

# JAVA M3 – ABPro5 - Ejercicio Grupal.

## Modelo entidad relación.

19-04-2022

Integrante: Claudio Vidal A.

Grupo 2

---

Parte 1: Crear entorno de trabajo.

- a) Crear una base de datos.
- b) Crear un usuario con todos los privilegios para trabajar con la base de datos recién creada.

Parte 2: Crear dos tablas.

- c) La primera almacena a los usuarios de la aplicación (id\_usuario, nombre, apellido, contraseña, zona horaria (por defecto UTC-3), género y teléfono de contacto).
- d) La segunda tabla almacena información relacionada a la fecha-hora de ingreso de los usuarios a la plataforma (id\_ingreso, id\_usuario y la fecha-hora de ingreso (por defecto la fecha-hora actual)).

Parte 3: Modificación de la tabla.

- e) Modifique el UTC por defecto. Desde UTC-3 a UTC-2.

Parte 4: Creación de registros.

- f) Para cada tabla crea 8 registros.

Parte 5:

- g) Justifique cada tipo de dato utilizado. ¿Es el óptimo en cada caso?

Parte 6:

- h) Creen una nueva tabla llamada Contactos (id\_contacto, id\_usuario, numero de teléfono, correo electrónico).

Parte 7:

- i) Modifique la columna teléfono de contacto, para crear un vínculo entre la tabla Usuarios y la tabla Contactos.
- j) El ejercicio debe ser subido a github y al Nodo Virtual.

## DESARROLLO:

### Parte 1: Crear entorno de trabajo.

#### a) Crear una base de datos.

The screenshot shows a database management interface with a Navigator pane on the left and a SQL editor on the right. The Navigator pane displays a tree of schemas, with 'abpro5' selected and highlighted by a red rectangle. The SQL editor contains the following code:

```
35 -- Parte 1: Crear entorno de trabajo.  
36 -- a) Crear una base de datos.  
37 • CREATE SCHEMA `abpro5` DEFAULT CHARACTER SET utf8;  
38 • USE abpro5;  
39
```

The Output pane at the bottom shows the execution results of the SQL statements:

#	Time	Action
1	21:06:41	CREATE SCHEMA `abpro5` DEFAULT CHARACTER SET utf8
2	21:06:41	USE abpro5

b) Crear un usuario con todos los privilegios para trabajar con la base de datos recién creada.

The screenshot displays the MySQL Workbench interface. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The 'Navigator' pane on the left shows the 'SCHEMAS' section with a search filter 'Filter objects'. Under the 'abpro5' database, there are icons for Tables, Views, Stored Procedures, and Functions. The 'Administration' tab is active, showing 'Schemas' and 'Information'. The 'Connection Details' section lists: Name: abpro5\_Session, Host: 127.0.0.1, Port: 3306, Login User: abpro5user (highlighted with a red box), Current User: abpro5user@localhost, and SSL cipher: TLS\_AES\_256\_GCM\_SHA384. The 'Server' section shows Product: MySQL Community Server - GPL and Version: 8.0.28. The 'Connector' section shows Version: C++ 8.0.28. The main editor window shows a SQL script with comments in Spanish and two SQL statements: `CREATE USER 'abpro5user'@'localhost' IDENTIFIED BY 'abpro5';` and `GRANT ALL PRIVILEGES ON abpro5 TO 'abpro5user'@'localhost';`. The 'Output' pane at the bottom shows 'Action Output' with columns for #, Time, Action, and Message.

```
40  /* b)  Crear un usuario con todos los privilegios para trabajar con la base
41  de datos recién creada. */
42  • CREATE USER 'abpro5user'@'localhost' IDENTIFIED BY 'abpro5';
43  • GRANT ALL PRIVILEGES ON abpro5 TO 'abpro5user'@'localhost';
44
45
46  -- Parte 2: Crear dos tablas.
47  /* c)  La primera almacena a los usuarios de la aplicación (id_usuario,
48  nombre, apellido, contraseña, zona horaria (por defecto UTC-3), género
49  y teléfono de contacto). */
50  /* d)  La segunda tabla almacena información relacionada a la fecha-hora
51  de ingreso de los usuarios a la plataforma (id_ingreso, id_usuario y la
52  fecha-hora de ingreso (por defecto la fecha-hora actual)). */
```

Connection Details

- Name: abpro5\_Session
- Host: 127.0.0.1
- Port: 3306
- Login User: abpro5user
- Current User: abpro5user@localhost
- SSL cipher: TLS\_AES\_256\_GCM\_SHA384

