

d90cc5a1-f8eb-4dcd-97ed-8a3a863b7f1c

November 16, 2024

1 ¡Hola !

Te escribe Lisandro Saez, soy revisor de código en Tripletten y tengo el agrado de revisar el proyecto que entregaste.

Para simular la dinámica de un ambiente de trabajo, si veo algún error, en primer instancia solo los señalaré, dándote la oportunidad de encontrarlos y corregirlos por tu cuenta. En un trabajo real, el líder de tu equipo hará una dinámica similar. En caso de que no puedas resolver la tarea, te daré una información más precisa en la próxima revisión.

Encontrarás mis comentarios más abajo - **por favor, no los muevas, no los modifiques ni los borres.**

¿Cómo lo voy a hacer? Voy a leer detenidamente cada una de las implementaciones que has llevado a cabo para cumplir con lo solicitado. Verás los comentarios de esta forma:

Comentario del revisor

Si todo está perfecto.

Comentario del revisor

Si tu código está bien pero se puede mejorar o hay algún detalle que le hace falta. Se aceptan uno o dos comentarios de este tipo en el borrador, pero si hay más, deberías hacer las correcciones. Es como una tarea de prueba al solicitar un trabajo: muchos pequeños errores pueden hacer que un candidato sea rechazado.

Comentario del revisor

Si de pronto hace falta algo o existe algún problema con tu código o conclusiones.

Puedes responderme de esta forma (no te preocupes, no es obligatorio):

Respuesta del estudiante

Hola, muchas gracias por tus comentarios y la revisión.

¡Empecemos!

Como parte del equipo de análisis, lo primero que debes hacer es evaluar la calidad de una muestra de datos recopilados y prepararla para analizarla posteriormente. Después, en la segunda parte de este proyecto en el segundo sprint, desarrollarás más tus habilidades y harás tu primer análisis completo, respondiendo a las necesidades del cliente.

Estos son los datos que el cliente nos proporcionó. Tienen el formato de una lista de Python, con las siguientes columnas de datos:

- **user_id:** Identificador único para cada usuario.
- **user_name:** El nombre del usuario.
- **user_age:** La edad del usuario.
- **fav_categories:** Categorías favoritas de los artículos que compró el usuario, como 'ELECTRONICS', 'SPORT' y 'BOOKS' (ELECTRÓNICOS, DEPORTES y LIBROS), etc.
- **total_spendings:** Una lista de números enteros que indican la cantidad total gastada en cada una de las categorías favoritas.

```
[ ]: users = [  
    ['32415', ' mike_reed ', 32.0, ['ELECTRONICS', 'SPORT', 'BOOKS'], [894, 213, 173]],  
    ['31980', 'kate morgan', 24.0, ['CLOTHES', 'BOOKS'], [439, 390]],  
    ['32156', ' john doe ', 37.0, ['ELECTRONICS', 'HOME', 'FOOD'], [459, 120, 99]],  
    ['32761', 'SAMANTHA SMITH', 29.0, ['CLOTHES', 'ELECTRONICS', 'BEAUTY'], [299, 679, 85]],  
    ['32984', 'David White', 41.0, ['BOOKS', 'HOME', 'SPORT'], [234, 329, 243]],  
    ['33001', 'emily brown', 26.0, ['BEAUTY', 'HOME', 'FOOD'], [213, 659, 79]],  
    ['33767', ' Maria Garcia', 33.0, ['CLOTHES', 'FOOD', 'BEAUTY'], [499, 189, 63]],  
    ['33912', 'JOSE MARTINEZ', 22.0, ['SPORT', 'ELECTRONICS', 'HOME'], [259, 549, 109]],  
    ['34009', 'lisa wilson ', 35.0, ['HOME', 'BOOKS', 'CLOTHES'], [329, 189, 329]],  
    ['34278', 'James Lee', 28.0, ['BEAUTY', 'CLOTHES', 'ELECTRONICS'], [189, 299, 579]],  
]
```

