

CHALLENGE - SEGURIDAD INFORMÁTICA 2022

CONTENIDO

RETO ENTREGADO	3
1. CONTEXTO DEL CHALLENGE	3
2. OBJETIVO DEL CHALLENGE	3
3. ENTREGABLES	4
4. BONUS.....	4
5. CONSIDERACIONES GENERALES.....	4
SOLUCION PLANTEADA	5
1. OBJETIVOS.....	5
a. Objetivo General.....	5
b. Objetivos Específicos	5
2. ALCANCE	5
3. IDENTIFICACION DE NECESIDADES.....	5
4. INVESTIGACION PRELIMINAR.....	6
a. ¿Cuáles son los estados de la información y cómo pueden ser asegurados?	6
b. Análisis de los Datos	6
6. ARQUITECTURA DE LA SOLUCIÓN	10
7. DESCRIPCIÓN DE LA APLICACIÓN REALIZADA	12
a. Descripción del Ambiente de Desarrollo y Código Fuente.....	12
b. Interfaces de Usuario	19
c. APIS	21
8. METODOLOGÍA DEL ANÁLISIS DE RIESGOS.....	22
d. Paso 1. Establecimiento del Contexto	23
e. Paso 2. Valoración del Riesgo	23
f. Paso 3. Analizar los Riesgos.....	24
g. Paso 4. Evaluación del Riesgo:	25
h. Paso 5. Tratamiento del Riesgo.....	25
i. Paso 6. Comunicación del Riesgo	26
j. Paso 7. Monitoreo y Revisión del riesgo.....	26

9. PROBLEMAS ENCONTRADOS EN LA GENERACIÓN DEL CHALLENGE	26
a. Error en el campo fecha	26
b. Error en el campo codigo_zip	29
b. Pruebas de SQL Injection.....	30
c. HTTPS (Hypertext Transfer Protocol Secure):.....	33
10. CONCLUSIONES.....	33

RETO ENTREGADO

1. CONTEXTO DEL CHALLENGE

El equipo de Seguridad Informática, con el que vas a interactuar esta semana, se encarga de velar por la seguridad de la información de todos nuestros procesos. Es por ello que están alineados a las mejores prácticas y/o a estándares de seguridad como por ejemplo PCI DSS, NIST, LGPD, etc.

Además el equipo es auditado anualmente por Entes Reguladores con el objetivo de evaluar el cumplimiento de los diferentes requerimientos de seguridad así como también el correcto tratamiento de los datos (sensibles).

Se te ha asignado un proyecto en el cuál deberás obtener desde un proveedor externo información de clientes, garantizando que la información esté asegurada **en todos sus estados** y disponibilizar recursos para que la misma sea accesible por los distintos sectores dentro de la empresa.

2. OBJETIVO DEL CHALLENGE

Dado el contexto otorgado, realice un análisis de cómo consumir y posteriormente almacenar estos datos en una base relacional o no relacional de manera segura, además, deberás disponibilizar esta información para que distintos equipos y aplicaciones de la empresa puedan consumirlos, teniendo en cuenta cada uno de los atributos que vienen desde este proveedor.

Se deberán realizar los controles que se estimen necesarios para asegurar esta información, queda a criterio del desarrollador establecer el tipo de control, la forma de obtener los datos y los posibles consumidores.

Para llevar a cabo esta tarea será necesario obtener toda la información del siguiente endpoint (get):

- <https://62433a7fd126926d0c5d296b.mockapi.io/api/v1/usuarios>

Ejemplo:

- "fec_alta": "2021-04-04T07:00:50.276Z",
- "user_name": "Filomena.Collins",
- "codigo_zip": "17919-7207",
- "credit_card_num": "6393-0943-6424-5954",
- "credit_card_ccv": "131",

- "cuenta_numero": "58757891",
- "direccion": "Mitchell Bypass",
- "geo_latitud": "-88.3967",
- "geo_longitud": "-70.5628",
- "color_favorito": "black",
- "foto_dni": "http://placeimg.com/640/480",
- "ip": "218.204.159.251",
- "auto": "Cadillac Volt",
- "auto_modelo": "1",
- "auto_tipo": "Coupe",
- "auto_color": "Jaguar PT Cruiser",
- "cantidad_compras_realizadas": 84978,
- "avatar": "https://cdn.fakercloud.com/avatars/muringa_128.jpg",
- "fec_birthday": "2022-03-28T21:18:02.439Z",
- "id": "1"

3. ENTREGABLES

Se espera:

- Código fuente de la aplicación (Repositorio Github)
- Instrucciones para la ejecución de la aplicación (incluida cualquier aplicación o librería a instalar para el correcto funcionamiento del programa).
- Descripción de la aplicación realizada, supuestos, problemas y soluciones con los que se encontró al realizar la misma con evidencias en png.
- Análisis de riesgo de la solución planteada.

4. BONUS

Los Bonus son factores que entregan valor agregado al desafío:

- Dockerizar la aplicación.
- Documentación del proceso (Diagrama de clases, Arquitectura, etc).

5. CONSIDERACIONES GENERALES

A tener en cuenta:

- El challenge debe ser realizado idealmente en Python o Golang.
- Se podrán crear todas las funciones/scripts/etc complementarios que se consideren necesarios para un correcto funcionamiento de la aplicación

SOLUCION PLANTEADA

1. OBJETIVOS

a. Objetivo General

Analizar la solución propuesta para identificar los riesgos de seguridad y definir buenas prácticas para el aseguramiento de las aplicaciones.

b. Objetivos Específicos

- Abordar cada proceso del desarrollo desde la visión integral de gestión de riesgos.
- Desarrollar el proceso de análisis y evaluación de los riesgos de seguridad existentes ligados a las vulnerabilidades para medir el impacto en la solución del Challenge.
- Recomendar la implementación de buenas prácticas y controles de seguridad necesarios para disminuir los riesgos a niveles aceptables de acuerdo con la norma ISO 27001, que permitan mitigar las vulnerabilidades de seguridad del sitio.

2. ALCANCE

Realizar una investigación integral de la solución desde el punto de vista de riesgos que abarque la identificación, medición, tratamiento y propuestas de control de los riesgos identificados en el desarrollo del Challenge para Mercado Libre (MELI).

Adicionalmente, definir un diseño de arquitectura de seguridad de la solución con el fin de garantizar la protección de los datos durante el ciclo de vida en la compañía.

3. IDENTIFICACION DE NECESIDADES

Se describen las necesidades de la aplicación requerida, con el objetivo de identificar las funcionalidades del producto mínimo viable:

ITEM	Necesidad
1	Obtener desde un proveedor externo información de clientes
2	Almacenar estos datos en una base relacional o no relacional de manera segura

3	La información debe estar asegurada en todos sus estados
4	Se debe disponibilizar la información para que distintos equipos/aplicaciones puedan consumirlos

4. INVESTIGACION PRELIMINAR

Se realiza una investigación inicial para entender algunos conceptos y validar los controles que puede ser asignados en los diferentes casos:

a. ¿Cuáles son los estados de la información y cómo pueden ser asegurados?

Los estados de la información son los siguientes:

- **En Reposo:** Con este término nos referimos a la información que no está siendo accedida, usada, ni procesada y que se encuentra almacenada en un medio físico o lógico. **Para proteger los datos en reposo, se pueden cifrar archivos confidenciales antes de almacenarlos.**
- **En Transito o Movimiento:** Información que viaja a través de cualquier tipo de canal privado o público de comunicación. Es información que se encuentra viajando de un punto a otro. **Para proteger los datos en tránsito, se elige usar HTTPS (Hyper Text Transfer Protocol Secure) para las conexiones encriptadas y proteger el contenido de los datos.**
- **En Uso:** Se habla de información en uso cuando es accedida por una o varias aplicaciones o personas para su tratamiento. **Para proteger los datos en tránsito, se elige cifrar los datos financieros a nivel de columna, para que solo las partes autorizadas pueden leerlos.**

b. Análisis de los Datos

Se realiza una descripción e identificación inicial de los tipos de datos que serán obtenidos del EndPoint y guardados en la base de datos, etiquetándolos así:

Campos	Descripción
fec_alta	Fecha de la primera compra
user_name	Nombre del cliente
codigo_zip	Código de identificación de la zona
credit_card_num	Número de la tarjeta de crédito del cliente
credit_card_ccv	Número de verificación ubicado en la tarjeta de crédito del cliente
cuenta_numero	Número de la cuenta
direccion	Dirección de residencia del cliente
geo_latitud	Coordenadas que indican latitud
geo_longitud	Coordenadas que indica longitud

color_favorito	Color favorito indicado por el cliente
foto_dni	Foto del documento de identidad del cliente
ip	IP de la ultima transacción realizada en el sitio por el cliente
auto	Último auto comprado por el cliente
auto_modelo	Último modelo del auto comprado por el cliente
auto_tipo	Último tipo del auto comprado por el cliente
auto_color	Último color del auto comprado por el cliente
cantidad_compras_realizadas	Cantidad de clientes realizadas por el cliente a través de la plataforma
avatar	Imagen elegida por el cliente para identificar su perfil
fec_birthday	Fecha de nacimiento del cliente
Id	Identificador del cliente

DATOS			
Personales	Geográficos	Financieros	Comerciales
user_name	codigo_zip	credit_card_num	fec_alta
foto_dni	ip	credit_card_ccv	cuenta_numero
	dirección		color_favorito
	geo_latitud		Auto
	geo_longitud		auto_modelo
			auto_tipo
			auto_color
			cantidad_compras_realizadas
			Avatar
			fec_birthday

Para los datos, se procede analizar la información asociada a los datos **personales** (aquella información que pueda ser relacionada con una persona, por ejemplo: la dirección de la casa) y **sensibles** (aquellos datos que afectan la intimidad del dueño de la información o cuyo uso indebido puede generar su discriminación, tales como: los datos relativos a la salud, a la vida sexual, y los datos biométricos) encontrando que **foto_dni** debe tener un tratamiento especial de guardado en la base de datos y ser accedida sólo por usuarios autorizados.

Para los **datos financieros** se realiza la validación del **Estándar de Seguridad PCI** identificando la siguiente información en sus apartados:

- ***“3.2.2 No guardar el valor o código de validación de tarjeta (número de tres o cuatro dígitos impresos en el anverso o reverso de una tarjeta de pago) utilizado para verificar las transacciones con tarjeta ausente”*** motivo por el cual este campo (credit_card_ccv) **NO** será almacenado en la solución planteada.
- ***“3.4 Asegurar que el Número de Cuenta Primario (PAN), como mínimo, sea ilegible en cualquier lugar en que esté guardado (incluyendo datos en medios portátiles, medios de respaldo, registros o bitácoras, y datos recibidos de redes inalámbricas o guardados en las mismas) utilizando los siguientes métodos:***
 - *Funciones hash de una sola vía (índices hash)*
 - *Números truncados*
 - *Tokens de índice y pads (el pad debe ser guardado bajo seguridad)*
 - *Criptografía de alta seguridad como el estándar 7 de 128 bits o el AES de 256 bits con procesos y procedimientos asociados de administración de claves.*

La información MÍNIMA sobre las cuentas que necesita estar en forma ilegible es el número de cuenta de la tarjeta de pago.”, se usará el método de función hash a través del algoritmo SHA-256 (Secure Hash Algorithm) para evitar que este campo (credit_card_num) este legible a los usuarios. Además, se realizará una validación para verificar su validez antes de ser guardados en la base de datos.

