GPIO en Turnero (Raspberry Pi)

Servicio Python + despachador PHP

Objetivo

Leer pulsadores físicos conectados a los pines GPIO y, al **apretar**, llamar a la API del Turnero:

- **17** → `/api/botones/nuevo.php`
- **27** → `/api/botones/siguiente.php`
- **22** → `/api/botones/anterior.php`

Esta guía instala todo bajo **`/var/www/html/turnero/pi`** y un **servicio systemd** que queda escuchando los pines.

Requisitos

- Raspberry Pi con Raspberry Pi OS.
- Paquetes del sistema:

```
sudo apt update
sudo apt install -y python3 python3-rpi.gpio php-curl
```

- Turnero funcionando (DB + API). Verifica que respondan:

```
curl -s -X POST http://127.0.0.1/turnero/api/botones/nuevo.php --data "queue_id=1"
curl -s -X POST http://127.0.0.1/turnero/api/botones/siguiente.php --data "queue_id=1"
curl -s -X POST http://127.0.0.1/turnero/api/botones/anterior.php --data "queue_id=1"
```

Cableado

- Cada pulsador entre **GPIO GND** (usaremos **pull-up interno**).
- Numeración **BCM**:
  - **BCM 17** (físico 11): Nuevo
  - **BCM 27** (físico 13): Siguiente
  - **BCM 22** (físico 15): Anterior

En reposo el pin lee **1**, al apretar (cerrar a GND) lee **0**.

Archivos

Crea la carpeta **`/var/www/html/turnero/pi`** si no existe:

```
sudo mkdir -p /var/www/html/turnero/pi
sudo chown -R www-data:www-data /var/www/html/turnero/pi
sudo chmod 755 /var/www/html/turnero/pi
```

1) monitor_pullup_multi.py (lector GPIO con pull-up)

Ruta: /var/www/html/turnero/pi/monitor_pullup_multi.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import argparse, time, json, requests
import RPi.GPIO as GPIO

def report(url, pin, value, queue_id=None):
    try:
        payload = {'pin': pin, 'value': value}
        if queue_id:
            payload['queue_id'] = queue_id
        r = requests.post(url, data=json.dumps(payload),
                          headers={'Content-Type':'application/json'},
                          timeout=2)
        print(f"[report] pin={pin} value={value} -> {r.status_code}")
    except Exception as e:
        print("[report] error:", e)

def main():
    ap = argparse.ArgumentParser()
    ap.add_argument('--pins', type=int, nargs='+', required=True, help='BCM pins')
    ap.add_argument('--report-url', required=True)
    ap.add_argument('--queue-id', type=int, default=1)
    ap.add_argument('--debounce', type=int, default=200, help='ms')
    ap.add_argument('--interval', type=float, default=0.02)
    args = ap.parse_args()

    GPIO.setmode(GPIO.BCM)
    for p in args.pins:
        GPIO.setup(p, GPIO.IN, pull_up_down=GPIO.PUD_UP)

    last = {p: GPIO.input(p) for p in args.pins}
    stable_since = {p: time.time() for p in args.pins}
    print(f"[multi] PULL-UP activo en BCM {args.pins}. Reposo=1, Apretado=0")

    try:
        while True:
            now = time.time()
            for p in args.pins:
```

```
                v = GPIO.input(p)
                if v != last[p]:
                    if (now - stable_since[p]) * 1000 >= args.debounce:
                        report(args.report_url, p, f"event:{v}", args.queue_id)
                        last[p] = v
                        stable_since[p] = now
                else:
                    stable_since[p] = now
            time.sleep(args.interval)
    except KeyboardInterrupt:
        pass
    finally:
        GPIO.cleanup()

if __name__ == "__main__":
    main()
```

2) gpio_dispatch.php (router → API botones)

Ruta: /var/www/html/turnero/pi/gpio_dispatch.php

```
<?php
declare(strict_types=1);
header('Content-Type: application/json; charset=utf-8');
error_reporting(0); ini_set('display_errors','0');

/* Config */
const DEFAULT_QUEUE_ID = 1; // cola por defecto si no mandan otra
$PIN_MAP = [
  17 => 'nuevo',
  27 => 'siguiente',
  22 => 'anterior',
];

/* Solo localhost */
$ip = $_SERVER['REMOTE_ADDR'] ?? '';
if (!in_array($ip, ['127.0.0.1','::1'], true)) {
  http_response_code(403);
  echo json_encode(['ok'=>false,'error'=>'Forbidden']); exit;
}

/* Leer JSON/form */
$raw  = file_get_contents('php://input') ?: '';
$data = json_decode($raw, true);
if (!is_array($data)) {
  $data = [
```

```
    'pin'      => isset($_POST['pin']) ? (int)$_POST['pin'] : null,
    'value'    => (string)($_POST['value'] ?? ''),
    'queue_id' => isset($_POST['queue_id']) ? (int)$_POST['queue_id'] : null,
  ];
}

