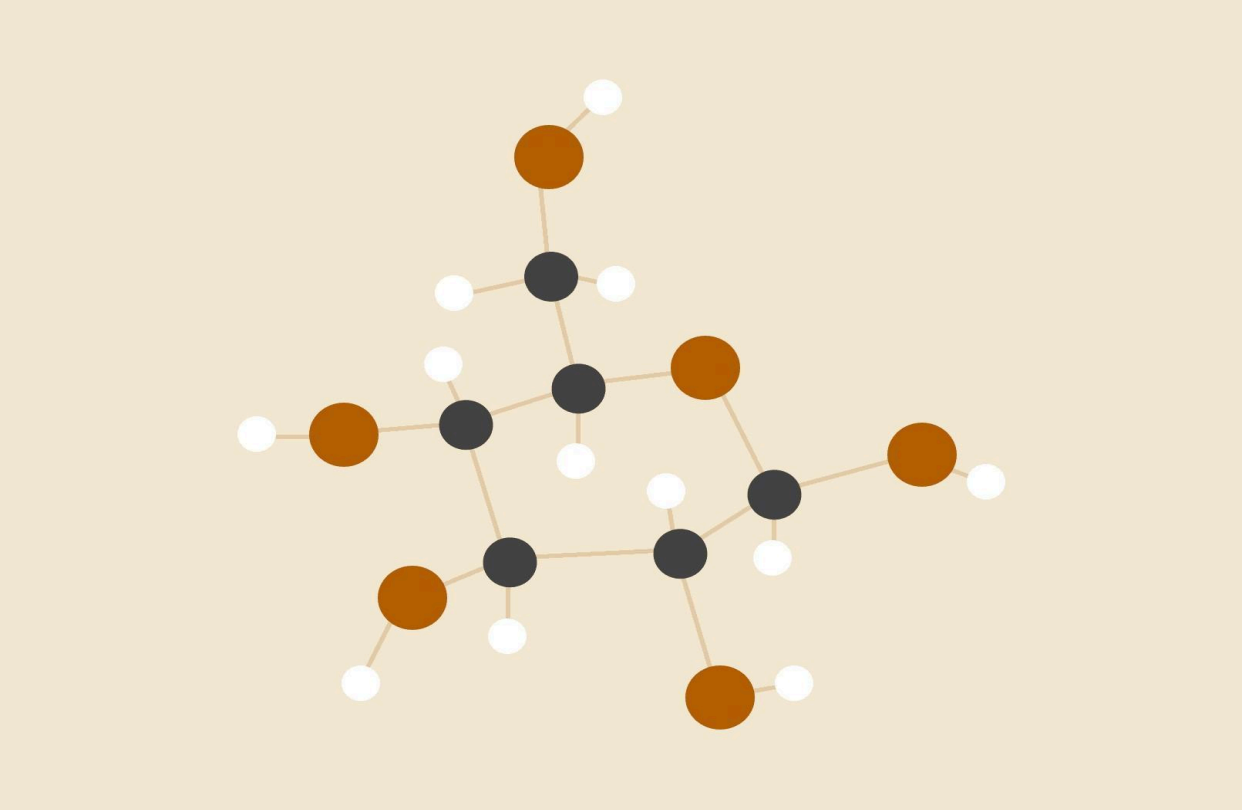
TRABAJO PRÁCTICO INTEGRADOR

IA3.5 Redes de Datos

Tecnicatura Universitaria en Inteligencia Artificial



Grimaldi Damián

Piccirill Mayra

2/7/2024

Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

INTRODUCCIÓN

- La base de datos elegida es el archivo prize.json.
- Trabajamos con el sistema operativo de Windows y Linux.
- Al momento de realizar el tp utilizamos un entorno virtual. Para crear el mismo, ejecutar el siguiente comando:

PARA WINDOWS:

```
python -m venv ./.venv #para crear un entorno virtual
.venv\Scripts\activate  #para activarlo
```

PARA LINUX:

```
python3 -m venv ./venv #para crear un entorno virtual
source ./.venv/bin/activate # para activarlo
```

