

Sistema de registro de Producción y Almacén | Oficina de Publicaciones e Impresiones | UNCP

Este sistema ha sido desarrollado con software libre pero el despliegue de la aplicación era necesario realizarlo en Windows xp (que es el requerimiento del cliente y este es el sistema operativo actual con el que se cuenta en la oficina), cabe señalar que la instalación de este software solo se realizará en una máquina (de la jefatura).

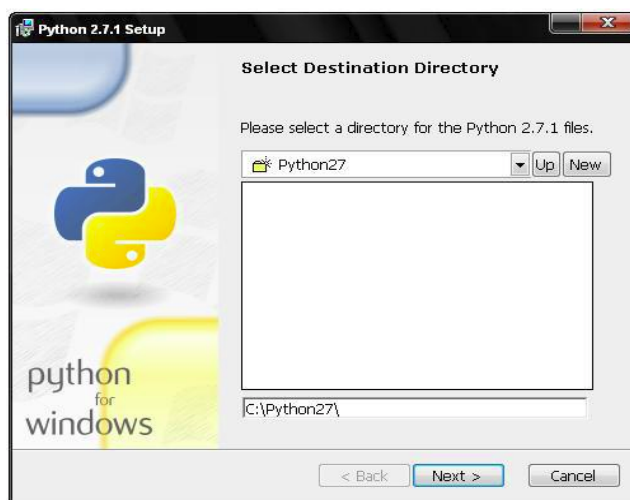
Los primeros pasos a realizar son la instalación de los siguientes programas:

Python

Python es un lenguaje de programación de alto nivel que se caracteriza por ser multifuncional, multiparadigma y es libre (Python Software Foundation License). Para empezar debemos descargarlo desde aquí <http://www.python.org/download/>. Luego al ejecutar el instalador se mostrará la ventana de inicio, seleccionaremos "Install for all users" luego a Next >.



Luego seleccionaremos donde se va a instalar Python en este caso será en el disco D, D:Python27, solo debemos seleccionarlo y darle click en Next >

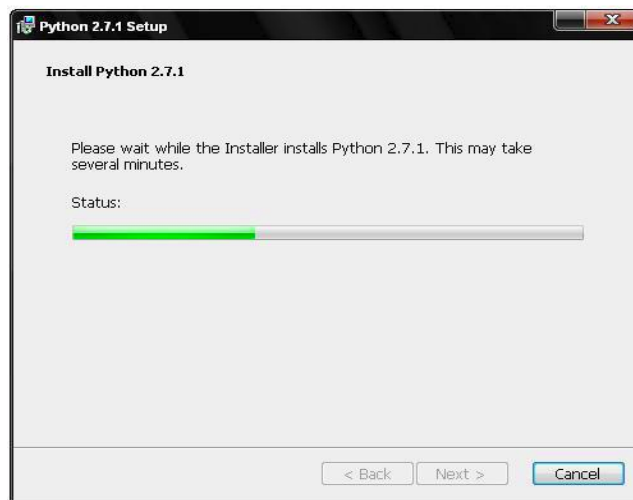


El siguiente paso es seleccionar que características de Python se desea instalar, lo dejaremos como esta por defecto y le damos click en Next > (si se quisiera modificar alguna característica se da click en los iconos que se desprenden del árbol de Python).

Manual de Despliegue



Luego podemos observar como se completa la instalación de Python

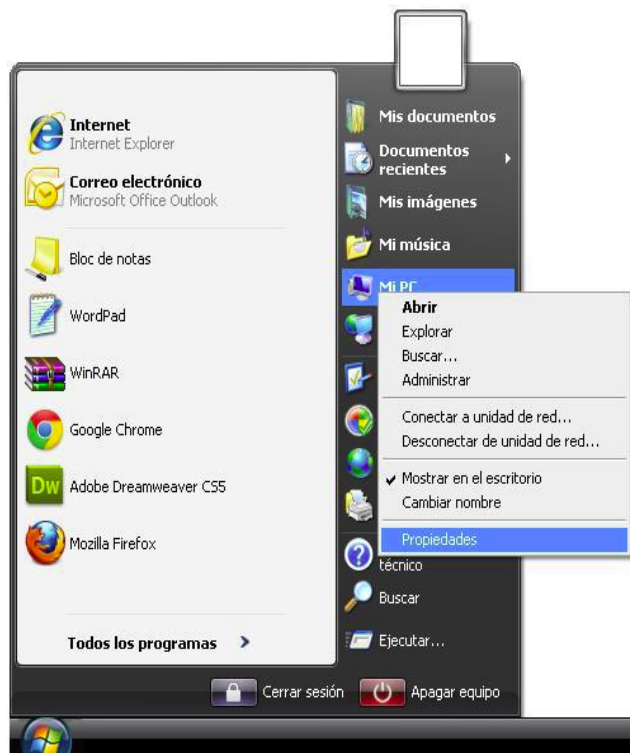


Y en la última ventana podremos ver que la instalación está lista y le damos click en Finish y de esta manera Python 2.7 ya está en la pc, ahora tendremos que seguir unos pasos de configuración para poder ejecutar Python desde el terminal de Windows XP.

