



# MAESTRIA EN INTELIGENCIA ARTIFICIAL APLICADA

## Pruebas de software y aseguramiento de la calidad

### 6.2 Ejercicio de programación 3

Profesor titular:

Dr. Gerardo Padilla Zárate

Estudiante:

Erick Johan Nájera Olivero (A01793506)

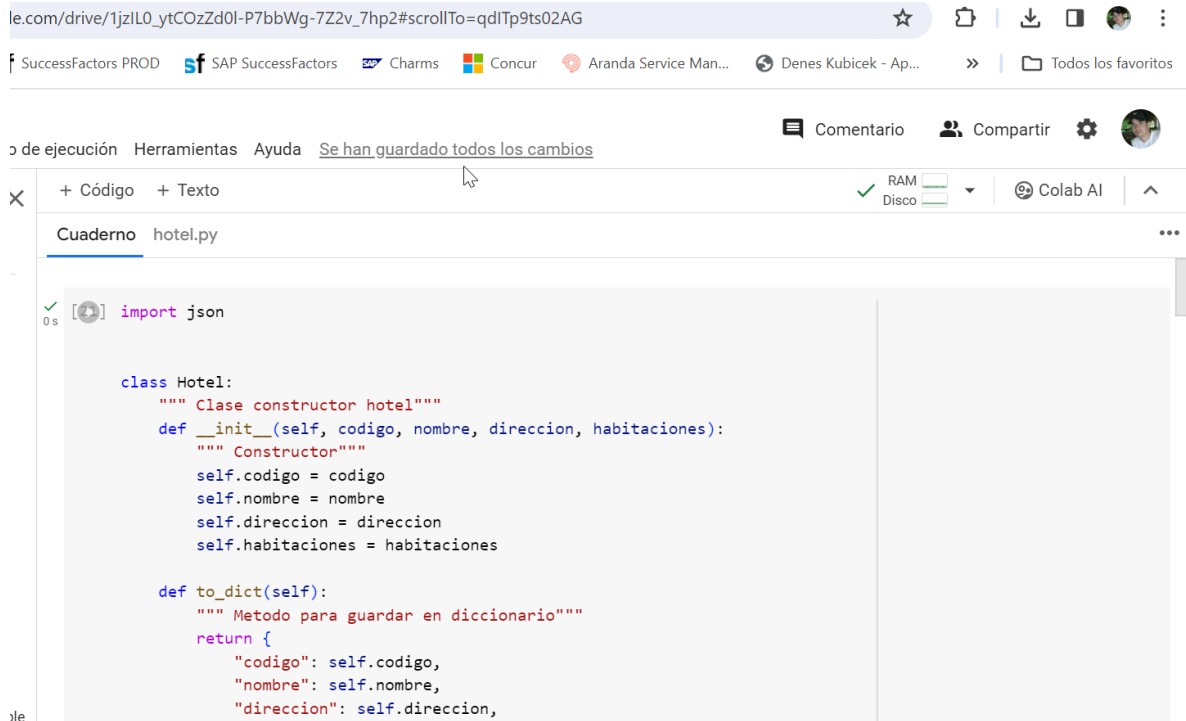
Febrero 18, 2024

## CLASES

**Req 1. Implement a set of classes in Python that implements two abstractions:**

### 1. Hotel

Ejemplo código fuente implementación clase hotel:



```
import json

class Hotel:
    """ Clase constructor hotel"""
    def __init__(self, codigo, nombre, direccion, habitaciones):
        """ Constructor"""
        self.codigo = codigo
        self.nombre = nombre
        self.direccion = direccion
        self.habitaciones = habitaciones

    def to_dict(self):
        """ Metodo para guardar en diccionario"""
        return {
            "codigo": self.codigo,
            "nombre": self.nombre,
            "direccion": self.direccion,
```

Prueba funcionamiento clase hotel:

Se crean dos hoteles, se guardan en archivo de texto.

se prueba la consulta y se escribe en pantalla para el hotel 1 (1000).

Se realiza reserva de la habitación 101 en hotel 1 (1000). Se consulta nuevamente hotel con habitaciones disponibles.

Se intenta reservar una habitación no existente en hotel, se captura el error y se muestra mensaje en pantalla. "la habitación no está disponible en este hotel".

Se intenta reservar habitación en hotel que no existe (1008): se captura el error y se muestra mensaje en pantalla "El hotel no existe".



Comentario

Compartir

ejecución

Herramientas

Ayuda

Se han guardado todos los cambios

+ Código

+ Texto

RAM

Disco

Colab AI

Cuaderno

hotel.py

0 s

# Example usage

hotel\_manager = HotelManager("hotels.txt")

hotel1 = Hotel("1000", "Hotel A", "City A", [101, 102, 103])

hotel2 = Hotel("2000", "Hotel B", "City B", [201, 202, 203])

hotel\_manager.create\_hotel(hotel1)

hotel\_manager.create\_hotel(hotel2)

print(hotel\_manager.display\_hotel\_info("1000").to\_dict())

hotel\_manager.reserve\_room("1000", 101)

print(hotel\_manager.display\_hotel\_info("1000").to\_dict())

hotel\_manager.cancel\_reservation("1001", 101)

print(hotel\_manager.display\_hotel\_info("1000").to\_dict())

{'codigo': '1000', 'nombre': 'Hotel A', 'direccion': 'City A', 'habitaciones': [102, 103]}

Error al reservar habitación: La habitación no está disponible en este hotel.

