

EVIDENCIA DE EXAMEN

Polo Avila Elias Moises

Profesor:Ebner Juárez Elías

Grupo: 1851

Tecnológico de Estudios Superiores de Ixtapaluca

ingeniería en Sistemas Computacionales

Asingnatura:Programación Logica Funcional

### **(python)**

### **Conexión a la base de datos**

python

CopiarEditar

import mysql.connector

# Conexión a la base de datos MySQL

conn = mysql.connector.connect(

host="localhost",

user="root",

password="", # Contraseña por defecto (en WAMP suele estar vacía)

database="calificaciones\_escuela"

)

cursor = conn.cursor() # Se crea un cursor para ejecutar consultas SQL

### **Diccionario de asignaturas por semestre**

python

CopiarEditar

# Diccionario que asocia cada semestre con una lista de asignaturas

asignaturas\_por\_semestre = {

1: ["CALCULO DIFERENCIAL", ..., "FUNDAMENTOS DE INVESTIGACION"],

...

9: ["INTELIGENCIA ARTIFICIAL", ..., "ALGORITMOS EVOLUTIVOS"]

}

* Permite saber qué materias le corresponden a un alumno, según su semestre.

### **Función para registrar un alumno**

python

CopiarEditar

def registrar\_alumno():

nombre = input("Ingrese el nombre del alumno: ")

semestre = int(input("Ingrese el semestre (1-9): "))

if 1 <= semestre <= 9:

cursor.execute("INSERT INTO alumno (nombre, semestre) VALUES (%s, %s)", (nombre, semestre))

conn.commit()

print(f"Alumno '{nombre}' registrado.")

else:

print("Semestre inválido.")

* Inserta un nuevo alumno en la tabla alumno, validando que el semestre esté en el rango 1-9.

### **Función para registrar calificaciones**

python

CopiarEditar

def registrar\_calificaciones():

nombre = input("Ingrese el nombre del alumno: ")

cursor.execute("SELECT id, semestre FROM alumno WHERE nombre = %s", (nombre,))

alumno = cursor.fetchone()

* Busca al alumno por nombre. Si no existe, termina el proceso.

python

CopiarEditar

semestre = alumno[1]

asignaturas = asignaturas\_por\_semestre.get(semestre, [])

* Obtiene las materias del semestre correspondiente al alumno.

python

CopiarEditar

for asignatura in asignaturas:

while True:

try:

calificacion = int(input(f"{asignatura} (0-10): "))

if 0 <= calificacion <= 10:

break

* Pide una calificación válida entre 0 y 10 para cada asignatura.

python

CopiarEditar

cursor.execute("INSERT INTO calificacion (alumno\_id, asignatura, calificacion) VALUES (%s, %s, %s)",

(alumno[0], asignatura, calificacion))

conn.commit()

print("Calificaciones registradas.")

* Inserta cada calificación en la tabla calificacion.

### **Función para consultar si un alumno es regular o irregular**

python

CopiarEditar

def consultar\_alumno():

nombre = input("Ingrese el nombre del alumno: ")

cursor.execute("SELECT id FROM alumno WHERE nombre = %s", (nombre,))

alumno = cursor.fetchone()

* Busca al alumno por nombre.

python

CopiarEditar

cursor.execute("SELECT calificacion FROM calificacion WHERE alumno\_id = %s", (alumno[0],))

calificaciones = [c[0] for c in cursor.fetchall()]

* Recupera todas las calificaciones del alumno.

python

CopiarEditar

estado = "regular" if all(c >= 7 for c in calificaciones) else "irregular"

print(f"Alumno {estado}.")

* Si todas son ≥7 → regular, si alguna es <7 → irregular.

### **Función para consultar cuántos reprobados hay**

python

CopiarEditar

def consultar\_reprobados():

semestre = input("Semestre (1-9) o 'todos': ")

* El usuario puede consultar por un semestre específico o todos.

python

CopiarEditar

if semestre.lower() == "todos":

cursor.execute("SELECT COUNT(DISTINCT alumno\_id) FROM calificacion WHERE calificacion < 7")

* Cuenta todos los alumnos que reprobaron al menos una materia.

python

CopiarEditar

else:

try:

semestre = int(semestre)

cursor.execute("""

SELECT COUNT(DISTINCT c.alumno\_id)

FROM calificacion c

INNER JOIN alumno a ON c.alumno\_id = a.id

WHERE a.semestre = %s AND c.calificacion < 7

""", (semestre,))

* Si se especificó un semestre, filtra por ese grupo.

