

---

Administració i Disseny de Sistemes

# MEMÒRIA DE LA PRÀCTICA FASE 2

---

Pol Piñol Castuera (pol.pinol)

Guillem Serra Cazorla (guillem.serra)

# Índex

<b>1</b>	<b>Introducció i Resum de la Pràctica</b>	<b>3</b>
<b>2</b>	<b>Instal·lacions prèvies Fase 1</b>	<b>3</b>
2.1	Sistema Operatiu i connexió SSH	3
2.2	Kernel Headers	3
2.3	GPIOs i interrupcions	4
<b>3</b>	<b>Instal·lacions prèvies Fase 2</b>	<b>5</b>
<b>4</b>	<b>Disseny de la interfície de la Web</b>	<b>7</b>
4.1	Inici de la Web	7
4.2	Registre d'un usuari	8
4.3	Menú amb totes les funcionalitats	9
4.4	Gestió d'usuaris	10
4.5	Programació de Crons	11
4.6	Filtrat de Paquets	12
4.7	Logs del sistema	13
4.8	Monitorització del sistema	14
4.9	Gestió de processos	15
4.10	Funcionalitats musicals	16
4.11	Llista de reproducció de la música	17
<b>5</b>	<b>Scripts Bash i Codi LKM</b>	<b>18</b>
5.1	Scripts per Menús	18
5.1.1	Index.sh	18
5.1.2	Webapp.sh	19
5.2	Scripts per Login	20
5.2.1	TryLogin.sh	20
5.3	Scripts per Signup	22
5.3.1	Signup.sh	22
5.3.2	TrySignup.sh	23
5.4	Scripts per Apagar	24
5.4.1	Apagar.sh	24
5.5	Scripts per Reiniciar	25
5.5.1	Reiniciar.sh	25
5.6	Scripts per Filtrar Paquets	26
5.6.1	Filter.sh	26
5.6.2	Filtratpaquets.sh	27
5.6.3	Veurefiltrats.sh	29
5.7	Scripts per Gestió de Logs	30
5.7.1	Gestiologs.sh	30
5.8	Scripts per Gestió d'Usuaris	31
5.8.1	Afegirusuari.sh	31
5.8.2	Eliminarusuari.sh	32
5.8.3	Gestiousuari.sh	33
5.8.4	Veuretotsusuaris.sh	34
5.9	Scripts per Monotorizació	35
5.9.1	Monotoritzacio.sh	35
5.10	Scripts per Gestió de Processos	36
5.10.1	Eliminarproces.sh	36
5.10.2	Eliminarprocessegons.sh	37
5.10.3	Estatproces.sh	38
5.10.4	Gestioprocessos.sh	39
5.10.5	KillInSeconds.sh	41

5.10.6	Veureprocessos.sh . . . . .	41
5.11	Scripts per Música . . . . .	42
5.11.1	Avansar.sh . . . . .	43
5.11.2	Llista.sh . . . . .	44
5.11.3	Musica.sh . . . . .	45
5.11.4	Playpause.sh . . . . .	46
5.11.5	Replay.sh . . . . .	47
5.11.6	Reproduir.sh . . . . .	48
5.11.7	Reproduir2.sh . . . . .	51
5.11.8	Retrocedir.sh . . . . .	51
5.11.9	Shuffle.sh . . . . .	52
5.11.10	Start.sh . . . . .	53
5.11.11	A.sh . . . . .	53
5.11.12	B.sh . . . . .	54
5.11.13	C.sh . . . . .	54
5.11.14	D.sh . . . . .	54
<b>6</b>	<b>Conclusions i Problemes Observats . . . . .</b>	<b>55</b>

# 1 Introducció i Resum de la Pràctica

---

En aquesta pràctica de l'assignatura Administració i Disseny de Sistemes programarem un driver per controlar un dispositiu bluetooth que és necessari per al servidor de testing i mantenir en bon estat el servidor d'impressió que utilitza tota l'empresa. També adaptarem el sistema a les necessitats oportunes de cada moment, sigui recompilant el Kernel i afegint nous LKM (Linux kernel modules).

En la primera fase de la pràctica es veurà com es pot afegir un LKM per a controlar una petita placa Raspberry Pi Zero amb uns botons auxiliars per encendre i apagar 2 LEDs en una protoboard.

En aquesta segona fase s'administrarà la resta del sistema operatiu. Per fer-ho caldrà dotar-lo d'un servidor WEB que el permeti gestionar remotament. En aquesta fase s'implementaran totes les funcionalitats del servidor WEB, configurant correctament el sistema operatiu. Aquest servidor Web podrà executar les següent funcionalitats: Login, Gestió de processos, Monitorització, Apagar i reiniciar, Gestió de logs, Gestió d'usuaris, Filtrat de paquets, Gestió de tasques preprogramades, Funcionalitats musicals.

## 2 Instal·lacions prèvies Fase 1

---

### 2.1 Sistema Operatiu i connexió SSH

Primer de tot cal especificar que s'ha treballat amb una Raspberry Pi Zero W v1.1 i amb una targeta Micro SD de 32GB.

Per el procés de la fase 1 de la pràctica s'ha començat amb la instal·lació del sistema operatiu Raspberry PI OS Lite de 32 bits. Primer de tot necessitem un ordinador auxiliar per a poder instal·lar Raspberry Pi Imager des del navegador. Un cop instal·lat i executat, seleccionarem el Sistema Operatiu que volem instal·lar a la Raspberry. En el nostre cas hem utilitzat Raspberry PI OS Full de 32 bits.

A continuació hem de pulsar Ctrl + Shift + X per activar les opcions avançades de la configuració. Ara activarem el SSH i declararem una contrasenya per connectar-nos. També activarem l'opció de wifi, ja que la nostra Raspberry suporta aquesta opció, per poder-nos connectar a través del SSH en un ordinador en la mateixa xarxa wifi.

Finalment escollim carregar tota la informació necessària i generada pel Raspberry Pi Imager a la targeta SD de 32GB. Un cop finalitzada la càrrega connectem la Raspberry i ens connectem via SSH a través del ordinador auxiliar en la mateixa xarxa wifi.

Creem una connexió SSH a través del programa MobaXterm a la Raspberry amb remote host i username que té la Raspberry per defecte (pi).

### 2.2 Kernel Headers

Els 'kernel headers' contenen els fitxers de capçalera C per al kernel de Linux. Aquests contenen les diferents definicions de funcions i estructures necessàries per compilar qualsevol codi que interaccioni amb el kernel, com ara mòduls del kernel o controladors de dispositius i alguns programes d'usuari.

Per tant, en el transcurs de la fase 1 és essencial instal·lar correctament els 'kernel headers'. Per fer-ho s'ha utilitzat la següent commanda que instal·larà els "kernel headers" necessaris per la versió de la nostra Raspberry.

1

```
sudo apt install raspberrypi-kernel-headers
```

## 2.3 GPIOs i interrupcions

La nostre intenció és modificar el funcionament dels GPIOs per efectuar la fase 1 de la pràctica. Un GPIO (General-purpose input/output) és un pin genèric d'un xip, el comportament del qual pot controlar l'usuari en temps d'execució.

Aquests GPIOs es poden controlar des de l'espai d'usuari de Linux mitjançant la interfície Sysfs GPIO. En el nostre cas utilitzarem un executable que ens permetrà activar els GPIOs que ens especifica l'enunciat per realitzar la fase 1. Destacar que es podden modificar el comportament de cada GPIO de Linux des de l'espai del kernel mitjançant les funcions que es descriuen a `linux/gpio.h`.

El driver LKM que volem crear ha de registrar una funció per controlar la interrupció, que defineix les accions que haurà de realitzar cada interrupció.

Aquesta funció s'inicia amb una sol·licitud d'interrupció (IRQ) mitjançant la funció `request_irq()`, el `irqNumber` que és el número d'interrupció associat al GPIO corresponent donat per l'enunciat. La nostre intenció és canviar per defecte el funcionament de la funció que controla la interrupció per una altre personalitzada.

Recordem que una interrupció és un senyal que s'envia a un microprocessador des del hardware, programa o circuit per indicar que s'ha produït un esdeveniment.

## 3 Instal·lacions prèvies Fase 2

Per a realitzar la instal·lació de tot el programari hem creat un script anomenat Install.sh. Aquest script es l'encarregat de actualitzar i instal·lar totes les dependències necessàries per a el correcte funcionament del sistema.

Dividirem aquest script en les següents fases:

Fase de instal·lació i actualització del programari. Nomes començar procedirem a fer un update i upgrade de tots els paquets del sistema. Aquesta acció tot i no ser del tot necessaria (el upgrade) es considera una bona praxis per tal de mantenir tot el programari actualitzat. En aquesta fase també descarregarem i instal·larem els headers.

A continuació ens baixarem el codi del repositori de github tot compilant els headers per els botons externs. També farà l'intent de instal·lar git per si fos el cas que no ho estigues instal·lat en la màquina.

Com a penúltim pas realitzem la gestió i adquisició de permisos per www-data i realitzem una cosa destacable, consisteix en buscar tots els fitxers del codi de scripts en busca de newlines de Windows i substituir-les per les de Unix. Aquesta línia la hem trobat donat a que ens hem anat trobant incompatibilitats durant el procés de desenvolupament del projecte.

Finalment iniciarem el servei remot i generarem la pipe per a donar-li els comandaments des de els altres scripts. Col·locarem la pipe a la localització /home/pi/Downloads tot i que ha estat un lloc per exemple, es podria modificar en qualsevol moment.

```
1  #!/bin/bash
2
3  sudo apt update
4
5  sudo apt install raspberrypi-kernel-headers
6
7  #TODO: demana el 'Y'
8  sudo apt install apache2
9
10 sudo apt-get install xrdp
11 sudo a2enmod cgid
12 sudo systemctl restart apache2
13
14 #TODO: Instalem git per crear els scripts al lloc corresponent
15 sudo apt-get install git
16 git clone https://github.com/GuillemSeCa/Raspberry_Spotify_Server $HOME/Desktop
17 cd $HOME/Desktop && git checkout dev
18
19 cd $HOME/Desktop/code && make && sudo insmod LKM.ko && cp A.sh /home/pi/A.sh &&
20 cp B.sh /home/pi/B.sh && cp C.sh /home/pi/C.sh && cp D.sh /home/pi/D.sh && chmod
21 +x /home/pi/A.sh /home/pi/B.sh /home/pi/C.sh /home/pi/D.sh
22
23 sudo cp -r $HOME/Desktop/* /usr/lib/cgi-bin
24
25 sudo chmod -R 755 /usr/lib/cgi-bin
26 echo 'www-data ALL=(ALL) NOPASSWD: ALL' | sudo EDITOR='tee -a' visudo
27
28 #Per transoformar \n s de Windows a format Unix
29 sudo apt-get install dos2unix
30 cd /usr/lib/cgi-bin && sudo find . -type f -print0 | sudo xargs -0 dos2unix
31
32 #Instalem les iptables
```

```

33 sudo apt-get install iptables
34
35 #Instalem el mpg123
36 sudo apt-get install mpg123
37
38 touch /home/pi/Desktop/logs.txt
39 sudo chmod 777 /home/pi/Desktop/logs.txt
40
41 touch /home/pi/Downloads/currentsong.txt
42 sudo chmod 777 /home/pi/Downloads/currentsong.txt
43 echo "1" > /home/pi/Downloads/currentsong.txt
44
45 touch /home/pi/Downloads/llista.txt
46 sudo chmod 777 /home/pi/Downloads/llista.txt
47 echo "1 song1.mp3" > /home/pi/Downloads/llista.txt
48 echo "2 song2.mp3" >> /home/pi/Downloads/llista.txt
49 echo "3 song3.mp3" >> /home/pi/Downloads/llista.txt
50 echo "4 song4.mp3" >> /home/pi/Downloads/llista.txt
51
52 sudo systemctl restart apache2

```

A més, per activar les funcionalitats musicals a través d'apache i el seu usuari www-data hem hagut de modificar els següents arxius. Modificarem el fitxer /etc/systemd/system/pulseaudio.service per afegir:

```

1 [Unit]
2 Description=PulseAudio system server
3
4 [Service]
5 Type=notify
6 ExecStart=pulseaudio --daemonize=no --system --realtime --log-target=journal
7
8 [Install]
9 WantedBy=multi-user.target

```

Farem enable dels serveis corresponents:

```

1 sudo systemctl --system enable pulseaudio.service
2 sudo systemctl --system start pulseaudio.service
3 sudo systemctl --system status pulseaudio.service

```

Finalment editarem el fitxer /etc/pulse/client.conf en els següent atributs:

```

1 default-server = /var/run/pulse/native
2 autospawn = no

