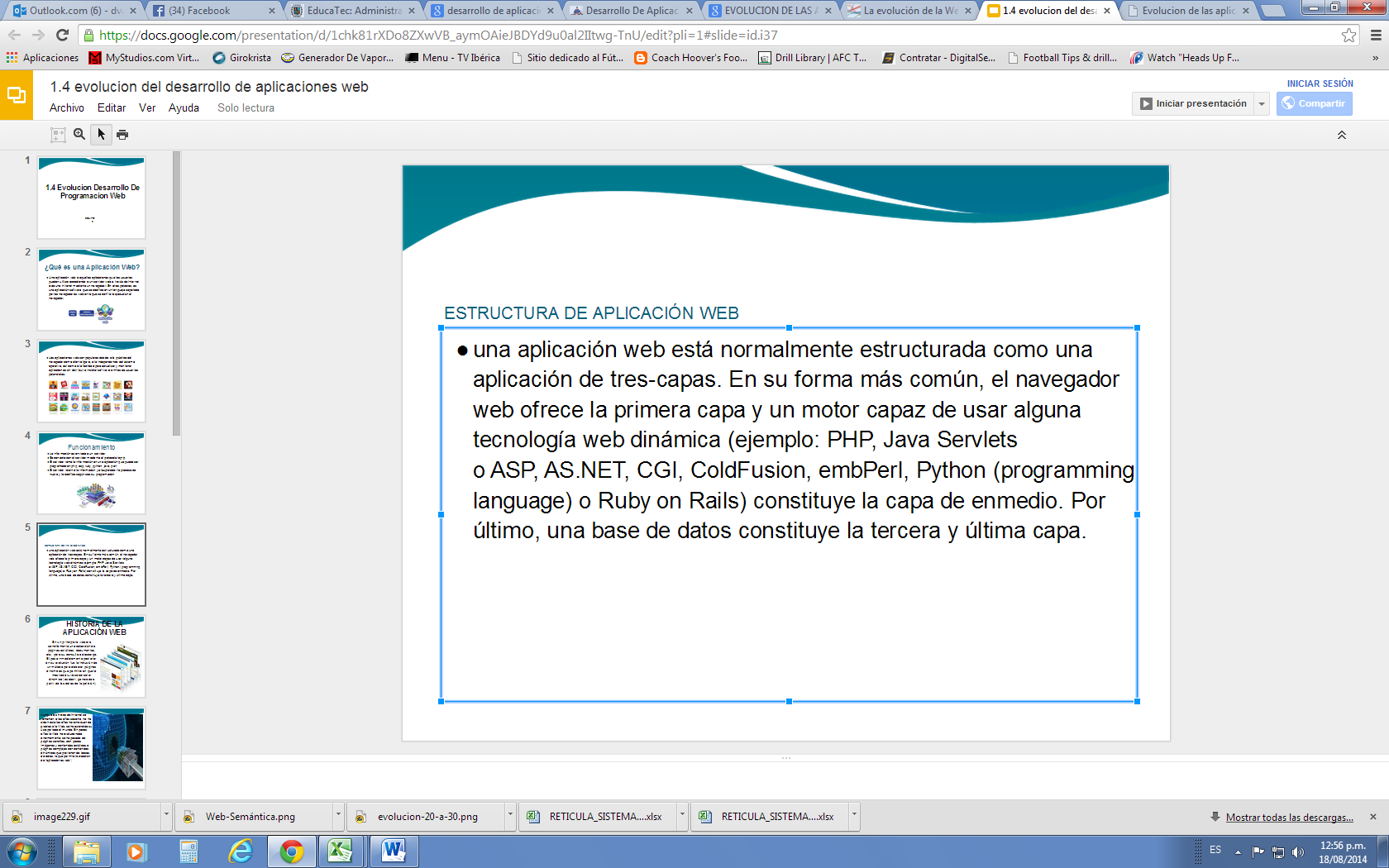
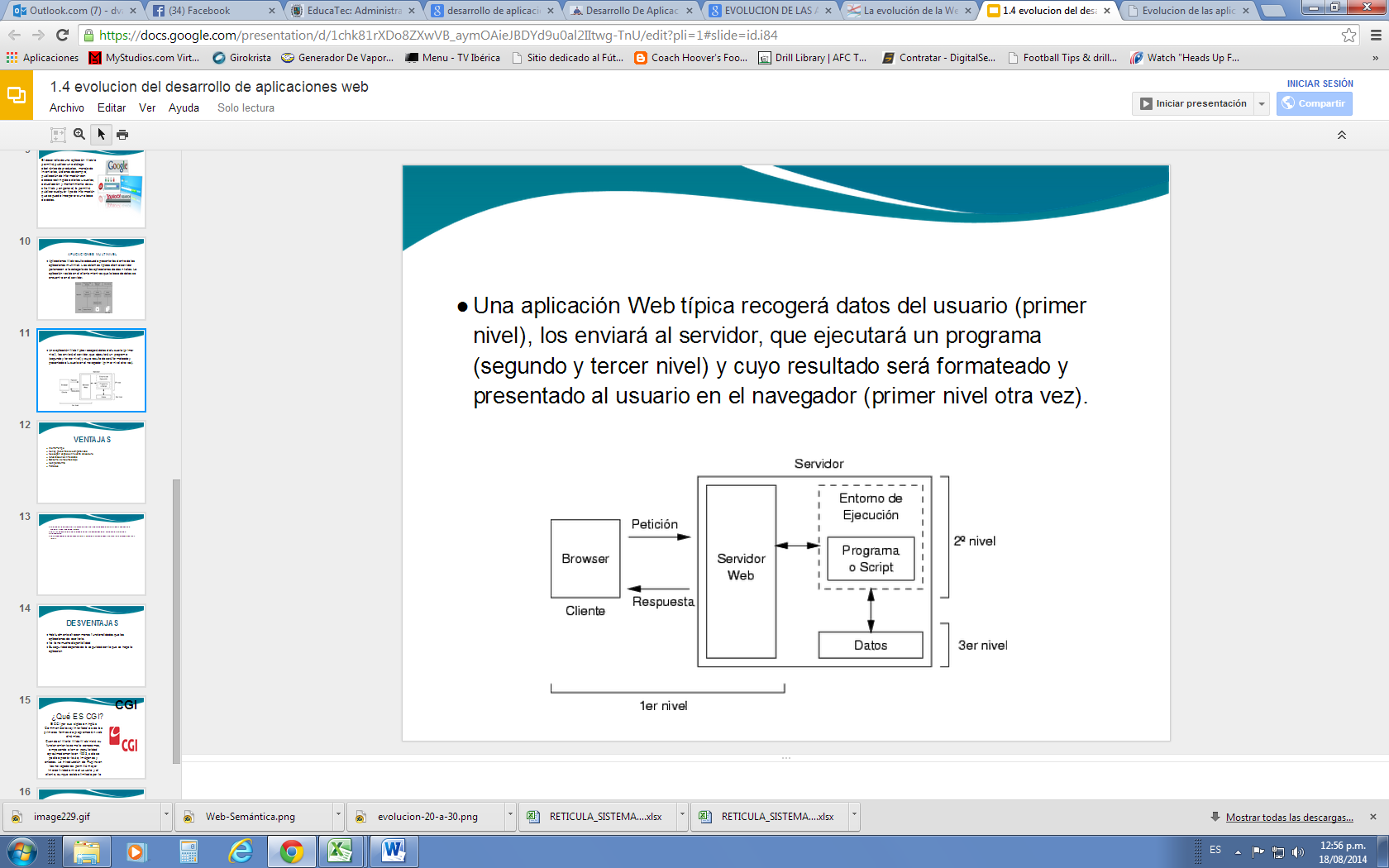
1.2. Arquitecturas de la tecnología Cliente –