Server

- Product: MySQL Community Server - GPL
- Version: 8.0.28

Connector

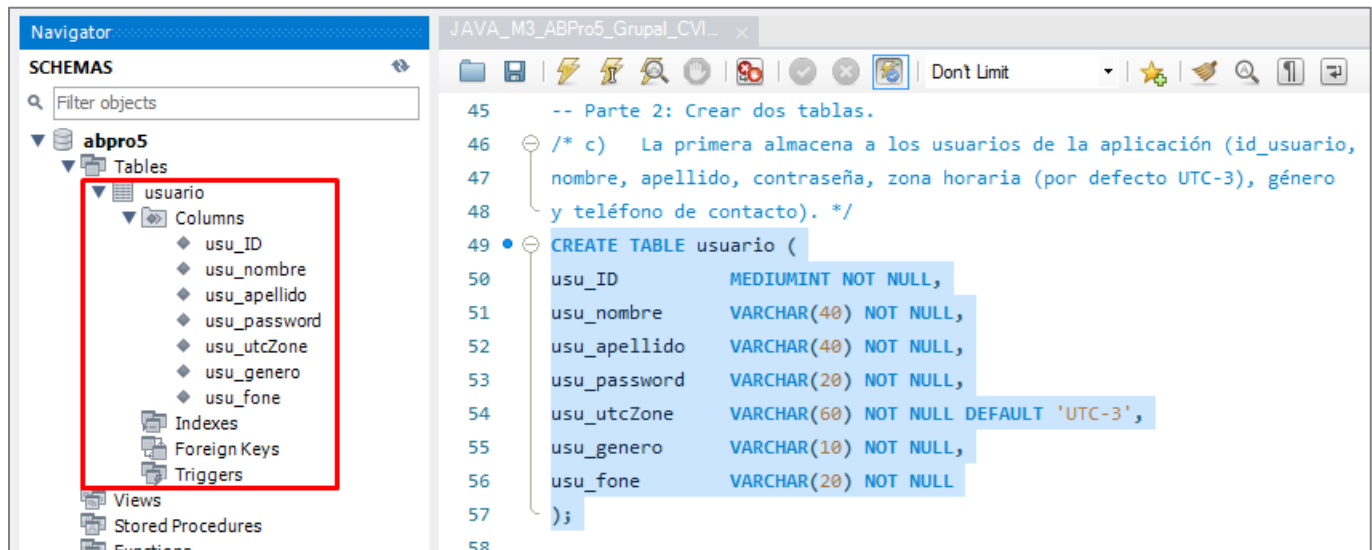
- Version: C++ 8.0.28

Output

#	Time	Action	Message
---	------	--------	---------

## Parte 2: Crear dos tablas.

- c) La primera almacena a los usuarios de la aplicación (id\_usuario, nombre, apellido, contraseña, zona horaria (por defecto UTC-3), género y teléfono de contacto).



The screenshot displays a database IDE with two main panes. The left pane, titled 'Navigator', shows the 'abpro5' schema with a tree view. Under 'Tables', the 'usuario' table is highlighted with a red box. The right pane, titled 'JAVA\_M3\_ABPro5\_Grupal\_CVI...', shows the SQL code for creating the 'usuario' table. The code is as follows:

```
45  -- Parte 2: Crear dos tablas.
46  /* c) La primera almacena a los usuarios de la aplicación (id_usuario,
47  nombre, apellido, contraseña, zona horaria (por defecto UTC-3), género
48  y teléfono de contacto). */
49  CREATE TABLE usuario (
50  usu_ID          MEDIUMINT NOT NULL,
51  usu_nombre      VARCHAR(40) NOT NULL,
52  usu_apellido    VARCHAR(40) NOT NULL,
53  usu_password    VARCHAR(20) NOT NULL,
54  usu_utcZone     VARCHAR(60) NOT NULL DEFAULT 'UTC-3',
55  usu_genero      VARCHAR(10) NOT NULL,
56  usu_fone        VARCHAR(20) NOT NULL
57  );
58
```

- d) La segunda tabla almacena información relacionada a la fecha-hora de ingreso de los usuarios a la plataforma (id\_ingreso, id\_usuario y la fecha-hora de ingreso (por defecto la fecha-hora actual)).

