporcionar una descripción detallada de cada arquitectura, incluyendo su estructura, componentes y cómo se integran entre sí.

3. **Ventajas:** esta columna debe incluir las ventajas de cada arquitectura, como mejoras en la eficiencia, reducción de costos o mejoras en la seguridad.
4. **Desventajas:** esta columna debe incluir las desventajas de cada arquitectura, como el costo de implementación o el tiempo requerido para la transición.
5. **Recomendación:** esta columna debe incluir una recomendación sobre si se debe utilizar o no cada arquitectura para la transición.
6. **Notas:** esta columna debe incluir cualquier otra información relevante sobre la arquitectura y sus componentes.
7. **Costo de implementación:** esta columna debe incluir una estimación del costo de implementación de cada arquitectura, incluyendo el costo de *hardware*, *software* y mano de obra.
8. **Tiempo de implementación:** esta columna debe incluir una estimación del tiempo que se requerirá para implementar cada arquitectura, incluyendo el tiempo de planificación, desarrollo y pruebas.
9. **Impacto en los procesos de negocio:** esta columna debe incluir una evaluación del impacto que tendrá cada arquitectura en los procesos de negocio, incluyendo cualquier cambio en los flujos de trabajo o en la forma en que se interactúa con los clientes.
10. **Impacto en los sistemas existentes:** esta columna debe incluir una evaluación del impacto que tendrá cada arquitectura en los sistemas existentes, incluyendo cualquier cambio o actualización necesaria en la infraestructura o el *software*.

Es importante tener en cuenta que estas son solo algunas de las columnas que pueden incluirse en una matriz de arquitecturas de transición de arquitectura empresarial y que la información que se incluye dependerá de las necesidades específicas de cada organización.



Actividades de aprendizaje recomendadas

Para las actividades de aprendizaje recomendadas de esta semana se necesita entender que las arquitecturas de transición son esenciales para una arquitectura empresarial y de aplicaciones, tomando en cuenta esto le invito a desarrollar las siguientes actividades:

- A. Analizar y diseñar la arquitectura en cada uno de los hitos marcados en el *Roadmap*.
- B. Realizar una presentación con los diseños de las arquitecturas de transición de cada uno de los hitos y objetivos empresariales.

Para el desarrollo de las actividades se recomienda seguir los siguientes pasos:

1. **Investigar las arquitecturas de transición:** el estudiante debe investigar las arquitecturas de transición y comprender su importancia en el desarrollo de una arquitectura empresarial y de aplicaciones. Debe comprender cómo las arquitecturas de transición permiten a la empresa avanzar hacia la arquitectura objetivo de manera efectiva.
2. **Analizar los hitos y objetivos empresariales:** el estudiante debe analizar los hitos y objetivos empresariales establecidos en el *Roadmap* para comprender cómo se relacionan con la arquitectura de transición. Debe identificar los componentes lógicos y físicos que se necesitan para lograr los hitos y objetivos, y comprender cómo estos componentes pueden ser diseñados e implementados en la arquitectura de transición.
3. **Diseñar las arquitecturas de transición:** el estudiante debe diseñar las arquitecturas de transición para cada uno de los hitos y objetivos empresariales establecidos en el *Roadmap*. Debe identificar las soluciones de arquitectura necesarias para lograr cada hito y objetivo, y diseñar las arquitecturas de transición que permitan avanzar hacia la arquitectura objetivo de manera efectiva. Los diseños deben ser claros, detallados y deben incluir diagramas y gráficos que faciliten su comprensión.

4. **Realizar una presentación:** el estudiante debe realizar una presentación que incluya los diseños de las arquitecturas de transición para cada uno de los hitos y objetivos empresariales establecidos en el *Roadmap*. La presentación debe ser clara, concisa y debe explicar cómo las arquitecturas de transición permitirán avanzar hacia la arquitectura objetivo de manera efectiva.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

En resumen, la actividad de aprendizaje recomendada es investigar las arquitecturas de transición, analizar los hitos y objetivos empresariales, diseñar las arquitecturas de transición y realizar una presentación. Al seguir estos pasos, el estudiante podrá comprender y diseñar las arquitecturas de transición necesarias para avanzar hacia la arquitectura objetivo de manera efectiva.



Semana 14

4.3. Modelo de gobierno de arquitectura

Una empresa puede gobernar su arquitectura empresarial mediante el uso de un marco de gobierno formal que describa las funciones, responsabilidades y procesos para gestionar y mantener la arquitectura empresarial a lo largo del tiempo. Este marco suele incluir los siguientes elementos:

- **Estructura de gobierno:** define la estructura organizativa y las relaciones necesarias para apoyar el gobierno de la arquitectura empresarial. Normalmente, incluye la creación de un consejo o comité de dirección de la arquitectura empresarial, responsable de establecer la dirección y las políticas generales de la arquitectura empresarial y de supervisar su aplicación y evolución.
- **Procesos de gobernanza:** define los procesos y procedimientos específicos necesarios para apoyar el gobierno de la arquitectura empresarial. Normalmente, incluye procesos para desarrollar y mantener la arquitectura empresarial, para alinear la arquitectura empresarial con las metas y objetivos de negocio de la empresa, y

para supervisar, evaluar y adaptar la arquitectura empresarial según sea necesario.

- **Normas de gobernanza:** define las normas y mejores prácticas necesarias para apoyar el gobierno de la arquitectura empresarial. Normalmente, incluye normas para el desarrollo de la arquitectura, para el gobierno de la arquitectura y para la gestión de la arquitectura, así como normas para ámbitos específicos como la arquitectura empresarial, la arquitectura de aplicaciones y la arquitectura tecnológica.

En general, un marco de gobernanza puede ayudar a una empresa a gestionar y mantener eficazmente su arquitectura empresarial a lo largo del tiempo, y a garantizar que la arquitectura empresarial esté alineada con las metas y objetivos de negocio de la empresa. Esto puede ayudar a mejorar el rendimiento, la eficiencia y la agilidad de la empresa, y permitirle adaptarse y responder a los cambios en su entorno empresarial.

Para gobernar las aplicaciones en la nube, puede seguir algunos pasos:

1. **Defina y documente sus políticas de gobierno de la nube:** cree un conjunto de políticas que describan cómo su organización utilizará la nube, incluyendo quién tiene acceso a la nube, qué tipos de cargas de trabajo se permiten ejecutar en la nube y cómo se asignarán y gestionarán los recursos.
2. **Utilice herramientas y servicios para gestionar y supervisar su entorno de nube:** hay muchas herramientas y servicios disponibles que pueden ayudarte a gestionar y supervisar tu entorno de nube, como plataformas de gestión de nube, herramientas de supervisión y servicios de seguridad. Estas herramientas pueden ayudarle a garantizar que su entorno en la nube cumple con sus políticas de gobernanza y que sus aplicaciones funcionan sin problemas.
3. **Implemente un equipo de gobernanza de la nube:** para garantizar el cumplimiento de sus políticas de gobernanza de la nube, considere la posibilidad de implementar un equipo dedicado a la gobernanza de la nube. Este equipo puede encargarse de supervisar la aplicación y el cumplimiento de sus políticas, así como de supervisar su entorno de nube y garantizar que funciona de forma eficiente y segura.

4. **Forme a sus empleados en el gobierno de la nube:** para asegurarse de que todos en su organización conocen y siguen sus políticas de gobierno de la nube, es importante proporcionar capacitación sobre estas políticas a todos sus empleados. Esto puede ayudar a evitar errores y garantizar que el entorno de la nube se utiliza de una manera que cumple con sus políticas.

En general, una gobernanza eficaz de la nube requiere una combinación de políticas bien definidas, el uso de herramientas y servicios para gestionar y supervisar su entorno de nube, un equipo dedicado a supervisar la gobernanza y formación para todos los empleados sobre sus políticas.