### **Función para ver el status de desempeño del alumno**

python

CopiarEditar

def ver\_status\_alumno():

nombre = input("Ingrese el nombre del alumno: ")

cursor.execute("SELECT id FROM alumno WHERE nombre = %s", (nombre,))

alumno = cursor.fetchone()

* Busca al alumno por nombre.

python

CopiarEditar

cursor.execute("SELECT calificacion FROM calificacion WHERE alumno\_id = %s", (alumno[0],))

calificaciones = [c[0] for c in cursor.fetchall()]

* Extrae todas sus calificaciones y calcula su promedio.

python

CopiarEditar

promedio = sum(calificaciones) / len(calificaciones)

if 7 <= promedio < 8:

status = "Aceptable"

elif 8 <= promedio < 9:

status = "Bueno"

elif 9 <= promedio <= 10:

status = "Excelente"

else:

status = "Insuficiente"

print(f"Promedio: {promedio:.2f} → Desempeño: {status}")

* Evalúa su rendimiento en base al promedio numérico.

### **Menú principal del programa**

python

CopiarEditar

def menu():

while True:

print("\n📚 Menú Académico")

...

opcion = input("Seleccione una opción: ")

* Muestra el menú con opciones para cada acción.

python

CopiarEditar

if opcion == "1":

registrar\_alumno()

elif opcion == "2":

registrar\_calificaciones()

elif opcion == "3":

consultar\_alumno()

elif opcion == "4":

consultar\_reprobados()

elif opcion == "5":

ver\_status\_alumno()

elif opcion == "6":

print("Saliendo...")

break

else:

print("Opción inválida.")

* Ejecuta la función correspondiente según la opción ingresada por el usuario.

### **Cierre de la conexión**

python

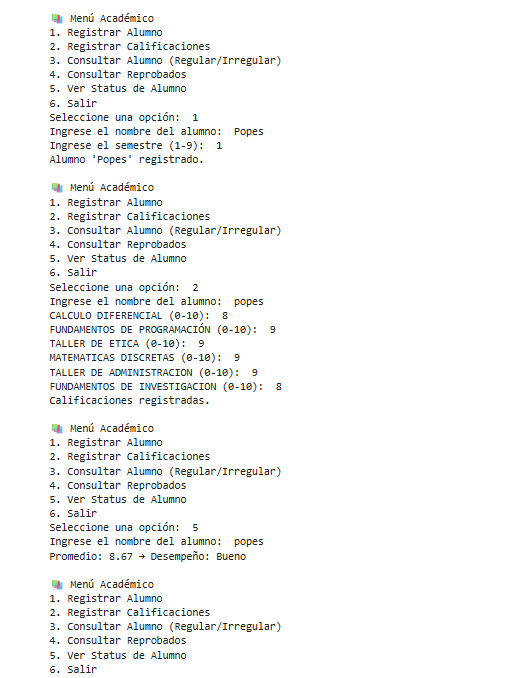
CopiarEditar

menu()

conn.close()

* Se ejecuta el menú, y al finalizar, se cierra la conexión a la base de datos correctamente.

-CONSULTAS



(prolog)

### **1. Hechos: alumnos registrados**

prolog

CopiarEditar

% Define los alumnos y su semestre correspondiente.

% Sintaxis: alumno(Nombre, Semestre).

alumno(juan, 1). % Juan está inscrito en el semestre 1

alumno(ana, 3). % Ana está inscrita en el semestre 3

alumno(carlos, 3). % Carlos está inscrito en el semestre 3

### **2. Hechos: asignaturas por semestre**

prolog

CopiarEditar

% Define las materias disponibles en cada semestre.

