

INSTITUTO SUPERIOR TECNOLÓGICO



JAPÓN

Amor al conocimiento

GUÍA METODOLÓGICA

INTEGRACIÓN DE
SISTEMAS INFORMÁTICOS
DESARROLLO DE SOFTWARE



COMPILADOR: MSC. DIANA MONCAYO
2019



1. IDENTIFICACIÓN DE

Nombre de la Asignatura: INTEGRACIÓN DE SISTEMAS INFORMÁTICOS	Componentes del Aprendizaje	COGNOSCITIVOS
<p>Resultado del Aprendizaje:</p> <ul style="list-style-type: none">• Conoce la importancia hoy en día de la integración de sistemas informáticos• Aprende sobre las diferentes arquitecturas y su evolución• Conoce aspectos fundamentales de la arquitectura en la nube• Entiende lo referente a servicios web y tecnologías asociadas.• Aprende a través de ejercicios prácticos a proveer y consumir servicios web• Identifica la necesidad de utilización de SOA en las organizaciones actuales• La capacidad de identificar, encontrar y obtener datos.• Adquiere conocimiento integral de métodos y técnicas aplicables y de sus limitaciones.• Conocer estándares y tecnologías de integración de sistemas informáticos• Capacidad para definir la arquitectura de un sistema orientado a servicios.• Capacidad de trabajo en equipo interdisciplinar.		



OBJETIVOS:

- Adquirir los conceptos y estructuras fundamentales de los sistemas de información.
- Reconocer y clasificar los distintos grupos y tipos de aplicaciones informáticas estándar.
- Plantear correctamente las etapas de desarrollo en la automatización de un problema de informática de gestión.
- Comprender la estructura general de una aplicación informática estándar.
- Aplica estrategias de tecnología y estándares adecuados en función de las particularidades de cada proyecto de integración.

COMPETENCIAS

- Capacidad para la integración de tecnologías, aplicaciones, servicios y sistemas propios de la Ingeniería Informática, con carácter generalista, y en contextos más amplios y multidisciplinares.
- Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
- Capacidad para analizar las necesidades de información que se plantean en un entorno y llevar a cabo en todas sus etapas el proceso de construcción de un sistema de información.
- Capacidad para la dirección general, dirección técnica y dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos, en el ámbito de la Ingeniería Informática.
- Capacidad de pensamiento creativo para desarrollar métodos nuevos y originales.
- Capacidad para analizar diferentes casos de uso en el ámbito de la integración atendiendo a las necesidades del negocio.
- Integrar soluciones informáticas para solucionar los problemas



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN
 GUIA DE APRENDIZAJE

Docente de Implementación: Msc. Diana Moncayo				
			Duración: 40 horas	
Unidades	Competencia	Resultados de Aprendizaje	Actividades	Tiempo de Ejecución
UNIDAD I Integración De Sistemas Informáticos	1.1 Introducción a la integración de datos 1.2 Definición de Integración de Datos 1.3 Necesidad de Integrar datos 1.4 Integración de datos en la NUBE 1.5 Fundamentos de la integración	Comprender los fundamentos teóricos de la integración de datos y buscan Integrar soluciones informáticas para ofrecer métodos robustos de TI, para trabajo empresarial.	Presentación de la asignatura, contenidos, normas, evaluación, etc. Desarrollo participativo, En Motivación para la integración de datos Participación Simulación de esquemas	8 horas
UNIDAD II Integración de fuentes de datos	2.1 Integración de fuentes heterogéneas 2.1.1 Emparejamiento de esquemas [schema matching] 2.1.2 Correspondencias entre esquemas [schema mapping] 2.1.3 Gestión de modelos 2.3 Arquitecturas distribuidas 2.3.1 Arquitectura de red Distribuida 2.4 Arquitectura de sistemas 2.4.1.- Niveles lógicos 2.4.2.- Sistemas monolíticos/centralizados 2.4.3.- Arquitectura cliente/servidor 2.4.4 Arquitectura distribuida 2.4.5 Cloud Computing	Identifica la arquitectura de sistemas, como enlazar distintos elementos que forman el sistema, plantea soluciones desde un punto de vista físico como lógico que iría en función de las necesidades de las empresas	Trabajo de arquitectura cliente servidor entre equipos Trabajo en la práctica: Análisis y selección de fuentes de datos a integrar	8 horas



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN
GUIA DE APRENDIZAJE

<p>UNIDAD III Arquitectura en la NUBE</p>	<p>3.1 Introducción 3.2 Soluciones en la Nube 3.3 Servicio SaaS de PaaS y IaaS 3.4 Arquitectura Nube 3.5 Seagate y el Almacenamiento en nube 3.6 Arquitectura orientada a servicios (SOA)</p>	<p>Comprende la importancia de la de una arquitectura en la nube para satisfacer necesidades y requerimientos específicos en cuanto a funcionalidad en , computación y almacenamiento</p>	<p>Trabajo de Investigación y cuestionario</p> <p>Se habilitará en el aula virtual dos cuestionarios de selección múltiple.</p> <p>Análisis técnicas visualización</p> <p>Diseño de la arquitectura en entorno distribuido par almacenamiento y consulta de fuentes. Justificación del diseño</p>	
<p>UNIDAD IV Servicios Web</p>	<p>4.1 Servicio web SOAP 4.1.1 Componentes de servicio web SOAP 4.1.2 Web Services Description Language (WSDL) 4.1.2 Ejemplo de documento WSDL 4.2 Servicios REST 4.2.1 Arquitectura web REST 4.2.2 JAX-RS Client y Servicios REST 4.2.3 JAX-RS Client</p>	<p>Capacidad de manejar aplicaciones de forma estándar a través de protocolos comunes como http y de manera independiente un lenguaje de programación.</p>	<p>Análisis y selección de los servicios web/ y/o herramientas para análisis de las fuentes e integración en entorno distribuido.</p> <p>Prueba objetiva teórica , de tipo objetivo</p>	8 horas
<p>UNIDAD V FRAMEWORKS</p>	<p>5.1 JERSEY 5.1.1 Ejemplo 5.1.2 Apache CXF vs Jersey</p>	<p>Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.</p>	<p>Presentación de proyectos con escenario formal</p> <p>Trabajo final: Arquitectura Web</p>	8 horas



2. CONOCIMIENTOS PREVIOS Y RELACIONADOS

Co-requisitos

- 1.- De acuerdo con el modelo educativo, se busca evidenciar el logro de los resultados de aprendizaje, con el desarrollo de un proyecto que cumpla los mecanismos de planificación, elaboración, diseño, aplicación, ejecución y demostración de un proyecto formativo.
2. El modulo debe profundizar en la formulación de un proyecto con el objetivo de poder identificar la problemática desarrollada, a fin de determinar si es aplicable la creación de un proyecto basado en una metodología de desarrollo.
3. Conocer metodologías de tecnología, para que el estudiante comprenda lo que requiere para iniciar un proyecto, identificando las fases a fin de estructurar de mejor manera el proyecto.
4. Proponer alternativas de iniciativa e innovación tecnológica por parte del Docente, para brindar soluciones al proceso de creación de un proyecto.

3. UNIDADES TEÓRICAS

• Desarrollo de las Unidades de Aprendizaje (contenidos)

A. Base Teórica



UNIDAD 1 :

TEMA 1: INTEGRACIÓN DE SISTEMAS INFORMÁTICOS

1.1 Introducción a la integración de datos

La integración de datos es una combinación de procesos técnicos y de negocio que se utilizan para combinar información de diferentes fuentes para convertirla en datos fiables y valiosos. Estas soluciones de integración ayudan a comprender, limpiar, monitorizar, transformar y entregar datos para que las empresas puedan estar seguras de que la fuente de información es confiable, consistente y está gestionada en tiempo real.

1.2 Definición de Integración de Datos

Es el proceso que permite combinar **datos heterogéneos** de muchas fuentes diferentes en la forma y estructura de una única aplicación. Esto facilita que diferentes tipos de información, tales como matrices de datos, documentos y tablas, sean fusionados por usuarios, organizaciones y aplicaciones para un uso personal, de procesos de negocio o de funciones.

Un proyecto de integración de datos generalmente implica los siguientes pasos:

- **Acceso a los datos** desde todas las fuentes y localizaciones tanto si se trata de locales, en la nube o de una combinación de ambos.
- **Integración de datos** de modo que los registros de una fuente de datos mapean registros en otra. Por ejemplo, incluso si un conjunto de datos utilizara “nombre, apellidos” y otro “nom, ape”, el conjunto integrado se asegurará de que en ambos casos los datos van al lugar correcto. Se trata de un tipo de preparación de datos esencial para que las analíticas y otras aplicaciones sean capaces de utilizar los datos con éxito.
- **Entrega de datos integrados** al negocio justo en el momento en que la empresa los necesita, ya sea por lotes, casi en tiempo real o en tiempo real.

1.3 Necesidad de Integrar datos.

El mundo empresarial se está centrando cada vez más en el consumidor. Enfocarse en el servicio al cliente y escucharle para obtener feedback era vital en el pasado, pero los



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

negocios de hoy en día necesitan alcanzar ideas más profundas de lo que los clientes quieren, recopilando datos que van desde el uso de patrones con productos, a publicaciones en medios sociales.

El uso de datos para tener expectativas más claras de las demandas de los clientes es vital, y hacerlo de manera rentable es esencial. Sin embargo, muchas empresas carecen de las herramientas para hacerlo. Un estudio de Experian encontró que en los departamentos de marketing, apenas un 30% de las organizaciones siente que tiene una buena integración de datos.

