TOPICOS DE INGENIERIA DE SOFTWARE

2023

Alumnas:
Marcela Paladino.
María Cecilia Pezzini.

FACULTAD DE INFORMATICA

UNIVERSIDAD NACIONAL DE LA PLATA

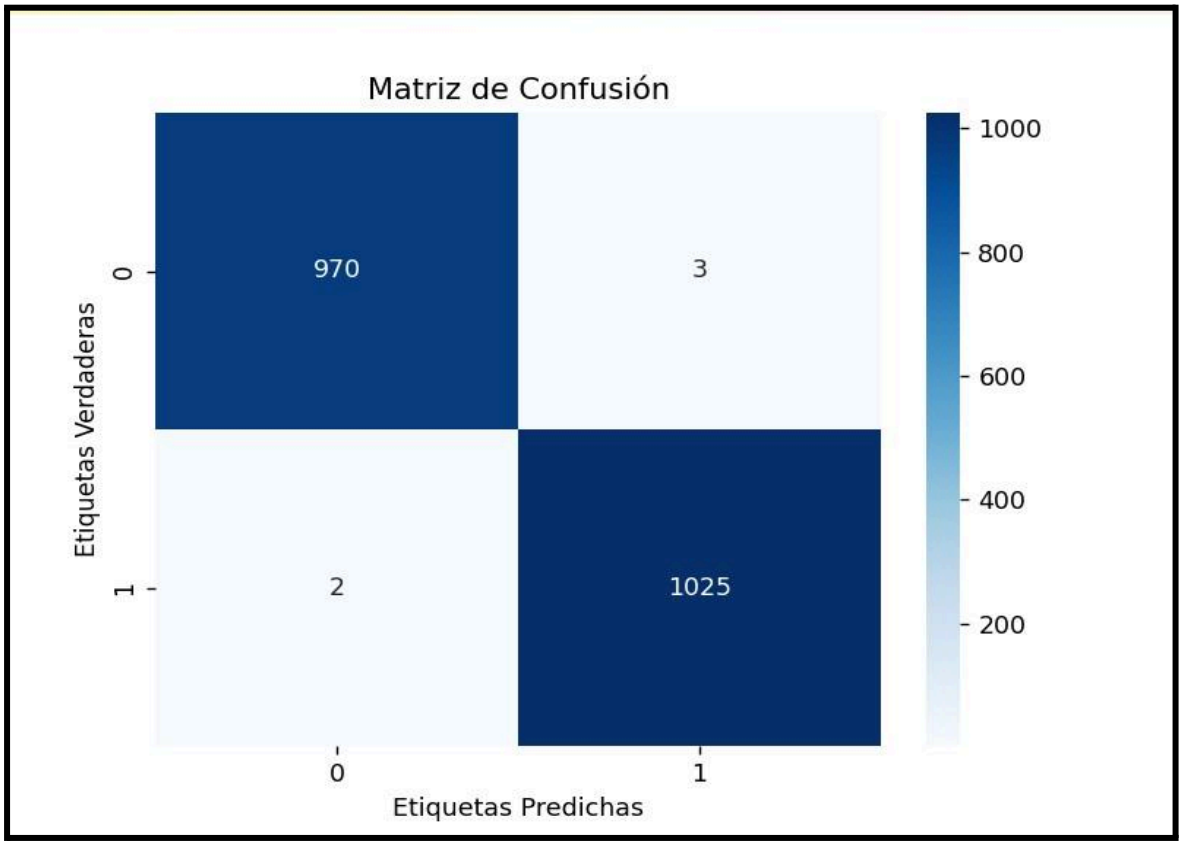
Índice de Contenidos:

1.- Instructivo para correr el modelo de Machine Learning.

El instructivo para correr el modelo de ML, se encuentra en el repositorio de GitHub, y se llama Instructivo_entrenamiento_ML.md.

Resultados del entrenamiento:

Matriz de Confusión



- Verdadero Positivo (TP): 970 instancias.

- Falso Negativo (FN): 3 instancias.
- Falso Positivo (FP): 2 instancias.
- Verdadero Negativo (TN): 1025 instancias.

La interpretación de la matriz de confusión:

- Diagonal Principal (Superior Izquierda a Inferior Derecha): Muestra las instancias que el modelo clasificó correctamente (TP=970 y TN=1025).
- Fuera de la Diagonal Principal: Muestra las instancias que el modelo clasificó incorrectamente (FP=2 y FN=3).