2 Paso 1

Store 1 tiene como objetivo garantizar la coherencia en la recopilación de datos. Como parte de esta iniciativa, se debe evaluar la calidad de los datos recopilados sobre los usuarios y las usuarias. Te han pedido que revises los datos recopilados y propongas cambios. A continuación verás datos sobre un usuario o una usuaria en particular; revisa los datos e identifica cualquier posible problema.

```
[1]: user_id = '32415'  
      user_name = ' mike_reed '  
      user_age = 32.0  
      fav_categories = ['ELECTRONICS', 'SPORT', 'BOOKS']
```

Opciones:

1. Se debe cambiar el tipo de dato de **user_id** de cadena a entero.
2. La variable **user_name** contiene una cadena que tiene espacios innecesarios y un guion bajo

entre el nombre y el apellido.

3. El tipo de dato de `user_age` es correcto y no hay necesidad de convertirlo.
4. La lista `fav_categories` contiene cadenas en mayúsculas. En lugar de ello, debemos convertir los valores de la lista en minúsculas.

Para cada una de las opciones, escribe en la siguiente celda markdown si la identificaste como un problema real en los datos o no. Justifica tu razonamiento. Por ejemplo, si crees que la primera opción es correcta, escríbelo y explica por qué piensas que es correcta.

#Ob1. No es necesario cambiar el dato a entero debido a que es información asignada, que no participa en operaciones matemáticas. #Ob2. Eliminar espacios innecesarios y guiones para uniformidad de formato de la información. #Ob3. Posiblemente no representa problemas el tipo float sin embargo la edad no suele ser representada así por lo que sería conveniente cambiar a int. #Ob4. Es recomendable mantener el formato de la información uniforme por lo que, lo conveniente es utilizar minúsculas.

Comentario del revisor

Gran trabajo! Es importante que antes de encarar cualquier problema que queramos resolver con Python, nos paremos a pensar bien en qué consisten los datos con los que vamos a trabajar!

3 Paso 2

Vamos a implementar los cambios que identificamos. Primero, necesitamos corregir los problemas de la variable `user_name` Como vimos, tiene espacios innecesarios y un guion bajo como separador entre el nombre y el apellido; tu objetivo es eliminar los espacios y luego reemplazar el guion bajo con el espacio.

```
[2]: user_name = ' mike_reed '  
user_name =user_name.strip()  
user_name =user_name.replace('_', ' ')  
  
print(user_name)
```

mike reed

*****Hint*****

Existe un método, `strip()`, que puede eliminar espacios al principio y al final de una cadena. Además, el método `replace()` se puede usar para reemplazar una parte de una cadena. En este caso, queremos reemplazar los guiones bajos (`_`) con espacios.

4 Paso 3

Luego, debemos dividir el `user_name` (nombre de usuario o usuaria) actualizado en dos subcadenas para obtener una lista que contenga dos valores: la cadena para el nombre y la cadena para el apellido.

```
[3]: user_name = 'mike reed'
      name_split = user_name.split()

      print(name_split)
```

```
['mike', 'reed']
```

```
*****Hint*****
```

El método `split()` se utiliza para dividir una cadena. Por defecto, utiliza un espacio como separador.

Comentario del revisor

Gran uso de los métodos `split()` y `strip()`!

5 Paso 4

¡Genial! Ahora debemos trabajar con la variable `user_age`. Como ya mencionamos, esta tiene un tipo de datos incorrecto. Arreglemos este problema transformando el tipo de datos y mostrando el resultado final.

```
[4]: user_age = 32.0
      user_age = int(user_age)

      print(user_age)
```

```
32
```

```
*****Hint*****
```

¿Qué tipo de datos eliminará la parte de coma flotante?

6 Paso 5

Como sabemos, los datos no siempre son perfectos. Debemos considerar escenarios en los que el valor de `user_age` no se pueda convertir en un número entero. Para evitar que nuestro sistema se bloquee, debemos tomar medidas con anticipación.