Estas son las 5 razones por las que una empresa debe poner el foco en integración de datos:

1. **Reduce la carga sobre los analistas de negocios.** Los profesionales del Business Intelligence se enfrentan a una carga de trabajo abrumadora tratando de filtrar las enormes cantidades de datos que entran a la empresa en el día a día. La eliminación de los silos de datos permite a los usuarios acceder a diferentes conjuntos de información basada en sus necesidades específicas. Dar a los equipos acceso directo a información relevante deja a los analistas con un tema menos del que preocuparse, permitiéndoles dedicarse a conjuntos de datos más complejos que generan valor para el negocio.
2. **Eliminar doble trabajo.** En muchas ocasiones, algunas organizaciones realizan un análisis de clientes para conocer sus demandas, y poco después descubren que un proyecto similar se completó hace unos meses en otro departamento. Con una integración de datos se puede evitar esa redundancia y no sólo en términos de grandes proyectos. Los problemas más comunes a los que se enfrentan las empresas tienen que ver con registros de datos de clientes en múltiples lugares, documentación de procesos en distintos sistemas, etc.
3. **Maximiza el valor de los datos.** Un análisis que puede ser eficaz para 20 de los empleados será considerablemente menos valioso si sólo cinco de ellos reciben el informe. Unificar datos a través de distintos canales permite a las organizaciones aprovechar distintos tipos de datos conjuntamente con otros para maximizar su potencial y garantizar que los grupos de usuarios tienen la visibilidad que necesitan. Esta transparencia puede extenderse a interesados tanto internos como externos impulsando la colaboración dentro de la empresa.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

4. **Mejora la toma de decisiones.** Dar a los usuarios acceso a datos clave incorporados en las aplicaciones y servicios que utilizan les permite tomar mejores decisiones al interactuar con clientes y colaboradores. La integración de datos en los sistemas relevantes hace que la información sea accionable en las operaciones diarias, proporcionando a los usuarios las ideas que necesitan para trabajar con la mayor inteligencia posible.
5. **Aprovechar los diversos tipos de datos.** Si bien el aprovechamiento de tipos de datos variados está relacionado con la maximización del valor de los datos, es importante reconocer que los distintos tipos de información crean desafíos únicos. La información de las hojas de cálculo, las bases de datos altamente estructuradas, informes de social media, diagramas, documentos técnicos, y una amplia gama de otras fuentes, deben unirse para obtener información completa de las operaciones, especialmente cuando tecnologías emergentes como el internet de las cosas traen todavía más datos a los ecosistemas empresariales. Esta diversidad puede llegar fácilmente a ser abrumadora y hacer que los datos se dejen abandonados si los usuarios no tienen acceso a ellos para aprovechar la información intuitivamente a través de sus departamentos.

1.4 integración de datos en la NUBE.

En un sistema de computación en nube, hay un cambio significativo en la carga de trabajo. Los ordenadores locales ya no tienen que hacer todo el trabajo pesado cuando se trata de ejecutar aplicaciones. La única cosa que el equipo del usuario debe ser capaz de ejecutar es el software de interfaz de sistemas de computación en nube, que puede ser tan simple como un navegador web y red se ocupa del resto.

A medida que la tecnología de la nube madura, la habilidad para transferir datos y procesos complejos a entornos Cloud mejora. Esto permite a las empresas integrar aplicaciones para obtener eficiencias inmediatas, al mismo tiempo que se disfruta de una mejor gestión de la información.

Una arquitectura de datos apropiadas que pueda soportar soluciones in situ, cloud y soluciones híbridas ayudará a controlar los costes, ya se trate de licencias, almacenamiento,



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

integración de datos o ancho de banda. Además, una arquitectura de nube alineada con el negocio aumenta la escalabilidad y la reutilización. Esto, a su vez, facilita las tareas de los trabajadores y mejora su capacidad para satisfacer las necesidades cambiantes del negocio.

La integración de datos en la nube ofrece las siguientes ventajas sobre los métodos más antiguos:

- Todos los usuarios pueden **acceder a datos personales en tiempo real desde cualquier dispositivo** con conexión a Internet.
- Es posible **integrar datos personales** tales como calendarios y listas de contactos ofrecidos por diferentes aplicaciones.
- Se puede **utilizar la misma información de inicio de sesión** para todas las aplicaciones personales.
- El sistema pasa eficientemente los **mensajes de control entre las aplicaciones**.
- Se evita el uso de silos de datos, **se mantiene la integridad de la información** y se eliminan conflictos de datos que pueden surgir de la redundancia.
- La integración de datos en la nube ofrece **escalabilidad para permitir una expansión futura** en términos de número de usuarios y de aplicaciones.

1.5 Fundamentos de la integración

Los responsables que han sido puestos a cargo de un proyecto de integración de datos a menudo no están seguros de por dónde comenzar ni qué hacer. Por ello, existen varios fundamentos que marcarán el punto de partida para saber cómo abordar este proceso:

- **Los metadatos lo son todo.** El origen de los datos guiará la forma en la que se debe iniciar la integración de los datos. La empresa debe comprender la información almacenada en los sistemas de origen y destino y encontrar una sola fuente fiable y verdadera.
- **Entender el flujo de información.** Una vez identificada la fuente, es necesario averiguar cómo los datos fluirán de un sistema a otro. Si bien la mayoría de los flujos de integración de datos son una replicación simple, también es posible cambiar la estructura



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

y el contenido de los información que fluye de un sistema a otro, para que la infraestructura de destino reciba datos nativos.

- **Unir Seguridad y Data Governance.** Son dos elementos que a menudo no se complementan bien en los entornos de integración de datos. Este problema se vuelve más importante a medida que avanzamos hacia la nube, ya que los datos a están físicamente fuera de nuestro control. Los encargados de la integración deben cifrar los datos, y una vez encriptados, la información será más segura.

En el contexto de la integración de datos, Data Governance implica la utilización políticas activas en torno al uso de datos, flujos, transformaciones, etc. Esto nos permite evitar que alguien cambie un flujo o varíe la estructura de un sistema objetivo y rompa la solución de integración.

Tanto si se trata de la primera vez que una organización realiza una integración de datos como si ya es un sistema consolidado, los fundamentos de la integración siguen siendo la clave para el éxito de su desarrollo.

UNIDAD 2

TEMA 1: Integración de fuentes de datos

2.1 Integración de fuentes heterogéneas

A continuación se presenta una alternativa a la integración de datos de fuentes textuales hacia una base de datos orientada a objetos para el dominio de equipos de computación. La base de datos integrada tiene el propósito de ser útil para emitir sugerencias de equipo de cómputo a diferentes usuarios con base en el equipo que mejor satisfaga sus requerimientos. En este caso es muy importante que la base de datos contenga los datos de equipos más actualizados sin importar el proveedor, esto dificulta que puedan establecerse acuerdos para solicitar los datos que se incorporan en la base de datos; como consecuencia, se considera más apropiado integrar los datos de las fuentes textuales hacia la base de datos marcando el proveedor, que en este trabajo se indica como un dato anónimo por cuestiones de derechos de autor de sus marcas registradas.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

La propuesta de “Benina Velázquez Ordoñez, Jesús Manuel Olivares Ceja, Miguel Patiño Ortíz, Julián Patiño Ortíz, Adolfo Guzmán Arenas” trabajo es una un módulo integrador de datos que toma las fuentes textuales heterogéneas que deben cumplir con una estructura de encabezado, donde se identifica el nombre de la clase principal y un cuerpo de documento donde se esperan las propiedades, valores y unidades de medida. Algunas de las propiedades pueden ser objetos que componen a la clase principal. Se realiza un análisis léxico de los elementos de las fuentes textuales para identificar los elementos que coinciden con la base de datos imagen1 El integrador utiliza un diccionario y un conjunto de reglas para encontrar los mapeos entre las palabras de un documento de texto con los elementos de la base de datos.

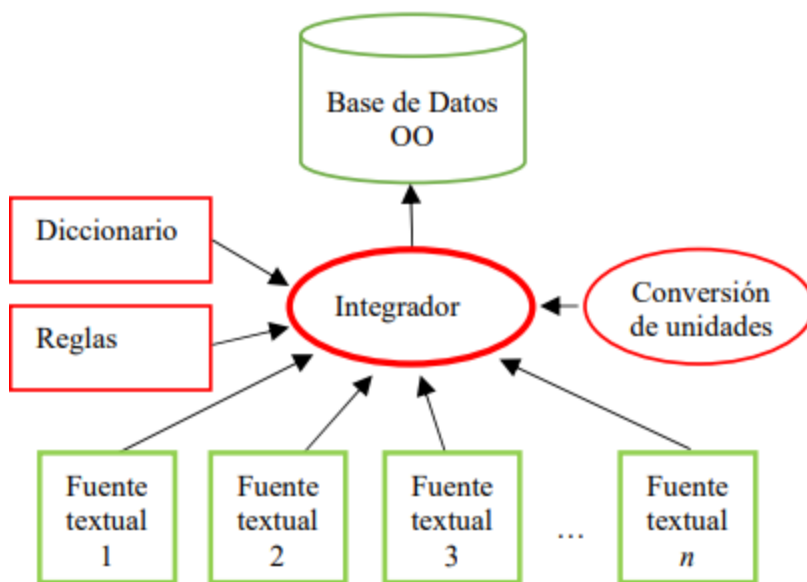


Imagen 1: Integración de fuentes textuales hacia una base de datos

Fuente: "<http://www.scielo.org.mx/pdf/poli/n51/n51a4.pdf>"

El modelo semántico de datos, tiene la ventaja de utilizar abstracciones cercanas al usuario facilitando la identificación de las principales entidades (clases) con un triángulo y a partir de las que se derivan las subclases identificadas con un círculo. Las clases formadas con la agrupación de propiedades (variables) se indica con un círculo con una cruz, los atributos



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

se indican con un ovalo y cuando son atributos multivaluados se preceden de un circulo con doble cruz.