En general, la solución debe garantizar una adecuada segregación de funciones a través de roles/perfiles de accesos, seleccionando los datos que pueden ser utilizados sin generar afectaciones en la legislación de habeas data y de PCI DSS:



Lo anterior, tiene la siguiente lógica de negocio:

- **User:** Este usuario no podrá consultar en texto claro a los datos financieros de los clientes, ni la información sensible. Sólo tendrá acceso a la información comercial de los clientes que son necesarias para cumplir sus funciones:
 - Fecha de Alta
 - Nombre
 - Código Postal
 - # Cuenta
 - Dirección
 - Color Favorito
 - Auto
 - Modelo Auto
 - Tipo Auto
 - Color Auto
 - # Compras Realizadas
 - Avatar
- **Authorized:** Este usuario no podrá consultar en texto claro los datos financieros de los clientes, pero podrá validar los datos sensibles para realizar las validaciones internas que el negocio requiera:
 - Nombre
 - # Tarjeta de Crédito (Cifrado)
 - Tipo de Tarjeta de Crédito

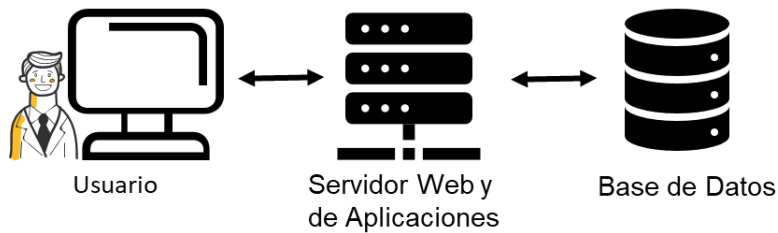
- # Cuenta
- Foto DNI
- **Admin:** Este usuario no podrá consultar en texto claro los datos financieros de los clientes cuando la tarjeta enviada por el proveedor es válida, pero si la tarjeta es inválida podrá verla en texto claro. Lo anterior, con el fin validar la información con el proveedor y validar si se está presentando alguna novedad. Los datos sensibles de los clientes no serán accesibles a este usuario:
 - Id
 - Fecha de Alta
 - Nombre
 - Código Postal
 - # Tarjeta de Crédito (Cifrada cuando es válida, en texto claro cuando es invalida)
 - Tipo de Tarjeta de Crédito
 - # Cuenta
 - Dirección
 - Latitud
 - Longitud
 - Color Favorito
 - IP
 - Auto
 - Modelo Auto
 - Tipo Auto
 - Color Auto
 - # Compras Realizadas
 - Avatar

6. ARQUITECTURA DE LA SOLUCIÓN

A continuación, se plantea la arquitectura técnica propuesta para la solución de este Challenge, considerando:

- Los lineamientos de seguridad en la infraestructura son los siguientes:
 - **Firewall** para servidores de acceso público como componente principal de la seguridad perimetral.
 - **Lista blanca de IPs** autorizadas para el ingreso a la aplicación.

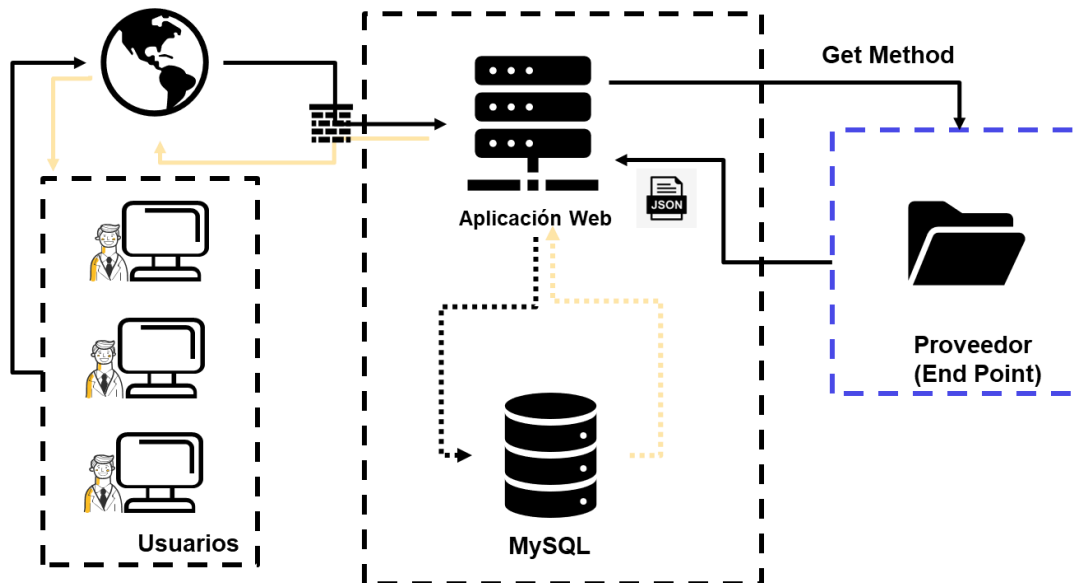
- Utilización y actualización permanente de **Antivirus** en los servidores de aplicación.
- La solución está basada en una arquitectura Cliente/Servidor:



Donde se cumplen las siguientes funciones:

- **Usuario (Cliente):** Se ejecuta en un navegador sobre el computador cliente del usuario final. El navegador se conecta a través del servidor web, el cual ejecuta la lógica de negocio y accede a los datos de la base de datos.
- **Servidor Web/App:** Identifica y envía peticiones desde los clientes web al servidor y regresa páginas de aplicación HTML. Adicionalmente, ejecuta procesos Batch, operaciones de la lógica de negocio como conectividad y acceso a la base de datos e integración con externos a través de APIs REST.
- **Base de Datos:** La base de datos que almacena toda la información usada por la aplicación. Los cambios se pueden realizar directamente en tiempo real.

En la siguiente imagen, se detalla un poco más la arquitectura de la solución:



7. DESCRIPCIÓN DE LA APLICACIÓN REALIZADA

En este apartado, se realizará una breve descripción de los diferentes componentes asociados a la aplicación, desde el momento del desarrollo hasta el momento del uso por parte de los usuarios, en los que se pensó para que sea lo más simple e intuitiva posible:

a. Descripción del Ambiente de Desarrollo y Código Fuente

Se hace uso de contenedores a través de **Docker** para facilitar la disponibilidad del entorno de desarrollo. A continuación, se realiza una descripción del ambiente y sus componentes:

- **Dockerfile:**

Son los archivos de texto plano que contienen las instrucciones necesarias para crear una imagen, para este caso se usaron dos archivos, uno para el contenedor de **MySQL** y el otro para el contenedor de **Python**:

```
mysql > Dockerfile > ...
1 # Descarga la imagen de MySQL - Última versión
2 FROM mysql:latest

=> => writing image sha256:70231e5644def10ad8536d3d5865a4278a9c6c7afbc4d84b1f9a0e8d25997a6d 0.0s
=> => naming to docker.io/library/docker_challenge_mysql_1 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
WARNING: Image for service pythonapp was built because it did not already exist. To rebuild this image you must use 'docker-compose build' or 'docker-compose up --build'.
docker_challenge_mysql_1 is up-to-date
Creating docker_challenge_mysql_1 ... done
PS D:\Melissa\MercadeoLibre\DOCKER_CHALLENGE> []
```

```
python > Dockerfile > ...
1 # Descarga la imagen de Python 3.9
2 FROM python:3.9
3
4 # Ubicación del proyecto en el SO de la imagen de Python
5 WORKDIR /app
6
7 # Copia el contenido del directorio actual en /app - Se copia el código fuente actual
8 COPY . /app
9
10 # Descargar e instala los paquetes necesarios para el proyecto
11 RUN pip3 --no-cache-dir install -r requirements.txt
12
13 # Ejecuta app.py cuando el contenedor se inicia
14 CMD ["python3", "src/app.py"]
15
16
17

=> => writing image sha256:70231e5644def10ad8536d3d5865a4278a9c6c7afbc4d84b1f9a0e8d25997a6d 1.1s
=> => naming to docker.io/library/docker_challenge_pythonapp_1 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
WARNING: Image for service pythonapp was built because it did not already exist. To rebuild this image you must use 'docker-compose build' or 'docker-compose up --build'.
docker_challenge_pythonapp_1 is up-to-date
Creating docker_challenge_pythonapp_1 ... done
PS D:\Melissa\MercadeoLibre\DOCKER_CHALLENGE> []
```

- **Docker-Compose:**
Para poder configurar el contenedor de Python en la misma red de MySQL, es necesario habilitar el Docker-Compose.

En este archivo se configuró:

- Se asigna nombre y contraseña de la base de datos.
- Los puertos por donde se conectará la base de datos.
- El puerto de la página web (se cambia del 4000 al 7000).

- Se habilita una carpeta externa para transmitir los cambios del código fuente en línea.

```

1  services:
2    pythonapp:
3      build: ./python/
4      ports:
5        - '7000:4000'
6      depends_on:
7        - mysql
8      volumes:
9        - D:\Melissa\Mercadeolibre\DOCKER_CHALLENGE\python\src:/app/src/
10
11   mysql:
12     build: ./mysql/
13     environment:
14       MYSQL_DATABASE: 'db'
15       MYSQL_ROOT_PASSWORD: 'root'
16     ports:
17       - '3306:3306'

```

- **Templates:**

Son archivos que muestran contenido estático y dinámico a los usuarios que visitan la aplicación:

- **admin.html:** Está asociada a la vista del usuario administrador, muestra la información de los clientes que se carga de la base de datos que contiene tarjetas de crédito válidas.
- **authorized.html:** Está asociada a la vista del usuario autorizado, muestra la información de los clientes que se cargan de la base de datos que contiene tarjetas de crédito válidas encriptadas y la información asociada a los datos sensibles (Foto DNI).
- **batch.html:** Está asociada a la vista del cargue masivo.
- **index.html:** Está asociada a la página principal, también llamada "home".
- **log.html:** Está asociada a la vista del usuario administrador, muestra la información de los clientes que se cargan de la base de datos que contiene tarjetas de crédito no validas.
- **login.html:** Está asociada a la vista de inicio de sesión.
- **logout.html:** Permite cerrar la sesión del usuario.
- **user.html:** Está asociada a la vista del usuario y con la información que se carga de la base de datos.

- **Utils.py**

Este archivo contiene funciones tales como:

- **Función Fecha:** Cambia el formato de la fecha que se obtiene del EndPoint.
- **Función Hash:** Aplica el algoritmo SHA-256 al campo credit_card_num antes de guardarlo en la base de datos.
- **Función Validación:** Realiza validaciones adicionales al campo credit_card_num para determinar si la tarjeta de crédito ingresada es válida o invalida.