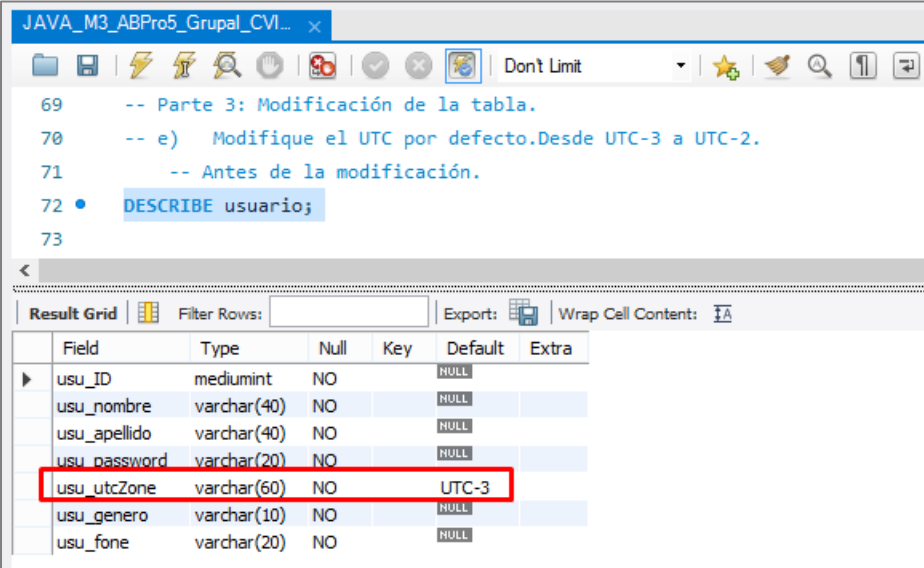
The screenshot displays a database IDE interface. On the left, the 'Navigator' pane shows the 'abpro5' schema with a 'Tables' folder expanded. The 'userlogin' table is highlighted with a red rectangle, showing its columns: 'userlog\_ID', 'usu\_ID', 'userlog\_dateTime', and 'userlog\_timesTamp'. On the right, the 'SQL Editor' pane shows the SQL code for creating the 'userlogin' table. The code is as follows:

```
59  /* d) La segunda tabla almacena información relacionada a la fecha-hora
60  de ingreso de los usuarios a la plataforma (id_ingreso, id_usuario y la
61  fecha-hora de ingreso (por defecto la fecha-hora actual)). */
62  CREATE TABLE userlogin (
63      userlog_ID          MEDIUMINT NOT NULL,
64      usu_ID              MEDIUMINT NOT NULL,
65      userlog_dateTime    DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
66      userlog_timesTamp   TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
67  );
68
```

### Parte 3: Modificación de la tabla.

e) Modifique el UTC por defecto. Desde UTC-3 a UTC-2.

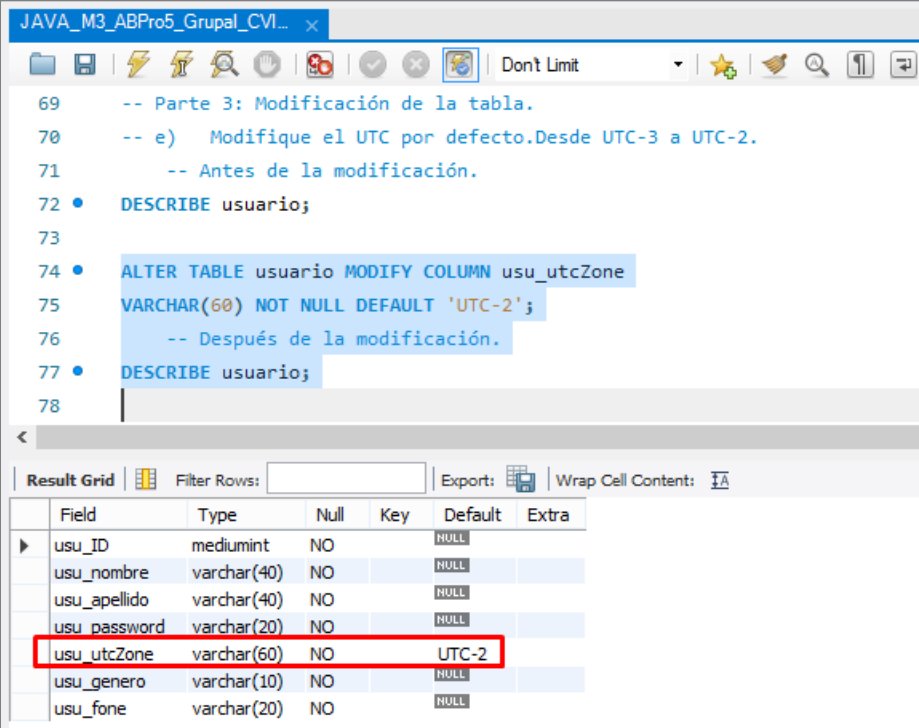
Antes de la modificación.



```
69 -- Parte 3: Modificación de la tabla.
70 -- e) Modifique el UTC por defecto.Desde UTC-3 a UTC-2.
71 -- Antes de la modificación.
72 • DESCRIBE usuario;
73
```

Field	Type	Null	Key	Default	Extra
usu_ID	mediumint	NO		NULL	
usu_nombre	varchar(40)	NO		NULL	
usu_apellido	varchar(40)	NO		NULL	
usu_password	varchar(20)	NO		NULL	
usu_utcZone	varchar(60)	NO		UTC-3	
usu_genero	varchar(10)	NO		NULL	
usu_fone	varchar(20)	NO		NULL	

Después de la modificación.