En la imagen 2 , se utiliza un modelo semántico para mostrar algunas de las entidades principales de la base de datos de ejemplo. A partir del modelo semántico se obtiene el modelo orientado a objetos que como diferencia del anterior las clases y superclases utilizan el mismo símbolo con lo cual se destacan las relaciones de herencia, composición, agregación y asociación.

En la imagen 3 se indican las clases que destacan las relaciones de la clase computadora con sus componentes y sus relaciones de herencia.

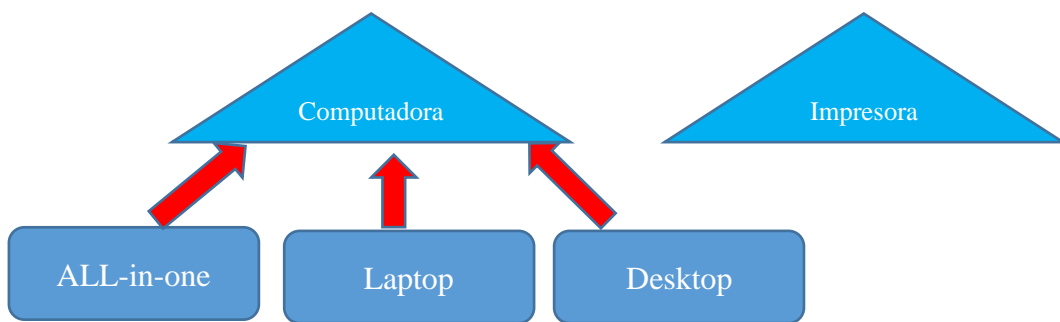


Imagen 2: Modelo semántico de la base de datos de ejemplo
Fuente: Desarrollo propio

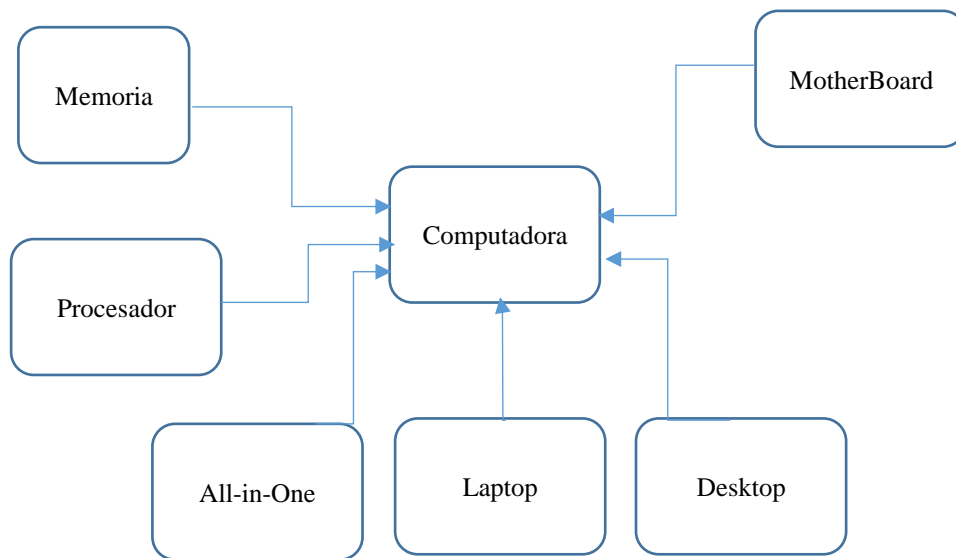


Imagen 3: Algunas clases del modelo orientado a objetos
Fuente: Desarrollo propio



2.1.1 Emparejamiento de esquemas [schema matching]

Schema matching tiene como objetivo identificar correspondencias semánticas entre estructuras de metadatos Figuras o modelos, como esquemas de bases de datos, formatos de mensajes XML y ontologías.

TÉCNICAS DE EMPAREJAMIENTO

Esquemas Matriz de similitud Entradas Dos esquemas S y T + Información potencialmente útil (datos reales, descripciones textuales...)

Salida Matriz de similitud que asigna a cada par de SxT un número entre 0 y 1 en función de su nivel de emparejamiento.

Familias de técnicas:

- Emparejamiento de nombres
- Emparejamiento de datos

Algoritmos sobre cadenas p.ej.

Distancia de edición

$$d(\text{"data mining"}, \text{"data minino"}) = 1$$

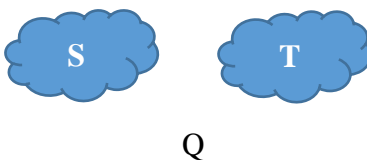
$$d(\text{"efecto"}, \text{"defecto"}) = 1$$

$$d(\text{"poda"}, \text{"boda"}) = 1$$

$$d(\text{"night"}, \text{"natch"}) =$$

$$d(\text{"natch"}, \text{"noche"}) = 3$$

Para comprender el emparejamiento de esquemas, se va a realizar unos ejemplos considerando declarar dos esquemas relacionales S y T.



Establecer la correspondencia más adecuada entre distintos esquemas es el problema más difícil en integración de datos (heterogeneidad semántica).



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

Una correspondencia semántica es una expresión (una consulta) que relaciona el esquema S con el esquema T.

Imaginemos que estamos diseñando el esquema integrado de un sistema de integración de datos con varias fuentes de datos:

- GAV [Global-as-View]: Describimos el esquema integrado como consultas sobre las fuentes de datos.
- LAV [Local-as-View]: Describimos las fuentes de datos como consultas sobre el esquema integrado.
- GLAV [Global-and-Local-as-View]: Hay que establecer correspondencias en ambos sentidos.

Ejemplo

```
DVD-VENDOR
Movies(id, title, year)
Products(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
Locations(lid, name, taxRate)

AGGREGATOR
Items(name, releaseInfo, classification, price)
```

Items

```
SELECT title AS name,
       releaseDate AS releaseInfo,
       rating AS classification,
       basePrice * (1 + taxRate) AS price
FROM Movies, Products, Locations
WHERE Movies.id = Products.mid
      AND Products.saleLocID = Locations.lid
```

Imagen 4: Ejemplo referencial

Fuente: <https://elvex.ugr.es/>

Fase 1: SEMANTIC MATCHING

Relaciones de conjuntos de elementos de un esquema S con conjuntos de elementos de otro esquema T (elicitadas usando conocimiento del dominio).

Emparejamientos uno a uno:

$Movies.title = Items.name$ $Products.rating = Items.classification$

Emparejamientos uno a muchos:

$Items.price = Products.basePrice * (1 + Locations.taxRate)$

2.1.2 Correspondencias entre esquemas [schema mapping]

A partir de las relaciones funcionales identificadas, se elaboran las consultas necesarias para establecer la correspondencia entre esquemas (p.ej. usando SQL).

Emparejamiento:

$Items.price = Products.basePrice * (1 + Locations.taxRate)$



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

Correspondencia:

```
SELECT (basePrice * (1 + taxRate)) AS Price  
FROM Product, Location  
WHERE Product.saleLocID = Location.lid
```

2.3 Arquitecturas Distribuidas

Son un conjunto de ordenadores que trabajan juntos de forma coordinada, a través del intercambio de mensajes. Para redes con muchos dispositivos, donde hay una cantidad muy grande de flujos, es implementada la solución de Arquitectura Distribuida. Se trata de un sistema de gestión en el cual son utilizados dispositivos llamados de colectores y consolidadores que colaboran para una mejor performance en el procesamiento y en la capacidad de colecta de informaciones, proporcionando una supervisión ágil y eficiente de toda la red, sin la pérdida de datos importantes.

Los colectores no son simples repasadores de información. Se trata de appliances que forman una arquitectura en cluster, y que actúan en el procesamiento de datos brutos, transformándolos en datos resumidos, que son tratados a partir de las necesidades del cliente para la obtención del análisis deseado. A partir de la recogida y del procesamiento en cada colector, las informaciones son transferidas para el dispositivo consolidador, appliance donde se concentran todos los datos ya procesados, para entregarlos al usuario para su completa visualización y análisis en las herramientas TRAFip y/o SLAview.

2.3.1 Arquitectura de red Distribuida

Una “empresa” es una compañía u organización conformada por dos o más instalaciones, ya sea en una misma ubicación o separadas entre sí. Una empresa podrían ser varias instalaciones agrupadas en la misma área, por ejemplo, un complejo hospitalario, en donde el edificio principal está rodeado de clínicas, consultorios, etc. O, podría ser una gran compañía multinacional compuesta por muchos edificios que pueden estar a cientos, incluso miles, de kilómetros entre sí.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

Los principales elementos de una arquitectura de red distribuida son

- I) la distribución de la toma de decisiones y del control a cada sede, y al mismo tiempo
- II) la conexión en red y sincronización simultáneas de las distintas sedes, a través de un centro de distribución.

Lo que quieren los usuarios finales y los integradores

La distribución de la toma de decisiones y del control a cada sede es esencial por dos razones. En primer lugar, los gerentes de las sedes necesitan flexibilidad para gestionar las necesidades de seguridad específicas de su sede. Desde la adición de personal hasta la modificación de derechos de acceso, es necesario que tengan la capacidad de administrar su sistema día a día, sin temor a perder conectividad a la red o ancho de banda con un servidor central externo.

En segundo lugar, desde el punto de vista de la escalabilidad, una arquitectura bien diseñada evita la transmisión innecesaria de datos y el consumo excesivo de banda ancha; por ejemplo, agregar un nuevo contratista y asignarle privilegios de acceso en una sede única no debería requerir comunicación con un servidor central.

