

Tarea semana 7: Bases de datos con Python y MySql

Jorge Peñaloza

Programación Avanzada II

Instituto IACC

4 de octubre de 2021

## **Desarrollo**

### **Respuestas del problema**

#### ***Contexto***

Lea atentamente el siguiente caso, analice la información de acuerdo a los contenidos revisados en la semana y conteste las preguntas que se presentan a continuación:

Recuerde que usted fue seleccionado para diseñar lógico de un sistema para el registro de estudiantes de colegios privados de la Región Metropolitana de Santiago, este sistema debe manejar las personas dentro del proceso como son: estudiantes con todos sus atributos (nombre, apellido, Rut, dirección) y apoderados también con sus atributos (nombre, apellido, Rut, dirección), que año y asignaturas cursan cada uno de los estudiantes, si realizan actividades extras, especificando cual actividad realizan.

Ahora bien, debe realizar lo siguiente en el gestor de bases de datos MySQL y el lenguaje de programación Python:

1. Realice y explique el acceso al servidor de la base de datos requerida (3 puntos).
2. Aplique las operaciones de las bases de datos (crear, ingresar y eliminar) para interactuar con el lenguaje Python (3 puntos).
3. Realice consultas, inserción y eliminación de datos en MySQL (3 puntos).

## ***Etapas 1***

Para el problema planteado se procede a realizar como primera etapa que es crear el modelo de base de datos y realizar un programa en Python que cree la base de datos y tablas, para lo cual se establece que los datos de la conexión serán:

- Servidor: 127.0.0.1
- Usuario: root
- Contraseña:
- Base de datos: colegio

La conexión a las bases de datos se implementará utilizando código de programación 1

Código de programación 1

*Código de conexión a base de datos*

```
import mysql.connector  
mydb = mysql.connector.connect( host="127.0.0.1", user="root", password=""  
,database="colegio" )
```

