

# 126- [PF] - Lab - Controladores de archivos y módulos para recuperar información sobre la insulina



# Creación de controladores de archivos y módulos para recuperar información sobre la insulina

# Información general sobre el laboratorio

En este laboratorio, deberá realizar lo siguiente:

- crear un módulo
- abrir un archivo y cargar los datos JSON que contiene con el uso del módulo integrado JSON de Python
- analizar la estructura JSON para acceder a los datos de insulina
- calcular el peso molecular aproximado de la insulina utilizando un código dado (similar al laboratorio Trabajo con la secuencia de cadena y el peso numérico de la insulina en Python)

## Tiempo de finalización estimado

25 minutos

#### Acceso al IDE de AWS Cloud9

- 1. Para activar el entorno de su laboratorio, desplácese hasta la parte superior de estas instrucciones y seleccione **Start Lab** (Iniciar laboratorio).
  - Se abrirá el panel **Start Lab** (Iniciar laboratorio), donde se muestra el estado del laboratorio.
- Espere hasta que aparezca el mensaje Lab status: ready (Estado del laboratorio: listo) y, a continuación, para cerrar el panel Start Lab (Iniciar laboratorio), haga clic en la X.
- 3. En la parte superior de estas instrucciones, elija **AWS**.
  - AWS Management Console se abrirá en una pestaña nueva del navegador. El sistema iniciará la sesión de forma automática.

Nota: Si no se abre una pestaña nueva del navegador, generalmente aparece un anuncio o un icono en la parte superior de este, el cual indica que el navegador no permite que se abran ventanas emergentes en el sitio. Haga clic en el anuncio o en el icono, y elija Allow pop ups (Permitir ventanas emergentes).

4. En AWS Management Console, elija **Services** (Servicios) > **Cloud9**. En el panel **Your environments** (Sus entornos), busque la tarjeta **reStart-python-cloud9** y elija **Open IDE** (Abrir IDE).

Se abre el entorno de AWS Cloud9.

Nota: Si se abre una ventana emergente con el mensaje .c9/project.settings have been changed on disk (Se ha modificado la configuración de .c9/project. en el disco), elija Discard (Descartar) para ignorarlo. Del mismo modo, si una ventana de diálogo le pide que Show third-party content (Muestre contenido de terceros), elija No para rechazar la indicación.

# Creación del archivo de ejercicio de Python

- En la barra de menú, elija File -> New from template -> Python File (Archivo > Nuevo a partir de plantilla > Archivo en Python).
  - Esta acción crea un archivo sin título.
- 2. Elimine el código de muestra del archivo de plantilla.
- 3. Elija **File -> Save As...** (Archivo > Guardar como...), proporcione un nombre adecuado para el archivo de ejercicios (por ejemplo, calc\_weight\_json.py) y guárdelo en el directorio */home/ec2-user/environment*.
- 4. Cree un segundo archivo y nómbrelo jsonFileHandler.py.
  - Nota: La extensión de los archivos en Python es .py.
- 5. Cree un directorio denominado files (archivos).

#### Acceso a la sesión del terminal

1. En su IDE de AWS Cloud9, elija el icono + y seleccione **New Terminal** (Nuevo terminal).

Se abre una sesión de terminal.

- 1. Para ver el directorio en el que está trabajando actualmente, escriba pwd. Este comando lleva a /home/ec2-user/environment.
- 2. En este directorio, localice el archivo que creó en la sección anterior.

# Ejercicio 1: Creación del archivo de datos de moléculas JSON

Este documento JSON almacena toda la información del laboratorio anterior, como las moléculas de insulina, los pesos numéricos de los aminoácidos y el peso real de la molécula de insulina

- 1. En la barra de menú, seleccione **File -> New File** (Archivo > Archivo nuevo).
- 2. Copie y pegue el siguiente código en el archivo recién creado:

```
{
   "molecules":{
      "lsInsulin": "malwmrllpllallalwgpdpaaa",
      "bInsulin": "fvnqhlcgshlvealylvcgergffytpkt",
      "aInsulin": "giveqcctsicslyqlenycn",
      "cInsulin":"rreaedlqvgqvelgggpgagslqplalegslqkr"
   "weights":{
      "A":89.09,
      "C":121.16,
      "D":133.10,
      "E":147.13,
      "F":165.19,
      "G":75.07,
      "H":155.16,
      "I":131.17,
      "K":146.19,
      "L":131.17,
      "M":149.21,
      "N":132.12,
      "P":115.13,
      "Q":146.15,
      "R":174.20,
      "S":105.09,
      "T":119.12,
      "V":117.15,
      "W":204.23,
      "Y":181.19
   },
   "molecularWeightInsulinActual":5807.63
}
```

- 1. Para guardar el archivo como **insulin.json** en la carpeta **files** (archivos), seleccione **File -> Save As...** (Archivo -> Guardar como...).
- 2. En la ventana emergente **Save As** (Guardar como) en **Filename**: (Nombre del archivo:), escriba insulin.json.
- 3. En **Folder**: (Carpeta:), escriba **files** o seleccione la carpeta **files** (archivos).