Si bien el control de las sedes es fundamental, las empresas, tal y como se definieron más arriba, también necesitan una sincronización de los datos de seguridad entre las sedes que sea fácil de configurar. Los factores fundamentales para una arquitectura de red son:

1. Sincronización instantánea y programada: los directores de seguridad necesitan recibir notificaciones en tiempo real sobre eventos y alarmas críticas y deben tener la capacidad de programar la sincronización de actividades no críticas para administrar mejor la utilización máxima de la red;
2. Tolerancia a interrupciones de la red: la sincronización entre múltiples sedes debe poder soportar interrupciones breves o prolongadas de la red. Cuando la sede o el



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

servidor central vuelvan a conectarse, la sincronización debe reiniciarse automáticamente;

3. “Escalabilidad” de lo grande a lo pequeño: en la mayoría de las empresas hay una combinación de pequeñas sedes por ejemplo, cuatro o cinco oficinas de ventas— y de grandes edificios de oficinas. Una verdadera arquitectura empresarial debe incorporar a la red todas las sedes, grandes y pequeñas.

Una arquitectura de red distribuida bien diseñada también ofrece a los directores de seguridad empresarial y de TI potentes funciones de administración centralizada. La gestión centralizada del personal –definición y modificación de privilegios “globales” de control de acceso, edición de detalles del personal – es el punto de partida más importante.

No obstante, la verdadera gestión centralizada va más allá del personal. Los directores de seguridad también necesitan tener la capacidad de i) crear un visualizador integrado de eventos en varias sedes; y de ii) administrar, monitorear y configurar de forma centralizada sus paneles y lectores.

Para satisfacer las necesidades de IT, los distintos servidores independientes de cada sitio de una arquitectura de red distribuida deben poder comunicarse con el centro de distribución a través, ya sea de una red de área local (LAN) o de una red de área extendida (WAN).

2.4 Arquitectura de sistemas

La **arquitectura de un sistema** trata de describir, tanto desde un punto de vista físico como lógico, la forma en la que trabajan los diferentes componentes del mismo. Existen diferentes tipos de **arquitecturas de sistemas** cuya estructura varía en función de las necesidades de las empresas y del momento histórico en el que se introdujeron. Vamos a describir de forma esquemática las visiones **lógica** y **física** de las mismas.

A continuación se detallan las arquitecturas de sistemas:

2.4.1 Niveles lógicos



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

Con independencia de la implementación física, se puede realizar una subdivisión lógica de los sistemas de información en tres niveles:

- **Interfaz de usuario:** nivel de presentación donde reside la lógica de presentación e interacción con el usuario.
- **La lógica de negocio:** donde residen las reglas de negocio y validación. Es el núcleo funcional de la aplicación.
- **El acceso a los datos:** nivel encargado de la persistencia e integridad de la información en el sistema.

Esta subdivisión es fundamental para entender la evolución de las arquitecturas.

2.4.2 Sistemas monolíticos/centralizados

Su implementación física consiste en disponer de un gran ordenador central (*mainframe*) que sirve a cientos o miles de usuarios conectados al mismo a través de una pantalla “tonta” (*dumb terminal*) que se utiliza para entrar o actualizar datos y acceder a información en el *mainframe*. Tanto la interfaz de usuario como las reglas de negocio y los datos residen en la misma máquina. Se produce, en consecuencia, un **acoplamiento** de niveles.

IBM popularizó en los setenta este tipo de **arquitectura** para satisfacer las necesidades de procesamiento grandes corporaciones. Sin embargo, en la actualidad muy pocas organizaciones utilizan exclusivamente este tipo arquitectura basada en un único ordenador central.



Imagen 3: Sistema monolítico/centralizado

Fuente: <https://www.iedge.eu/>



2.4.3 Arquitectura cliente/servidor

A partir de mediados de los ochenta se fue extendiendo el uso de terminales “**inteligentes**” en forma de **PC** de sobremesa o portátiles que, además de teclado y pantalla, incorporan elevadas capacidades de proceso y almacenaje, de modo que los usuarios pueden procesar información localmente, de forma descentralizada y autónoma respecto al *mainframe* u ordenador central. Comienzan a implementarse numerosas *redes* locales, departamentales y corporativas.

En esta arquitectura tanto interfaz de usuario, basada en **Windows**, como la lógica de negocio de las aplicaciones reside en las máquinas “cliente”. Los datos residen en el servidor y son compartidos por todas las estaciones cliente.

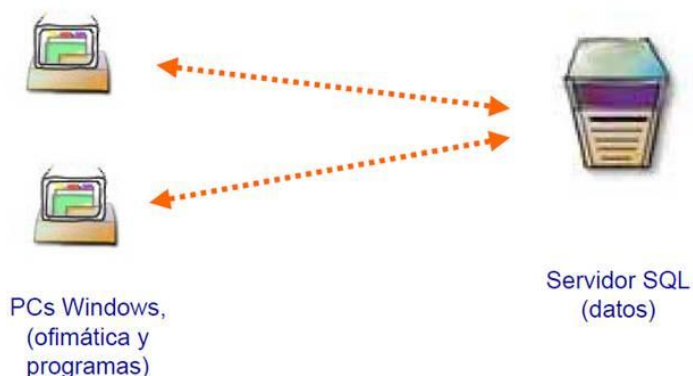


Imagen 4: Arquitectura cliente/servidor

Fuente: <https://www.iedge.eu/>

2.4.4 Arquitectura distribuida

A mediados de los noventa con el uso generalizado de Internet surge la arquitectura distribuida. Se establece una división entre los distintos niveles lógicos, necesaria para soportar las nuevas arquitecturas Internet. Implementación de dicha separación se consigue con la incorporación de un nuevo tipo de servidor: *el servidor de aplicaciones*. La lógica de negocio, que en la **arquitectura cliente/servidor** residía en el PC, es ejecutada ahora en los servidores de aplicaciones.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

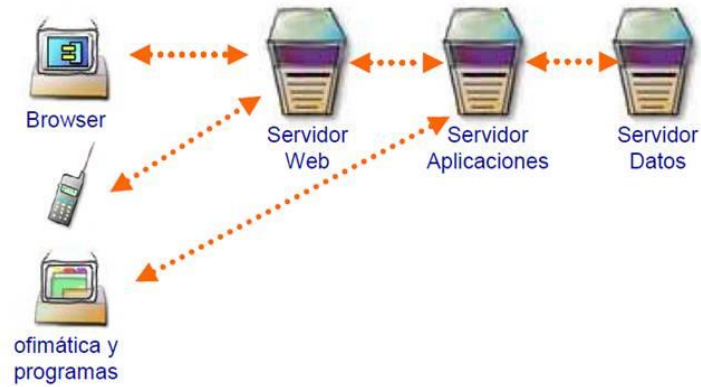


Imagen 5: Arquitectura distribuida

Fuente: <https://www.iedge.eu/>

Como vemos en la ilustración, en la arquitectura distribuida cada nivel lógico dispone de un tipo de servidor físico especializado:

- **Interfaz de usuario -> Servidor web**
- **La lógica de negocio -> Servidor de aplicaciones**
- **Acceso a los datos -> Servidor de base de datos**

2.4.5 Cloud Computing

Es una variante de cualquiera de los casos anteriores, pero con el servidor “en la nube” en lugar de estar físicamente en la oficina de la empresa. Ya no hablamos de servidor en red local, sino de servidor “remoto”, aunque para los usuarios debe ser indiferente que el servidor esté en la habitación de al lado o en un CPD (centro de proceso de datos) de Irlanda. Tiene un gran atractivo porque la empresa se desentiende del mantenimiento del servidor, incluyendo su ampliación si se queda pequeño, pero tiene un gran problema para sistemas de gestión documental: **el ancho de banda**. Los usuarios dependen completamente de la velocidad de su conexión a Internet para consultar archivos y añadir nuevos. Y algunos archivos pueden ser bastante “pesados”. Para PYMES con una sola oficina no parece justificada esta dependencia absoluta de la conexión a Internet, frente a la velocidad que proporciona una conexión en red local al servidor situado en la habitación contigua.



UNIDAD 3

TEMA 1: Arquitectura en la NUBE

3.1 Introducción

La computación en nube y el almacenamiento en nube se han convertido en temas relevantes para abordar problemas y retos comunes de informática a la vez que brindan nuevas oportunidades. Para algunos entornos, el fin principal es reducir costos, mientras que para otros es respaldar el crecimiento. Además, algunos ambientes necesitan mejorar los objetivos a nivel de servicio (SLOs) así como cumplir con los acuerdos a nivel de servicio (SLA, por sus siglas en inglés) sobre disponibilidad, rendimiento, seguridad y protección de los datos.

3.2 Soluciones en la Nube

Las soluciones en nube son herramientas para crear y almacenar contenido o información, así como estrategias sobre dónde y cómo consumirlo. Estas soluciones se utilizan para crear infraestructuras virtuales para organizaciones grandes y pequeñas para instalar aplicaciones o funciones de negocios, así como un lugar para desarrollar y crear nuevas capacidades. Adicionalmente, incluyen servicios o productos de pago inmediato (hardware, software y redes) y soluciones que usted puede adquirir para instalar en su entorno específico.

A continuación, se presentan algunos términos y frases comunes relacionados con las soluciones en nube:

- **Optimizadas y rentables:** alinea los recursos con los SLO para cumplir con los SLA
- **Menú de las opciones de servicio para seleccionar de:** jerarquización de recursos alineados con el costo y los SLA
- **Elástica y de tamaño adaptable con estabilidad:** respalda el crecimiento sin añadir complejidad
- **Elástica, flexible y dinámica:** se adapta a las necesidades cambiantes y puede estar disponible
- **Abastecimiento rápido o autoabastecimiento:** acceso a recursos y servicios con prontitud



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN

GUIA DE APRENDIZAJE

- **Segura y de arrendamiento múltiple:** separación segura de usuarios mientras se mantiene la integridad de los datos
- **Medida y administrada:** métrica para el reporte, análisis y manejo de los servicios
- **Escala de densidad:** aprovechar el multi-arrendamiento y las economías de escala para reducir costos

3.3 Servicio SaaS de PaaS y IaaS

Igualmente, Software como un servicio (SaaS) que se consume a través de soluciones en nube incluye entretenimiento personal o del consumidor (Netflix), noticias y redes de medios sociales (Facebook, Skype y Twitter), fotos compartidas, archivos compartidos (Dropbox), correo electrónico, música y servicios de respaldo en línea.

