Desarrollo de aplicaciones para ambientes distribuidos

Reyson Antonio Antonio

1.1 Capa de interfaz de usuario.

1.1 Capa de interfaz de usuario.

La capa de usuario consiste en el mecanismo de interacción entre el usuario y el sistema, comunicando y capturando la información de usuario en un mínimo proceso. Esta capa es la responsable de obtener datos de la capa siguiente, mostrarlos, validar entradas de datos, enviarlas a la siguiente capa donde pueden dividirse en: presentación, código de interfaz de usuario .

Bazurto, J. C., Florencia, A., Rojas, A., & Solórzano, J. (2018). Arquitectura de aplicaciones distribuidas. *Ciencia Huasteca Boletín Científico de la Escuela Superior de Huejutla*, *6*(11).

1.2 Capa de manejo de datos

1.2 Capa de manejo de datos

También conocida como capa de negocios, en donde residen los programas que se ejecutan, se reciben peticiones del usuario y se envían las respuestas después del proceso. Esta capa se comunica con la capa de presentación o interfaz de usuario y con la capa de datos, por lo cual es la capa intermedia entre las antes mencionadas.

Componentes de Capa de negocios.

Lógica de negocios: encargada de todas las operaciones, ya sea usando los datos obtenidos desde la capa de datos, o simplemente con datos externos.

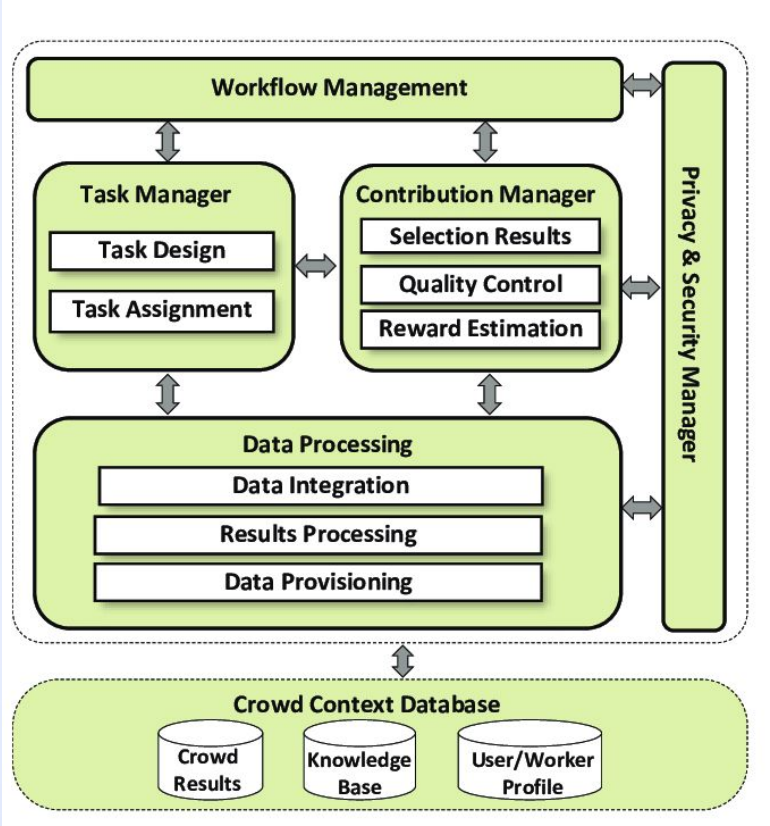
Componentes de Capa de negocios.

Lógica de acceso a datos: incluye los elementos necesarios para que la aplicación se conecte a orígenes de datos y recupere estructuras de datos que serán utilizadas por el resto de la aplicación.

