

Universidad de Cuenca

Informe sobre la estructura y funcionamiento del sistema de reserva de cubículos de la biblioteca del campus central de la Universidad de Cuenca

**Autor: Roger Aguirre
Luis González**

¹ Ingeniería Electrónica y Telecomunicaciones
14 de Julio de 2022

1 Introducción

El trabajo realizado consiste en desarrollar un sistema de comprobación de disponibilidad de los cubículos que se encuentran en la biblioteca del campus central de la Universidad de Cuenca, el cual se mostrara en monitores estratégicos para que los estudiantes puedan visualizar la disponibilidad de los cubículos.

2 Desarrollo

En la Ilustración 1 se muestra la variable que alberga el tiempo en segundos en que se van a ir mostrando cada pestaña que representa un cubículo, está en milisegundos, por defecto va a quedar en 10 segundos pero se lo puede cambiar a conveniencia, también está la variable que contiene el campus de la universidad, este debe estar en minúsculas para que se pueda extraer correctamente los datos del servidor.

```
// Tiempo que se muestra cada cubículo (milisegundos)
const intervalo = 500
// Nombre del Campus
const campus = "Campus Central"
```

Ilustración 1

En la ilustración 2 se define el usuario y contraseña, debe ser un usuario existente en el sistema Booked Scheduler.

```
// Usuario y contraseña
const credenciales = {
  username: "biblioteca@ucuenca.edu.ec",
  password: "biblioteca2020"
}
```

Ilustración 2

En la ilustración 3 se realiza la autenticación mediante una petición POST esta petición requiere de las credenciales y devuelve los datos de la sesión de los cuales se extraen las variables “sessionToken” y “userId”, que posteriormente se agregan al “localStorage” para mantener activa la sesión.

```
// Método POST para autenticación
const postAutenticacion = (credenciales) => {
  try {
    const res = await fetch('http://10.22.114.5/booked/Web/Services/index.php/Authentication/Authenticate', {
      method: "POST",
      body: JSON.stringify(credenciales),
      headers: { "Content-type": "application/json; charset=UTF-8" }
    })
    const data = await res.json()
    return data
  } catch (error) {
    console.log(error)
  }
}
```

Ilustración 3

En la Ilustración 4 se muestra como se obtienen los cubículos, se hace mediante una petición GET en la función “getCubiculos”. Esta petición requiere los headers “X-Booked-SessionToken” y “X-Booked-UserId”, que deben contener las variables “sessionToken” y “userId” anteriormente almacenadas en “localStorage”.

```
// Método GET para obtener todos los cubículos
const getCubiculos = async () => {
  try {
    const res = await fetch('http://10.22.114.5/booked/Web/Services/index.php/Resources', {
      method: "GET",
      headers: {
        "Content-type": "application/json; charset=UTF-8",
        "X-Booked-SessionToken": sessionToken,
        "X-Booked-UserId": userId
      }
    })
    const data = await res.json()
    return data
  } catch (error) {
    console.log(error)
  }
}
```

Ilustración 4

En la Ilustración 5 se obtienen las reservaciones, esto se hace mediante una petición GET en la función “getReservaciones”, esta petición también requiere los headers “X-Booked-SessionToken” y “X-Booked-UserId”, que deben contener las variables “sessionToken” y “userId” anteriormente almacenadas en “localStorage”.

```
// Método GET para obtener todas las reservaciones
const getReservaciones = async () => {
  try {
    const res = await fetch('http://10.22.114.5/booked/Web/Services/index.php/Reservations', {
      method: "GET",
      headers: {
        "Content-type": "application/json; charset=UTF-8",
        "X-Booked-SessionToken": sessionToken,
        "X-Booked-UserId": userId
      }
    })
    const data = await res.json()
    return data
  } catch (error) {
    console.log(error)
  }
}
```

Ilustración 5

En las siguientes ilustraciones se muestra la interfaz gráfica del sistema, cada pestaña representa un cubículo y van a ir pasando en un bucle cada tiempo que se haya especificado en el código. En el se visualiza de lado derecho el nombre del cubículo, el piso en el que se encuentra el cubículo y el estado de disposición el cual puede ser “OCUPADO” o “LIBRE”, dependiendo del estado cambiara su color de rojo a verde respectivamente. En el lado izquierdo se muestra todas las reservaciones que se tienen en el respectivo cubículo, las letras de color rojo representa que el cubículo está ocupado en ese momento, a continuación se muestran diferentes estados de la interfaz.

En la ilustración 6 se muestra que el cubículo es DEWEY que está en el 3er piso y se encuentra ocupado de 16:00 a 19:00 por eso están las letras en color rojo, también se puede ver que existen otras reservaciones.