Escribe un código que intente convertir la variable `user_age` en un número entero y asigna el valor transformado a `user_age_int`. Si el intento falla, mostramos un mensaje pidiendo al usuario o la usuaria que proporcione su edad como un valor numérico con el mensaje: `Please provide your age as a numerical value.` (Proporcione su edad como un valor numérico.)

```
[5]: user_age = 'treinta y dos'

      try:
          int(user_age)
      except:
          print("Please provide your age as a numerical value.")
```

Please provide your age as a numerical value.

*****Hint*****

Utiliza un bloque `try-except` para intentar la conversión; si falla, proporciona un mensaje claro indicando que la entrada debe ser numérica.

7 Paso 6

El equipo de dirección de Store 1 te pidió ayudarles a organizar los datos de sus clientes para analizarlos y gestionarlos mejor.

Tu tarea es ordenar esta lista por ID de usuario de forma ascendente para que sea más fácil acceder a ella y analizarla.

```
[6]: users = [
    ['32415', 'mike_reed ', 32.0, ['ELECTRONICS', 'SPORT', 'BOOKS'], [894, 213, 173]],
    ['31980', 'kate morgan', 24.0, ['CLOTHES', 'BOOKS'], [439, 390]],
    ['32156', 'john doe ', 37.0, ['ELECTRONICS', 'HOME', 'FOOD'], [459, 120, 99]],
    ['32761', 'SAMANTHA SMITH', 29.0, ['CLOTHES', 'ELECTRONICS', 'BEAUTY'], [299, 679, 85]],
    ['32984', 'David White', 41.0, ['BOOKS', 'HOME', 'SPORT'], [234, 329, 243]],
    ['33001', 'emily brown', 26.0, ['BEAUTY', 'HOME', 'FOOD'], [213, 659, 79]],
    ['33767', 'Maria Garcia', 33.0, ['CLOTHES', 'FOOD', 'BEAUTY'], [499, 189, 63]],
    ['33912', 'JOSE MARTINEZ', 22.0, ['SPORT', 'ELECTRONICS', 'HOME'], [259, 549, 109]],
    ['34009', 'lisa wilson ', 35.0, ['HOME', 'BOOKS', 'CLOTHES'], [329, 189, 329]],
    ['34278', 'James Lee', 28.0, ['BEAUTY', 'CLOTHES', 'ELECTRONICS'], [189, 299, 579]],
]

users.sort()

print(users)
```

```
[['31980', 'kate morgan', 24.0, ['CLOTHES', 'BOOKS'], [439, 390]], ['32156', 'john doe ', 37.0, ['ELECTRONICS', 'HOME', 'FOOD'], [459, 120, 99]], ['32415', 'mike_reed ', 32.0, ['ELECTRONICS', 'SPORT', 'BOOKS'], [894, 213, 173]], ['32761', 'SAMANTHA SMITH', 29.0, ['CLOTHES', 'ELECTRONICS', 'BEAUTY'], [299, 679, 85]], ['32984', 'David White', 41.0, ['BOOKS', 'HOME', 'SPORT'], [234, 329, 243]], ['33001', 'emily brown', 26.0, ['BEAUTY', 'HOME', 'FOOD'], [213, 659, 79]], ['33767', 'Maria Garcia', 33.0, ['CLOTHES', 'FOOD', 'BEAUTY'], [499, 189, 63]], ['33912', 'JOSE MARTINEZ', 22.0, ['SPORT', 'ELECTRONICS', 'HOME'], [259, 549, 109]], ['34009', 'lisa wilson ', 35.0, ['HOME', 'BOOKS', 'CLOTHES'], [329,
```

```
189, 329]], ['34278', 'James Lee', 28.0, ['BEAUTY', 'CLOTHES', 'ELECTRONICS'],
[189, 299, 579]]
```

*****Hint*****

Puedes utilizar el método `sort()` en la lista de usuarios para ordenarla de forma ascendente.