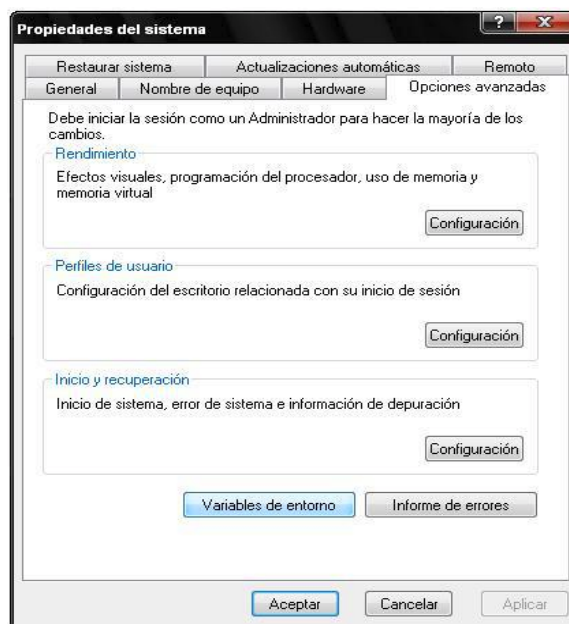


Para la siguiente configuración nos dirigimos a MI PC y damos click derecho para ver las Propiedades,

Manual de Despliegue

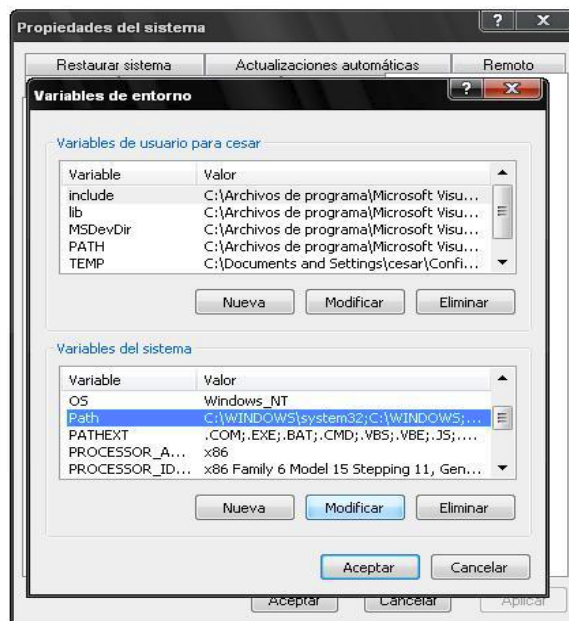


Seleccionaremos en la ventana, la pestaña de Opciones Avanzadas y damos click en Variables de entorno.

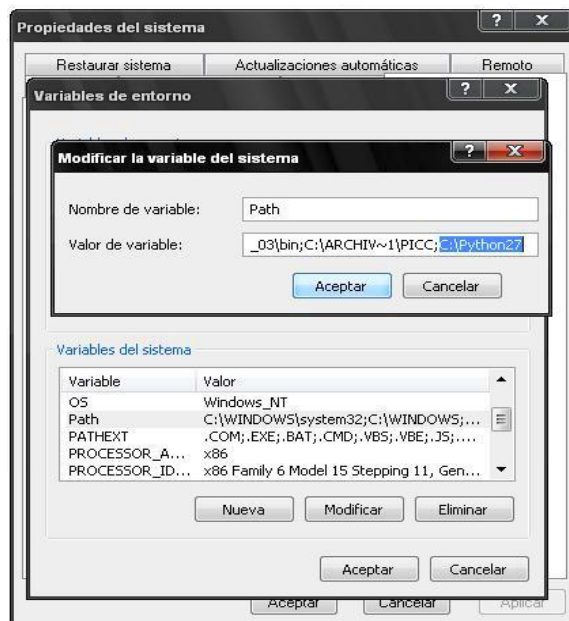


Luego en el área de Variables del Sistema, seleccionamos la variable Path y damos click en modificar,

Manual de Despliegue



Ya en la ventana que se nos muestra agregamos la ruta en la que se encuentra instalado Python, en este caso escribimos D:Python27, finalmente damos click en Aceptar.



Y con esto se termina la configuración de Python para que pueda ser utilizado incluso desde la terminal de Windows XP.

