

Integrantes:

Juan Alejandro Rodriguez Ribon

Brayan Steven Rodriguez Alba

Taller Procedimientos almacenados

- Punto 1
 - Código

```
1 delimiter //
```

```
2 • CREATE PROCEDURE holaMundo()
```

```
3 • BEGIN
```

```
4   select "¡Hola Mundo!";
```

```
5 END//
```

```
6 • call holaMundo()
```

- Resultado de la ejecución del código

	¡Hola Mundo!
▶	¡Hola Mundo!

- Punto 2:
 - Código

```

1 delimiter //
2 • CREATE PROCEDURE calificaciones(in _nombre varchar(45), in _nota int)
3 BEGIN
4     declare respuesta varchar(45);
5     IF _nota >= 0 AND _nota < 5 THEN
6         set respuesta = "Insuficiente";
7     ELSEIF _nota >= 5 AND _nota < 6 THEN
8         set respuesta = "Aprobado";
9     ELSEIF _nota >= 6 AND _nota < 7 THEN
10        set respuesta = "Bien";
11    ELSEIF _nota >= 7 AND _nota < 9 THEN
12        set respuesta = "Notable";
13    ELSEIF _nota >= 9 AND _nota <= 10 THEN
14        set respuesta = "Sobresaliente";
15    ELSE
16        set respuesta = "Nota no válida";
17    END IF;
18    insert into punto_2(nombre, nota, rango) values(_nombre, _nota, respuesta);
19 END//
20 • call calificaciones("Brayan", 5);
21 call calificaciones("Steven", 11);
22 call calificaciones("Angie", 9);
23 select * from punto_2

```

- Resultado de la ejecución del código

	id	nombre	nota	rango
▶	1	Brayan	5	Aprobado
	2	Steven	11	Nota no válida
	3	Angie	9	Sobresaliente

- Punto 3

- Código

```

1 delimiter //
2 • CREATE PROCEDURE cantidadProductos(in _buscar varchar(45))
3 BEGIN
4     select t.nombre as "tipo productos", count(t.nombre) as "cantidad de productos"
5     from punto_3 p, punto_3_tipoproductos t
6     where t.nombre=_buscar
7     and p.id_tipo_producto = t.id
8     group by t.nombre;
9 end//
10 • call cantidadProductos("granos");

```

- Resultado de la ejecución del código

	tipo productos	cantidad de productos
▶	granos	3

- Punto 4

- Código

```

1  delimiter //
2  • CREATE PROCEDURE preciosProductos(in _buscar varchar(45))
3  BEGIN
4      select max(precio) as "Precio maximo", min(precio) as "precio minimo", round(avg(precio)) as "promedio de los precios"
5      from punto_3 p, punto_3_tipoproductos t
6      where t.nombre=_buscar
7      and p.id_tipo_producto = t.id;
8  end//
9  • call preciosProductos("lacteos")

```

- Resultado de la ejecución del código

	Precio maximo	precio minimo	promedio de los precios
▶	3500	2700	3067

- Punto 5

- Código

```

1  delimiter //
2  • CREATE PROCEDURE funcionIVA()
3  BEGIN
4      select precio, round(precio*0.19) iva, round(precio+(precio*0.19)) total
5      from punto_3;
6  END//
7  • call funcionIVA()

```

- Resultado de la ejecución del código

	precio	iva	total
▶	2000	380	2380
	1600	304	1904
	2600	494	3094
	2700	513	3213
	3000	570	3570
	3500	665	4165
	5000	950	5950
	2600	494	3094
	4000	760	4760

- Punto 6

- Código

```

1  delimiter //
2  ● CREATE PROCEDURE sucursales(in _buscar varchar(45))
3  ○ BEGIN
4      select p.nombre pais, s.sucursal
5      from punto_6_sucursal s,punto_6_pais p
6      where s.id_pais=p.id
7      and _buscar=p.nombre;
8  end//
9  ● call sucursales("argentina")

```

- Resultado de la ejecución del código

	pais	sucursal
▶	argentina	buenos aires
	argentina	rosario

- Punto 7

- Código

```

1 • CREATE TABLE punto_7_edad (
2   id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
3   nombre VARCHAR (45),
4   fecha_nacimiento DATE
5 );
6 • insert into punto_7_edad(nombre, fecha_nacimiento) values ("Brayan", "2002-06-08");
7 • insert into punto_7_edad(nombre, fecha_nacimiento) values ("angie", "1999-06-03");
8 • insert into punto_7_edad(nombre, fecha_nacimiento) values ("nataly", "2013-02-26");
9 • insert into punto_7_edad(nombre, fecha_nacimiento) values ("jonathan", "2011-05-10");
10
11 • ALTER TABLE punto_7_edad ADD COLUMN edad INT;
12 • select * from punto_7_edad

```

- Resultado de la ejecución del código

	id	nombre	fecha_nacimiento	edad
▶	1	brayan	2002-06-08	NULL
	2	angie	1999-06-03	NULL
	3	nataly	2013-02-26	NULL
	4	jonathan	2011-05-10	NULL
✱	NULL	NULL	NULL	NULL

- Punto 8

- Código

```

1   DELIMITER //
2   CREATE FUNCTION calcularEdad(fecha DATE)
3   RETURNS INT
4   DETERMINISTIC
5   BEGIN
6     DECLARE edad INT;
7     SELECT timestampdiff(year, fecha, now()) INTO edad;
8     RETURN edad;
9   END//
10  DELIMITER ;
11  select calcularEdad("2002-06-08") edad

```

- Resultado de la ejecución del código

Result Grid	
	edad
▶	20

- Punto 9

- Código

```

1  delimiter //
2  • CREATE PROCEDURE actualizarColumnaEdad()
3  BEGIN
4      select nombre, fecha_nacimiento, calcularEdad(fecha_nacimiento) edad
5      from punto_7_edad;
6  end//
7  • call actualizarColumnaEdad()

```

- Resultado de la ejecución del código

	nombre	fecha_nacimiento	edad
▶	brayan	2002-06-08	20
	angie	1999-06-03	23
	nataly	2013-02-26	10
	jonathan	2011-05-10	11

- Punto 10

- Código

```

1  delimiter //
2  • CREATE PROCEDURE Rover(in _distancia float, in _tiempo float)
3  BEGIN
4      select concat((_distancia/_tiempo),"","m/s") as "velocidad medida en segundos";
5  end//
6  • call Rover(100,300)|

```

- Resultado de la ejecución del código

	velocidad medida en segundos
▶	0.333333333333333m/s