INSTITUTO TECNOLOGICO DE TUXTEPEC

INGENIERIA EN SISTEMAS COMPUTACIONALES

Asignatura:

"Programación Web"

Docente:

Investigacion:

UNIDAD V

Presentan:

- Pérez Pérez Cristian Eduardo
- Ramon Arano Zita Yamilet
- Vasquez Garcia Grisell
- Rosas Frutos Fernando
- Flores Nuñez Ana Faride
- Torres Gudiño María de Jesús

05/06/2019

INTRODUCCION

El objetivo de este tema es introducir en el entorno en el que se mueve la programación web y clarificar los conceptos fundamentales sobre los que nos vamos a ir moviendo en este curso. No se pretende que se dominen todos los términos pero sí obtener una idea general de donde van encajando las distintas tecnologías.

En toda conexión web existen dos partes bien separadas: cliente y servidor. El cliente suele ser la máquina del usuario que utiliza un navegador de páginas web, el servidor es quien recibe esa petición, es donde reside el código de las diferentes páginas y la base de datos y es donde, en principio, se realiza el procesamiento.

En entornos de desarrollo, a fin de poder hacer pruebas de manera más fácil, la máquina cliente y servidor es la misma, éste será el entorno que utilizaréis para realizar el trabajo final aunque se explicará como implantar una aplicación Django en servidor Apache y configurarla para servir en producción.

UNIDAD V

5.1 Conceptos Generales

La computación en la nube son servidores desde Internet encargados de atender las peticiones en cualquier momento. Se puede tener acceso a su información o servicio, mediante una conexión a internet desde cualquier dispositivo móvil o fijo ubicado en cualquier lugar. Sirven a sus usuarios desde varios proveedores de alojamiento repartidos frecuentemente por todo el mundo. Esta medida reduce los costos, garantiza un mejor tiempo de actividad y que los sitios web sean invulnerables a los delincuentes informáticos, a los gobiernos locales y a sus redadas policiales pertenecientes.

Cloud computing es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite incluso al usuario acceder a un catálogo de servicios estandarizados y responder con ellos a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado, o incluso gratuitamente en caso de proveedores que se financian mediante publicidad o de organizaciones sin ánimo de lucro.

El cambio que ofrece la computación desde la nube es que permite aumentar el número de servicios basados en la red. Esto genera beneficios tanto para los proveedores, que pueden ofrecer, de forma más rápida y eficiente, un mayor número de servicios, como para los usuarios que tienen la posibilidad de acceder a ellos, disfrutando de la ‘transparencia’ e inmediatez del sistema y de un modelo de pago por consumo. Así mismo, el consumidor ahorra los costes salariales o los costes en inversión económica (locales, material especializado, etc.).

Computación en nube consigue aportar estas ventajas, apoyándose sobre una infraestructura tecnológica dinámica que se caracteriza, entre otros factores, por un alto grado de automatización, una rápida movilización de los recursos, una elevada capacidad de adaptación para atender a una demanda variable, así como virtualización avanzada y un precio flexible en función del consumo realizado, evitando además el uso fraudulento del software y la piratería.

La computación en la nube ofrece servicios de tecnologías de la información. Ha tenido aceptación de manera eficaz de comprar y utilizar servicios de TI a precios accesibles e incluso gratuitos.

Los servicios en la nube son alternativas para operar algunos servicios de TI. Hay diferentes tipos de nubes disponibles que ofrecen diferentes tipos de servicios. Como lo son:

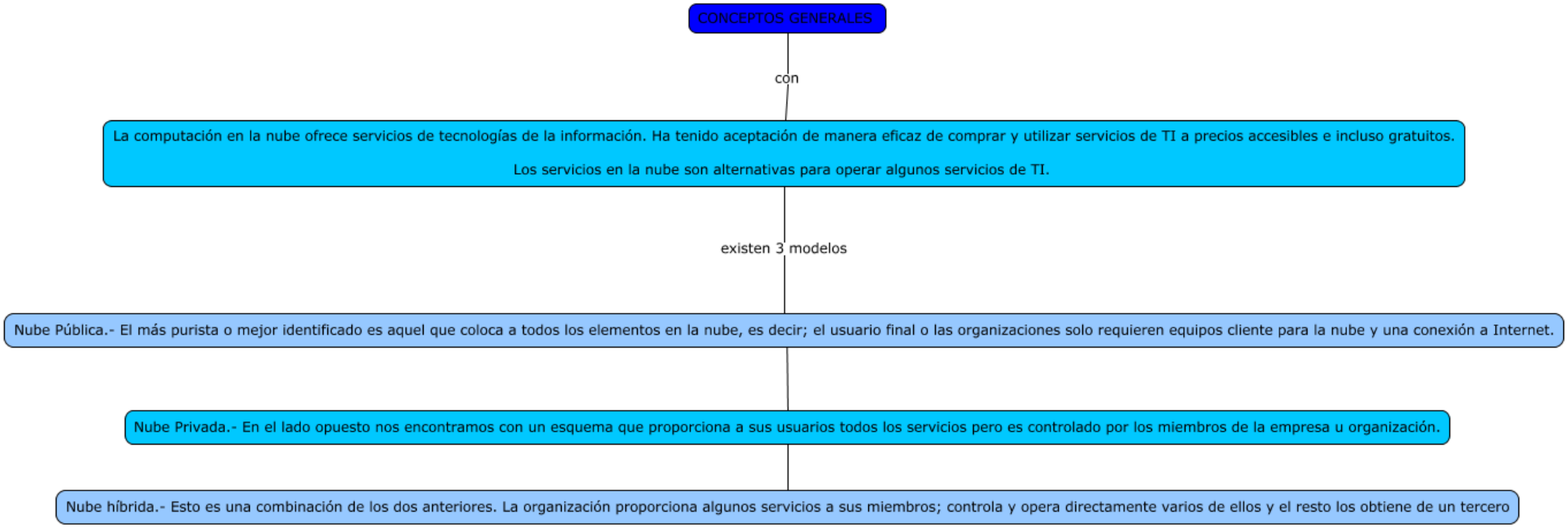
Las Nubes públicas ofrecen servicios por internet que permiten acceso de usuarios bajo demanda a través de utilidad a la computación, almacenamiento y aplicaciones de software. Las aplicaciones quedan hospedadas de forma segura en Data Centers remotos, en lugar de quedar en el local.