Django

Después de haber instalado Python, tenemos que instalar Django, para esto descargamos esto http://modwsgi.googlecode.com/files/mod_wsgi-win32-ap22py27-3.3.so

Manual de Despliegue

The screenshot shows the Django project documentation page for installation. The page is titled "Install Python" and "Install Apache and mod_wsgi". It provides instructions on how to install Django, including requirements for Python and Apache. A sidebar on the right contains a search bar, a table of contents, and a list of links to other documentation pages.

Install Python

Being a Python Web framework, Django requires Python.

It works with any Python version from 2.5 to 2.7 (due to backwards incompatibilities in Python 3.0, Django does not currently work with Python 3.0; see the [Django FAQ](#) for more information on supported Python versions and the 3.0 transition).

Get Python at <http://www.python.org>. If you're running Linux or Mac OS X, you probably already have it installed.

Django on Jython

If you use [Jython](#) (a Python implementation for the Java platform), you'll need to follow a few additional steps. See [Running Django on Jython](#) for details.

Install Apache and mod_wsgi

If you just want to experiment with Django, skip ahead to the next section; Django includes a lightweight web server you can use for testing, so you won't need to set up Apache until you're ready to deploy Django in production.

If you want to use Django on a production site, use Apache with mod_wsgi. mod_wsgi can operate in one of two modes: an embedded mode and a daemon mode. In embedded mode, mod_wsgi is similar to mod_perl - it embeds Python within Apache and loads Python code into memory when the server starts. Code stays in memory throughout the life of an Apache process, which leads to significant performance gains over other server arrangements. In daemon mode, mod_wsgi spawns an independent daemon process that handles requests. The daemon process can run as a different user than the Web server, possibly leading to improved security, and the daemon process can be restarted without restarting the entire Apache Web server, possibly making refreshing your codebase more seamless. Consult the mod_wsgi documentation to determine which mode is right for your setup. Make sure you have Apache installed, with the mod_wsgi module activated. Django will work with any version of Apache that supports mod_wsgi.

See [How to use Django with mod_wsgi](#) for information on how to configure mod_wsgi once you have it installed.

If you can't use mod_wsgi for some reason, fear not: Django supports many other deployment options. One is [uWSGI](#); it works very well with [nginx](#). Another is [FastCGI](#), perfect for using Django with servers other than Apache. Additionally, Django follows the WSGI spec ([PEP 3333](#)), which allows it to run on a variety of server platforms. See the [server-arrangements wiki page](#) for specific installation instructions for each platform.

Get your database running

If you plan to use Django's database API functionality, you'll need to make sure a database server is running. Django supports many different database servers and is officially supported with [PostgreSQL](#), [MySQL](#), [Oracle](#) and [SQLite](#) (although SQLite doesn't require a separate server to be running).

Search

Contents

- How to install Django
 - Install Python
 - Install Apache and mod_wsgi
 - Get your database running
 - Remove any old versions of Django
 - Install the Django code
 - Installing a distribution-specific package
 - Installing an official release
 - Installing the development version

Browse

- Prev: Using Django
- Next: Models and databases
- Table of contents
- General Index
- Python Module Index

You are here:

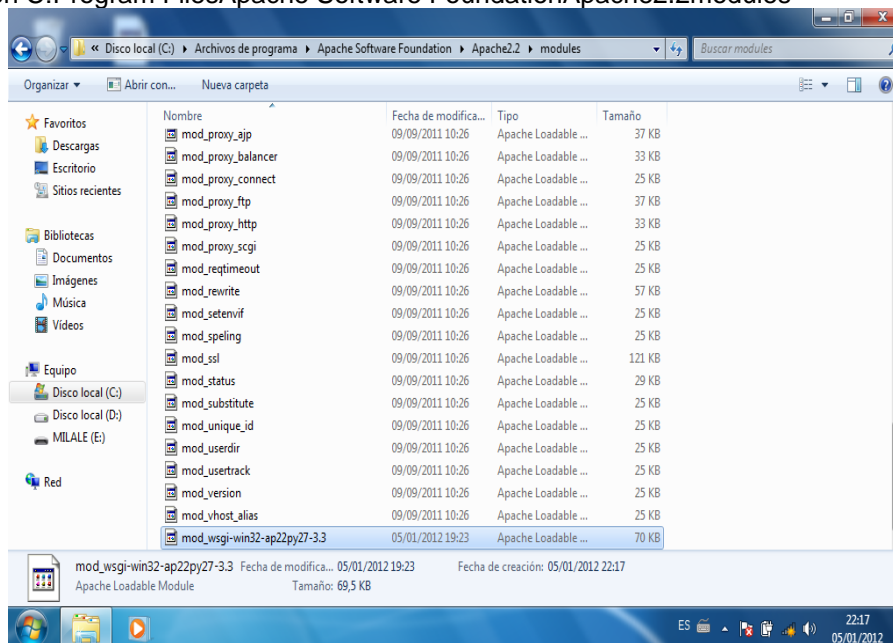
- Django dev documentation
 - Using Django
 - How to install Django

Download:

Offline (development version): [HTML](#) | [PDF](#) | [ePub](#)
Provided by [Read the Docs](#).