Para crear un plan de gobierno de la nube, una empresa necesita varias cosas:

- Una comprensión clara de sus objetivos y requisitos para el uso de la nube, incluidos los tipos de cargas de trabajo que se ejecutarán en la nube y cómo se asignarán y gestionarán los recursos.
- Un conjunto de políticas que describan cómo se utilizará la nube, quién tendrá acceso a ella y cómo se gestionará y supervisará.
- Herramientas y servicios para ayudar a gestionar y supervisar su entorno de nube, como plataformas de gestión de nube, herramientas de supervisión y servicios de seguridad.
- Un equipo dedicado a supervisar la aplicación y el cumplimiento de sus políticas de gobernanza de la nube, así como a supervisar su entorno de nube.
- Formación de los empleados sobre las políticas de gobierno de la nube de la empresa para garantizar que se cumplen.

Contar con estos elementos puede ayudar a garantizar que el entorno de nube de una empresa se utiliza de forma eficiente, segura y conforme a sus políticas.

Hay muchas políticas diferentes de gobierno de la nube que una organización puede optar por aplicar, en función de sus objetivos y requisitos específicos. Aquí hay 10 ejemplos de políticas comunes de gobierno de la nube:

1. **Política de control de acceso:** esta política define quién tiene acceso a la nube y qué nivel de acceso tiene.
2. **Política de seguridad:** esta política describe cómo se protegerá el entorno de la nube, incluyendo medidas como el cifrado y la autenticación.
3. **Política de cumplimiento:** esta política garantiza que el uso de la nube por parte de la organización cumple con las leyes y reglamentos pertinentes.
4. **Política de gestión de datos:** esta política describe cómo se almacenarán, accederán y gestionarán los datos en la nube.
5. **Política de asignación de recursos:** esta política define cómo se asignarán y gestionarán en la nube recursos como la computación, el almacenamiento y las redes.
6. **Política de supervisión e información:** esta política describe cómo se supervisará el entorno de la nube y cómo se generarán los informes sobre su rendimiento y uso.
7. **Política de respuesta a incidentes:** esta política define los procedimientos para responder a incidentes como brechas de seguridad o interrupciones en el entorno de nube.
8. **Política de recuperación de desastres:** esta política describe cómo la organización se recuperará de desastres tales como desastres naturales o pérdida de datos en la nube.
9. **Política de gestión de costes:** esta política define cómo la organización gestionará y controlará los costes asociados a su uso de la nube.
10. **Política de Acuerdos de Nivel de Servicio (SLA):** esta política describe los términos del acuerdo entre la organización y su proveedor de servicios en la nube, incluidos los niveles de servicio y soporte.

Estos son solo algunos ejemplos de políticas de gobierno de la nube. Una organización puede optar por aplicar otras políticas en función de sus necesidades y requisitos específicos.

No existe una respuesta única a cuál es la mejor arquitectura de un almacén de datos o *data warehouse*, ya que la arquitectura adecuada dependerá de las necesidades y requisitos específicos de la empresa. Sin embargo, hay algunas características clave que se suelen considerar parte de una buena arquitectura de almacén de datos, como:

- **Escalabilidad:** el almacén de datos debe ser capaz de manejar grandes volúmenes de datos y soportar el crecimiento del negocio.
- **Flexibilidad:** el almacén de datos debe ser capaz de adaptarse a los cambios en las fuentes de datos, la estructura de datos o los requisitos de negocio sin requerir un rediseño significativo o tiempo de inactividad.
- **Rendimiento:** el almacén de datos debe diseñarse para que las consultas y los informes sean rápidos y eficientes, utilizando técnicas como la indexación, la partición o el procesamiento paralelo.
- **Seguridad:** el almacén de datos debe contar con sólidos controles de seguridad para protegerlo de accesos no autorizados, filtraciones o pérdidas de datos.
- **Calidad de los datos:** el almacén de datos debe contar con procesos y controles que garanticen la calidad, precisión e integridad de los datos.
- **Gobierno de datos:** el almacén de datos debe contar con un marco de gobernanza de datos para gestionar y controlar los datos, incluyendo políticas, procesos y funciones y responsabilidades.
- **Acceso autoservicio:** el almacén de datos debe ofrecer a los usuarios la posibilidad de acceder a los datos y consultarlos en modo autoservicio, sin necesidad de que intervenga el departamento de TI.

En general, la mejor arquitectura para un almacén de datos será la que satisfaga las necesidades específicas de la empresa y permita a la organización obtener valor y conocimientos de sus datos.

4.3.1. DORA metrics

Las métricas DORA, o DevOps Research and Assessment (DORA) metrics, son un conjunto de métricas y puntos de referencia para medir

el rendimiento y la eficacia de las prácticas DevOps de una organización. DORA fue desarrollado por la Dra. Nicole Forsgren, Jez Humble y Gene Kim, y se basa en años de investigación y datos recogidos de miles de organizaciones. Las métricas DORA consisten en cuatro Indicadores Clave de Rendimiento (KPI):

- **Frecuencia de despliegue:** esta métrica mide la frecuencia con la que un equipo despliega código en producción. Una mayor frecuencia de despliegue indica que el equipo es capaz de entregar cambios a sus aplicaciones de forma rápida y fiable.
- **Tiempo de entrega de los cambios:** esta métrica mide el tiempo que tarda un cambio de código en desplegarse en producción. Un tiempo de espera más corto indica que el equipo es capaz de moverse rápida y eficientemente a través del proceso de desarrollo y despliegue.
- **Tiempo de restablecimiento del servicio:** esta métrica mide la rapidez con la que un equipo es capaz de restaurar el servicio después de un fallo o interrupción. Un tiempo más rápido para restablecer el servicio indica que el equipo está bien preparado y es capaz de responder rápidamente a los incidentes.
- **Tasa de fallos en los cambios:** esta métrica mide el porcentaje de implantaciones que acaban en fracaso. Un menor índice de fallos en los cambios indica que el equipo es capaz de desplegar los cambios de forma fiable y con confianza.

Las organizaciones suelen utilizar las métricas DORA para evaluar y mejorar sus prácticas DevOps y para comparar su rendimiento con los puntos de referencia del sector. Mediante el seguimiento y la mejora de estas métricas, las organizaciones pueden lograr mejores resultados y un mayor valor empresarial de sus iniciativas DevOps.

4.3.2. OKR

OKR son las siglas en inglés de Objectives and Key Results, que en español se traducen como Objetivos y Resultados Clave. Se trata de una herramienta de gestión que permite establecer objetivos claros y medibles, y definir los resultados clave que se deben lograr para alcanzar dichos objetivos.

Los OKR se utilizan comúnmente en empresas y organizaciones para establecer metas y objetivos a nivel individual, de equipo o de la organización en general, y para evaluar el progreso y el rendimiento en relación con dichos objetivos. Los OKR se componen de dos partes: los objetivos, que son declaraciones específicas y medibles sobre lo que se quiere lograr, y los resultados clave, que son los indicadores que se utilizan para medir el progreso y el éxito en la consecución de los objetivos.

Los OKR suelen ser establecidos para un período determinado de tiempo, como un trimestre o un semestre, y se revisan y actualizan regularmente para adaptarse a las necesidades cambiantes del negocio o la organización. Al establecer objetivos claros y medibles y definir los resultados clave que se deben lograr, los OKR pueden ayudar a alinear los esfuerzos y recursos de la organización hacia los objetivos estratégicos y mejorar el rendimiento y la eficacia en el logro de los mismos.

4.3.3. KPI

KPI son las siglas en inglés de Key Performance Indicators, que en español se traducen como Indicadores Clave de Desempeño. Se trata de una herramienta de medición que se utiliza para evaluar y medir el rendimiento y el éxito de una empresa u organización en relación con sus objetivos y metas estratégicas.

Los KPI se definen como métricas específicas y medibles que se utilizan para evaluar el progreso y el rendimiento de una empresa o una organización en relación con sus objetivos estratégicos. Estos indicadores son seleccionados cuidadosamente en función de los objetivos específicos de la organización y deben ser fácilmente medibles y verificables.

Los KPI pueden ser aplicados a diferentes áreas de la empresa, como finanzas, *marketing*, operaciones, recursos humanos, entre otras, y se utilizan para evaluar el éxito en relación con los objetivos y metas específicos en cada área. Los KPI también pueden ser utilizados para realizar comparaciones entre diferentes períodos de tiempo y para comparar el rendimiento de una empresa o una organización con el de sus competidores.