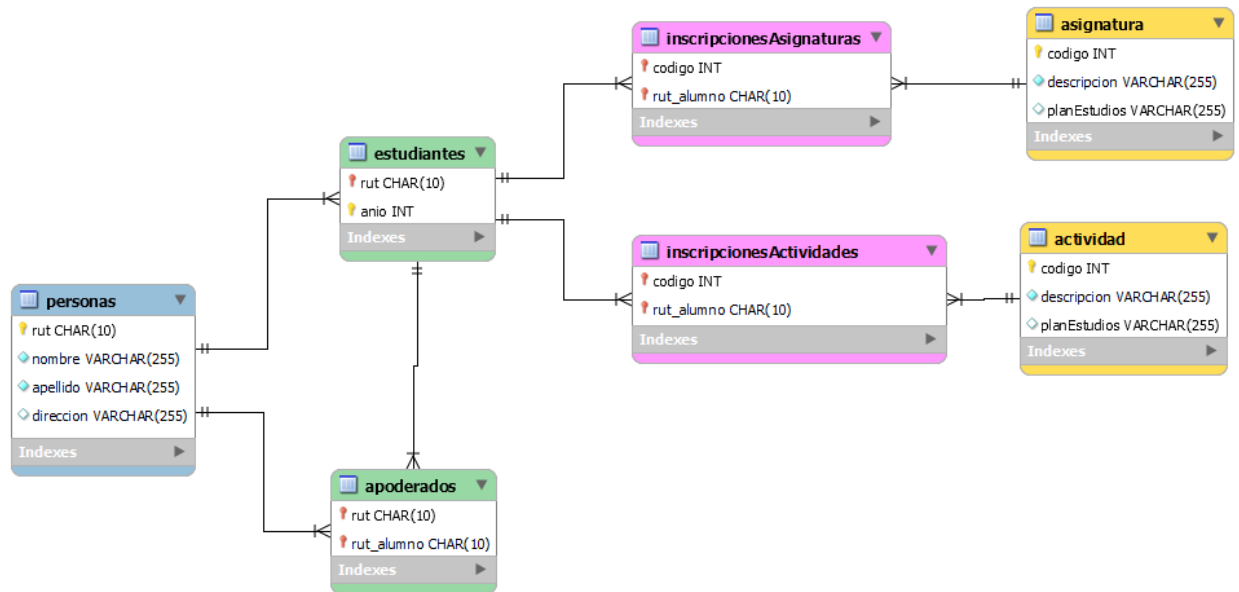
Fuente: Elaboración propia

El modelo que cumple los requisitos del trabajo se puede observar en la figura 1, la cual contiene la tabla *personas* que esta vinculada a las tablas *estudiantes* y *apoderados*, luego se pueden apreciar las tablas *inscripcionesActividades* y *inscripcionesAsignatura*, las cuales están relacionadas a la tabla *estudiantes* por medio del rut del alumno y a las tablas *asignatura* y *actividad* mediante llaves foráneas.

Las relaciones entre tablas permiten asegurar la integridad de la información debido a que si se eliminan o actualizan datos que tengan alguna restricción, las consultas generarán errores de ejecución evitando que los datos sean corrompidos por error.

Figura 1

*Modelo de base de datos*



Fuente: elaboración propia

Para realizar la creación de la base de datos se sincronizo el modelo de la base de datos y en proceso se obtuvo el código fuente para crear la base de datos, para poder ejecutar estas instrucciones se separo la consulta en 9 archivos de extensión sql que se pueden observar en códigos programación 2, códigos programación 3, códigos programación 4, códigos programación 5, códigos programación 6, códigos programación 7, códigos programación 8, códigos programación 9 y códigos programación 10.

Se implemento una rutina de ejecución vista en código de programación 11 de las los comando DDL de sql mediante la lectura de los archivos con extensión sql . Al ejecutar código de programación 11, se puede apreciar la creación de la base de datos y sus tablas en la figura 2.

## Código de programación 2

*Nombre archivo: db01.sql*

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
CREATE SCHEMA IF NOT EXISTS `colegio` DEFAULT CHARACTER SET utf8 ;
```

Fuente: Elaboración propia

## Código de programación 3

*Nombre archivo: db02.sql*

```
CREATE TABLE IF NOT EXISTS `colegio`.`personas` (
  `rut` CHAR(10) NOT NULL,
  `nombre` VARCHAR(255) NOT NULL,
  `apellido` VARCHAR(255) NOT NULL,
  `direccion` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`rut`),
  UNIQUE INDEX `rut_UNIQUE` (`rut` ASC))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;
```

Fuente: Elaboración propia

## Código de programación 4

*Nombre archivo: db03.sql*

```
CREATE TABLE IF NOT EXISTS `colegio`.`estudiantes` (
  `rut` CHAR(10) NOT NULL,
  `anio` INT(11) NOT NULL,
  INDEX `rut_a` (`rut` ASC),
  PRIMARY KEY (`rut`, `anio`),
  CONSTRAINT `rut_p`
    FOREIGN KEY (`rut`)
    REFERENCES `colegio`.`personas` (`rut`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;
```

Fuente: Elaboración propia

## Código de programación 5

*Nombre archivo: db04.sql*

```
CREATE TABLE IF NOT EXISTS `colegio`.`apoderados` (  
  `rut` CHAR(10) NOT NULL,  
  `rut_alumno` CHAR(10) NOT NULL,  
  INDEX `rut_a_idx` (`rut_alumno` ASC),  
  PRIMARY KEY (`rut`, `rut_alumno`),  
  CONSTRAINT `rut_p1`  
    FOREIGN KEY (`rut`)  
    REFERENCES `colegio`.`personas` (`rut`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `rut_a1`  
    FOREIGN KEY (`rut_alumno`)  
    REFERENCES `colegio`.`estudiantes` (`rut`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4;
```

Fuente: Elaboración propia

## Código de programación 6

*Nombre archivo: db05.sql*

