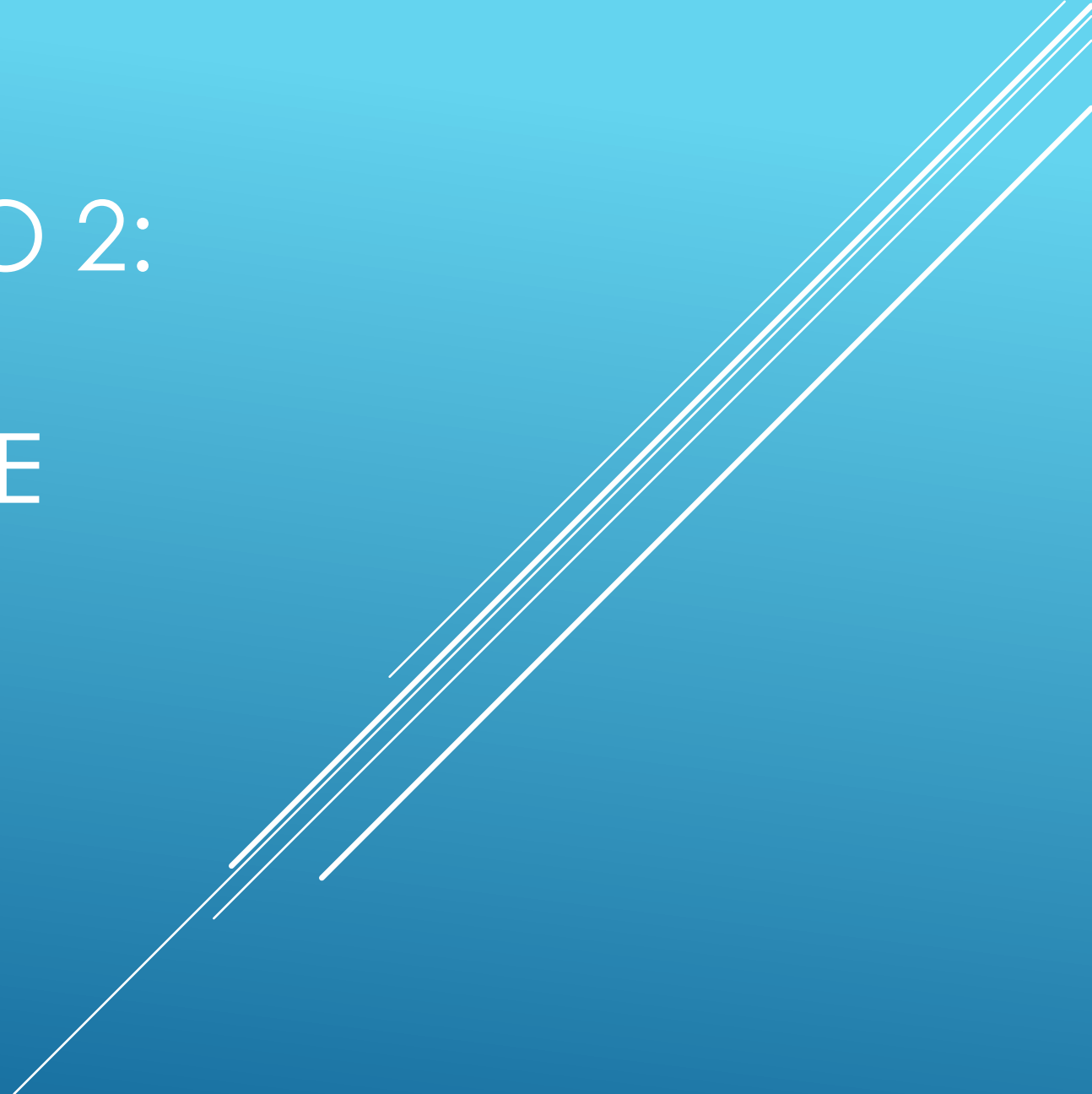


# UNIDAD DE TRABAJO 2: CONFIGURACIÓN Y ADMINISTRACIÓN DE SERVIDORES WEB.



# FUNCIONAMIENTO DE UN SERVIDOR WEB.

- Páginas **estáticas** es simple, puesto que solamente se necesita que el servidor web disponga de soporte html/xhtml/css
- En la instalación de Apache se crea una página web en /var/www/html/index.html referenciada a través del archivo /etc/apache2/sites-available/000-default.conf
- Una página **dinámica** se modificada mediante una interacción con el usuario y/o el navegador. Lenguajes dinámicos JavaScript, VBScript, PHP, ASP, JSP, etc.
- El protocolo **HTTPS** permite que la información viaje de forma segura entre el cliente y el servidor, por la contra el protocolo **HTTP** envía la información en texto claro. HTTPS utiliza cifrado sobre **SSL/TLS** y el puerto **443**.
- El protocolo **HTTPS** requiere de **certificados** y siempre y cuando sean validados la información será transferida cifrada, esto requiere un tiempo de computación.
- Una lista de **Entidades Certificadoras** que verifican, autentican y dan validez a los certificados.
- El estándar Extensiones Multipropósito de Correo de Internet o **MIME** (Multipurpose Internet Mail Extensions), especifica como un programa debe transferir archivos de texto, imagen, audio, vídeo o cualquier archivo que no esté codificado en US-ASCII.



# HOSTS VIRTUALES. CREACIÓN, CONFIGURACIÓN Y UTILIZACIÓN.

Existen tres tipos de virtualhost: **basados en nombre**, **basados en IP** y **basados en varios servidores principales**.

## Virtualhosts basados en nombre:

En la configuración de Apache2 existe un **directorio /etc/apache2/sites-available** donde se definen los virtualhosts, cada virtualhost en un fichero de texto de configuración distinto `empresa1.com.conf` y `empresa2.com.conf`

Ejemplo:

```
<VirtualHost 192.168.200.250>
