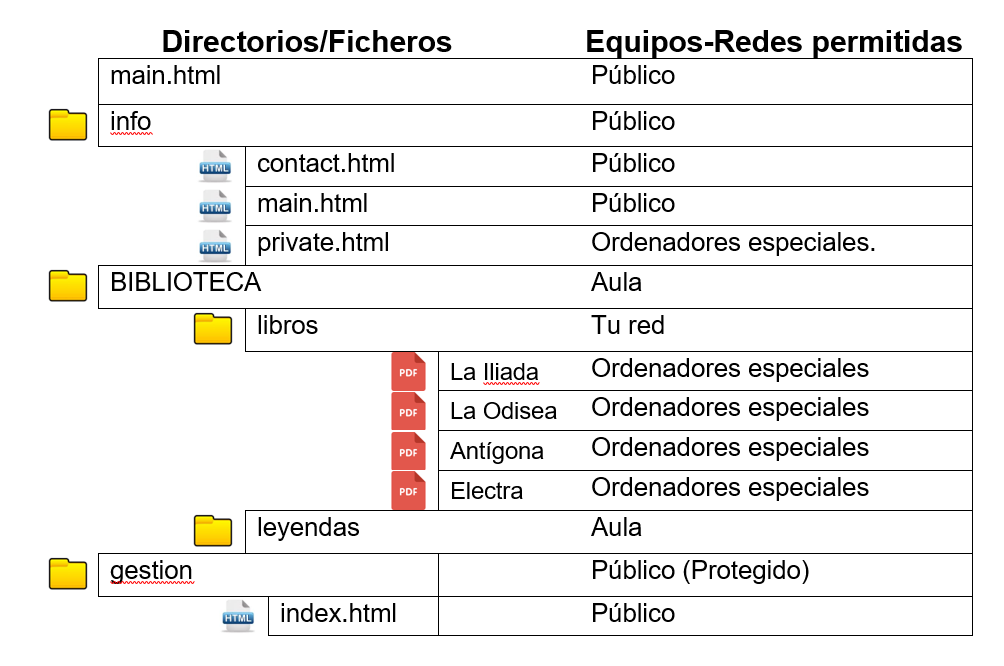
Para la realización de este reto, buscamos instalar y configurar un servidor web apache en el equipo de nuestra red interna llamado “ORÁCULO”. Para ello instalaremos sobre esta máquina (con Debian 9 como sistema operativo) el software de apache2 y realizaremos las configuraciones pertinentes para conseguir los puntos solicitados.

El contenido almacenado en nuestro servidor web consistirá de la siguiente estructura de carpetas con los permisos/método de acceso especificado.



Siendo los grupos de permisos los siguientes:

**Público**: Puede entrar desde cualquier lugar.

**Aula**: Pueden entrar todos los alumnos de las redes xxxx.olimpo.god, es decir, todos los ordenadores de las redes 10.130.x.0/24.

**Tu red**: Todos los ordenadores de la red 10.130.6.0/24 (siendo x el valor de tu red).

Además de la estructura de carpetas mostrada se busca desplegar una aplicación PHP con acceso a base de datos mysql y configurar el acceso a sitios web personales para algunos usuarios, explicadas ambas en sus secciones correspondientes.

INSTALACIÓN:

Antes de nada, tendremos que instalar el software necesario en nuestra máquina Oráculo:

# apt-get install apache2

Habiendo instalado apache, podemos proceder a configurar los puertos en los que escuchara nuestro servidor, que en este caso serán los puertos por defecto de HTTP (80) y HTTPS (443).

/etc/apache2/servidor.local.conf

NameVirtualHost \*:80

Listen 80

NameVirtualHost \*:443

Listen 443

CONFIGURACIÓN DE SITIOS Y PERMISOS

Habiendo instalado el software podemos proceder a la configuración del mismo; para ello nos moveremos al directorio “/etc/apache2/sites-available” y crearemos el que será nuestro archivo de configuración principal con el siguiente contenido.

/etc/apache2/sites-available/servidor.local.conf

<VirtualHost \*:80>

ServerName www.atenea.olimpo.god

DocumentRoot /var/www/servidor.local

ServerAdmin webmaster@servidor.local

DirectoryIndex main.html

</VirtualHost>

<Directory /var/www/servidor.local/gestion>

Options Indexes FollowSymLinks Multiviews

AllowOverride AuthConfig

Require all granted

</Directory>

<Directory /var/www/servidor.local/BIBLIOTECA/">

Options All

AllowOverride All

<RequireAll>

Require ip 10.130.0.0/24

Require ip 10.130.1.0/24

Require ip 10.130.2.0/24

Require ip 10.130.3.0/24

Require ip 10.130.4.0/24

Require ip 10.130.5.0/24

Require ip 10.130.6.0/24

</RequireAll>

</Directory>

Alias /leyendas/ "/var/www/servidor.local/BIBLIOTECA/leyendas/"

