**PRUEBA TÉCNICA ARQUITECTO DE DATOS JIKKOSOFT S.A.S**

Sección 6: Codificación y secuencias de comandos

1. Escriba un script de Python para conectarse a una base de datos y recuperar datos de una tabla. Maneje errores y asegure conexiones. Incluir operaciones adicionales, como inserción y actualización de datos.

Se realiza Script con las operaciones de recuperación de datos, inserción y actualización de registros en las tablas de ejemplo de MySql.

Script: scriptsPython\Seccion6Scrip1.py

import mysql.connector

from mysql.connector import Error

def conectar\_bd(server\_name, nombre\_usuario, pwd):

    connection = None

    try:

        connection = mysql.connector.connect(

            host=server\_name,

            user=nombre\_usuario,

            passwd=pwd

        )

        print("Conexión a la base de datos exitosa")

    except Error as err:

        print(f"Error conectando a la base de datos: {err}")

    return connection

def close\_bd(connection):

    if connection.is\_connected():

        connection.close()

        print("Conexión a la base de datos cerrada")

def ejecutar\_consulta(connection, query):

    cursor = connection.cursor()

    try:

        cursor.execute(query)

        connection.commit()

        print("La consulta ejecutada exitosamente")

    except Error as err:

        print(f"Error hejecutando la consulta: {err}")

def conusultar\_datos(connection, query):

    cursor = connection.cursor(dictionary=True)

    result = None

    try:

        cursor.execute(query)

        result = cursor.fetchall()

        return result

    except Error as err:

        print(f"Error ejecutando la consulta: {err}")

        return None

#Logica del proceso

# Parámetros de conexión

host = "localhost"

user = "root"

password = "Manuela04."

# Crear la conexión a la base de datos

connection = conectar\_bd(host, user, password)

#Si la connexión fue correcta, Realiza la recuperación de la información

if connection:

    # Recupera lo datos de la tabla y los imprime en pantalla

    select\_query = "SELECT \* FROM sakila.customer LIMIT 10"

    rows = conusultar\_datos(connection, select\_query)

    if rows:

        for row in rows:

            print(row)

 # Realizamos Inserciones a la tabla category

    insert\_query = "INSERT INTO sakila.category (name) VALUES('NuevaCategoria');"

    ejecutar\_consulta(connection, insert\_query)

 # Realizamos actaulizaciones a la tabla category

    update\_query = "UPDATE sakila.category SET name = 'CategoriaNuevoNombre'  WHERE category\_id = 21"

    ejecutar\_consulta(connection, update\_query)

# Cierra la conexión a ala base de datos

close\_bd(connection)

1. Cree un script ETL simple utilizando un lenguaje de su elección (por ejemplo, Python, Java) para extraer datos de un archivo CSV, transformarlo eliminando duplicados y cargarlo en una base de datos. Incluya transformaciones más complejas, como mapeo de campos y agregaciones.