1.3 CAPA DE PROCESAMIENTO DE DATOS

1.3 CAPA DE PROCESAMIENTO DE DATOS

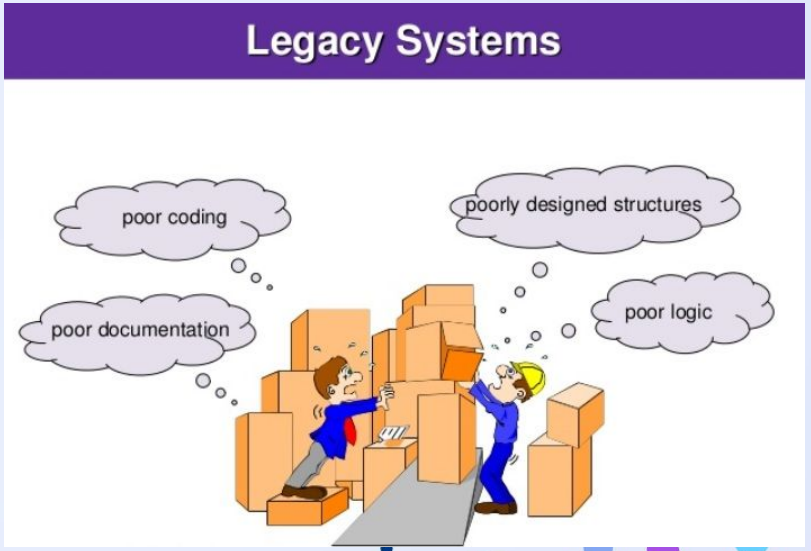
Esta capa es donde residen los datos y es la encargada de acceder a los mismos. Puede estar formada por uno o más gestores de base de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de manejo de datos.



1.4 Integración de sistemas heredados.

1.4 Integración de sistemas heredados.

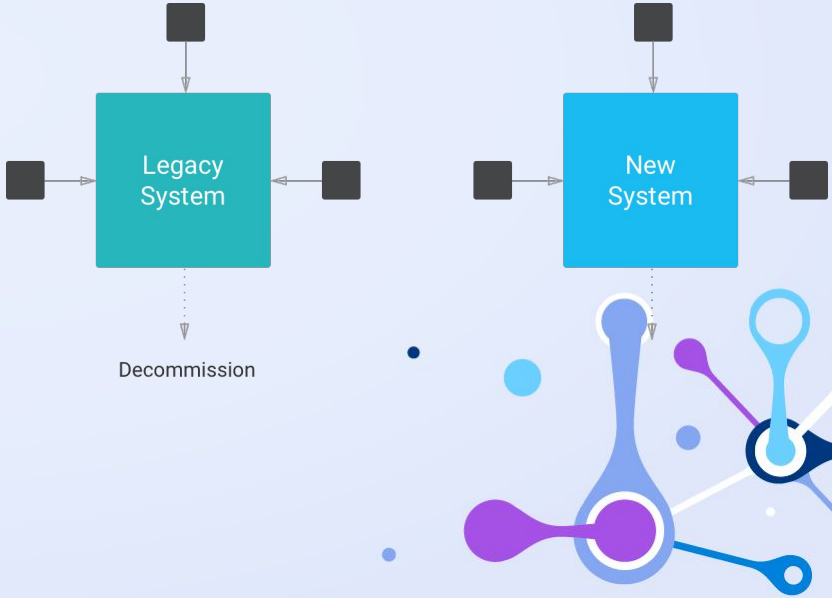
Los sistemas de información legacy o heredados fueron creados con la finalidad de automatizar procesos que antes de la invención de la informática se hacían de forma manual



1.4 Integración de sistemas heredados.

Un sistema heredado es un componente técnicamente obsoleto de un entorno de gestión de contenido.

Aunque la funcionalidad que un sistema heredado ofrece a los procesos empresariales puede estar disponible a través de una tecnología más moderna,



1.4 Integración de sistemas heredados.

La posibilidad de una interrupción del servicio durante la actualización de sistemas puede impedir una migración hacia el uso de sistemas más nuevos, o incluso la puede impedir dada la dificultad percibida en la conversión del contenido heredado para ajustarse a los nuevos modelos de contenido y formatos.

1.4 Integración de sistemas heredados.

La integración de sistemas heredados brinda un **método no intrusivo** para reutilizar aplicaciones críticas que residen en sistemas heredados, como un sistema mainframe o AS/400. El poder utilizar estos recursos existentes tiene muchas ventajas, entre ellas **un riesgo reducido** y **ahorros significativos**.

1.5. Distribución de elementos de una aplicación.

1.5. Distribución de elementos de una aplicación.

Se refiere a la construcción de software por partes, a las cuales les son asignadas un conjunto específico de **responsabilidades** dentro de un sistema.

1.5. Distribución de elementos de una aplicación.

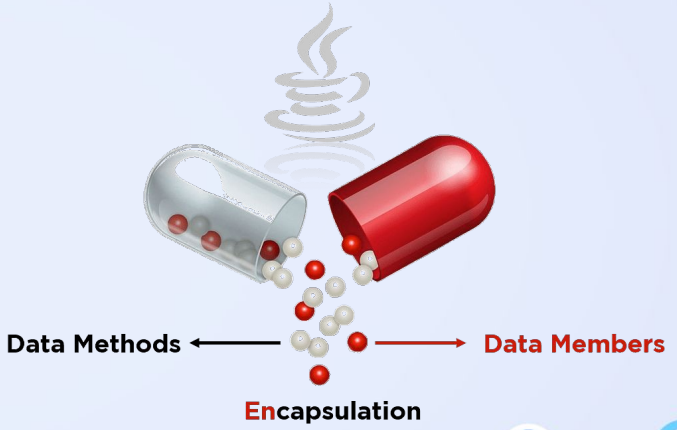
Entre los componentes que se encuentran en entornos separados.
Separación física y lógica de las partes de una aplicación:

- Separación física(niveles): considera **aspectos técnicos y económicos.**
- Separación Lógica(capas): conjunto de **servicios especializados** que son accedidos por múltiples clientes.