```

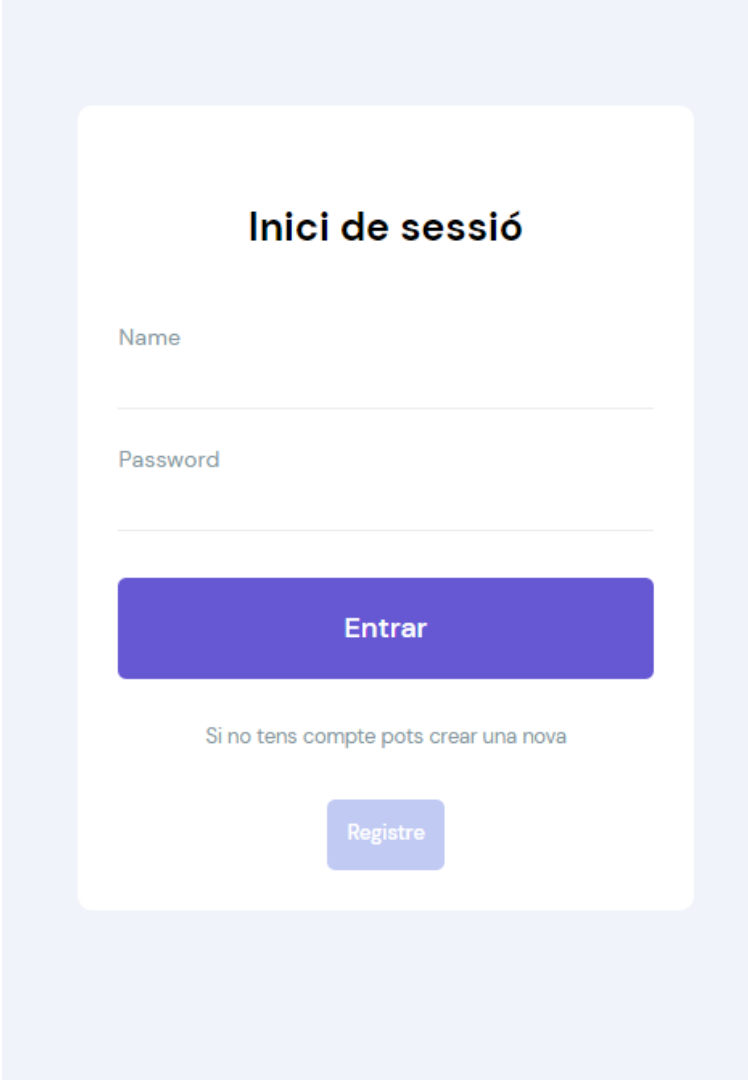
I afegirem el usuari www-data al grup corresponent: `sudo adduser www-data pulse-access`, `sudo adduser www-data audio` i `sudo adduser www-data pulse`. Finalment, fem un `restart systemctl -user` `restart pulseaudio.service`.

## 4 Disseny de la interfície de la Web

---

En aquest apartat es mostrarà el disseny final de la interfície web per controlar el servidor.

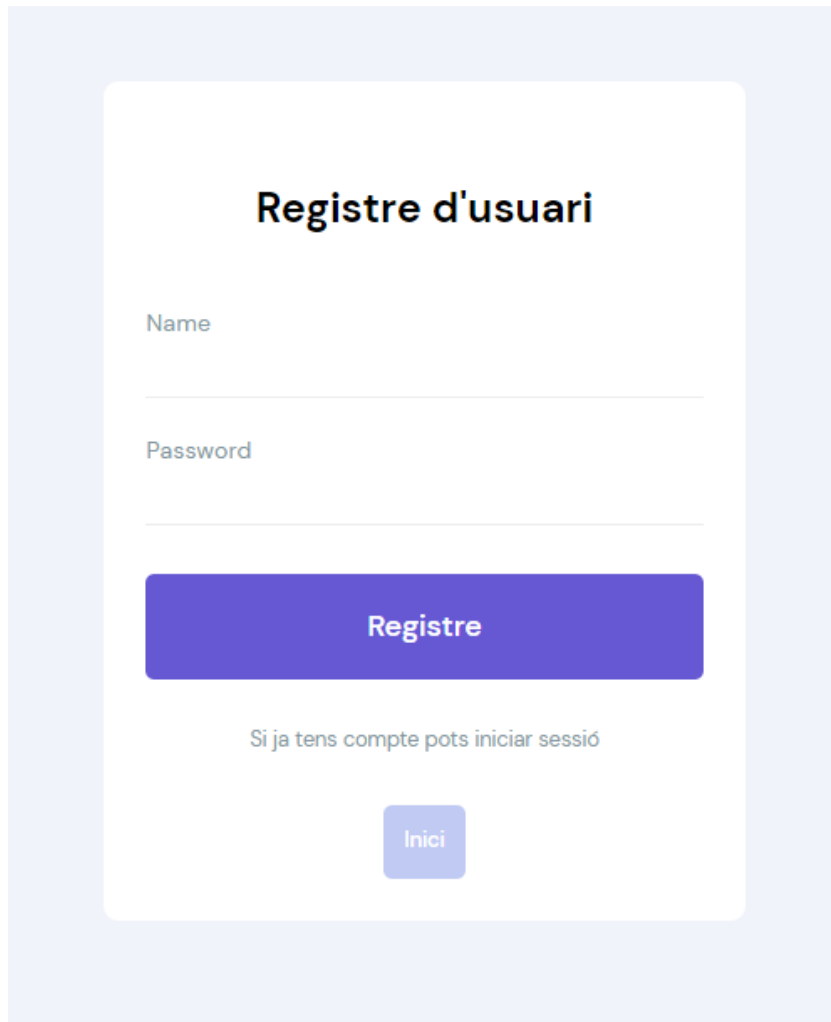
### 4.1 Inici de la Web



The image shows a login form design centered on a light blue background. The form itself is a white rounded rectangle. At the top of the form, the text "Inici de sessió" is displayed in a bold, black, sans-serif font. Below this, there are two input fields. The first is labeled "Name" in a small, grey font, followed by a thin horizontal line. The second is labeled "Password" in the same style, also followed by a thin horizontal line. Below the password field is a large, solid blue button with the word "Entrar" in white, bold, sans-serif font. Underneath the button, the text "Si no tens compte pots crear una nova" is written in a small, grey font. At the bottom of the form is a smaller, light blue button with the word "Registre" in a dark blue, sans-serif font.



## 4.2 Registre d'un usuari



The image shows a user registration form titled "Registre d'usuari". It features two input fields for "Name" and "Password", a prominent blue "Registre" button, and a link to log in with the text "Si ja tens compte pots iniciar sessió" and a light blue "Inici" button.

**Registre d'usuari**

Name

\_\_\_\_\_

Password

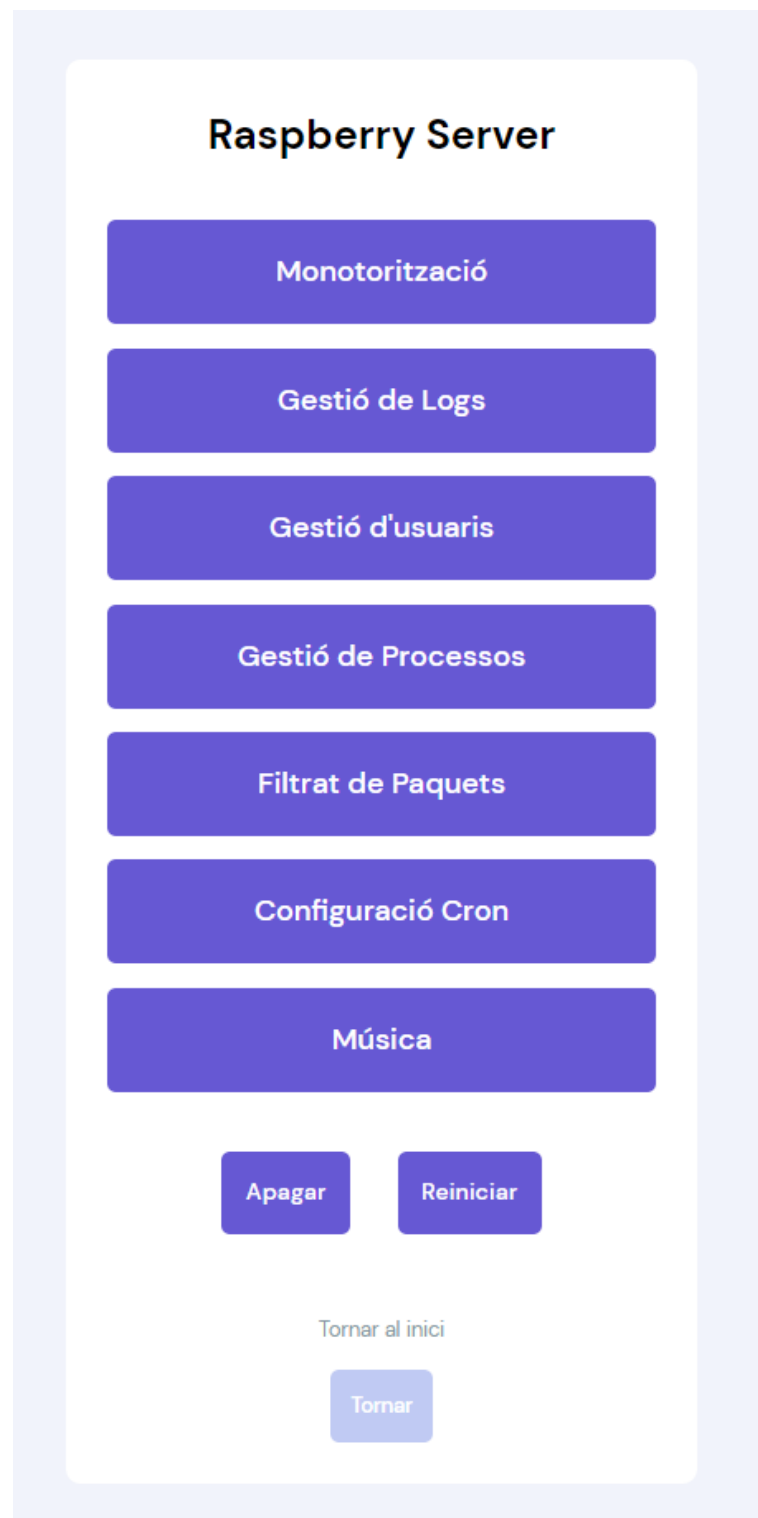
\_\_\_\_\_

**Registre**

Si ja tens compte pots iniciar sessió

Inici

### 4.3 Menú amb totes les funcionalitats



#### 4.4 Gestió d'usuaris

## Gestió d'usuaris

**Veure tots els usuaris**

[Mostra](#)

---

**Afegir un usuari**

Name

---

Password

---

[Afegir](#)

---

**Eliminar un usuari**

Name

---

[Eliminar](#)

[Tornar al inici](#)

## 4.5 Programació de Crons

### Configuració Cron

Veure tots els crons

Mostra

---

Afegir un cron

Name

---

Afegir

---

Eliminar un cron

Name

---

Eliminar

Tornar al inici

Tornar

## 4.6 Filtrat de Paquets

### Filtrat de Paquets

Veure tots els filtrats

Mostra

---

### Modificar un filtrat

Afegir o Eliminar (-A o -D)

---

Cadena (INPUT, OUTPUT o FORWARD)

---

Regla (ACCEPT o DROP)

---

Protocol

---

IP Origen

---

Port Origen

---

IP Destí

---

Port Destí

## 4.7 Logs del sistema

### Gestió de Logs

SYS\_LOGS: S'ha consultat la monotorització del sistema.

SYS\_LOGS: S'han mostrat la llista de reproducció de la música.

SYS\_LOGS: S'ha avançat una cançó de la llista de reproducció de la música.

SYS\_LOGS: S'han mostrat la llista de reproducció de la música.

SYS\_LOGS: S'han mostrat la llista de reproducció de la música.

SYS\_LOGS: S'han mostrat la llista de reproducció de la música.

SYS\_LOGS: S'han mostrat la llista de reproducció de la música.

SYS\_LOGS: S'han mostrat la llista de reproducció de la música.

