

Scripts de pruebas	2
Script de prueba para el blockchain:	2
Script de prueba para el Snort/Snorby:	3

Scripts de pruebas

Script de prueba para el blockchain:

```
# Este es un script de prueba para el blockchain

# Importar las bibliotecas necesarias
import requests

# URL base de la API del backend del blockchain
base_url = "http://localhost:5000/api"

# Función de prueba para el inicio de sesión
def test_login():
    url = f"{base_url}/login"
    data = {
        "username": "usuario",
        "password": "contraseña"
    }
    response = requests.post(url, json=data)
    print(response.json())

# Función de prueba para consultar el saldo
def test_check_balance():
    url = f"{base_url}/balance"
    response = requests.get(url)
    print(response.json())

# Función de prueba para realizar una transacción
def test_make_transaction():
    url = f"{base_url}/transaction"
    data = {
        "amount": 10,
        "recipient": "usuario2"
    }
    response = requests.post(url, json=data)
    print(response.json())

# Ejecutar las pruebas
test_login()
test_check_balance()
test_make_transaction()
```

Script de prueba para el Snort/Snorby:

```
# Este es un script de prueba para Snort/Snorby

# Comando de prueba para Snort
snort_command = "snort -c /etc/snort/snort.conf -i eth0"

# Comando de prueba para Snorby
snorby_command = "snorbyctl start"

# Ejecutar Snort en segundo plano
nohup $snort_command > snort.log 2>&1 &

# Ejecutar Snorby en segundo plano
nohup $snorby_command > snorby.log 2>&1 &

# Esperar un tiempo para que los servicios se inicien
sleep 10

# Comprobar el estado de Snort
snort_status=$(pgrep snort)
if [[ -n $snort_status ]]; then
    echo "Snort está en funcionamiento."
else
    echo "Snort no se está ejecutando correctamente."
fi

# Comprobar el estado de Snorby
snorby_status=$(pgrep snorby)
if [[ -n $snorby_status ]]; then
    echo "Snorby está en funcionamiento."
else
    echo "Snorby no se está ejecutando correctamente."
fi
```