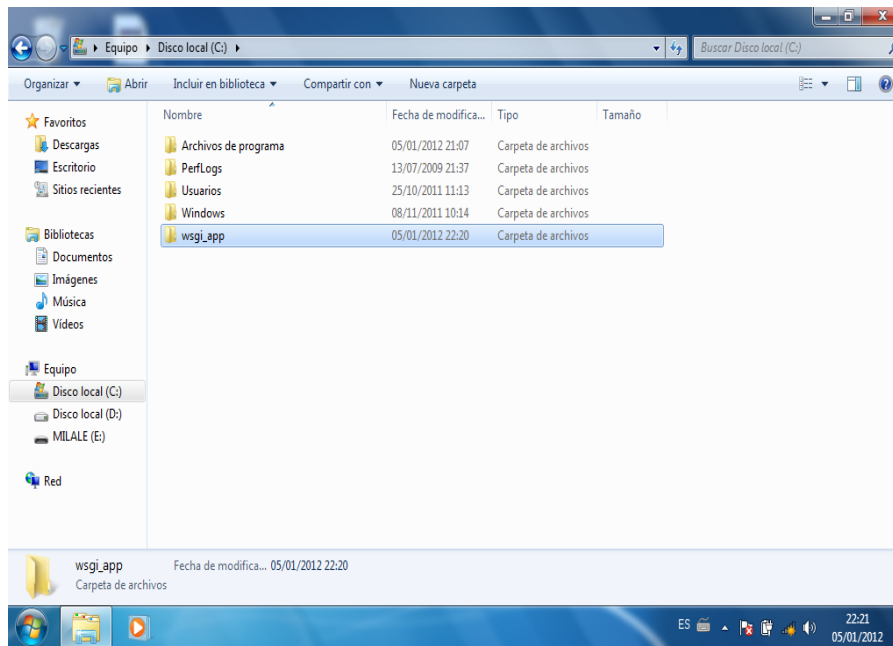
Documentation version: **dev**

luego haremos lo siguiente: este archivo deberemos copiarlo en la ruta donde hemos instalado Apache, en este caso en C:\Program Files\Apache Software Foundation\Apache2.2\modules

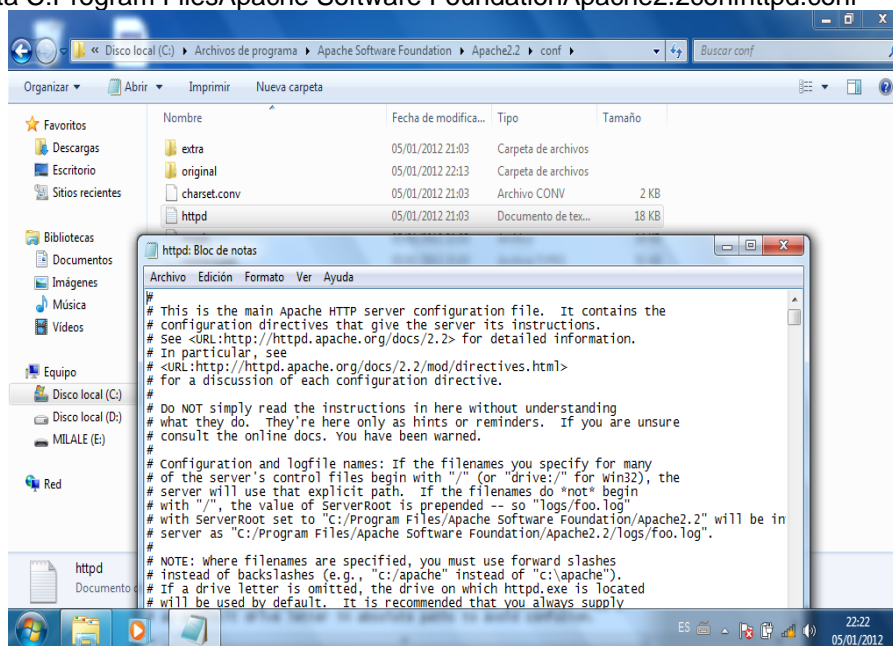


Después de esto debemos crear un directorio llamado wsgi_app fuera de la ruta en la que nos encontramos, por ejemplo en C:\wsgi_app desde la que se administrará el mod_wsgi-application

