FRANCISCO JOSÉ DE CALDAS DISTRICT UNIVERSITY

PROFESSOR:

Carlos Andrés Sierra Virgüez

Gabriela Martínez Eslava

(20202020117)

SYSTEMS ENGINEERING - FACULTY OF ENGINEERING

COMPUTING NETWORKING I

WORKSHOP II

Bogotá C.D.

2024

I. OBJECTIVE

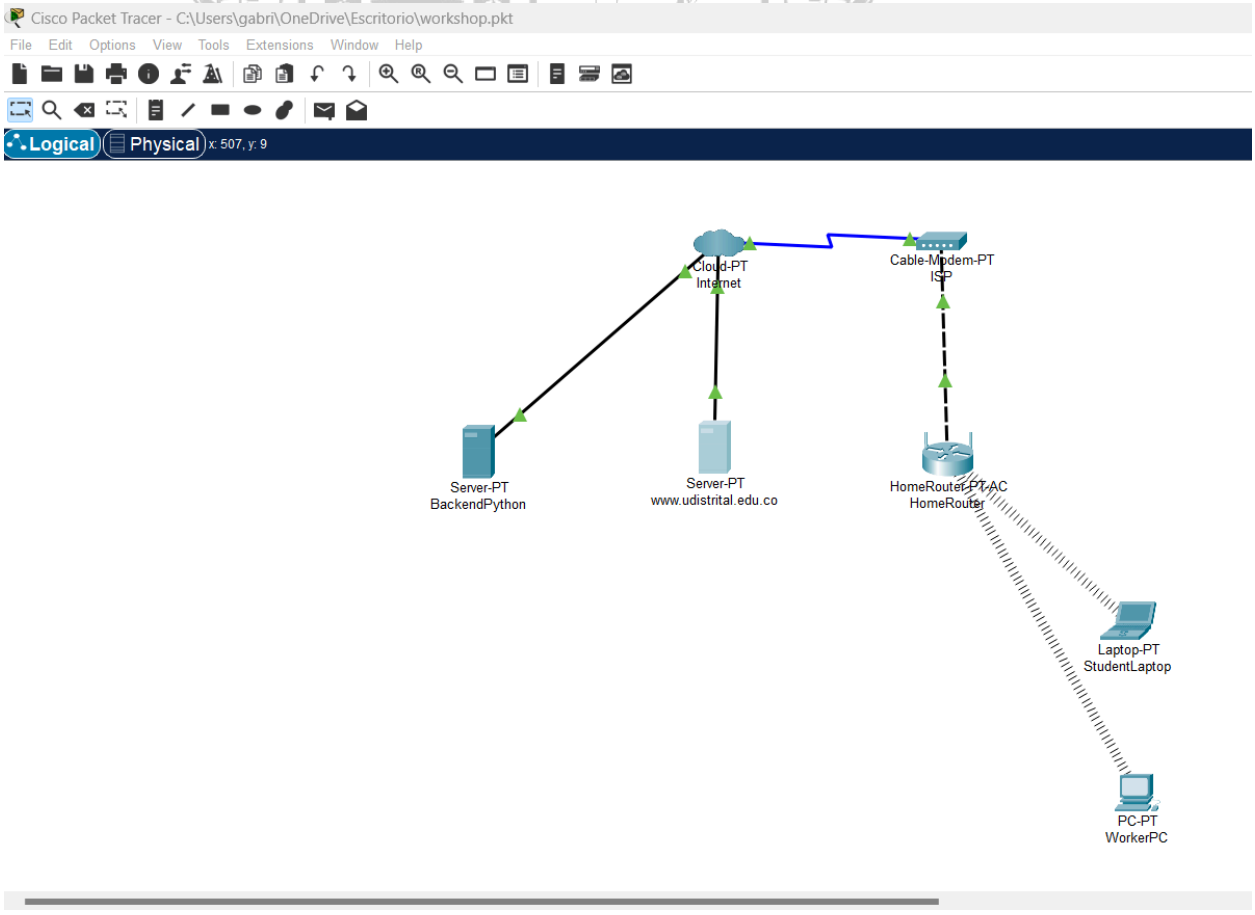
The main objective of this workshop is to understand and apply the use of sockets in network communication. Through practical exercises, you will explore how sockets function within the OSI model, gaining insights into the interaction between different network layers.

II. WORKSHOP

The steps for carrying out the workshop will be described below:

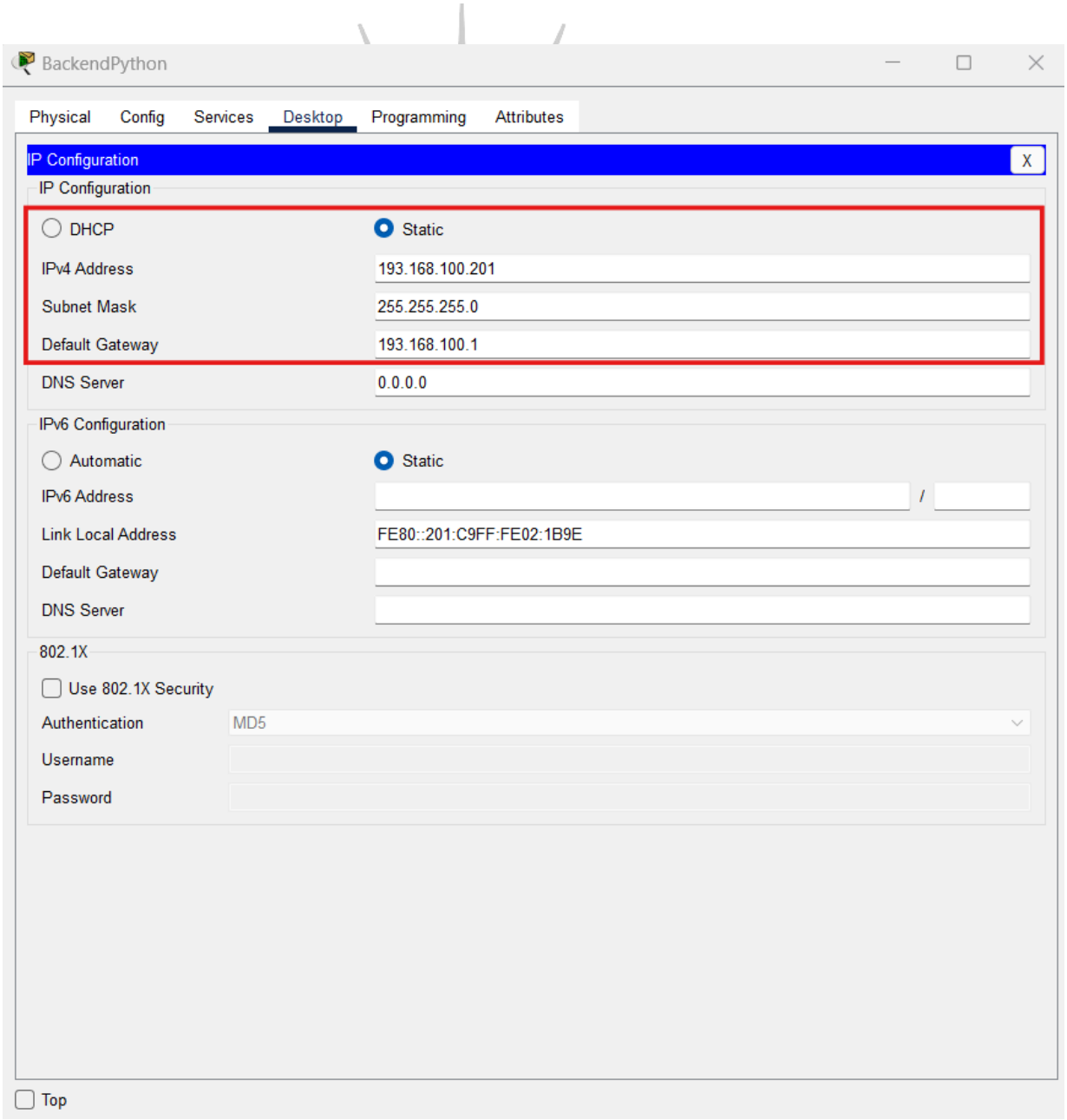
1. OPEN *WORKSHOP I* FILE AND ADD THE NEW SERVER

Let's open the file from the previous workshop and add the new 'BackendPython' server from end devices.