8 Paso 7

Tenemos la información de los hábitos de consumo de nuestros usuarios, incluyendo la cantidad gastada en cada una de sus categorías favoritas. La dirección está interesada en conocer la cantidad total gastada por el usuario.

Calculemos este valor y despluguémoslo.

```
[7]: fav_categories_low = ['electronics', 'sport', 'books']
    spendings_per_category = [894, 213, 173]

    total_amount = sum(spendings_per_category)

    print(total_amount)
```

1280

*****Hint*****

¿Cuáles son los tres métodos que se pueden aplicar a una lista para calcular sus valores mínimo, máximo y total?

Comentario del Revisor

Otra forma de hacer esto mismo podía ser utilizando `total_amount = spendings_per_category[0] + spendings_per_category[1] + spendings_per_category[2]`

Te felicito por haber intentado la manera más rápida de hacerlo, pero ten en cuenta también esta forma de sumar ya que puede servirte en algún momento donde `sum()` quizás no sea el mejor acercamiento al problema.

9 Paso 8

La dirección de la empresa nos pidió pensar en una manera de resumir toda la información de un usuario. Tu objetivo es crear una cadena formateada que utilice información de las variables `user_id`, `user_name` y `user_age`.

Esta es la cadena final que queremos crear: `User 32415 is mike who is 32 years old.` (El usuario 32415 es Mike, quien tiene 32 años).

```
[8]: user_id = '32415'
    user_name = ['mike', 'reed']
    user_age = 32

    user_info = f"User {user_id} is {user_name[0]} who is {user_age} years old."
```

```
print(user_info)
```

User 32415 is mike who is 32 years old.

*****Hint*****

Para crear una cadena, puedes utilizar el método `format()` o f-string. Para extraer el nombre de la lista `user_name`, puedes utilizar la segmentación.

10 Paso 9

La dirección también quiere una forma fácil de conocer la cantidad de clientes con cuyos datos contamos. Tu objetivo es crear una cadena formateada que muestre la cantidad de datos de clientes registrados.

Esta es la cadena final que queremos crear: Hemos registrado datos de X clientes.

```
[9]: users = [
    ['32415', 'mike_reed ', 32.0, ['ELECTRONICS', 'SPORT', 'BOOKS'], [894, 213, 173]],
    ['31980', 'kate morgan', 24.0, ['CLOTHES', 'BOOKS'], [439, 390]],
    ['32156', 'john doe ', 37.0, ['ELECTRONICS', 'HOME', 'FOOD'], [459, 120, 99]],
    ['32761', 'SAMANTHA SMITH', 29.0, ['CLOTHES', 'ELECTRONICS', 'BEAUTY'], [299, 679, 85]],
    ['32984', 'David White', 41.0, ['BOOKS', 'HOME', 'SPORT'], [234, 329, 243]],
    ['33001', 'emily brown', 26.0, ['BEAUTY', 'HOME', 'FOOD'], [213, 659, 79]],
    ['33767', 'Maria Garcia', 33.0, ['CLOTHES', 'FOOD', 'BEAUTY'], [499, 189, 63]],
    ['33912', 'JOSE MARTINEZ', 22.0, ['SPORT', 'ELECTRONICS', 'HOME'], [259, 549, 109]],
    ['34009', 'lisa wilson ', 35.0, ['HOME', 'BOOKS', 'CLOTHES'], [329, 189, 329]],
    ['34278', 'James Lee', 28.0, ['BEAUTY', 'CLOTHES', 'ELECTRONICS'], [189, 299, 579]],
]

user_info = f"We have registered data on {len(users)} clients."
print(user_info)
```

We have registered data on 10 clients.

*****Hint*****

Para crear una cadena, puedes utilizar el método `format()` o f-string. Para extraer la cantidad de clientes en la lista, puedes utilizar la función que devuelve la longitud de la lista.