{'codigo': '1000', 'nombre': 'Hotel A', 'direccion': 'City A', 'habitaciones': [102, 103]}

Error al cancelar la reserva: El hotel no existe.

{'codigo': '1000', 'nombre': 'Hotel A', 'direccion': 'City A', 'habitaciones': [102, 103]}

Se ejecuta librería Pylint, se realiza ajuste al código dando como resultado 10. De 10.

Comentario

Compartir

ejecución

Herramientas

Ayuda

Se han guardado todos los cambios

+ Código

+ Texto

RAM

Disco

Colab AI

Cuaderno

hotel.py

0 s

{'codigo': '1000', 'nombre': 'Hotel A', 'direccion': 'City A', 'habitaciones': [102, 103]}

Error al reservar habitación: La habitación no está disponible en este hotel.

{'codigo': '1000', 'nombre': 'Hotel A', 'direccion': 'City A', 'habitaciones': [102, 103]}

Error al cancelar la reserva: El hotel no existe.

{'codigo': '1000', 'nombre': 'Hotel A', 'direccion': 'City A', 'habitaciones': [102, 103]}

3

!pip install pylint

20

!pylint hotel.py

-----

Your code has been rated at 10.00/10 (previous run: 9.86/10, +0.14)

Se ejecuta librería Flake, se realizan los ajustes indicados obteniendo cero errores:

gle.com/drive/1jzIL0\_ytCoZZd0l-P7bbWg-7Z2v\_7hp2#scrollTo=EPR93Yc78Arh

SuccessFactors PROD SAP SuccessFactors Charms Concur Aranda Service Man... Denes Kubicek - Ap... Todos los favoritos

no de ejecución Herramientas Ayuda Comentario Compartir

+ Código + Texto
RAM Disco
Colab AI

Cuaderno hotel.py

[20]
Your code has been rated at 10.00/10 (previous run: 9.86/10, +0.14)

[25] !pip install flake8

0s !flake8 hotel.py

Se procede a crear el archivo test para las pruebas unitarias Unittest.

com/drive/1aLW0kcLI0P3INBsEDfx1BueL8IF9m6uH#scrollTo=cvGUuR3rB9X8

SuccessFactors PROD SAP SuccessFactors Charms Concur Aranda Service Man... Denes Kubicek - Ap... Todos los favoritos

de ejecución Herramientas Ayuda Se han guardado todos los cambios Comentario Compartir

+ Código + Texto
RAM Disco
Colab AI

Cuaderno hotel.py testhotel.py

```

1 # -*- coding: utf-8 -*-
2 """Untitled9.ipynb
3
4 Automatically generated by Colaboratory.
5
6 Original file is located at
7 | | https://colab.research.google.com/drive/1fPDyU4yTiXpGYTzJrmJ6WiKybWuUwtk
8 | |
9
10 import unittest
11 import os
12 from hotel import HotelManager
13
14 class TestHotelManager(unittest.TestCase):
15     def setUp(self):
16         # Crea un archivo de prueba para las pruebas
17         self.file_path = 'test_hoteles.csv'
18         self.manager = HotelManager(self.file_path)
19
20     def tearDown(self):

```

Se ejecuta código de las pruebas obteniendo resultados de acuerdo a lo especificado:

gle.com/drive/1aLW0kcLI0P3INBsEDfx1BueL8IF9m6uH#scrollTo=cvGUuR3rB9X8

SuccessFactors PROD SAP SuccessFactors Charms Concur Aranda Service Man... Denes Kubicek - Ap... Todos los favoritos

Comentario Compartir Se han guardado todos los cambios

+ Código + Texto RAM Disco Colab AI

Cuaderno hotel.py testhotel.py

```

!python testhotel.py

3000
...Habitacion reservada.
.
-----
Ran 4 tests in 0.002s

OK
  
```

Se ejecuta coverage para obtener el porcentaje de cubrimiento del código logrando un 88%.

com/drive/1aLW0kcLI0P3INBsEDfx1BueL8IF9m6uH#scrollTo=cvGUuR3rB9X8

SuccessFactors PROD SAP SuccessFactors Charms Concur Aranda Service Man... Denes Kubicek - Ap... Todos los favoritos

Comentario Compartir Se han guardado todos los cambios

+ Código + Texto RAM Disco Colab AI

Cuaderno hotel.py testhotel.py

```

[62] !pip install coverage

!coverage run -m unittest
!coverage report -m

3000
...Habitacion reservada.
.
-----
Ran 4 tests in 0.003s

OK
Name          Stmts  Miss  Cover   Missing
-----
hotel.py       66     11    83%    34, 38, 56, 58, 86, 92, 100-103, 105, 107
testhotel.py   36      1    97%      57
-----
TOTAL         102     12    88%
  