Existen básicamente 3 modelos de implementación de cómputo en la nube:

Nube Pública.- El más purista o mejor identificado es aquel que coloca a todos los elementos en la nube, es decir; el usuario final o las organizaciones solo requieren equipos cliente para la nube y una conexión a Internet.

Nube Privada.- En el lado opuesto nos encontramos con un esquema que proporciona a sus usuarios todos los servicios pero es controlado por los miembros de la empresa u organización.

Nube híbrida.- Esto es una combinación de los dos anteriores. La organización proporciona algunos servicios a sus miembros; controla y opera directamente varios de ellos y el resto los obtiene de un tercero.


CONCEPTOS GENERALES
con
La computación en la nube ofrece servicios de tecnologías de la información. Ha tenido aceptación de manera eficaz de comprar y utilizar servicios de TI a precios accesibles e incluso gratuitos.
Los servicios en la nube son alternativas para operar algunos servicios de TI.
existen 3 modelos
Nube Pública.- El más purista o mejor identificado es aquel que coloca a todos los elementos en la nube, es decir; el usuario final o las organizaciones solo requieren equipos cliente para la nube y una conexión a Internet.
Nube Privada.- En el lado opuesto nos encontramos con un esquema que proporciona a sus usuarios todos los servicios pero es controlado por los miembros de la empresa u organización.
Nube híbrida.- Esto es una combinación de los dos anteriores. La organización proporciona algunos servicios a sus miembros; controla y opera directamente varios de ellos y el resto los obtiene de un tercero

5.2 Tipos de servicios en la nube

Como hay diferentes tipos de nubes (Nubes públicas, nubes privadas, Nubes híbridas), existen distintos tipos de servicios dentro del cloud computing. Los mas importantes son:

- **Software as a Service (SaaS)**
- **Platform as a Service (PaaS)**
- **Infrastructure as a Service (IaaS)**

La diferencia principal entre los tres servicios es el grado del uso de componentes de software y hardware y su gestion por terceros o su gestion propia. Sus usos dependenden de la necesidad que un cliente/usuario tenga.

- **Software as a Service (SaaS) – Software como servicio**

Este servicio es el más conocido por el mercado. El software se aloja en servidores de los proveedores (compañía de TIC) y se accede con un navegador web o un cliente fino especializado, a través de internet. El software esta listo para ser usados por los clientes.
El usuario no tiene que preocuparse porque todo el mantenimiento, el soporte y la disponibilidad del software está manejado por el proveedor. El software puede ser consultado en cualquier ordenador a través de un servidor central colocado en la empresa proveedora de sistemas y no en la compañía del cliente.
Ejemplos: MS Office, Sharepoint etc.. Aplicaciones CRM de Salesforce etc. Correo, almacenamiento, juegos, aplicaciones ofimáticas y de colaboración, CRMs, redes sociales y un largo etcétera conforman este tipo de servicios.

- **Platform as a Service (PaaS) – Plataforma como servicio**

En este caso el usuario puede utilizar directamente a una carga de servicios que le permite alojar y desarrollar sus propias aplicaciones (desarrollos propios o licencias adquiridas) en una plataforma ie. una plataforma que engloba los recursos de infraestructura, sistemas operativos, middleware y runtimes. El proveedor ofrece el uso de su plataforma que a su vez se encuentra alojada en sus infraestructuras. Ejemplos: MS Windows Azure, Google App Engine.

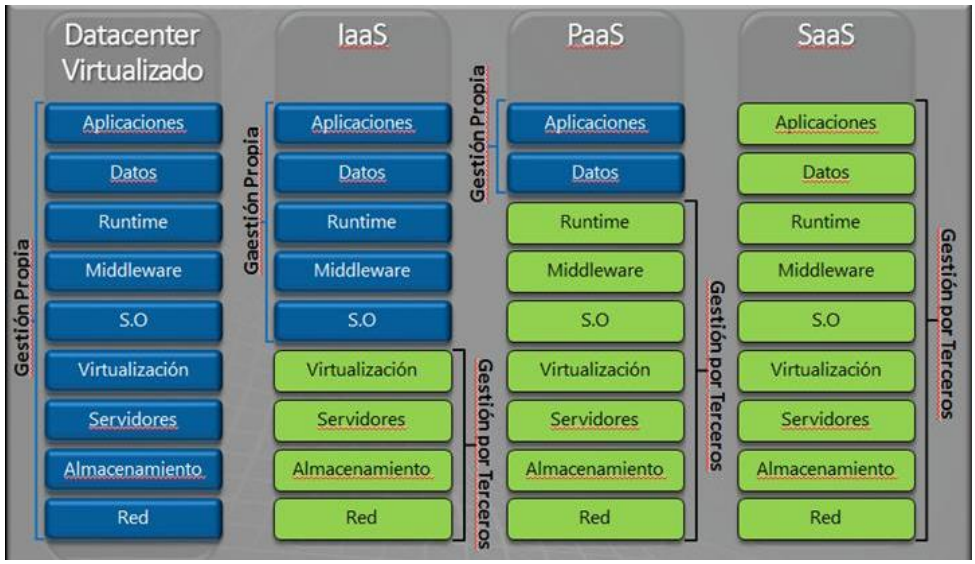
- **Infrastructure as a Service (IaaS) – Infraestructura como servicio**

