



Guía de Despliegue en Raspberry Pi / Mini-PC



Hardware Recomendado

Raspberry Pi

- **Modelo:** Raspberry Pi 4 Model B (4GB o 8GB RAM)
- **Almacenamiento:** MicroSD 32GB+ (Clase 10) o SSD USB
- **Cámara:** Raspberry Pi Camera Module v2 o Webcam USB
- **Fuente:** 5V 3A oficial
- **Opcional:** Case con ventilador para mejor refrigeración

Mini-PC

- **Procesador:** Intel i3 o AMD Ryzen 3 (mínimo)
- **RAM:** 4GB mínimo (8GB recomendado)
- **Almacenamiento:** 64GB+ SSD
- **Cámara:** Webcam USB o integrada



Instalación en Raspberry Pi

1. Preparar Sistema Operativo

```
# Instalar Raspberry Pi OS (64-bit recomendado)
# Usar Raspberry Pi Imager: https://www.raspberrypi.com/software/

# Actualizar sistema
sudo apt update && sudo apt upgrade -y

# Instalar dependencias del sistema
sudo apt install -y python3-pip python3-venv \
    libopencv-dev python3-opencv \
    libatlas-base-dev libopenblas-dev \
    libhdf5-dev libhdf5-serial-dev \
    libjpeg-dev zlib1g-dev libpng-dev \
    libavcodec-dev libavformat-dev libswscale-dev \
    v4l-utils
```

2. Configurar Cámara

Para Raspberry Pi Camera Module:

```
# Habilitar cámara
sudo raspi-config
# Interface Options -> Camera -> Enable

# Verificar
vcgencmd get_camera
# Debe mostrar: supported=1 detected=1
```

Para Webcam USB:

```
# Listar dispositivos
ls -l /dev/video*

# Probar cámara
v4l2-ctl --list-devices
```

3. Instalar Aplicación

```
# Clonar repositorio
cd /home/pi
git clone <tu-repositorio> attendance_system
cd attendance_system

# Crear entorno virtual
python3 -m venv venv
source venv/bin/activate

# Actualizar pip
pip install --upgrade pip setuptools wheel

# Instalar dependencias
pip install -r requirements.txt

# En Raspberry Pi, FAISS puede requerir compilación especial
# Si hay error con faiss-cpu, usar:
pip install faiss-cpu --no-cache-dir
```

4. Configurar Aplicación

```
# Copiar archivo de configuración
cp .env.example .env

# Editar configuración
nano .env
```

Configuración recomendada para Raspberry Pi:

```
SECRET_KEY=<generar-clave-segura>
FLASK_ENV=production
DEBUG=False

# Optimizar para Raspberry Pi
FACE_DETECTION_CONFIDENCE=0.5
FACE_SIMILARITY_THRESHOLD=0.55
CAMERA_WIDTH=640
CAMERA_HEIGHT=480
CAMERA_FPS=15

DEVICE_ID=rpi-001
DEVICE_NAME=Terminal Raspberry Pi
```

5. Inicializar Base de Datos

```
python scripts/init_db.py
```

6. Probar Sistema

```
python scripts/test_system.py
```



Configurar como Servicio Systemd

Para que la aplicación se inicie automáticamente:

1. Crear archivo de servicio

```
sudo nano /etc/systemd/system/attendance.service
```

2. Contenido del servicio:

```
[Unit]
Description=Sistema de Asistencia con Reconocimiento Facial
After=network.target

[Service]
Type=simple
User=pi
WorkingDirectory=/home/pi/attendance_system
Environment="PATH=/home/pi/attendance_system/venv/bin"
ExecStart=/home/pi/attendance_system/venv/bin/python run.py
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

3. Activar servicio

```
# Recargar systemd
sudo systemctl daemon-reload

# Habilitar servicio
sudo systemctl enable attendance.service

# Iniciar servicio
sudo systemctl start attendance.service

# Verificar estado
sudo systemctl status attendance.service

# Ver logs
sudo journalctl -u attendance.service -f
```

Configurar Acceso de Red

1. Encontrar IP de Raspberry Pi

```
hostname -I
```

2. Configurar IP Estática

```
sudo nano /etc/dhcpd.conf
```

Agregar al final:

```
interface eth0
static ip_address=192.168.1.100/24
static routers=192.168.1.1
static domain_name_servers=8.8.8.8
```

3. Reiniciar red

```
sudo systemctl restart dhcpd
```

4. Acceder desde otros dispositivos

```
http://192.168.1.100:5000
```



Configurar HTTPS con Nginx (Recomendado)

1. Instalar Nginx

```
sudo apt install nginx -y
```

2. Configurar proxy inverso

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

```
server {
    listen 80;
    server_name attendance.local;

    location / {
        proxy_pass http://127.0.0.1:5000;
        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;
    }
}
```

3. Activar sitio

```
sudo ln -s /etc/nginx/sites-available/attendance /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```



Optimización de Rendimiento

1. Overclocking Raspberry Pi 4 (Opcional)

```
sudo nano /boot/config.txt
```

Agregar:

```
over_voltage=6
arm_freq=2000
gpu_freq=750
```

2. Configurar Swap

```
sudo dphys-swapfile swapoff
sudo nano /etc/dphys-swapfile
# Cambiar CONF_SWAPSIZE=2048
sudo dphys-swapfile setup
sudo dphys-swapfile swapon
```

3. Usar SSD en lugar de MicroSD

- Mejora significativa de velocidad
- Mayor durabilidad
- Boot desde USB en Raspberry Pi 4



Monitoreo

Ver logs de aplicación

```
# Logs del sistema
tail -f /home/pi/attendance_system/logs/app.log

# Logs del servicio
sudo journalctl -u attendance.service -f
```

Monitorear recursos

```
# CPU y RAM
htop

# Temperatura
vcgencmd measure_temp

# Espacio en disco
df -h
```



Backup y Restauración

Backup automático

```
# Crear script de backup
nano /home/pi/backup.sh
```

```
#!/bin/bash
BACKUP_DIR="/home/pi/backups"
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR

# Backup de base de datos
cp /home/pi/attendance_system/data/database/attendance.db \
  $BACKUP_DIR/attendance_$DATE.db

# Backup de fotos
tar -czf $BACKUP_DIR/photos_$DATE.tar.gz \
  /home/pi/attendance_system/data/uploads/

# Limpiar backups antiguos (mantener últimos 7 días)
find $BACKUP_DIR -type f -mtime +7 -delete

echo "Backup completado: $DATE"
```

```
# Hacer ejecutable
chmod +x /home/pi/backup.sh

# Programar con cron (diario a las 2 AM)
crontab -e
# Agregar: 0 2 * * * /home/pi/backup.sh
```

Solución de Problemas

Aplicación no inicia

```
# Verificar logs
sudo journalctl -u attendance.service -n 50

# Verificar puerto
sudo netstat -tulpn | grep 5000

# Probar manualmente
cd /home/pi/attendance_system
source venv/bin/activate
python run.py
```

Cámara no funciona

```
# Verificar dispositivo
ls -l /dev/video*

# Permisos de usuario
sudo usermod -a -G video pi

# Probar cámara
raspistill -o test.jpg # Para Pi Camera
fswebcam test.jpg      # Para USB webcam
```

Bajo rendimiento

1. Reducir resolución de cámara en `.env`
2. Aumentar swap
3. Cerrar aplicaciones innecesarias
4. Considerar overclocking
5. Usar SSD en lugar de MicroSD

Actualización de la Aplicación

```
# Detener servicio
sudo systemctl stop attendance.service

# Actualizar código
cd /home/pi/attendance_system
git pull

# Activar entorno virtual
source venv/bin/activate

# Actualizar dependencias
pip install -r requirements.txt --upgrade

# Reiniciar servicio
sudo systemctl start attendance.service
```



Checklist de Despliegue

- ☐ Hardware preparado y conectado
 - ☐ Sistema operativo actualizado
 - ☐ Cámara configurada y funcionando
 - ☐ Aplicación instalada y configurada
 - ☐ Base de datos inicializada
 - ☐ Pruebas del sistema realizadas
 - ☐ Servicio systemd configurado
 - ☐ IP estática configurada (opcional)
 - ☐ Nginx configurado (opcional)
 - ☐ Backup automático configurado
 - ☐ Monitoreo configurado
 - ☐ Documentación para usuarios preparada
-

☒ **Sistema listo para producción**