Servidor.

1.2.1. Aplicaciones de 2, 3 y n capas.





La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles.

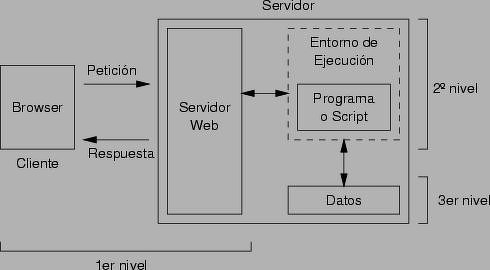
**El primer nivel** consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado.

**El segundo nivel** está referido habitualmente a algún tipo de programa o *script*. Finalmente, 

**el tercer nivel** proporciona al segundo los datos necesarios para su ejecución.

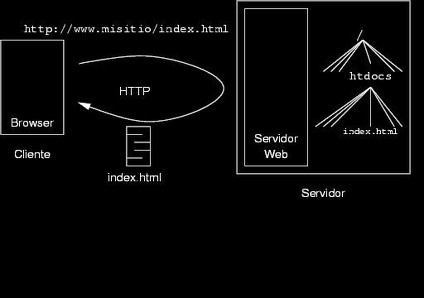
Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

**Arquitectura Web de tres niveles**



Lamentablemente, el uso de toda esta tecnología pasa por el dominio de técnicas de programación y de acceso a bases de datos.

Para abrir una página Web en un navegador, normalmente se teclea el correspondiente URL o se pica en el hiperenlace oportuno. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de ficheros y la envía de vuelta al navegador que la solicitó, según se muestra en la Figura, Arquitectura web básica.