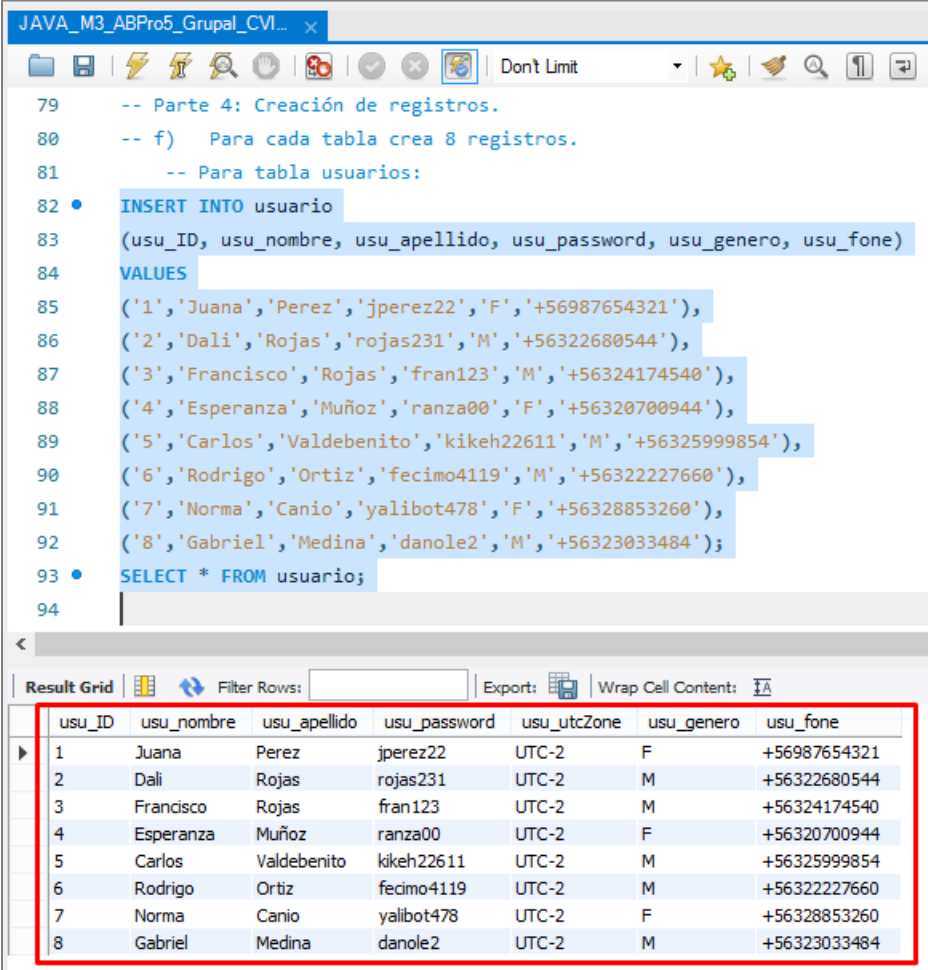
```
69 -- Parte 3: Modificación de la tabla.
70 -- e) Modifique el UTC por defecto.Desde UTC-3 a UTC-2.
71 -- Antes de la modificación.
72 • DESCRIBE usuario;
73
74 • ALTER TABLE usuario MODIFY COLUMN usu_utcZone
75 VARCHAR(60) NOT NULL DEFAULT 'UTC-2';
76 -- Después de la modificación.
77 • DESCRIBE usuario;
78
```

Field	Type	Null	Key	Default	Extra
usu_ID	mediumint	NO		NULL	
usu_nombre	varchar(40)	NO		NULL	
usu_apellido	varchar(40)	NO		NULL	
usu_password	varchar(20)	NO		NULL	
usu_utcZone	varchar(60)	NO		UTC-2	
usu_genero	varchar(10)	NO		NULL	
usu_fone	varchar(20)	NO		NULL	

#### Parte 4: Creación de registros.

f) Para cada tabla crea 8 registros.