Los KPI son una herramienta importante para la gestión y toma de decisiones en una empresa u organización, ya que permiten evaluar el

rendimiento en relación con los objetivos estratégicos y realizar ajustes y mejoras para alcanzar los resultados deseados.



Actividades de aprendizaje recomendadas

Le invito a realizar las siguientes actividades de aprendizaje recomendadas :

- A. Proponer el gobierno del portafolio de aplicaciones.
- B. Definir políticas de mantenimiento y KPI y OKR con base en la estrategia que seguirá la empresa donde se está aplicando la arquitectura empresarial.

Para el desarrollo de las actividades se recomienda:

1. **Identificar el portafolio de aplicaciones:** el estudiante debe identificar el portafolio de aplicaciones que se encuentra en la empresa. Debe comprender cómo estas aplicaciones se relacionan con la estrategia empresarial y cómo se ajustan a la arquitectura empresarial.
2. **Definir las políticas de mantenimiento:** el estudiante debe definir las políticas de mantenimiento que se deben seguir para garantizar que las aplicaciones del portafolio se mantengan actualizadas y operando de manera óptima. Debe identificar los procesos y procedimientos necesarios para el mantenimiento de las aplicaciones, así como los roles y responsabilidades de los involucrados.
3. **Definir los KPI y OKR :** el estudiante debe definir los KPI y OKR que se utilizarán para medir el rendimiento del portafolio de aplicaciones. Debe identificar los Indicadores Clave de Rendimiento (KPI) que se utilizarán para evaluar el rendimiento de las aplicaciones, así como los Objetivos Clave de Rendimiento (OKR) que se utilizarán para establecer metas específicas para mejorar el rendimiento.
4. **Proponer el gobierno del portafolio de aplicaciones:** el estudiante debe proponer un marco de gobierno del portafolio de aplicaciones que incluya las políticas de mantenimiento y los KPI y OKR definidos. Debe identificar los roles y responsabilidades de los involucrados

en el gobierno del portafolio de aplicaciones, y definir los procesos y procedimientos necesarios para el gobierno efectivo del portafolio.

5. **Hacer una presentación:** el estudiante debe hacer una presentación que describa el marco de gobierno del portafolio de aplicaciones propuesto. La presentación debe ser clara, concisa y debe incluir gráficos y diagramas que faciliten la comprensión del marco.

Nota: conteste las actividades en un cuaderno de apuntes o en un documento Word.

En resumen, la actividad de aprendizaje recomendada es identificar el portafolio de aplicaciones, definir las políticas de mantenimiento, definir los KPI y OKR , proponer el gobierno del portafolio de aplicaciones, y hacer una presentación. Al seguir estos pasos, el estudiante podrá proponer un marco de gobierno efectivo para el portafolio de aplicaciones que garantice el rendimiento óptimo de las aplicaciones y esté alineado con la estrategia empresarial.



Semana 15

4.4. Revisión y presentación a los stakeholders

Para presentar mi arquitectura de aplicaciones a los *stakeholders*, seguiría los siguientes pasos:

1. **Preparar una presentación visual:** una vez que he diseñado mi arquitectura de aplicaciones, prepararía una presentación visual que muestre la estructura y los componentes de la arquitectura de manera clara y concisa. Esto puede incluir diagramas, gráficos y otros tipos de visualización.
2. **Identificar a los stakeholders:** antes de presentar la arquitectura de aplicaciones, es importante identificar a todos los *stakeholders* que deben ser informados o involucrados en la decisión de implementar la arquitectura. Esto puede incluir a los ejecutivos, el equipo de TI, los usuarios finales y cualquier otra parte interesada.

- 3. Establecer objetivos de la presentación:** antes de presentar la arquitectura de aplicaciones, es importante tener en claro los objetivos de la presentación. Estos objetivos pueden incluir informar a los *stakeholders* sobre la arquitectura, obtener su aprobación para implementarla o discutir cualquier pregunta o preocupación que puedan tener.
- 4. Programar una reunión:** una vez que he preparado la presentación y he identificado a los *stakeholders*, programaría una reunión para presentar la arquitectura de aplicaciones. Aseguraría que se proporcione suficiente tiempo para la presentación y que se incluya tiempo para preguntas y discusión. También aseguraría que la reunión se realice en un lugar adecuado y que se proporcionen todos los recursos necesarios, como proyectores o pantallas.
- 5. Realizar la presentación:** durante la reunión, realizaría la presentación de la arquitectura de aplicaciones siguiendo los objetivos establecidos previamente. Aseguraría de ser claro y conciso al explicar la estructura y los componentes de la arquitectura y de responder cualquier pregunta o inquietud que puedan tener los *stakeholders*.
- 6. Escuchar y obtener retroalimentación:** durante la presentación, es importante escuchar atentamente a los *stakeholders* y obtener su retroalimentación. Esto puede incluir preguntas, comentarios o sugerencias sobre la arquitectura de aplicaciones. Tomar en cuenta esta retroalimentación puede ayudar a mejorar la arquitectura y a asegurar que se cumplan los objetivos de la organización.
- 7. Ajustar la arquitectura con base en la retroalimentación:** una vez que he obtenido la retroalimentación de los *stakeholders*, puedo ajustar la arquitectura de aplicaciones en consecuencia para asegurar que se cumplan los objetivos de la organización y se tenga en cuenta la perspectiva de todas las partes interesadas.
- 8. Obtener la aprobación para implementar la arquitectura:** una vez que he presentado la arquitectura de aplicaciones y he ajustado con base en la retroalimentación, es importante obtener la aprobación para implementar la arquitectura. Esto puede incluir obtener la aprobación de los ejecutivos o de cualquier otra parte interesada en la decisión.

9. **Programar una reunión:** una vez que he preparado la presentación y he identificado a los *stakeholders*, programaría una reunión para presentar la arquitectura de aplicaciones. Aseguraría que se proporcione suficiente tiempo para la presentación y que se incluya tiempo para preguntas y discusión. También aseguraría que la reunión se realice en un lugar adecuado y que se proporcionen todos los recursos necesarios, como proyectores o pantallas.
10. **Implementar la arquitectura:** una vez que he obtenido la aprobación para implementar la arquitectura de aplicaciones, es importante seguir un plan detallado para garantizar una implementación exitosa. Esto puede incluir la adquisición de *hardware* y *software* necesarios, la migración de datos y la capacitación del personal. Aseguraría también de monitorear el progreso de la implementación y de realizar ajustes según sea necesario.
11. **Monitorear y evaluar la arquitectura:** una vez que se ha implementado la arquitectura de aplicaciones, es importante monitorear y evaluar su rendimiento para asegurar que se está cumpliendo con los objetivos de la organización. Esto puede incluir la recopilación de datos sobre la eficiencia, la satisfacción del usuario y cualquier otra métrica relevante. Con base en esta información, puedo realizar ajustes a la arquitectura si es necesario para mejorar su rendimiento y cumplir con los objetivos de la organización.

En resumen, el proceso para presentar una arquitectura de aplicaciones a los *stakeholders* incluye preparar una presentación visual, identificar a los *stakeholders*, establecer objetivos de la presentación, programar una reunión, realizar la presentación, escuchar y obtener retroalimentación, ajustar la arquitectura con base en la retroalimentación y obtener la aprobación para implementar la arquitectura. Es importante seguir este proceso de manera cuidadosa para garantizar una implementación exitosa de la arquitectura de aplicaciones.



