Universidad Politécnica de Chiapas

Ingeniería en Desarrollo de Software

Redes

José Alonso Macias Montoya

Proyecto final – C3

Orozco Reyes Marcos Iván – 203413

Rios Mena Gustavo Vladimir - 203450

García Morales Yurandir - 201239

Proyecto integrador

Invernadero

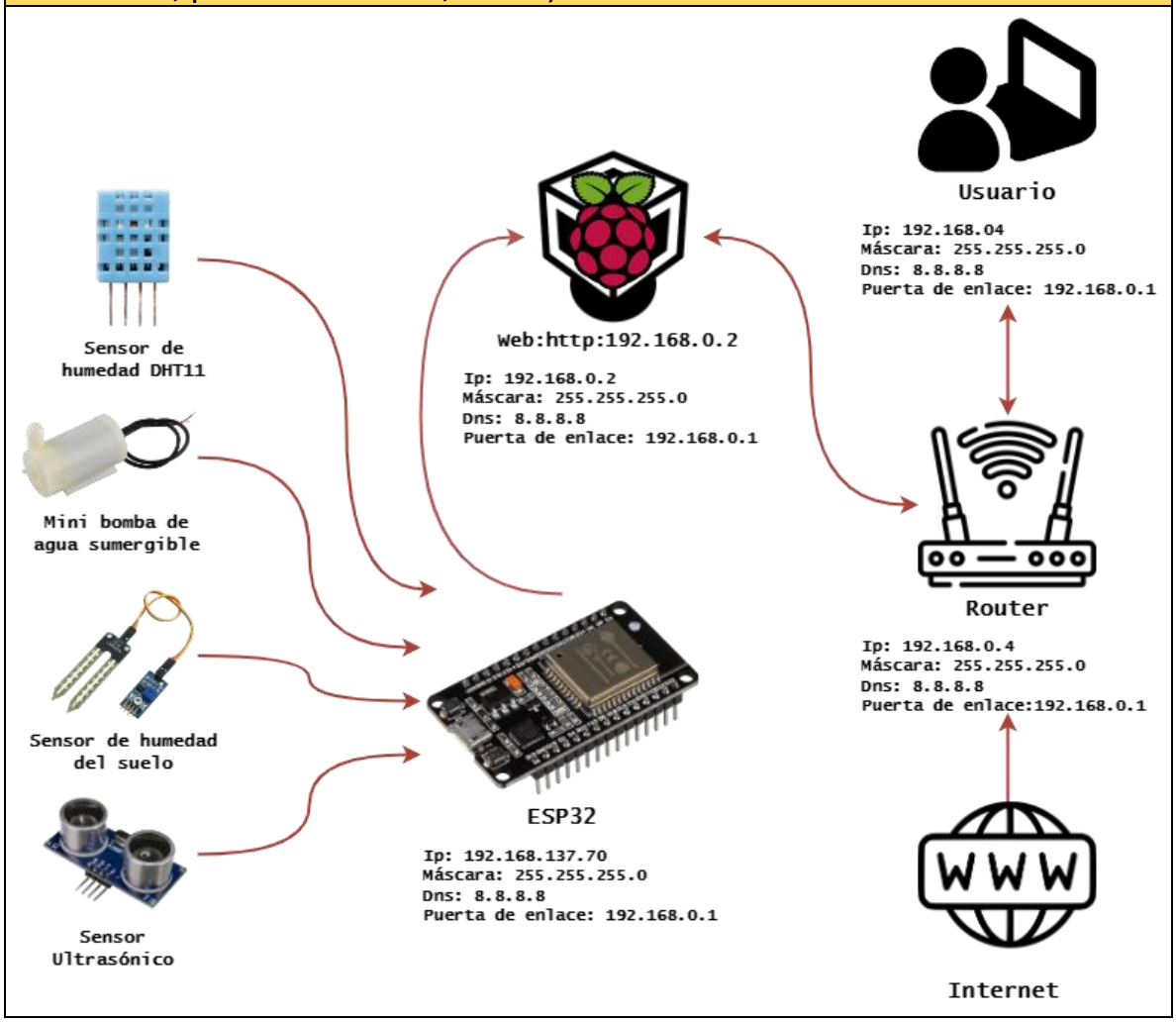
25 de junio de 2022

Introducción

En el siguiente reporte, se mostrarán los puntos o checklist de los requisitos a cumplir del proyecto integrador, relacionados a la materia de Redes. Los puntos a tratar son con respecto a la creación del servidor web, donde se está la base de datos, el back-end y el front-end alojado en los sitios disponibles de Apache. Aparte, estos puntos también se relacionan con la seguridad del sitio y la estructura de redes, es sumamente importante cumplir con estos puntos, para empezar a desarrollar buenas prácticas de desarrollo y ciberseguridad. A continuación, se mostrará la checklist con cada punto justificado y con capturas de pantalla que lo respalda.