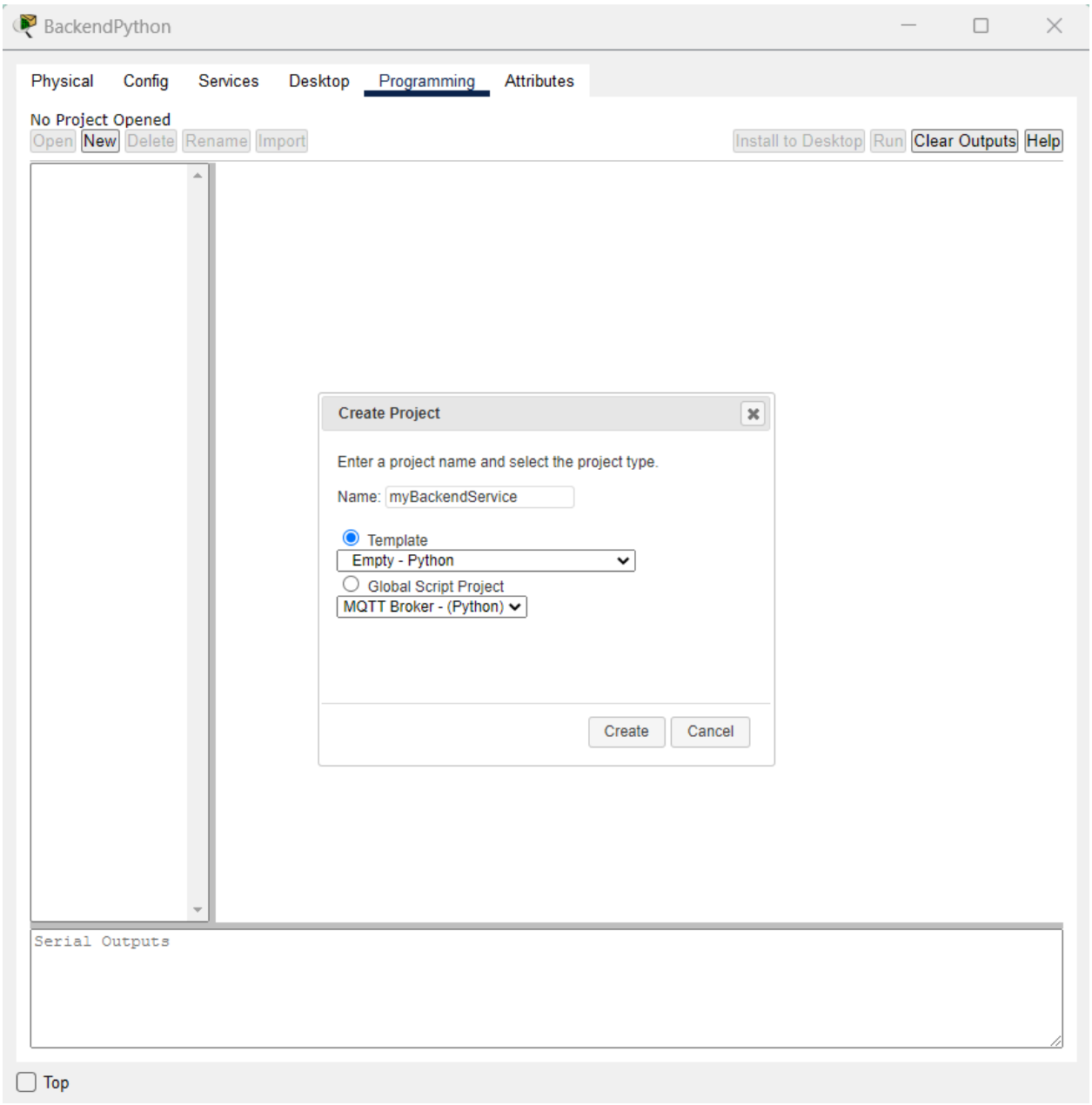


2. SETTING UP NEW SERVER "BackendPython" CONFIGURATION

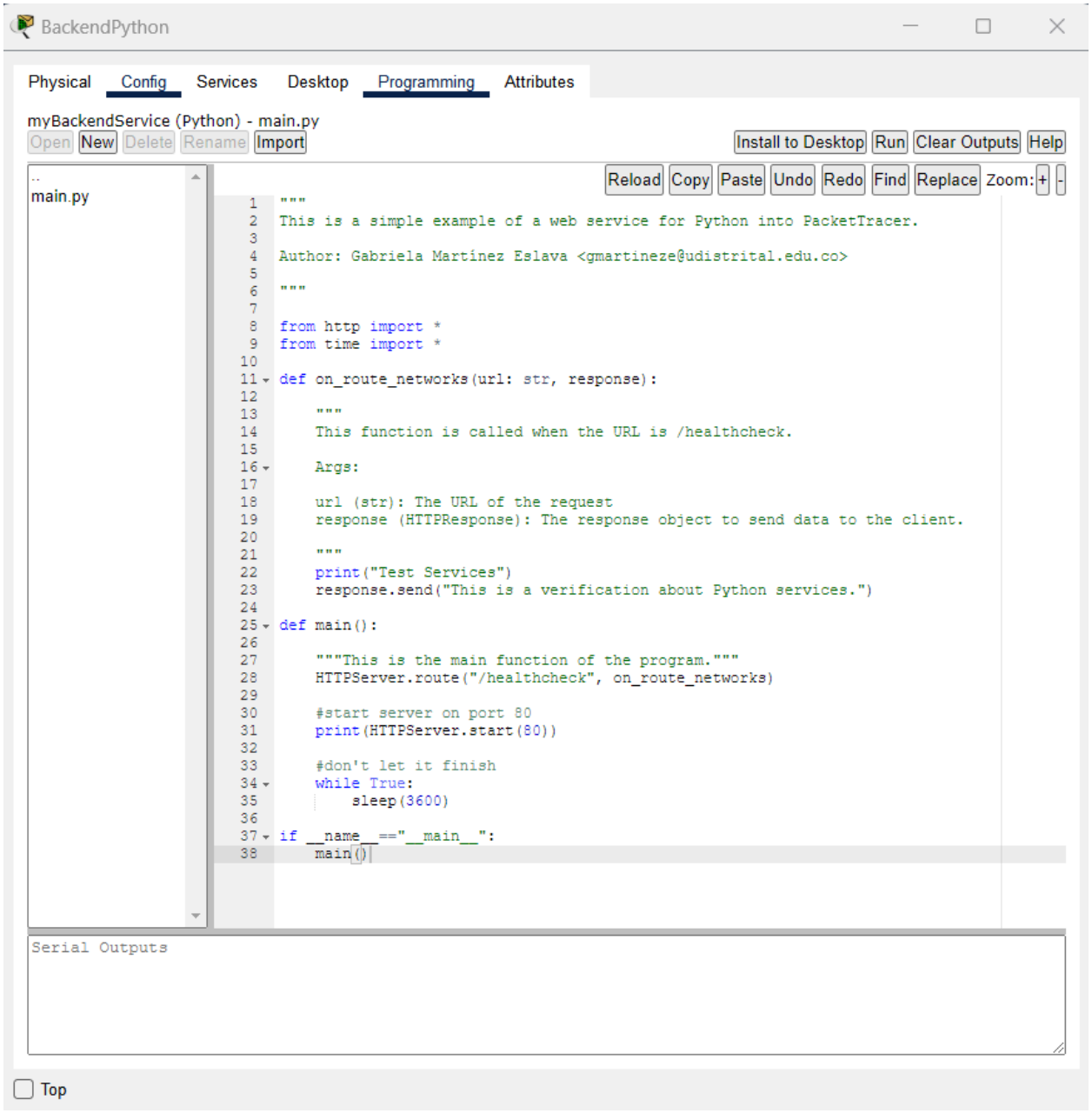
It is necessary setting up the data of the server: *IPv4, Subnet Mask, Default Gateway.*



Subsequently, in the *Programming* services section, we create a new project using ***Template:*** empty -- Python.