Además de ofrecer diversas funcionalidades para los consumidores, también los negocios grandes y pequeños utilizan soluciones en nube para mejorar la productividad. Por ejemplo, documentos compartidos (Google Docs), manejo de la relación con el cliente o CRM (Salesforce.com), reporte de gastos (Concur), nómina (ADP), correo electrónico, archivos compartidos, respaldo y archivado. Además de introducir SaaS, los proveedores de la nube también ofrecen herramientas y entornos para Plataforma como un servicio (PaaS) para respaldar el desarrollo y crear servicios SaaS entre otros.

Los tipos de capas de almacenamiento para Infraestructura como un Servicio (IaaS) incluyen capacidades como Web o máquinas virtuales (VMs), almacenamiento para archivo compartido en línea, respaldo o archivado, base de datos, búsqueda y herramientas de desarrollo. Estas capacidades permiten a los proveedores de la nube o a terceras partes crear soluciones individualizadas al combinar las diversas funcionalidades en nube o capas en servicios prestados.

Las soluciones de almacenamiento en nube de SaaS incluyen archivo, documento, música, foto y video compartido, respaldo/almacenamiento, BC y DR, junto con las capacidades de archivado. Otras opciones de almacenamiento en nube incluyen base de datos, análisis de datos grandes (incluyendo Hadoop y servicios basados en reducción de mapas), discos duros en nube y otras aplicaciones que aprovechan el back-end del almacenamiento en nube. Las soluciones de almacenamiento en nube también se extienden a productos y soluciones empleados para hacer uso de nubes públicas, privadas e híbridas.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN

GUIA DE APRENDIZAJE

Los productos y soluciones son los bloques de construcción de servicio de almacenamiento en nube más comunes de los sistemas de almacenamiento físico. La nube privada y los servicios públicos desde SaaS hasta PaaS y IaaS aprovechan el almacenamiento en niveles incluidas unidades de estado sólido (SSD) y unidades de disco duro (HDDs). Al igual que los entornos de almacenamiento empresarial tradicionales, los servicios en nube y los proveedores de servicio aprovechan una mezcla de niveles tecnológicos de almacenamiento diferentes que satisfacen diversos requerimientos de SLO y SLA. Por ejemplo, utilizar unidades de estado sólido rápidas para consolidación de E/S densa, respaldar diarios e índices de bases de datos, metadatos para una búsqueda rápida y otros datos transaccionales, permiten que se realice mayor cantidad de trabajo con menos energía en un espacio físico más denso y más rentable.

Utilizar una mezcla de SSD ultrarápidos junto con HDD de alta capacidad ofrece un balance del rendimiento y la capacidad para satisfacer los requerimientos de otro servicio con diferentes opciones de costo del servicio.

Con los servicios en nube, en lugar de especificar qué tipo de unidad física adquirir, los proveedores de la nube se ocupan de esto al ofrecer diversas opciones de disponibilidad, costo, capacidad, funcionalidad y rendimiento para satisfacer los requerimientos de SLA y SLO.

3.4 Arquitectura Nube

En el núcleo de la informática heredada, patrocinantes, proveedor de servicios administrado (MSP) y nubes se encuentran los bloques de construcción que incluyen redes y tecnologías de procesamiento y almacenamiento.

Diferentes tipos de servidores, redes y tecnologías de almacenamiento cumplen con los diversos requerimientos de computación en nube y almacenamiento en nube (servidores densos de estante y blade con diferentes números de receptáculos y núcleos a diversas velocidades GHz, roscas, cantidad de memoria y capacidades de expansión de E/S son sólo algunos ejemplos). Las opciones de redes incluyen redes veloces de 40 GbE y 100 GbE para circuitos de enlaces troncales, conjuntamente con el 10 GbE y 1 GbE más común para redes privadas virtuales (VPN) y optimización de ancho de banda.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

Las opciones o niveles de almacenamiento de datos incluyen SSD ultrarápidos, así como HDD de velocidad media y de alta capacidad. Las características de administración de almacenamiento incluyen protección de datos, alta disponibilidad (HA), respaldo (BC) y recuperación de desastres (DR), así como la reducción del espacio físico (DFR) para optimización del espacio, como compresión, de duplicación y aprovisionamiento dinámico, que permite mayor almacenamiento de información por períodos más largos a costos menores.

Las herramientas software también son muy importantes para crear servicios y soluciones, e incluyen API, software intermedio, bases de datos, aplicaciones, hipervisores para crear máquinas virtuales (VMs) e infraestructuras virtuales de sobremesa (VDI), conjuntamente con stackware en nube, como OpenStack y herramientas de administración asociadas. Ejemplos de hipervisores VMs y VDI incluyen Citrix/Xen, KVM, Microsoft Hyper-V, Oracle y VMware ESX/vSphere.

En los tres casos, el almacenamiento de datos es configurado en sistemas de almacenamiento, aplicaciones de almacenamiento y servidores de computación.

Se puede tener acceso a los servicios de las nubes públicas gratuitamente o mediante el pago de una tarifa que ofrezca diferentes funcionalidades, como los Servicios Web Amazon (AWS), Google Docs o el software de respaldo de datos de Seagate® EVault®. Las nubes públicas son controladas por sus respectivos propietarios, cuyos clientes eligen utilizar sus servicios. Por otra parte, las nubes privadas son propiedad o son operadas y controladas por organizaciones y son similares a los servicios ofrecidos por informática heredada. Sin embargo, obsérvese que las nubes privadas construidas utilizando componentes o servicios disponibles públicamente e instalaciones externas existentes en diferentes ubicaciones del proveedor en nube son conocidas como nubes híbridas.

3.5 Seagate y el Almacenamiento en nube

Seagate se conoce como el líder en almacenamiento empresarial y, evidentemente, también se encuentra en el núcleo de la infraestructura de la nube. Seagate, haciendo uso de décadas de experiencia en alta densidad, empresas de gran escala, instituciones, agencias



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

gubernamentales, servicios administrados y entornos de ubicación conjunta, aplica este conocimiento a los entornos de nubes públicas y privadas. Además de ser el líder de la industria en tecnología de almacenamiento, Seagate posee décadas de experiencia trabajando con numerosos socios y sus respectivas soluciones de almacenamiento, empaques, chasis y carcasas, así como procesos de prueba y verificación.

Como el surtidor clave para nubes privadas y públicas y proveedores de servicios administrados, la tecnología de Seagate se encuentra en entornos empresariales y centros de datos en nube, así como en proveedores de servicios administrados hasta pequeñas empresas y consumidores. En otras palabras, Seagate ha hecho posible la computación en nube y el almacenamiento en nube desde el centro de datos hasta su bolsillo durante algún tiempo.

Las opciones de almacenamiento de la computación en nube y los entornos de almacenamiento en nube de Seagate incluyen la familia Pulsar® de unidades de estados sólido de ultra alto rendimiento

3.6 Arquitectura orientadas a servicios (SOA)

Es comunicación con la necesidad del negocio y el sistema de software. Su finalidad es el de aportar flexibilidad, desde la automatización de las infraestructura y herramientas necesarias consiguiendo, al mismo tiempo, reducir los costes de integración. SOA se ocupa del diseño y desarrollo de sistemas distribuidos y es un potente aliado a la hora de llevar a cabo la gestión de grandes volúmenes de datos, datos en la nube y jerarquías de datos.

SOA es un estilo arquitectónico para la construcción de aplicaciones de software en base a servicios disponibles. Entre sus principales características destacan:

- Su flexibilidad, que permite la reutilización.
- Su versatilidad, que hace posible que los servicios puedan ser consumidos por los clientes en aplicaciones o procesos de negocio distintos.
- Sus posibilidades, que optimizan el trabajo con datos y su coordinación.

SOA permite la reutilización de activos existentes para nuevos servicios que se pueden crear a partir de una infraestructura de TI que ya se había diseñado. De esta forma, permite



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

a las empresas optimizar la inversión por medio de la reutilización que, además, conlleva otra ventaja: la interoperabilidad entre las aplicaciones y tecnologías heterogéneas.

La **arquitectura orientada a servicios** es fuente de ventaja competitiva ya que, por su configuración:

- Aumenta la eficiencia en los procesos.
- Amortiza la inversión realizada en sistemas.
- Reduce costes de mantenimiento.
- Fomenta la innovación orientada al desarrollo de servicios.
- Simplifica el diseño, optimizando la capacidad de organización.

UNIDAD 4

TEMA 1: Servicios Web

Un servicio web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet.

De una manera más clara se podría decir que un web service es una función que diferentes servicios o equipos utilizan; es decir, solo se envían parámetros al servidor (lugar donde está alojado el web service) y éste responderá la petición. Entre algunas que se manejan de utilizar servicios webs en las aplicaciones destacan las siguientes:

- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.

4.1 Servicio web SOAP

SOAP (Simple Object Access Protocol) es un protocolo que define cómo deben realizarse las comunicaciones entre máquinas. SOAP usa XML como lenguaje de intercambio de datos con una estructura compleja que es capaz de albergar todo tipo de datos sobre la solicitud o respuesta generada.