<Directory /var/www/servidor.local/BIBLIOTECA/leyendas/">

Options Indexes Multiviews

AllowOverride None

<RequireAll>

Require ip 10.130.0.0/24

Require ip 10.130.1.0/24

Require ip 10.130.2.0/24

Require ip 10.130.3.0/24

Require ip 10.130.4.0/24

Require ip 10.130.5.0/24

Require ip 10.130.6.0/24

</RequireAll>

</Directory>

<Files /var/www/servidor.local/info/private.html>

Options All

AllowOverride All

<RequireAll>

Require ip 10.130.6.0/24

</RequireAll>

</Files>

<Directory /var/www/servidor.local/BIBLIOTECA/libros/">

Options All

AllowOverride All

<RequireAll>

Require ip 10.130.6.0/24

</RequireAll>

</Directory>

Alias /Sofocles01 "/var/www/servidor.local/BIBLIOTECA/libros/Antigona.pdf"

<Files /var/www/servidor.local/BIBLIOTECA/libros/Antigona.pdf>

Options All

AllowOverride All

<RequireAll>

Require ip 10.130.6.0/24

</RequireAll>

</Files>

Alias /Sofocles02 "/var/www/servidor.local/BIBLIOTECA/libros/Electra.pdf"

<Files /var/www/servidor.local/BIBLIOTECA/libros/Electra.pdf>

Options All

AllowOverride All

<RequireAll>

Require ip 10.130.6.0/24

</RequireAll>

</Files>

Alias /Homero01 "/var/www/servidor.local/BIBLIOTECA/libros/Homero-Iliada.pdf"

<Files /var/www/servidor.local/BIBLIOTECA/libros/Homero-Iliada.pdf>

Options All

AllowOverride All

<RequireAll>

Require ip 10.130.6.0/24

</RequireAll>

</Files>

Alias /Homero02 "/var/www/servidor.local/BIBLIOTECA/libros/Homero-Odisea.pdf"

<Files /var/www/servidor.local/BIBLIOTECA/libros/Homero-Odisea.pdf>

Options All

AllowOverride All

<RequireAll>

Require ip 10.130.6.0/24

</RequireAll>

</Files>

Como podemos observar, es en este archivo es donde se realiza la configuración de la estructura de carpetas y sus permisos, además de algún alias si fuese necesario.

Sin entrar en muchos detalles; en este archivo configuraremos un host virtual mediante la etiqueta “<VirtualHost \*:80>” y en dicha etiqueta especificaremos que este virtual host será utilizado por todas las peticiones que accedan mediante el puerto 80 (puerto por defecto de HTTP). En esta sección es importante configurar sobre todo “DocumentRoot”, es decir la carpeta que almacenará todos los recursos a mostrar; y “DirectoryIndex”, mediante la que configuraremos el archivo que tratará de cargar por defecto, en este caso “main.html”.

Tras esta configuración genérica podemos pasar a configurar cada directorio/archivo de manera individual para especificar las propiedades distintas que tendrá cada recurso.

Esta configuración se hará mediante las etiquetas <Directory “Ruta/de/directorio”> y <File “Ruta/de/archivo”>, y para realizar la configuración de permisos utilizaremos la siguiente etiqueta para cada archivo/directorio en el que queramos añadir una restricción de acceso.

<RequireAll>

Require ip 10.130.6.0/24

</RequireAll>

En nuestro caso haremos estas restricciones basadas en IP, permitiendo acceso a individuos que provengan de alguna red en concreto, pero mediante está directiva es posible configurar el acceso basado en nombres también mediante instrucciones como “Require host nombre\_Host”.

Además en este archivo hemos configurado algunos alias para acceder a un recurso en concreto mediante un nombre distinto.

RESTRICCIÓN DE ACCESO POR CONTRASEÑA

Para sobrescribir el funcionamiento por defecto y configurar el acceso a cierto recurso por usuario y contraseña, dentro del directorio donde queremos restringir el acceso tendremos que cambiar a AllowOverride AuthConfig

<Directory /var/www/servidor.local/gestion>

Options Indexes FollowSymLinks Multiviews

AllowOverride AuthConfig

Require all granted

</Directory>

A continuación crearemos dos ficheros más dentro del directorio gestión. El primero se llamará “.htaccess” y su contenido será el siguiente:

/etc/apache2/servidor.local.conf/gestión/.htaccess

AuthType Basic

AuthName "Identifícate mamarracho"

AuthUserFile /var/www/servidor.local/gestion/.password

Require valid-user

El segundo, “.password”, es el archivo que hemos especificado como AuthUserFile en el apartado anterior, y en el introduciremos la lista de usuarios con sus contraseñas (hasheadas con MD5).

/etc/apache2/servidor.local.conf/gestión/.password

sibila:$apr1$1yciwtkl$qtyNuU9eOAZ7ThjmupM7M0

melisa:$apr1$2kb1hcxu$3kBJWtYuPfvvG6qwov.zH/

casandra:$apr1$mjs0de79$G/Kfsfy77jb1etwoaxb3t.

aquiles:$apr1$9dfrjo3m$eqo3OZZx6UgFlFH5tm9lN1

agamenon:$apr1$2i6z3ujc$XVQ38aNuWY0NltbIyLLj.1

ayax:$apr1$vgxdyftw$nPpWn50T9sZssn3BWIVIe0

Podemos obtener el hash de múltiples maneras, ya sea con herramientas de Linux como md5pass o con herramientas online como la [siguiente](https://www.web2generators.com/apache-tools/htpasswd-generator).