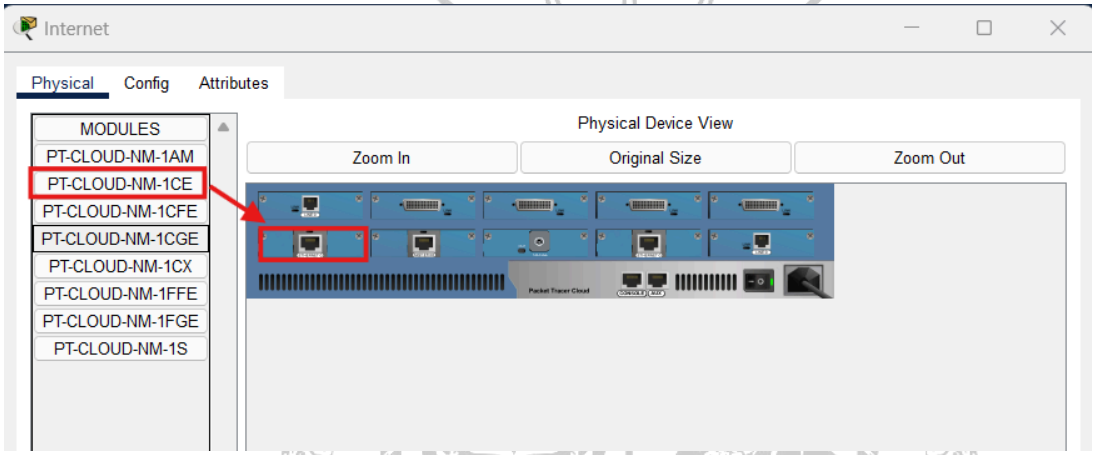


Once created our project, let's add the corresponding code at *main.py* file

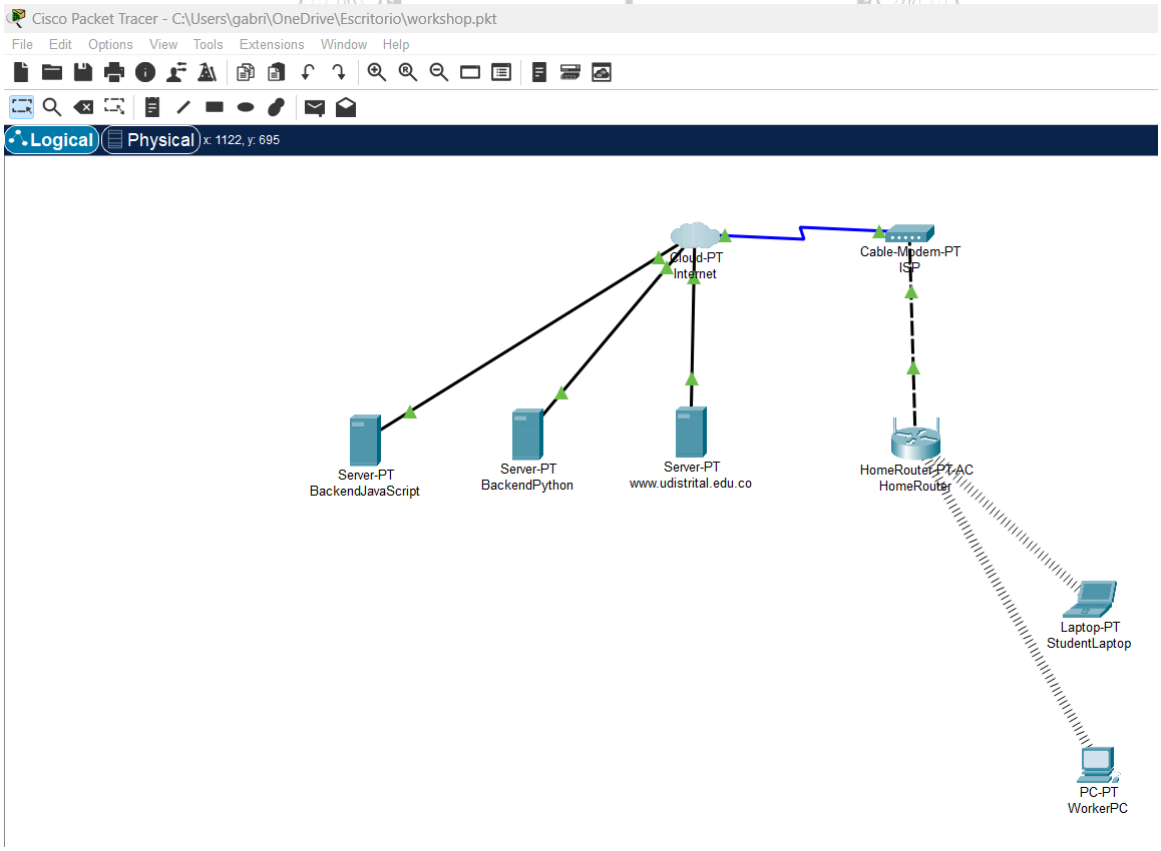


3. ADD ANOTHER NEW SERVER

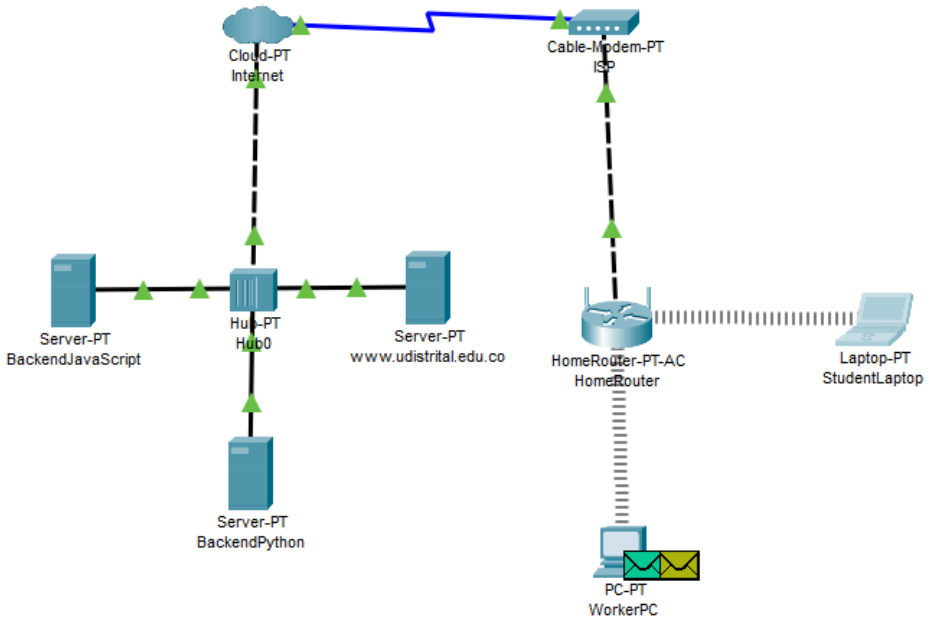
add the new 'BackendJavaScript' server from end devices. In this particular case, the Cloud-PT *Internet* did not have enough ports to connect this third server, so it was necessary to add a PT-CLOUD-NM-1CE module. After this adjustment, the new 'BackendJavaScript' server was linked.