Manual de Despliegue

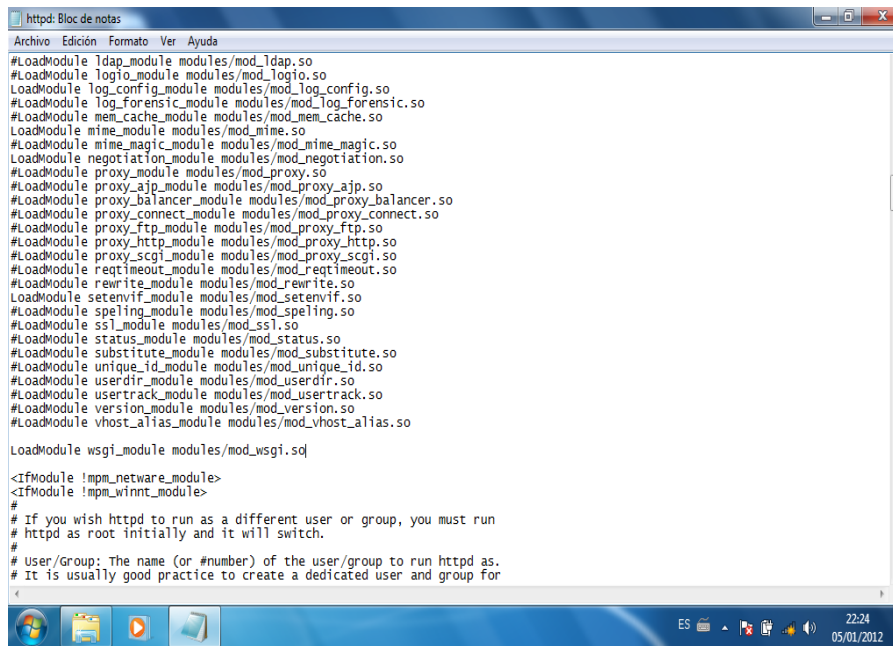


Debemos encontrar el archivo httpd.conf-file y abrirlo con un editor de textos, en mi caso se encuentra en la siguiente ruta C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf



Luego debemos buscar las líneas de "LoadModule" y donde estas terminen, agregar lo siguiente:
LoadModule wsgi_module modules/mod_wsgi.so

Manual de Despliegue



```
#LoadModule ldap_module modules/mod_ldap.so
#LoadModule logio_module modules/mod_logio.so
LoadModule log_config_module modules/mod_log_config.so
#LoadModule log_forensic_module modules/mod_log_forensic.so
#LoadModule mem_cache_module modules/mod_mem_cache.so
LoadModule mime_module modules/mod_mime.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
#LoadModule reqtimeout_module modules/mod_reqtimeout.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule spelling_module modules/mod_spelling.so
#LoadModule ssl_module modules/mod_ssl.so
#LoadModule status_module modules/mod_status.so
#LoadModule substitute_module modules/mod_substitute.so
#LoadModule unique_id_module modules/mod_unique_id.so
#LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so

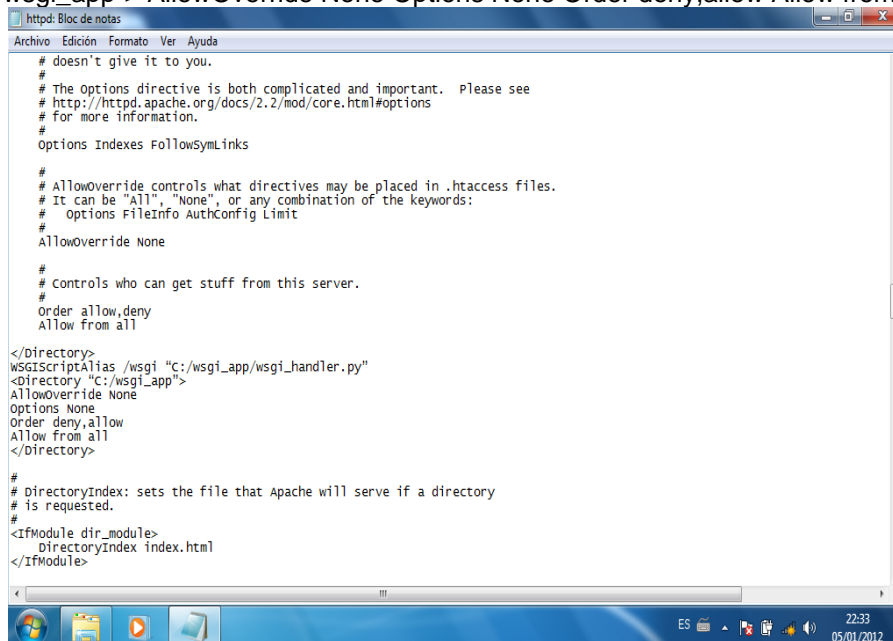
LoadModule wsgi_module modules/mod_wsgi.so

<IfModule !mpm_network_module>
<IfModule !mpm_winnt_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
```

Luego debemos ubicar el bloque <Directory> y escribiremos lo siguiente:

WSGIScriptAlias /wsgi "C:/wsgi_app/wsgi_handler.py"

<Directory "C:/wsgi_app"> AllowOverride None Options None Order deny,allow Allow from all </Directory>



```
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.2/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Order allow,deny
Allow from all

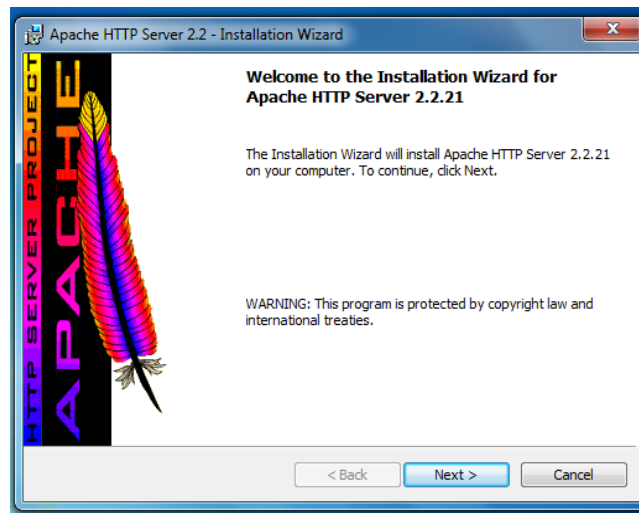
</Directory>
WSGIScriptAlias /wsgi "c:/wsgi_app/wsgi_handler.py"
<Directory "C:/wsgi_app">
AllowOverride None
Options None
Order deny,allow
Allow from all
</Directory>

#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

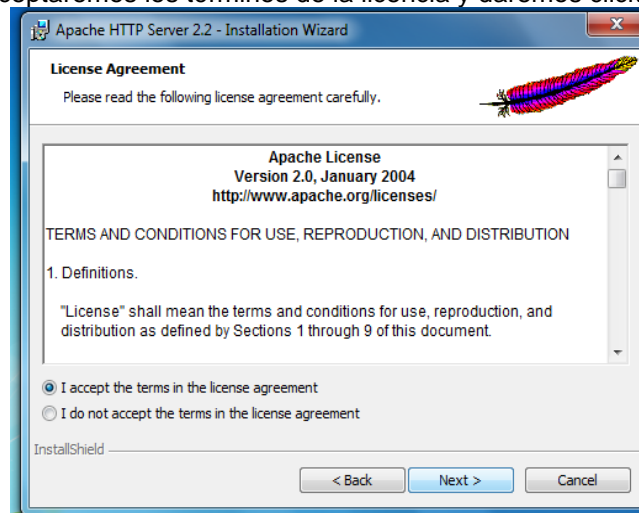
Apache

El servidor HTTP Apache es un servidor web HTTP que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Para empezar con la instalación, ingresaremos a httpd.apache.org/download.cgi y seleccionaremos el archivo llamado Win32 Binary including OpenSSL 0.9.8r (MSI Installer). Después de esto ejecutaremos el archivo obtenido, veremos esta ventana y le daremos click en Next >