4.1.1 Componentes de servicio web SOAP

Los componentes de un servicio web SOAP definen el objetivo del servicio web y la vía de comunicación del cliente de servicio web con el servicio web.

Un servicio web posee los siguientes componentes:

- Operaciones
Por otra parte, un servicio web puede contener una o varias operaciones. Cada operación corresponde a una acción en el servicio web.
- Web Services Description Language (WSDL)
Un WSDL es un documento XML que describe los protocolos, formatos y firmas de las operaciones de un servicio web.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

- Simple Object Access Protocol (SOAP)

SOAP es el protocolo de comunicación para los servicios web.

4.1.2 Web Services Description Language (WSDL)

A medida que los protocolos de comunicación y los formatos de mensajes están estandarizados en la comunidad web, se hace cada vez más posible e importante poder describir las comunicaciones de alguna manera estructurada. WSDL aborda esta necesidad definiendo una gramática XML para describir los servicios de red como colecciones de puntos finales de comunicación capaces de intercambiar mensajes. Las definiciones de servicio WSDL proporcionan documentación para sistemas distribuidos y sirven como una receta para automatizar los detalles involucrados en la comunicación de aplicaciones.

Un documento WSDL define los servicios como colecciones de puntos finales de red o puertos. En WSDL, la definición abstracta de puntos finales y mensajes se separa de su implementación de red concreta o enlaces de formato de datos. Esto permite la reutilización de definiciones abstractas: mensajes, que son descripciones abstractas de los datos que se intercambian, y tipos de puertos que son colecciones abstractas de operaciones. El protocolo concreto y las especificaciones de formato de datos para un tipo de puerto en particular constituyen un enlace reutilizable. Un puerto se define asociando una dirección de red con un enlace reutilizable, y una colección de puertos define un servicio. Por lo tanto, un documento WSDL utiliza los siguientes elementos en la definición de servicios de red:

- Tipos: un contenedor para definiciones de tipos de datos que utilizan algún sistema de tipos (como XSD).
- Mensaje: una definición abstracta y escrita de los datos que se comunican.
- Operación: una descripción abstracta de una acción respaldada por el servicio.
- Tipo de puerto: un conjunto abstracto de operaciones admitidas por uno o más puntos finales.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

- Enlace: un protocolo concreto y una especificación de formato de datos para un tipo de puerto en particular.
- Puerto: un único punto final definido como una combinación de un enlace y una dirección de red.
- Servicio: una colección de puntos finales relacionados.

Estos elementos se describen en detalle en la Sección 2. Es importante observar que WSDL no introduce un nuevo lenguaje de definición de tipo. WSDL reconoce la necesidad de sistemas de tipo rico para describir formatos de mensajes y admite la especificación de esquemas XML (XSD) [11] como su sistema de tipo canónico. Sin embargo, dado que no es razonable esperar que se use una gramática de sistema de un solo tipo para describir todos los formatos de mensajes presentes y futuros, WSDL permite usar otros lenguajes de definición de tipos mediante extensibilidad.

Además, WSDL define un mecanismo de enlace común. Esto se utiliza para adjuntar un protocolo específico o formato o estructura de datos a un mensaje abstracto, operación o punto final. Permite la reutilización de definiciones abstractas.

4.1.2 Ejemplo de documento WSDL

El siguiente ejemplo muestra la definición WSDL de un servicio simple que proporciona cotizaciones de acciones. El servicio admite una sola operación llamada GetLastTradePrice, que se implementa utilizando el protocolo SOAP 1.1 a través de HTTP. La solicitud toma un símbolo de ticker de tipo cadena y devuelve el precio como flotante.

```
<?xml version="1.0"?>
<definitions name="StockQuote"

targetNamespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
  xmlns:xsd1="http://example.com/stockquote.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
      xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="TradePriceRequest">
        <complexType>
          <all>
            <element name="tickerSymbol" type="string"/>
          </all>
        </complexType>
      </element>
    </schema>
  </types>
</definitions>
```



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN

GUIA DE APRENDIZAJE

```
</complexType>
</element>
<element name="TradePrice">
  <complexType>
    <all>
      <element name="price" type="float"/>
    </all>
  </complexType>
</element>
</schema>
</types>

<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>

<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:TradePrice"/>
</message>

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>

</definitions>
```

4.2 Servicios REST

REST (Representational State Transfer) es una interfaz para conectar varios sistemas basados en el protocolo HTTP (uno de los protocolos más antiguos) y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

El formato más usado en la actualidad es el formato JSON, ya que es más ligero y legible en comparación al formato XML. Elegir uno será cuestión de la lógica y necesidades de cada proyecto.

REST se apoya en HTTP, los verbos que utiliza son exactamente los mismos, con ellos se puede hacer GET, POST, PUT y DELETE. De aquí surge una alternativa a SOAP.

4.2.1 Arquitectura web REST

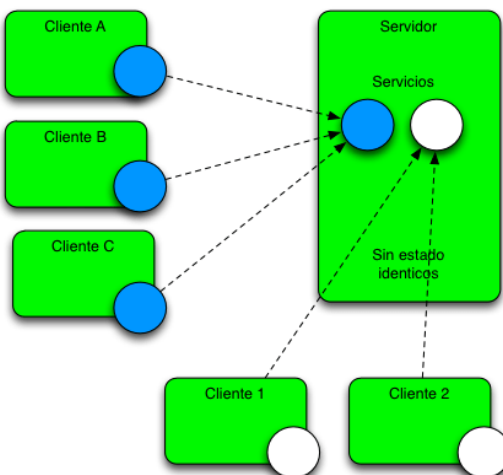
Cliente/Servidor : Como servicios web son cliente servidor y definen un interface de comunicación entre ambos separando completamente las responsabilidades entre ambas partes.



Imagen 6: Representación Cliente- Servidor

Fuente: <https://www.arquitecturajava.com/>

Sin estado : Son servicios web que no mantienen estado asociado al cliente .Cada petición que se realiza a ellos es completamente independiente de la siguiente . Todas las llamadas al mismo servicio serán idénticas.





INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN
GUIA DE APRENDIZAJE

Imagen 7: Representación Sin Estado

Fuente: <https://www.arquitecturajava.com/>

Cache : El contenido de los servicios web REST ha se puede cachear de tal forma que una vez realizada la primera petición al servicio el resto puedan apoyarse en la cache si fuera necesario.

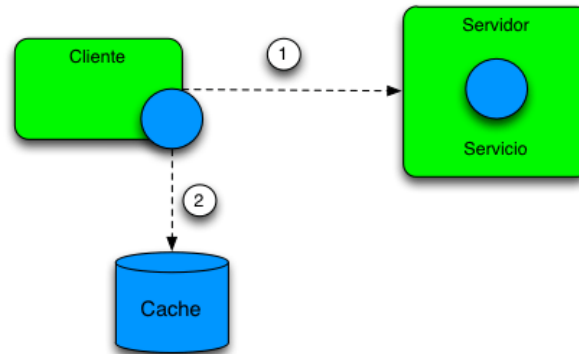


Imagen 8: Representación Cache

Fuente: <https://www.arquitecturajava.com/>

Servicios Uniformes : Todos los servicios REST compartirán una forma de invocación y métodos uniformes utilizando los métodos GET, POST, PUT, DELETE

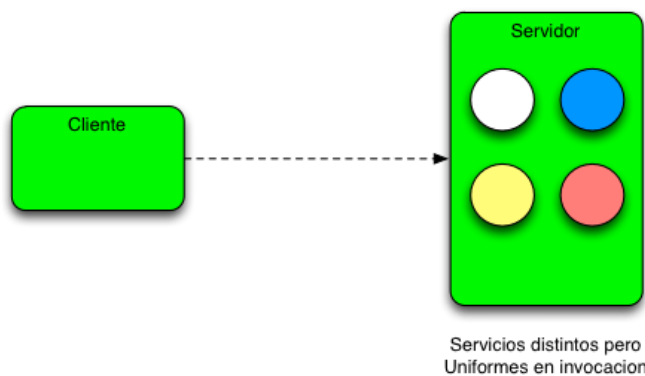


Imagen 9: Representación Servicios Uniformes

Fuente: <https://www.arquitecturajava.com/>

Arquitectura en Capas: Todos los servicios REST están orientados hacia la escalabilidad y un cliente REST no será capaz de distinguir entre si está realizando una petición directamente al servidor, o se lo está devolviendo un



sistema de caches intermedio o por ejemplo existe un balanceador que se encarga de redirigirlo a otro servidor.

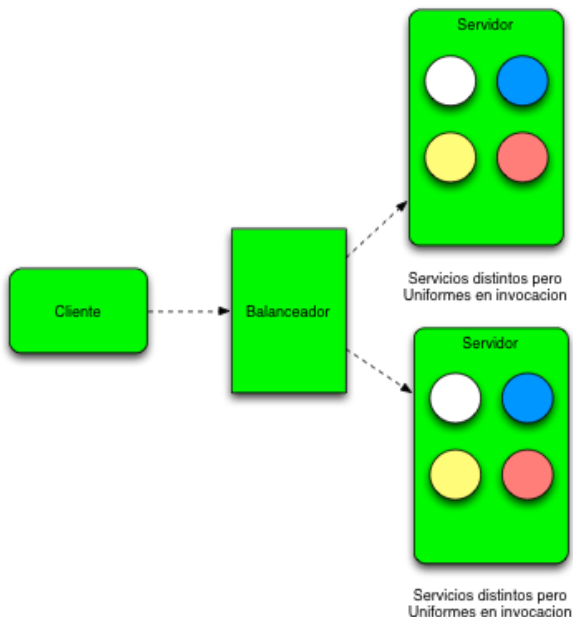


Imagen 10: Representación Arquitectura en Capas

Fuente: <https://www.arquitecturajava.com/>

Una vez vista una introducción al concepto de servicio REST en los siguientes POST nos encargaremos de construir uno usando los standards de la plataforma JEE.