```

## 2. Customers

Se crea la clase customer para gestionar operaciones de creación, actualización, consulta y borrado de customers.

le.com/drive/1JGiEt9t4YfcNwQflcDjDR91DyFKSSo\_#scrollTo=eLdnSzJEtpl

SuccessFactors PROD SAP SuccessFactors Charms Concur Aranda Service Man... Denes Kubicek - Ap... Todos los favoritos

o de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

Comentario Compartir

```

+ Código + Texto
Cuaderno customer.py x
9
10 import csv
11
12
13 class Mcustomer:
14
15     """ Clase customer"""
16     def __init__(self, file_path):
17         """ Inicializar clase cargar parh"""
18         self.file_path = file_path
19
20     def crear_customer(self, idc, nombre, genero, correo, telefono):
21         """ Metodo crear customer"""
22         with open(self.file_path, 'a', newline='', encoding='utf-8') as file:
23             writer = csv.writer(file)
24             writer.writerow([idc, nombre, genero, correo, telefono])
25
26     def borrar_customer(self, idc):
27         """ Metodo borrar customer"""
28         with open(self.file_path, 'r', encoding='utf-8') as file:

```

Se realizan pruebas del uso de los métodos:

com/drive/1JGiEt9t4YfcNwQflcDjDR91DyFKSSo\_#scrollTo=eLdnSzJEtpl

SuccessFactors PROD SAP SuccessFactors Charms Concur Aranda Service Man... Denes Kubicek - Ap... Todos los favoritos

de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

Comentario Compartir

```

+ Código + Texto
Cuaderno customer.py
0 s
file_path = 'customers.csv'
manager = Mcustomer(file_path)

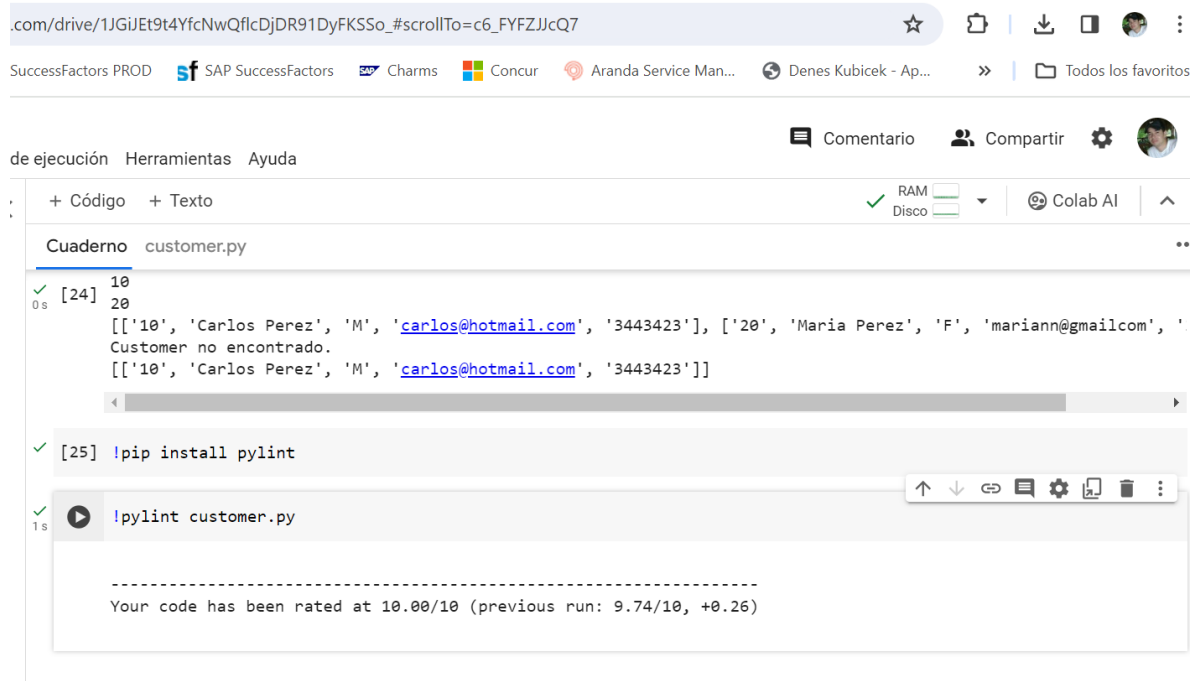
# Ejemplo de uso de los métodos
manager.crear_customer("10", "Carlos Perez", "M", "carlos@hotmail.com", "3443423")
manager.crear_customer("20", "Maria Perez", "F", "maria@hotmail.com", "35466667")
print(manager.consultar_customer())

manager.actualizar_customer("20", "mariann@gmailcom", "33324555")
print(manager.consultar_customer())
manager.borrar_customer("25")
manager.borrar_customer("20")
print(manager.consultar_customer())

[[['10', 'Carlos Perez', 'M', 'carlos@hotmail.com', '3443423'], ['20', 'Maria Perez', 'F', 'maria@hotmail.com',
10
20
[[['10', 'Carlos Perez', 'M', 'carlos@hotmail.com', '3443423'], ['20', 'Maria Perez', 'F', 'mariann@gmailcom', '
Customer no encontrado