Para tabla usuarios:



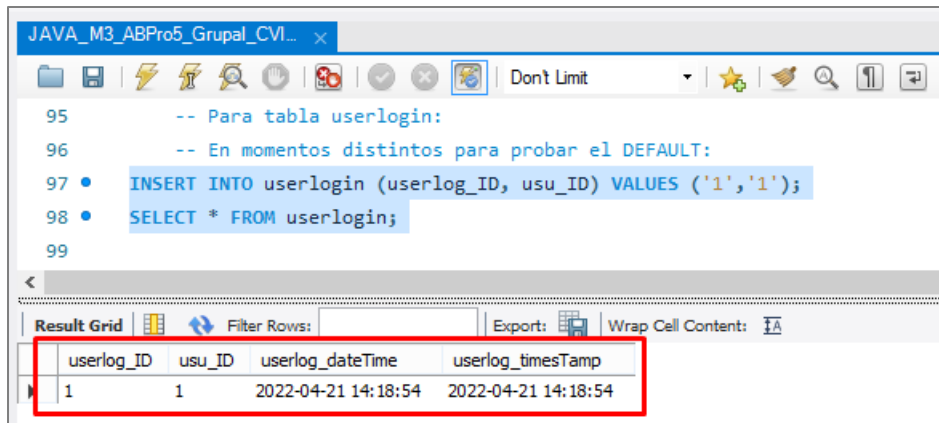
The screenshot shows a Java IDE window titled "JAVA\_M3\_ABPro5\_Grupal\_CVI...". The code editor contains SQL statements for creating and querying a table named "usuario". The code is as follows:

```
79 -- Parte 4: Creación de registros.
80 -- f) Para cada tabla crea 8 registros.
81 -- Para tabla usuarios:
82 • INSERT INTO usuario
83 (usu_ID, usu_nombre, usu_apellido, usu_password, usu_genero, usu_fone)
84 VALUES
85 ('1','Juana','Perez','jperez22','F','+56987654321'),
86 ('2','Dali','Rojas','rojas231','M','+56322680544'),
87 ('3','Francisco','Rojas','fran123','M','+56324174540'),
88 ('4','Esperanza','Muñoz','ranza00','F','+56320700944'),
89 ('5','Carlos','Valdebenito','kikeh22611','M','+56325999854'),
90 ('6','Rodrigo','Ortiz','fecimo4119','M','+5632227660'),
91 ('7','Norma','Canio','yalibot478','F','+56328853260'),
92 ('8','Gabriel','Medina','danole2','M','+56323033484');
93 • SELECT * FROM usuario;
94
```

Below the code editor, the "Result Grid" shows the output of the SELECT statement. The table has 8 rows and 7 columns: usu\_ID, usu\_nombre, usu\_apellido, usu\_password, usu\_utcZone, usu\_genero, and usu\_fone. The data is as follows:

	usu_ID	usu_nombre	usu_apellido	usu_password	usu_utcZone	usu_genero	usu_fone
▶	1	Juana	Perez	jperez22	UTC-2	F	+56987654321
	2	Dali	Rojas	rojas231	UTC-2	M	+56322680544
	3	Francisco	Rojas	fran123	UTC-2	M	+56324174540
	4	Esperanza	Muñoz	ranza00	UTC-2	F	+56320700944
	5	Carlos	Valdebenito	kikeh22611	UTC-2	M	+56325999854
	6	Rodrigo	Ortiz	fecimo4119	UTC-2	M	+5632227660
	7	Norma	Canio	yalibot478	UTC-2	F	+56328853260
	8	Gabriel	Medina	danole2	UTC-2	M	+56323033484

Para tabla userlogin y en momentos distintos, para probar el DEFAULT:

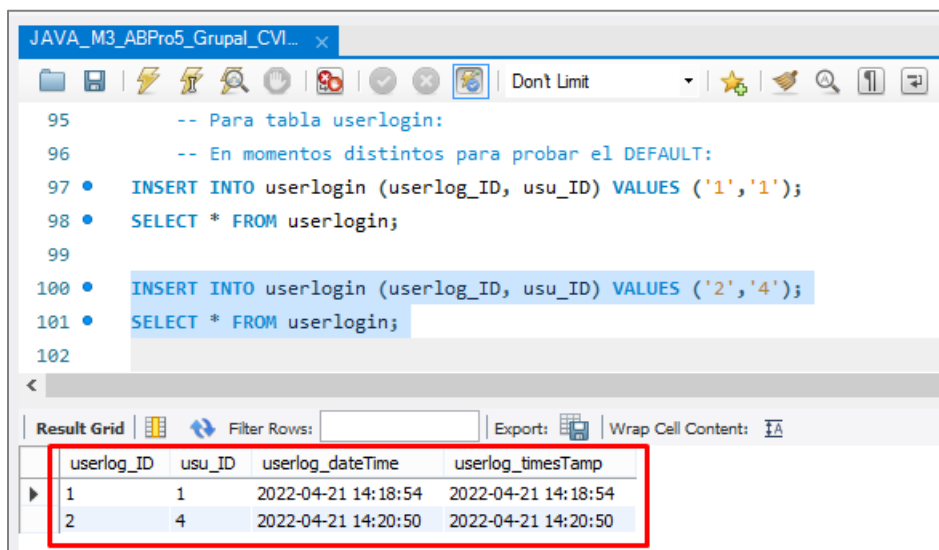


The screenshot shows a SQL IDE window titled "JAVA\_M3\_ABPro5\_Grupal\_CVI...". The SQL editor contains the following code:

```
95      -- Para tabla userlogin:
96      -- En momentos distintos para probar el DEFAULT:
97 •   INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('1','1');
98 •   SELECT * FROM userlogin;
99
```