1. **Precisión (Accuracy):**

$$\text{Precisión} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{970 + 1025}{970 + 1025 + 2 + 3} \approx \frac{1995}{2000} = 0.99$$

2. **Sensibilidad (Recall o Tasa de Verdaderos Positivos):**

$$\text{Sensibilidad} = \frac{TP}{TP + FN} = \frac{970}{970 + 3} \approx \frac{970}{973} \approx 0.997$$

3. **Especificidad (Tasa de Verdaderos Negativos):**

$$\text{Especificidad} = \frac{TN}{TN + FP} = \frac{1025}{1025 + 2} \approx \frac{1025}{1027} \approx 0.9981$$

- La precisión indica la proporción de predicciones correctas en el conjunto total de predicciones. En este caso, es bastante alta, lo que sugiere que el modelo tiene un buen rendimiento en términos de precisión global.
- La sensibilidad mide la proporción de instancias positivas reales que fueron correctamente identificadas por el modelo. En este caso, es también muy alto, indicando que el modelo identifica bien las instancias positivas.
- La especificidad mide la proporción de instancias negativas reales que fueron correctamente identificadas por el modelo. En este caso, es muy alto, lo que sugiere que el modelo identifica bien las instancias negativas.

- A continuación se muestra el resultado del entrenamiento luego de correrlo:

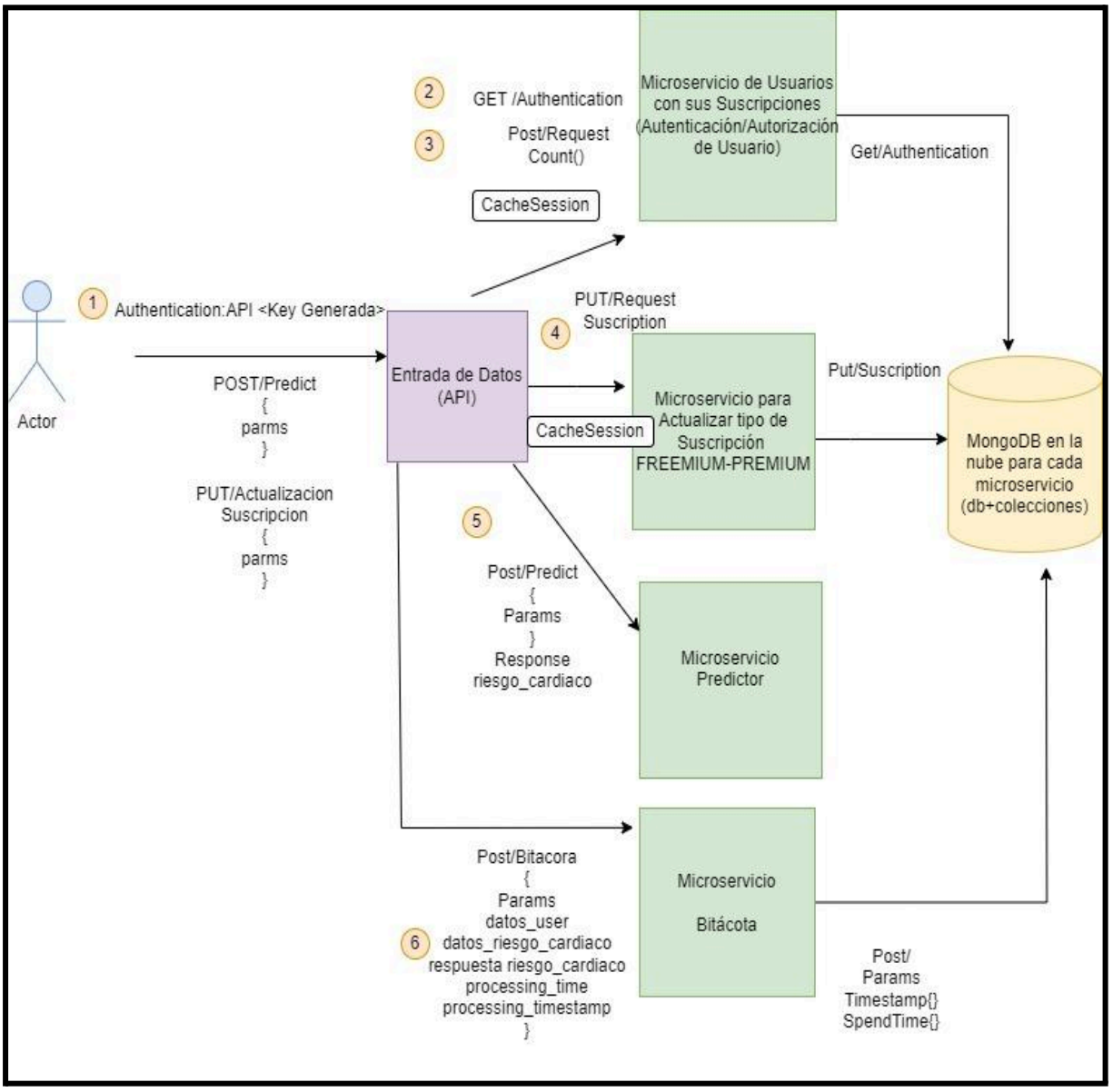
```
125/125 - 0s - loss: 0.0402 - accuracy: 0.9772 - 173ms/epoch - 1ms/step
Epoch 90/100
125/125 - 0s - loss: 0.0981 - accuracy: 0.9597 - 153ms/epoch - 1ms/step
Epoch 91/100
125/125 - 0s - loss: 0.0589 - accuracy: 0.9688 - 127ms/epoch - 1ms/step
Epoch 92/100
125/125 - 0s - loss: 0.0468 - accuracy: 0.9734 - 127ms/epoch - 1ms/step
Epoch 93/100
125/125 - 0s - loss: 0.0385 - accuracy: 0.9761 - 115ms/epoch - 922us/step
Epoch 94/100
125/125 - 0s - loss: 0.0447 - accuracy: 0.9760 - 116ms/epoch - 929us/step
Epoch 95/100
125/125 - 0s - loss: 0.0422 - accuracy: 0.9768 - 100ms/epoch - 803us/step
Epoch 96/100
125/125 - 0s - loss: 0.0388 - accuracy: 0.9761 - 111ms/epoch - 884us/step
Epoch 97/100
125/125 - 0s - loss: 0.0586 - accuracy: 0.9684 - 111ms/epoch - 885us/step
Epoch 98/100
125/125 - 0s - loss: 0.0354 - accuracy: 0.9774 - 110ms/epoch - 879us/step
Epoch 99/100
125/125 - 0s - loss: 0.0409 - accuracy: 0.9745 - 141ms/epoch - 1ms/step
Epoch 100/100
125/125 - 0s - loss: 0.0396 - accuracy: 0.9724 - 128ms/epoch - 1ms/step
63/63 [==============================] - 0s 1ms/step
63/63 [==============================] - 0s 770us/step
Accuracy: 0.9975
Precision: 0.9970817120622568
Recall: 0.9980525803310614
F1 Score: 0.9975669099756691
```

Estas métricas sugieren:

- **Accuracy (Precisión)**: 0.9975
  El 99.75% de las predicciones del modelo son correctas en el conjunto de datos total. Esto sugiere que el modelo tiene un rendimiento alto en términos de clasificación correcta.
- **Precisión (Precisión Positiva)**: 0.9971
  El 99.71% de las instancias clasificadas como positivas por el modelo son realmente positivas. Esto indica que el modelo tiene una alta precisión en la identificación de instancias positivas.
- **Recall (Sensibilidad o Tasa de Verdaderos Positivos)**: 0.9976
  El 99.76% de las instancias positivas reales fueron correctamente identificadas por el modelo. Esto sugiere que el modelo tiene capacidad para capturar instancias positivas.

2.- Diagrama de Arquitectura de Microservicios.

El diagrama muestra la arquitectura desarrollada para la implementación de lo solicitado en el trabajo final.



3.- Test de funcionamiento.