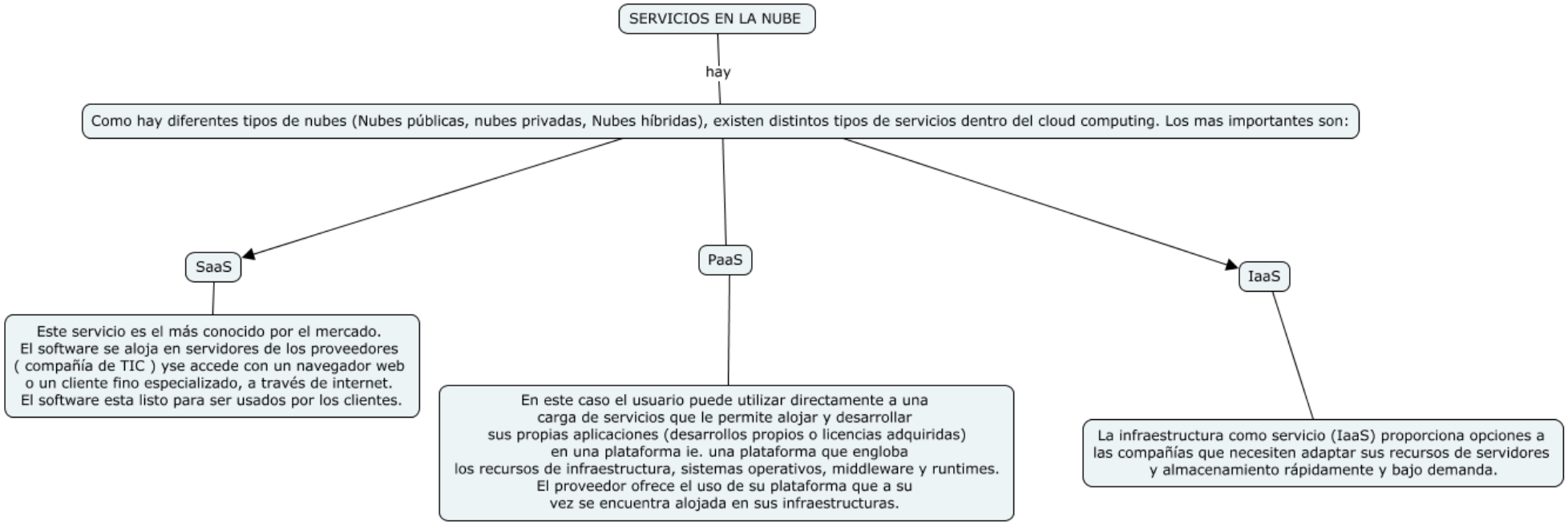
La infraestructura como servicio (IaaS) proporciona opciones a las compañías que necesiten adaptar sus recursos de servidores y almacenamiento rápidamente y bajo demanda.

Se desacopla las cargas de trabajo del hardware físico y su consolidación sobre servidores. En otras palabras, se construye una solución que comprenda los “Fabrics” (tejidos, estructuras) de red y almacenamiento y el conjunto de

elementos de cómputo incluidos en los servidores, con su capa de virtualización por encima, y ofrecerla a clientes internos y/o externos.

En ocasiones se refieren a IaaS como HaaS (Hardware as a Service). Como ejemplos de este tipo de servicios podemos mencionar GoGrid y Amazon EC2 (Elastic Compute Cloud).




SERVICIOS EN LA NUBE
hay
Como hay diferentes tipos de nubes (Nubes públicas, nubes privadas, Nubes híbridas), existen distintos tipos de servicios dentro del cloud computing. Los mas importantes son:
SaaS
PaaS
IaaS
Este servicio es el más conocido por el mercado.
El software se aloja en servidores de los proveedores
(compañía de TIC) yse accede con un navegador web
o un cliente fino especializado, a través de internet.
El software esta listo para ser usados por los clientes.
En este caso el usuario puede utilizar directamente a una
carga de servicios que le permite alojar y desarrollar
sus propias aplicaciones (desarrollos propios o licencias adquiridas)
en una plataforma ie. una plataforma que engloba
los recursos de infraestructura, sistemas operativos, middleware y runtimes.
El proveedor ofrece el uso de su plataforma que a su
vez se encuentra alojada en sus infraestructuras.
La infraestructura como servicio (IaaS) proporciona opciones a
las compañías que necesiten adaptar sus recursos de servidores
y almacenamiento rápidamente y bajo demanda.

5.3 Patrones de diseño

Hay una cosa que está clara: por muy específico que sea un problema al que te estés enfrentando durante el desarrollo de tu software, hay un 99% de posibilidades (cifra totalmente inventada, pero seguro que muy real) de que alguien se haya enfrentado a un problema tan similar en el pasado, que se pueda modelar de la misma manera.

Con modelado me estoy refiriendo a que la estructura de las clases que conforma la solución de tu problema puede estar ya inventada, porque estás resolviendo un problema común que otra gente ya ha solucionado antes. Si la forma de solucionar ese problema se puede extraer, explicar y reutilizar en múltiples ámbitos, entonces nos encontramos ante un patrón de diseño de software.
El concepto de patrón de diseño lleva existiendo desde finales de los 70, pero su verdadera popularización surgió en los 90 con el lanzamiento del libro de Design Pattern de la Banda de los Cuatro (Gang of Four), que aunque parezca que estamos hablando de los Trotamúsicos, es el nombre con el que se conoce a los creadores de este libro: Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. En él explican 23 patrones de diseño, que desde entonces han sido un referente.

¿Por qué son útiles los patrones de diseño?

Puede parecer una tontería, pero si no encuentras utilidad a las cosas acabarás por no usarlas. Los patrones de diseño son muy útiles por los siguientes motivos:

1. Te ahorran tiempo

Sé que te encantará encontrar una solución ingeniosa a un problema cuando estás modelando tu software, y es normal, a mí también me pasa. Como he comentado alguna vez, el desarrollo es un proceso casi artístico, y ese reto mental que supone revierte en una satisfacción personal enorme una vez que consigues un buen resultado.

Pero hay que ser sinceros: buscar siempre una nueva solución a los mismos problemas reduce tu eficacia como desarrollador, porque estás perdiendo mucho tiempo en el proceso. No hay que olvidar que el desarrollo de software también es una ingeniería, y que por tanto en muchas ocasiones habrá reglas comunes para solucionar problemas comunes.

2. Te ayudan a estar seguro de la validez de tu código

Un poco relacionado con lo anterior, siempre que creamos algo nuevo nos surge la duda de si realmente estamos dando con la solución correcta, o si realmente habrá una respuesta mejor. Y el tema es que es una duda muy razonable y que en muchos casos la respuesta sea la que no deseas:

Sí que hay una solución más válida, y has perdido tu valioso tiempo en implementar algo que, aunque funciona, podría haberse modelado mejor.