- **Database.py**

Este archivo contiene:

- La conexión a la base de datos.
Nota: Para este Challenge la contraseña de la conexión está en texto claro, esto debe ser modificado dado que no es una buena práctica en seguridad.
- Las consultas en la base de datos que permiten mostrar la información:
 - A los usuarios (Admin, User, Authorized) dependiendo de su rol.
 - A los terceros que será expuesta a través de las APIs.
Nota: El campo Fecha de Alta no pudo ser expuesto.
- El borrado (truncate) de la base de datos en el proceso batch (dado que se genera error al realizar insert porque los ids se repiten).
- El insert en la base de datos de la información obtenida del endpoint del proveedor.
- La encriptación de la contraseña de los usuarios, con el objetivo de no guardarla en texto claro en la base de datos.

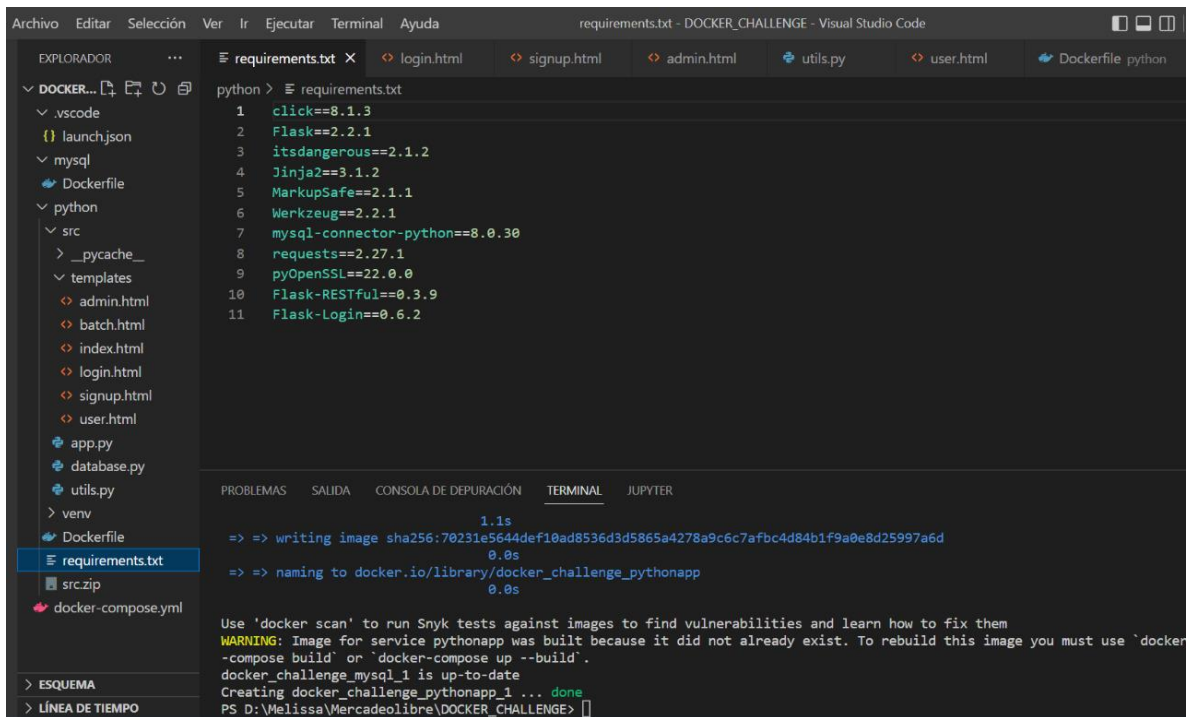
- **app.py**

Este archivo contiene:

- La inicialización de la aplicación.
- La configuración más importante, que me permite ejecutar la aplicación:

```
#####
#                               Ejecutar Aplicación
#####
if __name__ == '__main__':
    app.run(host="0.0.0.0", port=4000, debug=True, ssl_context='adhoc']
```

- Asigna la dirección del sitio.
 - El puerto
 - La posibilidad de ver los errores de la aplicación en la página o en la terminal (esto en modo desarrollo). Para el paso a producción, este parámetro se debe cambiar a **False** para impedir la visualización de los errores que se puedan presentar.
 - La seguridad a través del HTTPS.
- Manejo de sesiones de usuarios y páginas web.
- Consumir la información obtenida del endpoint del proveedor y guardarla en la base de datos, usando las consultas del archivo database.py
- De acuerdo con la información almacenada en la base de datos, se exponen las APIs usando el archivo database.py
- **requerimientos.txt**
Este archivo contiene la lista de todos los paquetes (librerías) instaladas en el entorno contenedor de Python. Además, permite automatizar la instalación de paquetes Python y por lo tanto agilizar esta parte del proceso de desarrollo de software.



The screenshot shows the Visual Studio Code interface with a Dockerfile open. The Dockerfile contains the following instructions:

```
python > requirements.txt
1 click==8.1.3
2 Flask==2.2.1
3 itsdangerous==2.1.2
4 Jinja2==3.1.2
5 MarkupSafe==2.1.1
6 Werkzeug==2.2.1
7 mysql-connector-python==8.0.30
8 requests==2.27.1
9 pyOpenSSL==22.0.0
10 Flask-RESTful==0.3.9
11 Flask-Login==0.6.2
```

The terminal output shows the Docker build process:

```
1.1s
=> => writing image sha256:70231e5644def10ad8536d3d5865a4278a9c6c7afbc4d84b1f9a0e8d25997a6d
0.0s
=> => naming to docker.io/library/docker_challenge_pythonapp
0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
WARNING: Image for service pythonapp was built because it did not already exist. To rebuild this image you must use `docker
-compose build` or `docker-compose up --build`.
docker_challenge_mysql_1 is up-to-date
Creating docker_challenge_pythonapp_1 ... done
PS D:\Melissa\MercadeoLibre\DOCKER_CHALLENGE>
```

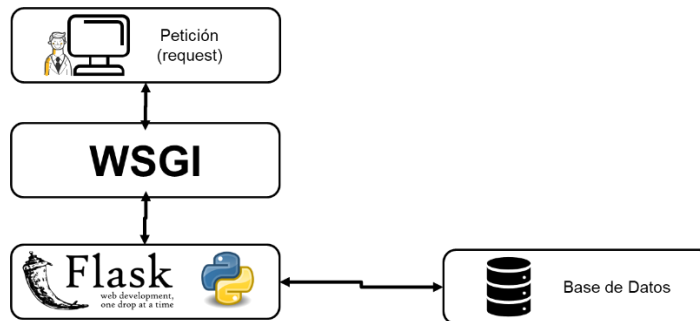
- **mysql-connector-python:** Este paquete contiene el módulo `mysql.connector`, que está escrito en Python. Tras instalar el paquete, ya se pueden realizar consultas sobre bases de datos MySQL.

```
#!/usr/bin/python

print "Resultados de MySQLdb:"
import MySQLdb
miConexion = MySQLdb.connect( host='localhost', user= 'USUARIO', passwd='PASS', db='neoguias' )
cur = miConexion.cursor()
cur.execute( "SELECT nombre, apellido FROM usuarios" )
for nombre, apellido in cur.fetchall() :
    print nombre, apellido
miConexion.close()
```

- **Flask:** Es un micro Framework escrito en Python y concebido para facilitar el desarrollo de aplicaciones web bajo el patrón MVC (Modelo Vista Controlador).

Es compatible con WSGI - Web Server Gateway Interface (es una especificación que describe cómo se comunica un servidor web con una aplicación web y cómo se pueden llegar a encadenar diferentes aplicaciones web para procesar una solicitud/petición (o request)).



Dependencias: Estas distribuciones se instalarán automáticamente al instalar Flask:

- **Werkzeug:** Implementa WSGI, la interfaz estándar de Python entre aplicaciones y servidores.
 - **Jinja2:** Es un lenguaje de plantillas que renderiza las páginas que sirve tu aplicación.
 - **MarkupSafe:** Viene con Jinja, escapa de la entrada no fiable cuando se renderizan las plantillas para evitar ataques de inyección.
 - **ItsDangerous:** Firma de forma segura los datos para asegurar su integridad, se utiliza para proteger la cookie de sesión de Flask.
 - **Click:** Es un marco para escribir aplicaciones de línea de comandos. Proporciona el comando flask y permite añadir comandos de gestión personalizados.
-
- **Flask-Login:** Permite administrar las sesiones del usuario tras la autenticación.
 - **Flask Restful:** Es una extensión que permite generar APIs REST muy fácilmente.
 - **Requests:** Librería usada para hacer una petición GET, ya sea para obtener el contenido de una web o para realizar una petición a un API. Para ello, simplemente tienes que invocar a la función get() indicando la URL a la que hacer la petición:

```
1. import requests
2.
3. resp = requests.get('https://www.google.com/')
```

La función devuelve un objeto Response, que en este caso se ha asignado a la variable resp, con toda la información de la respuesta.

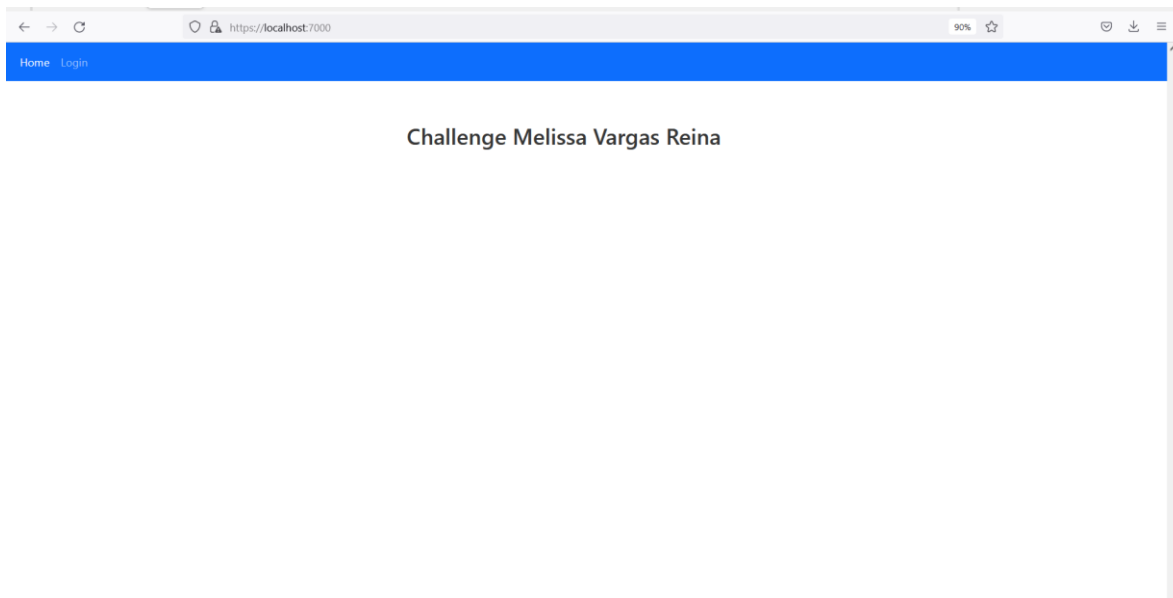
b. Interfaces de Usuario

A continuación, se detallan las diferentes GUIs que ofrece la aplicación, en la que se especifican las pantallas de la interfaz gráfica con la que interactuará el usuario. La aplicación consta de los siguientes menús:

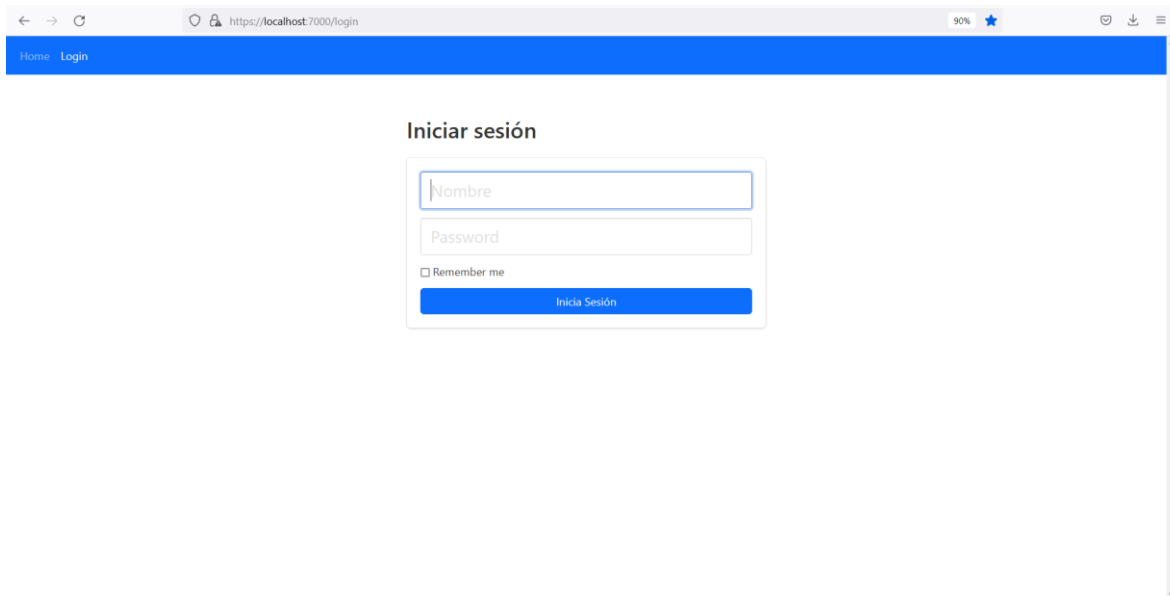
- **Home:** Muestra la página principal de la aplicación.
- **Login:** Inicia la sesión del usuario.
- **Batch:** Consume el API Rest del proveedor y guarda la información en la base de datos.
- **Admin:** Muestra en pantalla la información obtenida con tarjetas de crédito válidas (con el número de tarjeta de crédito encriptado).
- **User:** Muestra en pantalla la información de las ventas a los clientes filtrándola por los números de tarjeta válida.
- **Authorized:** Muestra en pantalla la información sensible de los clientes filtrándola por los números de tarjetas válidas.
- **Log:** Muestra en pantalla la información obtenida con tarjetas de crédito no válidas para el administrador (con el número de tarjeta de crédito en texto claro).
- **Logout:** Cierra la sesión del usuario.

Se detallan las pantallas que verá un usuario **Administrador**:

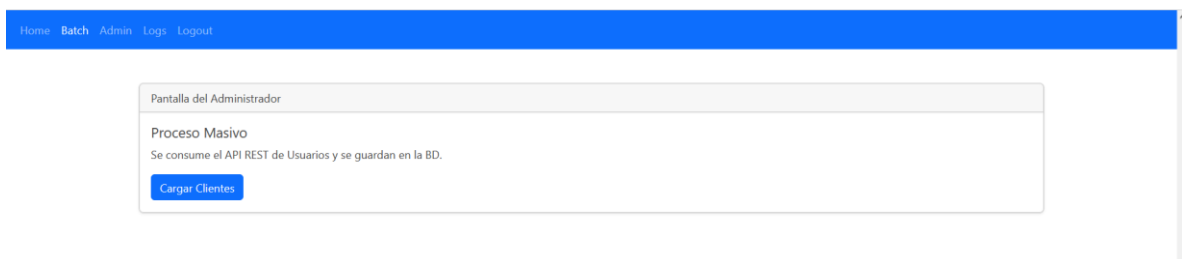
1. La página principal del sitio es: <https://localhost:7000>



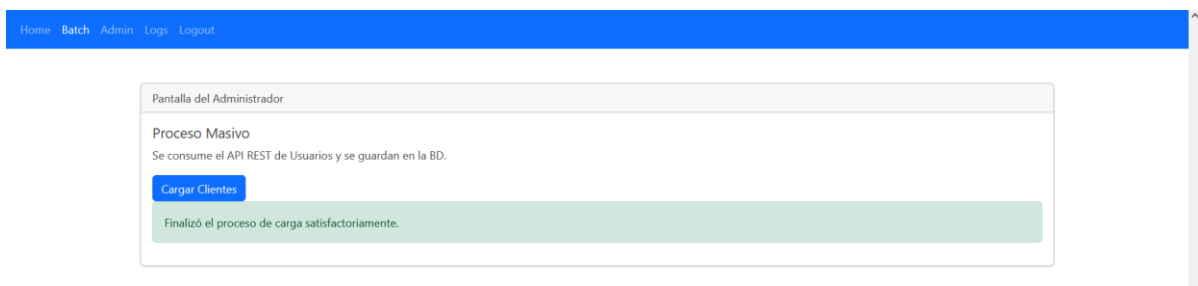
2. Para iniciar sesión, se debe dar clic al menú **Login** e ingresar datos del usuario:



3. Se debe ingresar al Menú **Batch** (cabe resaltar, que la aplicación ingresa directamente a este menú) para que la aplicación obtenga la información del EndPoint:



4. Cuando se finalice el proceso, el sistema arrojará un mensaje ***“Finalizó el proceso de carga satisfactoriamente.”*** que informa el estado de este:



5. Para validar la información, se tiene los menús **Admin** y **Logs**:
 - a. **Admin:** Muestra la información obtenida con tarjetas de crédito validas.

ID	Fecha de Alta	Nombre	Código Postal	# Tarjeta de Crédito	Tipo Tarjeta de Crédito	# Cuenta	Dirección	Latitud	Longitud	Color Favorito	IP	Auto	Modelo Auto	Tipo Auto	Color Auto	# Compras Realizadas	Avatar
13	2021-11-10 07:54:15	Dayana_Corkery70	56324-1007	6a36a262926a1e0797954a6fbb8bb6c51771b16d8960a1b1998747134	NONE--	61021827	Goodwin Plaza	74.4151	-99.2649	blue	143.207.128.77	Mercedes Benz Colorado	V10	Convertible	Auton Martin Countach	622	https://cdn.fakercloud.com/avatars/iamkethmason_128.jpg
15	2021-09-23 17:47:00	Ronny53	62409-8899	8875c7c1027b4a5dfbcb72bae4db374bba08f7c77d0f8abe4ac09112de469c2	NONE--	49452436	Ansel Square	-36.4495	91.6794	orange	208.162.190.88	Land Rover X75	El Camino	Wagon	Jaguar Silverado	71155	https://cdn.fakercloud.com/avatars/romgrosche_128.jpg
17	2021-08-01 22:43:43	Jayde_Spinka	00540-2686	67ab7b16baebd63f13d5c3b8111e0f9deacc3eaf5a6c583b0c220a601d84	NONE--	68330529	Raglan Mansors	-47.0884	-23.3069	plum	217.115.81.41	Maserati CX-9	LeBaron	Hatchback	Chrysler Civic	54614	https://cdn.fakercloud.com/avatars/davidhemphill_128.jpg
18	2022-01-26 19:59:40	Kara_O'Reilly33	26435-2521	a9e7a12295945a1a4894a0516ad5bae849949a515d38f277000e0eadda1	NONE--	93240266	Denesuk Square	-48.7134	78.6915	white	78.1487.210	Bugatti LeBaron	Aventador	Passenger Van	Lamborghini Aventador	18803	https://cdn.fakercloud.com/avatars/oktayelipe_128.jpg
20	2022-01-12 19:15:54	Alphonso_Jacobs	22205-3359	7aec468de9875e6eb7e80231509fccbe5f720f950bde13d7c6802356ba1862	NONE--	81553862	Susana Creek	-47.8670	-147.8133	fuchsia	223.124.252.193	Tesla Grand Canavan	Charger	Crew Cab Pickup	Chrysler Expedition	55637	https://cdn.fakercloud.com/avatars/afisck_128.jpg
30	2021-11-23 10:07:59	Jerald_Schimme90	81497-5007	37537344253466999446b2618212023246205023263a47396c67b5d9541c09	NONE--	96912083	Hyatt Knoll	-36.9224	-138.5087	fuchsia	199.104.156.90	Rolls Royce Spyder	Alpine	Sedan	Audi Model 3	76773	https://cdn.fakercloud.com/avatars/axel_128.jpg
39	2021-12-19 15:18:49	Olive31	43899-2202	ceef85677c7caeb3f13048a3086f710beeb3b5a3c8d8184d72ea1d29f98bc	NONE--	51641633	Rutherford Stravenus	44.1576	-99.0655	teal	97.2873.209	Cadillac Model T	Model T	Passenger Van	Smart PT Cruiser	48404	https://cdn.fakercloud.com/avatars/thunintestilden_128.jpg
41	2022-03-22 23:50:07	Lee_Metz	62747	264ab70b2b474bdf4bdc3246c3849c448b2982c63d9fca0495daae19fcb769	NONE--	64322366	Kenny Manor	5.3118	117.4413	white	78.253.247.155	Bugatti Element	2	SUV	Rolls Royce Model T	88762	https://cdn.fakercloud.com/avatars/ewgiles_128.jpg
49	2021-04-30 03:09:28	Julie_Sesser16	85970	6b8d44ee55a82d6207543ba2b304b542aaf6a74d64702b678d6d68b8f3170	NONE--	2773110	Kevin Road	-23.0884	-144.5966	purple	238.128.37.144	Bugatti Beetle	CX-9	Convertible	Maserati Fortwo	26011	https://cdn.fakercloud.com/avatars/codyanfilippo_128.jpg
50	2021-04-01 18:52:27	Karreen_Watkins13	33077	9df4de752d5a2caef7953c3dca3e93384459275d6dc592268a48f489f70283	NONE--	49607980	Beahan Expressway	48.5093	10.0501	plum	140.212.30.102	Chevrolet Element	Model S	Cargo Van	Polestar Taurus	90128	https://cdn.fakercloud.com/avatars/rhigdeterson_128.jpg
60	2021-07-14 15:18:30	Robaldo24	91264	b42df8d94c8ba01c457e02315ab957332753398a3d3dc56955e6f5822990	**MASTERCARD	51058311	Erica Lock	82.5174	-151.0458	violet	177.29.124.29	Kia I	LeBaron	Minivan	Nissan Iltia	12426	https://cdn.fakercloud.com/avatars/ronowhade_128.jpg
63	2021-10-29 20:49:51	Rhett33	64565-7353	05ae0376ce9838236c1c3091902579a55ec063a583669643c0238c103ca25ee4	**VISA	76565777	Chempin Lane	41.8289	-94.0734	white	126.97.206.122	Polestar Fortwo	Focus	SUV	Nissan El Camino	46733	https://cdn.fakercloud.com/avatars/cggauv_128.jpg
64	2021-06-11 19:33:42	Lou_Haley11	62861-8172	c8f123a252147a9f75d78b73276317843ab4fcdcb326bd07a5d442de19	NONE--	85785005	Lea Estates	-43.3101	-90.8868	orchid	164.106.217.11	BMW Wrangler	LeBaron	Hatchback	Bentley Mercatago	68227	https://cdn.fakercloud.com/avatars/mrgaelkoeman_128.jpg
65	2022-01-21 16:27:24	Shammi82	25168	07efab027e71e7a9959f6af02e46d58681831072defa7b15d6ebf11de31a9f1	NONE--	38772320	Winnia Radial	86.8526	147.0201	plum	123.36.7.35	Ferrari Golf	Accord	Minivan	Kia Impala	36765	https://cdn.fakercloud.com/avatars/iamjdelson_128.jpg
66	2021-12-10 15:18:34	Jalyn_Schamberger	04434-1979	f1ef68d7496eb319816409b1cd70a200acba0794c73cd53a53d830fcd25f8fa	**MASTERCARD	34224848	Maximilia Freeway	-4.3685	-126.8635	violet green	2.248.58.30	Toyota Grand Cherokee	Colorado	Extended Cab Pickup	Kia LeBaron	7601	https://cdn.fakercloud.com/avatars/calikkara_128.jpg
67	2021-08-18 16:07:39	Jeremy_Raynor	18027	fa3b491a0e9a32991339146d0d7b7c354708d09780295744a549eae0	NONE--	62580258	Pfeiffer Shores	36.9450	-20.8776	fuchsia	125.211.234.62	Jeep Malibu	Explorer	Convertible	Honda LeBaron	90697	https://cdn.fakercloud.com/avatars/rmpng_128.jpg
69	2021-10-09 18:05:21	Estefania72	88950	e40c2c341cc2b6396a32e4ab1d99f7875924930860744cc75931899f9b98	NONE--	41949500	Davon Keys	12.8700	38.3881	magenta	88.80.95.108	Audi Impala	LeBaron	Cargo Van	BMW Aventador	76994	https://cdn.fakercloud.com/avatars/iamteffen_128.jpg
73	2022-02-03 07:34:04	Reginald_Dickinson	09636-9101	cbaf3e4f71e71a62b14632aef9126c4b7790a0b6c5cb4998434afac492a	**MASTERCARD	70125070	Christiam Plaza	4.9369	-151.2644	orchid	250.102.50.61	Kia Golf	Escalade	Passenger Van	Fiat Spyder	18127	https://cdn.fakercloud.com/avatars/kinday_128.jpg
87	2021-06-01 19:15:09	Katherine_Metz3	38555-0388	a5ef0a0c872cc30cfa04810e22df8e157a3a12b78df93569190c078de9c53	NONE--	70321687	Moore Lakes	35.4874	-77.8118	orchid	197.218.90.114	Mazda Accord	Taurus	Crew Cab Pickup	Tesla Mercatago	75360	https://cdn.fakercloud.com/avatars/elbetargueta_128.jpg
94	2022-01-09	Okoy_Jacobi	01977-1280	177030a2724366718df6fdebf4711fae30522e4603917b64c5e6f7028770c0002	**AMERICA	94251493	Edell Rue	-21.9224	162.5258	salmon	70.205.32.111	Lamborghini	Charger	Extended Cab	Fiat Mercatago	56735	https://cdn.fakercloud.com/avatars