$pin   = isset($data['pin']) ? (int)$data['pin'] : null;
$value = (string)($data['value'] ?? '');
$qid   = isset($data['queue_id']) ? max(1,(int)$data['queue_id']) : DEFAULT_QUEUE_ID;

/* Log simple (últimos 200 eventos) */
$logf = __DIR__ . '/gpio_events.json';
$ev   = ['pin'=>$pin,'value'=>$value,'queue'=>$qid,'ts'=>time(),'ts_iso'=>date('c')];
$events = [];
if (is_file($logf)) { $prev = json_decode((string)@file_get_contents($logf), true); if (is_a
array_unshift($events, $ev); $events = array_slice($events,0,200);
@file_put_contents($logf, json_encode($events, JSON_PRETTY_PRINT|JSON_UNESCAPED_UNICODE));

/* Disparar solo en APRETADO (event:0) */
if (strpos($value,'event:')===0 ? substr($value,6)==='0' : $value==='pressed') {
  $action = $PIN_MAP[$pin] ?? null;
  if ($action) {
    $url = "http://localhost/turnero/api/botones/{$action}.php";
    $body = http_build_query(['queue_id'=>$qid], '', '&');
    $ch = curl_init($url);
    curl_setopt_array($ch, [
      CURLOPT_POST=>true, CURLOPT_POSTFIELDS=>$body,
      CURLOPT_HTTPHEADER=>['Accept: application/json','Content-Type: application/x-www-form-
      CURLOPT_RETURNTRANSFER=>true, CURLOPT_TIMEOUT=>5,
    ]);
    $resp = curl_exec($ch);
    $code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    curl_close($ch);
    if ($resp===false || $code<200 || $code>=300) {
      echo json_encode(['ok'=>false,'event'=>$ev,'action'=>$action,'forward_status'=>$code,'
    }
    $out = json_decode($resp,true);
    echo json_encode(['ok'=>true,'event'=>$ev,'action'=>$action,'result'=>$out]); exit;
  }
}

echo json_encode(['ok'=>true,'event'=>$ev,'note'=>'logged-only']);
```

3) Servicio `systemd`

Ruta: `/etc/systemd/system/gpio_buttons.service`

```
[Unit]
Description=Turnero GPIO Buttons (multipin pull-up)
After=network.target

[Service]
Type=simple
User=root
WorkingDirectory=/var/www/html/turnero/pi
Environment=PYTHONUNBUFFERED=1
ExecStart=/usr/bin/python3 /var/www/html/turnero/pi/monitor_pullup_multi.py --pins 17 27 22
Restart=always
RestartSec=2

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

Activar y ver estado:

```
sudo systemctl daemon-reload
sudo systemctl enable --now gpio_buttons.service
sudo systemctl status gpio_buttons.service --no-pager
journalctl -u gpio_buttons.service -f
```

Pruebas

1) **Simulación sin hardware** (empujar evento al dispatcher):

```
curl -s -X POST http://127.0.0.1/turnero/pi/gpio_dispatch.php   -H 'Content-Type: applicatio
```

2) **Ver eventos guardados**:

```
tail -n 10 /var/www/html/turnero/pi/gpio_events.json
```

3) **Estado de la pantalla**:

```
curl -s "http://127.0.0.1/turnero/api/queues_state.php?queue_id=1&next_limit=5&_=$(date +%s)
```

4) **Con hardware**: presiona el pulsador de **BCM17** → debe aparecer un turno nuevo en `next[]`. BCM27 **avanza** el turno; BCM22 **retrocede**.

Personalización

- **Otra cola**: cambia `--queue-id 1` en el `ExecStart` del servicio (o envía `queue_id` en el JSON si tu fuente lo permite).
- **Rebote**: ajusta `--debounce` (ms). 200–300 ms suele ir bien para pulsadores mecánicos.

- **Mapeo de pines**: edita `$PIN_MAP` en `gpio_dispatch.php` si quieres otras funciones o pines.

Seguridad

- `gpio_dispatch.php` **solo acepta localhost**. Si reportaras desde otra máquina, agrega un token (ej. header `X-Button-Key`) y valida en el PHP.
- `php-curl` es obligatorio para que el dispatcher pueda llamar a los endpoints de tu app.

Logs y mantenimiento

- Logs del servicio:

  ```
  journalctl -u gpio_buttons.service -f
  ```

- Reiniciar servicio tras cambios:

  ```
  sudo systemctl restart gpio_buttons.service
  ```

- Desinstalar:

  ```
  sudo systemctl disable --now gpio_buttons.service
  sudo rm /etc/systemd/system/gpio_buttons.service
  sudo systemctl daemon-reload
  ```

Árbol final recomendado

```
/var/www/html/turnero/
  pi/
     README.md                  ← este archivo
     monitor_pullup_multi.py    ← servicio lector GPIO
     gpio_dispatch.php          ← router → /api/botones/*
     gpio_events.json           ← log de últimos eventos
```

Con esto, el flujo queda: **GPIO → monitor (pull-up) → dispatch PHP → `/api/botones/*` → DB → `queues_state.php` → pantalla.**