4.2.2 JAX-RS Client y Servicios REST

El uso de servicios REST es cada día más común y han sido construidos para poder gestionar de una forma sencilla el intercambio de información entre sistemas heterogeneos. Estamos muy habituados a utilizar Ajax para comunicarnos con un servicio REST.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

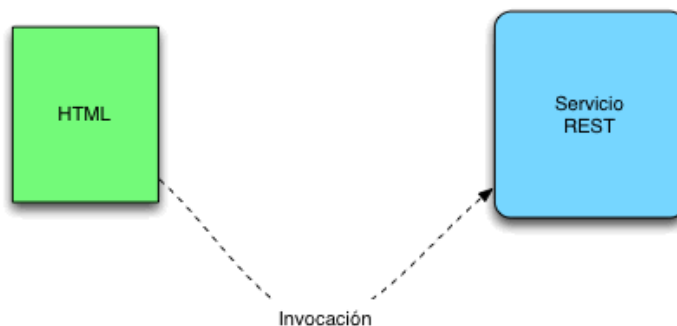


Imagen 11: Invocación Rest

Fuente: <https://www.arquitecturajava.com/>

Es también bastante habitual invocar servicios REST desde las APIs nativas de nuestras aplicaciones móviles como pueden ser aplicaciones desarrolladas con Android o IOS.

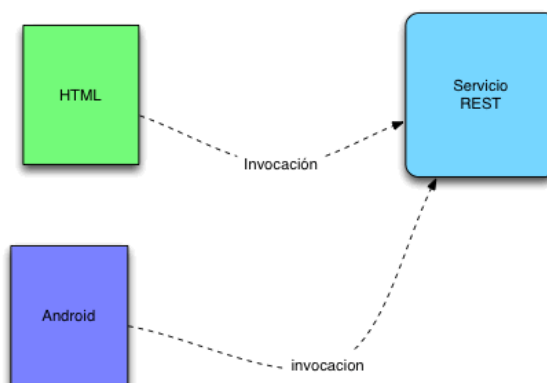


Imagen 12: Invocación Rest

Fuente: <https://www.arquitecturajava.com/>

Sin embargo pueden existir otras muchas situaciones en las que podemos querer invocar a un servicio REST, como por ejemplo desde el mundo Java Clásico.

4.2.3 JAX-RS Client

A partir de la versión 2.0 del API de REST (JAX-RS) de Java EE se soporta el uso de un API a nivel de Cliente de tal forma que podamos trabajar de una forma cómoda con ella (JAX-RS Client). Vamos a ver un ejemplo sencillo. Para ello lo primero que tendremos que hacer es construirnos un proyecto Maven con Eclipse y añadir las siguientes dependencias.



```
1. <dependency>
2. <groupId>javax.ws.rs</groupId>
3. <artifactId>javax.ws.rs-api</artifactId>
4. <version>2.0</version>
5. </dependency>
6.
7. <dependency>
8. <groupId>org.glassfish.jersey.core</groupId>
9. <artifactId>jersey-client</artifactId>
10. <version>2.12</version>
11. </dependency>
12.
13. <dependency>
14. <groupId>org.glassfish.jersey.media</groupId>
15. <artifactId>jersey-media-json-jackson</artifactId>
16. <version>2.11</version>
17. </dependency>
18. </dependencies>
```

Estamos añadiendo tres dependencias la primera el API de cliente de REST , la segunda las librerías que lo implementan y por ultimo las capacidades para trabajar con JSON a nivel de esta API . Una vez tenemos las dependencias definidas podemos construirnos un sencillo programa Java que accede a un servicio REST.

```
1. package com.arquitecturajava;
2. import java.io.Serializable;
3.
4. public class Factura implements Serializable {
5.
6.     private String id;
7.     private String concepto;
8.     private int importe;
9.     public String getId() {
10.         return id;
11.     }
12.     public void setId(String id) {
13.         this.id = id;
14.     }
15.     public String getConcepto() {
16.         return concepto;
17.     }
18.     public void setConcepto(String concepto) {
19.         this.concepto = concepto;
20.     }
21.     public int getImporte() {
22.         return importe;
23.     }
```

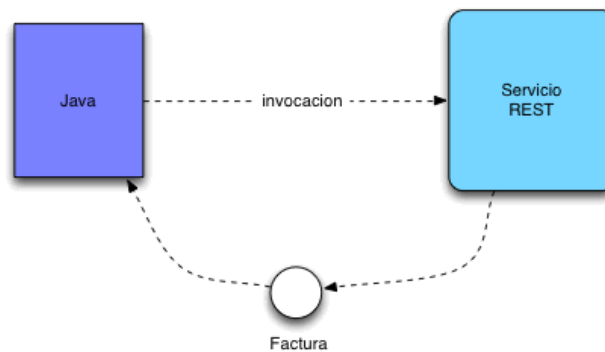


INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

```
24. public void setImporte(int importe) {  
25.     this.importe = importe;  
26. }  
27. }
```

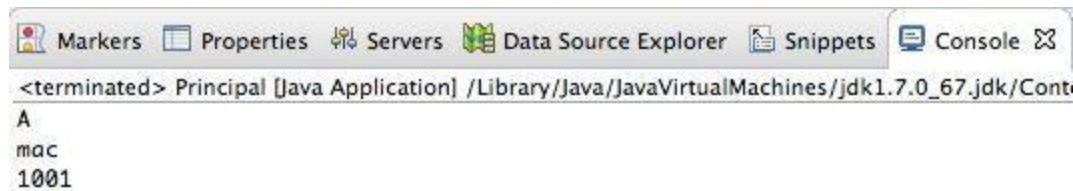
```
1. package com.arquitecturajava;  
2.  
3. import javax.ws.rs.client.Client;  
4. import javax.ws.rs.client.ClientBuilder;  
5. import javax.ws.rs.core.MediaType;  
6.  
7. public class Principal {  
8.  
9.     public static void main(String[] args) {  
10.  
11.         Client cliente = ClientBuilder.newClient();  
12.         Factura f = cliente.target("http://localhost:3000/mifactura")  
13.             .request(MediaType.APPLICATION_JSON_TYPE).get(Factura.class);  
14.  
15.         System.out.println(f.getId());  
16.         System.out.println(f.getConcepto());  
17.         System.out.println(f.getImporte());  
18.  
19.     }
```

De esta forma estaremos accediendo a la URL /mifactura de nuestro servidor que nos devolverá una sencilla factura en formato JSON





Realizada esta operación simplemente sacamos por pantalla los datos:



```
<terminated> Principal [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_67.jdk/Cont  
A  
mac  
1001
```

Imagen 13: Resultados en pantalla Invocación Rest- Factura

UNIDAD 5

TEMA 1: FRAMEWORKS

5.1 JERSEY

Jersey (JAX-RS)

El marco de servicios web RESTful de Jersey es un marco de código abierto, calidad de producción, marco para desarrollar servicios web RESTful en Java que proporciona soporte para las API JAX-RS y sirve como implementación de referencia JAX-RS (JSR 311 y JSR 339). El marco de Jersey es más que la implementación de referencia JAX-RS. Jersey proporciona su propia API que amplía el kit de herramientas JAX-RS con funciones y utilidades adicionales para simplificar aún más el servicio RESTful y el desarrollo del cliente.

5.1.1 Ejemplo

En este ejemplo de carga de archivos de Jersey 2, aprenderemos a cargar archivos binarios (por ejemplo, archivos PDF en este ejemplo) utilizando el soporte de datos de formularios de varias partes de Jersey. Aprenderemos a continuación los cambios necesarios para completar la funcionalidad.

1. Dependencia multiparte de Jersey

Para usar las funciones multiparte, debe agregar el módulo jersey-media-multipart a su archivo pom.xml:



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN

GUIA DE APRENDIZAJE

pom.xml

```
<dependency>
  <groupId>org.glassfish.jersey.media</groupId>
  <artifactId>jersey-media-multipart</artifactId>
  <version>2.19</version>
</dependency>
```

2. Agregue Jersey MultiPartFeature en web.xml

Además, debe agregar MultiPartFeature en la configuración de Jersey para informarle que utilizará solicitudes de varias partes. La manera más simple es agregar soporte a través del archivo web.xml.

web.xml

```
<servlet>
  <servlet-name>jersey-serlvet</servlet-name>
  <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>com.howtodoinjava.jersey</param-value>
  </init-param>
  <init-param>
    <param-name>jersey.config.server.provider.classnames</param-name>
    <param-value>org.glassfish.jersey.media.multipart.MultiPartFeature</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

3. Escribir API REST de carga de archivos Jersey

Ahorala API REST real para la carga de archivos, que recibirá y guardará el archivo.

```
@POST
@Path("/pdf")
@Consumes({MediaType.MULTIPART_FORM_DATA})
public Response uploadPdfFile( @FormDataParam("file") InputStream
fileInputStream,
                               @FormDataParam("file")
FormDataContentDisposition fileMetaData) throws Exception
{
  String UPLOAD_PATH = "c:/temp/";
  try
  {
    int read = 0;
    byte[] bytes = new byte[1024];

    OutputStream out = new FileOutputStream(new File(UPLOAD_PATH +
fileMetaData.getFileName()));
    while ((read = fileInputStream.read(bytes)) != -1)
    {
      out.write(bytes, 0, read);
    }
  }
}
```



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

```
        out.flush();
        out.close();
    } catch (IOException e)
    {
        throw new WebApplicationException("Error while uploading file.
Please try again !!");
    }
    return Response.ok("Data uploaded successfully !!").build();
}
```

4. Probar la carga de archivos usando el formulario HTML

Se debe crear un archivo 'fileUpload.html' en la carpeta 'webapp' y pegue este código.