Now the server is successfully connected.

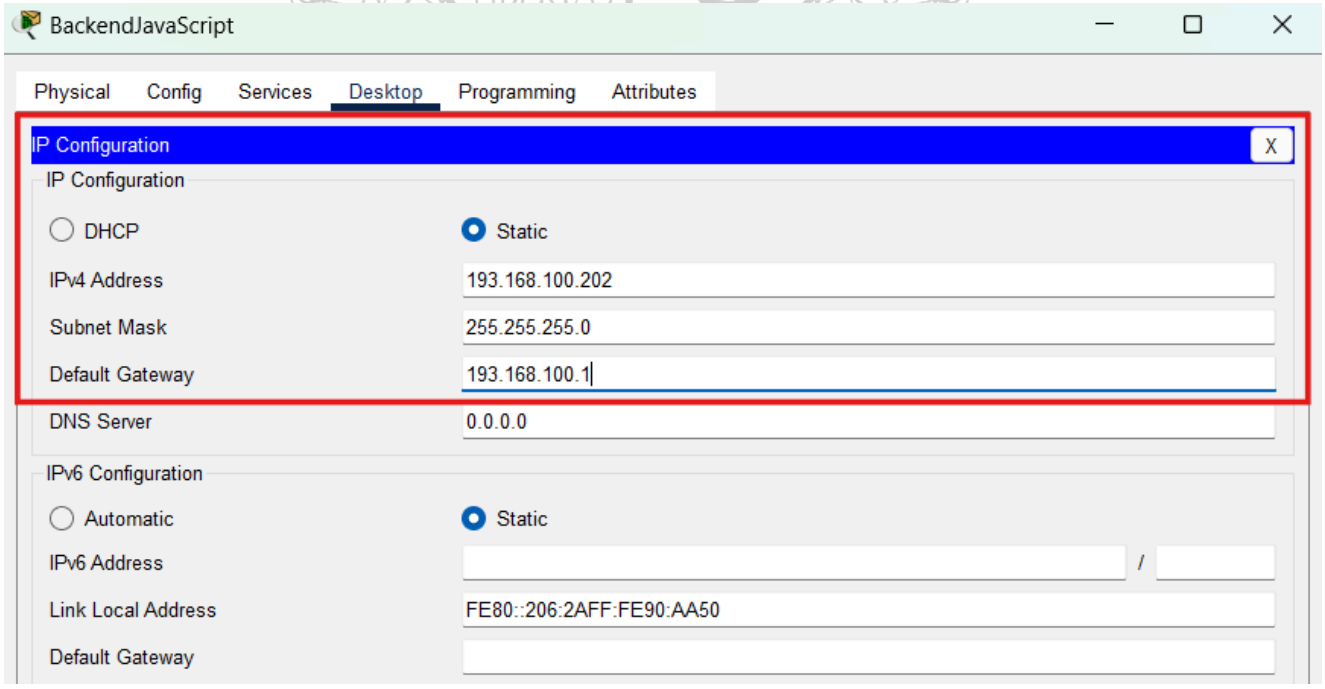


In the process of connecting the server I noticed that it will be better using a hub between servers to make the connection.