```

Procedemos a ejecutar librería Pylint y se depura con las correcciones sugeridas obteniendo 10 de 10:



The screenshot shows a Google Colab notebook with the following content:

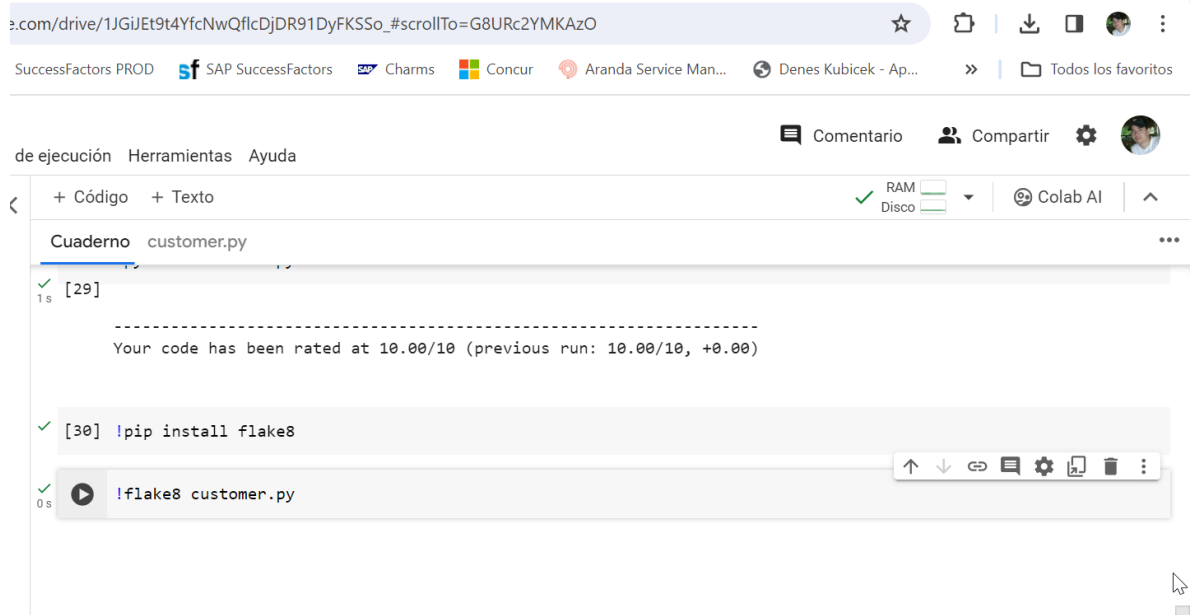
```
10
20
[[ '10', 'Carlos Perez', 'M', 'carlos@hotmail.com', '3443423'], [ '20', 'Maria Perez', 'F', 'mariann@gmailcom', '3443423']]
Customer no encontrado.
[[ '10', 'Carlos Perez', 'M', 'carlos@hotmail.com', '3443423']]
```

[25] !pip install pylint

[26] !pylint customer.py

-----  
Your code has been rated at 10.00/10 (previous run: 9.74/10, +0.26)

Se procede a ejecutar librería flake8, se realizan los ajustes recomendados y se obtiene código sin error posteriormente:



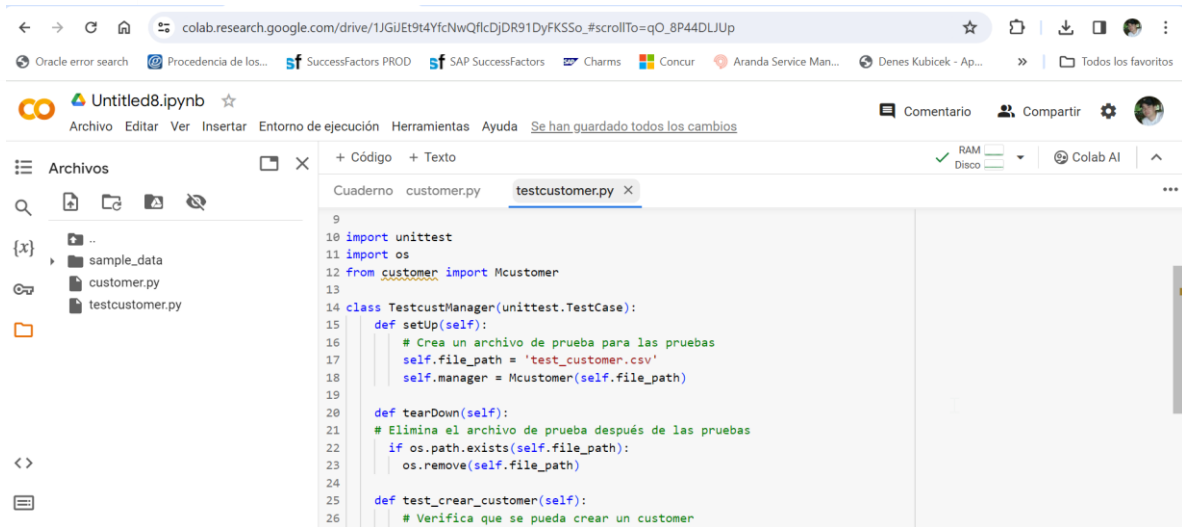
The screenshot shows a Google Colab notebook with the following content:

```
-----  
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

[30] !pip install flake8

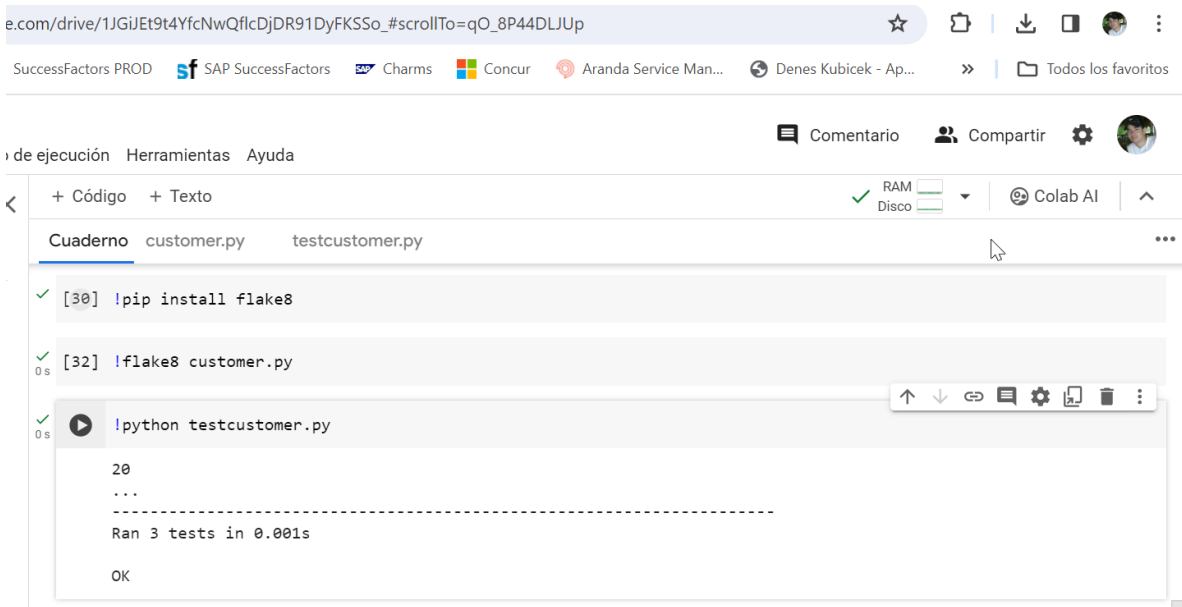
[31] !flake8 customer.py

Creamos el archivo testcustomer.py para las pruebas unitarias de Unittest.



```
9
10 import unittest
11 import os
12 from customer import Mcustomer
13
14 class TestcustManager(unittest.TestCase):
15     def setUp(self):
16         # Crea un archivo de prueba para las pruebas
17         self.file_path = 'test_customer.csv'
18         self.manager = Mcustomer(self.file_path)
19
20     def tearDown(self):
21         # Elimina el archivo de prueba después de las pruebas
22         if os.path.exists(self.file_path):
23             os.remove(self.file_path)
24
25     def test_create_customer(self):
26         # Verifica que se pueda crear un customer
```

Procedemos a ejecutar el archivo para validar las pruebas unitarias según lo especificado, obtenemos el ok.