Los patrones de diseño son estructuras probadas por millones de desarrolladores a lo largo de muchos años, por lo que si eliges el patrón adecuado para modelar el problema adecuado, puedes estar seguro de que va a ser una de las soluciones más válidas (si no la que más) que puedas encontrar.

3. Establecen un lenguaje común

Todas las demás razones palidecen ante esta. Modelar tu código mediante patrones te ayudará a explicar a otras personas, conozcan tu código o no, a entender cómo has atajado un problema. Además ayudan a otros desarrolladores a comprender lo que has implementado, cómo y por qué, y además a descubrir rápidamente si esa era la mejor solución o no.

Listado de patrones de diseño

Este es el listado de los patrones de diseño más conocidos. Poco a poco iré escribiendo artículos sobre cada uno de ellos y los iré enlazando aquí. Es un proceso largo porque son muchos y quiero encontrar la mejor forma de explicarlos, pero llegará el día en que tengamos aquí un listado completo que te sirva de referencia.

Los patrones se dividen en distintos grupos según el tipo de problema que resuelven, a saber:

Patrones creacionales

Son los que facilitan la tarea de creación de nuevos objetos, de tal forma que el proceso de creación pueda ser desacoplado de la implementación del resto del sistema.

Los patrones creacionales están basados en dos conceptos:

Encapsular el conocimiento acerca de los tipos concretos que nuestro sistema utiliza. Estos patrones normalmente trabajarán con interfaces, por lo que la implementación concreta que utilicemos queda aislada.
Ocultar cómo estas implementaciones concretas necesitan ser creadas y cómo se combinan entre sí.

Los patrones creacionales más conocidos son:

Abstract Factory: Nos provee una interfaz que delega la creación de un conjunto de objetos relacionados sin necesidad de especificar en ningún momento cuáles son las implementaciones concretas.

Factory Method: Expone un método de creación, delegando en las subclases la implementación de este método.

Builder: Separa la creación de un objeto complejo de su estructura, de tal forma que el mismo proceso de construcción nos puede servir para crear representaciones diferentes.

Singleton: limita a uno el número de instancias posibles de una clase en nuestro programa, y proporciona un acceso global al mismo.

Prototype: Permite la creación de objetos basados en «plantillas». Un nuevo objeto se crea a partir de la clonación de otro objeto.

Patrones estructurales

Son patrones que nos facilitan la modelización de nuestros software especificando la forma en la que unas clases se relacionan con otras.

Estos son los patrones estructurales que definió la Gang of Four:

Adapter: Permite a dos clases con diferentes interfaces trabajar entre ellas, a través de un objeto intermedio con el que se comunican e interactúan.

Bridge: Desacopla una abstracción de su implementación, para que las dos puedan evolucionar de forma independiente.

Composite: Facilita la creación de estructuras de objetos en árbol, donde todos los elementos emplean una misma interfaz. Cada uno de ellos puede a su vez contener un listado de esos objetos, o ser el último de esa rama.

Decorator: Permite añadir funcionalidad extra a un objeto (de forma dinámica o estática) sin modificar el comportamiento del resto de objetos del mismo tipo.

Facade: Una facade (o fachada) es un objeto que crea una interfaz simplificada para tratar con otra parte del código más compleja, de tal forma que simplifica y aísla su uso. Un ejemplo podría ser crear una fachada para tratar con una clase de una librería externa.

Flyweight: Una gran cantidad de objetos comparte un mismo objeto con propiedades comunes con el fin de ahorrar memoria.

Proxy: Es una clase que funciona como interfaz hacia cualquier otra cosa: una conexión a Internet, un archivo en disco o cualquier otro recurso que sea costoso o imposible de duplicar.

Patrones de comportamiento

En este último grupo se encuentran la mayoría de los patrones, y se usan para gestionar algoritmos, relaciones y responsabilidades entre objetos.
Los patrones de comportamiento son:

Command: Son objetos que encapsulan una acción y los parámetros que necesitan para ejecutarse.

Chain of responsibility: se evita acoplar al emisor y receptor de una petición dando la posibilidad a varios receptores de consumirlo. Cada receptor tiene la opción de consumir esa petición o pasárselo al siguiente dentro de la cadena.

Interpreter: Define una representación para una gramática así como el mecanismo para evaluarla. El árbol de sintaxis del lenguaje se suele modelar mediante el patrón Composite.

Iterator: Se utiliza para poder movernos por los elementos de un conjunto de forma secuencial sin necesidad de exponer su implementación específica.