1.5. Distribución de elementos de una aplicación.

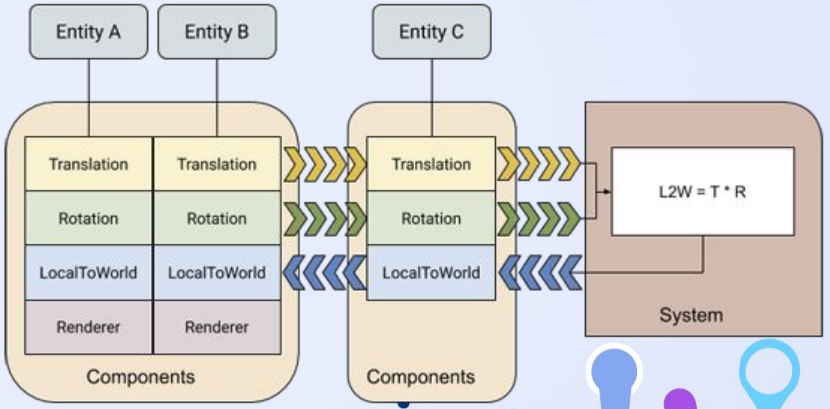
Componente: Es un elementos de software que **encapsula** una serie de **funcionalidades**.

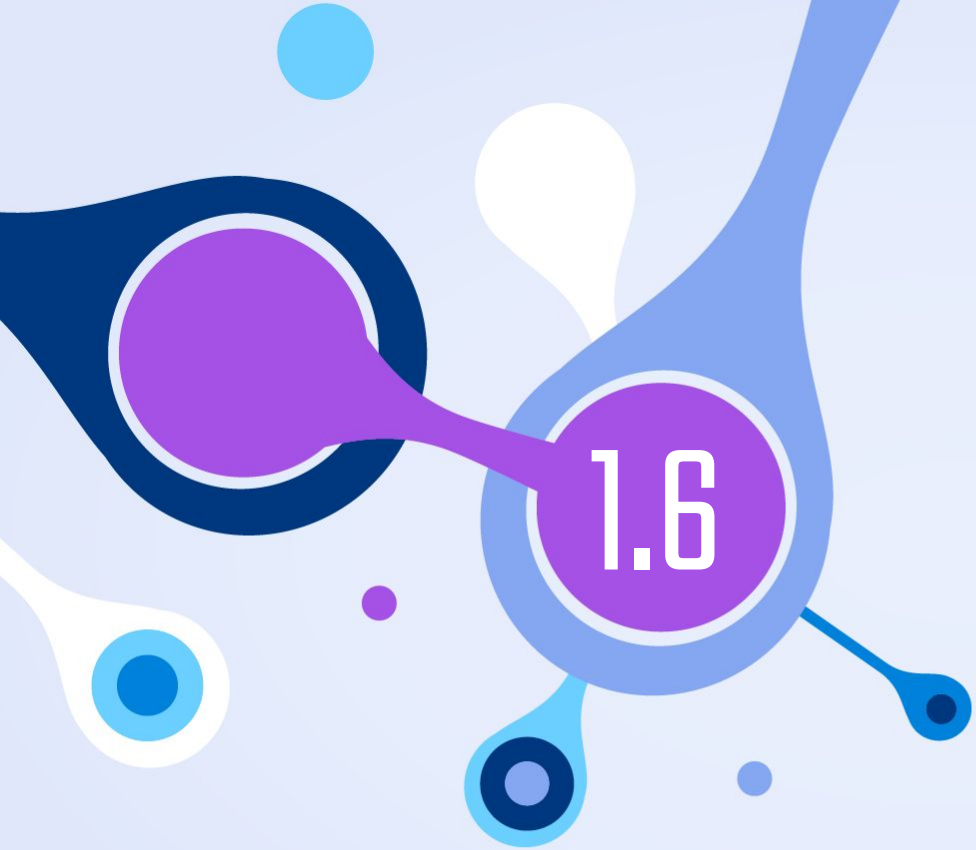
Es una unidad independiente(puede estar compuesto por **clases** ó recursos complementarios archivos imágenes entre otros)