Below the editor is a "Result Grid" showing the results of the SELECT statement. The grid has four columns: userlog\_ID, usu\_ID, userlog\_dateTime, and userlog\_timesTamp. The first row contains the values 1, 1, 2022-04-21 14:18:54, and 2022-04-21 14:18:54. The entire result grid is highlighted with a red rectangle.

userlog_ID	usu_ID	userlog_dateTime	userlog_timesTamp
1	1	2022-04-21 14:18:54	2022-04-21 14:18:54



The screenshot shows the same SQL IDE window. The SQL editor now contains the following code:

```
95      -- Para tabla userlogin:
96      -- En momentos distintos para probar el DEFAULT:
97 •   INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('1','1');
98 •   SELECT * FROM userlogin;
99
100 •  INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('2','4');
101 •  SELECT * FROM userlogin;
102
```

The "Result Grid" now shows two rows of data. The first row is the same as before, and the second row contains the values 2, 4, 2022-04-21 14:20:50, and 2022-04-21 14:20:50. The entire result grid is highlighted with a red rectangle.

userlog_ID	usu_ID	userlog_dateTime	userlog_timesTamp
1	1	2022-04-21 14:18:54	2022-04-21 14:18:54
2	4	2022-04-21 14:20:50	2022-04-21 14:20:50



JAVA\_M3\_ABPro5\_Grupal\_CVI...

Don't Limit

```
95 -- Para tabla userlogin:
96 -- En momentos distintos para probar el DEFAULT:
97 • INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('1','1');
98 • SELECT * FROM userlogin;
99
100 • INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('2','4');
101 • SELECT * FROM userlogin;
102
103 • INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('3','7');
104 • SELECT * FROM userlogin;
105
```

Result Grid

	userlog_ID	usu_ID	userlog_dateTime	userlog_timesTamp
▶	1	1	2022-04-21 14:18:54	2022-04-21 14:18:54
	2	4	2022-04-21 14:20:50	2022-04-21 14:20:50
	3	7	2022-04-21 14:22:00	2022-04-21 14:22:00

JAVA\_M3\_ABPro5\_Grupal\_CVI...

Don't Limit

```

95      -- Para tabla userlogin:
96      -- En momentos distintos para probar el DEFAULT:
97      • INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('1','1');
98      • SELECT * FROM userlogin;
99
100     • INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('2','4');
101     • SELECT * FROM userlogin;
102
103     • INSERT INTO userlogin (userlog_ID, usu_ID) VALUES ('3','7');
104     • SELECT * FROM userlogin;
105
106     -- INSERT normal:
107     • INSERT INTO userlogin
108       (userlog_ID, usu_ID, userlog_dateTime, userlog_timesTamp)
109     VALUES
110       ('4','2','2022-04-18 13:06:45','2022-04-18 13:06:45'),
111       ('5','3','2022-04-17 10:05:41','2022-04-17 10:05:41'),
112       ('6','5','2022-04-15 16:59:52','2022-04-15 16:59:52'),
113       ('7','6','2022-04-02 18:47:36','2022-04-02 18:47:36'),
114       ('8','8','2022-04-01 23:05:00','2022-04-18 23:05:00');
115     • SELECT * FROM userlogin;
116

```

Result Grid

Filter Rows:  Export: Wrap Cell Content:

	userlog_ID	usu_ID	userlog_dateTime	userlog_timesTamp
▶	1	1	2022-04-21 14:18:54	2022-04-21 14:18:54
	2	4	2022-04-21 14:20:50	2022-04-21 14:20:50
	3	7	2022-04-21 14:22:00	2022-04-21 14:22:00
	4	2	2022-04-18 13:06:45	2022-04-18 13:06:45
	5	3	2022-04-17 10:05:41	2022-04-17 10:05:41
	6	5	2022-04-15 16:59:52	2022-04-15 16:59:52
	7	6	2022-04-02 18:47:36	2022-04-02 18:47:36
	8	8	2022-04-01 23:05:00	2022-04-18 23:05:00

## Parte 5:

### g) Justifique cada tipo de dato utilizado. ¿Es el óptimo en cada caso?

Para el caso de las variables usu\_nombre, usu\_apellido, usu\_password, usu\_utcZone, usu\_genero y usu\_fone, se usó varchar con extensión de string variable, lo que permite ajustar a rangos razonables el espacio ocupado en memoria.

