

Primer Parcial

Serie I

2. a) save()
4. a) sqrt()
6. a) round()
8. a) mean()
10. c) acos()
12. a) size()
14. a) mean()
16. a) round()
18. b) fft()
20. a) save()
22. c) PsqI
24. c) \l
26. a) CREATE TABLE
28. c) INSERT INTO
30. a) UPDATE
32. a) GNU General Public License
34. a) Create Database
36. b) Una estructura de datos que acelera consultas
38. c) Una tabla virtual que se genera a partir de una consulta
40. a) Una situación en la que dos o más transacciones quedan bloqueadas definitivamente
42. a) UPDATE
44. d) INSERT
46. b) ALTER
48. a) SELECT
50. d) ORDER BY

Serie II

2. Es un conjunto de campos en una tabla de base de datos que sirve para identificar de forma única cada registro en la tabla
4. La operación que se utiliza es DELETE
6. Es un tipo de filtro que permite seleccionar los registros que cumplan una determinada condición
8. Ventana que nos permite ver los datos de una o varias tablas
10. Permite realizar tareas complejas con datos
12. Código:

$$A = [2, 4, 6; 8, 10, 12]$$

$$B = [2, 5, 7; 9, 1, 11]$$

$$C = A \times B$$

}

Multiplicación de matrices
14. Se calcula utilizando la función `mean()`

$$v = [1, 2, 3]$$

$$\text{mean}(v);$$
16. Se calcula con el comando `sqrt`

$$C = 64$$

$$A = \text{sqrt}(C)$$
18. Utilizando la función `max`

$$v = [1, 2, 3, 4]$$

$$\text{max}(v)$$
20. Con la función `rand()`

$\text{rand}() \Rightarrow$ Genera aleatorio entre 0 y 1

$\text{rand}(10) =$

✓

✓

✓

0 y 10

Serie III

- Programa 1

```
import psycopg2

def connect_to_database():
    try:
        connection = psycopg2.connect(
            user="postgres",
            password="2405",
            host="localhost",
            port="5433",
            database="0980 Proyectos"
        )
        return connection
    except (Exception, psycopg2.Error) as error:
        print("Error al conectar a la base de datos:", error)
        return None

def insert_student(connection, nombre, edad, genero, direccion):
    try:
        cursor = connection.cursor()
        insert_query = '''INSERT INTO estudiantes (nombre, edad,
genero, direccion)
                        VALUES (%s, %s, %s, %s);'''
        data = (nombre, edad, genero, direccion)
        cursor.execute(insert_query, data)
        connection.commit()
        cursor.close()
        print("Estudiante agregado con éxito.")
    except (Exception, psycopg2.Error) as error:
        print("Error al agregar estudiante:", error)

def update_student(connection, estudiante_id, campo, nuevo_valor):
    try:
        cursor = connection.cursor()
        update_query = f'''UPDATE estudiantes
                        SET {campo} = %s
                        WHERE id = %s;'''
        data = (nuevo_valor, estudiante_id)
        cursor.execute(update_query, data)
        connection.commit()
        cursor.close()
        print("Información del estudiante actualizada con éxito.")
    except (Exception, psycopg2.Error) as error:
```

```

•         print("Error al actualizar información del estudiante:",
•         error)
•
•
•     def delete_student(connection, estudiante_id):
•         try:
•             cursor = connection.cursor()
•             delete_query = '''DELETE FROM estudiantes WHERE id = %s;'''
•             cursor.execute(delete_query, (estudiante_id,))
•             connection.commit()
•             cursor.close()
•             print("Estudiante eliminado de la base de datos.")
•         except (Exception, psycopg2.Error) as error:
•             print("Error al eliminar estudiante:", error)
•
•
•     def main():
•         connection = connect_to_database()
•         if connection:
•             while True:
•                 print("\nOpciones:")
•                 print("1. Agregar estudiante")
•                 print("2. Editar información de estudiante")
•                 print("3. Eliminar estudiante")
•                 print("4. Salir")
•                 opcion = input("Selecciona una opción: ")
•
•                 if opcion == "1":
•                     nombre = input("Nombre del estudiante: ")
•                     edad = int(input("Edad: "))
•                     genero = input("Género: ")
•                     direccion = input("Dirección: ")
•                     insert_student(connection, nombre, edad, genero,
•                     direccion)
•
•                 elif opcion == "2":
•                     estudiante_id = int(input("ID del estudiante a editar:
•                     "))
•
•                     campo = input("Campo a editar
•                     (nombre/edad/genero/direccion): ")
•                     nuevo_valor = input(f"Nuevo valor para {campo}: ")
•                     update_student(connection, estudiante_id, campo,
•                     nuevo_valor)
•
•                 elif opcion == "3":
•                     estudiante_id = int(input("ID del estudiante a
•                     eliminar: "))
•
•                     delete_student(connection, estudiante_id)
•
•                 elif opcion == "4":