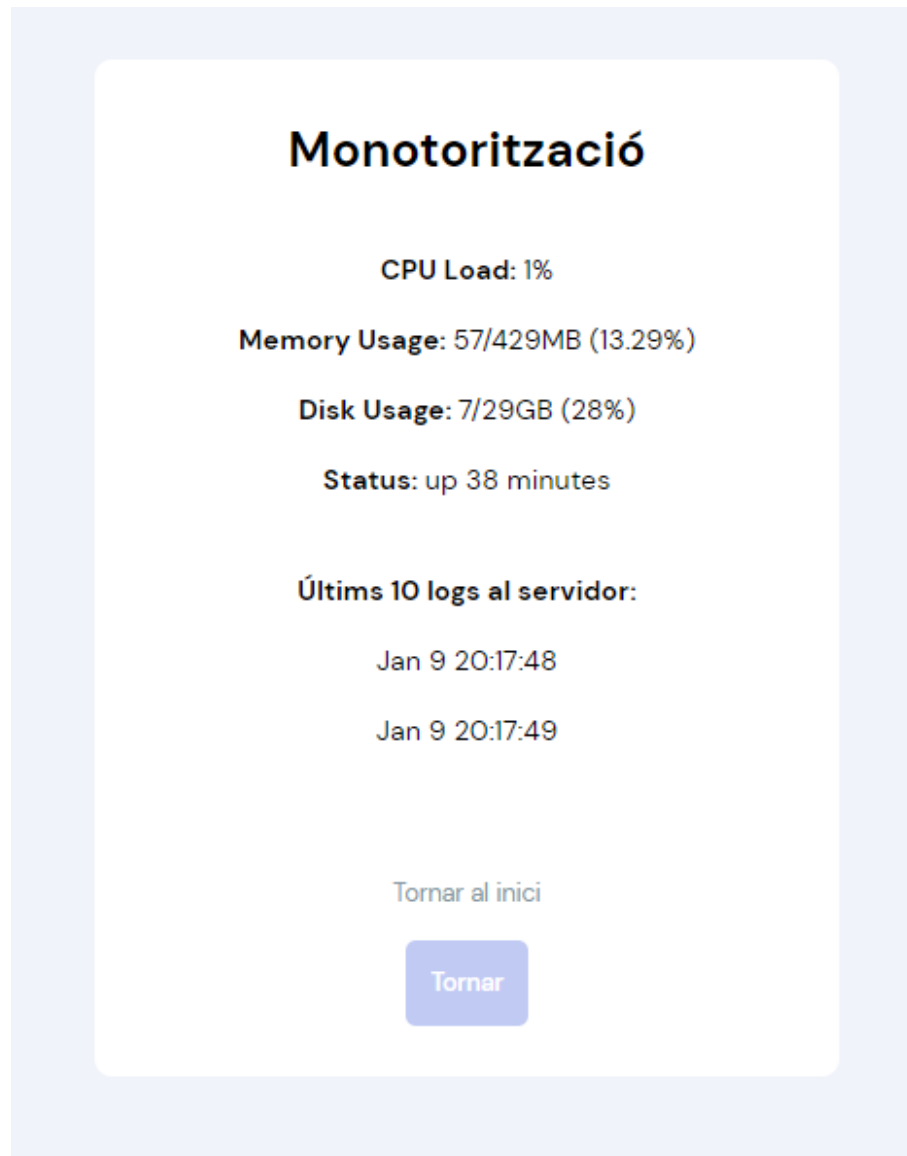
SYS\_LOGS: S'han mostrat la llista de reproducció de la música.

SYS\_LOGS: S'han fet shuffle a la llista de reproducció de la música.

SYS\_LOGS: S'han mostrat la llista de reproducció de la música.

SYS\_LOGS: S'ha fet replay/unreplay a la música.

## 4.8 Monitorització del sistema



## 4.9 Gestió de processos

### Gestió de Processos

Veure tots els processos

Mostra

---

Consulta estat procés

PID

---

Consulta

---

Elimina un procés

PID

---

Elimina

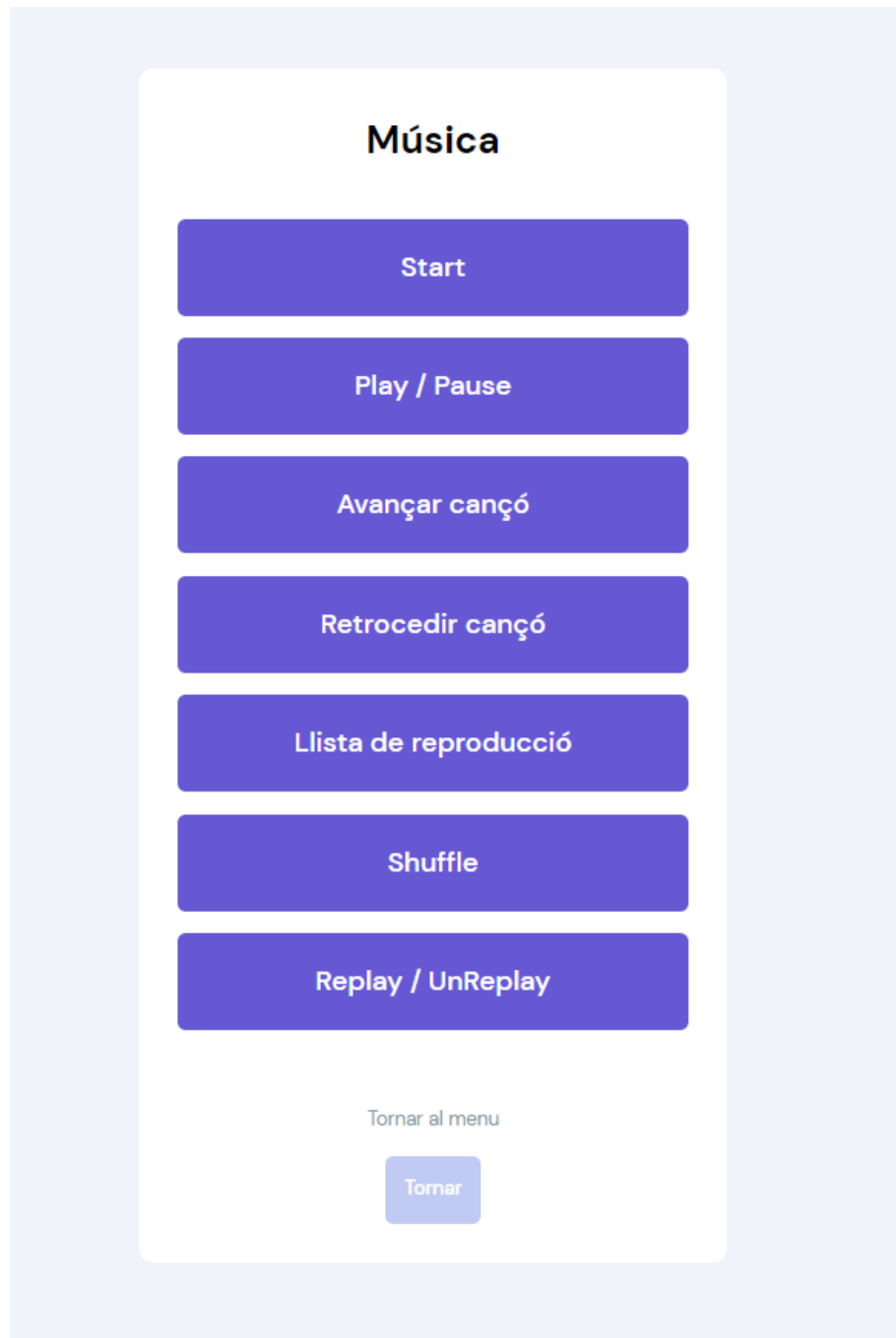
---

Elimina un procés en segons

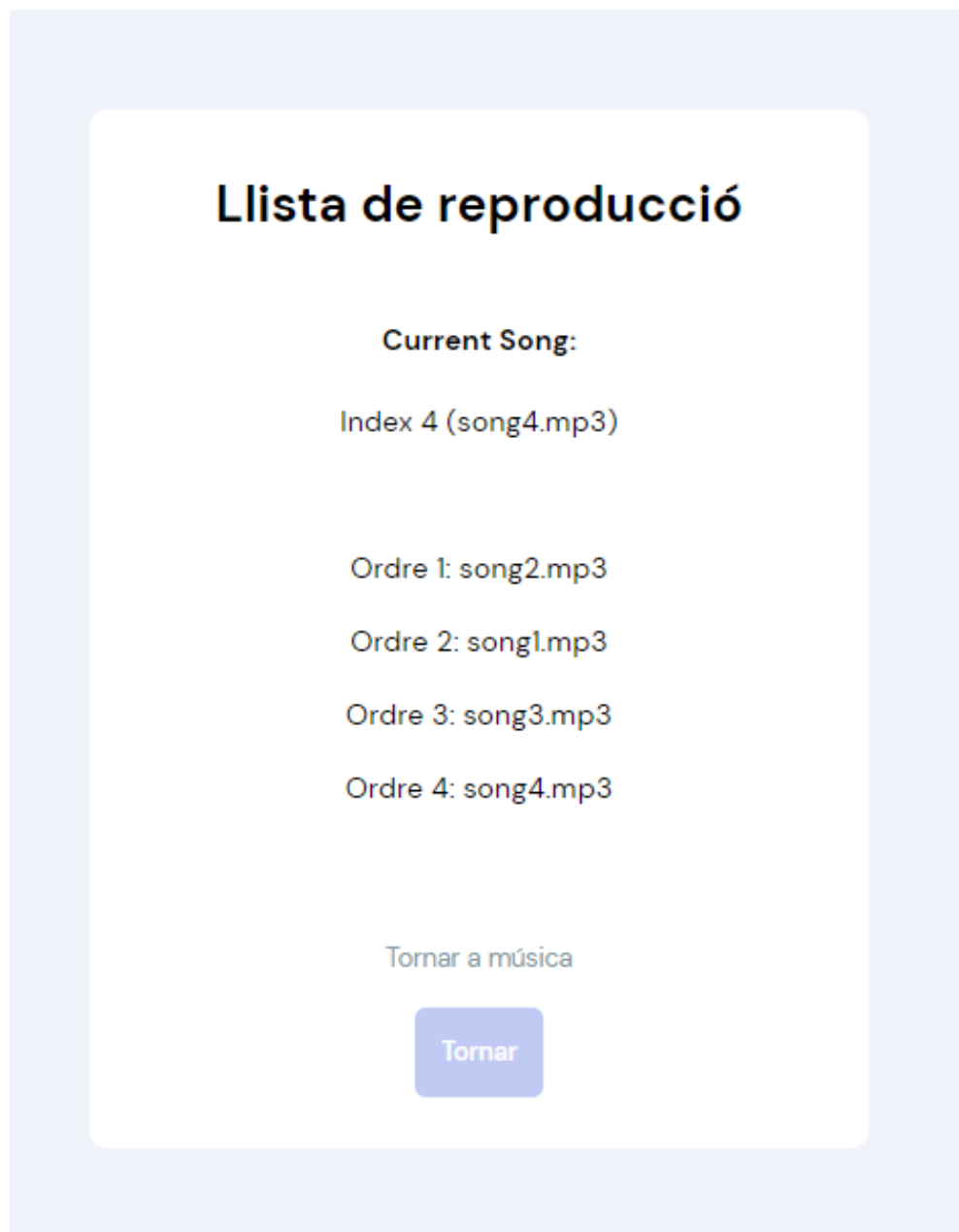
PID



#### 4.10 Funcionalitats musicals



#### 4.11 Llista de reproducció de la música



## 5 Scripts Bash i Codi LKM

### 5.1 Scripts per Menús

En aquesta secció es descriuran els scripts utilitzats per gestionar les diferents opcions que té l'usuari en la pàgina web.

#### 5.1.1 Index.sh

El script "index.sh" és el script encarregat de mostrar per HTML un formulari on es recolliran les dades (nom i contrasenya) per enviar al següent script "tryLogin.sh" que comprovarà si el usuari i la contrasenya introduïda són correctes a través de peticions GET. També es pot redirigir al script "signup.sh" per registrar-te en el sistema.

```
1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4      echo -e "
5      <!DOCTYPE html><html>
6          <head>
7              <meta charset="UTF-8">
8          </head>
9          <body>
10             <div class="container">
11                 <div class="card">
12                     <form action="LOGIN/tryLogin.sh" class="card-form">
13                         <div class="center"><h2 class="card-heading">Inici de sessió
14                         </h2></div>
15                         <div class="input">
16                             <input type="text" class="input-field" name="name"
17                             required/>
18                             <label class="input-label">Name</label>
19                         </div>
20                         <div class="input">
21                             <input type="password" class="input-field" name="password"
22                             required/>
23                             <label class="input-label">Password</label>
24                         </div>
25                         <div class="action">
26                             <input class="action-button" type="submit"
27                             value="Entrar">
28                         </div>
29                     </form>
30                     <div class="card-info">
31                         <p>Si no tens compte pots crear una nova</p>
32                         <div class="action">
33                             <form action="SIGNUP/signup.sh">
34                                 <input class="action-button2" type="submit"
35                                 value="Registre">
36                             </form>
37                         </div>
38                     </div>
39                 </div>
40             </div>
41         </body>
42     </html>"
```