```
CREATE TABLE IF NOT EXISTS `colegio`.`asignatura` (  
  `codigo` INT(11) NOT NULL,  
  `descripcion` VARCHAR(255) NOT NULL,  
  `planEstudios` VARCHAR(255) NULL DEFAULT NULL,  
  PRIMARY KEY (`codigo`),  
  UNIQUE INDEX `codigo_UNIQUE` (`codigo` ASC))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4;
```

Fuente: Elaboración propia

## Código de programación 7

*Nombre archivo: db06.sql*

```
CREATE TABLE IF NOT EXISTS `colegio`.`actividad` (  
  `codigo` INT(11) NOT NULL,  
  `descripcion` VARCHAR(255) NOT NULL,  
  `planEstudios` VARCHAR(255) NULL DEFAULT NULL,  
  PRIMARY KEY (`codigo`),
```

```
    UNIQUE INDEX `codigo_UNIQUE` (`codigo` ASC))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;
```

Fuente: Elaboración propia

Código de programación 8

*Nombre archivo: db07.sql*

```
CREATE TABLE IF NOT EXISTS `colegio`.`inscripcionesAsignaturas` (
  `codigo` INT(11) NOT NULL,
  `rut_alumno` CHAR(10) NOT NULL,
  INDEX `rut_a_idx` (`rut_alumno` ASC),
  INDEX `rut_a` (`rut_alumno` ASC),
  INDEX `codigo_a` (`codigo` ASC),
  PRIMARY KEY (`codigo`, `rut_alumno`),
  CONSTRAINT `codigo_a2`
    FOREIGN KEY (`codigo`)
    REFERENCES `colegio`.`asignatura` (`codigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `rut_a2`
    FOREIGN KEY (`rut_alumno`)
    REFERENCES `colegio`.`estudiantes` (`rut`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;
```

Fuente: Elaboración propia

Código de programación 9

*Nombre archivo: db08.sql*

```
CREATE TABLE IF NOT EXISTS `colegio`.`inscripcionesActividades` (
  `codigo` INT(11) NOT NULL,
  `rut_alumno` CHAR(10) NOT NULL,
  INDEX `rut_a_idx` (`rut_alumno` ASC),
  INDEX `rut_a` (`rut_alumno` ASC),
  INDEX `codigo_a` (`codigo` ASC),
  PRIMARY KEY (`codigo`, `rut_alumno`),
  CONSTRAINT `codigo_a3`
    FOREIGN KEY (`codigo`)
    REFERENCES `colegio`.`actividad` (`codigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
```

```

CONSTRAINT `rut_a3`
  FOREIGN KEY (`rut_alumno`)
  REFERENCES `colegio`.`estudiantes` (`rut`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;

```

Fuente: Elaboración propia

Código de programación 10

*Nombre archivo: db09.sql*

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

Fuente: Elaboración propia

Código de programación 11

*Nombre archivo: crear\_db.py*

```

import mysql.connector
mydb = mysql.connector.connect( host="127.0.0.1", user="root", password=""
, database="colegio" )
mycursor = mydb.cursor()

for i in range(1,10):
    archivos = "db0"+str(i) + ".sql"
    if i == 1 or i == 9:
        f = open(archivos, "r")
        lineas = f.readlines()
        f.close()
        for query in lineas:
            print(query)
            mycursor.execute(query)

    else:
        f = open (archivos, 'r')
        query = f.read()
        f.close()
        print(query)
        mycursor.execute(query)

