

# PRUEBA TÉCNICA QA

Componente practico

Wilder Adolfo Perez Amador

RappiPay

# ★ contextualizacion

Para resolver el ejercicio primero se construyó una base de datos relacional con las tablas `students`, `careers` y `career_student`, respetando claves primarias y foráneas.

Sobre esta base se ejecutaron consultas de lectura, actualización y eliminación, cuidando la integridad referencial y validando los resultados antes y después de cada operación.

de igual forma se adjunta un archivo llamado `sql-exercises.sql` el cual tiene los scripts de creación de la bd y los ejercicios aca presentados.

# ★ Seleccione el total de estudiantes Masculinos Registrados

```
SELECT COUNT(*) AS total_masculinos  
FROM students  
WHERE gender = 'M';
```

The screenshot shows a SQL IDE interface. At the top, a comment in green text reads: `--a. Seleccione el total de estudiantes Masculinos Registrados`. Below it, the SQL query is entered and highlighted in blue: `SELECT COUNT(*) AS total_masculinos  
FROM students  
WHERE gender = 'M';`. Below the query editor, another comment in green text reads: `--b. Elimine los estudiantes nacidos antes 2000 (por probar)`. Below the query editor, there is a tab labeled "Resultados 1" with a close button. Below the tab, the query is repeated: `SELECT COUNT(*) AS total_masculino`. To the right of the query, there is a text box with the placeholder "Enter a SQL expression to filter results (use Ctrl+Space)". Below the text box, there is a table with one row and one column. The row is labeled "1" and the column is labeled "total\_masculinos". The value in the cell is "123".

	total_masculinos
1	123

# ★ Elimine los estudiantes nacidos antes 2000

```
SELECT student_id, name, birthdate
FROM students
WHERE birthdate < '2000-01-01'
ORDER BY birthdate;

--2 eliminar estudiantes nacidos antes del 2000
DELETE FROM career_student
WHERE student_id IN (
SELECT student_id
FROM students
WHERE birthdate < '2000-01-01'
);
SELECT changes() AS carreras_borradas;

DELETE FROM students
WHERE birthdate < '2000-01-01';
SELECT changes() AS estudiantes_borrados;
```

SQL query editor showing the following query:

```
SELECT student_id, name, birthdate
FROM students
WHERE birthdate < '2000-01-01'
ORDER BY birthdate;
```

Results table:

student_id	name	birthdate
9.000.189	ELENA CASTRO	1972-02-18
9.000.185	PEDRO GOMEZ	1985-10-30
9.000.194	MONICA FIGUEROA	1987-05-06
9.000.190	MATIAS PEREZ	1993-11-01
9.000.186	JUAN HERNANDEZ	1995-01-25
9.000.193	CAROLINA LUNA	1997-07-21

123 carreras\_borradas

10
----

123 estudiantes\_borrados

1	8
---	---

★ Actualice la descripción de la carrera con id '10026' de “Administración” a “Administración de Empresas”

```
UPDATE careers
SET description = 'Administración de Empresas'
WHERE career_id = 10026;
```

```
UPDATE careers
SET description = 'Administración de Empresas'
WHERE career_id = 10026;
```

	Value
Rows	1
Time	0,005s
Date	Mon Feb 02 21:03:54 COT 2026
Time	Mon Feb 02 21:03:54 COT 2026
SQL	UPDATE careers SET description = 'Administración de Empresas' WHERE career_id = 10026

# ★ Muestre los nombres de los estudiantes que estén cursando dos o más materias

```
SELECT s.name
FROM students s
JOIN career_student cs ON cs.student_id =
s.student_id
GROUP BY s.student_id, s.name
HAVING COUNT(*) >= 2;
```

--d. Muestre los nombres de los estudiantes que estén cursando

```
SELECT s.name
FROM students s
JOIN career_student cs ON cs.student_id = s.student_id
GROUP BY s.student_id, s.name
HAVING COUNT(*) >= 2;
```

ents 1 X

ATE careers SET description = 'Ar' Enter a SQL expression to filter results (use Ctrl+Sp)

<input checked="" type="radio"/> A-Z name ▼	
PEDRO GOMEZ	
JUAN HERNANDE	
MATIAS PEREZ	
SEBASTIAN VARG	

# ★ Calificación promedio de todos los estudiantes que estén cursando carreras

```
SELECT AVG(cs.calification) AS  
promedio_calificacion  
FROM career_student cs;
```

The screenshot shows a SQL IDE with a query editor at the top and a results pane at the bottom. The query in the editor is: `--e. Calificación promedio de todos los es  
SELECT AVG(cs.calification) AS promedio_co  
FROM career_student cs;`. The results pane shows a single row with the column name `123 promedio_calificacion` and the value `76,4615384615`.

123 promedio_calificacion
76,4615384615

# ★ Muestra el nombre, carrera id, correos de los estudiantes que tienen una calificación mayor 70

```
SELECT s.name,  
       cs.career_id,  
       s.email,  
       cs.calification  
FROM students s  
JOIN career_student cs ON cs.student_id =  
s.student_id  
WHERE cs.calification > 70;
```

```
--f. Muestra el nombre, carrera id, correos de los estudiantes que tienen un  
SELECT s.name,  
       cs.career_id,  
       s.email,  
       cs.calification  
FROM students s  
JOIN career_student cs ON cs.student_id = s.student_id  
WHERE cs.calification > 70;
```

	A-Z name	123 career_id	A-Z email	123 calification
	PEDRO GOMEZ	10.025	PEDRO.GOMEZ@GMAIL.COM	
	JUAN HERNANDEZ	10.030	Juan.Hernandez@gmail.com	
	SEBASTIAN VARGAS	10.028	SEBASTIAN.VARGAS@OUTLOC	
	ELENA CASTRO	10.025	elena.casto@hotmail.com	
	SEBASTIAN VARGAS	10.026	SEBASTIAN.VARGAS@OUTLOC	



# ★ Cuántos estudiantes que estén cursando 2 o más carreras

```
SELECT COUNT(*) AS  
estudiantes_con_2_o_mas_carreras  
FROM (  
SELECT cs.student_id  
FROM career_student cs  
GROUP BY cs.student_id  
HAVING COUNT(*) >= 2  
) t;
```

```
--g. Cuántos estudiantes que estén cursando 2 o más carreras  
SELECT COUNT(*) AS estudiantes_con_2_o_mas_carreras  
FROM (  
  SELECT cs.student_id  
  FROM career_student cs  
  GROUP BY cs.student_id  
  HAVING COUNT(*) >= 2  
) t;
```

Resultados 1 ✕

SELECT s.name, cs.career\_id, s.email, c | Enter a SQL expression to filter results (use Ctrl+Spa

123	estudiantes_con_2_o_mas_carreras	
		4