**Arquitecturas de la tecnología Cliente-Servidor**

Arquitectura web básica

Para abrir una página Web en un navegador, normalmente se teclea el correspondiente URL o se pica en el hiperenlace oportuno. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de ficheros y la envía de vuelta al navegador que la solicitó como se muestra en la imagen.

Aplicaciones multinivel

Al hablar de desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: interfaz de presentación, lógica de la aplicación y los datos.

La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos.

El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o *script*. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

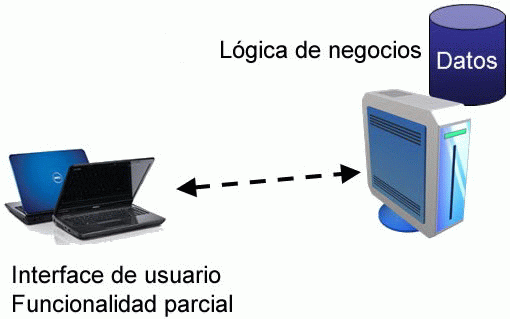
Aplicaciones de N capas

El surgimiento de la tecnología de componentes distribuidos es la clave de las arquitecturas de n-capas. Estos sistemas de computación utilizan un número variable de componentes individuales que se comunican entre ellos utilizando estándares predefinidos y frameworks de comunicación.

**2.1. Arquitectura de las aplicaciones Web**

La arquitectura tradicional de cliente/servidor tamién es conocida como arquitectura de dos capas. Requiere una interfaz de usuario que se instala y se ejecuta en una PC o estación de trabajo y envía solicitudes a un servidor para ejecutar operaciones complejas.

Por ejemplo, una estación de trabajo utilizada como cliente puede ejecutar una aplicación de interfaz de usuario que interroga a un servidor central de ases de datos.



**Ventajas del Sistema de Dos Capas:**

El desarrollo de aplicaciones en un ambiente de dos capas funciona adecuadamente, pero no es necesariamente lo más eficiente. Las herramientas para el desarrollo con dos capas son robustas y ampliamente evaluadas.

Las técnicas de ingeniería de software de prototipo se emplean fácilmente. Las soluciones de dos capas trabajan en ambientes no dinámicos estales, pero no se ejecutan bien en organizaciones rápidamente cambiantes.

**Desventajas del sistema de dos capas:**

Los ambientes de dos capas requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación cuando se les hacen cambios. Esto se debe al hecho de que la mayoría de la aplicación lógica existe en la estación de trabajo del cliente.

La seguridad del sistema en un diseño de dos capas es compleja y a menudo requiere administración de las bases de datos; esto es deido al número de dispositivos con acceso directo al amibente de esas bases de datos.

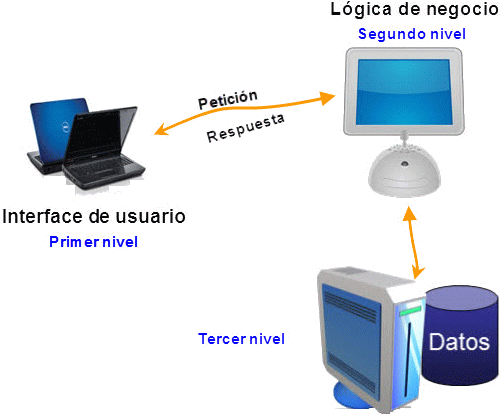
Las herramientas del cliente y de la base de datos, utilizadas en diseños de dos capas, constantemente están cambiando. La dependencia a largo plazo de cualquier herramienta, puede complicar el escalamiento futuro o las implementaciones.

**Arquitectura de tres capas**

La arquitectura de tres capas es un diseño reciente que introduce una capa intermedia en el proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final (el cliente). La arquitectura basada en Web transforma la interfaz de búsqueda existente (el explorador de Web), en la interfaz del usuario final.

**La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles**

* El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de presentar los datos un formato adecuado.
* El segundo nivel está referido habitualmente a algún tipo de programa o script.
* Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución. Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).



Las diferentes capas suelen ser:

* **Capa 1**: Cliente de aplicación: Navegador Web
* **Capa 2**: Servidor de Aplicaciones: Apache, Servidor Tomcat con servlet’s
* **Capa 3**: Servidor de Datos: base de datos, servidor SMTP…

**Ventajas de la arquitectura de tres capas:**

Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexiles que en el diseño de dos capas, ya que la estación solo necesita transferir parámetros a la capa intermedia.

Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.

El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseíado en formato modular.

La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.

**Desventajas de las Arquitecturas de Tres Capas y basadas en Web**

Los ambientes de tres capas pueden incrementar el tráfico en la red y requiere más balance de carga u tolerancia a las fallas.

Los exploradores actuales no son todos iguales.

La estandarización entre diferentes proveedores ha sido lenta en desarrollarse. Muchas organizaciones son forzadas a escoger uno en lugar de otro, mientras que cada uno ofrece sus propias y distintas ventajas.