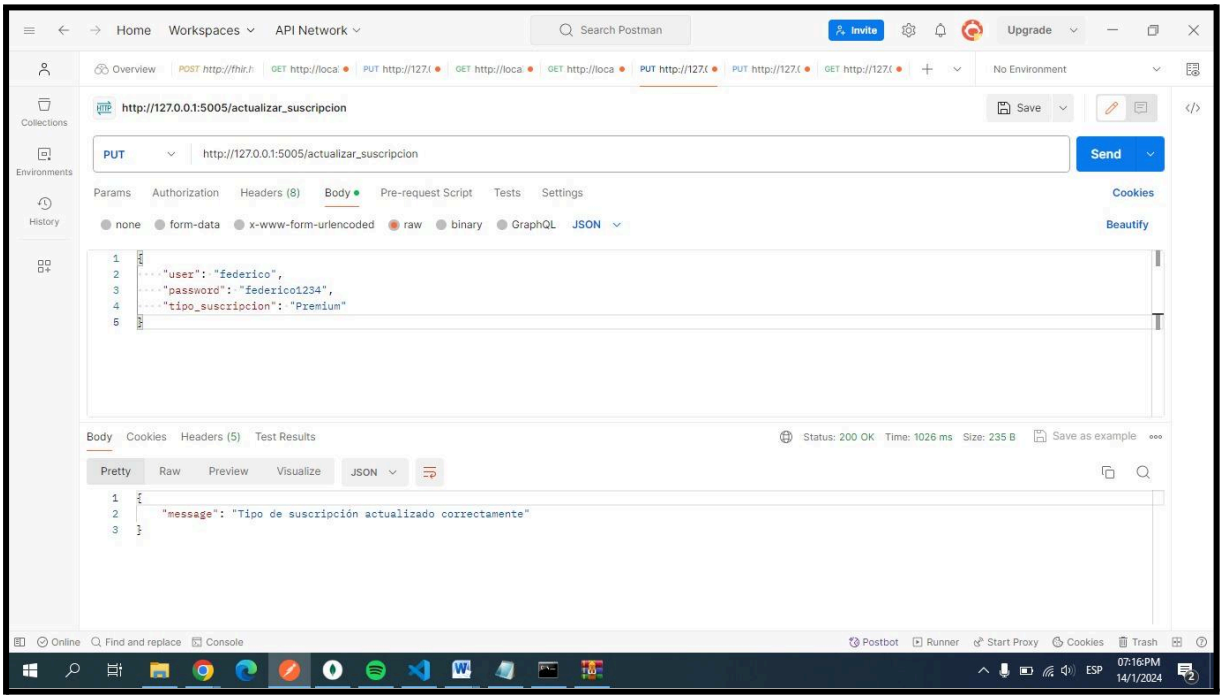
3.1 Test para recuperar el tipo de suscripción.

Nombre del archivo: recuperaSuscripcion.py
GET:http://localhost:5001/obtener_suscripcion/ceci/ceci1234 (tipo suscripción:Premium)
GET:http://localhost:5001/obtener_suscripcion/marce/marce1234 (tipo suscripción:Freemium)

3.2 Test para cambiar el tipo de suscripción.

Nombre del archivo: cambiaSuscripcion.py
Parámetros: PUT:http://localhost:5000/llamar_actualizar_suscripcion

```
{
    "user": "federico",
    "password": "federico1234",
    "tipo_suscripcion": "Freemium"
}
```



3.3 Test Predicción.

Nombre del archivo: app_riesgocardiaco.py
POST: http://localhost:5002/predict
Parámetros1:

```
  "nivel_colesterol": 1.8,
 "presion_arterial": 1.2,
 "glucosa": 1.0,
 "edad": 40,
 "sobrepeso": 0,
 "tabaquismo": 0
}
```

Respuesta:

"prediction": "Paciente sin riesgo cardíaco"

Parámetros2:


```
{
  "nivel_colesterol": 1.8,
 "presion_arterial": 1.2,
 "glucosa": 1.9,
 "edad": 50,
 "sobrepeso": 1,
 "tabaquismo": 0
}
```


Respuesta:


```
 "prediction": "Paciente en riesgo cardíaco"
```


3.4 Test de aplicación.

El archivo de la aplicación se llama **index.py.** Es quien define y configura los endpoints para ejecutar Flask. Se puede utilizar Postman para interactuar con los endpoints definidos.
Se deberá tener en ejecución previamente: bitacora.py, recuperaSuscripcion.py, app_riesgocardiaco.py, actualizaSuscripcion.py. Podrán descargarse de GitHub.

