

INSTALACIÓN DEL SISTEMA

Introducción

Producción en una PC con Linux Mint (Debian/Ubuntu-based), contemplando:

- Backend FastAPI que también sirve fronts estáticos.
- Servicio de teclados (el “agente/servicio” que manda POST al backend).
- Arranque automático con `systemd`.
- Acceso en LAN vía `nginx` (opcional pero recomendado).
- Logs, permisos y estructura prolija.

Nota: Proxy delante, proceso gestionado, reinicio automático. FastAPI recomienda usar proxy como Nginx/Traefik y para workers, Uvicorn recomienda Gunicorn con `uvicorn.workers.UvicornWorker`

Estructura de directorios

```
BOTONERA/  
  app/  
    main.py  
    web_static/  
      api/ config/ models/ services/ utils/  
    devices_services/  
      teclados_fisicos/  
        input_devices_service.py  
    data/  
    logs/  
    venv/  
  venv/  
  logs/  
  config.json
```

Flujo del sistema

```
[Nginx :80]
  v
[Gunicorn + UvicornWorker]
  v
[FastAPI app.main:app]
  |
  +--> /monitor-simple -> HTML/JS/CSS
  +--> /moderacion     -> HTML/JS/CSS
  +--> /pantalla       -> HTML/JS/CSS
  +--> /bancas/*.png   -> imágenes
  +--> /estados/*      -> JSON estado
  +--> /entradas/tecla -> POST desde servicio teclados
```

En paralelo:

```
[Servicio teclados]
  |
  +--> POST /entradas/tecla
```

En el mismo servidor, hablando por `127.0.0.1`.

Instalación de recursos

Dependencias:

```
sudo apt update

sudo apt install -y git python3 python3-venv python3-pip nginx
```

Usuario de servicio y directorio:

```
sudo adduser --system --group --home /opt/botonera botonera

sudo mkdir -p /opt/botonera

sudo chown -R botonera:botonera /opt/botonera

sudo chmod -R 775 /opt/botonera
```

Agregar usuario de uso al grupo `botonera` :

```
sudo usermod -aG botonera <Tu_usuario>
```

Reiniciar para que tome el grupo.

Clonar repositorio:

```
sudo -u botonera git clone https://github.com/TU_USUARIO/TU_REPO.git  
/opt/botonera/BOTONERA
```

Entornos virtuales

Backend

```
cd /opt/botonera/BOTONERA  
  
sudo -u botonera python3 -m venv venv  
  
sudo -u botonera ./venv/bin/python -m pip install --upgrade pip  
  
sudo -u botonera ./venv/bin/pip install -r requirements-backend.txt
```

Servicio teclados

```
cd /opt/botonera/BOTONERA/devices_services/teclados_fisicos  
  
sudo -u botonera python3 -m venv venv  
  
sudo -u botonera ./venv/bin/python -m pip install --upgrade pip  
  
sudo -u botonera ./venv/bin/pip install -r requirements.txt
```

Permisos para teclados:

```
sudo usermod -aG input botonera
```

(Reiniciar PC).

Crear servicios systemd

Backend (gunicorn)

Crear:

```
sudo nano /etc/systemd/system/botonera-backend.service
```

Contenido:

```
[Unit]

Description=Botonera Backend (FastAPI)

After=network.target

[Service]

User=botonera

Group=botonera

WorkingDirectory=/opt/botonera/BOTONERA

Environment=PYTHONUNBUFFERED=1

ExecStart=/opt/botonera/BOTONERA/venv/bin/gunicorn -w 1 -k
uvicorn.workers.UvicornWorker app.main:app --bind 127.0.0.1:8000

Restart=always

RestartSec=2

StandardOutput=journal

StandardError=journal

[Install]

WantedBy=multi-user.target
```

Activar:

```
sudo systemctl daemon-reload

sudo systemctl enable --now botonera-backend

sudo systemctl status botonera-backend
```

Logs:

```
journalctl -u botonera-backend -f
```

Servicio teclados

Crear:

```
sudo nano /etc/systemd/system/botonera-teclados.service
```

Contenido:

```
[Unit]

Description=Botonera Servicio Teclados

After=network.target botonera-backend.service

[Service]

User=botonera

Group=botonera

WorkingDirectory=/opt/botonera/BOTONERA/devices_services/teclados_fisicos

Environment=PYTHONUNBUFFERED=1

Environment=API_BASE_URL=http://127.0.0.1:8000

ExecStart=/opt/botonera/BOTONERA/devices_services/teclados_fisicos/venv/bin/python input_devices_service.py

Restart=always

RestartSec=2

StandardOutput=journal

StandardError=journal

[Install]

WantedBy=multi-user.target
```

Activar:

```
sudo systemctl daemon-reload  
  
sudo systemctl enable --now botonera-teclados  
  
sudo systemctl status botonera-teclados
```

Logs:

```
journalctl -u botonera-teclados -f
```

Crear comandos para mapear y presentar estado

Crear el comando `teclados-menu`

```
sudo nano /usr/local/bin/teclados-menu
```

Contenido:

```
#!/usr/bin/env bash  
  
cd /opt/botonera/BOTONERA/devices_services/teclados_fisicos  
  
# Paramos el servicio para evitar que ambos escuchen teclas a la vez  
  
sudo systemctl stop botonera-teclados  
  
# Abrimos el menú interactivo  
  
sudo -u botonera ./venv/bin/python input_devices_service.py  
  
# Al salir del menú, volvemos a levantar el servicio  
  
sudo systemctl start botonera-teclados
```

Permisos:

```
sudo chmod +x /usr/local/bin/teclados-menu
```

Uso:

```
teclados-menu
```

Crear el comando `teclados-log`

Crea:

```
sudo nano /usr/local/bin/teclados-log
```

Contenido:

```
#!/usr/bin/env bash  
  
exec journalctl -u botonera-teclados -f
```

Permisos:

```
sudo chmod +x /usr/local/bin/teclados-log
```

Uso:

```
teclados-log
```

✓ log en vivo del servicio.