1.5. Distribución de elementos de una aplicación.

La distribución habla de que las partes o **componentes** se encuentran en entornos separados, entonces para realizar la separación física, primero se debe de tener clara la **separación lógica** de las partes de una aplicación





1.6. Integración de tecnologías heterogéneas y homogéneas.

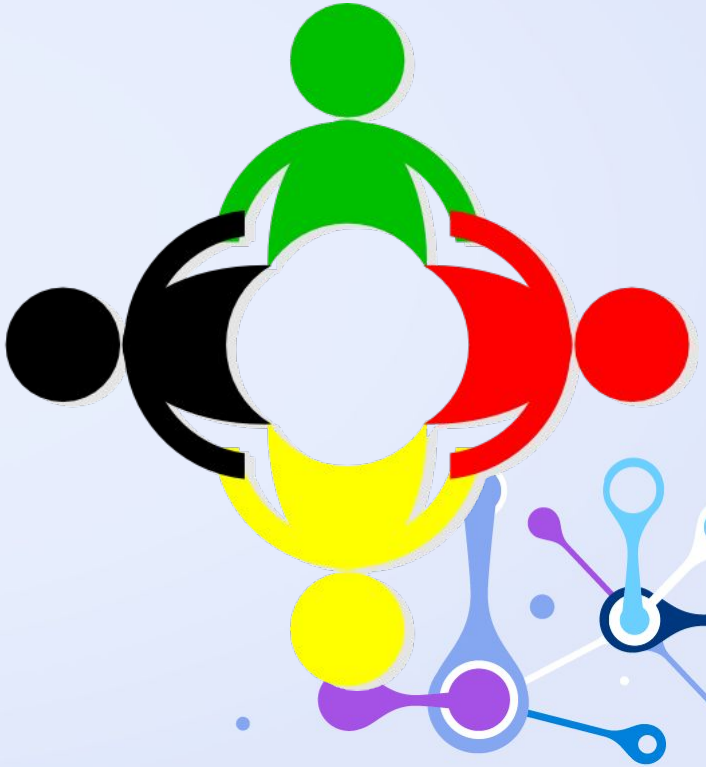
1.6. Integración de tecnologías heterogéneas y homogéneas.

Existen diferentes motivos para la heterogeneidad y homogeneidad. Una razón son los **cambios tecnológicos que siempre se dan en un periodo de tiempo corto**. En este contexto, dichos cambios se refieren a **mejor calidad**, mejor **desempeño**, costos más **económicos**, **seguridad**, entre otras características que se toman en cuenta.

1.6. Integración de tecnologías heterogéneas y homogéneas.

HOMOGÉNEO:

En los sistemas homogéneos, todos los sitios **emplean idéntico software** de gestión de base de datos, son **conscientes** de la existencia de los demás sitios y **acuerdan cooperar** en el **procesamiento de las solicitudes** de los usuarios



1.6. Integración de tecnologías heterogéneas y homogéneas.

HETEROGÉNEO: Son aquellas donde Sitios diferentes utilizan diferentes DBMS, siendo cada uno esencialmente autónomo.

Es posible que algunos sitios **no sean conscientes** de la existencia de los demás y quizás proporcionen **facilidades limitadas** para la cooperación en el procesamiento de transacciones. La heterogeneidad se debe a que **los datos de cada BD son de diferentes tipos o formatos.** El enfoque heterogéneo es más complejo que el enfoque homogéneo.

1.7. SERVICIOS DE LA ARQUITECTURA (EMAIL, WEB, BASE DE DATOS, APLICACIONES, TRANSACCIONES, SISTEMAS OPERATIVOS, FIREWALL)

Servicio Web:

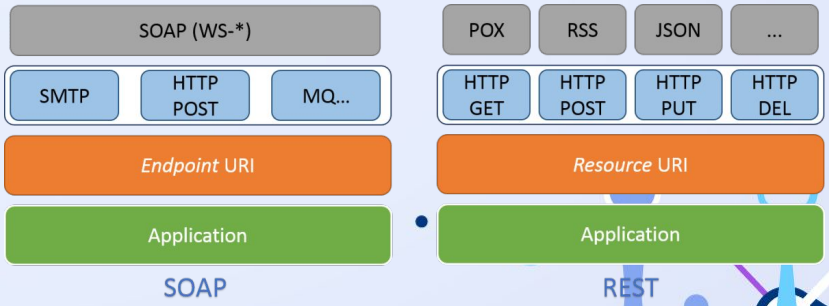
Es un conjunto de **protocolos** y estándares que sirven para **intercambiar datos** entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet.