### 5.1.2 Webapp.sh

El script "webapp.sh" és el script principal que ordena totes les possibles opcions que tenen els usuaris un cop han fet el login. Les diferents opcions que l'usuari pot clicar estan dividides en diferents forms d'HTML per executar cada script corresponent.

```
1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4      echo -e "
5      <!DOCTYPE html><html>
6          <head>
7              <meta charset="UTF-8">
8          </head>
9          <body>
10             <div class="container">
11                 <div class="card">
12                     <div class="center">
13                         <h2 class="card-heading">Raspberry Server</h2>
14                     </div>
15
16                     <form action="MONOTORITZACIO/monotoritzacio.sh" class="card-form">
17                         <div class="action">
18                             <input class="action-button" type="submit"
19                                 value="Monotorització">
20                         </div>
21                     </form>
22
23                     <form action="GESTIO_LOGS/gestiologs.sh" class="card-form">
24                         <div class="action">
25                             <input class="action-button" type="submit"
26                                 value="\Gestió de Logs">
27                         </div>
28                     </form>
29
30                     <form action="GESTIO_USUARIS/gestiousuaris.sh" class="card-form">
31                         <div class="action">
32                             <input class="action-button" type="submit"
33                                 value="\Gestió d'usuaris">
34                         </div>
35                     </form>
36
37                     <form action="GESTIO_PROCESSOS/gestioprocessos.sh"
38                         class="card-form">
39                         <div class="action">
40                             <input class="action-button" type="submit"
41                                 value="\Gestió de Processos">
42                         </div>
43                     </form>
44
45                     <form action="FILTRAT_PAQUETS/filtratpaquets.sh"
46                         class="card-form">
47                         <div class="action">
48                             <input class="action-button" type="submit"
49                                 value="\Filtrat de Paquets">
50                         </div>
51                     </form>
52
```

```

53         <form action="CRON/cron.sh" class="card-form">
54             <div class="action">
55                 <input class="action-button" type="submit"
56                 value="\Configuració Cron\">
57             </div>
58         </form>
59
60         <form action="MUSICA/musica.sh" class="card-form">
61             <div class="action">
62                 <input class="action-button" type="submit"
63                 value="Música">
64             </div>
65         </form>
66         </br>
67
68         <div class="flex-row">
69             <form action="APAGAR/apagar.sh" class="card-form">
70                 <div class="action">
71                     <input class="action-button3" type="submit"
72                     value="Apagar">
73                 </div>
74             </form>
75
76             <form action="REINICIAR/reiniciar.sh" class="card-form">
77                 <div class="action">
78                     <input class="action-button3" type="submit"
79                     value="Reiniciar">
80                 </div>
81             </form>
82         </div>
83
84         </br>
85
86         <div class="card-info">
87             <p>Tornar al inici</p>
88             <div class="action">
89                 <form action="index.sh">
90                     <input class="action-button2"
91                     type="submit" value="Tornar">
92                 </form>
93             </div>
94         </div>
95     </div>
96 </div>
97 </body>
98 </html>"

```

## 5.2 Scripts per Login

### 5.2.1 TryLogin.sh

A través de les peticions GET, aquest script farà split d'aquestes i es guardarà en variables el nom d'usuari i la contrasenya rebuda. Es comprovarà que el usuari existeixi en el sistema i que la contrasenya coincideixi amb la guardada a */etc/shadow*. Si alguna d'aquestes casuístiques falla, es mostrarà per la pàgina web el error pertinent.

```

1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4      echo -e "
5      <!DOCTYPE html><html>
6          <head>
7              <meta charset="UTF-8">
8          </head>
9          <body>
10             <div class="container">
11                 <div class="card">"
12
13      aux1=`echo $QUERY_STRING | awk -F= '{print $2}'`
14      name=`echo $aux1 | awk -F\& '{print $1}'`
15      password=`echo $QUERY_STRING | awk -F\= '{print $3}'`
16      pass2=$(sudo grep -w $name /etc/shadow | cut -d: -f2 2>&1)
17
18          if [ `id -u $name 2>/dev/null || echo -1` -ge 0 ];
19          then
20              if [ "$password" = "$pass2" ];
21              then
22                  echo "SYS_LOGS: Usuari $name ha fet login
23                  al sistema." >> /home/pi/Desktop/logs.txt
24                  echo -e "          <div class="center"><h2
25                  class="card-heading">Inici de sessió</h2>
26                      <p>El inici de sessió ha sigut
27                      satisfactori.</p></div>
28                      <form action="../webapp.sh"
29                      class="card-form">
30                          <div class="action">
31                              <input class="action-button"
32                              type="submit" value="Entrar">
33                          </div>
34                      </form>
35                      <div class="card-info">
36                          <p>Tornar al inici</p>
37                          <div class="action">
38                              <form action="../index.sh">
39                                  <input class="action-button2
40                                  " type="submit" value="Tornar">
41                              </form>
42                          </div>
43                      </div>"
44              else
45                  echo "SYS_LOGS: Intent de login al sistema amb
46                  nom: $name. Contrasenya fallida." >> /home/pi/Desktop/logs.txt
47                  echo -e "          <div class="center"><h2 class="card-heading">
48                  Inici de sessió</h2>
49                      <p>La contrasenya no coincideix amb el
50                      nom d'usuari.</p></div>
51                      <div class="card-info">
52                          <p>Tornar al inici</p>
53                          <div class="action">
54                              <form action="../index.sh">
55                                  <input class="action-button2"
56                                  type="submit" value="Tornar">
57                              </form>
58                      </div>

```

```

59                                     </div>"
60
61             fi
62         else
63             echo "SYS_LOGS: Intent de login al sistema amb nom:
64             $name. Usuari no registrat." >> /home/pi/Desktop/logs.txt
65             echo -e "                <div class='center'><h2 class='card-heading'>
66             Inici de sessió</h2>
67                                     <p>El usuari no existeix.</p></div>
68                                     <div class='card-info'>
69                                         <p>Tornar al inici</p>
70                                         <div class='action'>
71                                             <form action='../index.sh'>
72                                                 <input class='action-button2'
73                                                 type='submit' value='Tornar'>
74                                         </form>
75                                     </div>
76             fi
77             echo -e "
78                                     </div>
79             </div>
80         </body>
81     </html>"

```

## 5.3 Scripts per Signup

### 5.3.1 Signup.sh

El script "signup.sh" és el script encarregat de mostrar per HTML un formulari on es recolliran les dades (nom i contrasenya) per enviar al següent script "trySignup.sh" que comprovarà si el usuari pot ser guardat a través de peticions GET.

```

1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4      echo -e "
5      <!DOCTYPE html><html>
6          <head>
7              <meta charset='UTF-8'>
8          </head>
9          <body>
10             <div class='container'>
11                 <div class='card'>
12                     <form action='trySignup.sh' class='card-form'>
13                         <div class='center'><h2 class='card-heading'>
14                         Registre d'usuari</h2></div>
15                         <div class='input'>
16                             <input type='text' class='input-field'
17                             name='name' required/>
18                             <label class='input-label'>Name</label>
19                         </div>
20                         <div class='input'>
21                             <input type='password' class='input-field'
22                             name='password' required/>
23                             <label class='input-label'>Password</label>
24                         </div>
25                     <div class='action'>

```

```

26         <input class="action-button" type="submit"
27         value="Registre">
28     </div>
29 </form>
30 <div class="card-info">
31     <p>Si ja tens compte pots iniciar sessió</p>
32     <div class="action">
33         <form action=" ../index.sh">
34             <input class="action-button2" type="submit"
35             value="Inici">
36         </form>
37     </div>
38 </div>
39 </div>
40 </div>
41 </body>
42 </html>"

```

### 5.3.2 TrySignup.sh

A través de les peticions GET, aquest script farà split d'aquestes i es guardarà en variables el nom d'usuari i la contrasenya rebuda. Es comprovarà que el usuari no existeixi en el sistema i a continuació, a través de la comanda *useradd* es guardarà el nou usuari. Si el usuari ja existeix, es mostrarà un error per pantalla.

```

1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4  echo -e "
5  <!DOCTYPE html><html>
6      <head>
7          <meta charset="UTF-8">
8      </head>
9      <body>
10         <div class="container">
11             <div class="card">"
12
13         aux1=`echo $QUERY_STRING | awk -F= '{print $2}'`
14         name=`echo $aux1 | awk -F\& '{print $1}'`
15         password=`echo $QUERY_STRING | awk -F\= '{print $3}'`
16         try=$(sudo useradd -m $name -p $password 2>&1)
17
18         if [ -z $try ];
19         then
20             echo "SYS_LOGS: S'ha fet un signup correcte al sistema."
21             >> /home/pi/Desktop/logs.txt
22             echo -e "          <div class="center"><h2 class="card-heading">
23             Registre d'usuari</h2>
24                 <p>El registre del usuari ha sigut
25                 satisfactori.</p></div>
26                 <div class="card-info">
27                 <p>Tornar al inici</p>
28                 <div class="action">
29                     <form action=" ../index.sh">
30                         <input class="action-button2"
31                         type="submit" value="Tornar">

```



```

32                                     </form>
33                                     </div>
34                                     </div>"
35     else
36         echo "SYS_LOGS: S'ha fet un signup incorrecte al sistema."
37         >> /home/pi/Desktop/logs.txt
38         echo -e "                <div class=\"center\"><h2 class=\"card-heading\">
39         Registre d'usuari</h2>
40                 <p>Hi ha hagut un
41                 error en el registre del usuari.</p></div>
42                 <div class=\"card-info\">
43                     <p>Tornar al inici</p>
44                     <div class=\"action\">
45                         <form action=\"../index.sh\">
46                         <input class=\"action-button2\"
47                         type=\"submit\" value=\"Tornar\">
48                             </form>
49                     </div>
50                 </div>"
51     fi
52     echo -e "
53                                     </div>
54                                     </div>
55                                     </body>
56                                     </html>"

```

## 5.4 Scripts per Apagar

### 5.4.1 Apagar.sh

El següent script executarà la comanda `sudo shutdown now` que apaga el sistema en aquell mateix moment.

```

1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4
5      echo "SYS_LOGS: S'ha apagat el sistema." >> /home/pi/Desktop/logs.txt
6
7      echo -e "
8      <!DOCTYPE html><html>
9          <head>
10             <meta charset=\"UTF-8\">
11          </head>
12          <body>
13             <div class=\"container\">
14                 <div class=\"card\">
15                     <div class=\"center\">
16                         <h2 class=\"card-heading\">Apagar el servidor</h2>
17                     </div><div class=\"center\">
18                         <p>Apagant...</p>
19                         <br>
20                     </div>
21                     <div class=\"card-info\">
22                         <p>Tornar al inici</p>
23                         <div class=\"action\">
24                             <form action=\"../index.sh\">

```

```

25                                     <input class="action-button2"
26                                     type="submit" value="Tornar">
27                                     </form>
28                                     </div>
29                                 </div>
30                            </div>
31                        </div>
32                    </body>
33    </html>"
34
35    sudo shutdown now

```

## 5.5 Scripts per Reiniciar

### 5.5.1 Reiniciar.sh

El següent script executarà la comanda `sudo reboot` que reiniciarà el sistema.

```

1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4
5      echo "SYS_LOGS: S'ha reiniciat el sistema." >> /home/pi/Desktop/logs.txt
6
7      echo -e "
8      <!DOCTYPE html><html>
9          <head>
10             <meta charset="UTF-8">
11          </head>
12          <body>
13             <div class="container">
14                 <div class="card">
15                     <div class="center">
16                         <h2 class="card-heading">Reiniciar el servidor</h2>
17                     </div><div class="center">
18                         <p>Reiniciant...</p>
19                         <br>
20                     </div>
21                     <div class="card-info">
22                         <p>Tornar al inici</p>
23                         <div class="action">
24                             <form action=" ../index.sh">
25                                 <input class="action-button2"
26                                 type="submit" value="Tornar">
27                             </form>
28                         </div>
29                     </div>
30                 </div>
31             </div>
32          </body>
33    </html>"
34
35    sudo reboot

```

## 5.6 Scripts per Filtrar Paquets

### 5.6.1 Filter.sh

A través de les peticions GET, aquest script farà split d'aquestes i es guardarà en variables totes les regles obtingudes per filtrar. Com algunes regles poden ser buides, hem de comprovar-ho per cada una (amb sentències if else). Finalment per filtrar els paquets fem servir la comanda iptables amb els seus arguments corresponents depenent de les peticions GET.

```
1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  echo "SYS_LOGS: S'ha modificat un filtrat de paquets del sistema."
6      >> /home/pi/Desktop/logs.txt
7
8  aux1=`echo $QUERY_STRING | awk -F= '{print $2}'`
9  AoD=`echo $aux1 | awk -F\& '{print $1}'`
10
11 aux2=`echo $QUERY_STRING | awk -F\& '{print $2}'`
12 cad=`echo $aux2 | awk -F\= '{print $2}'`
13
14 aux3=`echo $QUERY_STRING | awk -F\& '{print $3}'`
15 regla=`echo $aux3 | awk -F\= '{print $2}'`
16
17 aux4=`echo $QUERY_STRING | awk -F\& '{print $4}'`
18 prot=`echo $aux4 | awk -F\= '{print $2}'`
19
20 aux5=`echo $QUERY_STRING | awk -F\& '{print $5}'`
21 src=`echo $aux5 | awk -F\= '{print $2}'`
22
23 aux6=`echo $QUERY_STRING | awk -F\& '{print $6}'`
24 sport=`echo $aux6 | awk -F\= '{print $2}'`
25
26 aux7=`echo $QUERY_STRING | awk -F\& '{print $7}'`
27 dest=`echo $aux7 | awk -F\= '{print $2}'`
28
29 dport=`echo $QUERY_STRING | awk -F\= '{print $9}'`
30
31 if [ -z "$prot" ];
32 then
33     protT=""
34 else
35     protT="-p ${prot}"
36 fi
37
38 if [ -z "$src" ];
39 then
40     srcT=""
41 else
42     srcT="-s ${src}"
43 fi
44
45 if [ -z "$dest" ];
46 then
47     destT=""
48 else
49     destT="-d ${dest}"
50 fi
```

```

51
52 if [ -z "$dport" ];
53 then
54     dportT=""
55 else
56     dportT="-dport ${dport}"
57 fi
58
59 if [ -z "$sport" ];
60 then
61     sportT=""
62 else
63     sportT="-sport ${sport}"
64 fi
65
66 iptable="sudo iptables ${AoD} ${cad} ${srcT} ${sportT} ${protT}
67     ${destT} ${dportT} -j ${regla}"
68 eval "$iptable"
69
70 echo -e "
71 <!DOCTYPE html><html>
72     <head>
73         <meta charset='UTF-8'>
74     </head>
75     <body>
76         <div class='container'>
77             <div class='card'>
78                 <div class='center'>
79                     <h2 class='card-heading'>Modificar un filtrat</h2>
80                 </div><div class='center'>
81                     <p>Filtrat de paquet afegit / modificat</p>
82                     <br>
83                 </div>
84                 <div class='card-info'>
85                     <p>Tornar al filtrat de paquets</p>
86                     <div class='action'>
87                         <form action='filtratpaquets.sh'>
88                             <input class='action-button2'
89                                 type='submit' value='Tornar'>
90                         </form>
91                     </div>
92                 </div>
93             </div>
94         </div>
95     </body>
96 </html>"