11 Paso 10

Apliquemos ahora todos los cambios a la lista de clientes. Para simplificar las cosas, te proporcionaremos una más corta. Debes: 1. Eliminar todos los espacios iniciales y finales de los nombres, así como cualquier guion bajo. 2. Convertir todas las edades en números enteros. 3. Separar todos los nombres y apellidos en una sublista.

Guarda la lista modificada como una nueva lista llamada `users_clean` y muéstrala en la pantalla.

```
[10]: users = [
    ['32415', ' mike_reed ', 32.0, ['ELECTRONICS', 'SPORT', 'BOOKS'], [894, 213, 173]],
    ['31980', 'kate morgan', 24.0, ['CLOTHES', 'BOOKS'], [439, 390]],
    ['32156', ' john doe ', 37.0, ['ELECTRONICS', 'HOME', 'FOOD'], [459, 120, 99]]
]

users_clean = [user_clean_1, user_clean_2, user_clean_3]

# Procesa al primer usuario
user_name_1 = users[0]
user_age_1 = int(user_name_1[2])
user_name_1_2 = (user_name_1[1].strip()).replace('_', ' ')
user_clean_1 = f"{users[0]}{[user_name_1_2, user_age_1]}"

#test
print(user_name_1)
print(user_age_1)
print(user_name_1_2)
print(user_clean_1)

# Procesa al segundo usuario
user_name_2 = users[1]
user_age_2 = int(user_name_2[2])
user_name_2_2 = (user_name_2[1].strip()).replace('_', ' ')
user_clean_2 = f"{users[1]}{[user_name_2_2, user_age_2]}"

#test
print(user_name_2)
print(user_age_2)
print(user_name_2_2)
print(user_clean_2)

# Procesa al tercer usuario
user_name_3 = users[2]
```



```

user_age_3 =int(user_name_3[2])
user_name_3_2 =(user_name_3[1].strip()).replace('_', ' ')
user_clean_3=f"{users[2]}{[user_name_3_2,user_age_3]}"

#test
print(user_name_3)
print(user_age_3)
print(user_name_3_2)
print(user_clean_3)

print(users_clean)

#Dudas:

# Procesa al primer usuario - (dudas en el caso de cualquier
↳usuario)
user_name_1 = # escribe tu código aquí - (traje la lista del usuario a
↳corregir)
user_age_1 = # escribe tu código aquí - (se corrige el formato de edad)
user_name_1 = # escribe tu código aquí - (se corrigen espacios y
↳guiones), (por qué repetir el nombre de la variable "user_name_1"? Las
↳renombré agregando al final "_2")
users_clean.# escribe tu código aquí - (las tomé como variables en
↳cada usuario agregando "_1", 2 o 3 según el usuario)(es difícil apegarse a
↳las instrucciones ya que la función append sólo puede agregar un dato al
↳final y el resultado esperado anuncia la sustitución de los datos en el
↳orden original lo cual requeriría las funciones pop e insert)

users_clean - (a partir del punto anterior,
↳el objetivo presentar la lista definitiva no es eficaz)

```

Cell In[10], line 53

```

user_name_1 = # escribe tu código aquí - (traje la lista del usuario,
↳a corregir)

```

SyntaxError: invalid syntax

Comentario del Revisor

Prar próximos sprints, recuerda que todos tus bloques de código deben correr sin errores.

*****Hint*****

Para procesar a cada usuario, comienza por acceder a los elementos requeridos de la lista de usuarios. Utiliza el método `strip()` para eliminar los espacios iniciales y finales y el método `replace('_', ' ')` para reemplazar los guiones bajos por espacios en los nombres. Convierte la edad a un número entero utilizando `int()`. Separa el nombre completo en nombre y apellido utilizando el método

`split()`. Por último, **append** (agrega) los datos limpios a la lista `users_clean`.

Comentario del revisor

Felicitaciones por el excelente trabajo que hiciste durante todo el sprint!