CONFIGURACIÓN DE VIRTUALHOST CON HTTPS:

Si queremos usar conexiones seguras mediante el protocolo HTTPS para acceso a determinadas páginas, tendremos que configurar Apache para que cargue el módulo ssl, aunque en algunas versiones de Apache 2 suele venir integrado.

# a2enmod ssl

Para utilizar el protocol seguro, es obligatorio tener un certificado digital a nuestro nombre, para ello necesitaremos instalar openSSL y crearnos un certificado, para ello:

# apt-get install openssl

# make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/certificados/subastas.local.pem openssl req $@ -new -x509 -days 365 -nodes -out

Configuraremos con protocolo seguro el acceso a la página de subastas, y para su configuración crearemos un nuevo archivo en sites-available, especificando allí el certificado que utilizará.

/etc/apache2/sites-available/subastas.local.conf

<VirtualHost \*:443>

ServerName subastas.atenea.olimpo.god

DocumentRoot /var/www/subastas.local

ServerAdmin webmaster@subastas.local

DirectoryIndex index.php

SSLEngine on

SSLCertificateFile /etc/apache2/certificados/subastas.local.pem

</VirtualHost>

APLICACIÓN PHP/MYSQL

Esta aplicación necesita para funcionar correctamente PHP y una base de datos MySQL/MariaDB, por lo que para que nuestro servidor pueda lanzar correctamente la app necesitaremos instalar algunos programas.

Primero necesitamos instalar la base de datos, para ello

# wget https://dev.mysql.com/get/mysql-apt-config\_0.8.10-1\_all.deb

# dpkg –i mysql-apt-config\_0.8.10-1\_all.deb

# apt update

# apt install mysql-server

Ahora podemos configurar la base de datos con el script que la aplicación trae por defecto

# mysql\_secure\_installation

Siguiendo los pasos que indica el script tendremos la base de datos lista y configurada con el usuario “root” con la contraseña que hayamos especificado, y ya podremos habilitar el servicio para comenzar a utilizarlo.

# systemctl start mysql

Para la administración de la base de datos instalaremos phpMyAdmin; para ello:

# wget https://files.phpmyadmin.net/phpMyAdmin/5.0.4/phpMyAdmin-5.0.4-all-languages.tar.xz

# tar xf phpMyAdmin-5.0.4-all-languages.tar.xz -C /var/www/subastas.local/

# mv /var/www/subastas.local/phpMyAdmin-5.0.4-all-languages/ /var/www/subastas.local/phpmyadmin

# chown www-data /var/www/subastas.local/phpmyadmin/

Finalmente tendremos que instalar un intérprete de PHP, para ello:

# apt-get install php libapache2-mod-php php-mysql

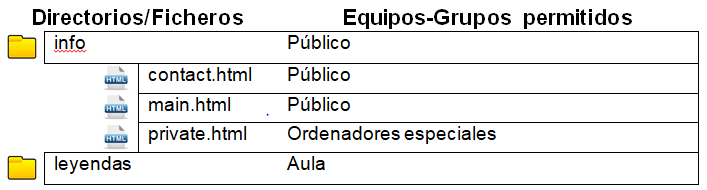
# sudo a2enmod php7

Hecho esto ya tendremos todo el software necesario instalado y podemos proceder a desplegar nuestra aplicación en el servidor apache (subiendo el código e importando la base de datos).

SITIOS WEB PERSONALES

Finalmente, se desea que además del sitio web principal y la web de subastas, se disponga de páginas web personales para los usuarios Sibila, Melisa y Aquiles.

Por comodidad, se quiere poder utilizar el carácter especial ~ para acceder ellas. Todas estas Webs personales tendrán la misma estructura de directorios que se muestra a continuación:



Para obtener este resultado tendremos que habilitar otro módulo de Apache, en este caso userdir.

# a2enmod userdir

Además configuraremos el modulo con los siguientes parámetros:

/etc/apache2/mods-available/userdir.conf

<IfModule mod\_userdir.c>

UserDir public\_html

UserDir disabled root

<Directory /home/\*/public\_html>

AllowOverride FileInfo AuthConfig Limit Indexes

Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec

Require method GET POST OPTIONS

</Directory>

</IfModule>

Y crearemos para los usuarios que hemos especificado anteriormente esta carpeta “public\_html” en sus directorios personales.

# mkdir /home/sibila/public\_html

# mkdir /home/melisa/public\_html

# mkdir /home/aquiles/public\_html

Y crearemos la estructura de archivos en cada carpeta.

La configuración ya está lista, pero todavía no está habilitada, para ello necesitaremos ejecutar el siguiente comando para crear un enlace simbólico en sites-enabled de los archivos almacenados en sites-available, para que de esta manera nuestro servidor apache conozca que esta es la configuración que deberá aplicar.

# a2ensite subastas.local.conf

# a2ensite servidor.local.conf

Finalmente reiniciaremos/recargaremos apache2 para que aplique todos los cambios realizados.

