Universidad Nacional de Ingeniería
Facultad de Ciencias
Escuela Profesional de Ciencia de la Computación
Administración de Redes

Partial Exam (Practice)

CC312 **Date: 10/19/2024 Duration: 90 minutes** **Semester: 2024-II**

Norms:

1. The exam is IN-PERSON
2. .Do not share answers/queries with your colleagues through chats, social networks, other written and/or digital media.
3. Class notes are allowed.
4. The solutions will be sent to the teacher's email account.
5. Any unethical act will be reprimanded and recorded in the student's record.

Last Name: _____ First Name: _____

In spanish.

1. [5.0 puntos.] Cada pregunta vale 1 punto.

1.1 ¿Cómo añadiría con el comando ***ip***, el IP ***192.168.1.50/20*** a la interface ***eth0****?*

1.2 ¿Cómo deshabilitaría con el comando **ip**, la interface de red ***enp3s0***?

1.3 El comando ***neighbour*** se utiliza para ver la dirección MAC de los dispositivos conectados a su sistema. ¿Cómo añadiría con el comando ***ip,*** la entrada ***ARP*** con el IP ***192.168.1.50/24*** a la interface ***enp3s0****?*

1.4 Si tenemos la ruta **192.168.4.0/20** que apunta a la interfaz ***enp3s0*** ¿cómo la eliminamos?

1.5 Explique qué resultado obtendrá con:
$ sudo nmcli connection add type ethernet ifname enp0s5

2. [5.0 points] The question will be answered in the jupyter notebook ***exercise-1_examen-partial_2024-2.ipynb*** attached to the exam.

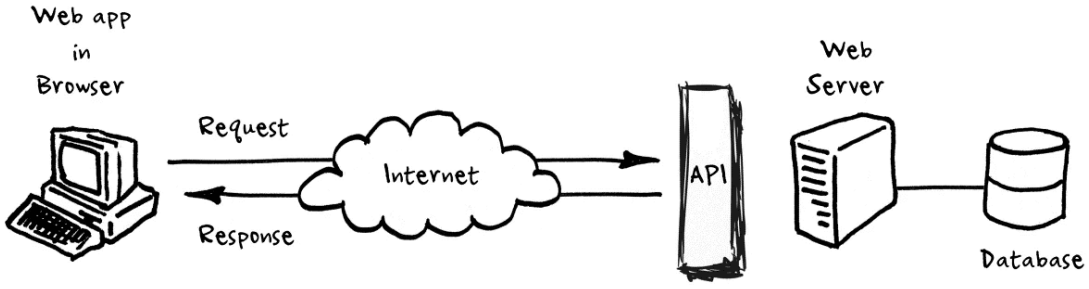
3. [5.0 points] Work: Building API with ***FastAPI***

What is an API?

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. When you use an application on your phone, the application connects to the Internet and sends data to a server. The server then processes the data and sends it back to your phone. The application on your phone then interprets the data and presents it to you in a readable way.

An API is like a waiter in a restaurant. The waiter takes your order and sends it to the kitchen. The kitchen then prepares the food and sends it back to the waiter. The waiter then brings the food to you.

In the same way, an API takes a request from an application and sends it to a server. The server then processes the request and sends the data back to the application. The application then interprets the data and presents it to the user.



What is FastAPI

FastAPI is a high-performing web framework for building APIs with Python 3.7+ based on

standard Python type hints. It helps developers build applications quickly and efficiently. FastAPI is built on top of the Starlette web server and includes features that make building web applications easier, such as automatic data validation, error handling, and interactive API docs.

We will look at all these features individually in this section. First, let's look at key features as pointed out in the original documentation of FastAPI.

- **Performance**: On par with NodeJS and the Go language.
- **Speed**: Increase the development speed 2-3X.
- **Easy**: Great editor support. Completion everywhere. Easy to learn and use.
- **Robust**: Production-ready code with automatic interactive documentation.
- **OpenAPI based**: Fully compatible with OpenAPI and JSON Schema.

Installing FastAPI

FastAPI requires Python 3.7+. It can be installed using pip. You will need to install FastAPI and the ASGI server `uvicorn

```
# install fastapi

Complete install Complete

# install uvicorn
Complete install Complete
```

Create a simple API

Let's directly get into creating a very simple toy API. I am using VS Code to implement this, but you can use any editor you like

```
from typing import Union

from fastapi import FastAPI

app = Complete

@app.get("")

Complete read_root():

    return {"Hola": "CC 312"}


@app.get("/items/{item_id}")

Complete read_item(item_id: int, q: Union[str, None] = None):

    return {"item_id": item_id, "q": q}
```

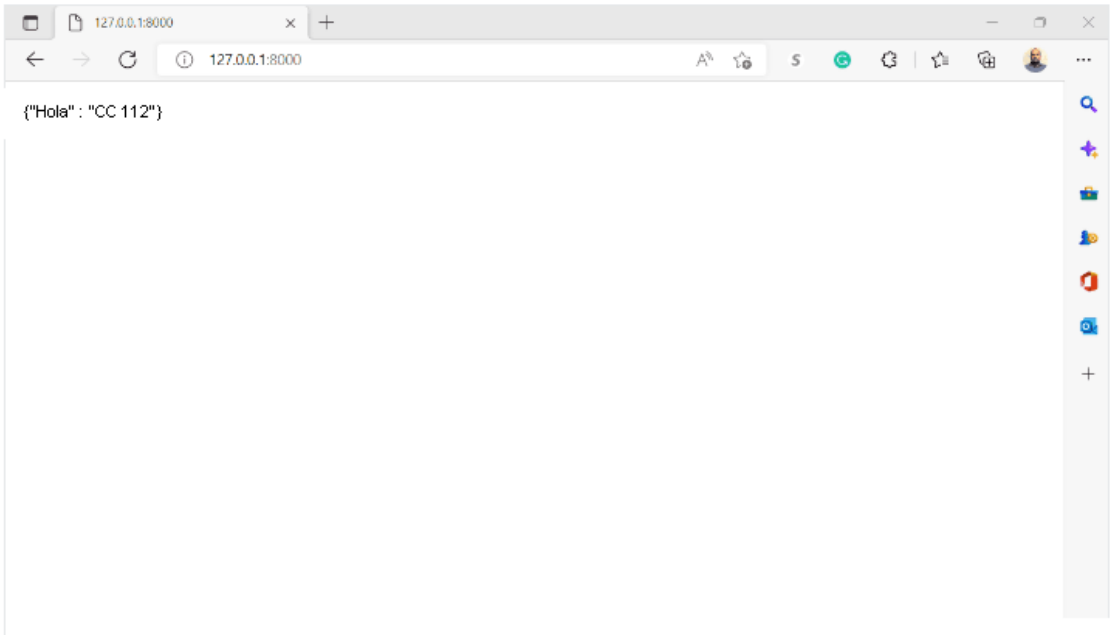
Now using a command line terminal, run this API with the following command:

```
uvicorn main:app –reload
```

`main` is the name of the Python file, and `app` is the variable that stores the FastAPI class. You can name them whatever you like. Once you run the above command, you will see something like this on your terminal:

```
(base) C:\Users\owner\fastapi>uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Users\\owner\\fastapi']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [19124] using statreload
INFO:     Started server process [32540]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Head over to the link in your browser, and if you see a page that shows `Hola CC 312` the API is up and running.



Congratulations on building your first API!!!