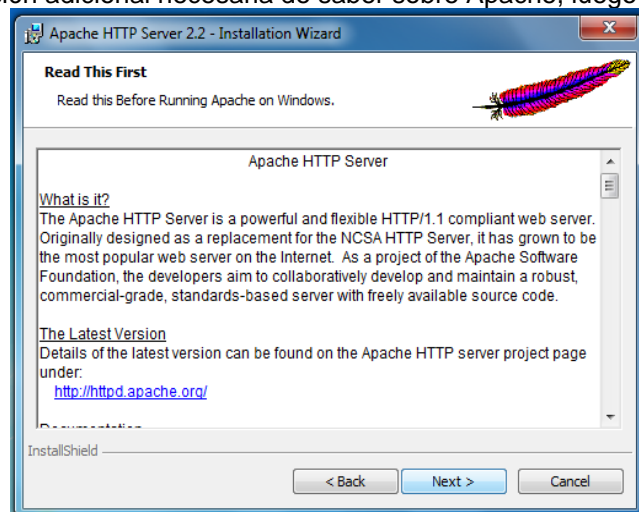
Manual de Despliegue



Luego en esta ventana aceptaremos los términos de la licencia y daremos click en Next >



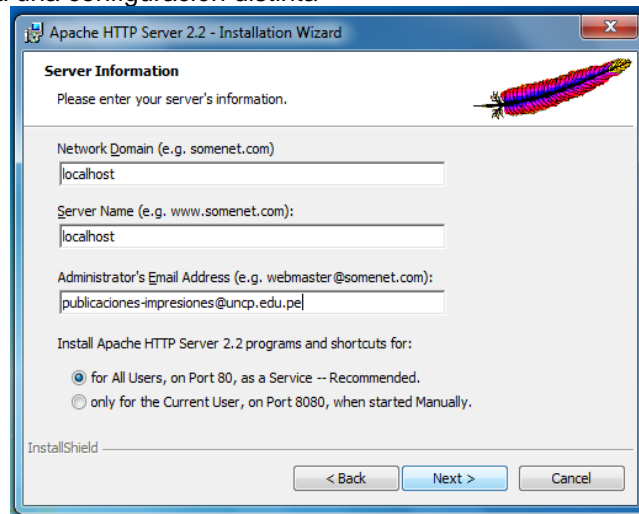
Ahora leeremos información adicional necesaria de saber sobre Apache, luego damos click en Next >



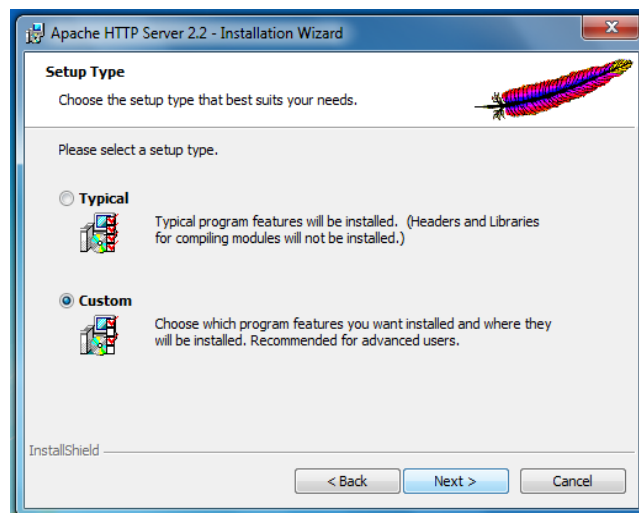
Nos aparecerá la siguiente ventana en la cual deberemos agregar información, en este caso en el campo de Network Domain y de Server Name colocaremos localhost, esto es porque debemos probar en nuestra pc si Apache estará funcionando correctamente y en el email address colocar el mail del administrador, en este caso es el mail de la oficina de Publicaciones e Impresiones, en las opciones

Manual de Despliegue

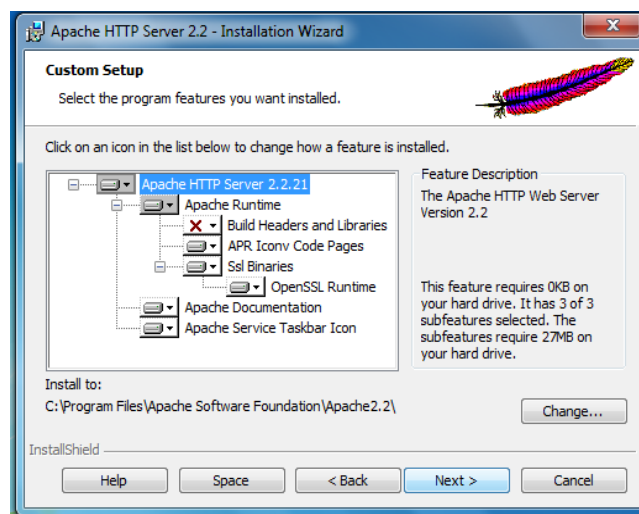
debajo seleccionaremos que Apache se instale para todos los usuarios de la pc en el puerto 80 ya que de lo contrario se necesitaría una configuración distinta