```

### 5.6.2 Filtratpaquets.sh

El script "filtratpaquets.sh" és el script encarregat de mostrar per HTML un formulari on es recolliran les dades per enviar al següent script "filter.sh" o bé veure tots els filtres de paquets al script "veurefiltrats.sh".

```

1 #!/bin/bash
2 echo Content-Type: text/html
3 echo
4 echo -e "

```

```

5 <!DOCTYPE html><html>
6   <head>
7     <meta charset="UTF-8">
8   </head>
9   <body>
10     <div class="container">
11       <div class="card">
12         <div class="center"><h2 class="card-heading">
13           Filtrat de Paquets</h2></div>
14
15         <div class="center"><h3>Veure tots els filtrats</h3></div>
16
17         <form action="veurefiltrats.sh" class="card-form">
18           <div class="action-top">
19             <input class="action-button" type="submit"
20               value="\Mostra\">
21           </div>
22         </form>
23
24         <br>
25         <br>
26         <hr>
27         <br>
28
29         <div class="center"><h3>Modificar un filtrat</h3></div>
30         <form action="filter.sh" class="card-form">
31           <div class="input">
32             <input type="text" class="input-field" name="AoD"/>
33             <label class="input-label">Afegir o Eliminar
34               (-A o -D)</label>
35           </div>
36
37           <div class="input">
38             <input type="text" class="input-field" name="cad"/>
39             <label class="input-label">Cadena (INPUT, OUTPUT
40               o FORWARD)</label>
41           </div>
42
43           <div class="input">
44             <input type="text" class="input-field" name="regla"/>
45             <label class="input-label">Regla (ACCEPT o DROP)</label>
46           </div>
47
48           <div class="input">
49             <input type="text" class="input-field" name="prot"/>
50             <label class="input-label">Protocol</label>
51           </div>
52
53           <div class="input">
54             <input type="text" class="input-field" name="src"/>
55             <label class="input-label">IP Origen</label>
56           </div>
57
58           <div class="input">
59             <input type="text" class="input-field" name="sport"/>
60             <label class="input-label">Port Origen</label>
61           </div>
62

```

```

63         <div class="input">
64             <input type="text" class="input-field" name="dest"/>
65             <label class="input-label">IP Destí</label>
66         </div>
67
68         <div class="input">
69             <input type="text" class="input-field" name="dport"/>
70             <label class="input-label">Port Destí</label>
71         </div>
72
73         <div class="action">
74             <input class="action-button" type="submit"
75                 value="Afegir">
76         </div>
77     </form>
78
79     </br>
80     </br>
81     <div class="card-info">
82         <p>Tornar al inici</p>
83         <div class="action">
84             <form action="../webapp.sh">
85                 <input class="action-button2"
86                     type="submit" value="Tornar">
87             </form>
88         </div>
89     </div>
90 </div>
91 </div>
92 </body>
93 </html>"

```

### 5.6.3 Veurefiltrats.sh

El script "veurefiltrats.sh" és el script encarregat de mostrar per tots els filtrats de paquets a través de la comanda `sudo iptables -S` que mostra totes les iptables existents al sistema.

```

1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  aux=`echo "$(sudo iptables -S)"`
6  echo "SYS_LOGS: S'han mostrat tots els filtrats del sistema."
7      >> /home/pi/Desktop/logs.txt
8
9  echo -e "
10 <!DOCTYPE html><html>
11     <head>
12         <meta charset="UTF-8">
13     </head>
14     <body>
15         <div class="container">
16             <div class="card">
17                 <div class="center">
18                     <h2 class="card-heading">Veure tots els filtrats</h2>
19                 </div><div class="center">"
20

```

```

21         while IFS="" read -r -a line;
22         do
23             echo "<p>$line</p>";
24         done <<< "$aux"
25
26     echo -e "                </br>
27                </div>
28                <div class="card-info">
29                    <p>Tornar a la gestió de filrats</p>
30                    <div class="action">
31                        <form action="filtratpaquets.sh">
32                            <input class="action-button2"
33                                type="submit" value="Tornar">
34                        </form>
35                    </div>
36                </div>
37            </div>
38        </div>
39    </body>
40 </html>"

```

## 5.7 Scripts per Gestió de Logs

### 5.7.1 Gestilogs.sh

El script “veurefiltrats.sh” és el script encarregat de mostrar per tots els logs que s’han fet en el sistema. Aquests logs estaran guardats en un fitxer (/home/pi/Desktop/logs.txt) i cada cop que un usuari executi una comanda al sistema es farà un append en aquest fitxer amb una descripció. Per exemple: echo "SYS\_LOGS: S'ha modificat un filtrat de paquets del sistema.»> /home/pi/Desktop/logs.txt.

```

1     #!/bin/bash
2     echo Content-Type: text/html
3     echo
4
5     echo -e "
6     <!DOCTYPE html><html>
7         <head>
8             <meta charset="UTF-8">
9         </head>
10        <body>
11            <div class="container">
12                <div class="card">
13                    <div class="center">
14                        <h2 class="card-heading">Gestió de Logs</h2>
15                    </div>
16
17                    while IFS="" read -r -a line;
18                    do
19                        echo "<p>$line</p>";
20                    done <<< "$(sudo cat "/home/pi/Desktop/logs.txt")"
21
22                    echo -e "</br>
23                    <div class="card-info">
24                        <p>Tornar al inici</p>
25                        <div class="action">
26                            <form action=" ../webapp.sh">
27                                <input class="action-button2"

```

```

28                                     type="submit" value="Tornar">
29                                     </form>
30                                 </div>
31                            </div>
32                        </div>
33                    </div>
34                </body>
35            </html>"

```

## 5.8 Scripts per Gestió d'Usuaris

### 5.8.1 Afegirusuari.sh

Aquest script actua igual que el script trySignup.sh.

```

1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4      echo -e "
5      <!DOCTYPE html><html>
6          <head>
7              <meta charset="UTF-8">
8          </head>
9          <body>
10             <div class="container">
11                 <div class="card">"
12
13                 aux1=`echo $QUERY_STRING | awk -F= '{print $2}'`
14                 name=`echo $aux1 | awk -F\& '{print $1}'`
15                 password=`echo $QUERY_STRING | awk -F\= '{print $3}'`
16                 try=$(sudo useradd -m $name -p $password 2>&1)
17
18                 if [ -z $try ];
19                 then
20                     echo "SYS_LOGS: Usuari $name afegit al sistema." >>
21                         /home/pi/Desktop/logs.txt
22                     echo -e "                <div class="center"><h2 class="card-heading">
23                         Afegir un usuari</h2>
24                                     <p>El registre del usuari ha sigut satisfactori
25                                     .</p></div>
26                                     <div class="card-info">
27                                         <p>Tornar al inici</p>
28                                         <div class="action">
29                                             <form action="gestiousuaris.sh">
30                                                 <input class="action-button2" type="submit"
31                                                 value="Tornar">
32                                                 </form>
33                                         </div>
34                                     </div>"
35                 else
36                     echo "SYS_LOGS: Intent de registre d'usuari fallit." >>
37                         /home/pi/Desktop/logs.txt
38                     echo -e "                <div class="center"><h2 class="card-heading">
39                         Afegir un usuari</h2>
40                                     <p>Hi ha hagut un error en el registre del
41                                     usuari.</p></div>
42                                     <div class="card-info">

```



```

43         <p>Tornar a la gestió d'usuaris</p>
44         <div class="action">
45             <form action="gestiousuaris.sh">
46                 <input class="action-button2"
47                     type="submit" value="Tornar">
48             </form>
49         </div>
50     </div>"
51 fi
52 echo -e "
53         </div>
54     </div>
55 </body>
56 </html>"

```

### 5.8.2 Eliminarusuari.sh

A través de les peticions GET, es guardarà el nom del usuari a eliminar per a cotnuació exectuar la següent comanda sudo userdel \$name i sudo rm -rf /home/\$name per eliminar el usuari del sistema i el seu directori. Es comprova, al igual forma que al registrar un usuari, que abans d'eliminar existeixi el usuari entrat.

```

1     #!/bin/bash
2     echo Content-Type: text/html
3     echo
4     echo -e "
5     <!DOCTYPE html><html>
6         <head>
7             <meta charset="UTF-8">
8         </head>
9         <body>
10             <div class="container">
11                 <div class="card">"
12
13                 aux1=`echo $QUERY_STRING | awk -F= '{print $2}'`
14                 name=`echo $aux1 | awk -F\& '{print $1}'`
15
16                 if [ `id -u $name 2>/dev/null || echo -1` -ge 0 ];
17                 then
18                     sudo userdel $name
19                     sudo rm -rf /home/$name
20                     echo "SYS_LOGS: Usuari $name eliminat del sistema."
21                     >> /home/pi/Desktop/logs.txt
22                     echo -e "         <div class="center"><h2 class="card-heading">
23                         Eliminar un usuari</h2>
24                     <p>El eliminat del usuari ha sigut
25                         satisfactori.</p></div>
26                         <div class="card-info">
27                             <p>Tornar al inici</p>
28                             <div class="action">
29                                 <form action="gestiousuaris.sh">
30                                     <input class="action-button2"
31                                         type="submit" value="Tornar">
32                                 </form>
33                             </div>
34                         </div>"

```

```

35         else
36             echo "SYS_LOGS: Intent d'eliminar usuari $name fallit."
37             >> /home/pi/Desktop/logs.txt
38             echo -e "          <div class='center'><h2 class='card-heading'>
39             Eliminar un usuari</h2>
40             <p>Hi ha hagut un error en el eliminat del usuari.
41             </p></div>
42             <div class='card-info'>
43             <p>Tornar a la gestió d'usuaris</p>
44             <div class='action'>
45                 <form action='gestiousuaris.sh'>
46                 <input class='action-button2'
47                 type='submit' value='Tornar'>
48                 </form>
49             </div>
50             </div>"
51         fi
52     echo -e "
53             </div>
54             </div>
55             </body>
56             </html>"

```

### 5.8.3 Gestiousuari.sh

El script "gestiousuari.sh" és el script encarregat de mostrar per HTML un formulari on es recolliran les dades per enviar al següents script "Afegirusuari.sh" o "Eliminarusuari.sh" o bé veure tots els usuaris existents del sistema al script "Veuretotsusuaris.sh".

```

1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4      echo -e "
5      <!DOCTYPE html><html>
6          <head>
7              <meta charset='UTF-8'>
8          </head>
9          <body>
10             <div class='container'>
11                 <div class='card'>
12                     <div class='center'><h2 class='card-heading'>
13                     Gestió d'usuaris</h2></div>
14
15                     <div class='center'><h3>Veure tots els usuaris</h3>
16                     </div>
17                     <form action='veuretotsusuaris.sh' class='card-form'>
18                         <div class='action-top'>
19                             <input class='action-button' type='submit'
20                             value='Mostra'>
21                         </div>
22                     </form>
23
24                     </br>
25                     </br>
26                     <hr>
27                     </br>
28

```

```

29         <div class="center"><h3>Afegir un usuari</h3></div>
30         <form action="afegirusuari.sh" class="card-form">
31             <div class="input">
32                 <input type="text" class="input-field"
33                     name="name" required/>
34                 <label class="input-label">Name</label>
35             </div>
36             <div class="input">
37                 <input type="password" class="input-field"
38                     name="password" required/>
39                 <label class="input-label">Password</label>
40             </div>
41             <div class="action">
42                 <input class="action-button" type="submit"
43                     value="Afegir">
44             </div>
45         </form>
46
47         </br>
48         </br>
49         <hr>
50         </br>
51
52         <div class="center"><h3>Eliminar un usuari</h3></div>
53         <form action="eliminarusuari.sh" class="card-form">
54             <div class="input">
55                 <input type="text" class="input-field"
56                     name="name" required/>
57                 <label class="input-label">Name</label>
58             </div>
59             <div class="action">
60                 <input class="action-button" type="submit"
61                     value="Eliminar">
62             </div>
63         </form>
64
65         </br>
66         </br>
67         <div class="card-info">
68             <p>Tornar al inici</p>
69             <div class="action">
70                 <form action=" ../webapp.sh">
71                     <input class="action-button2"
72                         type="submit" value="Tornar">
73                 </form>
74             </div>
75         </div>
76     </div>
77 </div>
78 </body>
79 </html>"
80