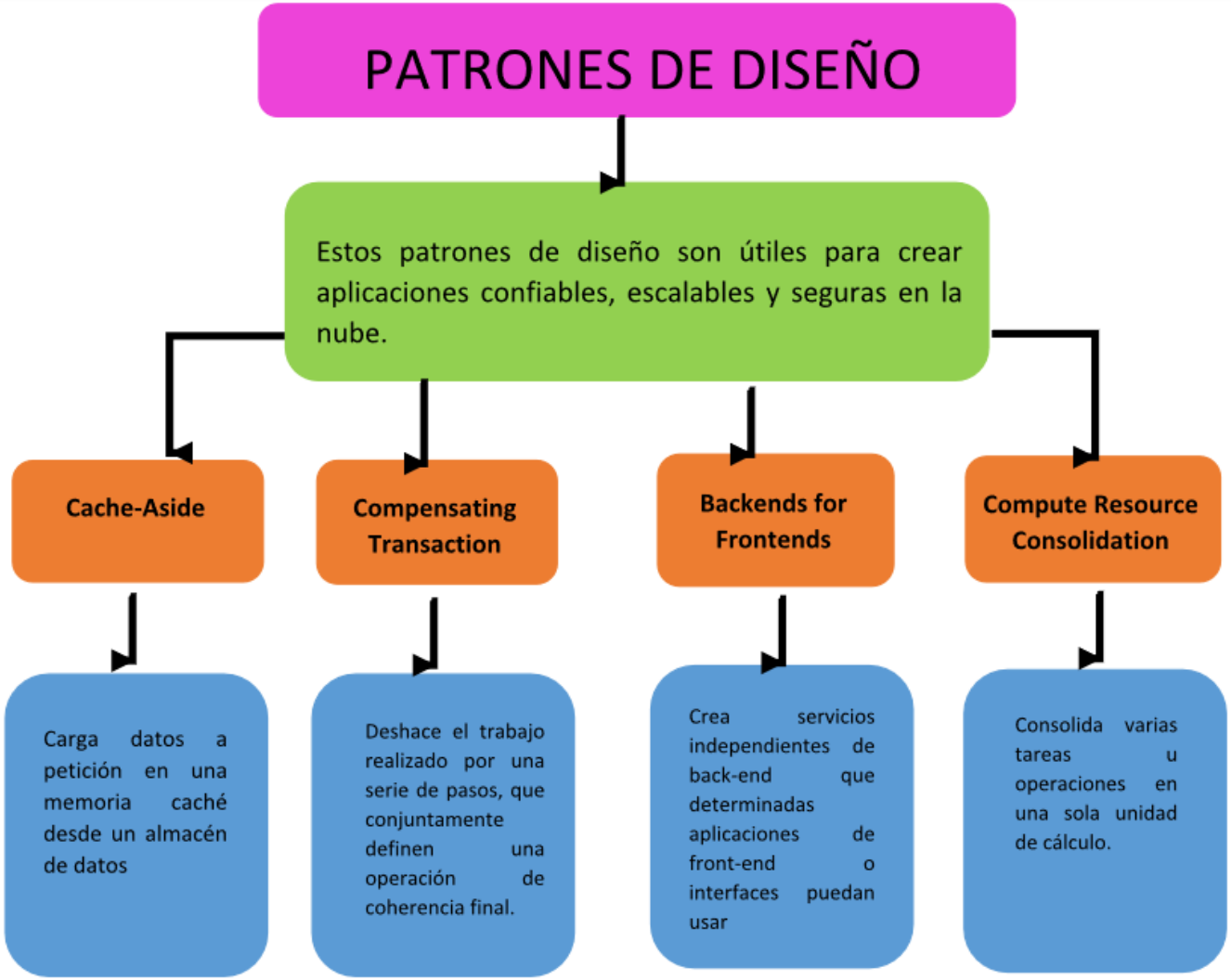
Mediator: Objeto que encapsula cómo otro conjunto de objetos interactúan y se comunican entre sí.

Memento: Este patrón otorga la capacidad de restaurar un objeto a un estado anterior
Observer: Los objetos son capaces de suscribirse a una serie de eventos que otro objetivo va a emitir, y serán avisados cuando esto ocurra.
State: Permite modificar la forma en que un objeto se comporta en tiempo de ejecución, basándose en su estado interno.

Strategy: Permite la selección del algoritmo que ejecuta cierta acción en tiempo de ejecución.

Template Method: Especifica el esqueleto de un algoritmo, permitiendo a las subclases definir cómo implementan el comportamiento real.

Visitor: Permite separar el algoritmo de la estructura de datos que se utilizará para ejecutarlo. De esta forma se pueden añadir nuevas operaciones a estas estructuras sin necesidad de modificarlas.


PATRONES DE DISEÑO
Estos patrones de diseño son útiles para crear aplicaciones confiables, escalables y seguras en la nube.
Cache-Aside
Compensating Transaction
Backends for Frontends
Compute Resource Consolidation
Carga datos a petición en una memoria caché desde un almacén de datos
Deshace el trabajo realizado por una serie de pasos, que conjuntamente definen una operación de coherencia final.
Crea servicios independientes de back-end que determinadas aplicaciones de front-end o interfaces puedan usar
Consolida varias tareas u operaciones en una sola unidad de cálculo.

5.4 Estándares de servicios

Uno de los atributos clave de estándares de Internet es que se centran en protocolos y no en implementaciones. Internet se compone de tecnologías heterogéneas que operan conjuntamente de modo satisfactorio mediante protocolos compartidos. Esto impide que los proveedores individuales impongan un estándar en Internet. El desarrollo del software de código fuente abierto desempeña un rol fundamental para proteger la interoperatividad de implementaciones de estándares del proveedor.

Los estándares siguientes desempeñan roles clave en servicios Web: UDDI (Universal Description, Discovery and Integration), WSDL (Web Services Description Language), WSIL (Web Services Inspection Language), SOAP y WS-I (Web Services Interoperability).

La especificación UDDI define estándares abiertos independientes de la plataforma que permiten a las empresas compartir información en un registro de empresa global, encontrar servicios en el registro y definir cómo actúan conjuntamente en Internet. Si desea más información sobre UDDI.

WSIL es una especificación abierta basada en XML que define un método de descubrimiento de servicios distribuidos que suministra referencias a descripciones de servicio en el punto de ofertas del proveedor de servicios, especificando cómo comprobar si hay servicios Web disponibles en un sitio Web. Un documento WSIL define las ubicaciones en un sitio Web donde se pueden buscar descripciones del servicio Web. Dado que WSIL se centra en el descubrimiento de servicios distribuidos, la especificación WSIL complementa UDDI facilitando el descubrimiento de servicios que están disponibles en sitios Web que quizá no se enumeren aún en un registro UDDI. En un tema aparte de esta documentación se describe la Relación entre UDDI y WSIL. Si desea más información sobre WSIL.