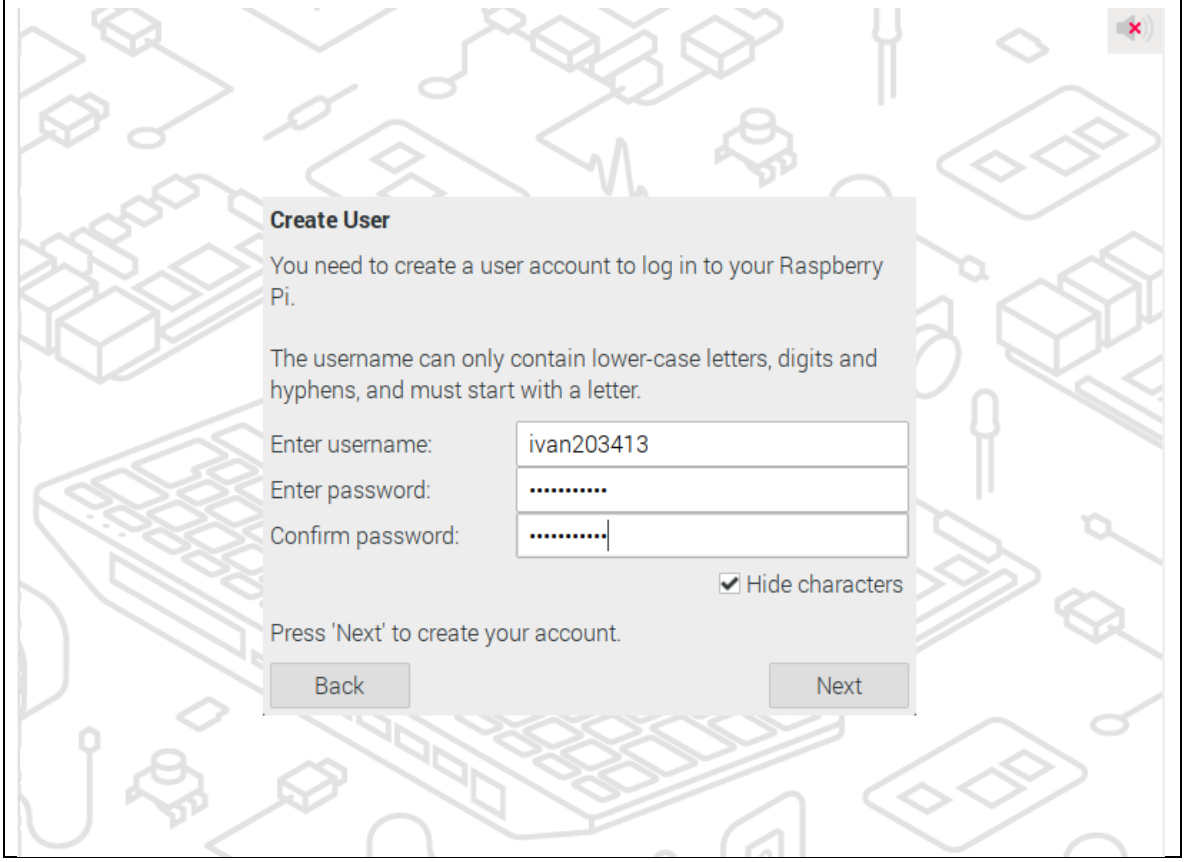
Link a la documentación completa (resto de materias):
https://github.com/203413/CarpetaDeEvidencias_ProyectoIntegrador

1. **Diagrama de red que incluye direccionamiento** (IP fija, máscara de red, puerta de enlace, DNS's)



2. **Crear nueva cuenta de usuario con permisos de administrador** y **eliminar cuenta por defecto.**