# systemctl restart apache2

Como es habitual, tenemos múltiples comandos para la gestión del servicio, entre los más importantes:

# systemctl stop apache2

# systemctl start apache2

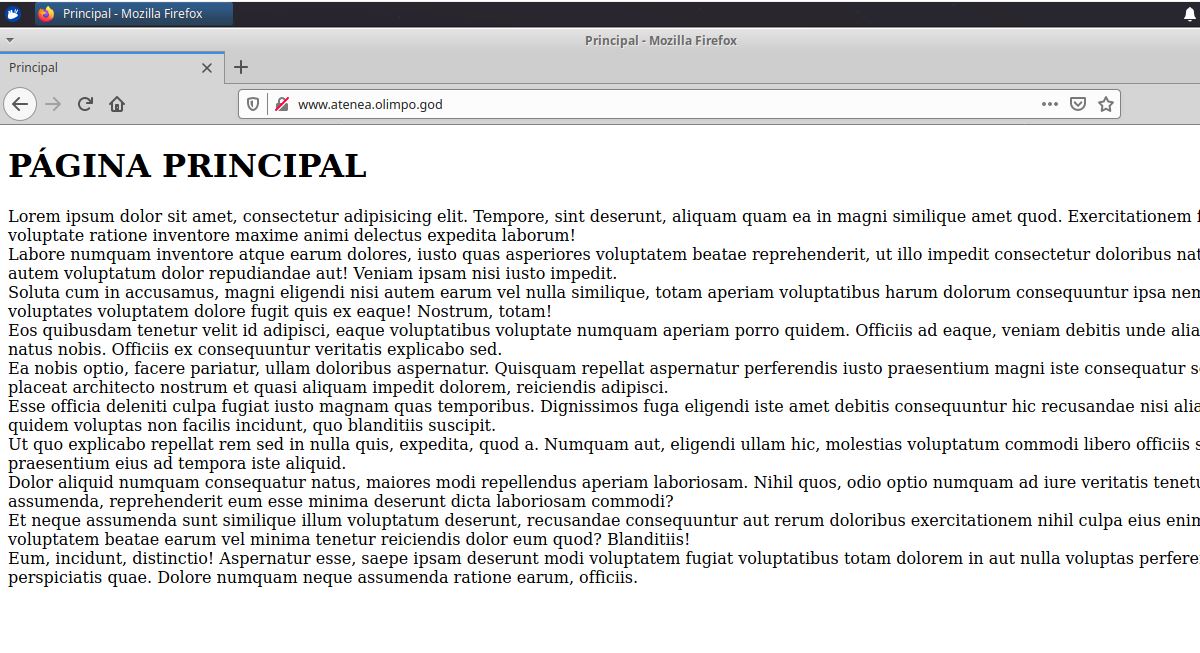
# systemctl enable apache2

# systemctl disable apache2

# systemctl reload apache2

COMPROBACIONES:

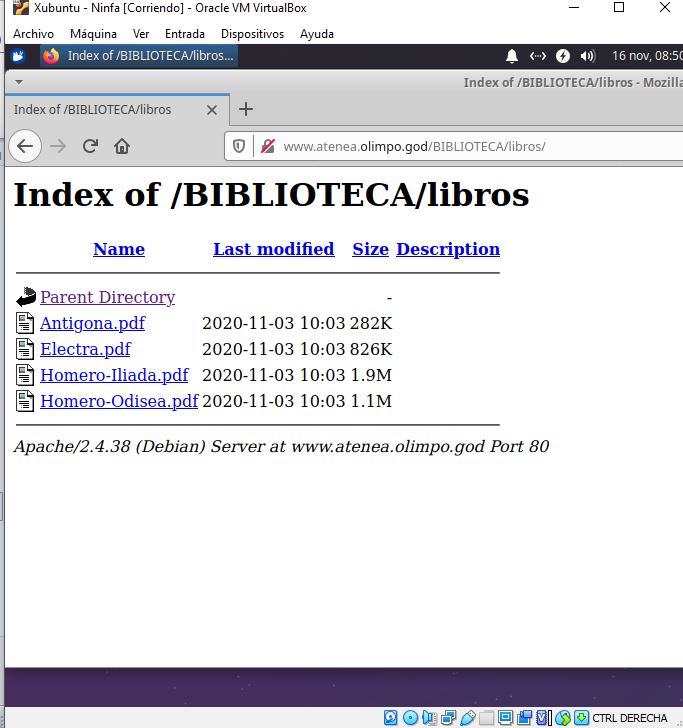
a)



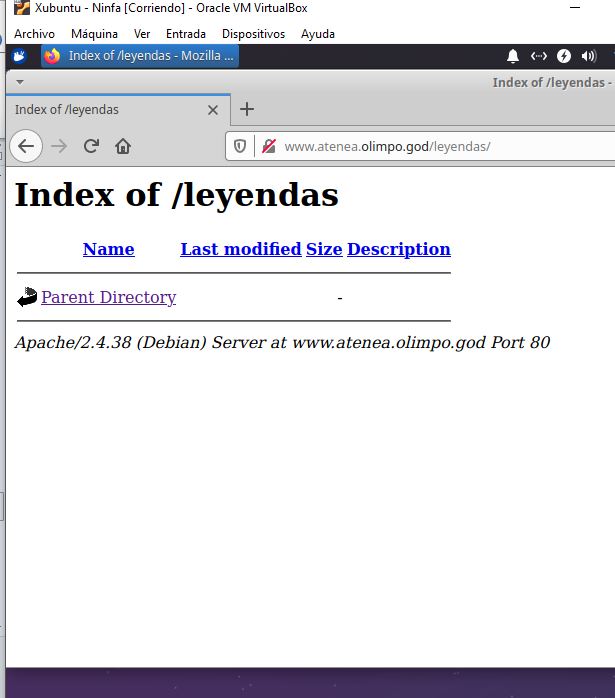
b)