- **GET:** http://localhost:5000/index?user=marce&password=marce1234&nivel_colesterol=1.0&presion_arterial=1.3&glucosa=0.7&edad=50&sobrepeso=1&tabaquismo=1

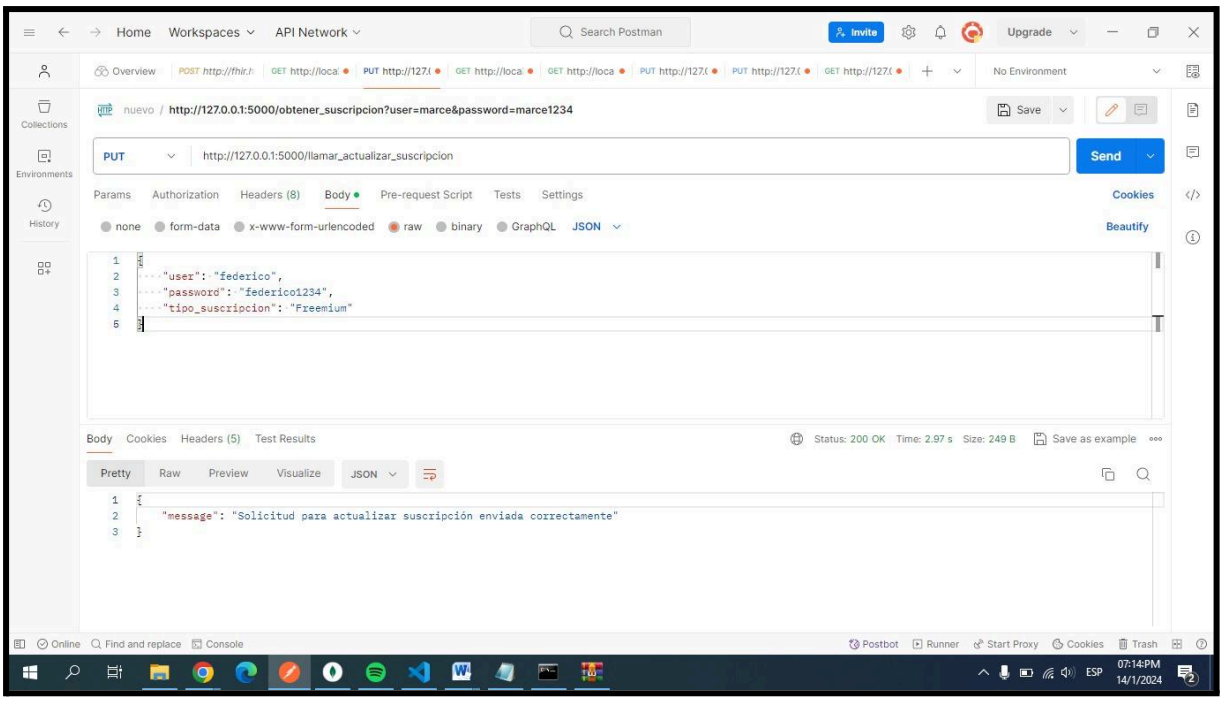
En los archivos especificados a continuación (disponibles en GitHub), se encuentran las respuestas obtenidas en los diferentes test:

response1: Paciente CON Riesgo Cardiaco.
response2: Paciente SIN Riesgo Cardiaco.

response3: Respuesta obtenida: "Se ha alcanzado el límite de consultas para marce/marce1234."
response4: Nivel de colesterol fuera del rango permitido (1.0, 3.0).
response5: Respuesta obtenida: "Usuario No Encontrado."

- **PUT**

  http://127.0.0.1:5000/llamar_actualizar_suscripcion



4.- Architecture Decision Records (ADR)

A continuación se recopilan las decisiones de arquitectura adoptadas durante el desarrollo del proyecto. Los mismos están disponibles en GitHub.

Estos son:

ADR #1: Arquitectura basada en Microservicios.
ADR #2: Escenario de Respuesta a Fallos.

ADR #3: Red Neuronal para Predicción de Riesgo Cardíaco.
ADR #4: Uso de Caché para Datos de Lectura y Baja Volatilidad.
ADR #5: Elección de Python y MongoDB Atlas.
ADR #6: Uso del Patrón Gateway para Orquestación.

5.- Consideraciones.

- Hemos decidido no limpiar la base de datos para que se pueda preservar y visualizar el trabajo realizado durante el proceso de desarrollo.
- Para poder acceder a los datos, dejamos disponible el siguiente archivo: base.py, disponible en GitHub.