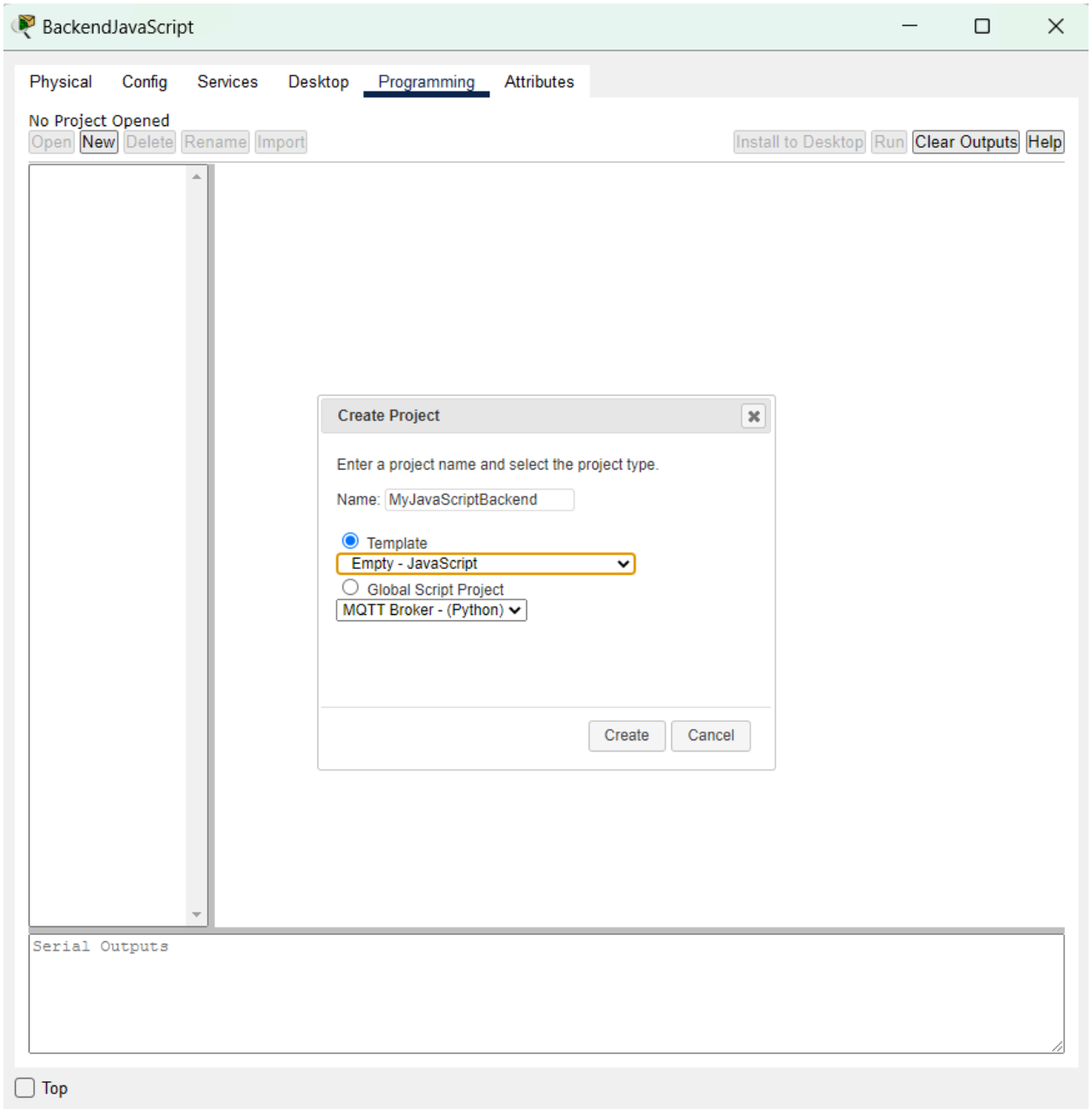


4. SETTING UP NEW SERVER "BackendJavaScript" CONFIGURATION

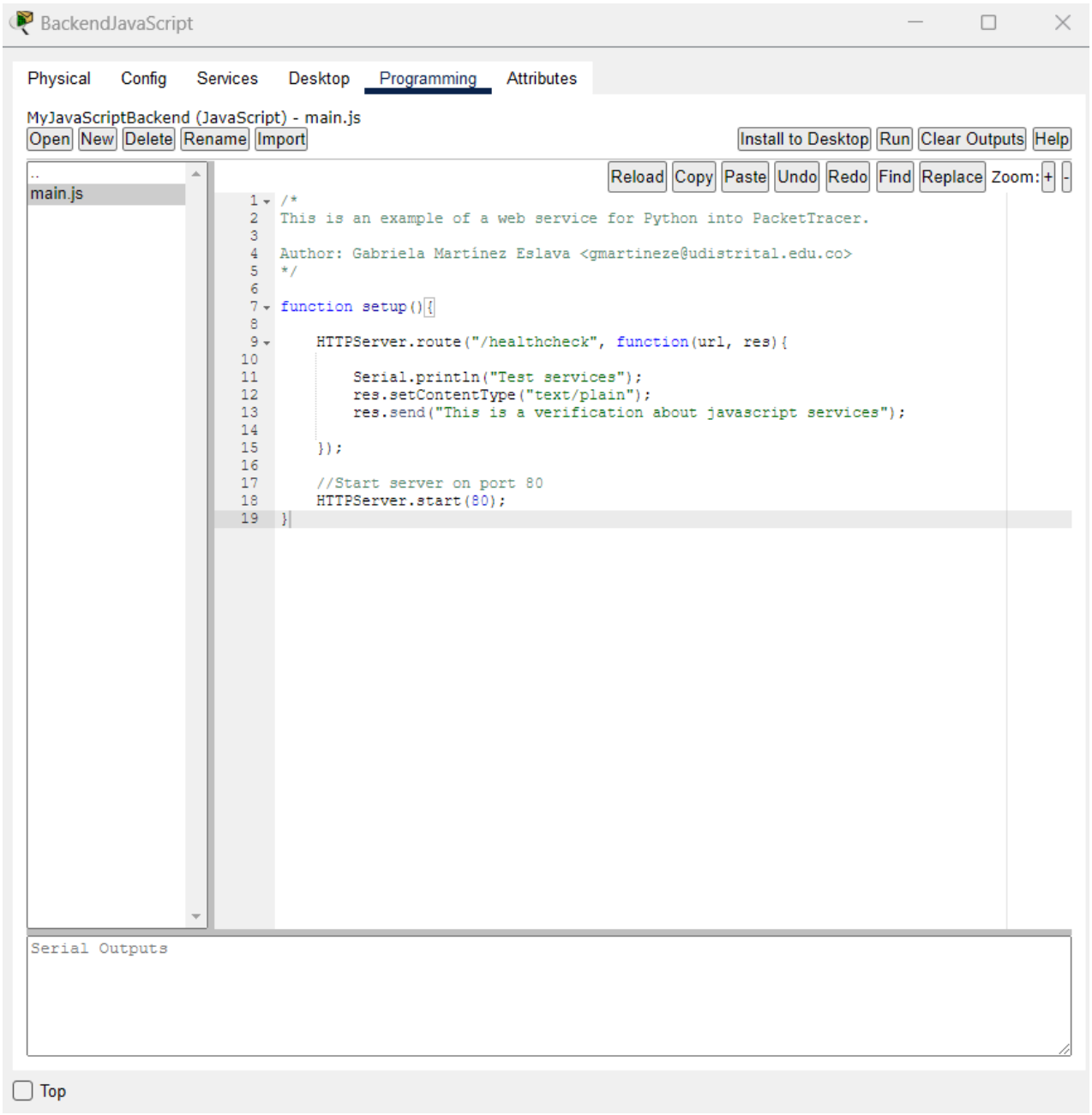
set up the data of the server: *IPv4, Subnet Mask, Default Gateway.*



Then, in the *Programming* services section, we create a new project using ***Template:*** empty -- JavaScript.



Once created our project, let's add the corresponding code at *main.js* file



Then, it is necessary to build the frontend service using HTML, CSS and JavaScript.

HTML file:

index.HTML

```html
<html>
    <head>
        <title>Workshop 4 - Networks</title>
        <meta charset="UTF-8">
        <meta name="description" content="This is a simple example to explore OSI">
        <link rel="stylesheet" type="text/css" href="styles.css">
        <script src="functions.js"></script>
    </head>
    <body>
        <h1>Workshop 4 - Networks</h1>
        <h2>OSI Layers</h2>
        <p>Click on the buttons to explore services on the network</p>
        <div>
            <button onclick="callPython()">Python Message</button>
            <button onclick="callJavaScript()">JavaScript Message</button>
        </div>
        <div id="result"></div>
    </body>
</html>
```

CSS file:

I changed it to make it prettier, nothing personal :P

styles.CSS

```css
body {
    font-family: "Arial", sans-serif;
    font-size: 16px;
    background-color: #f8f4f9;
    color: #5a5a5a;
    text-align: center;
    padding: 20px;
}

h1 {
    color: #ff6b6b;
    font-size: 28px;
    margin-bottom: 20px;
}

h2 {
    color: #6b7aff;
    font-size: 22px;
    margin-bottom: 15px;
}


button {
    background-color: #ffcb77;
    color: #5a5a5a;
    border: none;
    padding: 12px 24px;
    font-size: 16px;
    font-weight: bold;
    cursor: pointer;
    border-radius: 12px;
    transition: background 0.3s, transform 0.2s;
    box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);
}

button:hover {
    background-color: #ffd97d;
```

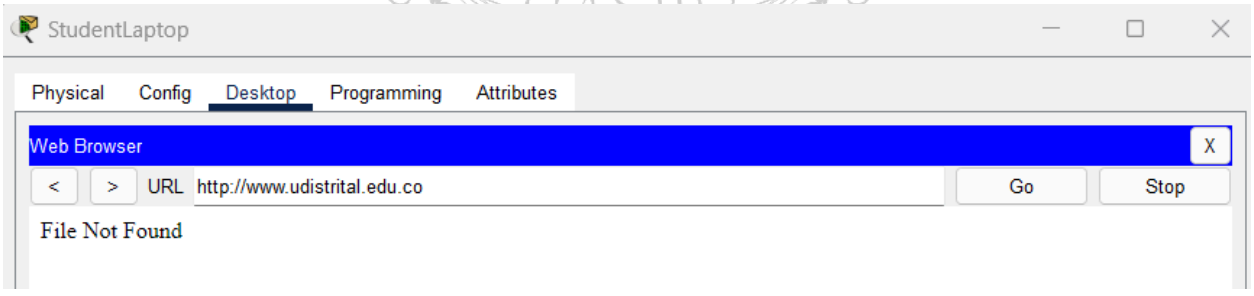
styles.CSS

```css
button {
    border: none;
    padding: 12px 24px;
    font-size: 16px;
    font-weight: bold;
    cursor: pointer;
    border-radius: 12px;
    transition: background 0.3s, transform 0.2s;
    box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);
}

button:hover {
    background-color: #ffd97d;
    transform: scale(1.05);
}

.result {
    margin-top: 20px;
    padding: 15px;
    border: 2px solid #b8c0ff;
    background-color: #e3e8ff;
    border-radius: 12px;
    box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);
    display: inline-block;
    min-width: 200px;
}
```