SISTEMA DE CUBÍCULOS

DEWEY
CAMPUS CENTRAL 3ER PISO

OCUPADO

RESERVAS

	Inicio	Fin
DEWEY	Viernes 8 Julio 14:00	Viernes 8 Julio 16:00
SICED	Viernes 8 Julio 16:00	Viernes 8 Julio 19:00
VUJR		
AMANUENSE		
COLOFÓN		
TESAURO		
MISCELANEA		
SIGNATURA		

Ilustración 6

En la Ilustración 7 se puede ver que el cubículo es AMANUENSE se encuentra en el cuarto piso, su estado es libre en el horario actual por eso se encuentra de color verde, también se puede observar que existe una reservación de 13:00 a 16:00.

SISTEMA DE CUBÍCULOS

AMANUENSE
CAMPUS CENTRAL 4TO PISO

LIBRE

RESERVAS

	Inicio	Fin
DEWEY	Viernes 8 Julio 13:00	Viernes 8 Julio 16:00
SICED		
VUJR		
AMANUENSE		
COLOFÓN		
TESAURO		
MISCELANEA		
SIGNATURA		

Ilustración 7

En la Ilustración 8 se muestra el cubículo VLIR que se encuentra en el 3er piso, su estado es libre por lo tanto esta de color verde, en el lado derecho se muestra que no existe ninguna reservación.



Ilustración 8

Raspberrypi 3

Para ponerla en funcionamiento se le instaló el sistema operativo Raspbian mediante NOOBS (New Out of Box Software) para realizar la instalación primero hay que formatear la tarjeta micro SD, se lo realizó con el programa SDFormatter:

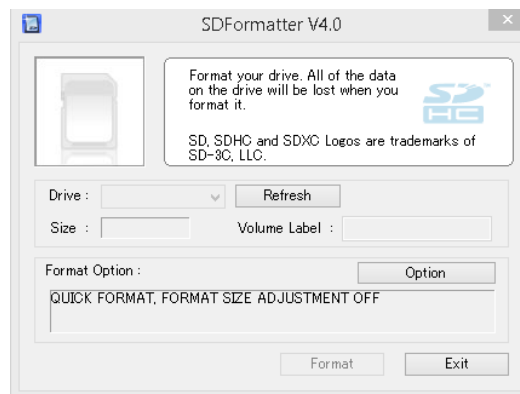


Ilustración 9

Para descargar NOOBS se puede hacer desde el siguiente link: <https://github.com/raspberrypi/noobs>, este link nos llevará a descargar la versión más reciente, también se pueden descargar otras versiones. Después todos los archivos que se descargaron los copiamos y pegamos en la micro SD y procedemos a encender la raspberrypi y seleccionamos Raspbian.

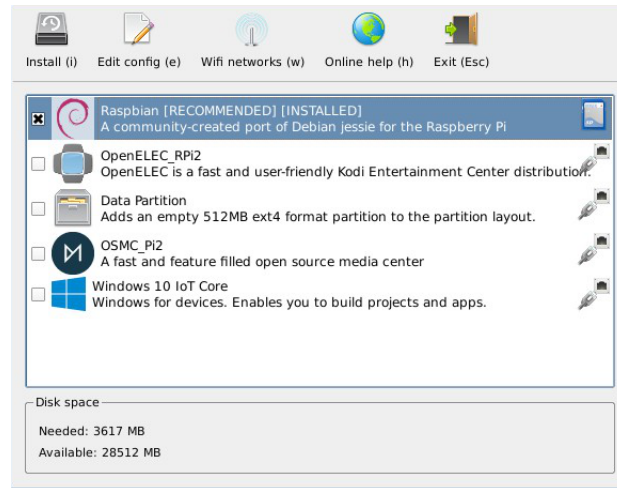


Ilustración 10

Hacer automático el inicio del sistema en la Raspberry pi

Para que el sistema de reserva de cubículos se inicie al encender el dispositivo Raspberry pi, el primer paso es crear en el escritorio un archivo bash que tiene extensión .sh, en este caso se lo llamo biblio.sh. A este archivo se le deben dar permisos de ejecución, para esto se introduce en la terminal el siguiente comando:

```
sudo chmod +x Desktop/biblio.sh
```

Lo siguiente es escribir el código de biblio.sh el cual abrirá en pantalla completa el sistema de reserva de cubículos además tiene otros comandos para que no pase ningún error por pantalla, tampoco ningún aviso como el de restaurar las páginas en caso de que se apague intempestivamente la Raspberrypi el cual es el siguiente:

```
Chromium-browser --noerrors --disable-session-crashedbubble --start-fullscreen --incognito http://10.22.114.5/reserva-biblioteca-ucuenca
```