Protocolos utilizados:

* XML: Es el formato estándar para los datos que se vayan a intercambiar.

* SOAP o XML-RPC: Protocolos sobre los que se establece el intercambio.

* HTTP, FTP, o SMTP: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales ya bien conocidos.

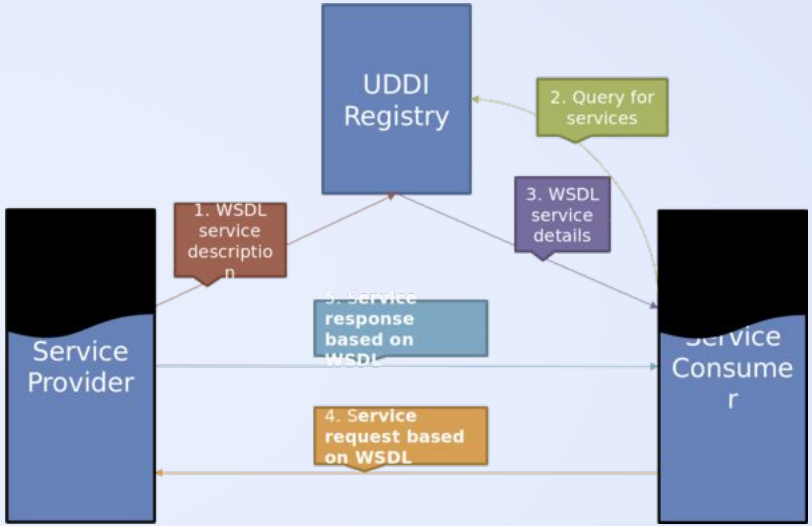


Protocolos utilizados:

* WSDL: Es el lenguaje de la interfaz pública para los servicios Web.

* UDDI: Protocolo para publicar la información de los servicios Web.

* WS-Security: Protocolo de seguridad aceptado como estándar por OASIS.

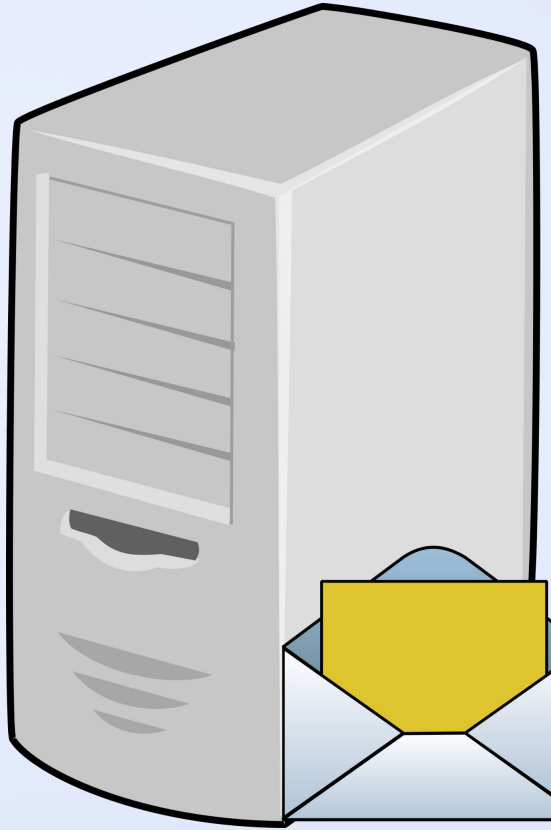


1.7. SERVICIOS DE LA ARQUITECTURA (EMAIL, WEB, BASE DE DATOS, APLICACIONES, TRANSACCIONES, SISTEMAS OPERATIVOS, FIREWALL)

Estos servicios proporcionan **mecanismos de comunicación** estándares entre diferentes aplicaciones, que interactúan entre sí para presentar **información dinámica** al usuario. Para proporcionar **interoperabilidad** y **extensibilidad** entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

SERVICIO EMAIL

Servicio de red que permite a los usuarios **enviar y recibir mensajes** rápidamente también denominados **mensajes electrónicos** o cartas electrónicas mediante sistemas de comunicación electrónicos.