Actividades de aprendizaje recomendadas

Le invito a desarrollar las siguientes actividades de aprendizaje recomendadas para esta semana:

- A. Preparar una presentación completa para ser presentada a los **Stakeholders** de la empresa donde se está levantando la arquitectura empresarial, considerar lo siguiente:
 1. **Identificar los stakeholders:** el estudiante debe identificar los **Stakeholders** de la empresa que necesitan conocer los detalles de la arquitectura empresarial. Debe comprender sus roles, responsabilidades y sus objetivos empresariales específicos.
 2. **Definir los objetivos de la presentación:** el estudiante debe definir los objetivos de la presentación y asegurarse de que sean claros y específicos. Debe identificar los mensajes clave que desea transmitir a los **Stakeholders** y cómo estos mensajes se relacionan con los objetivos de la empresa.
 3. **Preparar el contenido de la presentación:** el estudiante debe preparar el contenido de la presentación con base en los objetivos definidos anteriormente. Debe incluir detalles sobre la arquitectura empresarial, los objetivos empresariales, la estrategia, los hitos importantes y las arquitecturas de transición. También debe incluir información sobre los KPI y OKR, las políticas de mantenimiento y el gobierno del portafolio de aplicaciones.
 4. **Diseñar la presentación:** el estudiante debe diseñar la presentación de manera efectiva para garantizar que sea clara, fácil de entender y visualmente atractiva. Debe utilizar gráficos, diagramas y otros elementos visuales para ayudar a transmitir la información de manera efectiva.
 5. **Practicar la presentación:** el estudiante debe practicar la presentación varias veces para asegurarse de que pueda transmitir los mensajes clave de manera efectiva y responder a las preguntas de los **Stakeholders** de manera adecuada.

- 6. Realizar la presentación:** finalmente, el estudiante debe realizar la presentación a los *Stakeholders* de la empresa. Debe estar preparado para responder preguntas y comentarios de manera efectiva, y para ajustar la presentación según sea necesario para satisfacer las necesidades de los *Stakeholders*.

Nota: conteste las actividades en un cuaderno de apuntes, en un documento Word. Para desarrollar las presentaciones, utilice el programa recomendado por el tutor.

En resumen, la actividad de aprendizaje recomendada es identificar los *Stakeholders*, definir los objetivos de la presentación, preparar el contenido de la presentación, diseñar la presentación, practicar la presentación y realizar la presentación. Al seguir estos pasos, el estudiante podrá presentar la arquitectura empresarial de manera efectiva y satisfacer las necesidades de los *Stakeholders* de la empresa.

- B. Realizar la siguiente autoevaluación:**



Autoevaluación 4

1. ¿Qué es el *Roadmap* en la arquitectura de aplicaciones?
 - a. Una lista de tareas diarias para el equipo de arquitectura.
 - b. Una planificación de los hitos y entregables de la arquitectura en un horizonte de tiempo determinado.
 - c. Una representación gráfica de la arquitectura de la aplicación.
2. ¿Qué son las arquitecturas de transición en la arquitectura de aplicaciones?
 - a. Arquitecturas que cambian con frecuencia según los requerimientos del negocio.
 - b. Arquitecturas que se utilizan durante el proceso de migración de una aplicación.
 - c. Arquitecturas que se utilizan en aplicaciones de prueba.
3. ¿Qué es el Gobierno de Arquitectura en la arquitectura de aplicaciones?
 - a. La gestión de los riesgos y el cumplimiento de las políticas de seguridad en la arquitectura de la aplicación.
 - b. La gestión del personal del equipo de arquitectura.
 - c. La gestión del presupuesto y los recursos de la arquitectura de la aplicación.
4. ¿Qué son las DORA Metrics en la arquitectura de aplicaciones?
 - a. Métricas que miden la velocidad y la estabilidad en el proceso de desarrollo y despliegue de la aplicación.
 - b. Métricas que miden la cantidad de usuarios que utilizan la aplicación.
 - c. Métricas que miden el consumo de recursos de la aplicación.

5. ¿Qué es la Presentación de Arquitectura a *stakeholders* en la arquitectura de aplicaciones?
 - a. Un proceso de presentación formal de la arquitectura de la aplicación a los interesados del negocio.
 - b. Una presentación informal de la arquitectura de la aplicación a los miembros del equipo de desarrollo.
 - c. Una presentación visual de la arquitectura de la aplicación.
6. ¿Cuál es el objetivo principal del Gobierno de Arquitectura?
 - a. Establecer políticas y estándares para el desarrollo de la arquitectura empresarial.
 - b. Implementar soluciones de arquitectura que satisfagan los requerimientos de negocio.
 - c. Gestionar y monitorear la implementación de la arquitectura empresarial.
7. ¿Qué es una arquitectura de transición?
 - a. Una arquitectura que se enfoca en la capa de presentación de una aplicación.
 - b. Una arquitectura que se enfoca en la capa de lógica de una aplicación.
 - c. Una arquitectura que describe el proceso de cambio de una arquitectura existente a una nueva.
8. ¿Qué son las DORA Metrics?
 - a. Métricas que miden la eficiencia de la implementación de soluciones de arquitectura.
 - b. Métricas que miden la eficacia de la implementación de soluciones de arquitectura.
 - c. Métricas que miden la calidad y el rendimiento de los procesos de desarrollo y entrega de software.

9. ¿Cuál es el propósito de un *Roadmap* de Arquitectura?
- a. Proporcionar una visión de alto nivel de la estrategia de la organización y los objetivos a largo plazo.
 - b. Identificar los componentes de TI necesarios para implementar soluciones de arquitectura.
 - c. Establecer políticas y estándares para el desarrollo de la arquitectura empresarial.
10. ¿Cuál es el enfoque principal al presentar la arquitectura a los stakeholders?
- a. Detallar la implementación técnica de la arquitectura.
 - b. Centrarse en los beneficios de negocio y la alineación con la estrategia de la organización.
 - c. Demostrar el rendimiento y la eficiencia de la arquitectura empresarial.



Nota: estimado estudiante, las respuestas las puede encontrar en la sección de solucionario, en la autoevaluación 4.

[Ir a solucionario](#)



Semana 16

El segundo examen parcial es una oportunidad para demostrar sus habilidades y conocimientos en los temas fundamentales de la arquitectura de aplicaciones que se ha aprendido hasta ahora. Este es un momento crucial en su formación académica y profesional.

- **Diseño de arquitectura:** el diseño de arquitectura es esencial para el éxito de cualquier proyecto de arquitectura de aplicaciones. Prepararte en este tema te permitirá comprender cómo se lleva a cabo un diseño efectivo de arquitectura y cómo se aplican las mejores prácticas en este ámbito.
- **Gap análisis:** el gap análisis es una herramienta importante en la arquitectura de aplicaciones que ayuda a identificar las brechas entre la arquitectura actual y la arquitectura objetivo. Prepararte en este tema te permitirá comprender cómo se lleva a cabo el gap análisis y cómo se aplican las mejores prácticas en este ámbito.
- **Gestión de Riesgos:** la gestión de riesgos es esencial para cualquier proyecto de arquitectura de aplicaciones. Prepararte en este tema te permitirá comprender cómo se lleva a cabo la gestión de riesgos y cómo se aplican las mejores prácticas en este ámbito.
- **Hoja de ruta:** la hoja de ruta es esencial para la planificación y gestión de proyectos de arquitectura de aplicaciones. Prepararte en este tema te permitirá comprender cómo se lleva a cabo una hoja de ruta efectiva y cómo se aplican las mejores prácticas en este ámbito.
- **Arquitecturas de transición:** las arquitecturas de transición son esenciales para cualquier proyecto de arquitectura de aplicaciones, ya que ayudan a garantizar la continuidad del negocio durante la transición entre la arquitectura actual y la arquitectura objetivo. Prepararte en este tema te permitirá comprender cómo se lleva a cabo una arquitectura de transición efectiva y cómo se aplican las mejores prácticas en este ámbito.
- **Gobierno de arquitectura:** el gobierno de arquitectura es esencial para garantizar el éxito a largo plazo de la arquitectura de aplicaciones. Prepararte en este tema te permitirá comprender cómo se lleva a

cabo el gobierno de arquitectura y cómo se aplican las mejores prácticas en este ámbito.

- **Revisión con stakeholders:** la revisión con *stakeholders* es esencial para garantizar que la arquitectura de aplicaciones cumpla con los objetivos empresariales y las necesidades de los *stakeholders*. Prepararte en este tema te permitirá comprender cómo se lleva a cabo una revisión efectiva con *stakeholders* y cómo se aplican las mejores prácticas en este ámbito.