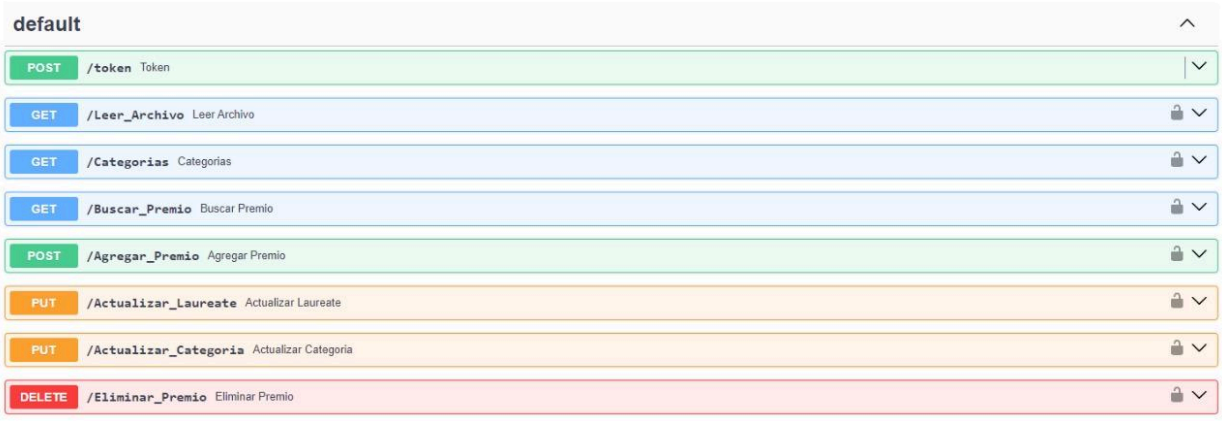
- Una vez realizado esto, se instalan las librerías necesarias:

```
pip install fastapi #para la creación de api
pip install requests #para abrir url
pip install uvicorn #para poder ejecutar el servidor api
pip install python-jose # para poder generar claves token
pip install pydantic # para crear un modelo de datos para uso de las clases
```

DESCRIPCIÓN DEL ENTORNO DE TRABAJO DEL SERVIDOR

Para configurar el servidor trabajamos en Python.

Configuraciones realizadas:



- **POST/token:** valida usuario y contraseña. Si los datos son correctos, lo transforma en un token, que será utilizado para las demás funciones.
- **GET/ leer archivo:** todos los tipos de usuarios pueden leer el archivo.
- **GET/Categorías:** todos los tipos de usuarios pueden ver las categorías.

- **GET/buscar_premio:** todos los tipos de usuarios pueden buscar premios.
- **POST/agregar premio:** los usuarios administradores pueden agregar premios.
- **PUT/actualizar_Laurete:** los usuarios administradores pueden actualizar Laureate.
- **PUT/actualizar_Categoria:** los usuarios administradores pueden actualizar categorías.
- **DELETE/eliminar_premio:** los usuarios administradores pueden eliminar premios.

Algunos problemas encontrados:

- desconocimiento sobre uso de tokens.
- problemas para validar los tokens.
- dificultades respecto al manejo de objetos al momento de desarrollar el código.

DESCRIPCIÓN DEL ENTORNO DE TRABAJO DEL DEL CLIENTE

Para configurar el cliente utilizamos Python.

Configuraciones realizadas:

```
def login():···

#Aca al token lo transforma de esta manera para poder mandarlo, porque sino produce error

def get_headers(token):···

def verArchivo():···

def VerCategorias():···

def BuscarPremio():···

def agregarLaureate():···

def AgregarPremio(token):···

def ActualizarLaureate(token):···

def ActualizarCategoria(token):···

def EliminarPremio(token):···

def menu(token):···

valor=login()
token = get_headers(valor)
menu(token)
```

- **def login:** se loguea el usuario.

- **def get_headers:** transformar el token obtenido en un formato que sea aceptable.
- **def ver_archivo:** todos los usuarios ven los archivos.
- **def ver_Categorias:** todos los usuarios pueden ver las categorías.
- **def buscar_premio:** todos los usuarios pueden buscar premios.
- **def actualizar_Laureate:** sólo los usuarios administradores pueden actualizar Laureate.
- **def actualizar_Categoria:** sólo los usuarios administradores pueden actualizar categorías.
- **def eliminar_premio:** sólo los usuarios administradores pueden eliminar premios.
- **def menu:** dependiendo del tipo de usuario, se mostrará ….

DESCRIPCIÓN DE LA BASE DE DATOS ELEGIDA

El archivo prize.json contiene lo siguiente:
El archivo cuenta con un total de un encabezado, denominado prizes, y posee los siguientes atributos:

- year y su tipo: str
- category y su tipo: str
- overallMotivation y su tipo: str

Laureates, es una lista de diccionario con los siguientes claves y valores:

- id y su tipo: str
- firstname y su tipo: str
- surname y su tipo: str
- motivation y su tipo: str
- share y su tipo: str

Algunos problemas encontrados:

- dificultades para desarrollar código: la idea era generar objetos para el desarrollo de cliente, y que el servidor los convierta en formato adecuado.

PROCEDIMIENTO

- Para levantar el servidor usar el comando:

uvicorn main:app –host 0.0.0.0 –reload

- El host sirve para que cualquier dispositivo de la misma red se pueda conectar. Luego, con una página web ingresar:

http://localhost:8000/docs

- Ejecutar el archivo clienteApi.py

INSTRUCCIONES DE USO DEL CLIENTE

- Ejecución y uso cliente: ejecutar archivo clienteApi.py
- Entorno: el archivo debe ejecutarse en un entorno virtual. En cualquier sistema operativo.
- Información adicional: hay 2 tipos de usuarios:
    - administrador, que tiene todos los permisos para ejecutar todas las tareas.
    - usuario final, que tiene permisos limitados.

APÉNDICE A

El código del servidor se encuentra en carpeta .zip.

APÉNDICE B

El código del cliente se encuentra en carpeta .zip.