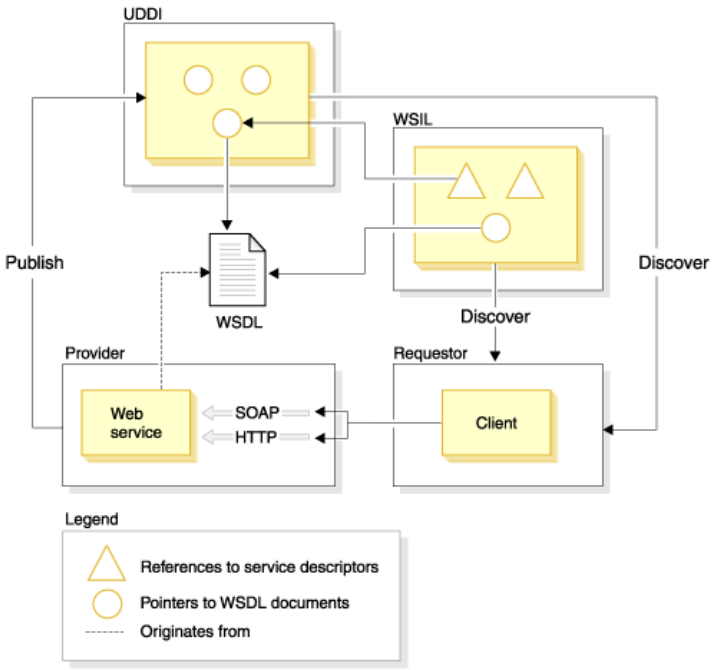
WSDL es una especificación abierta basada en XML que describe las interfaces y las instancias de servicios Web en la red. Es ampliable, de modo que se pueden describir los puntos finales independientemente de los formatos de mensaje o de los protocolos de red

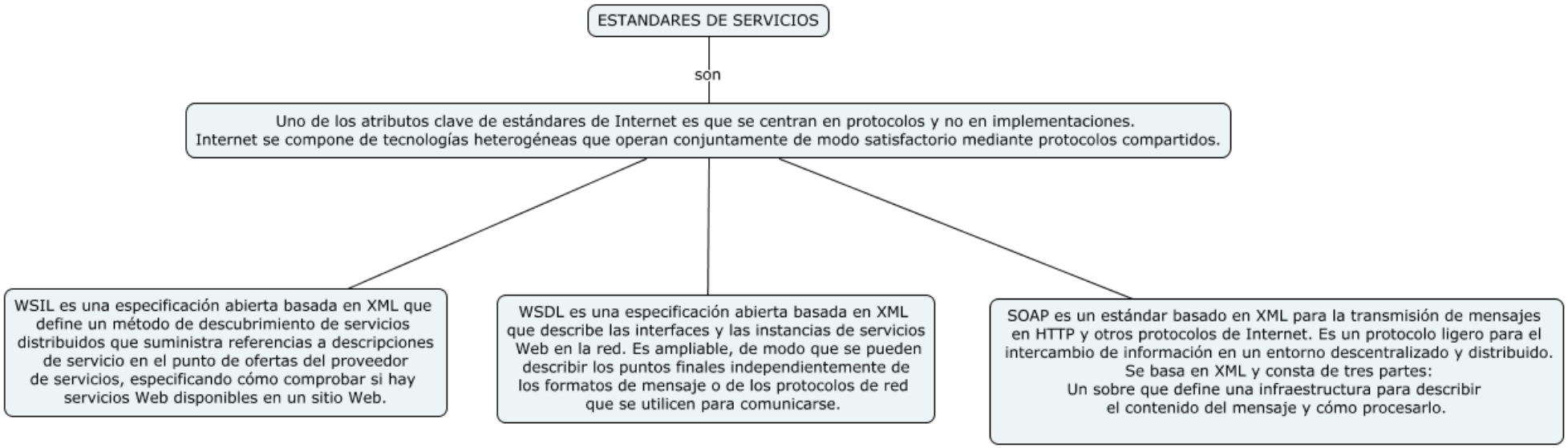
que se utilicen para comunicarse. Las empresas pueden poner a disposición de sus servicios Web los documentos WSDL mediante UDDI, WSIL o divulgando los URL a su WSDL mediante correo electrónico o sitios Web. WSDL se describe en un tema aparte de esta documentación. Si desea más información sobre WSDL.

SOAP es un estándar basado en XML para la transmisión de mensajes en HTTP y otros protocolos de Internet. Es un protocolo ligero para el intercambio de información en un entorno descentralizado y distribuido. Se basa en XML y consta de tres partes:
Un sobre que define una infraestructura para describir el contenido del mensaje y cómo procesarlo.

Un conjunto de normas de codificación para expresar instancias de tipos de datos definidos por la aplicación.
Una convención para representar llamadas y respuestas a procedimiento remoto.

SOAP permite el enlace y la utilización de servicios Web encontrados definiendo una ruta de mensaje para el direccionamiento de mensajes. Se puede utilizar SOAP para consultar UDDI para servicios Web. Si desea más información acerca de Apache SOAP.




ESTANDARES DE SERVICIOS
son
Uno de los atributos clave de estándares de Internet es que se centran en protocolos y no en implementaciones.
Internet se compone de tecnologías heterogéneas que operan conjuntamente de modo satisfactorio mediante protocolos compartidos.
WSIL es una especificación abierta basada en XML que define un método de descubrimiento de servicios distribuidos que suministra referencias a descripciones de servicio en el punto de ofertas del proveedor de servicios, especificando cómo comprobar si hay servicios Web disponibles en un sitio Web.
WSDL es una especificación abierta basada en XML que describe las interfaces y las instancias de servicios Web en la red. Es ampliable, de modo que se pueden describir los puntos finales independientemente de los formatos de mensaje o de los protocolos de red que se utilicen para comunicarse.
SOAP es un estándar basado en XML para la transmisión de mensajes en HTTP y otros protocolos de Internet. Es un protocolo ligero para el intercambio de información en un entorno descentralizado y distribuido.
Se basa en XML y consta de tres partes:
Un sobre que define una infraestructura para describir el contenido del mensaje y cómo procesarlo.

5.5 Plataformas de tecnologías

Arquitectura de la información web: es la disciplina encargada de la organización y estructuración de la información en los websites.

Servidor web: es un programa diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language). Estos son textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores musicales. El programa implementa el protocolo HTTP (HyperText Transfer Protocol) que pertenece a la capa de aplicación del modelo OSI. Ejecutado en un ordenador, el servidor web se mantiene a la espera de peticiones por parte de un navegador web y responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando un mensaje si se detectó algún error.

A manera de ejemplo, si escribimos www.google.com en nuestro navegador, éste hará una petición HTTP al servidor de dicha dirección. Acto seguido, el servidor responderá al cliente enviando el código HTML de la página. Luego el cliente recibe el código, lo interpreta y lo exhibe en pantalla. La interpretación del código HTML consiste en mostrar las fuentes, los colores y la disposición de los textos y objetos de la página. Por su parte, el servidor se limita a transferir el código de la página sin realizar interpretación alguna.