```

```

•         print("Saliendo del programa.")
•         break
•     else:
•         print("Opción no válida. Por favor, selecciona una
opción válida.")
•         connection.close()
•
• if __name__ == "__main__":
•     main()
•

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

2. Editar información de estudiante
3. Eliminar estudiante
4. Salir
Selecciona una opción: 1
Nombre del estudiante: Francisco
Edad: 21
Género: M
Dirección: Casa 30. Lomas IV
Estudiante agregado con éxito.

```

Data Output Messages Notifications					
	id [PK] integer	nombre character varying (255)	edad integer	genero character varying (255)	direccion character varying (255)
1	1	Francisco	21	M	Casa 30. Lomas IV

- Segundo

```

• import psycopg2
•
• def connect_to_database():
•     try:
•         connection = psycopg2.connect(
•             user="postgres",
•             password="2405",
•             host="localhost",
•             port="5433",
•             database="0980 Proyectos"
•         )
•         return connection
•     except (Exception, psycopg2.Error) as error:

```

```

•         print("Error al conectar a la base de datos:", error)
•         return None
•
•
• def insert_expense(connection, fecha, categoria, descripcion, monto):
•     try:
•         cursor = connection.cursor()
•         insert_query = '''INSERT INTO gastos (fecha, categoria,
•             descripcion, monto)
•                 VALUES (%s, %s, %s, %s);'''
•         data = (fecha, categoria, descripcion, monto)
•         cursor.execute(insert_query, data)
•         connection.commit()
•         cursor.close()
•         print("Gasto agregado con éxito.")
•     except (Exception, psycopg2.Error) as error:
•         print("Error al agregar gasto:", error)
•
•
• def generate_report(connection):
•     try:
•         cursor = connection.cursor()
•         report_query = '''SELECT fecha, categoria, descripcion, monto
•             FROM gastos
•             ORDER BY fecha DESC;'''
•         cursor.execute(report_query)
•         results = cursor.fetchall()
•         cursor.close()
•         print("Reporte de gastos:")
•         for row in results:
•             print(f"Fecha: {row[0]}, Categoría: {row[1]}, Descripción:
• {row[2]}, Monto: {row[3]}")
•     except (Exception, psycopg2.Error) as error:
•         print("Error al generar reporte:", error)
•
•
• def adjust_budget(connection, categoria, nuevo_monto):
•     try:
•         cursor = connection.cursor()
•         update_query = f'''UPDATE presupuestos
•             SET monto = %s
•             WHERE categoria = %s;'''
•         data = (nuevo_monto, categoria)
•         cursor.execute(update_query, data)
•         connection.commit()
•         cursor.close()
•         print("Presupuesto ajustado con éxito.")
•     except (Exception, psycopg2.Error) as error:

```

```

•         print("Error al ajustar presupuesto:", error)
•
•
• def main():
•     connection = connect_to_database()
•     if connection:
•         while True:
•             print("\nOpciones:")
•             print("1. Agregar gasto")
•             print("2. Ver informe de gastos")
•             print("3. Ajustar presupuesto")
•             print("4. Salir")
•             opcion = input("Selecciona una opción: ")
•
•             if opcion == "1":
•                 fecha = input("Fecha del gasto (YYYY-MM-DD): ")
•                 categoria = input("Categoría del gasto: ")
•                 descripcion = input("Descripción del gasto: ")
•                 monto = float(input("Monto del gasto: "))
•                 insert_expense(connection, fecha, categoria,
descripcion, monto)
•             elif opcion == "2":
•                 generate_report(connection)
•             elif opcion == "3":
•                 categoria = input("Categoría del presupuesto a
ajustar: ")
•                 nuevo_monto = float(input("Nuevo monto del
presupuesto: "))
•                 adjust_budget(connection, categoria, nuevo_monto)
•             elif opcion == "4":
•                 print("Saliendo del programa.")
•                 break
•             else:
•                 print("Opción no válida. Por favor, selecciona una
opción válida.")
•                 connection.close()
•
• if __name__ == "__main__":
•     main()
•
•

```

```
1. Agregar gasto
2. Ver informe de gastos
3. Ajustar presupuesto
4. Salir
Selecciona una opción: 1
Fecha del gasto (YYYY-MM-DD): 2023-02-09
Categoría del gasto: Comida
Descripción del gasto: 1 Almuerzo
Monto del gasto: 50.00
Gasto agregado con éxito.
```

Data Output Messages Notifications

	id [PK] integer	fecha date	categoria character varying (255)	descripcion character varying (255)	monto numeric (10,2)
1	1	2023-08-21	Comida	almuerzo	50.00
2	2	2023-05-08	comida	1 almuerzo	30.00
3	3	2023-09-14	Comida	Cena	50.00

- Tercero:

```
import psycopg2

def connect_to_database():
    try:
        connection = psycopg2.connect(
            user="postgres",
            password="2405",
            host="localhost",
            port="5433",
            database="0980 Proyectos"
        )
        return connection
    except (Exception, psycopg2.Error) as error:
        print("Error al conectar a la base de datos:", error)
        return None

def insert_product(connection, nombre, descripcion, cantidad, precio):
    try:
```



```

        cursor = connection.cursor()
        insert_query = '''INSERT INTO inventarios (nombre, descripcion,
cantidad, precio)
                        VALUES (%s, %s, %s, %s);'''
        data = (nombre, descripcion, cantidad, precio)
        cursor.execute(insert_query, data)
        connection.commit()
        cursor.close()
        print("Producto agregado con éxito.")
    except (Exception, psycopg2.Error) as error:
        print("Error al agregar producto:", error)

def main():
    connection = connect_to_database()
    if connection:
        while True:
            print("\nOpciones:")
            print("1. Agregar producto")
            print("2. Salir")
            opcion = input("Selecciona una opción: ")

            if opcion == "1":
                nombre = input("Nombre del producto: ")
                descripcion = input("Descripción del producto: ")
                cantidad = int(input("Cantidad del producto: "))
                precio = float(input("Precio del producto: "))
                insert_product(connection, nombre, descripcion, cantidad,
precio)
            elif opcion == "2":
                print("Saliendo del programa.")
                break
            else:
                print("Opción no válida. Por favor, selecciona una opción
válida.")
        connection.close()

if __name__ == "__main__":
    main()

```

```
1. Agregar producto
2. Salir
Selecciona una opción: 1
Nombre del producto: Tornillos
Descripción del producto: Tornillos de acero
Cantidad del producto: 25
Precio del producto: 32.00
Error al agregar producto: transacción abortada, las órdenes serán ignoradas hasta el fin de bloque de transacción
```

Data Output

Messages

Notifications

≡+

▼

▼

	id [PK] integer	nombre character varying (255)	descripcion character varying (255)	cantidad integer	precio numeric (10,2)

- Cuarto:

- Quinto:

```

• import psycopg2
•
• def connect_to_database():
•     try:
•         connection = psycopg2.connect(
•             user="postgres",
•             password="2405",
•             host="localhost",
•             port="5433",
•             database="0980 Proyectos"
•         )
•         return connection
•     except (Exception, psycopg2.Error) as error:
•         print("Error al conectar a la base de datos:", error)
•         return None
•
• def insert_sale(connection, fecha, producto, cantidad, precio):
•     try:
•         cursor = connection.cursor()
•         insert_query = '''INSERT INTO ventas (fecha, producto,
cantidad, precio)
VALUES (%s, %s, %s, %s);'''
•         data = (fecha, producto, cantidad, precio)
•         cursor.execute(insert_query, data)
•         connection.commit()