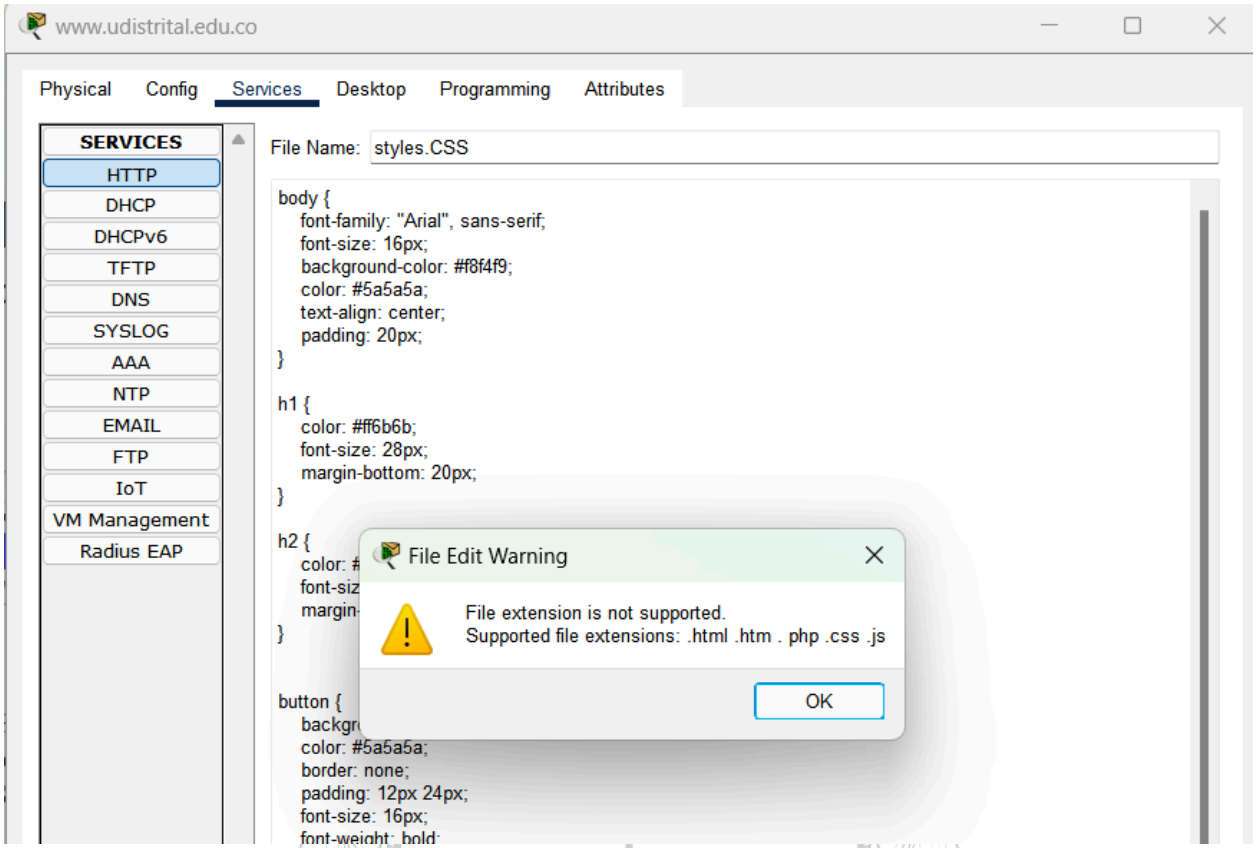
JavaScript file:

5. TEST ACCESS IN THE DEVICES

First, when I try to access *www.udistrital.edu.co* from *StudentLaptop* or *WorkerPC* file was not found.

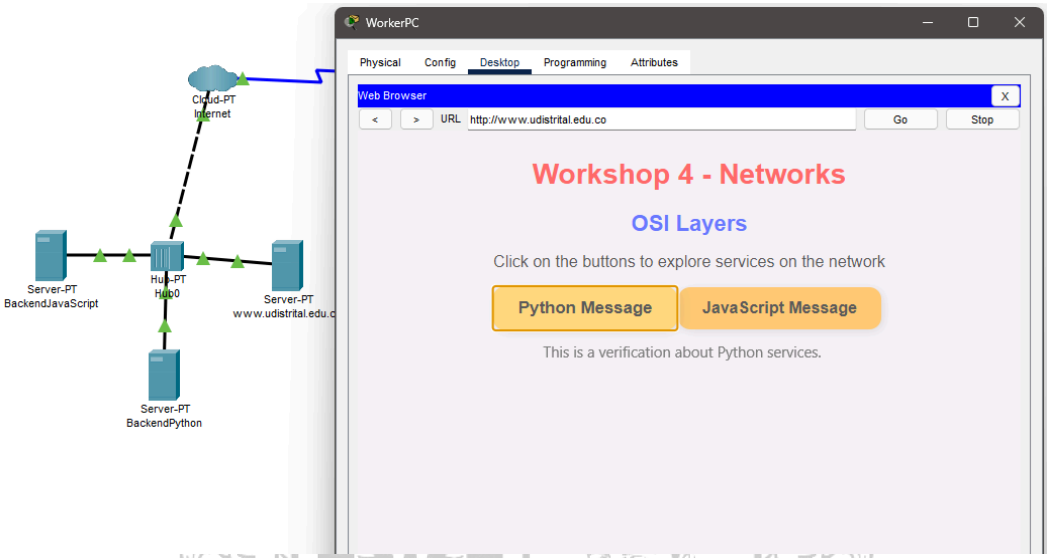


So I checked the .HTML, .CSS, and .js files, and the issue was the extension, which must be in lowercase. The problem was fixed by renaming the file from index.HTML to index.html, and the same for style.CSS