Aplicaciones para envío y recepción de emails

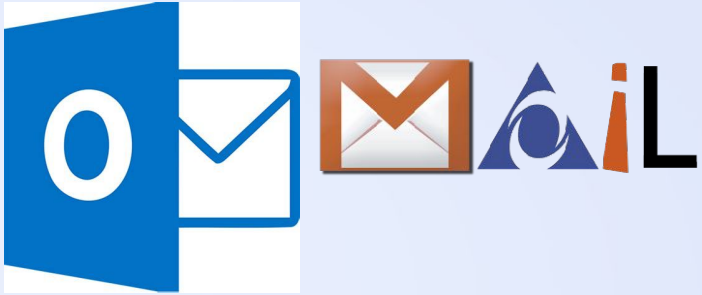
Los mensajes de email son **intercambiados entre hosts** que emplea el **protocolo SMTP**, con programas llamados **agentes de transferencia** de emails.

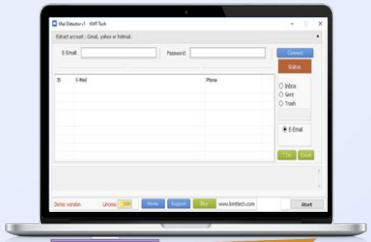
En tanto, los usuarios pueden recibir sus emails de los servidores empleando protocolos como **POP o IMAP**; o protocolos propietarios específicos como Lotus Notes o Microsoft Exchange Servers. También pueden emplear **servicios de webmail** para acceder a los emails desde un navegador

Ejemplos De Servicio Email

- Aplicaciones online: Existen servicios online dedicados al webmail como Gmail, Hotmail, Yahoo! Mail, etc. En general, un usuario debe registrarse al servicio de e-mail para obtener una cuenta de correo electrónico.

- Aplicaciones de computadora: Algunas aplicaciones que se utilizan para el envío y recepción de e-mails son: Microsoft Outlook Express, Microsoft Outlook, Netscape Mail, Eudora y Pegasus Mail.





BASE DE DATOS

Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Un servidor de base de datos **es un programa que provee servicios de base de datos a otros programas u otras computadoras**, como es definido por el modelo cliente-servidor. También puede hacer referencia a aquellas computadoras (servidores) dedicadas a ejecutar esos programas, prestando el servicio.

BASE DE DATOS

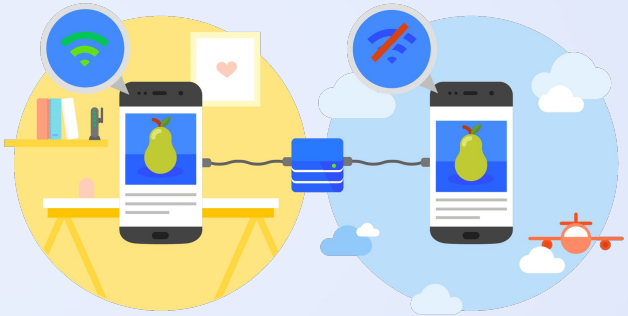
Los servidores de datos deben proporcionar **mecanismos de comunicación óptimos**, pues de cómo se envíe la información dependen parámetros tan importantes como la **velocidad de acceso** a los datos. Todos los sistemas gestores analizados cuentan con múltiples configuraciones de protocolos, adaptándose a los protocolos existentes y estandarizados de la actualidad: **TCP/IP, IPX, Banyan**, ect; es importante no sólo el canal de comunicaciones que está disponible para los servidores de datos sino también cómo es transmitida la información.

APLICACIONES

Programa informáticos que permiten a un usuario utilizar una computadora **con un fin específico**. Son parte del software de una computadora y suelen **ejecutarse sobre el sistema operativo.**

¿Qué es un proveedor de servicios de aplicación?

El modelo de negocios Proveedor de Servicios de Aplicación o más conocido en inglés como ASP (Application Service Provider) permite a una organización utilizar un software de aplicación bajo el concepto de servicio, sin la necesidad de comprar licencias, equipos y otros activos, pagando sólo una cuota mensual.



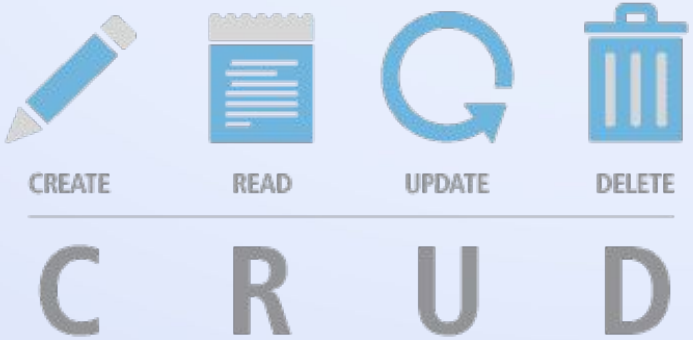
El proveedor está especializado en manejar volúmenes más eficientes con costos fijos consiguiendo una mayor calidad de servicio.