# Ejercicio 2: Creación del módulo controlador de archivos JSON

En esta tarea, creará un módulo que lee el archivo JSON y devuelve el documento JSON.

- 1. Elija el archivo jsonFileHandler.py.
- 2. Importe JSON para comenzar el trabajo:

```
import json
```

1. Defina la función que leerá el archivo:

```
def readJsonFile(fileName):
```

1. Debajo de la definición de archivo, agregue una variable de datos como una cadena vacía:

```
data=""
```

1. Para el cuerpo de la función, abra el archivo JSON con la función **open** y analice el archivo con **json.load**.

```
def readJsonFile(fileName):
    data = ""
    with open('files/insulin.json') as json_file:
        data = json.load(json_file)
    return data
```

La función **open** devuelve un controlador de archivos al archivo **files/insulin.json**.

**json.load** lee el archivo JSON y devuelve el contenido como un diccionario de Python.

1. Agregue un bloque **try/except** para hacer que esta función sea más confiable:

```
import json

def readJsonFile(fileName):
    data = ""
    try:
        with open(fileName) as json_file:
            data = json.load(json_file)
    except IOError:
        print("Could not read file")
    return data
```

En caso de que el archivo no se pueda abrir, el programa mostrará el error *Could not read file* (No se pudo leer el archivo).

La cadena de **datos** devuelta está vacía en caso de que falle el método "Open File".

Creó un módulo **jsonFileHandle** que puede importar en otros archivos Python para acceder a la función **readJsonFile**.

# Ejercicio 3: Creación del programa principal

Crea el programa principal que analiza los datos JSON y calcula el peso molecular como lo hizo en un laboratorio anterior.

1. Primero, importe el módulo **jsonFileHandle**. Abra el archivo **calc\_weight\_json.py** y agregue lo siguiente:

```
import jsonFileHandler
```

1. Recupere los datos JSON y almacénelos en una variable de **datos**.

```
data = jsonFileHandler.readJsonFile('files/insulin.json')
```

1. Compruebe si los datos devueltos no están vacíos y obtenga los datos de insulina.

```
if data != "" :
    bInsulin = data['molecules']['bInsulin']
    aInsulin = data['molecules']['aInsulin']
    insulin = bInsulin + aInsulin
    molecularWeightInsulinActual = data['molecularWeightInsulinActual']
    print('bInsulin: ' + bInsulin)
    print('aInsulin: ' + aInsulin)
    print('molecularWeightInsulinActual: ' + str(molecularWeightInsulinActual))
else:
    print("Error. Exiting program")
```

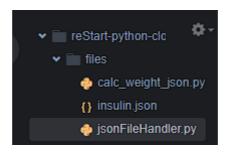
1. Puede ejecutar el programa para ver si los datos se recuperaron bien. Los resultados deberían ser los siguientes:

```
bInsulin: fvnqhlcgshlvealylvcgergffytpkt
aInsulin: giveqcctsicslyqlenycn
molecularWeightInsulinActual: 5807.63
```

1. También puede probar lo que sucede si no se encuentra el archivo. Por ejemplo, cambie el nombre del archivo a 'files/insuline.json' y ejecute el programa. Recibirá el siguiente mensaje:

```
Could not read file
Error. Exiting program
```

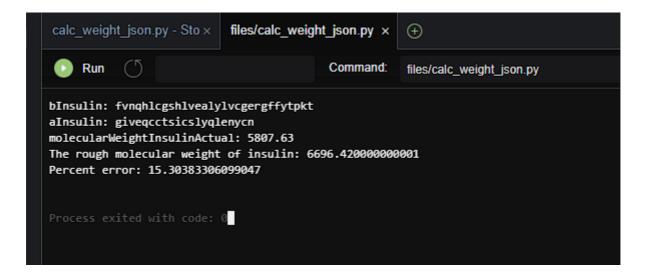
 Deshaga el último cambio de manera que el archivo se llame files/insulin.json de nuevo.



2. En la sección **if** del código debajo de la última función **print**, agregue el siguiente código:

1. Ejecute el programa. Obtendrá lo siguiente:

```
bInsulin: fvnqhlcgshlvealylvcgergffytpkt
aInsulin: giveqcctsicslyqlenycn
molecularWeightInsulinActual: 5807.63
The rough molecular weight of insulin: 6696.420000000001
Percent error: 15.30383306099047
```



### Finalizar laboratorio

¡Felicitaciones! Ha llegado al final del laboratorio.

1. Elige **End Lab** (Finalizar laboratorio) en la parte superior de esta página y, a continuación, selecciona Yes (Sí) para confirmar que deseas finalizar el laboratorio.

Un panel muestra el mensaje *DELETE has been initiated... You may close this message box now* (Se ha iniciado la ELIMINACIÓN... Ya puedes cerrar este mensaje).

1. Aparece brevemente el mensaje *Ended AWS Lab Successfully* (El laboratorio de AWS finalizó correctamente), que indica que el laboratorio ha finalizado.