```

```

•         cursor.close()
•         print("Venta agregada con éxito.")
•     except (Exception, psycopg2.Error) as error:
•         print("Error al agregar venta:", error)
•
•
• def generate_report(connection):
•     try:
•         cursor = connection.cursor()
•         report_query = '''SELECT fecha, producto, cantidad, precio
•                             FROM ventas
•                             ORDER BY fecha DESC;'''
•
•         cursor.execute(report_query)
•         results = cursor.fetchall()
•         cursor.close()
•         print("Reporte de ventas:")
•         for row in results:
•             print(f"Fecha: {row[0]}, Producto: {row[1]}, Cantidad:
• {row[2]}, Precio: {row[3]}")
•         except (Exception, psycopg2.Error) as error:
•             print("Error al generar reporte:", error)
•
•
• def analyze_data(connection):
•     try:
•         cursor = connection.cursor()
•         analysis_query = '''SELECT producto, SUM(cantidad) AS
cantidad,
•
•                             SUM(precio) AS precio
•                             FROM ventas
•                             GROUP BY producto;'''
•
•         cursor.execute(analysis_query)
•         results = cursor.fetchall()
•         cursor.close()
•         for row in results:
•             print(f"Producto: {row[0]}, Cantidad: {row[1]}, Precio:
{row[2]}")
•         print("Patrones y tendencias:")
•         # Aquí puedes agregar tu código para analizar los datos y
encontrar patrones y tendencias.
•         # Por ejemplo, puedes calcular el crecimiento de las ventas,
el producto más vendido, etc.
•         except (Exception, psycopg2.Error) as error:
•             print("Error al analizar datos:", error)
•
•
• def main():
•     connection = connect_to_database()

```

```

•     if connection:
•         while True:
•             print("\nOpciones:")
•             print("1. Agregar venta")
•             print("2. Generar reporte")
•             print("3. Analizar datos")
•             print("4. Salir")
•             opcion = input("Selecciona una opción: ")
•
•             if opcion == "1":
•                 fecha = input("Fecha de la venta (YYYY-MM-DD): ")
•                 producto = input("Producto: ")
•                 cantidad = int(input("Cantidad: "))
•                 precio = float(input("Precio: "))
•                 insert_sale(connection, fecha, producto, cantidad,
precio)
•
•                 elif opcion == "2":
•                     generate_report(connection)
•                 elif opcion == "3":
•                     analyze_data(connection)
•                 elif opcion == "4":
•                     print("Saliendo del programa.")
•                     break
•                 else:
•                     print("Opción no válida. Por favor, selecciona una
opción válida.")
•                     connection.close()
•
• if __name__ == "__main__":
•     main()
•

```

```

1. Agregar venta
2. Generar reporte
3. Analizar datos
4. Salir
Selecciona una opción: 1
Fecha de la venta (YYYY-MM-DD): 2022-06-09
Producto: TV
Cantidad: 6
Precio: 50.00
Venta agregada con éxito.

```

```

Opciones:
1. Agregar venta
2. Generar reporte
3. Analizar datos
4. Salir
Selecciona una opción: 2
Reporte de ventas:
Fecha: 2023-09-16, Producto: Mouse, Cantidad: 2, Precio: 200.00
Fecha: 2023-04-05, Producto: tele, Cantidad: 1, Precio: 100.00
Fecha: 2022-06-09, Producto: TV, Cantidad: 6, Precio: 50.00
Fecha: 2021-05-03, Producto: laptop, Cantidad: 5, Precio: 500.00

Opciones:
1. Agregar venta

```

Data Output Messages Notifications						
	id [PK] integer	fecha date	producto character varying (255)	cantidad integer	precio numeric (10,2)	
1	1	2023-04-05	tele	1	100.00	
2	2	2021-05-03	laptop	5	500.00	
3	3	2023-09-16	Mouse	2	200.00	
4	4	2022-06-09	TV	6	50.00	