Al dar click en Next > en la siguiente ventana seleccionaremos la instalación personalizada (custom) y damos click a Next >

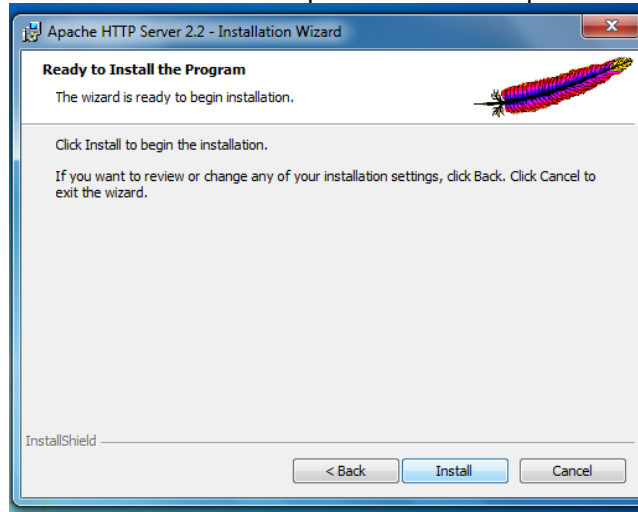


Luego, en la venta siguiente veremos el árbol que nos indica todo lo que se instalará en la pc, damos click en Next >

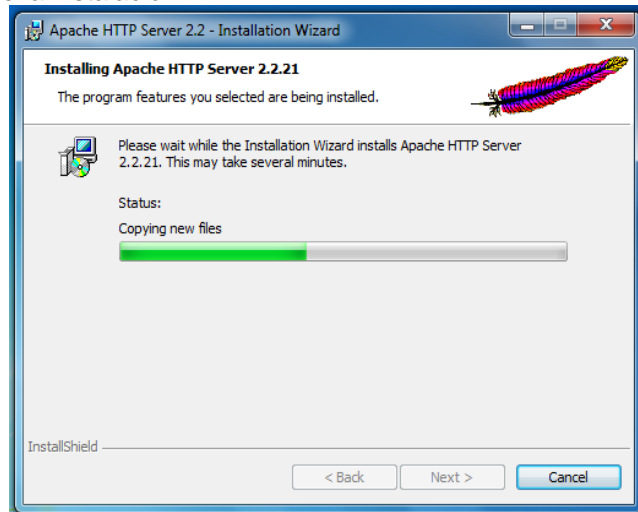


Manual de Despliegue

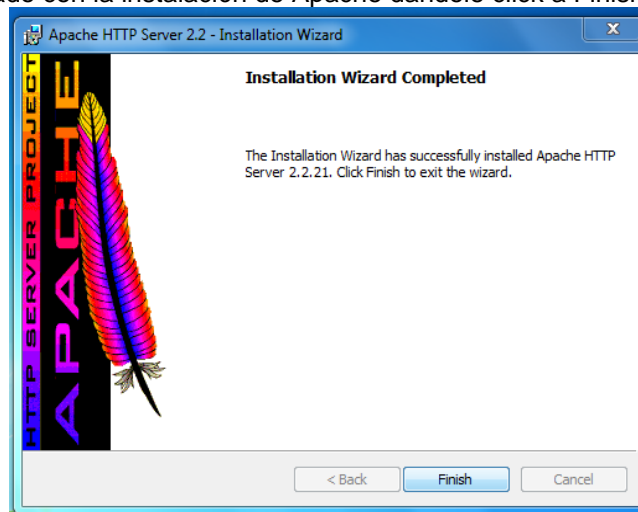
En la siguiente ventana indicaremos al dar Next > que estamos listos para instalar.



Esperamos el progreso de la instalación

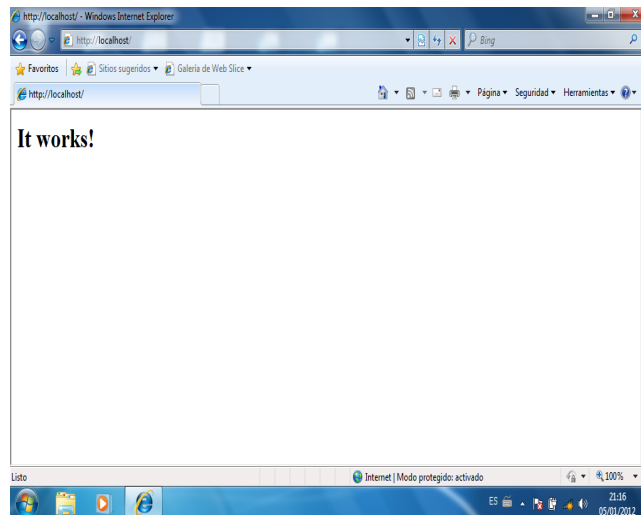


Y listo, habremos terminado con la instalación de Apache dándole click a Finish

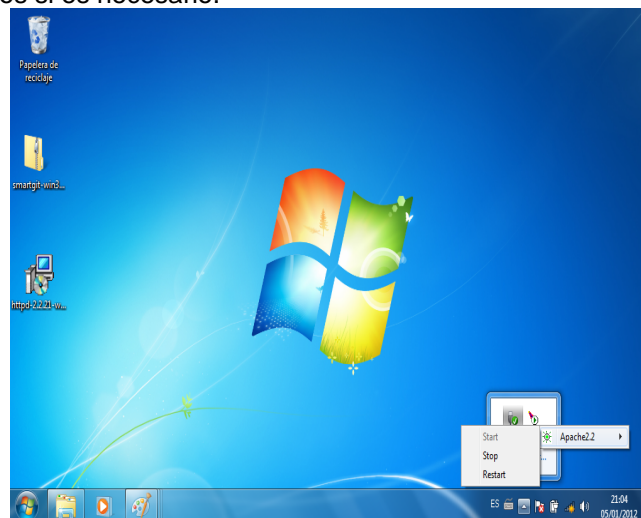


Desde cualquier browser al entrar al localhost podremos ver que el servidor Apache está funcionando

Manual de Despliegue



En el desktop podremos ver como ya Apache se encuentra funcionando y nos muestra las opciones de Stop y Restart para usarlos si es necesario.



SmartGit