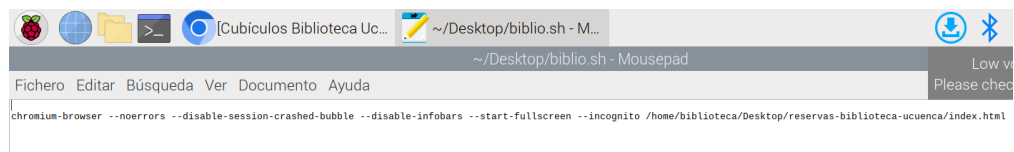
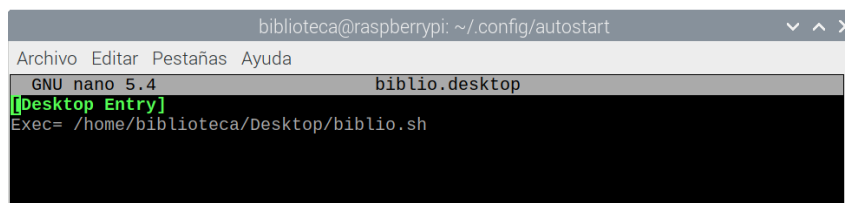


Ilustración 11

Después para que el programa biblio.sh se ejecute al encender la Raspberry hay que ir a la ruta **/home/pi/.config/** en ella hay que crear una carpeta, en este caso se la llamo “autostart”. En esa carpeta hay que hacer un archivo con extensión .desktop en este caso se lo llamo “biblio.desktop” en ella se le va a ingresar el siguiente código:



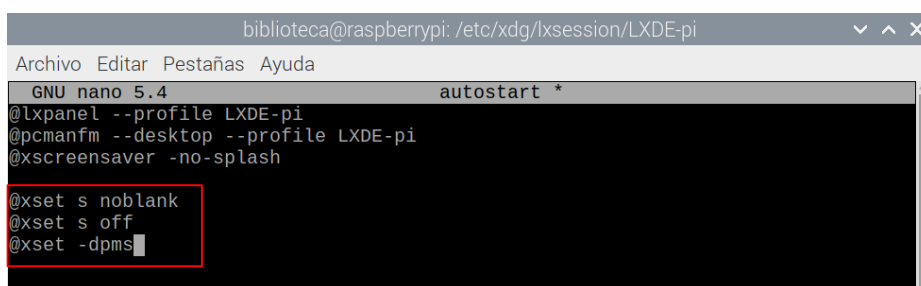
```

biblioteca@raspberrypi: ~/config/autostart
GNU nano 5.4 biblio.desktop
[Desktop Entry]
Exec= /home/biblioteca/Desktop/biblio.sh

```

Ilustración 12

La Raspberry viene por defecto que cuando no se está realizando ninguna tarea se va a suspender sola luego de cierto tiempo. Para evitar que esto pase se debe dirigir a la siguiente ruta **sudo nano /etc/xdg/lxsession/LXDE-pi/autostart** y agregarle los siguientes comandos señalados en un cuadro:



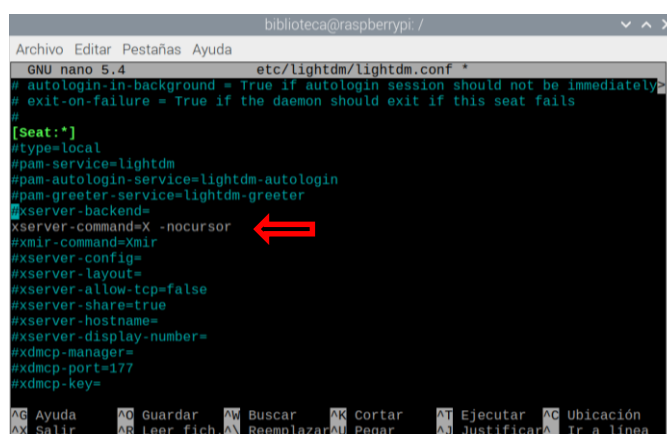
```

biblioteca@raspberrypi: /etc/xdg/lxsession/LXDE-pi
GNU nano 5.4 autostart *
@lxpanel --profile LXDE-pi
@pcmanfm --desktop --profile LXDE-pi
@xscreensaver -no-splash
@xset s noblank
@xset s off
@xset -dpms

```

Ilustración 13

Una vez que se hayan hecho todas las comprobaciones de que funciona todo correctamente, se puede esconder el cursor del mouse que puede estar interfiriendo, para que esto no suceda hay que dirigirse a la ruta siguiente y editar el texto descomentando y agregándole la parte de “-nocursor”, en este caso se realizó con “nano”: **sudo nano /etc/lightdm/lightdm.conf**, de esta manera al reiniciar la Raspberrypi ya no se podrá ver el cursor.



```

biblioteca@raspberrypi: /
GNU nano 5.4 etc/lightdm/lightdm.conf *
# autologin-in-background = True if autologin session should not be immediately
# exit-on-failure = True if the daemon should exit if this seat fails
#
[Seat:*]
#type=local
#pam-service=lightdm
#pam-autologin-service=lightdm-autologin
#pam-greeter-service=lightdm-greeter
xserver-backend=
xserver-command=X -nocursor
xmirc-command=Xmir
xserver-config=
xserver-layout=
xserver-allow-tcp=false
xserver-share=true
xserver-hostname=
xserver-display-number=
xdmcp-manager=
xdmcp-port=177
xdmcp-key=

```

Ilustración 14