```
[30] !pip install flake8

[32] !flake8 customer.py

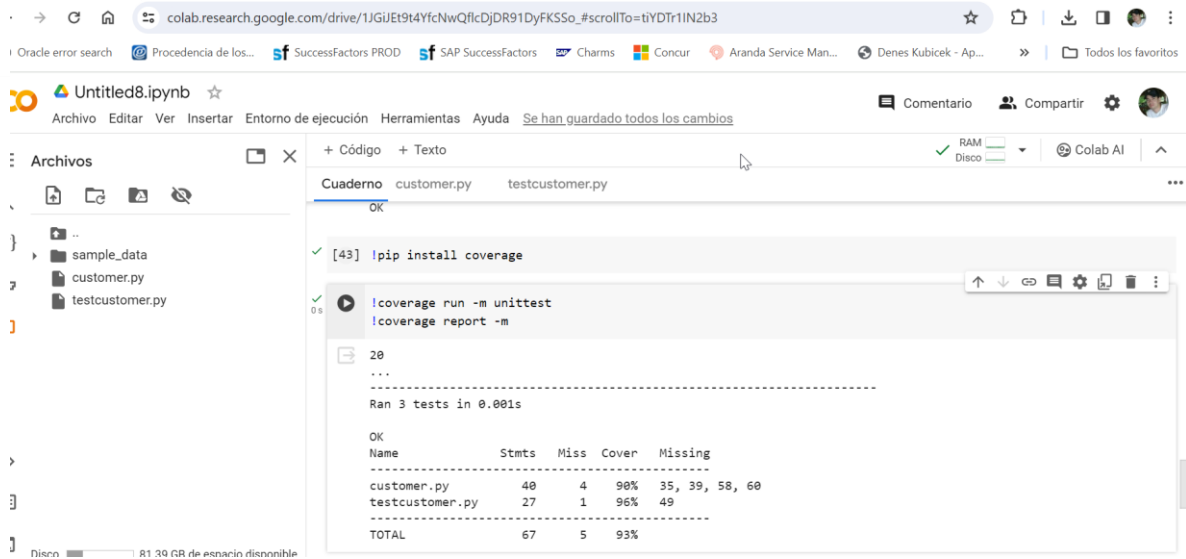
[33] !python testcustomer.py

20
...
-----
Ran 3 tests in 0.001s

OK
```

Ahora procedemos a ejecutar el reporte para visualizar el porcentaje de cobertura, obteniendo una cobertura del 93% del código:





colab.research.google.com/drive/1JGUEt9t4YfcNwQfCjDR91DyFKSSo\_#scrollTo=tiYDTr1IN2b3

Oracle error search | Procedencia de los... | SuccessFactors PROD | SAP SuccessFactors | Charms | Concur | Aranda Service Man... | Denes Kubicek - Ap... | Todos los favoritos

Untitled8.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

Archivos

- sample\_data
- customer.py
- testcustomer.py

+ Código + Texto

Cuaderno customer.py testcustomer.py

OK

[43] !pip install coverage

!coverage run -m unittest  
!coverage report -m

20  
...

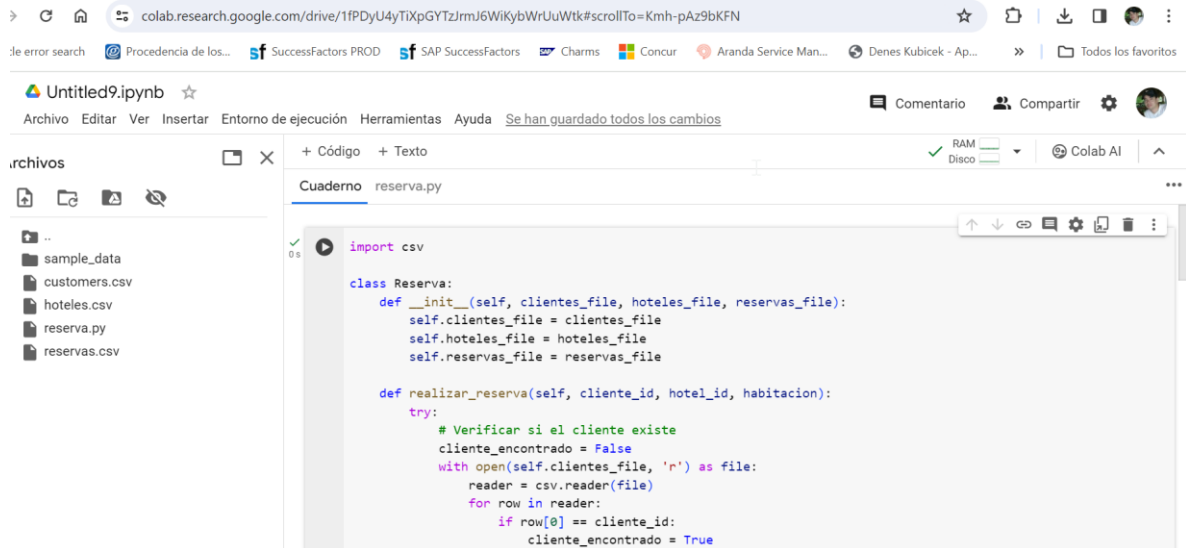
Ran 3 tests in 0.001s

OK	Name	Stmts	Miss	Cover	Missing
	customer.py	40	4	90%	35, 39, 58, 60
	testcustomer.py	27	1	96%	49
	TOTAL	67	5	93%	

Disco 11.34 GB de espacio disponible

### 3. Reservas

Se crea la clase reserva para gestionar en archivo las reservas realizadas por los clientes creados desde la clase customer hacia las habitaciones gestionadas desde la clase hotel:



colab.research.google.com/drive/1fPDyU4yTixpGYTzJrmJ6WiKbWUuWtk#scrollTo=Kmh-pAz9bKFN

Oracle error search | Procedencia de los... | SuccessFactors PROD | SAP SuccessFactors | Charms | Concur | Aranda Service Man... | Denes Kubicek - Ap... | Todos los favoritos

Untitled9.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

Archivos

- sample\_data
- customers.csv
- hoteles.csv
- reserva.py
- reservas.csv

+ Código + Texto

Cuaderno reserva.py

