

LabWebUQ Versión 1.0

LABORATORIO WEB

UNIQUINDIO

Manual de instalación

1 Instrucciones de instalación

El proceso de ejecución del software se realiza siguiendo una serie de pasos y configuraciones previas. Este manual muestra el proceso de instalación de dependencias en una computadora personal que puede servir como host local para generar una red interna, o instalación en un hosting pago; En cuanto a la instalación en el servidor local, se proporciona el manual para la instalación en los sistemas operativos GNU/Linux y Windows.

Dependencias del sistema:

- python >= 3.7
- pip

Dependencias de Python:

- django >= 2.1
- django-extensions
- opencv-contrib-python
- pyserial
- numpy

Las siguientes páginas explican cómo instalar las dependencias.

2 Configuración en un servidor local

Esta sección muestra el proceso de instalación de las dependencias correspondientes para configurar el software LabWebUQ en un computador personal para generar redes personales o crear proyectos individuales.

Se puede instalar en los siguientes sistemas operativos:

- Windows
- GNU/Linux (basado en Ubuntu)
- GNU/Linux (basado en Arch Linux)

Las siguientes páginas explican cómo instalar las dependencias.

2.1 Proceso inicial en GNU/Linux (Todas las distribuciones)

El software esta creado con el framework llamado django, el cual tiene como lenguaje de programación python, el paso para la instalación es el siguiente:

Basado en Ubuntu:

```
sudo apt install python3  
sudo apt install python3-pip
```

Basado en Arch Linux:

```
sudo pacman -S python  
sudo pacman -S python-pip
```

Nota: Muchas veces las distribuciones de Linux tienen instaladas las dependencias por defecto, en este caso el sistema informará que los paquetes ya están instalados.

Ahora se procede a instalar django y las dependencias necesarias:

```
sudo pip3 install django django-extensions opencv-contrib-python pyserial numpy
```

2.2 Proceso inicial en Microsoft Windows.

Antes de instalar django, es necesario instalar python junto a las dependencias del entorno de trabajo, para ello primero debe verificar la versión de Windows, para ello presione la tecla de Windows + Pause / break, esto abrirá la información de tu sistema, ahí visualiza si tu versión es de 64 bits o de 32 bits. Ahora con la versión de su sistema operativo, vaya a la página siguiente y descargue la última versión de python 3 <https://www.python.org/downloads/windows/>; debe tener en cuenta la versión de su sistema operativo e instalar el programa

Nota: cuando esté instalando el programa seleccione la opción: «Add python 3.X to PATH»

Ahora se debe instalar django y sus dependencias, para ello presione las teclas Windows + R y escribe cmd, se desplegará una ventana conocida como línea de comandos, allí se ejecuta:

```
python3 -m pip install --upgrade pip
```

Y finalmente se ejecuta:

```
pip install django django-extensions opencv-contrib-python pyserial numpy
```

3 Finalización de la configuración local y ejecución

Cuando tenga instaladas las dependencias se procede a ejecutar el proyecto, la siguiente configuración es la misma para windows y para GNU/Linux.

3.1 Configuración inicial

Nota: Este procedimiento debe hacerlo una vez

Para crear la base de datos, se procede a generar la migración de las estructuras de las tablas, para lo cual se ingresa a la carpeta LabWebUQ del proyecto, allí se ingresan los siguientes comandos:

```
python3 manage.py makemigrations  
python3 manage.py migrate
```

Ahora se debe crear el usuario administrador:

```
python3 manage.py createsuperuser
```

Se pedirán los siguientes datos del administrador:

- Username
- Email
- Password

En este momento la aplicación está totalmente configurada

3.2 Ejecución

Cada vez que quiera iniciar la aplicación, debe ingresar a la carpeta del proyecto «LabWebUQ» y escribir el siguiente comando:

```
python3 manage.py runserver
```

Entonces debe ir a la URL por defecto que es <http://127.0.0.1:8000>, allí encontrará la aplicación ejecutándose, puede ingresar con el usuario y contraseña generados en el paso anterior.

3.3 Correr como servidor

Si desea tener una computadora como servidor local para generar una red local alrededor de la aplicación, debe conectarse a cualquier red WiFi, al igual que los hosts que deseen acceder a la aplicación, allí debe saber la dirección IP asignada e iniciar la aplicación con el comando:

```
python3 manage.py runserver <IP>:8000
```

Los otros hosts deben acceder al navegador web y escribir `http://<IP>:8000`, con esto se tiene la aplicación configurada correctamente

3.4 Uso de las tarjetas

Cada planta es manejada por una tarjeta que se comunica por puerto serial con el servidor, para poder accederla desde el sistema web debe ingresar, desde la línea de comandos, a la carpeta del proyecto «LabWebUQ» y digitar lo siguiente:

```
python manage.py runscript puerto --script-args <puerto>
```

<puerto> corresponde al puerto usado por la tarjeta. Debe abrir una ventana diferente por cada planta a la que se desee acceder.

4 Administrador

El perfil de administrador puede realizar las siguientes operaciones:

- Agregar, ver y editar usuarios
- Crear y editar Plantas
- Crear y editar Prácticas
- Crear horarios para las prácticas
- Asignar usuarios a las prácticas

5 Usuario no administrador

El perfil de usuario no administrador puede realizar las siguientes operaciones:

- Solicitar citas en el laboratorio remoto
- Realizar la práctica acorde a la cita solicitada

6 Sobre el proyecto

Este trabajo fue realizado por:

- Jaiber Evelio Cardona Aristizábal (jaibercardona@uniquindio.edu.co),
- José Gabriel Hoyos Gutiérrez (josegabrielh@uniquindio.edu.co),
- Pablo Andrés Muñoz Gutiérrez (pamunoz@uniquindio.edu.co),
- Diego Fernando Ramírez Jiménez (dframirez@uniquindio.edu.co),

Universidad del Quindío 2021