b. Logs: Muestra la información obtenida con tarjetas de crédito no validas.

Home Batch Admin Logs Logout																	

c. APIS

Se configuran todos los métodos para disponibilizar la información, uno de ellos entrega todos los datos (excepto la fecha) y el otra que entrega la información por id. Se habilita un API REST con el método GET:

app.py

database.py

```

#####
# APIs REST - GET
#####
#API REST GET SALES
class getSales(Resource):
    def get(self):
        ventas = database.getAllSales()
        return ventas

#API REST GET SALES BY ID
class getSalesById(Resource):
    def get(self, id):
        venta = database.getSale(id)
        return venta

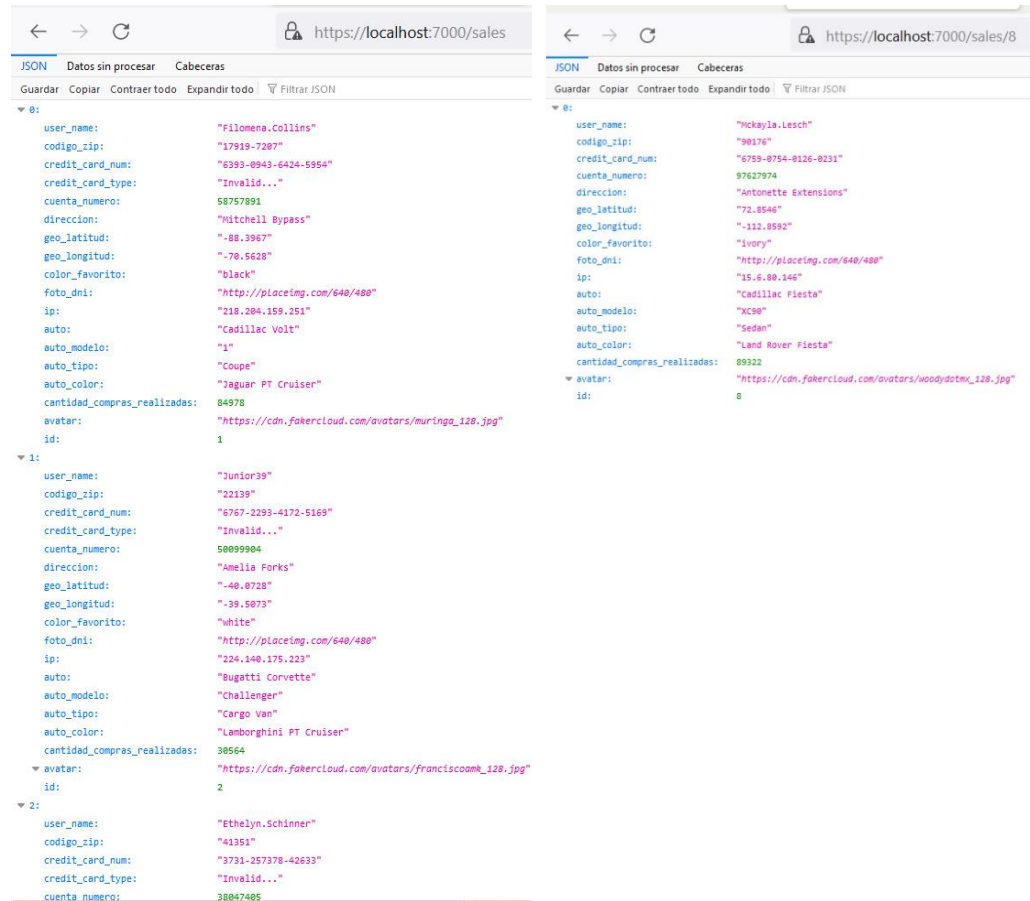
api.add_resource(getSales, '/sales')
api.add_resource(getSalesById, '/sales/<id>')

```

```

35
36 #Consultas para generar los APIs REST - GET
37
38 def getAllSales():
39     connect = open()
40     cursor = connect.cursor(dictionary=True)
41     cursor.execute("Select user_name,codigo_zip,credit_card_num,credit_card_type,cuenta_nu
42     sales = cursor.fetchall()
43     connect.close()
44     return sales
45
46
47 def getSale(id):
48     connect = open()
49     cursor = connect.cursor(dictionary=True)
50     #cursor.execute("Select user_name,codigo_zip,credit_card_num,cuenta_numero,direccion,s
51     cursor.execute("Select user_name,codigo_zip,credit_card_num,cuenta_numero,direccion,ge
52     sales = cursor.fetchall()
53     connect.close()

```

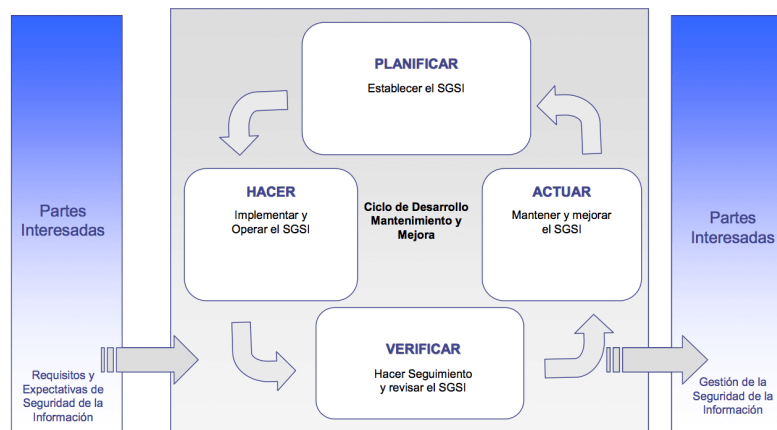


NOTA: Está pendiente implementar métodos de autenticación.

8. METODOLOGÍA DEL ANÁLISIS DE RIESGOS

Para la aplicación en la gestión de los riesgos, se tomó como referencia las actividades y procesos expuestos en la NTC-ISO/IEC 27005 dado que la administración del riesgo establece lineamientos precisos acerca del tratamiento, manejo y seguimiento a los riesgos aplicando el ciclo planificar, hacer, verificar y actuar (en adelante PHVA).

En la siguiente imagen, podemos detallar en resumen las actividades de este ciclo:



Fuente: NORMA TÉCNICA NTC-ISO/IEC COLOMBIANA 27001

Dado lo anterior, se realiza el siguiente análisis que nos permite aplicar el modelo en la solución planteada:

d. Paso 1. Establecimiento del Contexto

La empresa se dedica a la venta de automóviles, específicamente para personas a naturales:

DOFA	
Debilidades	Oportunidades
<ul style="list-style-type: none"> * Limitada capacidad financiera de algunos clientes. * Falta de marketing 	<ul style="list-style-type: none"> * Aumentar participación del mercado. * Lanzar líneas de lujo.
Fortalezas	Amenazas
<ul style="list-style-type: none"> * Buena percepción desde las entidades bancarias. * Conocimiento de clientes y proveedores. * Trato personalizado a potenciales clientes. 	<ul style="list-style-type: none"> * Inestabilidad económica e inflación. * Alzas en los precios debido a los costos de importación. * Competencia con marcas de renombre y de experiencia.

e. Paso 2. Valoración del Riesgo

La valoración del riesgo consta de las siguientes etapas:

- **Identificación de Activos:** Cualquier información, elemento o proceso que tienen valor y por consiguiente deben protegerse. No basta solo con

identificarlos, sino que se debe realizarse una breve descripción y calificación en términos de confidencialidad, disponibilidad e integridad para identificar la clasificación de la información:

CLASIFICACION DE INFORMACION (ACTIVOS)		
NOMBRE	VALOR	DESCRIPCION
No Aplica	Resultado es igual a 0	Se da por entendido que el activo es un proceso.
Pública	Resultado entre 1 y 3	Es aquella información que ha sido declarada de conocimiento público. Esta información puede ser entregada o publicada sin restricciones a terceros, funcionarios o cualquier persona sin ocasionar daños a terceros ni a
Uso interno	Resultado entre 4 y 6	Es aquella información que es utilizada para realizar las labores designadas en los procesos de la compañía y que no puede ser utilizada por terceros sin autorización del propietario.
Confidencial	Resultado entre 7 y 10	Información sensible, interna a áreas o proyectos a los que debe tener acceso controlado un grupo reducido de personas y no toda la empresa, que debe ser protegida por su impacto en los intereses de la empresa, de

Nota: Cuando el activo es un proceso no se realiza la calificación.