Tendremos como actividad calificada la evaluación bimestral, que consiste en un examen con preguntas objetivas que abarcan los contenidos de todo el segundo bimestre. Prepare con anticipación su examen que consistirá en la aplicación tanto de sus habilidades de comprensión teórica como de la aplicación de ejemplos y casos de razonamiento. Le sugiero que siga esta estrategia para su preparación:

1. Revise los contenidos de cada uno de los temas tratados en el bimestre.
2. Revise las tablas, ejemplos prácticos y casos propuestos.
3. Realice la autoevaluación al finalizar cada unidad, para comprender cuáles son los conceptos que debe reforzar en su preparación.
4. Revise los apuntes de su bitácora para validar sus resúmenes y los puntos de refuerzo que ha considerado para cada semana.

¡Qué tenga mucho éxito en el desarrollo de su evaluación bimestral!



4. Solucionario

Autoevaluación 1		
Pregunta	Respuesta	Retroalimentación
1	a	El Metamodelo es esencial para garantizar que todos los miembros del equipo de arquitectura empresarial estén en la misma página y hablen el mismo idioma.
2	a	Los principios de arquitectura son esenciales para asegurar la coherencia y la integridad de la arquitectura empresarial, y proporcionan una guía clara para la toma de decisiones.
3	a	Las capacidades de negocio son una parte crítica de la arquitectura empresarial, y deben ser consideradas al diseñar soluciones de arquitectura empresarial, ya que representa a lo que el negocio puede hacer.
4	a	Los building blocks son componentes reutilizables de la arquitectura empresarial y proporcionan una forma eficiente de diseñar soluciones de arquitectura que satisfagan los requerimientos de negocio.
5	c	Los requerimientos empresariales se refieren a las necesidades, objetivos y metas de la organización, y son la base para el desarrollo de una arquitectura de aplicaciones efectiva y eficiente.
6	a	Un modelo de arquitectura en TOGAF es una representación simplificada y estructurada de la arquitectura empresarial. Los modelos de arquitectura son una herramienta esencial para comunicar la arquitectura empresarial a todas las partes interesadas, y permiten una comprensión clara de la estructura y los componentes de la arquitectura empresarial.
7	c	La cadena de valor en TOGAF se utiliza para identificar las actividades de negocio que agregan valor a la organización. La cadena de valor es esencial para garantizar que la arquitectura empresarial se alinee con los objetivos y requerimientos de negocio de la organización.

Autoevaluación 1		
Pregunta	Respuesta	Retroalimentación
8	b	Los insumos de la fase de arquitectura de aplicaciones proporcionan la información necesaria para tomar decisiones informadas sobre la arquitectura empresarial y para diseñar soluciones de arquitectura empresarial que sean efectivas y eficientes.
9	c	La fase de arquitectura de aplicaciones es la fase en la que se diseña la arquitectura de aplicaciones detallada, se seleccionan los building blocks necesarios para la solución de arquitectura y se definen los servicios de aplicación. También se desarrolla un plan de transición que establece cómo se implementará la arquitectura de aplicaciones en la organización. En resumen, la fase de arquitectura de aplicaciones es la fase en la que se diseña e implementa la solución de arquitectura que se alinea con los requerimientos de negocio y la visión empresarial.
10	a	TOGAF proporciona un lenguaje común para la comunicación entre los arquitectos empresariales y otros miembros de la organización, lo que reduce la confusión y mejora la eficiencia. En resumen, el beneficio principal de utilizar TOGAF es proporcionar una estructura y un conjunto de herramientas para el desarrollo de la arquitectura empresarial y establecer un lenguaje común para la comunicación entre los miembros de la organización.

[Ir a la autoevaluación](#)

Autoevaluación 2		
Pregunta	Respuesta	Retroalimentación
1	b	Esta opción describe lo que es la arquitectura de microservicios, que se caracteriza por dividir una aplicación en servicios independientes y altamente cohesivos que se comunican entre sí a través de protocolos como HTTP y TCP.
2	c	REST es un protocolo de comunicación que se basa en una arquitectura cliente-servidor y utiliza el protocolo HTTP para intercambiar datos en formatos como JSON o XML.
3	c	ETL es un proceso utilizado en la integración de datos para extraer datos de diversas fuentes, transformarlos en un formato común y cargarlos en una base de datos u otro destino.
4	a	<i>Clean Code</i> es un conjunto de buenas prácticas y principios que se utilizan para escribir código limpio, legible y fácil de mantener.
5	c	La arquitectura de datos es un enfoque sistemático y estructurado para la gestión y organización de los datos de una aplicación o empresa. Se centra en la definición de políticas, estándares y modelos para la captura, almacenamiento, integración, distribución y uso de los datos. La arquitectura de datos también aborda aspectos como la seguridad, la privacidad, la calidad y el gobierno de los datos.
6	b	REST es el protocolo de comunicación más comúnmente utilizado en arquitecturas de microservicios debido a su simplicidad, escalabilidad y flexibilidad.
7	b	SOA (Service-Oriented Architecture), ya que es un enfoque arquitectónico que se centra en el uso de servicios como componentes fundamentales para el desarrollo de sistemas empresariales, promoviendo la reutilización de servicios y la interoperabilidad entre sistemas heterogéneos.
8	c	gRPC, ya que es un enfoque de arquitectura de software que se centra en el diseño de interfaces de comunicación claramente definidas y contratos bien definidos entre servicios, lo que permite la comunicación eficiente y confiable entre los componentes del sistema.
9	a	ETL (Extract, Transform, Load), ya que es una tecnología comúnmente utilizada para la integración de datos en sistemas empresariales, permitiendo la extracción de datos de múltiples fuentes, la transformación de los datos en un formato común y la carga de los datos en un destino final.

Autoevaluación 2		
Pregunta	Respuesta	Retroalimentación
10	b	CSR (Client-Side Rendering), ya que se utiliza comúnmente en arquitecturas de aplicaciones para mejorar el rendimiento y la experiencia del usuario al permitir que el procesamiento de la interfaz de usuario se realice en el lado del cliente, reduciendo la carga en el servidor y mejorando la capacidad de respuesta del sistema.

Ir a la
autoevaluación

Autoevaluación 3		
Pregunta	Respuesta	Retroalimentación
1	b	Los componentes de TI se refieren a todos los elementos de <i>hardware, software, redes y datos</i> que conforman la infraestructura tecnológica de una organización.
2	b	Los contenedores son una tecnología de virtualización que permite ejecutar aplicaciones de forma aislada y portátil.
3	b	La arquitectura objetivo describe el estado futuro deseado de la organización y sirve como guía para la toma de decisiones en cuanto a la implementación de soluciones de arquitectura.
4	b	La monetización de API implica generar ingresos a través de la utilización de las mismas, ya sea mediante la venta de acceso a las API o mediante el cobro por el uso de las mismas.
5	b	El gap análisis y gestión de riesgos en arquitectura de aplicaciones implica identificar y evaluar los riesgos asociados con la implementación de soluciones de arquitectura y establecer medidas para mitigar dichos riesgos.
6	b	Los contenedores permiten empaquetar todo lo necesario para ejecutar la aplicación, incluyendo dependencias y configuraciones, lo que facilita su despliegue y escalabilidad en la nube.
7	b	Los microservicios permiten dividir la aplicación en componentes independientes y escalables, lo que facilita la gestión y el mantenimiento de la arquitectura.
8	c	Permite adaptar la tarifa al uso y la demanda de la API, lo que puede generar mayores ingresos y rentabilidad para el negocio.
9	b	Permite dividir la aplicación en componentes independientes y escalables, lo que facilita la gestión y el mantenimiento de la arquitectura y su escalabilidad.
10	a	Permite identificar las brechas entre la arquitectura actual y la objetivo, lo que facilita la toma de decisiones y la mitigación de riesgos asociados al diseño de la arquitectura.