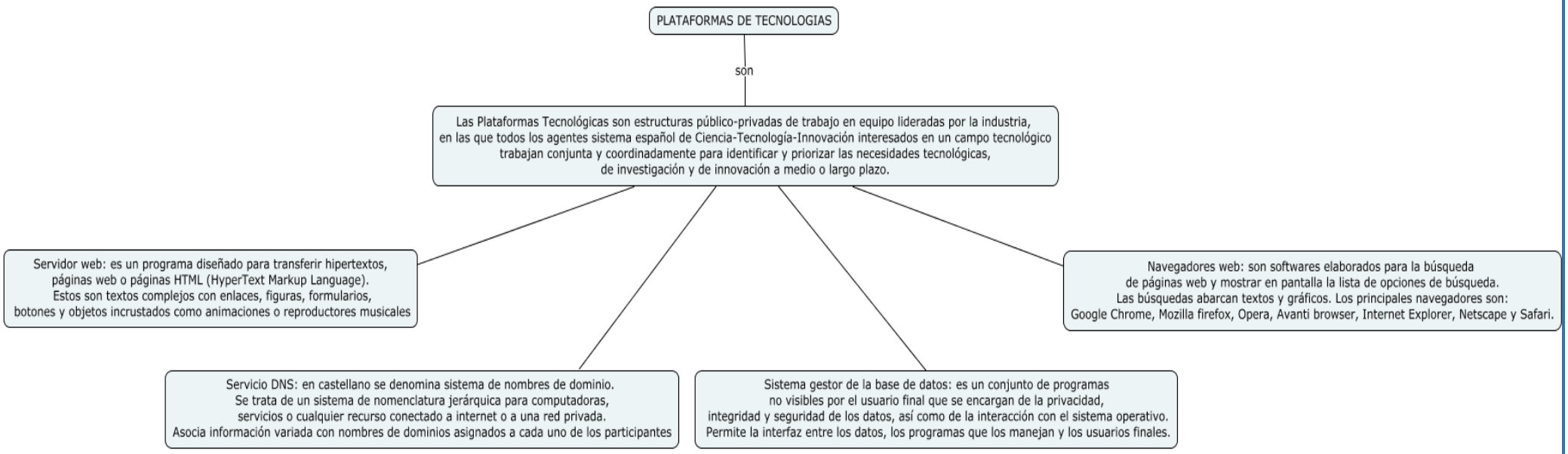
Servicio DNS: en castellano se denomina sistema de nombres de dominio. Se trata de un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a internet o a una red privada. Asocia información variada con nombres de dominios asignados a cada uno de los participantes. Su función más importante es "traducir" nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, con la finalidad de localizar y direccionar estos equipos mundialmente. Otra función clave de los protocolos DNS es la asignación de nombres a direcciones IP. Cada sitio web cuenta con una dirección IP, pero para tener acceso a ellas no digitamos dichas direcciones sino su nombre precedido de www y con una terminación como .org, .com, etc. La mayoría de usuarios acceden a los sitios web empleando estos nombres y no las direcciones IP.

Sistemas operativos: quizá es el programa más complejo e importante para una computadora. El sistema operativo "despierta" a la PC y le hace reconocer al CPU, la memoria, el teclado, el sistema de vídeo, la impresora y las unidades de disco. También facilita la comunicación para que los usuarios se comuniquen con la computadora y sirve de plataforma a partir de la cual se corran programas de aplicación.

Los sistemas operativos más conocidos son: DOS, Windows 3.1, OS/2, Mac OS y UNIX.

Sistema gestor de la base de datos: es un conjunto de programas no visibles por el usuario final que se encargan de la privacidad, integridad y seguridad de los datos, así como de la interacción con el sistema operativo. Permite la interfaz entre los datos, los programas que los manejan y los usuarios finales.

Navegadores web: son softwares elaborados para la búsqueda de páginas web y mostrar en pantalla la lista de opciones de búsqueda. Las búsquedas abarcan textos y gráficos. Los principales navegadores son: Google Chrome, Mozilla firefox, Opera, Avanti browser, Internet Explorer, Netscape y Safari.


PLATAFORMAS DE TECNOLOGIAS
son
Las Plataformas Tecnológicas son estructuras público-privadas de trabajo en equipo lideradas por la industria, en las que todos los agentes sistema español de Ciencia-Tecnología-Innovación interesados en un campo tecnológico trabajan conjunta y coordinadamente para identificar y priorizar las necesidades tecnológicas, de investigación y de innovación a medio o largo plazo.
Servidor web: es un programa diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language). Estos son textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores musicales
Navegadores web: son softwares elaborados para la búsqueda de páginas web y mostrar en pantalla la lista de opciones de búsqueda. Las búsquedas abarcan textos y gráficos. Los principales navegadores son: Google Chrome, Mozilla firefox, Opera, Avanti browser, Internet Explorer, Netscape y Safari.
Servicio DNS: en castellano se denomina sistema de nombres de dominio. Se trata de un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a internet o a una red privada. Asocia información variada con nombres de dominios asignados a cada uno de los participantes
Sistema gestor de la base de datos: es un conjunto de programas no visibles por el usuario final que se encargan de la privacidad, integridad y seguridad de los datos, así como de la interacción con el sistema operativo. Permite la interfaz entre los datos, los programas que los manejan y los usuarios finales.

5.6 Seguridad e interoperabilidad

La característica principal de un Servicio Web es que le permite cierto grado de flexibilidad, accesibilidad yinteroperabilidad. Esto permite que los desarrolladores abstraigan la lógica de negocio y se centren en el desarrollo del servicio sin preocuparse de los criterios anteriormente citados.

Como objetivos básicos a cubrir por la seguridad de un servicio WEB
El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) define interoperabilidad como la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

La característica principal de un Servicio Web es que le permite cierto grado de flexibilidad, accesibilidad y interoperabilidad. Esto permite que los desarrolladores abstraigan la lógica de negocio y se centren en el desarrollo del servicio sin preocuparse de los criterios anteriormente citados.