```

```
    DocumentRoot "/var/www/empresa1"
```

```
    ServerName empresa1.com
```

```
    ServerAlias www.empresa1.com empresa1.es www.empresa1.es
```

```
</VirtualHost>
```

Si no tienes configurado un servidor DNS con las entradas de dominio necesarias, puedes generar estas entradas modificando el archivo **/etc/hosts**, añadiéndolas al final del mismo:

```
#IP nombre-dominio
```

```
192.168.200.250 www.empresa1.com empresa1.com www.empresa1.es empresa1.es
```



# HOSTS VIRTUALES. CREACIÓN, CONFIGURACIÓN Y UTILIZACIÓN.

**Virtualhosts basados en IP:** Directorio `/etc/apache2/sites-available` donde se definen los virtualhosts, cada virtualhost en un fichero de texto de configuración distinto `empresa3.com.conf` y `empresa4.com.conf`

Ejemplo:

```
<VirtualHost IP1_Servidor_Web>
```

```
    DocumentRoot "/var/www/empresa3"
```

```
    ServerName empresa3.com
```

```
    ServerAlias www.empresa3.com empresa3.es www.empresa3.es
```

```
</VirtualHost>
```

Este método no aporta ventajas sobre el anterior, es más, aún puede ser más difícil de mantener si las IP del servidor web se modifican con cierta frecuencia. Requiere servidor con 2 adaptadores de red.

**Virtualhosts basados en varios servidores principales** es el más complejo de todos, solo tiene sentido cuando quieras tener varios archivos de configuración `apache2.conf` independientes organizando cada uno sus propios hosts virtuales.



# MÓDULOS

La importancia de un servidor web radica en su: estabilidad, disponibilidad y escalabilidad. Es muy importante poder dotar al servidor web de nuevas funcionalidades de forma sencilla, así como del mismo modo quitárselas. Es por esto que la posibilidad que nos otorga el servidor web Apache mediante sus módulos sea uno de los servidores web más manejables y potentes que existen. Módulos SSL, PHP, LDAP, etc.