Ir a la
autoevaluación

Autoevaluación 4		
Pregunta	Respuesta	Retroalimentación
1	b	<i>Roadmap</i> es una planificación estratégica que establece los hitos y entregables de la arquitectura en un horizonte de tiempo determinado.
2	b	Las arquitecturas de transición se utilizan durante el proceso de migración de una aplicación, para permitir la continuidad del negocio y reducir el impacto de la transición.
3	a	Gobierno de Arquitectura se encarga de la gestión de los riesgos y el cumplimiento de las políticas de seguridad en la arquitectura de la aplicación.
4	a	DORA Metrics miden la velocidad y la estabilidad en el proceso de desarrollo y despliegue de la aplicación, para ayudar a mejorar la calidad y la eficiencia del proceso.
5	a	Presentación de Arquitectura a <i>stakeholders</i> es un proceso formal de presentación de la arquitectura de la aplicación a los interesados del negocio para obtener su aprobación y su apoyo.
6	a	Gobierno de Arquitectura tiene como objetivo principal establecer políticas y estándares para el desarrollo de la arquitectura empresarial, garantizando que los proyectos de TI cumplan con los objetivos y estrategias de la organización.
7	c	Una arquitectura de transición describe el proceso de cambio de una arquitectura existente a una nueva, permitiendo que la organización planifique y gestione la transición de manera efectiva.
8	c	DORA Metrics son métricas que miden la calidad y el rendimiento de los procesos de desarrollo y entrega de software, lo que permite a las organizaciones evaluar y mejorar su capacidad para entregar software de manera rápida y confiable.
9	a	El propósito de un <i>Roadmap</i> de Arquitectura es proporcionar una visión de alto nivel de la estrategia de la organización y los objetivos a largo plazo, lo que permite a la organización planificar y priorizar iniciativas de TI en función de su estrategia y objetivos.
10	b	El enfoque principal al presentar la arquitectura a los <i>stakeholders</i> es centrarse en los beneficios de negocio y la alineación con la estrategia de la organización, lo que permite a los <i>stakeholders</i> entender el valor de la arquitectura empresarial y su impacto en la organización.

[Ir a la autoevaluación](#)



5. Glosario

- **ADM:** Architecture Development Method.
- **BPM:** Business Process Management.
- **BPMN:** Business Process Modeling Notation.
- **CRM:** Customer Relationship Management.
- **ERP:** Enterprise Resource Planning.
- **ETL:** Extract Transform and Load.
- **POS:** Point of Sale.
- **SBS:** Solution Building Block.
- **SBS :** Solution Buildings Blocks (SBS).
- **SRM:** Supplier Resource Management.
- **TOGAF:** The Open Group Framework.
- **OMG:** OMG (Object Management Group).
- **SSR:** Server Side Rendering.
- **CSR:** Client Side Rendering.
- **COTS:** Comercial off-the-shelf (Componente Comercial Salido del Estante).



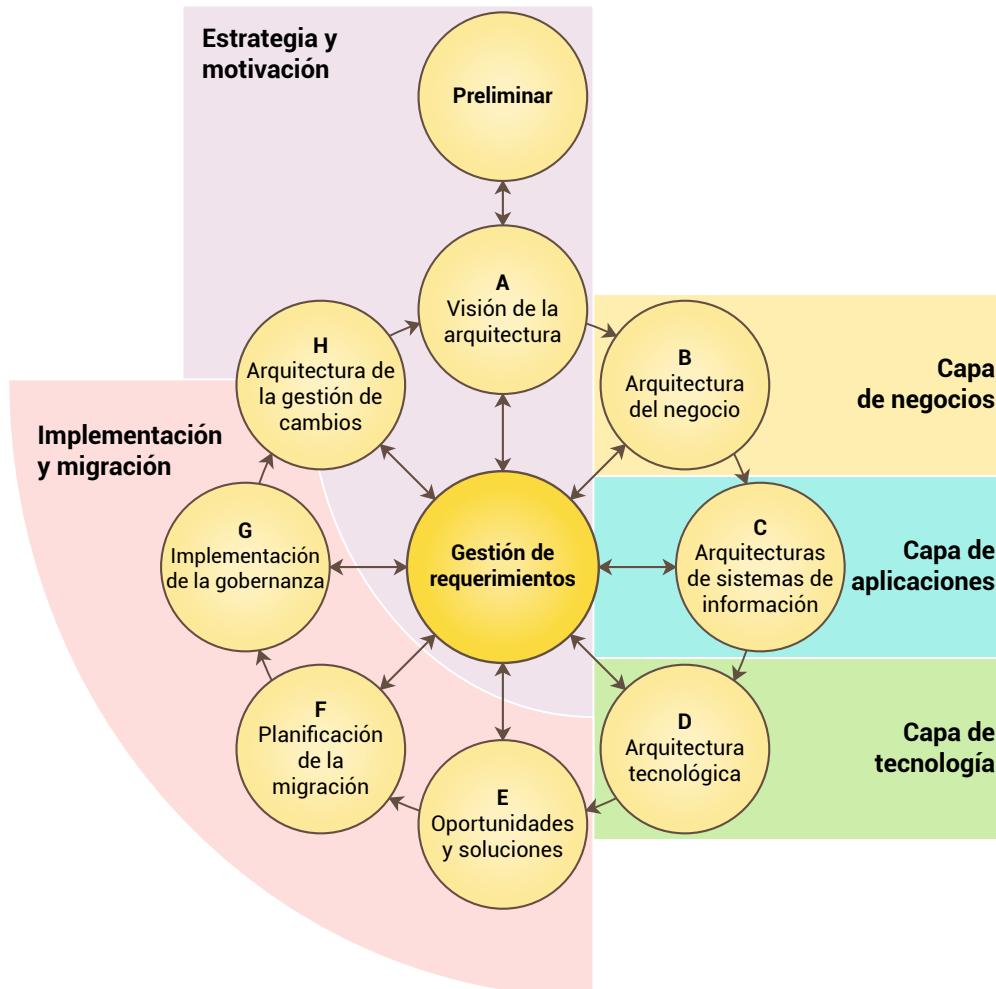
6. Referencias bibliográficas

- Luisi, J. V. (2014). *You have institutional access, Pragmatic Enterprise Architecture.*
- Bhardwaj, M. (19 de Nov de 2018). *LinkedIn.* Obtenido de Application Portfolio Management using TIME: <https://www.linkedin.com/pulse/application-portfolio-management-using-time-manoj-bhardwaj>
- Raymond, P. D. (2014). *Modeling Enterprise Architecture with TOGAF.*
- Stefan Bente, U. B. (2012). *Collaborative Enterprise Architecture.*
- Martin, R. (2008). *Clean Code.*
- Group, O. M. (s.f.). *OMG.* Obtenido de OMG: <https://www.omg.org/>
- Insider, C. M. (23 de Julio de 2020). *Obtenido de Cloud Management Insider:* <https://www.cloudmanagementinsider.com/>
- The Open Group. (2018). *TOGAF Standar Versión 9.2.*