Como objetivos básicos a cubrir por la seguridad de un servicio WEB

- Es necesario asegurar que existe una autenticación mutua entre el cliente que accede a los servicios web y el proveedor de dichos servicios.

- Se debe mantener una política de autorización del acceso a recursos y, más importante, a operaciones y procesos en un entorno en el que debe administrarse y controlarse el acceso de clientes, proveedores, vendedores, competidores y los posibles ataques que reciban de personal externo.

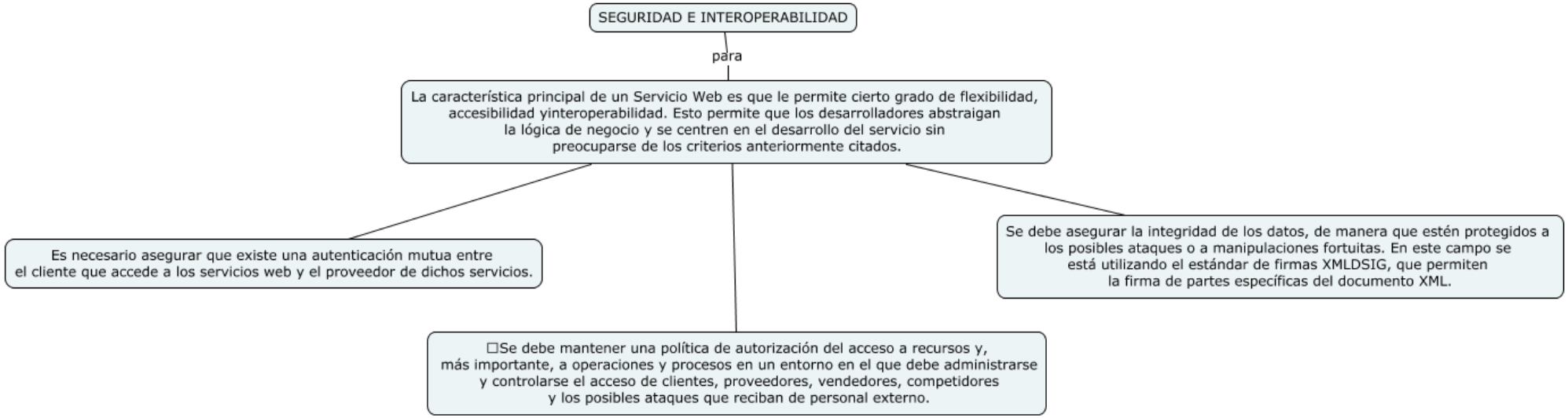
- Mantener al cliente identificado, de manera que se identifique una sola vez y pueda acceder a servicios en diversos sistemas, sin que resulte necesario identificarse nuevamente en cada uno de ellos.

- Controlar y asegurar la confidencialidad de los datos intercambiados, ya que SOAP no es capaz de cifrar la información, la cual viaja en claro a través de la red.Es necesario asegurar la comunicación con algún estándar que permita crear un canal seguro de comunicación. El estándar ya firmemente establecido de creación de canales seguros SSL y el cifrado de partes específicas de documentos mediante el cifrado XML son las direcciones que se están siguiendo en este terreno.

- Se debe asegurar la integridad de los datos, de manera que estén protegidos a los posibles ataques o a manipulaciones fortuitas. En este campo se está utilizando el estándar de firmas XMLDSIG, que permiten la firma de partes específicas del documento XML.
- Comprobar que no se repudian las operaciones, para lo cual es necesario mantener firmas en XML.

- A continuación se ofrece una tabla resumen con los los principales elementos de seguridad dentro de los servicios Web, así como las recomendaciones de madeja al respecto.

Seguridad		Recomendación	Elemento	Mecanisno
Autenticación de servicios	HTTP autenticación	No		
	SSL x509 autenticación	No		
	WS-Security-tokens	Si		
Autnticación de usuarios	SAML	Si		
Integridad	SSL	Si		
	Ws-Signature			
No repudio	WS/Signature	Si		
	WS-Adressing			
	Logs			
Confidencialidad	SSL	Si		


SEGURIDAD E INTEROPERABILIDAD
para
La característica principal de un Servicio Web es que le permite cierto grado de flexibilidad, accesibilidad yinteroperabilidad. Esto permite que los desarrolladores abstraigan la lógica de negocio y se centren en el desarrollo del servicio sin preocuparse de los criterios anteriormente citados.
Es necesario asegurar que existe una autenticación mutua entre el cliente que accede a los servicios web y el proveedor de dichos servicios.
☐Se debe mantener una política de autorización del acceso a recursos y, más importante, a operaciones y procesos en un entorno en el que debe administrarse y controlarse el acceso de clientes, proveedores, vendedores, competidores y los posibles ataques que reciban de personal externo.
Se debe asegurar la integridad de los datos, de manera que estén protegidos a los posibles ataques o a manipulaciones fortuitas. En este campo se está utilizando el estándar de firmas XMLDSIG, que permiten la firma de partes específicas del documento XML.

CONCLUSION

Llegamos a la conclusión que aprendimos a utilizar diferentes herramientas para llegar a nuestra meta, el análisis, el diseño, el proceso distintas partes del sistema que nos llevaron a comprender y aprender como realizar partiendo de cero.

También pudimos notar que un buen análisis y diseño nos da como resultado un sistema con buen funcionamiento y escalable.

En la programación Web php, html son lenguajes que permite codificar o preparar documentos de hipertexto, que viene a ser los lenguajes más comunes para la construcción de las páginas Web.Con el comienzo de Internet y la programación web, se desfasaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, se ha optado más por el diseño sencillo y de fácil comprensión.

Referencias:

https://programacion1class.wordpress.com/6-3-seguridad-e-interoperabilidad/
https://www.esan.edu.pe/apuntes-empresariales/2016/08/plataformas-tecnologicas-para-el-desarrollo-de-sitios-web/
https://www.ibm.com/support/knowledgecenter/es/SSCLKU_7.5.5/org.eclipse.jst.ws.doc.user/concepts/cwsstandards.html
https://devexperto.com/patrones-de-diseno-software/
https://www.eoi.es/blogs/volkerbachmann/2012/01/14/tipos-de-servicios-dentro-del-cloud-computing/