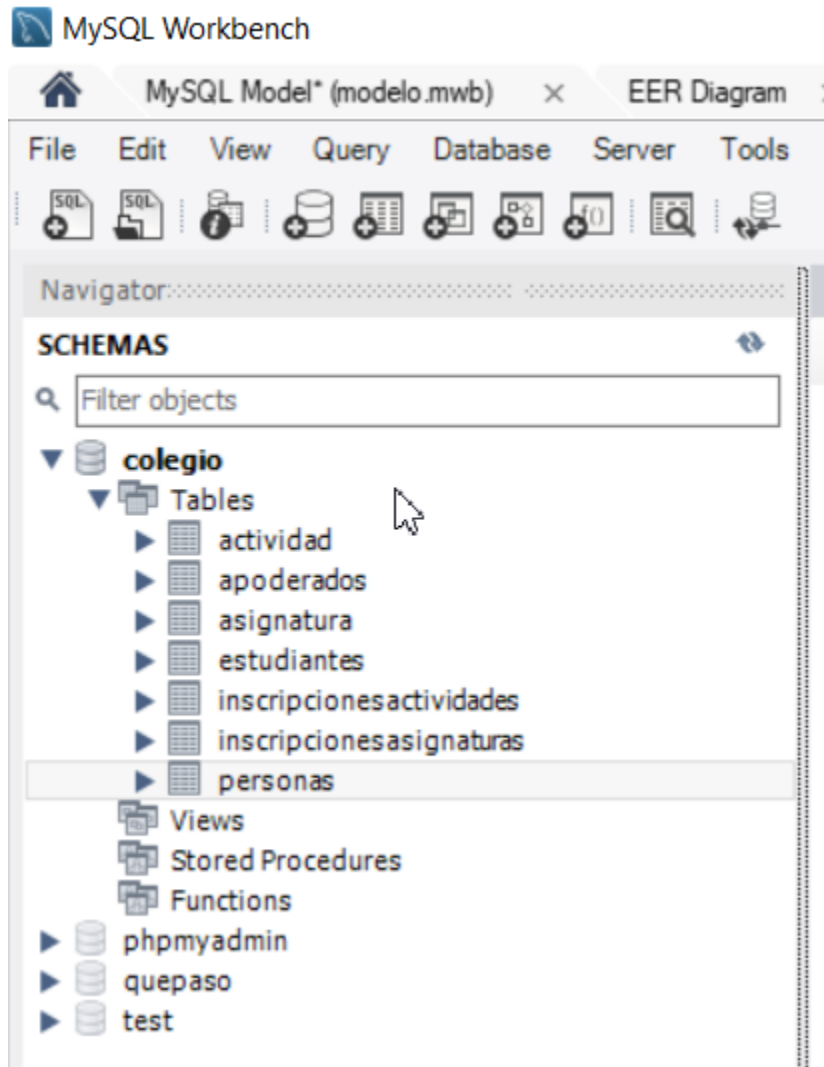
```

Fuente: Elaboración propia



Figura 2

*Vista de MySQL WorkBench de base de datos creada*



Fuente: elaboración propia

## ***Etapas 2***

Para realizar las actividades 2 y 3 se implementó una aplicación en Python un sistema de menús por consola que permiten actualizar las bases de datos. El script se puede ver en código de programación 12 y su vista de ejecución en la figura 3. En este menú se puede observar las siguientes opciones:

1. Estudiantes: Es un CRUD para mantener la tabla *estudiantes*.
2. Apoderados: Es un CRUD para mantener la tabla *apoderados*.
3. Asignaturas: Es un CRUD para mantener la tabla *asignatura*.
4. Actividades: Es un CRUD para mantener la tabla *actividad*.
5. Agregar alumno a asignaturas: Permite agregar alumnos a asignaturas que ya tengan un código creado.
6. Agregar alumnos a actividades: Permite agregar alumnos a actividades que ya tengan un código creado
7. Salir: Permite cerrar la aplicación

Código de programación 12

*Nombre archivo: menú.py*

```
import os
import estudiantes
import apoderados
import actividades
import asignaturas
import addAsignaturas
import addActividades

def menu():
    print("1-Estudiantes")
    print("2-Apoderados")
    print("3-Asignaturas")
    print("4-Actividades")
    print("5-Agregar alumno a asignaturas")
    print("6-Agregar alumno a actividades")
    print("7-salir")
    print("Selecciones una opcion")

op = 0
while( op != 7):
    os.system("cls")
    menu()
    op = int(input())
    if op == 1:
```