Las ventajas del modelo son muchas, entre las cuales podemos destacar:

- Menores costos de implementación
- Costos de mantenimiento predecible y acordado.
- Ahorra la inversión inicial para construir la infraestructura requerida para correr las aplicaciones.
- No se requiere disponer de recursos técnicos para la implementación.
- Rápida puesta en producción y con menor riesgo.
- Mayor seguridad.
- Mayor escalabilidad.

TRANSACCIONES

Una transacción es una interacción con una estructura de datos compleja compuesta por varios procesos que se han de aplicar uno después del otro.



FIREWALL

Es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado permitiendo a los mismos tiempos comunicaciones autorizadas cortafuegos, mecanismo de seguridad en internet frente a acceso no autorizados.

2.1 Diseño de la base de datos distribuidas

https://docs.google.com/document/preview?hgd=1&id=1odrBK-0PSINWQ7wvpFJBysyxxUYelWVCPJ_0D5WMzYI#

Diseño de BD

El diseño de una base de datos consiste en **definir la estructura de los datos** que debe tener la base de datos de un sistema de información determinado. En el caso relacional, esta estructura será un conjunto de esquemas de **relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, etc.**

Proceso:

Se **descompone** el diseño de bases de datos en tres etapas:

1) Etapa del diseño **conceptual**

2) Etapa del diseño **lógico**

3) Etapa del diseño **físico**

Etapa del diseño conceptual:

Se obtiene una estructura de la información de la futura BD independiente de la tecnología que hay que emplear.

No se tiene en cuenta todavía qué tipo de base de datos se utilizará relacional, orientada a objetos, jerárquica, etc.; en consecuencia, tampoco se tiene en cuenta con qué SGBD ni con qué lenguaje concreto se implementará la base de datos.

Etapa del diseño conceptual:

Así pues, la etapa del diseño conceptual nos permite concentrarnos únicamente en la problemática de la estructuración de la información, sin tener que preocuparnos al mismo tiempo de resolver cuestiones tecnológicas.

Etapa del diseño conceptual:

El resultado de la etapa del diseño conceptual se expresa mediante algún modelo de datos de alto nivel. Uno de los más empleados es el modelo entidad- interrelación (entity relationship), que abreviaremos con la sigla ER.

Etapa del diseño lógico:

Se parte del resultado del diseño conceptual, que se transforma de forma que se adapte a la tecnología que se debe emplear. Más concretamente, es preciso que se **ajuste al modelo del SGBD** con el que se desea implementar la base de datos. Por ejemplo, si se trata de un SGBD relacional, esta etapa obtendrá un conjunto de relaciones con sus atributos, claves primarias y claves foráneas.

Etapa del diseño lógico:

Esta etapa parte del hecho de que ya se ha resuelto la problemática de la estructuración de la información en un ámbito conceptual, y permite concentrarnos en las cuestiones tecnológicas relacionadas con el modelo de base de datos.

Etapa del diseño físico

Se transforma la estructura obtenida en la etapa del diseño lógico, con el objetivo de conseguir una mayor eficiencia; además, se completa con aspectos de implementación física que dependerán del SGBD.

Etapa del diseño físico

Por ejemplo, si se trata de una base de datos relacional, la transformación de la estructura puede consistir en lo siguiente: tener almacenada alguna relación que sea la combinación de varias relaciones que se han obtenido en la etapa del diseño lógico, partir una relación en varias, añadir algún atributo calculable a una relación, etc.

Etapa del diseño físico

Los aspectos de implementación física que hay que completar consisten normalmente en la elección de estructuras físicas de implementación de las relaciones, la selección del tamaño de las memorias inter- medias (buffers) o de las páginas, etc.

Etapa del diseño físico

En la etapa del diseño físico –con el objetivo de conseguir un buen rendimiento de la base de datos–, se deben tener en cuenta las características de los procesos que consultan y actualizan la base de datos, como por ejemplo los caminos de acceso que utilizan y las frecuencias de ejecución. También es necesario considerar los volúmenes que se espera tener de los diferentes datos que se quieren almacenar.

https://bettyvp.wordpress.com/2014/05/07/2-1-el-proceso-de-diseno-de-una-base-de-datos/