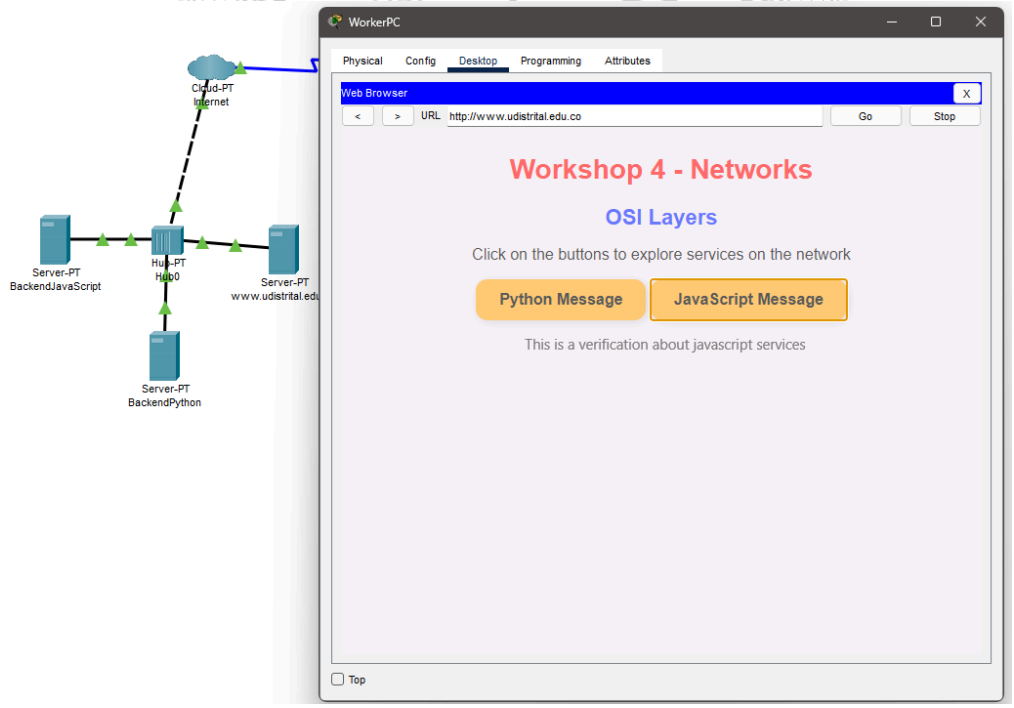


After this, it finally worked 😀

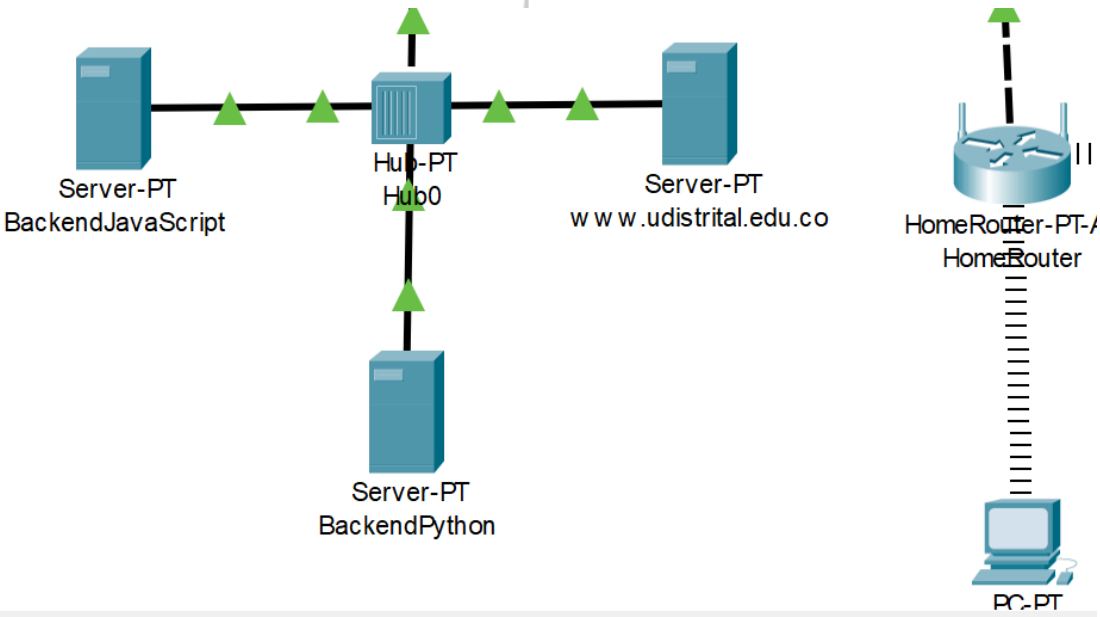
First I tried calling the python backend through its button, and the backend answered by sending the corresponding message.

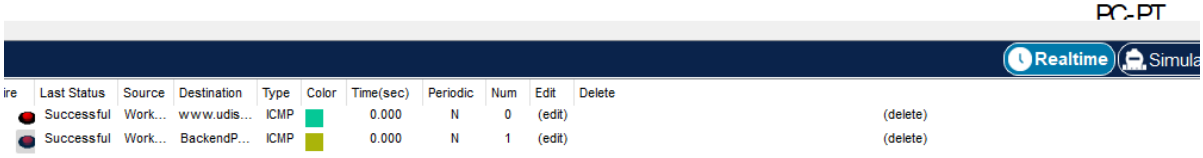


Then i tried calling Java Script backend and it worked as well 😀

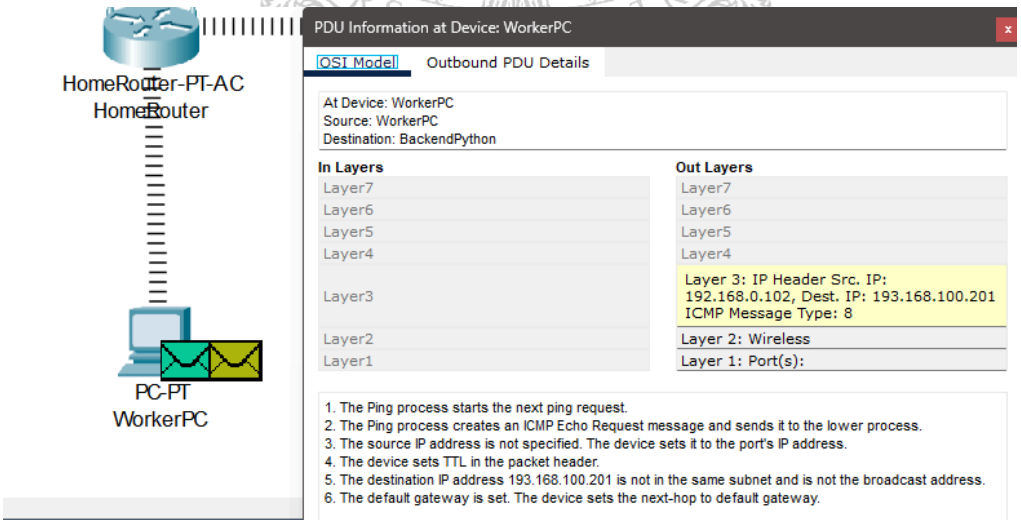


I made the simulations trying to call the frontend from Worker's PC and both of them worked.

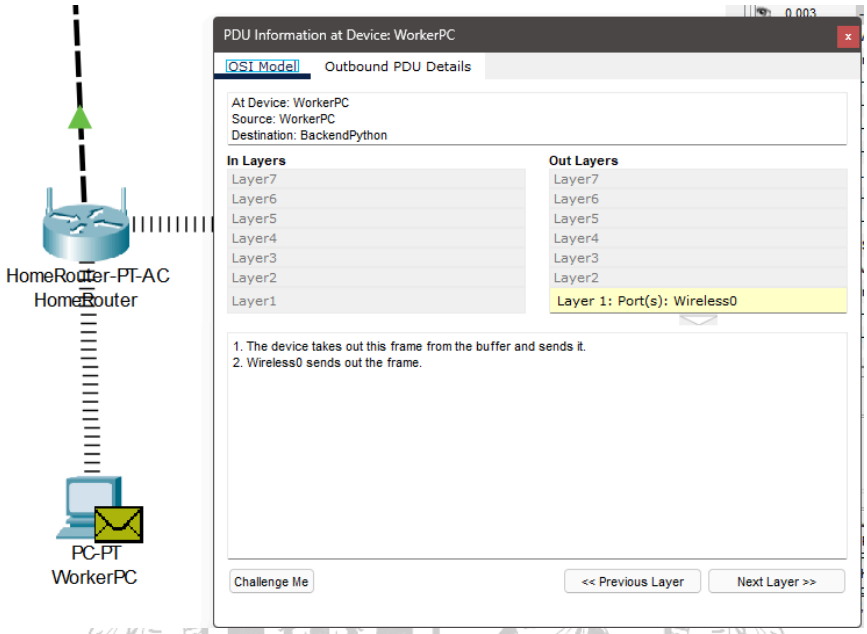




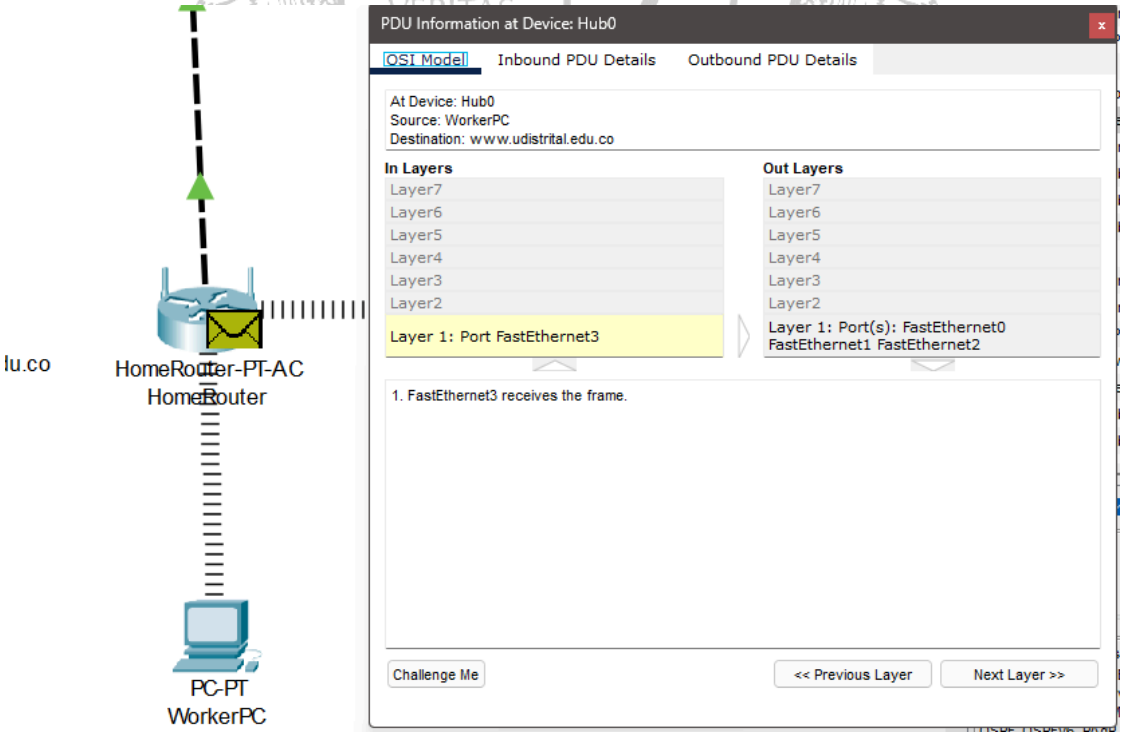
When i send the package from Worker's PC it starts from network layer adding the source and destiny IP's.