Para el caso de la variable usu\_ID se usó el tipo mediumint considerando un caso normal de uso (con signo el rango de valores va desde -8.388.608 a 8.388.607, sin signo el rango va desde 0 A16777215).

```
120 • DESCRIBE usuario;
121 • DESCRIBE userlogin;
122
```

	Field	Type	Null	Key	Default	Extra
▶	usu_ID	mediumint	NO		NULL	
	usu_nombre	varchar(40)	NO		NULL	
	usu_apellido	varchar(40)	NO		NULL	
	usu_password	varchar(20)	NO		NULL	
	usu_utcZone	varchar(60)	NO		UTC-2	
	usu_genero	varchar(10)	NO		NULL	
	usu_fone	varchar(20)	NO		NULL	

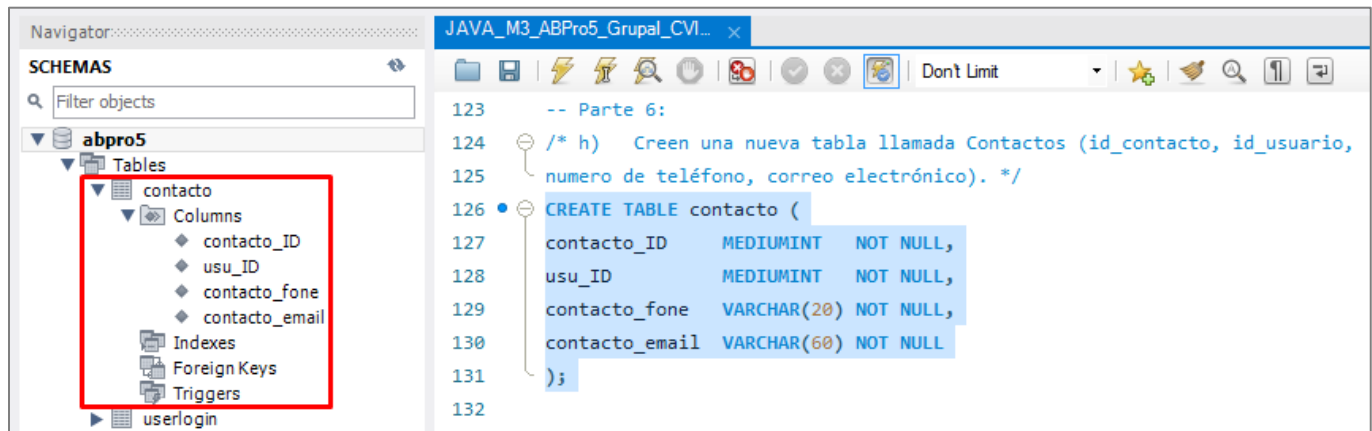
Para el caso de las variables userlog\_dateTime y userlog\_timesTamp, se usaron del tipo DateTime y Timestamp respectivamente, de modo de almacenar fecha y hora acorde al estándar. Se agregaron de los dos tipos a la tabla, con la finalidad de hacer pruebas con cambios de zonas horarias. Para las variables userlog\_ID y usu\_ID, se usaron del tipo médiumint, lo que permitirá dar una holgura razonable y optimización en uso de memoria.

```
121 • DESCRIBE userlogin;
122
```

	Field	Type	Null	Key	Default	Extra
▶	userlog_ID	mediumint	NO		NULL	
	usu_ID	mediumint	NO		NULL	
	userlog_dateTime	datetime	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED
	userlog_timesTamp	timestamp	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED

## Parte 6:

- h) Creen una nueva tabla llamada Contactos (id\_contacto, id\_usuario, numero de teléfono, correo electrónico).



Navigator: SCHEMAS

Filter objects

abpro5

Tables

contacto

Columns

- contacto\_ID
- usu\_ID
- contacto\_fone
- contacto\_email

Indexes

Foreign Keys

Triggers

userlogin

SQL Editor: JAVA\_M3\_ABPro5\_Grupal\_CVI...