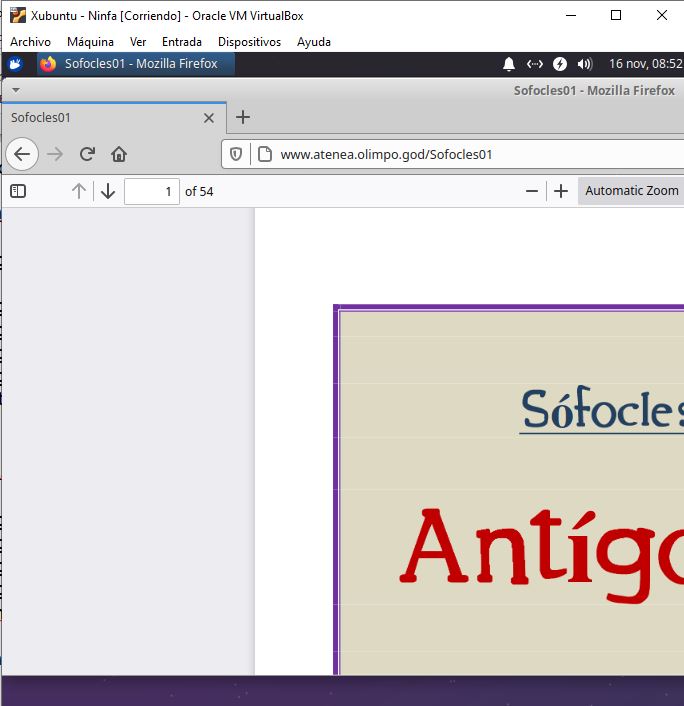
c)



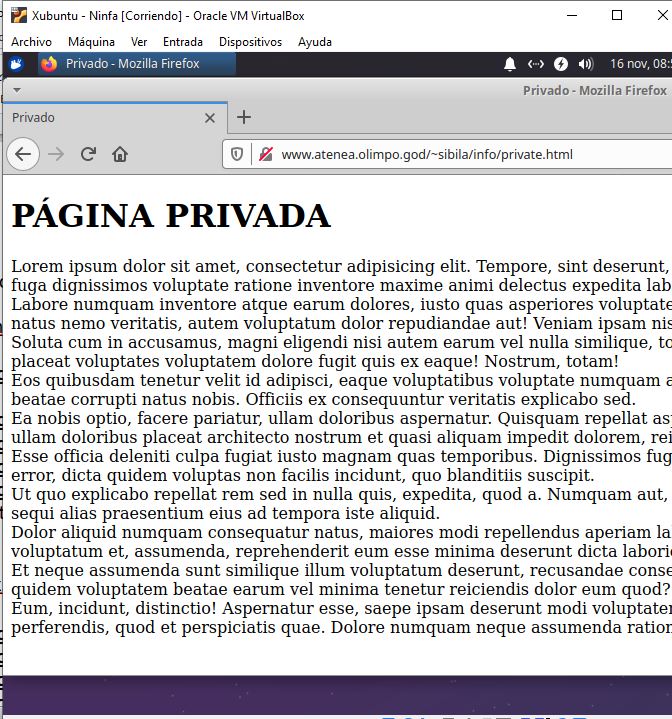
d)



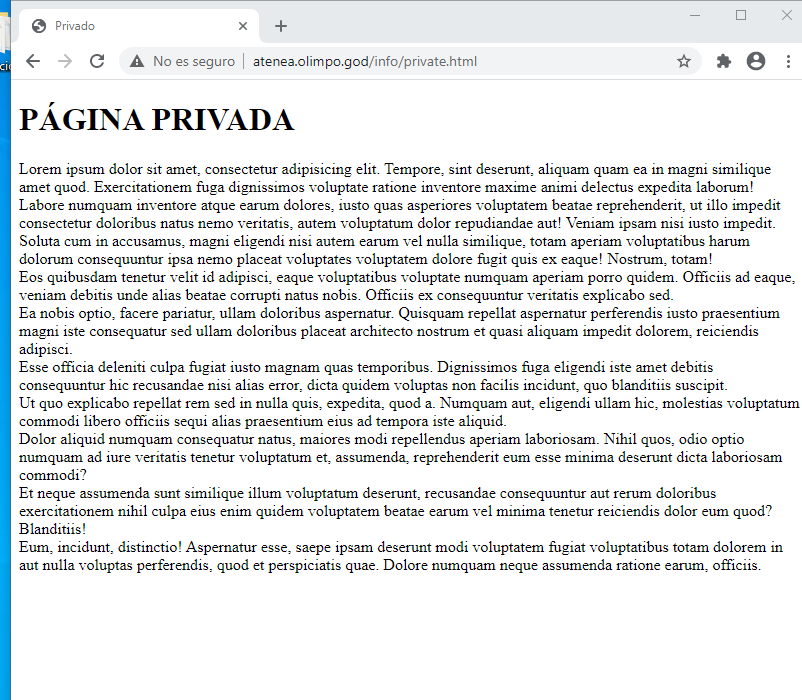
e)