```

#### 5.8.4 Veure tots usuaris.sh

El script "Veure tots usuaris.sh" és el script encarregat de mostrar per tots els usuaris existents del sistema a través de la comanda `cut -d: -f1 /etc/passwd`.

```

1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4
5      users=`echo "$(cut -d: -f1 /etc/passwd)"`
6      echo "SYS_LOGS: S'han mostrat tots els usuaris del sistema."
7      >> /home/pi/Desktop/logs.txt
8
9      echo -e "
10     <!DOCTYPE html><html>
11         <head>
12             <meta charset="UTF-8">
13         </head>
14         <body>
15             <div class="container">
16                 <div class="card">
17                     <div class="center">
18                         <h2 class="card-heading">Veure tots els
19                         usuaris</h2>
20                     </div><div class="center">
21
22                         while IFS="" read -r -a line;
23                         do
24                             echo "<p>$line</p>";
25                         done <<< "$users"
26
27     echo -e "                 </br>
28                 </div>
29                 <div class="card-info">
30                     <p>Tornar a la gestió d'usuaris</p>
31                     <div class="action">
32                         <form action="gestiousuaris.sh">
33                             <input class="action-button2"
34                             type="submit" value="Tornar">
35                         </form>
36                     </div>
37                 </div>
38             </div>
39         </div>
40     </body>
41 </html>"

```

## 5.9 Scripts per Monotorització

### 5.9.1 Monotoritzacio.sh

El script "monotoritzacio.sh" és el script encarregat de mostrar tota la informació de CPU Usage, Memory Usage, Inicis de Sessió, Uptime i Disk Usage. En aquestes informació s'ha fet servir l'ús de la comanda awk per filtrar i sumar les columnes que ens interessaven depenent de la informació buscada.

```

1      #!/bin/bash
2      echo Content-Type: text/html
3      echo
4
5      cpuusage=`echo $[100-$(vmstat 1 2|tail -1|awk '{print $15}')]`%"`
6      memusage=`free -m | awk 'NR==2{printf "%s/%sMB (%.2f%%)\n", $3,$2,$3*100/$2 }'`

```

```

7 auxlogs=`echo "$(sudo cat /var/log/auth.log | grep
8 Accepted | awk 'NR==1,NR==10 {printf "%s %s %s\n", $1,$2,$3}')"`
9 uptime=`echo $(uptime -p)`
10 diskusage=`df -h | awk '$NF=="/" {printf "%d/%dGB (%s)\n", $3,$2,$5}'`
11 echo "SYS_LOGS: S'ha consultat la monitorització del sistema."
12 >> /home/pi/Desktop/logs.txt
13
14 echo -e "
15 <!DOCTYPE html><html>
16     <head>
17         <meta charset="UTF-8">
18     </head>
19     <body>
20         <div class="container">
21             <div class="card">
22                 <div class="center">
23                     <h2 class="card-heading">Monitorització</h2>
24                 </div>
25                 <div class="center">
26                     <p><h4>CPU Load: </h4>${cpuusage}</p>
27                 <p><h4>Memory Usage: </h4>${memusage}</p>
28                 <p><h4>Disk Usage: </h4>${diskusage}</p>
29                 <p><h4>Status: </h4>${uptime}</p>
30                 <br>
31                 <h4>Últims 10 logs al servidor:</h4>
32                 while IFS="" read -r -a line;
33                 do
34                     echo "<p>${line}</p>";
35                 done <<< "$auxlogs"
36
37                 echo -e "<br>"
38             </div>
39             <div class="card-info">
40                 <p>Tornar al inici</p>
41                 <div class="action">
42                     <form action=" ../webapp.sh">
43                         <input class="action-button2"
44                             type="submit" value="Tornar">
45                     </form>
46                 </div>
47             </div>
48         </div>
49     </div>
50 </body>
51 </html>"

```

## 5.10 Scripts per Gestió de Processos

### 5.10.1 Eliminarproces.sh

A través de les peticions GET, aquest script enviarà una senyal de kill al PID rebut per les peticions GET. Es comprova abans de fer-ho que aquest PID existeixi en el sistema.

```

1 #!/bin/bash
2 echo Content-Type: text/html
3 echo
4

```

```

5 aux1=`echo $QUERY_STRING | awk -F= '{print $2}'`
6 pid=`echo $aux1 | awk -F\& '{print $1}'`
7
8 echo -e "
9 <!DOCTYPE html><html>
10     <head>
11         <meta charset="UTF-8">
12     </head>
13     <body>
14         <div class="container">
15             <div class="card">
16                 <div class="center">
17                     <h2 class="card-heading">Elimina un procés</h2>
18                 </div><div class="center">
19
20                     if [ -n "$(ps -p $pid -o pid=)" ];
21                     then
22                         echo "SYS_LOGS: El Procés amb PID $pid
23                             s'ha eliminat del sistema." >> /home/pi/Desktop/logs.txt
24                         echo -e "<p>El Procés amb PID $pid s'ha
25                             eliminat del sistema.</p>"
26                         sudo kill -9 $pid
27                     else
28                         echo -e "<p>El Procés amb PID $pid no està running.</p>"
29                         echo "SYS_LOGS: Intent de fer kill al PID
30                             $pid." >> /home/pi/Desktop/logs.txt
31                     fi
32
33                 </div>
34             </div>
35             <div class="card-info">
36                 <p>Tornar a la gestió de processos</p>
37                 <div class="action">
38                     <form action="gestioprocessos.sh">
39                         <input class="action-button2"
40                             type="submit" value="Tornar">
41                     </form>
42                 </div>
43             </div>
44         </div>
45     </div>
46 </body>
47 </html>"

```

### 5.10.2 Eliminarprocessegons.sh

A través de les peticions GET, aquest script executarà en paral·lel "killInSeconds.sh" per enviar una senyal de kill al PID al cap de X segons rebut tot per les peticions GET. Es comprova abans de fer-ho que aquest PID existeixi en el sistema.

```

1 #!/bin/bash
2 echo Content-Type: text/html
3 echo
4
5 aux1=`echo $QUERY_STRING | awk -F= '{print $2}'`
6 pid=`echo $aux1 | awk -F\& '{print $1}'`

```

```

7 seconds=`echo $QUERY_STRING | awk -F\"= \" '{print $3}'`
8
9 echo -e "
10 <!DOCTYPE html><html>
11     <head>
12         <meta charset="UTF-8">
13     </head>
14     <body>
15         <div class="container">
16             <div class="card">
17                 <div class="center">
18                     <h2 class="card-heading">Elimina un procés
19                         en segons</h2>
20                 </div><div class="center">"
21
22                 if [ -n "$(ps -p $pid -o pid=)" ];
23                 then
24                     echo "SYS_LOGS: El Procés amb PID $pid s'elimina
25 del sistema en $seconds segons." >> /home/pi/Desktop/logs.txt
26                     echo -e "<p>El Procés amb PID $pid s'elimina del
27 sistema en $seconds segons.</p>"
28                     nohup bash killInSeconds.sh $seconds $pid >
29 /dev/null 2>&1 & disown
30                 else
31                     echo -e "<p>El Procés amb PID $pid no està running.</p>"
32                     echo "SYS_LOGS: Intent de fer kill al PID $pid."
33                     >> /home/pi/Desktop/logs.txt
34                 fi
35
36 echo -e "                </br>
37                </div>
38                <div class="card-info">
39                    <p>Tornar a la gestió de processos</p>
40                    <div class="action">
41                        <form action="gestioprocessos.sh">
42                            <input class="action-button2"
43                                type="submit" value="Tornar">
44                        </form>
45                    </div>
46                </div>
47            </div>
48        </div>
49    </body>
50 </html>"

```

### 5.10.3 Estatproces.sh

A través de les peticions GET, aquest script farà split d'aquestes i es guardarà en variables totes les regles obtingudes per filtrar. Com algunes regles poden ser buides, hem de comprovar-ho per cada una (amb sentències if else). Finalment per filtrar els paquets fem servir la comanda iptables amb els seus arguments corresponents depenent de les peticions GET.

```

1 #!/bin/bash
2 echo Content-Type: text/html
3 echo
4

```

```

5 echo "SYS_LOGS: S'ha consultat un procés del sistema."
6   >> /home/pi/Desktop/logs.txt
7 aux1=`echo $QUERY_STRING | awk -F= '{print $2}'`
8 pid=`echo $aux1 | awk -F\& '{print $1}'`
9 status=`echo "$(ps -q $pid -o state --no-headers)"`
10
11 echo -e "
12 <!DOCTYPE html><html>
13     <head>
14         <meta charset="UTF-8">
15     </head>
16     <body>
17         <div class="container">
18             <div class="card">
19                 <div class="center">
20                     <h2 class="card-heading">Consulta estat
21                         procés</h2>
22                 </div><div class="center">
23                     <p>Estat del procés consultat: $status</p>
24                     <br>
25                 </div>
26                 <div class="card-info">
27                     <p>Tornar a la gestió de processos</p>
28                     <div class="action">
29                         <form action="gestioprocessos.sh">
30                             <input class="action-button2"
31                                 type="submit" value="Tornar">
32                         </form>
33                     </div>
34                 </div>
35             </div>
36         </div>
37     </body>
38 </html>"

```

#### 5.10.4 Gestiprocessos.sh

El script "gestiprocessos.sh" és el script encarregat de mostrar per HTML un formulari on es recolliran les dades per enviar al següent script "Eliminarproces.sh" o "Eliminarprocessegons.sh" o "Estatproces.sh" o bé veure tots els processos existents del sistema al script "Veureprocessos.sh".

```

1 #!/bin/bash
2 echo Content-Type: text/html
3 echo
4 echo -e "
5 <!DOCTYPE html><html>
6     <head>
7         <meta charset="UTF-8">
8     </head>
9     <body>
10         <div class="container">
11             <div class="card">
12                 <div class="center"><h2 class="card-heading">
13                     Gestió de Processos</h2></div>
14
15                 <div class="center"><h3>Veure tots els processos
16                 </h3></div>

```



```

17         <form action="veureprocessos.sh" class="card-form">
18             <div class="action-top">
19                 <input class="action-button" type="submit"
20                     value="\Mostra\">
21             </div>
22         </form>
23
24     </br>
25 </br>
26 <hr>
27 </br>
28
29     <div class="center"><h3>Consulta estat procés
30 </h3></div>
31     <form action="estatproces.sh" class="card-form">
32         <div class="input">
33             <input type="text" class="input-field"
34                 name="name" required/>
35             <label class="input-label">PID</label>
36         </div>
37         <div class="action">
38             <input class="action-button" type="submit"
39                 value="Consulta">
40         </div>
41     </form>
42
43 </br>
44 </br>
45 <hr>
46 </br>
47
48     <div class="center"><h3>Elimina un procés</h3></div>
49     <form action="eliminarproces.sh" class="card-form">
50         <div class="input">
51             <input type="text" class="input-field"
52                 name="name" required/>
53             <label class="input-label">PID</label>
54         </div>
55         <div class="action">
56             <input class="action-button" type="submit"
57                 value="Elimina">
58         </div>
59     </form>
60
61 </br>
62 </br>
63 <hr>
64 </br>
65
66     <div class="center"><h3>Elimina un procés en
67 segons</h3></div>
68     <form action="eliminarprocessegons.sh" class="card-form">
69         <div class="input">
70             <input type="text" class="input-field"
71                 name="name" required/>
72             <label class="input-label">PID</label>
73         </div>
74         <div class="input">

```

```

75         <input type="text" class="input-field"
76         name="temps" required/>
77         <label class="input-label">Temps</label>
78     </div>
79     <div class="action">
80         <input class="action-button"
81         type="submit" value="Elimina">
82     </div>
83 </form>
84
85 </br>
86 </br>
87 <div class="card-info">
88     <p>Tornar al inici</p>
89     <div class="action">
90         <form action=" ../webapp.sh">
91             <input class="action-button2"
92             type="submit" value="Tornar">
93         </form>
94     </div>
95 </div>
96 </div>
97 </div>
98 </body>
99 </html>"

```

### 5.10.5 KillInSeconds.sh

Aquest script només executà una senyal kill al procés per paràmetre al cap de X segons.

```

1 #!/bin/bash
2 sleep $1 && sudo kill -9 $2

```

### 5.10.6 Veureprocessos.sh

El script "Veureprocessos.sh" és el script encarregat de mostrar per tots els processos existents del sistema a través de la comanda ps -aux.

```

1 #!/bin/bash
2 echo Content-Type: text/html
3 echo
4
5 echo "SYS_LOGS: S'han consultat tots els processos del sistema."
6     >> /home/pi/Desktop/logs.txt
7
8 echo -e "
9 <!DOCTYPE html><html>
10     <head>
11         <meta charset="UTF-8">
12     </head>
13     <body>
14         <div class="container">
15             <div class="card">
16                 <div class="center">
17                     <h2 class="card-heading">Veure tots els

```

```

18         processos</h2>
19     </div><table>"
20
21     while IFS="" read -r -a line;
22     do
23         echo -e "<tr>"
24         var=0
25         for word in $line
26         do
27             if [ "$var" -gt 10 ];
28             then
29                 echo -e " $word";
30             else
31                 if [ "$var" == 10 ];
32                 then
33                     echo -e "<td>$word";
34                 else
35                     echo -e "<td>$word</td>";
36                 fi
37             fi
38             var=$((var+1))
39         done
40         echo -e "</td></tr>"
41     done <<< "$(ps -aux)"
42
43     echo -e "</table><div class='card-info'"
44         <p>Tornar a la gestió de processos</p>
45         <div class='action'"
46             <form action='gestioprocessos.sh'"
47                 <input class='action-button2"
48                     type='submit' value='Tornar'"
49             </form>
50         </div>
51     </div>
52 </div>
53 </div>
54 </body>
55 </html>"

```

## 5.11 Scripts per Música

Aquest grup de scripts ha estat el que més problemes ens ha donat i més hores s'han dedicat en comparació a tots els altres. Un dels primers problemes que ens hem trobat a estat que tot i que podíem fer funcionar el so no sabíem si realment estava sonant ja que no teníem cap dispositiu amb que comprovar que l'àudio funcionés correctament. Aquest problema el vam adreçar utilitzant la verbosity del programa i veien el output del so podíem saber que estava funcionant correctament.