- Sexto:
- Séptimo
- Octavo
- Noveno
- Decimo
- Onceavo
- Doceavo:

```

• import psycopg2
•
• def connect_to_database():
•     try:
•         connection = psycopg2.connect(
•             user="postgres",
•             password="2405",
•             host="localhost",
•             port="5433",
•             database="0980 Proyectos"
•         )
•         return connection
•     except (Exception, psycopg2.Error) as error:
•         print("Error al conectar a la base de datos:", error)
•         return None
•
• def show_all_questions(connection):
•     try:
•         cursor = connection.cursor()
•         query = "SELECT pregunta, respuesta FROM futbol"
•         cursor.execute(query)
•         results = cursor.fetchall()
•         cursor.close()
•
•         print("Listado de preguntas:")
•         for row in results:
•             print(f"Pregunta: {row[0]}, Respuesta: {row[1]}")
•     except (Exception, psycopg2.Error) as error:
•         print("Error al obtener preguntas:", error)
•
• def generate_random_question(connection):
•     try:
•         cursor = connection.cursor()
•         query = "SELECT pregunta, respuesta FROM futbol ORDER BY
random() LIMIT 1"
•         cursor.execute(query)
•         results = cursor.fetchall()
•         cursor.close()

```

```

•
•         if results:
•             return results[0]
•         else:
•             return None
•     except (Exception, psycopg2.Error) as error:
•         print("Error al generar pregunta aleatoria:", error)
•         return None
•
• def play_game(connection):
•     lives = 3
•     points = 0
•
•     while lives > 0:
•         question = generate_random_question(connection)
•         print(question[0])
•         answer = input("Tu respuesta: ")
•
•         if answer == question[1]:
•             points += 1
•             print("¡Correcto! Ganaste 1 punto.")
•         else:
•             lives -= 1
•             print("¡Incorrecto! Perdiste una vida.")
•
•         if lives == 0:
•             print("Has perdido el juego.")
•             break
•
•         print("Tu puntuación actual es:", points)
•
•     print("Has ganado", points, "puntos.")
•
• def main():
•     connection = connect_to_database()
•     if connection:
•         while True:
•             print("\nOpciones:")
•             print("1. Jugar")
•             print("2. Ver preguntas")
•             print("3. Salir")
•             opcion = input("Selecciona una opción: ")
•
•             if opcion == "1":
•                 play_game(connection)

```

```
•         elif opcion == "2":
•             show_all_questions(connection)
•         elif opcion == "3":
•             print("Saliendo del programa.")
•             break
•         else:
•             print("Opción no válida. Por favor, selecciona una
opción válida.")
•
•         connection.close()
•
• if __name__ == "__main__":
•     main()
•
```


PS D:\Desktop\Proyectos\Primer Parcial> & C:/Users/Pancho2405/AppData

Opciones:

1. Jugar
2. Ver preguntas
3. Salir

Selecciona una opción: 1

¿Quien fue el ultimo campeon del mundo?

Tu respuesta: Brasil

¡Incorrecto! Perdiste una vida.

Tu puntuación actual es: 0

¿En qué año se jugó el primer Mundial de Fútbol?

Tu respuesta: 1390

¡Incorrecto! Perdiste una vida.

Tu puntuación actual es: 0

¿Quien fue el ultimo campeon del mundo?

Tu respuesta: Alemania

¡Incorrecto! Perdiste una vida.

Has perdido el juego.

Has ganado 0 puntos.

Data Output Messages Notifications



	pregunta character varying (255)	respuesta character varying (255)
1	¿En qué año se jugó el primer Mundial de Fútbol?	1930
2	¿Cuál es el país con más títulos mundiales de fútbol?	Brasil
3	¿Quién es el máximo goleador de la historia en las eliminatorias mundialist...	Carlos Ruiz
4	¿Quién es el jugador más joven en anotar un gol en un Mundial de Fútbol?	Pelé
5	¿Quien fue el ultimo campeon del mundo?	Argentina

Serie IV