```
import csv

class Reserva:
    def __init__(self, clientes_file, hoteles_file, reservas_file):
        self.clientes_file = clientes_file
        self.hoteles_file = hoteles_file
        self.reservas_file = reservas_file

    def realizar_reserva(self, cliente_id, hotel_id, habitacion):
        try:
            # Verificar si el cliente existe
            cliente_encontrado = False
            with open(self.clientes_file, 'r') as file:
                reader = csv.reader(file)
                for row in reader:
                    if row[0] == cliente_id:
                        cliente_encontrado = True
```

Pruebas de reserva de habitaciones, prueba de habitación no disponible y pruebas de hotel no encontrado.



google.com/drive/1fPDyU4yTiXpGYTzJrmJ6WiKybWrUuWtk#scrollTo=Kmh-pAz9bKFN

Sf SuccessFactors PROD Sf SAP SuccessFactors SAP Charms Concur Aranda Service Man... Denes Kubicek - Ap... Todos los favoritos

Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios Comentario Compartir

+ Código + Texto

Cuaderno reserva.py

```
# Ejemplo de uso
clientes_file = 'customers.csv'
hoteles_file = 'hoteles.csv'
reservas_file = 'reservas.csv'

reserva = Reserva(clientes_file, hoteles_file, reservas_file)
reserva.realizar_reserva("10", "1000", "1")
reserva.realizar_reserva("20", "1000", "1")
reserva.realizar_reserva("10", "5000", "1")
```

```
1000
Reserva realizada con éxito.
Cliente no encontrado.
1000
2000
Hotel no encontrado.
False
```

Procedemos a ejecutar pylint, se realizan los ajustes recomendados, obteniendo posteriormente puntuación 10 de 10.

colab.research.google.com/drive/1fPDyU4yTiXpGYTzJrmJ6WiKybWrUuWtk#scrollTo=wka-92F3hOOs

Oracle error search Procedencia de los... Sf SuccessFactors PROD Sf SAP SuccessFactors Charms Concur Aranda Service Man... Denes Kubicek - Ap... Todos los favoritos

Untitled9.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

Archivos

sample\_data customers.csv hoteles.csv reserva.py reservas.csv

+ Código + Texto

Cuaderno reserva.py

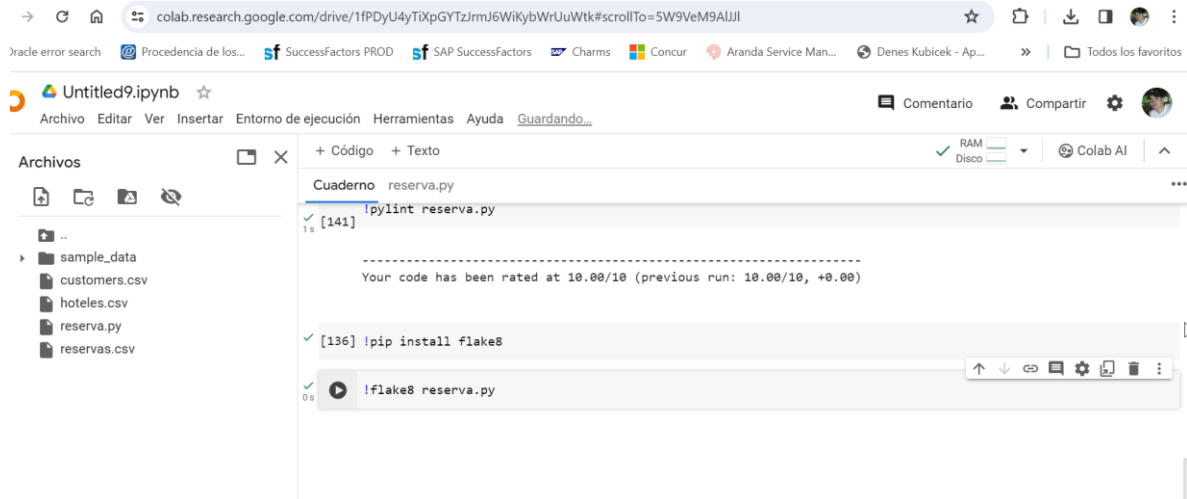
```
1000
[133] Reserva realizada con éxito.
      Cliente no encontrado.
      1000
      2000
      Hotel no encontrado.
      False
```