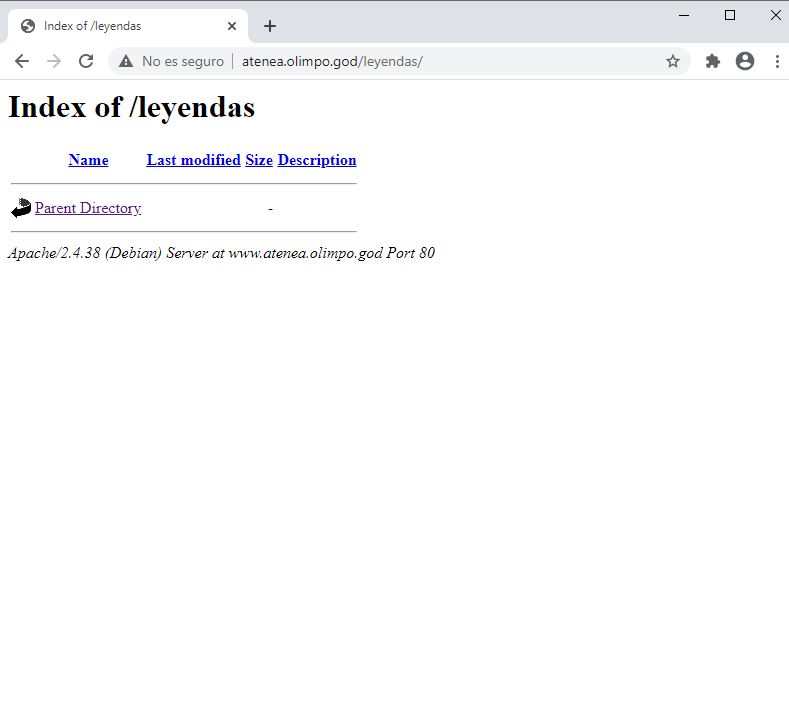
f)



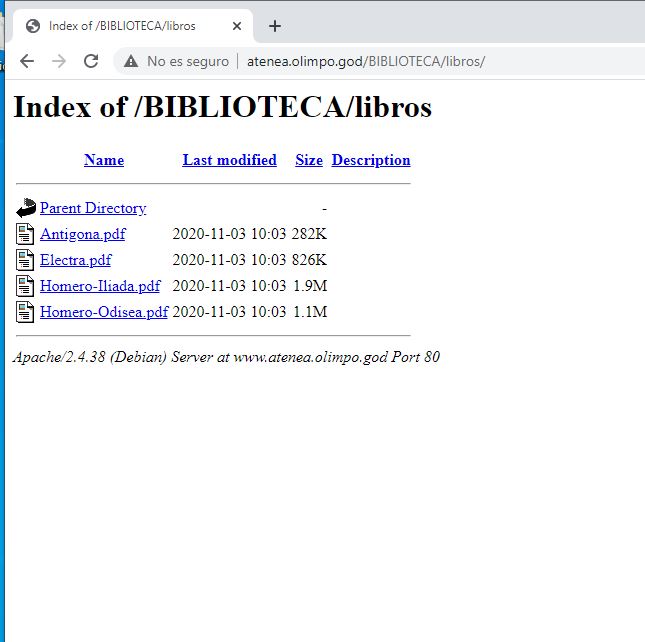
g)



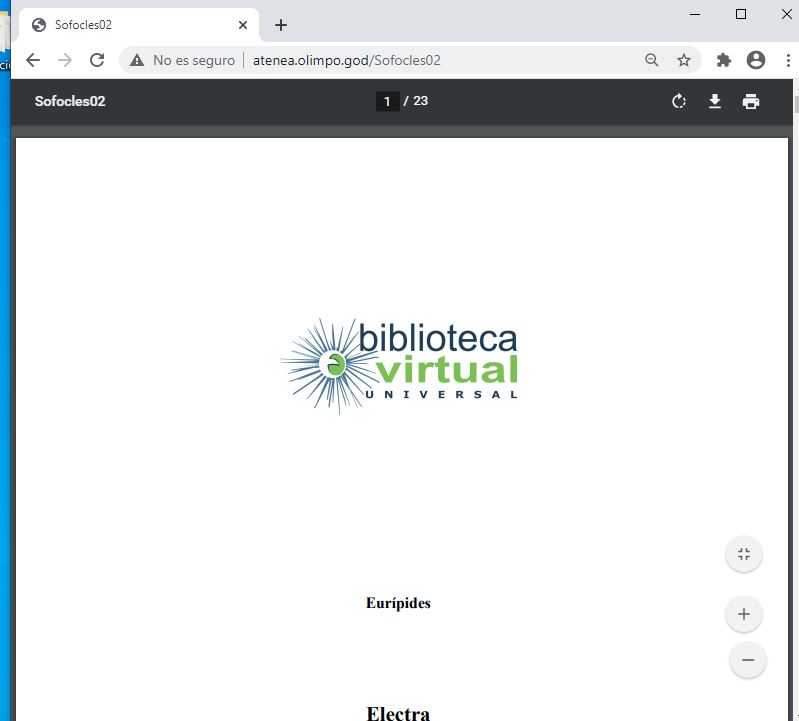
h)