- Comando **apt-get install**: Instalar un módulo.
- Comando **apt-get remove**: Desinstalar un módulo.
- Comando **a2enmod**: Utilizado para habilitar un módulo de apache.
- Comando **a2dismod**: Utilizado para deshabilitar un módulo de Apache.

Si no dispones de **a2enmod** y **a2dismod** para poder habilitar y deshabilitar módulos Apache simplemente haces lo que ellos: crear los enlaces simbólicos correspondientes desde `/etc/apache2/mods-enabled/` hasta `/etc/apache2/mods-available/`.

Los módulos de Apache puedes instalarlos, desinstalarlos, habilitarlos o deshabilitarlos, así, puedes tener un módulo instalado pero no habilitado.

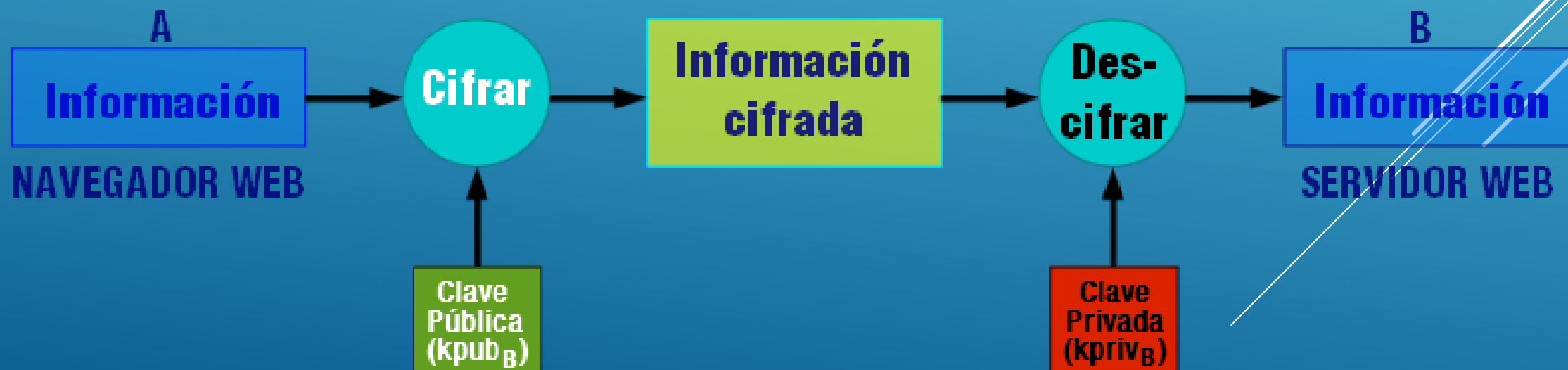
Una vez habilitado o deshabilitado los módulos Apache sólo reconocerá estos cambios cuando recargues su configuración, **/etc/init.d/apache2 restart**



# ACCESO A CARPETAS SEGURAS.

El cifrado al que nos referimos es el cifrado de clave pública o asimétrico: clave pública (**kpub**) y clave privada (**kpriv**). La **kpub** interesa publicarla para que llegue a ser conocida por cualquiera, la **kpriv** no interesa que nadie la posea, solo el propietario de la misma. Ambas son necesarias para que la comunicación sea posible, una sin la otra no tiene sentido, así una información cifrada mediante la **kpub** solamente puede ser descifrada mediante la **kpriv** y una información cifrada mediante la **kpriv** solo puede ser descifrada mediante la **kpub**.

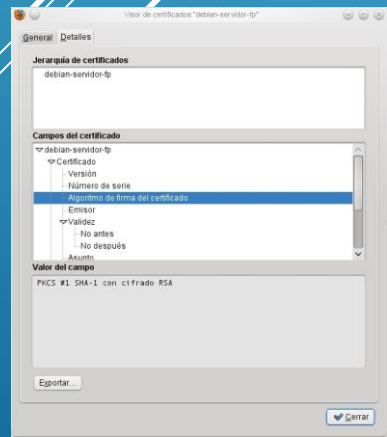
**Certificado** debe ser emitido por un entidad certificadora de confianza.