2.2 Diseño de la fragmentación distribuida

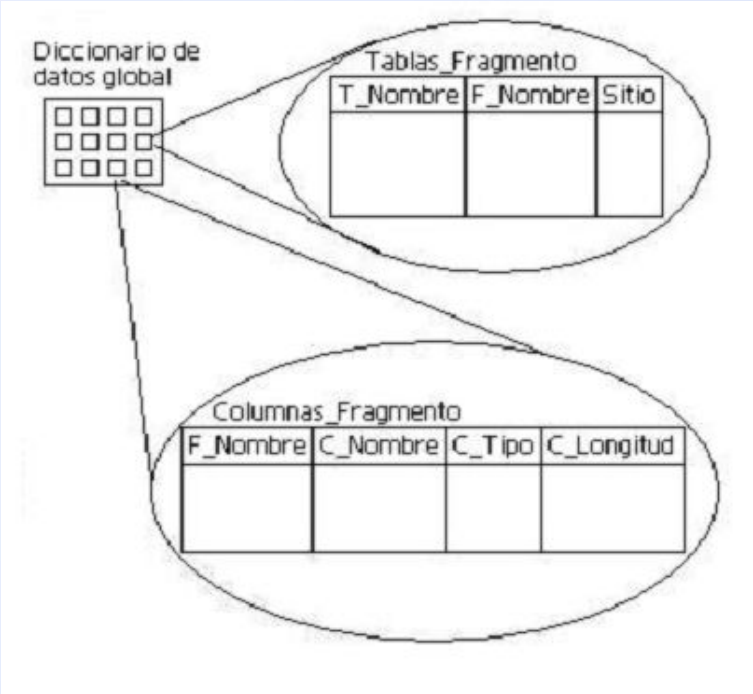
2.2 Diseño de la fragmentación distribuida

En el proceso de diseño de una Base de Datos Distribuida, **considerando un diseño descendente**, se parte de diseñar la base de datos global **como si está fuera a implementarse de manera centralizada.** Una vez elaborado este diseño, mismo que se refleja en el esquema global en la Arquitectura de la Base de Datos Distribuida, se procede a **definir los fragmentos** en que dividirá la base de datos global y que constituirán la base de datos fragmento.

https://adonaihg.wordpress.com/2017/02/20/diseno-de-base-de-datos-distribuidas/

2.2 Diseño de la fragmentación distribuida

El Esquema de Fragmentación **expresa**, para todos los fragmentos que constituyen la base de datos fragmento, su **correspondencia** con los objetos de datos que constituyen la base de datos global, de los cuales particularmente se han derivado.



https://adonaihg.wordpress.com/2017/02/20/diseno-de-base-de-datos-distribuidas/

2.2 Diseño de la fragmentación distribuida

Aún más, en el Esquema de Fragmentación no solo se refleja la correspondencia entre los fragmentos con los objetos de datos de la base de datos global de los que se han derivado, sino también la correspondencia de ciertos fragmentos, con los fragmentos de los que se han derivado.

https://adonaihg.wordpress.com/2017/02/20/diseno-de-base-de-datos-distribuidas/

2.2 Diseño de la fragmentación distribuida

Es decir, las consideraciones de diseño de la fragmentación, en la mayoría de los casos de diseño de Bases de Datos Distribuidas, implican la necesidad de dividir a su vez en fragmentos, algunos fragmentos que se han derivado de objetos de datos de la base de datos global, e incluso que se han derivado de otros fragmentos ya definidos.

https://adonaihg.wordpress.com/2017/02/20/diseno-de-base-de-datos-distribuidas/

1.-Fragmentación Horizontal

Una tabla T se divide en subconjuntos, T1, T2, ...Tn. Los fragmentos se definen a través de una operación de selección y su reconstrucción se realizará con una operación de unión de los fragmentos componentes. Cada fragmento se sitúa en un nodo. Pueden existir fragmentos no disjuntos: combinación de fragmentación y replicación

https://bettyvp.wordpress.com/2014/05/07/2-1-el-proceso-de-diseno-de-una-base-de-datos/

2.-Fragmentación Vertical

Una tabla T se divide en subconjuntos, T1, T2, ...Tn. Los fragmentos se definen a través de una operación de proyección. Cada fragmento debe incluir la clave primaria de la tabla. Su reconstrucción se realizará con una operación de join de los fragmentos componentes, pueden existir fragmentos no disjuntos: combinación de fragmentación y replicación.

https://bettyvp.wordpress.com/2014/05/07/2-1-el-proceso-de-diseno-de-una-base-de-datos/

3.-Fragmentación Mixta

Como el mismo nombre indica es una combinación de las dos anteriores vistas he aquí un ejemplo a partir de una tabla fragmentada horizontalmente.

https://bettyvp.wordpress.com/2014/05/07/2-1-el-proceso-de-diseno-de-una-base-de-datos/

2.3 Diseño de la distribución de datos

2.4 Diseño general del sistema