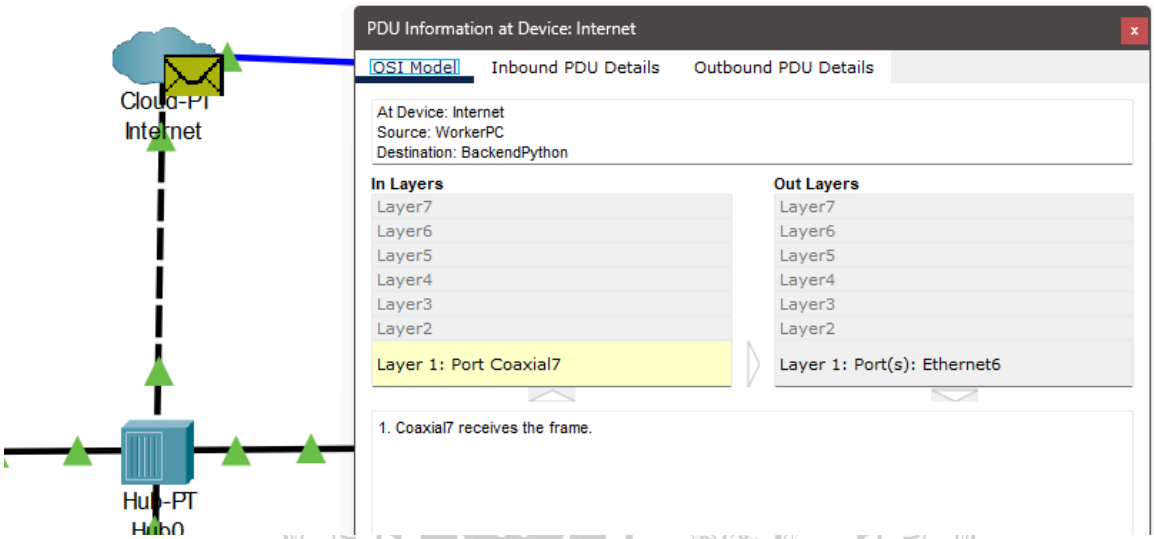
Then it goes through data link layer adding the MAC and goes through the wires in the physical layer.



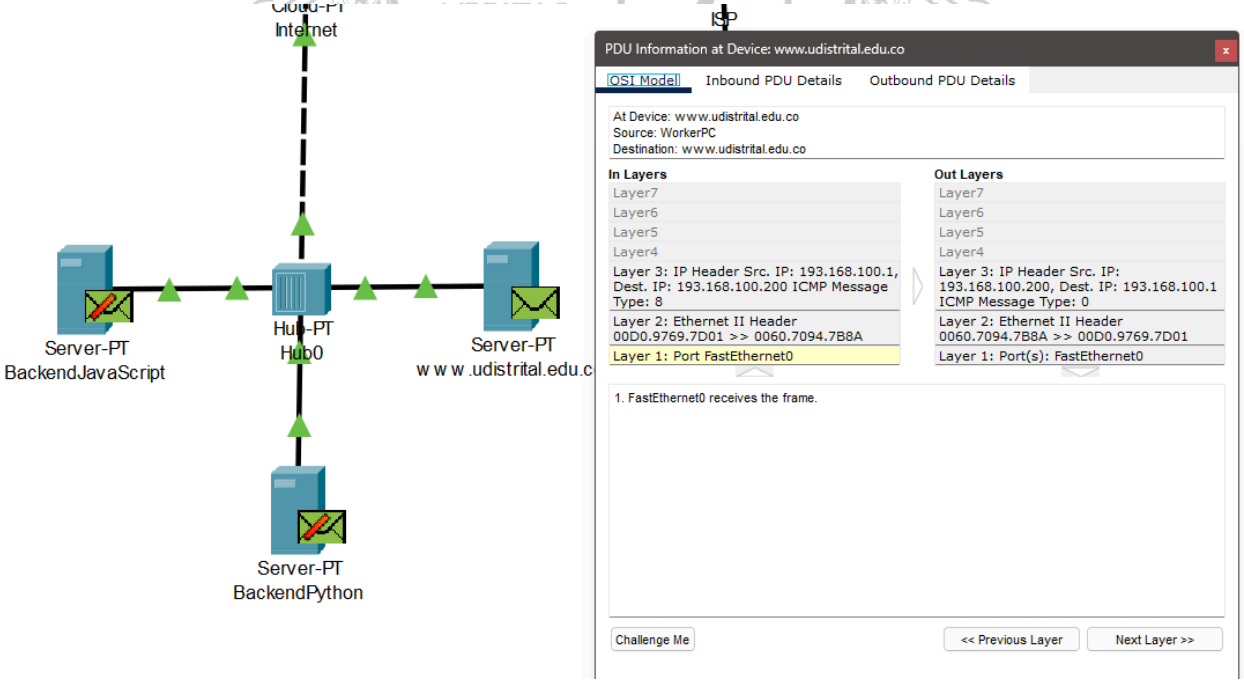
The package travels through wireless connection to the Home Router using the first “in” layer of the OSI model.



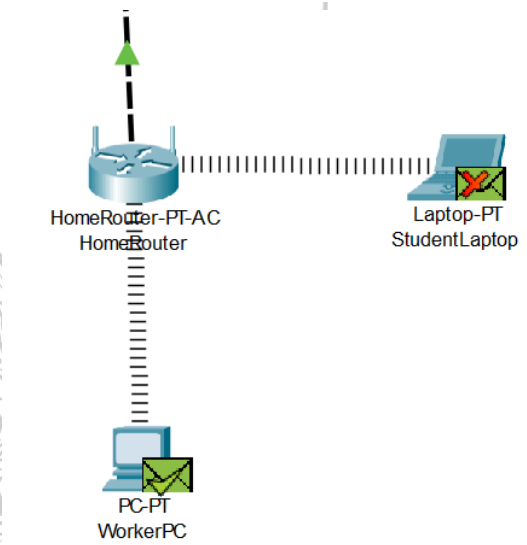
It continues travelling through the physical layer, searching for the destination device



Then after reaching the hub, it starts unpacking the payload and search for the device's IP and MAC in the datalink and network layer.



Then it sends back an answer to the source device notifying a successful connection.



III. CONCLUSIONS

- Networks are deeply embedded in most of the technologies we rely on today, serving as the backbone of essential fields such as web development. Their seamless integration enables efficient communication, data exchange, and connectivity, making them a fundamental pillar of modern digital systems.

- The OSI model provides a robust and well-structured framework that enables seamless and efficient communication between systems, ensuring interoperability across diverse networks.