A continuació podíem fer funcionar tot el sistema amb una playlist fent la comanda `mpg123 *.mp3` i investigant per el man del programa hem vist que existien uns controls mitjançant tecles del teclat. La pròxima idea ha estat utilitzant un simulador de clics `xdotool` simular aquests clicks quan es cliquessin els diferents comandos i tecles. El resultat d'aquest procediment ha estat erroni ja que el programa `mpg123` no escolta aquests events. Investigant hem trobat que molts programes tenen com per defecte que no facin cas d'aquests events però que es pot desactivar aquesta mesura de protecció a la configuració del programa. En aquest cas `mpg123` no té un fitcher ni un atribut de configuració amb que activar aquesta possibilitat amb el que hem hagut de conseguir un altre camí a seguir.

La següent idea ha estat utilitzar els file descriptors que tindrà el procés, accedint a `/proc/<PID>/fd`. Aquesta idea tot i que introduïa el que volíem a dins del sistema no executava les comandes i no simulava com si s'haguessin premut aquelles tecles. Per tant tampoc era satisfactori.

Tornant al inici ja que les idees que teníem no estaven donant resultat hem tornat a llegir el man de el programa i ens hem adonat de un concepte interessant amb el atribut `-fifo` ja que permet assignar una pipe on enviar les coses i el programa podria llegir-ho com si fos el `stdin` normal del sistema. La complicació amb aquest mètode és que moltes de les funcionalitats com el `shuffle`, `next song`, `reverse song`... que ja venien per defecte les perdiem al estar en estat remot.

Per tant, finalment hem decidit executar `mpg123` en format remot enviar echos a la pipe per així controlar amb certes comandes (`LOADLIST` i `PAUSE`), además de controlar la cançó que sona a través d'un txt, al igual forma que la llista de reproducció amb un `.txt`. D'aquesta forma algunes funcionalitats venen donades en fer per exemple un `shuffle` a un fitxer o només deixar una línia a la llista de reproducció per fer `replay`.

### 5.11.1 Avansar.sh

Aquest script té com a objectiu avançar una cançó de la llista de reproducció. Per fer-ho, primer elimina totes les reproduccions que existeixen per després executar el script "reproduir.sh" amb argument 1 (en el seu apartat s'explicarà el funcionament d'aquest script).

```
1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  echo "SYS_LOGS: S'ha avançat una cançó de la llista de reproducció de la música."
6  >> /home/pi/Desktop/logs.txt
7
8  sudo kill -9 -f reproduir.sh
9  nohup bash reproduir.sh 1 > /dev/null 2>&1 & disown
10
11 echo -e "
12 <!DOCTYPE html><html>
13     <head>
14         <meta charset="UTF-8">
15     </head>
16     <body>
17         <div class="container">
18             <div class="card">
19                 <div class="center">
20                     <h2 class="card-heading">Avançar cançó</h2>
21                 </div>
22                 <div class="center">
23                     <p>S'ha avançat una cançó de la llista de reproducció de la música</p>
24                 </div>
25                 <br>
26
27                 <div class="card-info">
28                     <p>Tornar a música</p>
29                     <div class="action">
30                         <form action="musica.sh">
31                             <input class="action-button2"
32                                 type="submit" value="Tornar">
33                         </form>
34                     </div>
35                 </div>
36             </div>
37         </div>
38     </body>
39 </html>
```

```

36         </div>
37     </div>
38 </body>
39 </html>"

```

### 5.11.2 Llista.sh

Aquests cript té com a objectiu llistar totes les cançons de la llista de reproducció de la música del sistema. Per fer-ho es fa cat del fitxer /home/pi/Downloads/llista.txt que estan guardades les cançons (inicialment carregat amb el fitxer de configuració) i cat del fitxer /home/pi/Downloads/currentsong.txt on està la cançó que actualment està sonant per tenir un millor control del sistema.

```

1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  songs=`echo "$(cat /home/pi/Downloads/playlist3.m3u)"`
6
7  echo "SYS_LOGS: S'han mostrat la llista de reproducció de la música."
8  >> /home/pi/Desktop/logs.txt
9
10 echo -e "
11 <!DOCTYPE html><html>
12     <head>
13         <meta charset="UTF-8">
14     </head>
15     <body>
16         <div class="container">
17             <div class="card">
18                 <div class="center">
19                     <h2 class="card-heading">Llista de reproducció</h2>
20                 </div>
21                 <div class="center">
22                     </div><div class="center">"
23
24                 var=1
25                 songs=`cat /home/pi/Downloads/llista.txt`
26                 current=`cat /home/pi/Downloads/currentsong.txt`
27
28                 echo -e "<p><h4>Current Song: </h4> Index ${current}
29                     (song${current}.mp3)</p><br>";
30
31                 while IFS="" read -r -a line;
32                 do
33                     aux=($line)
34                     echo "<p>Ordre $var: ${aux[1]}</p>";
35                     var=$((var+1))
36                 done <<< "$songs"
37
38
39
40
41 echo -e "
42         </div></br>
43         <div class="card-info">
44             <p>Tornar a música</p>
45             <div class="action">

```

```

45                                     <form action="musica.sh">
46                                     <input class="action-button2"
47                                     type="submit" value="Tornar">
48                                     </form>
49                                 </div>
50                            </div>
51                        </div>
52                    </div>
53                </body>
54            </html>"

```

### 5.11.3 Musica.sh

El script "Musica.sh" és el script encarregat de mostrar per HTML un formulari on es recolliran les dades per enviar als següents script que gestionen totes les funcionalitats implementades de la música.

```

1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  echo -e "
6  <!DOCTYPE html><html>
7      <head>
8          <meta charset="UTF-8">
9      </head>
10     <body>
11         <div class="container">
12             <div class="card">
13                 <div class="center">
14                     <h2 class="card-heading">Música</h2>
15                 </div>
16
17                 <form action="start.sh" class="card-form">
18                     <div class="action">
19                         <input class="action-button" type="submit"
20                         value="\Start\">
21                     </div>
22                 </form>
23
24                 <form action="playpause.sh" class="card-form">
25                     <div class="action">
26                         <input class="action-button" type="submit"
27                         value="\Play / Pause\">
28                     </div>
29                 </form>
30
31                 <form action="avansar.sh" class="card-form">
32                     <div class="action">
33                         <input class="action-button" type="submit"
34                         value="\Avançar cançó\">
35                     </div>
36                 </form>
37
38                 <form action="retrocedir.sh" class="card-form">
39                     <div class="action">
40                         <input class="action-button" type="submit"
41                         value="\Retrocedir cançó\">

```

```

42         </div>
43     </form>
44
45     <form action="llista.sh" class="card-form">
46         <div class="action">
47             <input class="action-button" type="submit"
48                 value="\Llista de reproducció">
49         </div>
50     </form>
51
52     <form action="shuffle.sh" class="card-form">
53         <div class="action">
54             <input class="action-button" type="submit"
55                 value="\Shuffle">
56         </div>
57     </form>
58
59     <form action="replay.sh" class="card-form">
60         <div class="action">
61             <input class="action-button" type="submit"
62                 value="\Replay / UnReplay">
63         </div>
64     </form>
65
66     </br>
67
68     <div class="card-info">
69         <p>Tornar al menu</p>
70         <div class="action">
71             <form action=" ../webapp.sh">
72                 <input class="action-button2"
73                     type="submit" value="Tornar">
74             </form>
75         </div>
76     </div>
77 </div>
78 </div>
79 </body>
80 </html>"

```

#### 5.11.4 Playpause.sh

Aquest script té com a objectiu fer play o pause (depenent de l'estat anterior) de la llista de reproducció. Per fer-ho només cal enviar echo "pause» /home/pi/Downloads/pipe degut a com ho hem configurat inicialment en el start.sh.

```

1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  echo "pause" > /home/pi/Downloads/pipe
6  echo "SYS_LOGS: S'ha fet play/pause a la música."
7      >> /home/pi/Desktop/logs.txt
8
9  echo -e "
10 <!DOCTYPE html><html>

```

```

11     <head>
12         <meta charset="UTF-8">
13     </head>
14     <body>
15         <div class="container">
16             <div class="card">
17                 <div class="center">
18                     <h2 class="card-heading">Play / Pause</h2>
19                 </div>
20                 <div class="center">
21                     <p>Play / Pause realitzat a la cançó.</p>
22                 </div>
23                 <br>
24
25                 <div class="card-info">
26                     <p>Tornar a música</p>
27                     <div class="action">
28                         <form action="musica.sh">
29                             <input class="action-button2"
30                                 type="submit" value="Tornar">
31                         </form>
32                     </div>
33                 </div>
34             </div>
35         </div>
36     </body>
37 </html>"

```

### 5.11.5 Replay.sh

Aquest script té com a objectiu fixar la current song i repetir-la indefinidament. Per fer-ho es guarda temporalment la llista.txt en un fitxer /home/pi/Downloads/replay.txt i es substitueix el fitxer /home/pi/Downloads/llista.txt per la única cançó que estava sonant. Així es quedarà en bucle sonant (en replay) Si es volen desfer els canvis, només s'ha d'executar el script ja que, si el fitxer temporal /home/pi/Downloads/replay.txt no és buit, es desfaràn els canvis, carregant un altre cop tota la llista a /home/pi/Downloads/llista.txt.

```

1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  echo "SYS_LOGS: S'ha fet replay/unreplay a la música."
6      >> /home/pi/Desktop/logs.txt
7
8  if [ -f /home/pi/Downloads/replay.txt ];
9  then
10      cat /home/pi/Downloads/replay.txt > /home/pi/Downloads/llista.txt
11      sudo rm -f /home/pi/Downloads/replay.txt
12  else
13      sudo touch /home/pi/Downloads/replay.txt
14      sudo chmod 777 /home/pi/Downloads/replay.txt
15      cat /home/pi/Downloads/llista.txt > /home/pi/Downloads/replay.txt
16
17      var=0
18      current=`cat /home/pi/Downloads/currentsong.txt`
19      songs=`cat /home/pi/Downloads/llista.txt`

```



```

20 > /home/pi/Downloads/llista.txt
21 while IFS="" read -r -a line;
22 do
23     aux=($line)
24     if [ "${aux[0]}" != "current" ];
25     then
26         echo "${aux[0]} ${aux[1]}" > /home/pi/Downloads/llista.txt
27     fi
28 done <<< "$songs"
29 fi
30
31 echo -e "
32 <!DOCTYPE html><html>
33     <head>
34         <meta charset="UTF-8">
35     </head>
36     <body>
37         <div class="container">
38             <div class="card">
39                 <div class="center">
40                     <h2 class="card-heading">Replay / UnReplay</h2>
41                 </div>
42                 <div class="center">
43                     <p>Replay / UnReplay realitzat a la cançó.</p>
44                 </div>
45                 <br>
46                 <div class="card-info">
47                     <p>Tornar a música</p>
48                     <div class="action">
49                         <form action="musica.sh">
50                             <input class="action-button2"
51                                 type="submit" value="Tornar">
52                         </form>
53                     </div>
54                 </div>
55             </div>
56         </div>
57     </body>
58 </html>"
59

```

### 5.11.6 Reproduir.sh

Aquest script té com a objectiu reproduir la llista de reproducció gràcies a mpg123. Degut a la fifo configurada al script start.sh, per carregar una llista de reproducció només ens cal emprar una comanda molt senzilla: `echo "LOADLIST $index /home/pi/Downloads/playlist3.m3u" > /home/pi/Downloads/pipe`. Però, en aquest script es recullen tres formes de executar aquest LOADLIST al mpg123. Si no es passa cap argument (`$ -eq 0`) el script entendre que es comença a reproduir la música indefinidament en bucle (fent sleep de 60 segons) des del principi. Si es passa un paràmetre (`$ -eq 1`), el script interpreta que executarà la llista de reproducció avançant una posició de la current song. Finalment si existeixen dos paràmetres el script interpreta que executarà la llista de reproducció retrocedint una posició (menys quan es la primera posició). Per fer aquests algorismes ens hem basat en emprar comandes com while i pipes a fitxer per recorre els fitxers que ens indiquen la llista i la current song.

```

1 #!/bin/bash
2 sudo chmod 777 /home/pi/Downloads/pipe