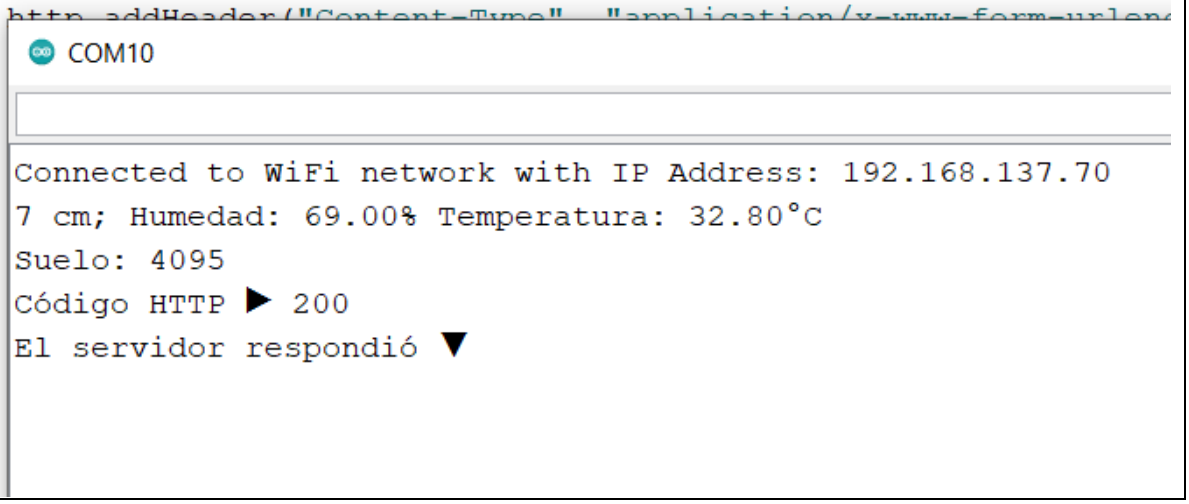
Originalmente se iba a trabajar con raspberries físicas, pero gracias a varios problemas técnicos, se decidió usar la virtualización, llamada “Raspberry desktop” (también llamado “Rasbian” ya que está basado en Debian). En la versión más reciente de Raspberry desktop, el usuario “pi” es eliminado y cambiado por uno creado por el usuario en la instalación. Así que el borrar el usuario por defecto y crear uno nuevo se puede omitir.



3. Justificación del uso del WIFI

El WIFI en el proyecto integrador es de suma importancia, ya que por este medio se reciben los datos enviados por medio de peticiones HTTP por el esp32, además de permitirnos instalar herramientas para el proyecto.
Gracias a la librería **<WIFI.H>** el esp32 se puede conectar a internet. En el código del esp32 le indicamos que una vez que se conecte, muestre la ip que se le ha asignado.
Una vez conectado a internet, con la librería **<HTTPClient.h>** puedes hacer peticiones HTTP con el esp32, y en este caso, le hace petición al servidor web para alojar los datos recolectados a la base de datos.

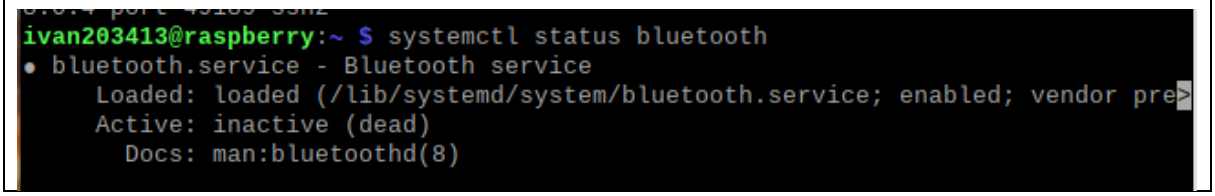
```
http.begin("http://192.168.0.4:3001/api/datos");
```



4. Se ha desactivado bluetooth

El servicio de bluetooth fue desactivado, gracias al comando: **sudo rfkill block bluetooth.** Ya que, aparte de verlo como no necesario para el proyecto, abre menos posibilidad a ataques y menos vulnerabilidad.

Con el comando **systemctl status bluetooth** se revisa el status del servicio de bluetooth, este dice que está inactivo.