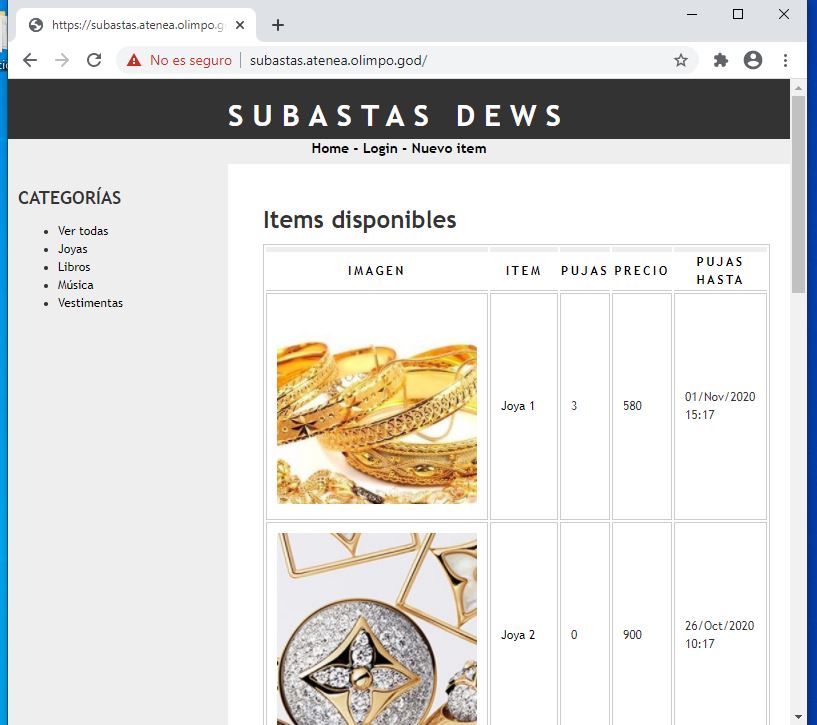
i)



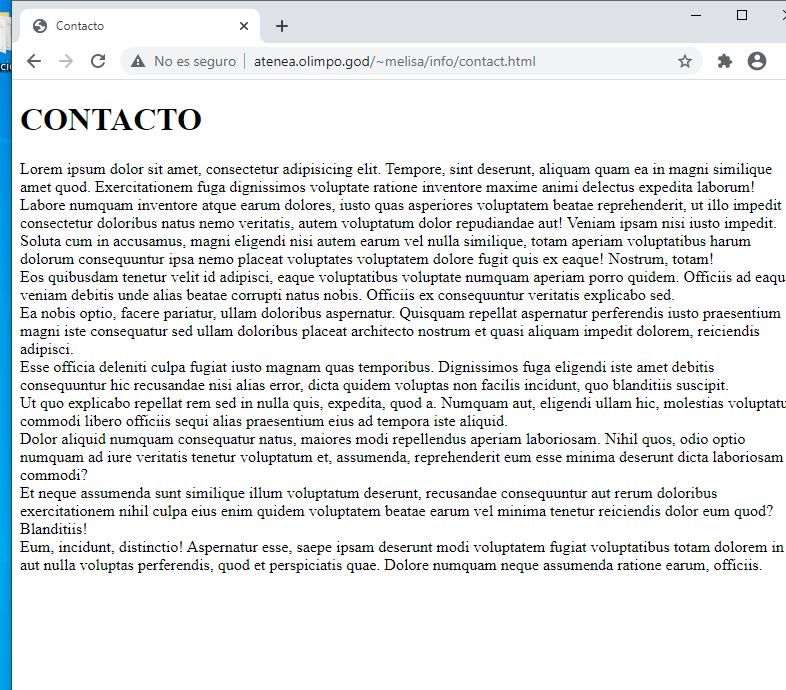
j)



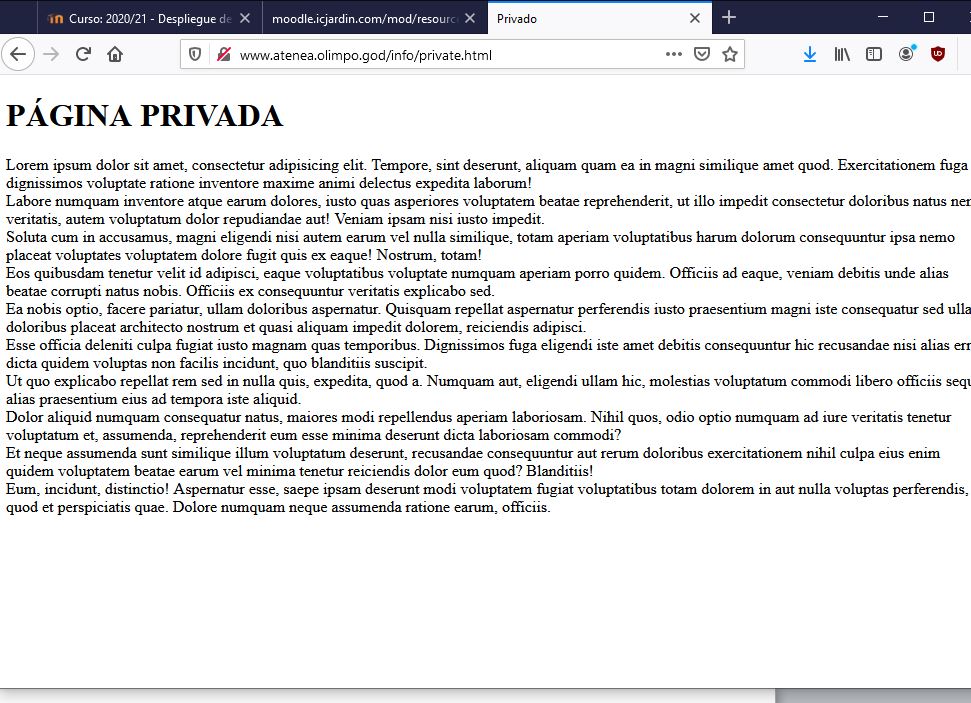
k)