7. Anexos

Anexo 1: Capas de la arquitectura empresarial



Nota. Esta foto de Autor desconocido está bajo licencia CC BY-NC-ND

Anexo 2: Repositorio arquitectónico

Repositorio de la Empresa

Repositorio de la Arquitectura

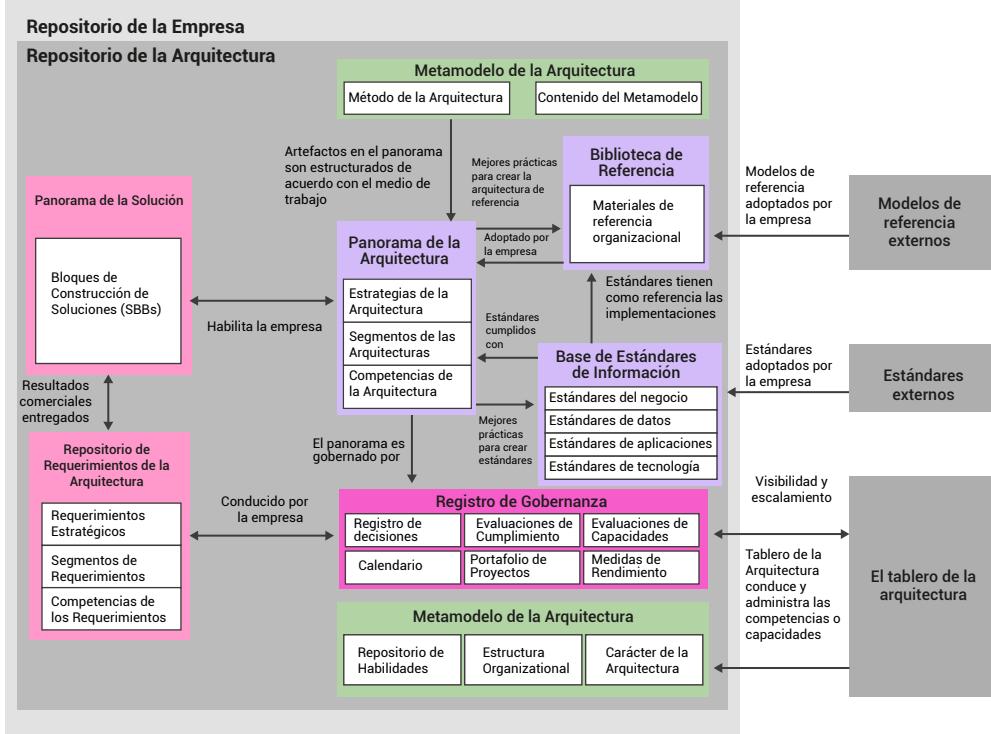


Tabla 1 Fase Preliminar del Ciclo ADM de TOGAF

Tabla 1.

Fase Preliminar del Ciclo ADM de TOGAF

Objetivos

1. Determinar la capacidad arquitectónica deseada por la organización
 - a. Revisar el contexto arquitectónico para determinar la Arquitectura Empresarial.
 - b. Identificar y determinar el alcance de los elementos de la organización (empresa) que serán afectados por las competencias arquitectónicas.
 - c. Identificar los marcos de trabajo, métodos, y procesos utilizados y sus intersecciones con la Capacidad Arquitectónica.
 - d. Establecer la capacidad de madurez objetivo.

Que s

5. Establecer la capacidad arquitectónica
 - a. Definir y establecer el modelo de Arquitectura Empresarial de la Organización.
 - b. Definir, establecer y detallar proceso y recursos para la Gobernanza Arquitectónica.
 - c. Seleccionar e implementar herramientas para soportar las Capacidades de la Arquitectura.
 - d. Definir los principios arquitectónicos.

Entradas

1. Elementos Arquitectónicos

- a. Modelos de arquitectura empresarial en la organización (si existen) y deben contemplar:
 - a. Alcance de los elementos organizacionales impactados.
 - b. Evaluación de madurez, análisis de brechas y enfoque de resolución.
 - c. Roles y responsabilidades para el equipo de definición de la arquitectura.
 - d. Cartera de requerimientos.
 - e. Estrategia de gobernanza y apoyo.
- b. Marco de trabajo de la arquitectura existente (si existe), que debe contemplar.
 - a. Método arquitectónico.
 - a. Contenido de la arquitectura.
 - b. Herramientas configuradas y desplegadas.
 - c. Principios arquitectónicos.
 - d. Repositorio de la arquitectura.

Para la definición de estos elementos se puede utilizar:

- Librería de TOGAF (si se encuentra implementado).
- Otros marcos de referencia de Arquitectura Empresarial (si aplica).

2. Elementos No Arquitectónicos

- a. Tableros de estrategias y tableros de planes de negocio, planificación de TI, principios de negocio como Visión, Misión, Objetivos, y los elementos de planeación estratégica existentes.
- b. Portafolio de gestión de proyectos, documentos de respaldo de la operación del negocio.
- c. Marcos de trabajo de gobernanza, políticas, recursos legales, etc.
- d. Contratos de tercerización, alianzas estratégicas y acuerdos contractuales.

Pasos

El ADM de TOGAF es un método genérico, destinado a ser usado por una amplia variedad de empresas diferentes, y en conjunción con una amplia variedad de otros marcos de trabajo arquitectónicos, si es preciso. Por lo tanto, el enfoque y conjunto de herramientas o procedimientos de arquitectura, estos pasos, pueden seguir diferentes órdenes dependiendo del nivel de detalle de la organización, estructura interna, nivel de madurez u otro tipo de estrategia de Gobernanza Arquitectónica implementada, los pasos podrían ser los siguientes:

1. Alcance de los elementos empresariales en la organización que son o han sido impactados.
2. Marcos de trabajo de gobernanza y apoyo confirmados.
3. Definición del equipo arquitectónico y de la organización que se ha definido.
4. Identificación de los principios arquitectónicos establecidos e identificados.
5. Implementación del marco de trabajo de TOGAF o cualquier otra arquitectura seleccionada.
6. Desarrollar una estrategia o plan de implementación para técnicas y herramientas.

Salidas

Las salidas de esta fase pueden incluir los siguientes elementos:

1. Modelo Organizacional para la Arquitectura Empresarial.
2. Marco de trabajo de la arquitectura implementada.
3. Repositorio inicial de la arquitectura.
4. Reformulación o referencia a los principios de negocio, objetivos del negocio, directrices del negocio.
5. Solicitudes de trabajo arquitectónico.
6. Marco de trabajo de Gobernanza Arquitectónica.
7. Catálogos.

Enfoque

Esta fase define el “dónde, qué, por qué, quién y cómo se hará la arquitectura” en términos de lo concerniente a la arquitectura, los aspectos más importantes son:

1. Definición de la arquitectura.
2. Identificar directrices clave y elementos en el contexto organizacional.
3. Definir los requerimientos para el trabajo arquitectónico.
4. Definir los principios arquitectónicos que informarán cualquier trabajo arquitectónico.
5. Definir el marco de trabajo que será usado.
6. Definir las relaciones entre los marcos de trabajo gestionados.
7. Evaluar la madurez de la Arquitectura Empresarial.

Nota. Adaptado de Preliminary Phase (The Open Group, 2018).

Tabla 2 Fase A - Visión de la Arquitectura

Tabla 2.

Fase A: Visión de la Arquitectura

Objetivos

1. Desarrollar la visión deseada de alto nivel de las capacidades y valor empresarial que es entregada como resultado de la arquitectura empresarial propuesta.
2. Obtener la aprobación de los involucrados para la Declaración de Trabajo de Arquitectura que define el programa de trabajo para desarrollar e implementar la arquitectura delineada en la Visión de la Arquitectura.

Entradas

1. Elementos Arquitectónicos
 - a. Modelos de la organización
 - a. Alcance de las organizaciones o áreas de la empresa afectadas.
 - b. Evaluaciones de madurez, brechas y enfoques de resolución.
 - c. Roles y responsabilidades del o los equipos de trabajo.
 - d. Restricciones para el trabajo de arquitectura.
 - e. Requerimientos reutilizables.
 - f. Cartera de requerimientos.
 - g. Solicitudes de cambio.
 - h. Estrategias de gobernanza y apoyo.
 - b. Marco de trabajo de la arquitectura a medida
 - a. Método arquitectónico a medida.
 - a. Contenido de la arquitectura a medida (entregables y artefactos).
 - b. Principios de la arquitectura.
 - c. Configuración y despliegue de herramientas.
 - b. Repositorio de la arquitectura poblado con documentación de arquitecturas existentes, si aplica, podrían ser los siguientes elementos:
 - a. Descripción del marco de trabajo.
 - b. Descripción de la arquitectura.
 - c. Descripción de la línea base, etc.
 3. Elementos No Arquitectónicos
 - a. Solicitud de trabajo de Arquitectura.
 - b. Principios de negocio, objetivos del negocio, directrices del negocio.

Pasos

El nivel de detalle direccionado en esta fase dependerá del alcance y objetivos de la Solicitud de Trabajo de Arquitectura, o del subconjunto de alcances y objetivos asociados con el desarrollo de iteraciones en el desarrollo de la arquitectura.