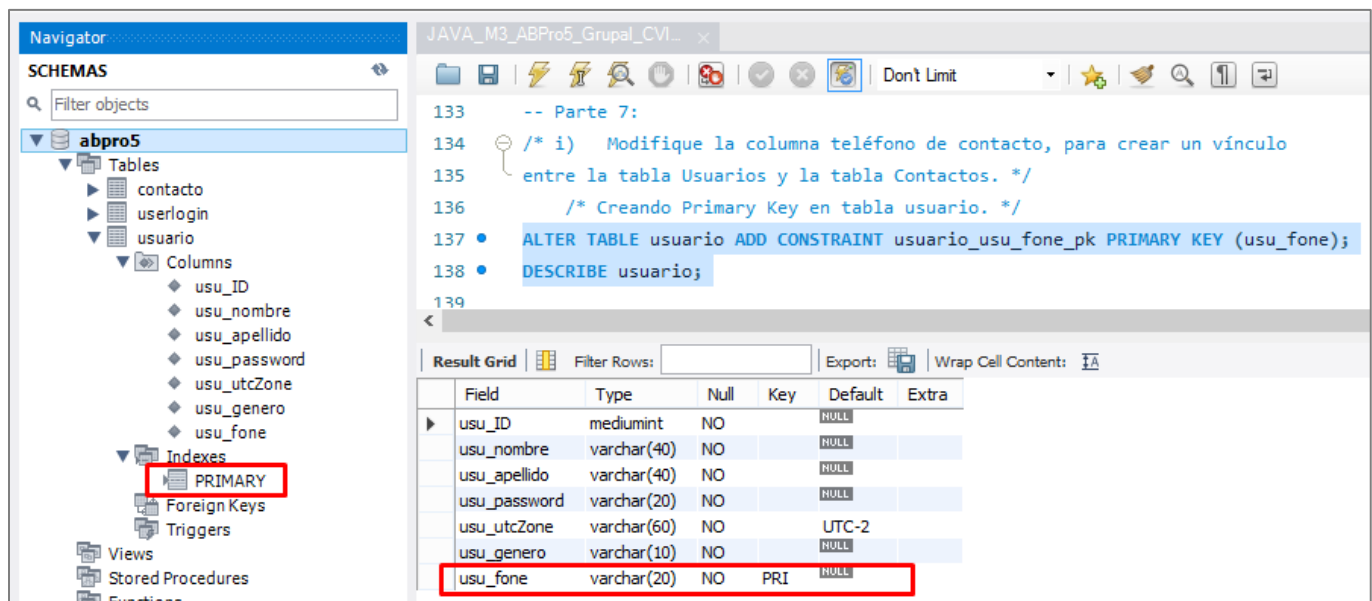
```

123 -- Parte 6:
124 /* h) Creen una nueva tabla llamada Contactos (id_contacto, id_usuario,
125 numero de teléfono, correo electrónico). */
126 CREATE TABLE contacto (
127 contacto_ID MEDIUMINT NOT NULL,
128 usu_ID MEDIUMINT NOT NULL,
129 contacto_fone VARCHAR(20) NOT NULL,
130 contacto_email VARCHAR(60) NOT NULL
131 );
132
  
```

## Parte 7:

- i) Modifique la columna teléfono de contacto, para crear un vínculo entre la tabla Usuarios y la tabla Contactos.

Creando Primary Key en tabla usuario.



Navigator: SCHEMAS

Filter objects

abpro5

Tables

contacto

userlogin

usuario

Columns

- usu\_ID
- usu\_nombre
- usu\_apellido
- usu\_password
- usu\_utcZone
- usu\_genero
- usu\_fone

Indexes

PRIMARY

Foreign Keys

Triggers

Views

Stored Procedures

Functions

SQL Editor: JAVA\_M3\_ABPro5\_Grupal\_CVI...

```

133 -- Parte 7:
134 /* i) Modifique la columna teléfono de contacto, para crear un vínculo
135 entre la tabla Usuarios y la tabla Contactos. */
136 /* Creando Primary Key en tabla usuario. */
137 ALTER TABLE usuario ADD CONSTRAINT usuario_usu_fone_pk PRIMARY KEY (usu_fone);
138 DESCRIBE usuario;
139
  
```

Result Grid

Field	Type	Null	Key	Default	Extra
usu_ID	mediumint	NO		NULL	
usu_nombre	varchar(40)	NO		NULL	
usu_apellido	varchar(40)	NO		NULL	
usu_password	varchar(20)	NO		NULL	
usu_utcZone	varchar(60)	NO		UTC-2	
usu_genero	varchar(10)	NO		NULL	
usu_fone	varchar(20)	NO	PRI	NULL	

Creando Foreign Key en tabla contacto.

```
135  entre la tabla Usuarios y la tabla Contactos. */
136  -- Creando Primary Key en tabla usuario.
137  • ALTER TABLE usuario ADD CONSTRAINT usuario_usu_fone_pk PRIMARY KEY (usu_fone);
138  • DESCRIBE usuario;
139
140  -- Creando Foreign Key en tabla contacto.
141  • ALTER TABLE contacto ADD CONSTRAINT contacto_contacto_fone_fk FOREIGN KEY (contacto_fone)
142  REFERENCES usuario(usu_fone);
143  • DESCRIBE contacto;
144
145
```

Field	Type	Null	Key	Default	Extra
contacto_ID	mediumint	NO		NULL	
usu_ID	mediumint	NO		NULL	
contacto_fone	varchar(20)	NO	MUL	NULL	
contacto_email	varchar(60)	NO		NULL	

j) El ejercicio debe ser subido a github y al Nodo Virtual.