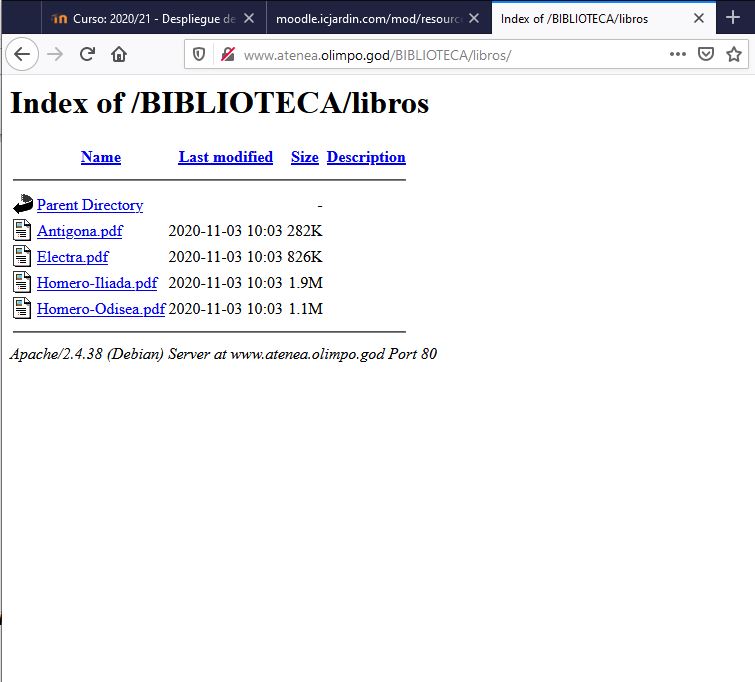
l)



m)



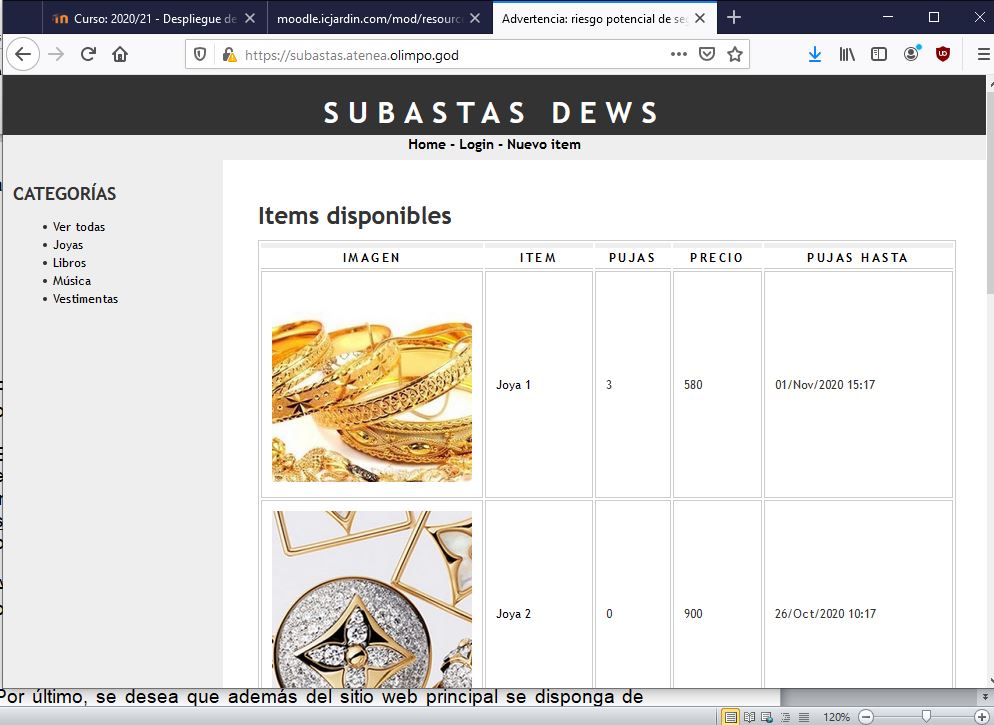
n)



o)



p)



q)