# CERTIFICADOS DIGITALES Y AC.

- Un **certificado digital** es un documento electrónico que asocia una clave pública con la identidad de su propietario, individuo o máquina, por ejemplo un servidor web, y es emitido por autoridades en las que pueden confiar los usuarios.
- La **Tercera Parte Confiable (TPC)** que se encarga de la firma digital de los certificados de los usuarios de un entorno de clave pública se conoce con el nombre de **Autoridad de Certificación (AC)**.
- Existen varios formatos para certificados digitales, pero los más comúnmente empleados se rigen por el estándar **UIT-T X.509** que contiene los siguientes campos: versión, nº de serie del certificado, identificador del algoritmo de firmado, nombre del emisor, periodo de validez, nombre del sujeto, información de clave pública del sujeto, identificador único del emisor, identificador único del sujeto y extensiones.
- El módulo **ssl** es quien permite cifrar la información entre navegador y servidor web. En la instalación por defecto éste módulo no viene activado, así que debes ejecutar el siguiente comando para poder activarlo: **a2enmod ssl**
- Utilizar el Apartado **4.3.2** para Crear un servidor virtual seguro en Apache (III)



# AUTENTICACIÓN Y CONTROL DE ACCESO.

Puede que interese impedir el acceso a determinadas páginas ofrecidas por el servidor web y suelen emplearse bases de datos SQL o LDAP para la autenticación de usuarios, siendo **OpenLDAP** una de las alternativas más empleadas.

HTTP proporciona un método de autenticación básico de usuarios: **basic**.

Para el buen funcionamiento de lo expuesto a continuación se asume que tanto Apache2 como OpenLDAP están instalados y configurados:

1. Habilita el soporte LDAP para Apache2.
2. Configura el virtualhost **autenticacion-ldap-apache**. La directiva **AllowOverride All** es necesaria para habilitar ficheros **.htaccess**
3. Crea el fichero `/var/www/autenticacion-ldap/.htaccess` que permite configurar la autenticación ldap para el virtualhost anterior. La directiva **Require ldap-user admin** permite la autenticación al usuario **user1LDAP**, todos los demás usuarios tienen el acceso denegado.
4. Accede a la URL: `www.autenticacion-ldap.empresa-proyecto.com`





# MONITORIZACIÓN DEL ACCESO: ARCHIVOS DE REGISTRO (LOGS).

- Es necesario guardar los archivos de registro al menos durante **1 año** según la Ley de Servicios de la Sociedad de Información (**LSSI/CE**).
- Esta información es guardada en formato **CLF** (Common Logon Format) por defecto. Cada línea identifica una solicitud al servidor web y contiene varios campos separados con espacios.
- Como los archivos de registro a medida que pasa el tiempo van incrementando su tamaño, debe existir una política de mantenimiento de registros para que éstos no consuman demasiados recursos en el servidor, así es conveniente **rotar** los archivos de registro, esto es, hay que depurarlos, comprimirlos y guardarlos. Dos opciones para rotar tus registros: **rotatelogs** y **logrotate**.



# DESPLIEGUE DE APLICACIONES SOBRE SERVIDORES WEB.

El servidor web puede tener soporte **php**, pero el soporte **sql** debe ser ofrecido por otro servidor al que pueda acceder el servidor web. Este servidor con soporte sql puede estar configurado en el mismo equipo que el servidor web o en otro.

El procedimiento suele ser el siguiente:

1. Se descarga la aplicación.
2. Se configura para que sea visible a través del servidor web.
3. Suele traer una página de instalación que verifique si el servidor web cumple los requisitos para la instalación de la aplicación.
4. Es necesaria antes de finalizar el proceso de instalación autenticarse al servidor sql con un usuario con permisos para crear/modificar una base de datos. Puede que previamente se tenga que crear la base de datos para que el proceso de instalación genere las tablas necesarias en la misma.
5. Se pide un usuario y contraseña para poder acceder a la aplicación web.
6. Fin de la instalación.

Existen varias aplicaciones de software libre en el mercado para tiendas virtuales, como: **OpenCart**, **Magento**, **osCommerce**.