```
<html>
  <body>
    <h1>File Upload Example - howtodoinjava.com</h1>

    <form action="rest/upload/pdf" method="post"
    enctype="multipart/form-data">

      <p>Select a file : <input type="file" name="file"
    size="45" accept=".pdf" /></p>
      <input type="submit" value="Upload PDF" />

    </form>

  </body>
</html>
```

Ahora se debe acceder a la URL: "http://localhost:8080/ JerseyDemos / fileUpload.html" y mostrará un control de archivo HTML para explorar el archivo. Seleccione cualquier archivo PDF y haga clic en el botón "Cargar PDF".

Su archivo se cargará y recibirá el mensaje: "¡Datos cargados con éxito!"

5. Probar la carga de archivos usando Jersey Client

Si está buscando cargar archivos usando clientes java, puede modificar el siguiente código de trabajo según sus necesidades.

Carga de archivos de Jersey utilizando el ejemplo FormDataMultiPart.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN GUIA DE APRENDIZAJE

```
public static void main(String[] args) throws IOException
{
    final Client client =
    ClientBuilder.newBuilder().register(MultiPartFeature.class).build();

    final FileDataBodyPart filePart = new FileDataBodyPart("file", new
    File("C:/temp/sample.pdf"));
    FormDataMultiPart formDataMultiPart = new FormDataMultiPart();
    final FormDataMultiPart multipart = (FormDataMultiPart)
    formDataMultiPart.field("foo", "bar").bodyPart(filePart);

    final WebTarget target =
    client.target("http://localhost:8080/JerseyDemos/rest/upload/pdf");
    final Response response =
    target.request().post(Entity.entity(multipart,
    multipart.getMediaType()));

    //Use response object to verify upload success

    formDataMultiPart.close();
    multipart.close();
}
```

5.1.2 Apache CXF vs Jersey

Apache CXF : Un marco de servicios de código abierto. Le ayuda a construir y desarrollar servicios utilizando API de programación frontend, como JAX-WS y JAX-RS. Estos servicios pueden hablar una variedad de protocolos como SOAP, XML / HTTP, RESTful HTTP o CORBA y funcionan en una variedad de transportes como HTTP, JMS o JBI.

Jersey : Un marco REST que proporciona una implementación JAX-RS *. Es un marco de código abierto, calidad de producción, para desarrollar servicios web RESTful en Java que proporciona soporte para las API JAX-RS y sirve como implementación de referencia JAX-RS (JSR 311 y JSR 339). Proporciona su propia API que amplía el kit de herramientas JAX-RS con funciones y utilidades adicionales para simplificar aún más el servicio RESTful y el desarrollo del cliente.

Apache CXF pertenece a la categoría "Marco de servicio web componible" de la pila tecnológica, mientras que Jersey se puede clasificar principalmente en "Microframeworks (Backend)".

B. Base de Consulta

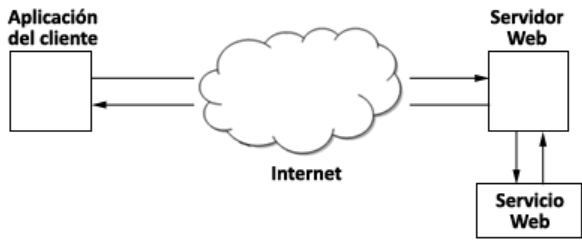



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN
 GUIA DE APRENDIZAJE

TÍTULO	AUTOR	EDICIÓN	AÑO	IDIOMA	EDITORIAL
Data Virtualization Going Beyond Traditional Data Integration to Achieve Business Agility.	Judith R. Davis and Robert Eve.	1	2011	Español	Composite Software.
Principles of Data Integration.	AnHai Doan, Alon Halevy, and Zachary Ives. Morgan Kaufmann.	1	2012	Inglés	
SOA-Based Enterprise Integration,.	Waseem Roshen,.	1	2009	Inglés	McGraw Hill Professional,
Web 2.0 Architectures	James Governor, Duane Nickull, Dion Hinchcliffe	1	2009	Inglés	O'Reilly Media
BIG Data Analytics and Cloud Computing	Trovati, M., Hill, R., Anjum, A., Zhu, S.Y., Liu, L. (Eds.).	1	2015	Inglés	Springer
Sistemas distribuidos: principios y paradigmas	Tanenbaum, Andrew S.	1	2008	Español	México Pearson Educación
Computación en la nube: estrategias de cloud computing en las empresas	Joyanes Aguilar, Luis	1	2012	Español	México Alfaomega
Java web services: up and running	Kalin, Martin	2	2013	Inglés	California O'Reilly
SOA Approach to Integration: XML, Web Services	Matjaz, B. Juric	1	2007	Inglés	Packt Publishing
https://www.powerdata.es/integracion-de-datos	Cap 1				
http://www.scielo.org.mx/pdf/poli/n51/n51a4.pdf					
Semantic database modeling: survey, applications, and research issues	R. Hull and R. King	19	1987	Inglés	ACM Computing Surveys
https://elvex.ugr.es/decsai/information-systems/slides/42%20Data%20Integration%20-%20Schema%20Matching%20&%20Mapping.pdf					



C. Base práctica con ilustraciones

TRABAJOS A PRESENTAR	DETALLE
	2.- Presentación de la creación de un proyecto , comprendiendo sistemas distribuidos para comunicar información, apoyándose de un lenguaje de programación para crear solución HTTP
	3- Presentación del Proyecto Impreso Debe contener carátula, encabezado y pie de página , Índice, inicio a normas APA

4. ESTRATEGIAS DE APRENDIZAJE

ESTRATEGIA DE APRENDIZAJE 1: Análisis y Planeación

Descripción:

- Discusión sobre las lecturas, artículos y videos.
- Desarrollar, habilidades y destrezas, con los conocimientos desarrollados en la comunidad para identificar los factores de riesgo y su oportuna intervención.
- Elaborar un proyecto, entendiendo conceptos de marco teórico, estado del arte, hipótesis, etc.
- Implementar un proyecto de desarrollo social como respuesta a un problema previamente identificado en el ámbito comunitario, el cual sustenta con claridad y precisión.
- Definir, analizar, implementar y gestionar las herramientas de programación, mostrando un conocimiento sólido respecto al uso de las Tecnologías de la Información y Comunicación y aplicando las técnicas elementales para buscar información en internet, procesarla y almacenarla.



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN
GUIA DE APRENDIZAJE

- Realizar prácticas en clase, para comprender el desarrollo de un proyecto metodológico , apoyado en las tecnologías de información y que sean un aporte social comunitario
- Crear debates de participación.
- Prácticas en laboratorio de cómputo para el desarrollo del proyecto de programación, apoyado de una herramienta de TI.

Ambiente(s) requerido:

Aula amplia con buena iluminación, y laboratorios.

Material (es) requerido:

- ✓ Aula de clase
- ✓ Aulas virtuales
- ✓ Bibliotecas, páginas web
- ✓ Videos a fines al tema impartido
- ✓ Proyector

Computador

Docente:

Con conocimiento de la materia.

5. ACTIVIDADES

- Controles de lectura
- Exposiciones
- Presentación del Trabajo final
- CD con contenido del Proyecto
- Habilidad y esfuerzo en el proyecto entregado

6. EVIDENCIAS Y EVALUACIÓN

Tipo de Evidencia	Descripción (de la evidencia)
De conocimiento:	Creación de un Proyecto aplicando una metodología de desarrollo de software, que permita estructurar el proyecto y que



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN
GUIA DE APRENDIZAJE

	siga los lineamientos de las normas APA , que permitirá al estudiante tener los elementos y lineamientos para trabajos colaborativos.
Desempeño:	Trabajo individual presentación del trabajo sobre la creación de una proyecto, usando herramientas de tecnologías de información para la programación de un sistema que pueda dar un aporte y se convierta en una solución empresarial Exposición individual del proyecto educacional
De Producto:	✓ Desarrollo de un proyecto innovador, que debe ser promocionado a un entorno social, utilizando medios para difundir la información, generando interés significativo y positivo. Intervención mediante una práctica de los estudiante. ✓ Exposiciones oral sobre los temas de investigación individuales asignados a los estudiantes.
De Innovación	Se revisará la participación investigativa por parte del alumno en cuanto refiere a la innovación y desempeño al proyecto entregado.
Criterios de Evaluación (Mínimo 5 Actividades por asignatura)	Identificar la metodología seleccionada, medios de difundir la información, herramienta tecnológica utilizada, Normas Apa establecidas en el proyecto, creatividad e investigación social.

ANEXO 1 EVIDENCIAS DE APRENDIZAJE

Msc. Diana Moncayo	Alexis Benavides	Milton Altamirano



INSTITUTO TECNOLÓGICO SUPERIOR JAPÓN
GUIA DE APRENDIZAJE

Elaborado por: (Docente)	Revisado Por: (Coordinador)	Reportado Por: (Vicerrector)
------------------------------------	---------------------------------------	--

Bibliografía

ANEXO 1



Guía metodológica de integración de sistemas informáticos

Desarrollo de software

Msc. Diana Moncayo

2019

Coordinación editorial general:

Mgs. Milton Altamirano Pazmiño

Ing. Alexis Benavides Vinueza

Mgs. Lucía Begnini Dominguez

Diagramación: Sebastián Gallardo Ramírez

Corrección de estilo: Mgs. Lucía Begnini Dominguez

Diseño: Sebastián Gallardo Ramírez

Imprenta: JKIMPRIMA

Instituto Superior Tecnológico Japón

AMOR AL CONOCIMIENTO

ISBN: 978-9942-811-99-8