```

```

3  if [ $# -eq 0 ];
4  then
5      while true
6      do
7          songs=`cat /home/pi/Downloads/llista.txt`
8          while IFS="" read -r -a line;
9              do
10                 aux=($line)
11                 echo "LOADLIST ${aux[0]}
12                     /home/pi/Downloads/playlist3.m3u" > /home/pi/Downloads/pipe
13                 echo "${aux[0]}" > /home/pi/Downloads/currentsong.txt
14                 sleep 60
15             done <<< "$songs"
16         done
17     else
18         # MÈTODE AVANÇAR
19         if [ $# -eq 1 ];
20         then
21             #sudo touch /home/pi/Downloads/prova.txt
22             #echo
23             bandera=0
24             songs=`cat /home/pi/Downloads/llista.txt`
25             current=`cat /home/pi/Downloads/currentsong.txt`
26             while IFS="" read -r -a line;
27                 do
28                     aux=($line)
29
30                     if [ "$bandera" -eq 1 ];
31                     then
32                         echo "LOADLIST ${aux[0]}
33                             /home/pi/Downloads/playlist3.m3u" >
34                             /home/pi/Downloads/pipe
35                         echo "${aux[0]}" > /home/pi/Downloads/currentsong.txt
36                         sleep 60
37                     else
38                         if [ "$current" == "${aux[0]}" ];
39                         then
40                             bandera=1
41                         fi
42                     fi
43                 done <<< "$songs"
44
45             #bucle infinit
46             songs=`cat /home/pi/Downloads/llista.txt`
47             while true
48             do
49                 songs=`cat /home/pi/Downloads/llista.txt`
50                 while IFS="" read -r -a line;
51                     do
52                         aux=($line)
53                         echo "LOADLIST ${aux[0]}
54                             /home/pi/Downloads/playlist3.m3u" >
55                             /home/pi/Downloads/pipe
56                         echo "${aux[0]}" > /home/pi/Downloads/currentsong.txt
57                         sleep 60
58                     done <<< "$songs"
59                 done
60             else

```

```

61      #METODE RETROCEDIR
62
63      #buscar abans canço
64      bandera=0
65      ite=0
66      songs=`cat /home/pi/Downloads/llista.txt`
67      current=`cat /home/pi/Downloads/currentsong.txt`
68      while IFS="" read -r -a line;
69      do
70          aux=($line)
71
72          if [ "$bandera" -neq 1 ];
73          then
74              if [ "$current" == "${aux[0]}" ];
75              then
76                  bandera=1
77              else
78                  ite=$((ite+1))
79              fi
80          fi
81      done <<< "$songs"
82
83
84      var=1
85      songs=`cat /home/pi/Downloads/llista.txt`
86      while IFS="" read -r -a line;
87      do
88          aux=($line)
89
90          if [ $var -ge $ite ];
91          then
92              echo "LOADLIST ${aux[0]}"
93                  /home/pi/Downloads/playlist3.m3u" >
94                  /home/pi/Downloads/pipe
95              echo "${aux[0]}" > /home/pi/Downloads/currentsong.txt
96              sleep 60
97          fi
98          var=$((var+1))
99      done <<< "$songs"
100
101
102      #bucle infinit
103      songs=`cat /home/pi/Downloads/llista.txt`
104      while true
105      do
106          songs=`cat /home/pi/Downloads/llista.txt`
107          while IFS="" read -r -a line;
108          do
109              aux=($line)
110              echo "LOADLIST ${aux[0]}"
111                  /home/pi/Downloads/playlist3.m3u"
112                  > /home/pi/Downloads/pipe
113              echo "${aux[0]}" > /home/pi/Downloads/currentsong.txt
114              sleep 60
115          done <<< "$songs"
116      done
117  fi
118

```

119

fi

### 5.11.7 Reproduir2.sh

Aquest script només el farem servir per iniciar la música. S'executa la comanda mpg123 de forma remota (-R) i amb una fifo ubicada sempre a /home/pi/Downloads/pipe. Fins que no arribi un LOADLIST estarà a l'espera sense sonar música.

```
1 #!/bin/bash
2 mpg123 -R --fifo /home/pi/Downloads/pipe
```

### 5.11.8 Retrocedir.sh

Aquest script té com a objectiu retrocedir una cançó de la llista de reproducció. Per fer-ho, primer elimina totes les reproduccions que existeixen per després executar el script "reproduir.sh" amb argument 1 1.

```
1 #!/bin/bash
2 echo Content-Type: text/html
3 echo
4
5 echo "SYS_LOGS: S'ha retrocedit una cançó de la llista de reproducció de la música."
6   >> /home/pi/Desktop/logs.txt
7
8 sudo pkill -9 -f reproduir.sh
9 nohup bash reproduir.sh 1 1 > /dev/null 2>&1 & disown
10
11 echo -e "
12 <!DOCTYPE html><html>
13     <head>
14         <meta charset="UTF-8">
15     </head>
16     <body>
17         <div class="container">
18             <div class="card">
19                 <div class="center">
20                     <h2 class="card-heading">Retrocedir
21                     cançó</h2>
22                 </div>
23                 <div class="center">
24                     <p>S'ha retrocedit una cançó de la llista de
25                     reproducció de la música</p>
26                 </div>
27                 <br>
28
29                 <div class="card-info">
30                     <p>Tornar a música</p>
31                     <div class="action">
32                         <form action="musica.sh">
33                             <input class="action-button2"
34                             type="submit" value="Tornar">
35                         </form>
36                     </div>
37                 </div>
38             </div>
```

```

39         </div>
40     </body>
41 </html>"

```

### 5.11.9 Shuffle.sh

Aquest script té com a objectiu reordenar de forma aleatòria la llista de reproducció de la música. Per fer-ho, en essència utilitzarem la comanda shuf /home/pi/Downloads/llista.txt que ens fa exactament el que volem.

```

1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  sudo touch /home/pi/Downloads/temp.txt
6  sudo chmod 777 /home/pi/Downloads/temp.txt
7  sudo shuf /home/pi/Downloads/llista.txt >
8      /home/pi/Downloads/temp.txt
9  cat /home/pi/Downloads/temp.txt > /home/pi/Downloads/llista.txt
10 sudo rm -f /home/pi/Downloads/temp.txt
11
12 echo "SYS_LOGS: S'han fet shuffle a la llista de reproducció
13     de la música." >> /home/pi/Desktop/logs.txt
14
15 echo -e "
16 <!DOCTYPE html><html>
17     <head>
18         <meta charset="UTF-8">
19     </head>
20     <body>
21         <div class="container">
22             <div class="card">
23                 <div class="center">
24                     <h2 class="card-heading">Shuffle</h2>
25                 </div>
26                 <div class="center">
27                     <p>Shuffle realitzat a la llista de reproducció.</p>
28                 </div>
29                 <br>
30
31                 <div class="card-info">
32                     <p>Tornar a música</p>
33                     <div class="action">
34                         <form action="musica.sh">
35                             <input class="action-button2"
36                                 type="submit" value="Tornar">
37                         </form>
38                     </div>
39                 </div>
40             </div>
41         </div>
42     </body>
43 </html>"

```

### 5.11.10 Start.sh

Aquest script té com a objectiu inicialitzar les comandes necessàries per executar la llista de reproducció. Per fer-ho s'eliminaran tots els processos anteriors que puguin donar problemes (mpg123 i reproduir.sh). Després executarem els scripts ja explicats "reproduir.sh" "reproduir2.sh" en paral·lel.

```
1  #!/bin/bash
2  echo Content-Type: text/html
3  echo
4
5  sudo kill -9 -f mpg123
6  sudo kill -9 -f reproduir.sh
7  nohup bash reproduir2.sh > /dev/null 2>&1 & disown
8  sudo chmod 777 /home/pi/Downloads/pipe
9  nohup bash reproduir.sh > /dev/null 2>&1 & disown
10
11  echo "SYS_LOGS: S'ha començat a reproduir la llista de
12      reproducció de la música." >> /home/pi/Desktop/logs.txt
13
14  echo -e "
15  <!DOCTYPE html><html>
16      <head>
17          <meta charset='UTF-8'>
18      </head>
19      <body>
20          <div class='container'>
21              <div class='card'>
22                  <div class='center'>
23                      <h2 class='card-heading'>Start</h2>
24                  </div>
25
26                  <div class='center'><p>S'ha començat a reproduir
27                      la llista de cançons.</p></div>
28
29                  <br>
30
31                  <div class='card-info'>
32                      <p>Tornar a música</p>
33                      <div class='action'>
34                          <form action='musica.sh'>
35                              <input class='action-button2'
36                                  type='submit' value='Tornar'>
37                          </form>
38                      </div>
39                  </div>
40              </div>
41          </div>
42      </body>
43  </html>"
```

### 5.11.11 A.sh

Aquest script, com diu l'enunciat, ha de controlar retrocedir una cançó. Per fer-ho es reutilitzen les funcionalitats ja implementades en els scripts anteriors.

```
1  #!/bin/bash
2  sudo kill -9 -f reproduir.sh
```

```
3 nohup bash /usr/lib/cgi-bin/MUSICA/reproduir.sh 1 1
4 > /dev/null 2>&1 & disown
```

### 5.11.12 B.sh

Aquest script, com diu l'enunciat, ha de controlar l'aturat de la música. Per fer-ho es reutilitzen les funcionalitats ja implementades en els scripts anteriors.

```
1 #!/bin/bash
2 sudo pkill -9 -f mpg123
3 sudo pkill -9 -f reproduir.sh
```

### 5.11.13 C.sh

Aquest script, com diu l'enunciat, ha de controlar la reproducció de la música. Per fer-ho es reutilitzen les funcionalitats ja implementades en els scripts anteriors.

```
1 #!/bin/bash
2
3 sudo pkill -9 -f mpg123
4 sudo pkill -9 -f reproduir.sh
5 nohup bash /usr/lib/cgi-bin/MUSICA/reproduir2.sh
6 > /dev/null 2>&1 & disown
7 sudo chmod 777 /home/pi/Downloads/pipe
8 nohup bash /usr/lib/cgi-bin/MUSICA/reproduir.sh
9 > /dev/null 2>&1 & disown
```

### 5.11.14 D.sh

Aquest script, com diu l'enunciat, ha de controlar avançar una cançó. Per fer-ho es reutilitzen les funcionalitats ja implementades en els scripts anteriors.

```
1 #!/bin/bash
2 sudo pkill -9 -f reproduir.sh
3 nohup bash /usr/lib/cgi-bin/MUSICA/reproduir.sh 1
4 > /dev/null 2>&1 & disown
```

## 6 Conclusions i Problemes Observats

---

Aquesta pràctica ens ha permès endinsar-nos a el ecosistema de linux tot treballant moltíssim amb bash i apache. Al inici del desenvolupament, sent de les primeres coses que vam realitzar va ser estructurar el sistema de fitxers que tindria el projecte. Aquesta decisió ens va anar molt bé de cara a poder treballar conjuntament tot cadascun tocant el seu propi script.

Una de les coses que hem trobat més a faltar ha estat tenir una raspberry més potent que ens permetés corre la instal·lació amb més rapidesa, ja que en el nostre cas tardava aproximadament 1 hora si fèiem el upgrade del sistema. Tot i que es una molt bona practica mantenir el upgrade per actualitzar tot el programari del sistema comentàvem comunament la línia per tal de evitar tenir aquests temps d'espera.

Una de les coses que hem trobat més a faltar ha estat permetre'ns utilitzar altres llenguatges de programació per simular la premuda de tecles per el mpg123. Donat que a la normativa de la pràctica no es permetia i que s'havia de utilitzar tot bash, ens ha complicat molt la feina i hem dedicat molt temps a aquesta part.

Un dels problemes que més treball ens han portat ha estat a causa del sistema de encoding de windows i linux, alhora de penjar els scripts a vegades ens fallaven per aquesta qüestió. Donat això vam afegir una línia a la instal·lació per poder tractar aquest error.

En quant a la musica hem trobat a faltar descobrir o comprar algun adaptador per poder visualitzar o sentir la musica per tal de poder debugar amb més certesa i facilitat. En aquest apartat també hem trobat a faltar una mini guia que expliqués com obtenir la capacitat de simular els clicks a el programa. Nosaltres vam estar intentant treballar amb xdotool tot i que vam concloure que no estava funcionant a causa que el procés detectava aquests events com a que no es tenien que tractar. Molts programes a el seu fitxer de configuració es pot modificar aquest comportament però en el cas de mpg123 no era el cas. Finalment vam aconseguir un funcionament molt semblant utilitzant la pipe que genera amb l'atribut -fifo.

Destacar que ens hem trobat molts cops que el mpg123 no estava funcionant com corresponia i a vegades fent el mateix si que actuava i bastants Segmentation faults que no hem aconseguit descobrir a que es causaven. Un altre treball que hem realitzat es permetre donant permisos a tots els usuaris córrer el mpg123 ja que al inici i per defecte no es permetia.