Configurar auto-inicio en una terminal (Linux Mint)

1. Menú → Configuración → Sistema → Sesión e inicio
2. Pestaña “Inicio automático de aplicaciones”
3. Añadir
4. Completar:

Nombre: “Logs_en_inicio”

Descripción: “Logs de sistemas teclado al inicio”

Orden: (elegí según escritorio):

Cinnamon / GNOME:

```
gnome-terminal -- bash -c "teclados-log"
```

MATE:

```
mate-terminal -- bash -c "teclados-log"
```

XFCE:

```
xfce4-terminal --hold -e "teclados-log"
```

Desencadenar: Al iniciar sesión

Guardar.

Nginx Server

Archivo:

```
sudo nano /etc/nginx/sites-available/botonera
```

Contenido:

```
server {  
  
    listen 80;  
  
    server_name _;  
  
    location / {  
  
        proxy_pass http://127.0.0.1:8000;  
  
        proxy_set_header Host $host;  
  
        proxy_set_header X-Real-IP $remote_addr;  
  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
  
        proxy_set_header X-Forwarded-Proto $scheme;  
  
    }  
  
}
```

Activar:

```
sudo ln -sf /etc/nginx/sites-available/botonera  
/etc/nginx/sites-enabled/botonera  
  
sudo nginx -t  
  
sudo systemctl reload nginx
```

Verificación

Ejecución como usuario

Backend:

```
cd /opt/botonera/BOTONERA
sudo -u botonera opt/botonera/BOTONERA./venv/bin/python -m uvicorn
app.main:app --host 127.0.0.1 --port 8000
```

Debe estar detenido el servicio: "/etc/systemd/system/botonera-backend.service" (sudo systemctl stop botonera-backend)

En otra terminal probar:

```
curl -I http://127.0.0.1:8000/monitor-simple/
curl -i http://127.0.0.1:8000/estados/estado_global ; echo
```

Cortar uvicorn con **Ctrl+C**.

Servicio teclados:

```
cd /opt/botonera/BOTONERA
sudo -u botonera ./devices_services/teclados_fisicos/venv/bin/python
./devices_services/teclados_fisicos/input_devices_service.py
```

Debe estar detenido el servicio: "/etc/systemd/system/botonera-teclados.service" (sudo systemctl stop botonera-teclados)

servicios

Backend

```
sudo systemctl status botonera-backend --no-pager
```

```
curl -I http://127.0.0.1:8000/monitor-simple/
curl -i http://127.0.0.1:8000/estados/estado_global ; echo
```

Desde otra PC en la LAN:

```
curl -I http://IP_DE_SERVIDOR/monitor-simple
```

Ver logs en vivo:

```
journalctl -u botonera-backend -f
```

Teclados

Servicio:

```
sudo systemctl status botonera-teclados --no-pager
```

Log desde journalctl:

```
journalctl -u botonera-teclados -f
```

log con comando:

```
teclados-log
```

Comando mapeo:

```
teclados-menu
```

Script update

Crear:

```
sudo nano /opt/botonera/update.sh
```

Contenido:

```
#!/usr/bin/env bash

set -e

cd /opt/botonera/BOTONERA

echo "→ git sync"

sudo -u botonera git fetch --all

sudo -u botonera git reset --hard origin/main
```

```
echo "→ backend deps"

sudo -u botonera ./venv/bin/pip install -r requirements-backend.txt

echo "→ teclados deps"

sudo -u botonera ./devices_services/teclados_fisicos/venv/bin/pip install -r
devices_services/teclados_fisicos/requirements.txt

echo "→ restart"

sudo systemctl restart botonera-backend

sudo systemctl restart botonera-teclados

echo "✓ OK"
```

Permisos:

```
sudo chmod +x /opt/botonera/update.sh
```

Desactivar la protección para ese path:

```
git config --global --add safe.directory /opt/botonera/BOTONERA
```

Uso:

```
sudo /opt/botonera/update.sh
```

Resumen Comandos

Nota: Arranque normal de PC → autoinicia botonera-backend + botonera-teclados + logs en vivo teclados

Ver estados de los servicios:

```
sudo systemctl status botonera-backend --no-pager  
sudo systemctl status botonera-teclados --no-pager
```

Ver logs de servicios:

```
sudo systemctl status botonera-backend --no-pager  
teclados-log
```

On/off de servicios:

```
sudo systemctl stop botonera-backend  
sudo systemctl stop botonera-teclados  
sudo systemctl restart botonera-backend  
sudo systemctl restart botonera-teclados
```

Mapear teclas y herramientas de inputs debugging:

```
teclados-menu
```

Actualizar:

```
sudo /opt/botonera/update.sh
```