El orden y tiempo de ejecución de los pasos, dependerá de la organización establecida por la Gobernanza arquitectónica y de cómo han ido iniciando y finalizado formalmente cada uno de los pasos, estos pasos podrían ser los siguientes:

1. Establecer el proyecto de arquitectura.
2. Identificar involucrados, sus preocupaciones y requerimientos de negocio.
3. Confirmar y elaborar objetivos del negocio, directrices de negocio y restricciones.
4. Evaluar capacidades.
5. Evaluación de la preparación para la transformación del negocio.
6. Definición del alcance.
7. Confirmación y eliminación de los principios de arquitectura.
8. Desarrollo de la visión arquitectónica.
9. Definir la arquitectura objetivo y los indicadores clave de proceso.
10. Identificar riesgos de la transformación tecnológica y sus estrategias de mitigación.
11. Desarrollar y aprobar el documento de Declaración de Trabajo de Arquitectura.

Salidas

Las salidas de esta fase pueden incluir:

1. Documento de Declaración de Trabajo de Arquitectura, debidamente aprobado.
2. Redefinición de las declaraciones de los principios de negocio, objetivos de negocio y directrices de negocio.
3. Principios de la arquitectura.
4. Evaluación de capacidades.
5. Marco de trabajo de la arquitectura a medida.
6. Visión de la arquitectura.
7. Borrador del documento de Definición de la Arquitectura.
8. Plan de comunicación.
9. Matrices de mapeo de involucrados.
10. 1 Diagramas.
 - a. Diagrama del modelo de negocio.
 - b. Mapa de capacidades del negocio.
 - c. Mapa de flujo de valor.
 - d. Diagrama de la cadena de valor.
 - e. Diagramas de conceptos de la solución.

Enfoque

Esta fase inicia con la recepción de la Solicitud de Trabajo de Arquitectura, que es un documento emitido por la sección de la organización que auspicia el trabajo de arquitectura. En este documento se declara de manera clara cuáles son los móviles de la propuesta de trabajo de arquitectura. Adicionalmente, se define qué está dentro y fuera del alcance de la solución considerando los elementos de evaluación de recursos y su disponibilidad.

Otro factor importante que debe ser evaluado en esta fase es el conjunto de restricciones y las estrategias de evaluación y mitigación de potenciales riesgos identificados en la fase anterior (identificación de la visión de la arquitectura).

Esta fase además debe reforzar la creación de la visión arquitectónica y proponerla como una herramienta clave para la exposición de los beneficios de la propuesta de evaluación a todos los interesados y todas las líneas de gestión que influyen en la toma de decisiones dentro de la empresa. La visión arquitectónica describe de manera clara cómo las nuevas capacidades ayudarán a cumplir los objetivos y estrategias del negocio mediante la alineación de todos los elementos y factores neurálgicos del negocio.

Nota. Adaptado de Phase A: Architecture Vision (The Open Group, 2018).

Tabla 3 Fase B - Arquitectura del Negocio

Tabla 3.

Fase B: Arquitectura del Negocio

Objetivos

1. Desarrollar la arquitectura de negocios, objetivo que describa cómo necesita operar la empresa para alcanzar sus objetivos de negocio y responder a las directrices estratégicas planteadas en el documento de visión.
2. Identificar los componentes de la hoja de ruta de la arquitectura candidata basada en la reducción de brechas entre la línea base y las arquitecturas de negocio objetivo.

Entradas

1. Elementos Arquitectónicos
 - a. Modelos de la organización.
 - a. Alcance de las organizaciones o áreas de la empresa afectadas.
 - b. Evaluaciones de madurez, brechas y enfoques de resolución.
 - c. Roles y responsabilidades del o los equipos de trabajo.
 - d. Restricciones para el trabajo de arquitectura.
 - e. Requerimientos reutilizables.
 - f. Cartera de requerimientos.
 - g. Solicitudes de cambio.
 - h. Estrategias de gobernanza y apoyo.
 - b. Marco de trabajo de la arquitectura a medida.
 - a. Método arquitectónico a medida.
 - b. Contenido de la arquitectura a medida (entregables y artefactos).
 - c. Principios de la arquitectura.
 - d. Configuración y despliegue de herramientas.
 - c. Declaración de Trabajo de Arquitectura aprobado.
 - d. Principio de Arquitectura incluyendo principios de negocio pre – existentes.
 - e. Continuum empresarial.
 - f. Repositorio de arquitectura.
 - a. Bloques de construcción re – utilizables.
 - b. Modelos de referencia públicamente disponibles.
 - c. Modelos de referencia de especificación de la organización.
 - d. Estándares de la organización.
 - g. Visión de la arquitectura.
 - a. Descripción del problema.
 - b. Objetivos de la Declaración del Trabajo de Arquitectura.
 - c. Resumen de vistas.
 - d. Escenarios de negocio.
 - e. Requerimientos clave de los involucrados refinados a alto nivel.

Entradas

- h. Borrador del Documento de Definición de Arquitectura
 - a. Línea base de la Arquitectura del Negocio, versión 0.1.
 - b. Línea base de la Arquitectura Tecnológica, versión 0.1.
 - c. Línea base de la Arquitectura de Datos, versión 0.1.
 - d. Línea base de la Arquitectura de Aplicaciones, versión 0.1.
 - e. Arquitectura de Negocios Objetivo, versión 0.1.
 - f. Arquitectura Tecnológica Objetivo, versión 0.1.
 - g. Arquitectura de Datos Objetivo, versión 0.1.
 - h. Arquitectura de Aplicaciones Objetivo, versión 0.1.
- 2.. Elementos No Arquitectónicos
 - a. Solicitud de trabajo de Arquitectura.
 - a. Principios de negocio, objetivos del negocio, directrices del negocio.
 - b. Evaluación de capacidades.
 - c. Plan de comunicación.

Pasos

En esta fase cobra mucha importancia el desarrollo y administración de los artefactos del negocio, siendo estos los que se convierten en elementos temporizadores para el inicio o fin de los diferentes pasos definidos en el proceso. Por lo tanto, los artefactos podrán ser llevados a través del proceso con el propósito de apoyar en el desarrollo del trabajo arquitectónico, es decir, esta fase establece procedimientos que deberán ser ejecutados sin un orden específico, pero deberá ser evaluados de acuerdo con lo establecido en la Gobernanza Arquitectónica y para cumplir su objetivo y necesidades. Al tratarse de procesos que generan documentos, es importante que una vez que se haya finalizado un paso, este esté correctamente documentado y sea publicado para facilitar su acceso y manipulación. Los pasos que se podrían seguir en esta fase son:

1. Seleccionar modelos de referencia, puntos de vista y herramientas.
2. Desarrollar la descripción de la Línea Base de la Arquitectura.
3. Desarrollar la descripción de la Línea Base de la Arquitectura de Negocio Objetivo.
4. Definir los componentes de la hoja de ruta candidata.
5. Resolver los impactos a través del panorama de la arquitectura.
6. Conducir las revisiones formales de artefactos con los involucrados.
7. Finalizar la Arquitectura de Negocio.
8. Crear el Documento de Definición de la Arquitectura.

Salidas

Las salidas de esta fase pueden incluir:

1. Visión de la Arquitectura refinada y actualizada, la cual puede incluir:
 - a. Declaración del Trabajo de Arquitectura, actualizado si es necesario.
 - b. Principios de negocio, objetivos de negocio, directrices de negocio, etc., actualizados si es necesario.
 - c. Principio de la Arquitectura.
 - d. Borrador del Documento de Definición de Arquitectura
 - e. Línea Base de la Arquitectura de Negocio.
 - f. Arquitectura de Negocio Objetivo.
 - g. Borrador del Documento de Especificación de Requerimientos de Arquitectura.
 - h. Hoja de ruta de los componentes arquitectónicos de la Arquitectura de Negocio.
- 2.. Catálogos.
3. Matrices.
9. Diagramas.

Enfoque

La Arquitectura de Negocio es una representación holística, multidimensional de las vistas del negocio en lo relacionado a: capacidades, valor entregado de punto a punto, información y estructura organizacional y todas las relaciones existentes entre las vistas y estrategias del negocio, productos, iniciativas e involucrados.

La Arquitectura de Negocio relaciona los elementos del negocio con los objetivos de este y todos sus elementos en otros dominios.

Nota. Adaptado de *Phase B: Business Architecture (The Open Group, 2018)*.