5. Configuración de UFW con reglas de acceso a servicios

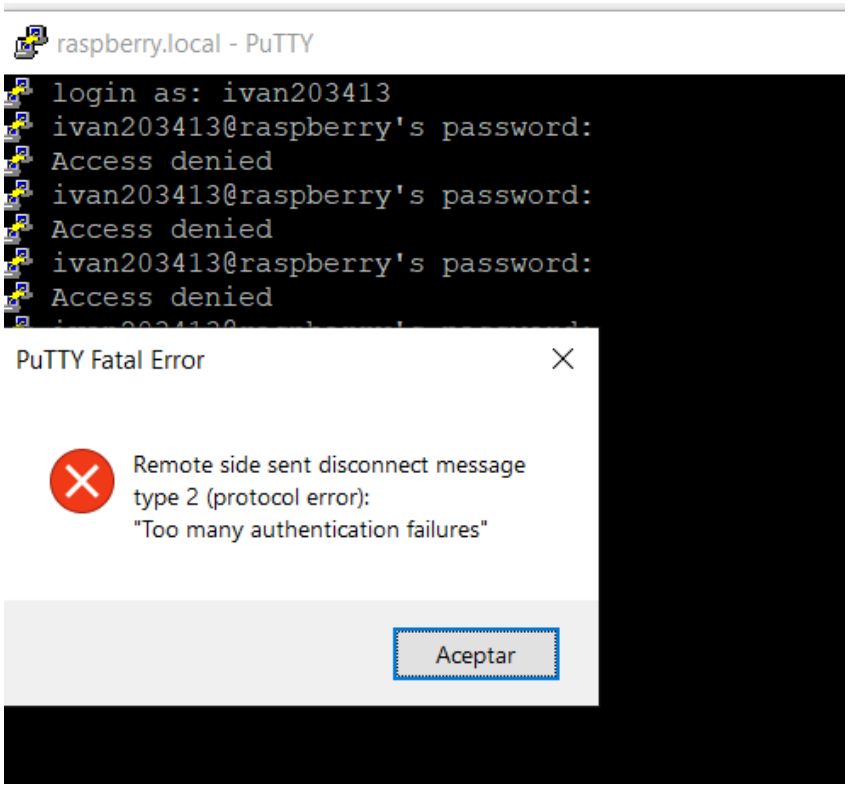
Una vez actualizado el sistema, se instalará el firewall UFW, con el típico comando de instalación: sudo apt-get install UFW
Con el programa ya instalado, se procede a hacer dos simples configuraciones: permitir el puesto 22 (dedicado al SSH) y el puerto 80 (dedicado a web).

```
 route insert NUM RULE          insert route RULE at NUM
 reload                         reload firewall
 reset                          reset firewall
 status                         show firewall status
 status numbered                show firewall status as numbered list of RULES
 status verbose                 show verbose firewall status
 show ARG                       show firewall report
 version                        display version information

Application profile commands:
 app list                       list application profiles
 app info PROFILE               show information on PROFILE
 app update PROFILE             update PROFILE
 app default ARG                set default application policy

ivan203413@raspberry:~ $ ufw allow ssh
ERROR: You need to be root to run this script
ivan203413@raspberry:~ $ sudo ufw allow ssh
Rules updated
Rules updated (v6)
ivan203413@raspberry:~ $ sudo ufw allow 80
Rules updated
Rules updated (v6)
ivan203413@raspberry:~ $
```

6. Configuración de fail2ban para SSH

Una vez permitido el puerto 22 para SSH, se debe de proteger, aquí es donde usará fail2ban. Se instalará con sudo apt install fail2ban, y aún sin configurar nada, ya entrará en funcionamiento, bloqueando y baneando de forma temporal, ip's que excedan un número de intentos fallidos de conectarse a nuestra máquina con SSH
Se hicieron unas pruebas con Putty para ver el funcionamiento del bloqueo de intentos fallidos. Se puede ver el registro de estos intentos con el comando: **grep "Failes password" /var/log/auth.log**

```
ivan203413@raspberry:~ $ grep "Failed password" /var/log/auth.log
Jul 25 05:52:01 raspberry sshd[2060]: Failed password for ivan203413 from 192.16
8.0.4 port 49189 ssh2
Jul 25 05:52:08 raspberry sshd[2060]: Failed password for ivan203413 from 192.16
8.0.4 port 49189 ssh2
Jul 25 05:52:13 raspberry sshd[2060]: Failed password for ivan203413 from 192.16
8.0.4 port 49189 ssh2
Jul 25 05:52:21 raspberry sshd[2060]: Failed password for ivan203413 from 192.16
8.0.4 port 49189 ssh2
ivan203413@raspberry:~ $
```



7. Justificación de SSH

El SSH se usó primordialmente para la edición de código en el backend, usando un plugin de VSCode para conectarnos a la máquina virtual mediante SSH y editar código de forma más fácil, ya que no se puede usar VSCode en el rasbian. También se ocupó para probar la efectividad de fail2ban.