% Sintaxis: asignatura(Semestre, NombreAsignatura).

asignatura(1, 'CALCULO DIFERENCIAL'). % Materia del semestre 1

asignatura(1, 'FUNDAMENTOS DE PROGRAMACIÓN'). % Otra materia del semestre 1

asignatura(3, 'CALCULO VECTORIAL'). % Materia del semestre 3

asignatura(3, 'ESTRUCTURA DE DATOS'). % Otra materia del semestre 3

### **3. Hechos: calificaciones de cada alumno**

prolog

CopiarEditar

% Registra qué calificación obtuvo cada alumno en cada materia.

% Sintaxis: calificacion(NombreAlumno, Asignatura, Calificacion).

calificacion(juan, 'CALCULO DIFERENCIAL', 9).

calificacion(juan, 'FUNDAMENTOS DE PROGRAMACIÓN', 8).

calificacion(ana, 'CALCULO VECTORIAL', 6).

calificacion(ana, 'ESTRUCTURA DE DATOS', 7).

calificacion(carlos, 'CALCULO VECTORIAL', 10).

calificacion(carlos, 'ESTRUCTURA DE DATOS', 9).

### **4. Reglas: alumno regular o irregular**

prolog

CopiarEditar

% Un alumno es regular si todas sus calificaciones son mayores o iguales a 7

regular(Nombre) :-

alumno(Nombre, \_), % El alumno debe estar registrado

findall(C, calificacion(Nombre, \_, C), Calificaciones), % Obtener todas sus calificaciones

Calificaciones \= [], % Verificar que tenga calificaciones

forall(member(C, Calificaciones), C >= 7). % Todas deben ser ≥ 7

% Un alumno es irregular si al menos una de sus calificaciones es menor que 7

irregular(Nombre) :-

alumno(Nombre, \_),

findall(C, calificacion(Nombre, \_, C), Calificaciones),

Calificaciones \= [],

member(C, Calificaciones), C < 7.

### **5. Regla para calcular el promedio**

prolog

CopiarEditar

% Calcula el promedio de calificaciones de un alumno

% Sintaxis: promedio(NombreAlumno, ResultadoPromedio)

promedio(Nombre, Promedio) :-

findall(C, calificacion(Nombre, \_, C), Calificaciones), % Lista de calificaciones

Calificaciones \= [], % Verificar que existan

sumlist(Calificaciones, Suma), % Sumar todas

length(Calificaciones, Total), % Contar cuántas son

Promedio is Suma / Total. % Dividir para obtener promedio

### 🏅 **6. Clasificación del desempeño según el promedio**

prolog

CopiarEditar

% Define el estado del alumno según su promedio

desempeno(Nombre, 'Excelente') :-

promedio(Nombre, P),

P >= 9, P =< 10.

desempeno(Nombre, 'Bueno') :-

promedio(Nombre, P),

P >= 8, P < 9.

desempeno(Nombre, 'Aceptable') :-

promedio(Nombre, P),

P >= 7, P < 8.

desempeno(Nombre, 'Insuficiente') :-

promedio(Nombre, P),

P < 7.

### ❌ **7. Reprobados por semestre**

prolog

CopiarEditar

% Encuentra alumnos que tienen al menos una materia reprobada en cierto semestre

% Una calificación menor a 7 se considera reprobada

reprobado\_en\_semestre(Semestre, Nombre) :-

alumno(Nombre, Semestre), % El alumno debe estar en el semestre especificado

calificacion(Nombre, \_, C), % Buscar sus calificaciones

C < 7. % Verificar si alguna es menor a 7

### **Ejemplos de uso en consola de Prolog:**

prolog

CopiarEditar

?- regular(juan).

true.

?- regular(ana).

false.

?- promedio(carlos, P).

P = 9.5.

?- desempeno(ana, D).

D = 'Aceptable'.

?- reprobado\_en\_semestre(3, ana).

true.

Consultas

?- regular(juan).

true.

?- regular(ana).

false.

?- promedio(carlos, P).

P = 9.5.

?- desempeno(carlos, D).

D = 'Excelente'.

?- reprobado\_en\_semestre(3, ana).

true.