- **Identificación de Amenazas:** Una vez se determinan los activos, hay que listar las posibles amenazas que se pueden explotar.
- **Identificación de Controles:** Se realiza verificando y listando los controles que se tienen activos o que se pueden implementar por cada amenaza.

f. Paso 3. Analizar los Riesgos

En esta etapa de acuerdo con las causas, consecuencias positivas o negativas, fuentes de riesgo, se determina la probabilidad y el impacto, variables que nos permiten establecer el nivel de riesgo:

- **Probabilidad:** Se define como la posibilidad de ocurrencia en el tiempo del riesgo. Esta puede ser medida con criterios de frecuencia:

PROBABILIDAD		
Valor	Calificación	Frecuencia
1	Minima	Se puede materializar el riesgo al menos 1 vez en la compañía en la última década .
2	Baja	Se puede materializar el riesgo al menos 1 vez en la compañía en el último año .
3	Moderado	Se puede materializar el riesgo al menos 1 vez en la compañía en el último mes .
4	Alto	Se puede materializar el riesgo al menos 1 vez en una semana .
5	Muy Alto	Se puede materializar el riesgo al menos 1 vez en un día .

- **Impacto:** Se define como las consecuencias que puede ocasionar a la empresa la materialización del riesgo:

IMPACTO		
Valor	Calificación	Consecuencias
1	Inferior	No hay afectación significativa en los objetivos del proceso
2	Menor	Puede haber afectación baja en los objetivos del proceso.
3	Importante	Puede haber afectación significativa en los objetivos del proceso
4	Mayor	Puede haber afectación grave de los objetivos del proceso
5	Superior	Puede haber afectación desastrosa de los objetivos del proceso

- **Nivel del riesgo:** De acuerdo con el resultado entre la intersección entre la probabilidad y el impacto se establece el nivel del riesgo, el cual se determinará basado en la siguiente tabla:

PROBABILIDAD	Nivel						
	5	BAJO	MODERADO	ALTO	EXTREMO	EXTREMO	
	4	BAJO	MODERADO	ALTO	ALTO	EXTREMO	
	3	BAJO	MODERADO	MODERADO	ALTO	EXTREMO	
	2	BAJO	BAJO	MODERADO	ALTO	EXTREMO	
	1	BAJO	BAJO	MODERADO	ALTO	EXTREMO	
		1	2	3	4	5	Nivel
		IMPACTO					

g. Paso 4. Evaluación del Riesgo:

La evaluación del riesgo siempre se debe realizar con los dueños de proceso y la gerencia de la empresa ya que en este paso se compara el nivel de riesgo encontrado en el paso anterior y el nivel de aceptación que tiene la empresa.

A partir de esta comparación se debe determinar si la empresa acepta o no los riesgos

h. Paso 5. Tratamiento del Riesgo

Luego del resultado de la valoración del riesgo inherente se debe dar tratamiento a los riesgos que tienen calificaciones muy altas, implementando alguna de las estrategias ya conocidas:

- Reducir o mitigar
- Asumir o retener
- Evitar
- Transferir

En nuestro caso, se realizan estrategias de mitigación a través de controles. Cabe resaltar que al menos debe existir un control preventivo y uno detectivo para que el nivel de riesgo disminuya y proceder a calcular el nivel de riesgo residual.

Los siguientes pasos no fueron tenidos en cuenta para el este ejercicio, dado que se tratan de actividades recurrentes y de seguimiento. Sin embargo, hacen parte de la metodología planteada:

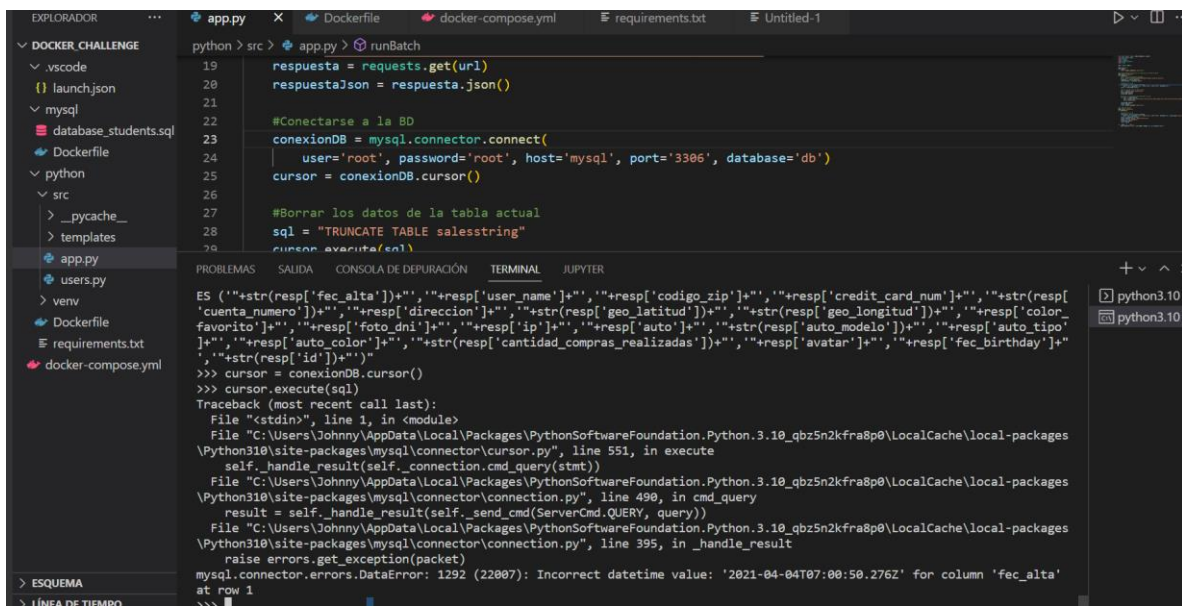
- i. Paso 6. Comunicación del Riesgo
- j. Paso 7. Monitoreo y Revisión del riesgo

9. PROBLEMAS ENCONTRADOS EN LA GENERACIÓN DEL CHALLENGE

a. Error en el campo fecha

- **Guardar en la BD:**

El formato del campo de fecha contiene las letras TyZ, lo cual no permitía almacenar la información en la base de datos:



The screenshot shows a VS Code editor with a file explorer on the left containing files like .vscode, launch.json, mysql, database_students.sql, Dockerfile, python, src, _pycache_, templates, app.py, users.py, venv, Dockerfile, requirements.txt, and docker-compose.yml. The main editor shows a Python script (app.py) with the following code:

```
python > src > app.py > runBatch
19 respuesta = requests.get(url)
20 respuestaJson = respuesta.json()
21
22 #Conectarse a la BD
23 conexionDB = mysql.connector.connect(
24     user='root', password='root', host='mysql', port='3306', database='db')
25 cursor = conexionDB.cursor()
26
27 #Borrar los datos de la tabla actual
28 sql = "TRUNCATE TABLE salesstring"
29 cursor.execute(sql)
```

The terminal at the bottom shows the execution of the script and a traceback error:

```
ES (''+str(resp['fec_alta'])+'''+'''+str(resp['user_name'])+'''+'''+str(resp['codigo_zip'])+'''+'''+str(resp['credit_card_num'])+'''+'''+str(resp[
'cuenta_numero'])+'''+'''+str(resp['direccion'])+'''+'''+str(resp['geo_latitud'])+'''+'''+str(resp['geo_longitud'])+'''+'''+str(resp['color_
favorito'])+'''+'''+str(resp['foto_dni'])+'''+'''+str(resp['ip'])+'''+'''+str(resp['auto'])+'''+'''+str(resp['auto_modelo'])+'''+'''+str(resp['auto_tipo'
])+'''+'''+str(resp['auto_color'])+'''+'''+str(resp['cantidad_compras_realizadas'])+'''+'''+str(resp['avatar'])+'''+'''+str(resp['fec_birthday'])+'''+
'''+str(resp['id'])+'''+''')
>>> cursor = conexionDB.cursor()
>>> cursor.execute(sql)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "C:\Users\Johnny\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages
\Python310\site-packages\mysql\connector\cursor.py", line 551, in execute
      self._handle_result(self._connection.cmd_query(stmt))
    File "C:\Users\Johnny\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages
\Python310\site-packages\mysql\connector\connection.py", line 490, in cmd_query
      result = self._handle_result(self._send_cmd(ServerCmd.QUERY, query))
    File "C:\Users\Johnny\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages
\Python310\site-packages\mysql\connector\connection.py", line 395, in _handle_result
      raise errors.get_exception(packet)
mysql.connector.errors.DataError: 1292 (22007): Incorrect datetime value: '2021-04-04T07:00:50.276Z' for column 'fec_alta'
at row 1
>>>
```

Después de mucho revisar información y sitios web, además de intentar diferentes soluciones (por ejemplo: con STR_TO_DATE de MySql), se encontró que con la librería **datetime** de Python se puede almacenar la fecha (incluyendo la máscara):

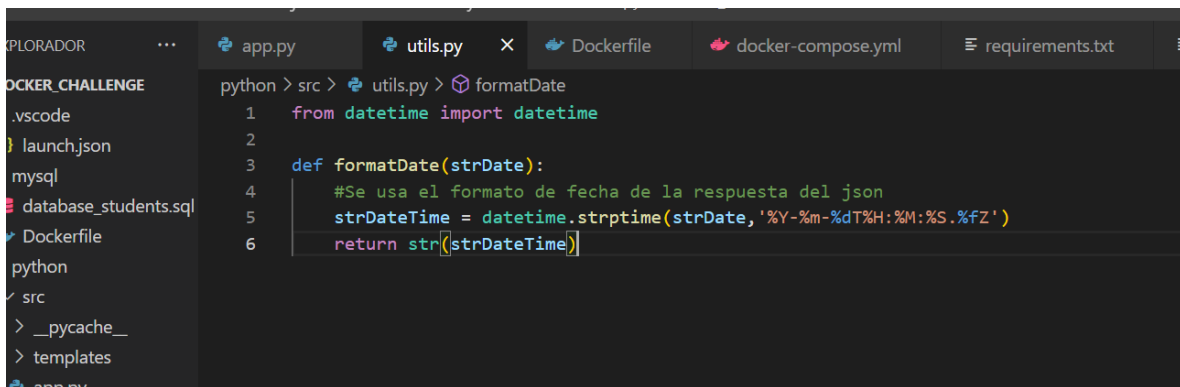
strftime() and strptime() Behavior

`date`, `datetime`, and `time` objects all support a `strftime(format)` method, to create a string representing the time under the control of an explicit format string.

Conversely, the `datetime.strptime()` class method creates a `datetime` object from a string representing a date and time and a corresponding format string.