```
[120] !pip install pylint
```

```
[15] !pylint reserva.py
```

```
-----
Your code has been rated at 10.00/10 (previous run: 9.84/10, +0.16)
```

Procedemos a ejecutar flake8, se realizan os ajustes recomendados, obteniendo posteriormente 0 errores:



colab.research.google.com/drive/1fPDyU4yTiXpGYTzrmJ6WiKybWrUuWtk#scrollTo=5W9VeM9AIJl

Oracle error search | Procedencia de los... | SuccessFactors PROD | SAP SuccessFactors | Charms | Concur | Aranda Service Man... | Denes Kubicek - Ap... | Todos los favoritos

Untitled9.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Guardando...

Archivos

- ..
- sample\_data
  - customers.csv
  - hoteles.csv
  - reserva.py
  - reservas.csv

+ Código + Texto

Cuaderno reserva.py

```
[141] !pylint reserva.py
```

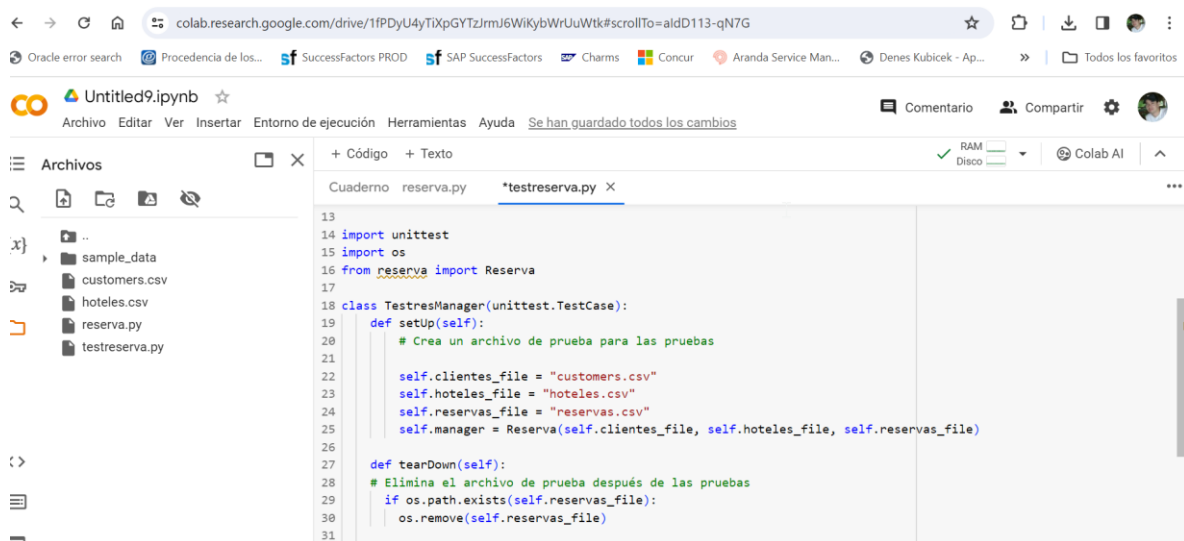
-----

Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

```
[136] !pip install flake8
```

```
!flake8 reserva.py
```

Creamos el archivo testreserva.py para realizar las pruebas unitarias de unittest.



colab.research.google.com/drive/1fPDyU4yTiXpGYTzrmJ6WiKybWrUuWtk#scrollTo=aldD113-qN7G

Oracle error search | Procedencia de los... | SuccessFactors PROD | SAP SuccessFactors | Charms | Concur | Aranda Service Man... | Denes Kubicek - Ap... | Todos los favoritos

Untitled9.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

Archivos

- ..
- sample\_data
  - customers.csv
  - hoteles.csv
  - reserva.py
  - testreserva.py

+ Código + Texto







Cuaderno reserva.py \*testreserva.py

```
13
14 import unittest
15 import os
16 from reserva import Reserva
17
18 class TestresManager(unittest.TestCase):
19     def setUp(self):
20         # Crea un archivo de prueba para las pruebas
21
22         self.clientes_file = "customers.csv"
23         self.hoteles_file = "hoteles.csv"
24         self.reservas_file = "reservas.csv"
25         self.manager = Reserva(self.clientes_file, self.hoteles_file, self.reservas_file)
26
27     def tearDown(self):
28         # Elimina el archivo de prueba después de las pruebas
29         if os.path.exists(self.reservas_file):
30             os.remove(self.reservas_file)
31
```

Ejecutamos librería para lograr total cobertura del código

.com/drive/1fPDyU4yTiXpGYTzJrmJ6WiKybWrUuWtk#scrollTo=aldD113-qN7G



SuccessFactors PROD  SAP SuccessFactors  Charms  Concur  Aranda Service Man...  Denes Kubicek - Ap...  Todos los favoritos


de ejecución Herramientas Ayuda

 Comentario  Compartir  

+ Código + Texto


✓ RAM  Disco  Colab AI 

Cuaderno reserva.py testreserva.py

 0 s

-----

Ran 2 tests in 0.002s

 FAILED (errors=1)

Name	Stmts	Miss	Cover	Missing
-----				
reserva.py	72	24	67%	27-28, 42-43, 45-46, 56-58, 65-86
testreserva.py	24	1	96%	46
-----				
TOTAL	96	25	74%	