The table below provides a high-level comparison of `strftime()` versus `strptime()`:

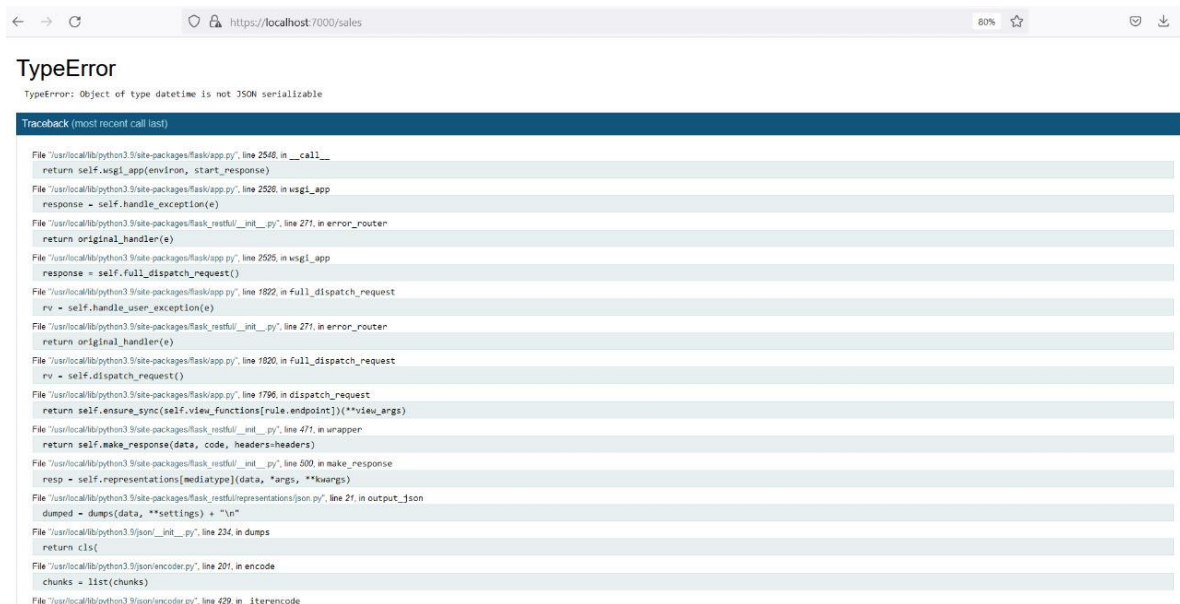
	<code>strftime</code>	<code>strptime</code>
Usage	Convert object to a string according to a given format	Parse a string into a <code>datetime</code> object given a corresponding format
Type of method	Instance method	Class method
Method of	<code>date</code> ; <code>datetime</code> ; <code>time</code>	<code>datetime</code>
Signature	<code>strftime(format)</code>	<code>strptime(date_string, format)</code>



```
python > src > utils.py > formatDate
1 from datetime import datetime
2
3 def formatDate(strDate):
4     #Se usa el formato de fecha de la respuesta del json
5     strDateTime = datetime.strptime(strDate, '%Y-%m-%dT%H:%M:%S.%fZ')
6     return str(strDateTime)
```

- **API**

Al realizar el envío de la información con el campo fecha, se genera el siguiente error:



```
TypeError: Object of type datetime is not JSON serializable

Traceback (most recent call last):
  File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 2548, in __call__
    return self.wsgi_app(environ, start_response)
  File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 2526, in wsgi_app
    response = self.handle_exception(e)
  File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 271, in error_router
    return original_handler(e)
  File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 2526, in wsgi_app
    response = self.full_dispatch_request()
  File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1822, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 271, in error_router
    return original_handler(e)
  File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1820, in full_dispatch_request
    rv = self.dispatch_request()
  File "/usr/local/lib/python3.9/site-packages/flask/app.py", line 1766, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
  File "/usr/local/lib/python3.9/site-packages/flask_restful/_init.py", line 471, in wrapper
    return self.make_response(data, code, headers=headers)
  File "/usr/local/lib/python3.9/site-packages/flask_restful/_init.py", line 500, in make_response
    resp = self.representations[mediatype](data, *args, **kwargs)
  File "/usr/local/lib/python3.9/site-packages/flask_restful/representations/json.py", line 21, in output_json
    dumped = dumps(data, **settings) + "\n"
  File "/usr/local/lib/python3.9/site-packages/flask/json/_init.py", line 234, in dumps
    return cls(
  File "/usr/local/lib/python3.9/site-packages/flask/json/encoder.py", line 201, in encode
    chunks = list(chunks)
  File "/usr/local/lib/python3.9/site-packages/flask/json/encoder.py", line 429, in _iterencode
```

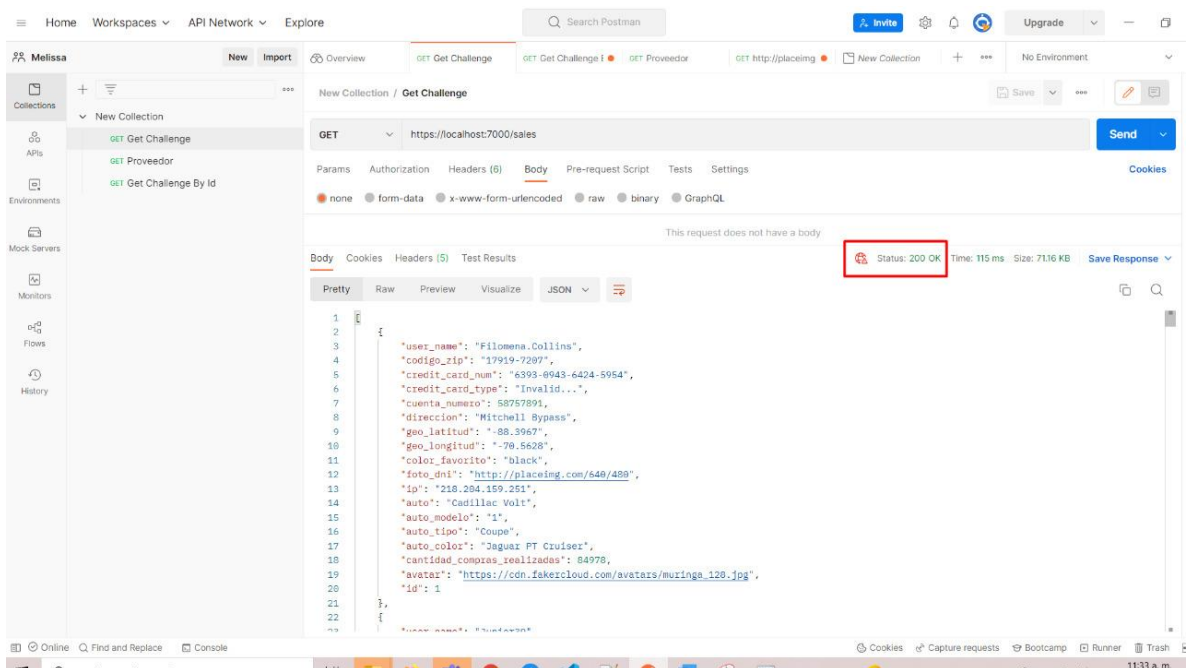
Se evidencia que el campo que está generando la inconsistencia es el de la fecha:

```
34
35
36 #Consultas para generar los APIs REST - GET
37
38 def getAllSales():
39     connect = open()
40     cursor = connect.cursor(dictionary=True)
41     cursor.execute("Select fec_alta, user_name, codigo_zip, credit_card_num, credit_card_type, cuenta_numero, dire
42     sales = cursor.fetchall()
43     connect.close()
44     return sales
45
46
47 def getSale(id):
48     connect = open()
49     cursor = connect.cursor(dictionary=True)
50     #cursor.execute("Select user_name, codigo_zip, credit_card_num, cuenta_numero, direccion, geo_latitud, geo_long
51     cursor.execute("Select user_name, codigo_zip, credit_card_num, cuenta_numero, direccion, geo_latitud, geo_longi
52     sales = cursor.fetchall()
53     connect.close()
54     return sales
```

Dado lo anterior, se modifican las consultas de los APIs para que no se incluya dicho campo y funciona correctamente:

```
36 #Consultas para generar los APIs REST - GET
37
38 def getAllSales():
39     connect = open()
40     cursor = connect.cursor(dictionary=True)
41     cursor.execute("Select user_name, codigo_zip, credit_card_num, credit_card_type, cuenta_numero, direccion, geo
42     sales = cursor.fetchall()
43     connect.close()
44     return sales
45
46
47 def getSale(id):
48     connect = open()
49     cursor = connect.cursor(dictionary=True)
50     #cursor.execute("Select user_name, codigo_zip, credit_card_num, cuenta_numero, direccion, geo_latitud, geo_long
51     cursor.execute("Select user_name, codigo_zip, credit_card_num, cuenta_numero, direccion, geo_latitud, geo_longi
52     sales = cursor.fetchall()
53     connect.close()
```

Se realizan pruebas en Postman:

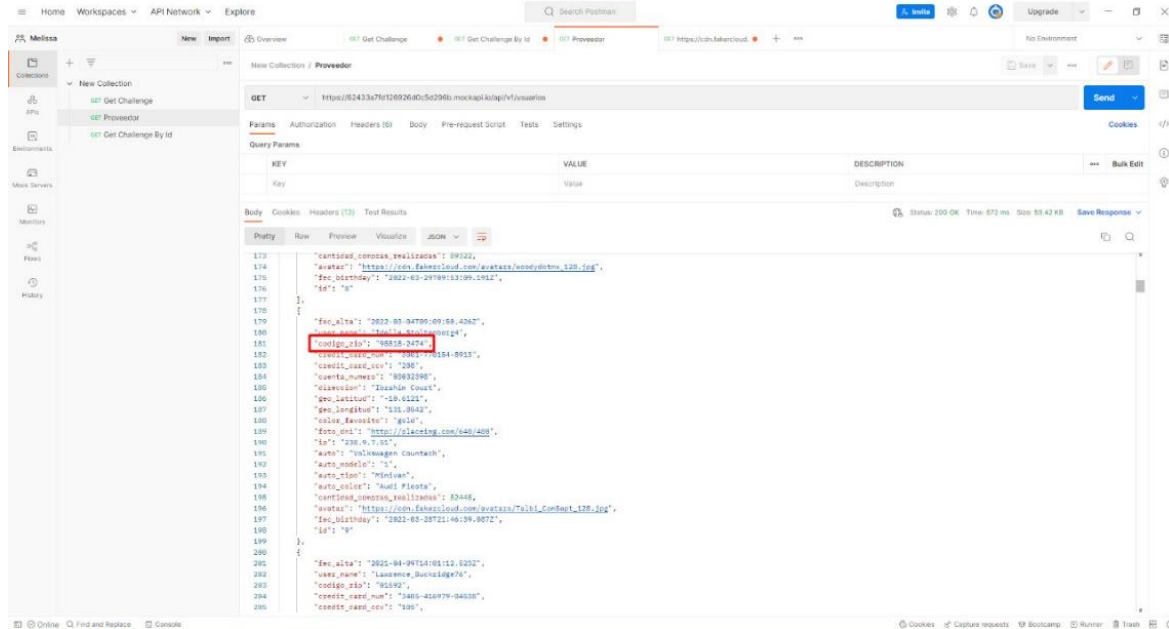


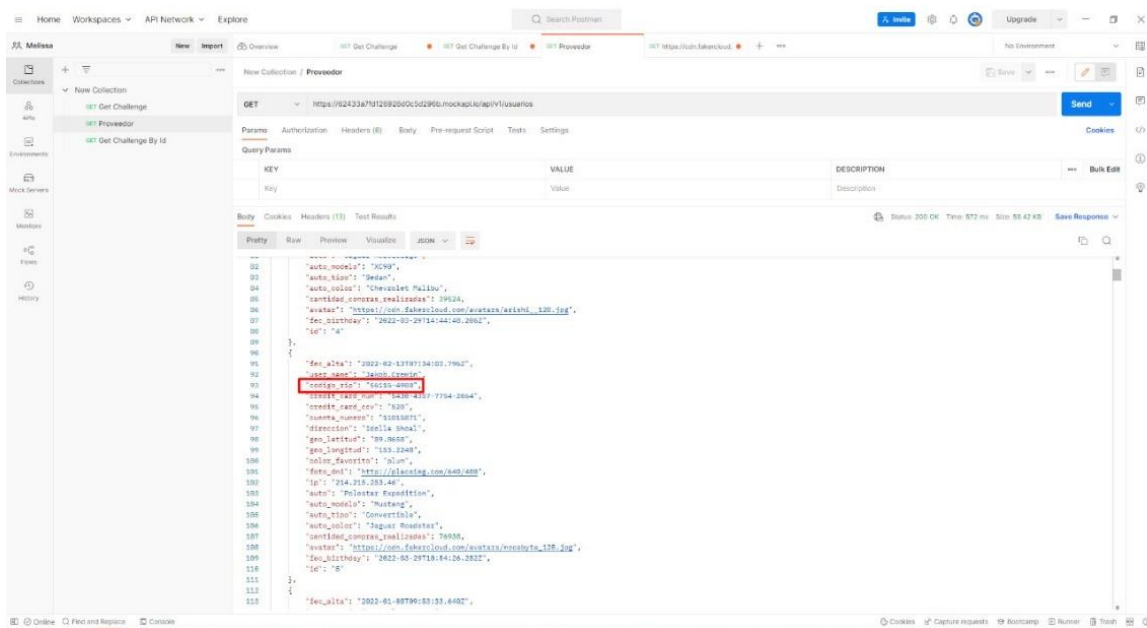
b. Error en el campo codigo_zip

Se creó el campo en la BD como tipo numérico, pero generaba el siguiente error:

```
File "C:\Users\Johnny\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages
Python310\site-packages\mysql\connector\cursor.py", line 551, in execute
    self._handle_result(self._connection.cmd_query(stmt))
File "C:\Users\Johnny\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages
Python310\site-packages\mysql\connector\connection.py", line 490, in cmd_query
    result = self._handle_result(self._send_cmd(ServerCmd.QUERY, query))
File "C:\Users\Johnny\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages
Python310\site-packages\mysql\connector\connection.py", line 395, in _handle_result
    raise errors.get_exception(packet)
mysql.connector.errors.DatabaseError: 1265 (01000): Data truncated for column 'codigo_zip' at row 1
>>> print(sql)
INSERT INTO sales (fec_alta,user_name,codigo_zip,credit_card_num,cuenta_numero,direccion,geo_latitud,geo_longitud,color_fav
orito,foto_dni,ip,auto,auto_modelo,auto_tipo,auto_color,cantidad_compras_realizadas,avatar,fec_birthday,id) VALUES ('2021-0
1-04 07:00:50.276000','Filomena.Collins','17919-7207','6393-0943-6424-5954','58757891','Mitchell Bypass','-88.3967','-70.56
28','black','http://placeimg.com/640/480','218.204.159.251','Cadillac Volt','1','Coupe','Jaguar PT Cruiser','84978','https:
/cdn.fakercloud.com/avatars/muringa_128.jpg','2022-03-28T21:18:02.439Z','1')
\\>>>
```

Se valida la información del GET desde el Postman (en el Endpoint del Proveedor):



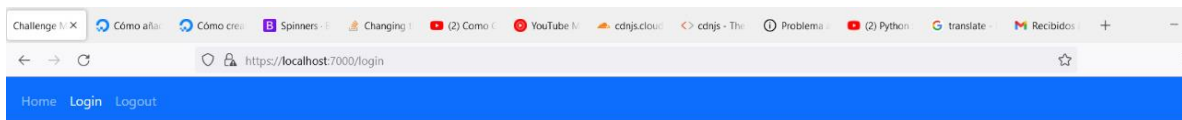


Por lo anterior, se determina que la mejor opción es modificar el campo de un tipo numérico a un tipo texto (varchar).

b. Pruebas de SQL Injection

Se realizan dos pruebas de SQL Injection:

- Ingresando ' (comilla simple) en el campo usuario:



Iniciar sesión

☐ Remember me

Inicia Sesión

El error retorna el nombre del motor de base de datos que se está usando:

ProgrammingError

mysql.connector.errors.ProgrammingError: 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''' and password=''' at line 1

Traceback (most recent call last)

```
File "usr/local/lib/python3.9/site-packages/mysql/connector/connection_cext.py", line 555, in cmd_query
    self._cmysql.query(
```

The above exception was the direct cause of the following exception:

```
File "usr/local/lib/python3.9/site-packages/flask/app.py", line 2548, in __call__
    return self.wsgi_app(environ, start_response)
```

```
File "usr/local/lib/python3.9/site-packages/flask/app.py", line 2528, in wsgi_app
    response = self.handle_exception(e)
```

```
File "usr/local/lib/python3.9/site-packages/flask_restful/_init_.py", line 271, in error_router
    return original_handler(e)
```

```
File "usr/local/lib/python3.9/site-packages/flask/app.py", line 2525, in wsgi_app
    response = self.full_dispatch_request()
```

```
File "usr/local/lib/python3.9/site-packages/flask/app.py", line 1822, in full_dispatch_request
    rv = self.handle_user_exception(e)
```

```
File "usr/local/lib/python3.9/site-packages/flask_restful/_init_.py", line 271, in error_router
    return original_handler(e)
```

- Ingresando **admin';;--** en el campo usuario:

[Home](#) [Login](#) [Logout](#)

Iniciar sesión

☐ Remember me

El usuario inicia sesión en la aplicación (sin necesidad de ingresar el password) y muestra la información:

Challenge X

https://localhost:7000/admin

Home Admin Batch Logout

Id	Fecha de Alta	Nombre	Código Postal	# Tarjeta de Crédito	# Cuenta	Dirección	Latitud	Longitud	Color Favorito	Foto	IP	Auto	Modelo Auto	Tipo Auto	Color Auto
1		Filomena.Collins	17919-7207	3	58757891	Mitchell Bypass	-88.3967	-70.5628	black	http://placeimg.com/640/480	218.204.159.251	Cadillac Volt	1	Coupe	Jaguar PT Cruiser
2		Junior39	22139	3	50099904	Amelia Forks	-40.0728	-39.5073	white	http://placeimg.com/640/480	224.140.175.223	Bugatti Corvette	Challenger	Cargo Van	Lamborghini PT Cruiser
3		Ethelyn.Schinner	41351	3	38047405	Shaniya Springs	10.3752	-105.7502	gold	http://placeimg.com/640/480	190.130.230.168	Volkswagen Cruze	Model S	Hatchback	Jaguar Camaro
4		Lonie85	50853	3	35431462	Brakus Isle	11.4957	55.9517	azure	http://placeimg.com/640/480	156.89.51.113	Jaguar Mercielago	XC90	Sedan	Chevrolet Malibu
5		Jakob.Cremin	56115-4988	3	11015871	Idella Shoal	89.8658	153.2248	plum	http://placeimg.com/640/480	214.215.253.46	Polestar Expedition	Mustang	Convertible	Jaguar Roadster
6		Juvenal.Larson92	59210	3	72588644	Quigley Shoal	-35.5623	59.5388	lime	http://placeimg.com/640/480	63.31.243.1	Maserati Grand Caravan	Civic	Crew Cab Pickup	Polestar Model T
7		Jeff.Greenholt	93737	3	78676257	Sophia Lodge	-38.7054	-92.9974	magenta	http://placeimg.com/640/480	76.249.54.121	Smart Durango	Volt	Extended Cab Pickup	Chrysler Corvette
8		Mckayla.Lesch	90176	3	97627974	Antonette Extensions	72.8546	-112.8592	ivory	http://placeimg.com/640/480	15.6.80.146	Cadillac Fiesta	XC90	Sedan	Land Rover Fiesta

Escribe aquí para buscar

11°C Nublado 10:41 p.m. 14/06/2022

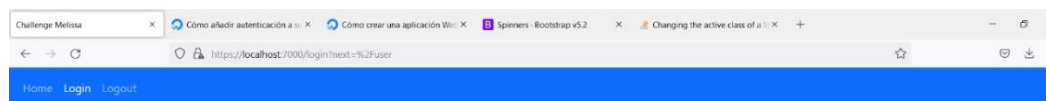
Se modificó la sentencia para consultar el usuario y el password en la base de datos por parámetros:

```
#Validación del usuario y el password
def validateUser(u, p):
    connect = open()
    cursor = connect.cursor(dictionary=True)
    #sql = "select count(1) from user where name='"+u+"' and password='"+p+"'
    #cursor.execute(sql)

    pHashed = hashlib.sha256(p.encode('utf-8')).hexdigest()
    pHashedStr = str(pHashed)

    cursor.execute("select responsibility from user where name = %(u)s and password = %(p)s", {'u': u, 'p': pHashedStr})
    result = cursor.fetchone()
    connect.close()
    return result
```

Este cambio, solucionó el ingreso a través de SQL Injection:



Iniciar sesión

Please log in to access this page.

Nombre

Password

☐ Remember me

Inicia Sesión

c. HTTPS (Hypertext Transfer Protocol Secure):

Para este error no se tiene imagen de evidencia. Sin embargo, llegar a la solución conlleva a modificar el contenedor de Python (con Sistema Operativo Alpine) y su respectiva imagen, dado que no soportaba los paquetes (librerías) que se estaban instalando.

Por lo anterior, se procedió a descargar una nueva imagen de Python (con Sistema Operativo Debian GNU/Linux) que soportó la librería que se estaba usando para habilitar el HTTPS de la aplicación.

10. CONCLUSIONES

Las diferentes empresas deben considerar la seguridad de la información como un valor que agrega confiabilidad para sus clientes y proveedores y no como un costo adicional, el costo-beneficio de implementar controles debe enfocarse en contrarrestar en la medida de lo posible, las nuevas amenazas que día a día emergen con el único propósito de identificar y aprovechar las vulnerabilidades informáticas con fines delictivos.

La gestión de riesgos se ha constituido en una de las principales herramientas para estar preparados contra todas las amenazas de nuestros sistemas de información, lo cual conlleva a considerar el uso de las mejores prácticas y estándares. Logrando como eje primordial ser preventivos para mitigar los riesgos dado que el monitoreo de estos desde etapas tempranas nos permite reducir los costos y pérdidas asociados en la materialización de los riesgos, para enfrentarlos de manera proactiva y con la misma dinámica con que aparecen.