```

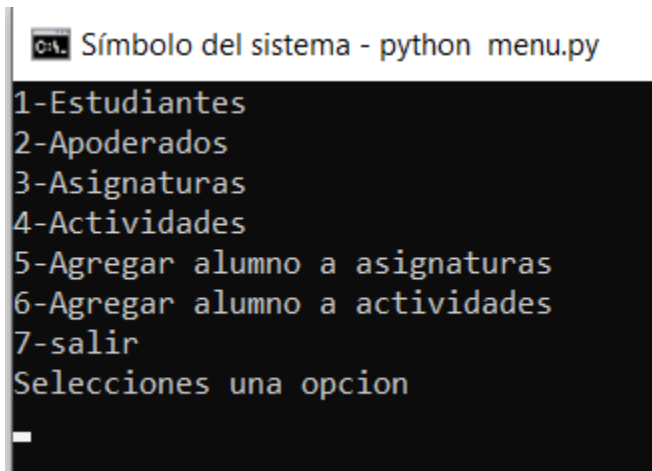
    estudiantes.principal()
if op == 2:
    apoderados.principal ()
if op == 3:
    asignaturas.principal()
if op == 4:
    actividades.principal()
if op == 5:
    addAsignaturas.principal()
if op == 6:
    addActividades.principal()

```

Fuente: Elaboración propia

Figura 3

### *Menú principal*



```

Símbolo del sistema - python menu.py
1-Estudiantes
2-Apoderados
3-Asignaturas
4-Actividades
5-Agregar alumno a asignaturas
6-Agregar alumno a actividades
7-salir
Selecciones una opcion
_

```

Fuente: elaboración propia

Cuando seleccione la opción estudiante se vera la ejecución del script de código de programación 13, el cual posee el CRUD con la implementación de las consultas DML necesarias en conjunto con un menú que permite listar los estudiantes y realizar el mantenimiento de sus datos que se puede apreciar en la figura 4

Al presionar la opción 1 del menú de estudiantes se solicita que ingrese el RUT del estudiante y así de esta forma realiza la búsqueda de datos que será mostrada tal como se ve en la figura 5.

## Código de programación 13

Nombre archivo: *estudiantes.py*

```
import os
import mysql.connector

def menu():
    print("1-Actualizar estudiante")
    print("2-Listas estudiantes")
    print("3-salir")
    print("Selecciones una opcion")

def buscar(rut):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `personas` WHERE rut = '"+rut+"'"
    mycursor.execute(query)
    datos1 = mycursor.fetchall()
    filas1 = mycursor.rowcount
    query = "SELECT * FROM `estudiantes` WHERE rut = '"+rut+"'"
    mycursor.execute(query)
    datos2 = mycursor.fetchall()
    filas2 = mycursor.rowcount
    if filas2 == 1:
        an = datos2[0][1]
    else:
        an = ""
    if( filas1 == 1):
        return (True, datos1, an)
    else:
        return (False, datos1, an)

def insertar(rut,nombre,apellido, direccion,anio):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "INSERT INTO `personas` (`rut`, `nombre`, `apellido`, `direccion`)"
    query += " VALUES ('"+rut+"', '"+nombre+"', '"+apellido+"', '"+direccion+"');"
    mycursor.execute(query)
```

```

        mydb.commit()
        query = "INSERT INTO `estudiantes` (`rut`, `anio`) VALUES ('"+rut+"',
        '"+anio+"');"
        mycursor.execute(query)
        mydb.commit()

def actualizar(rut,nombre,apellido, direccion, anio):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="",database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "UPDATE `personas` SET `nombre` = '"+nombre+"', `apellido` = '
    "+apellido
    query += "', `direccion` = '"+direccion+" WHERE `personas`.`rut` = '"
    +rut+"';"
    mycursor.execute(query)
    mydb.commit()
    query = "UPDATE `estudiantes` SET `anio` = '"+anio+" WHERE `estudiant
    es`.`rut` = '"+rut+"';"
    mycursor.execute(query)
    mydb.commit()

def eliminar(rut):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="",database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "DELETE FROM `estudiantes` WHERE `estudiantes`.`rut` = \'" +rut
    +"\'"
    mycursor.execute(query)
    mydb.commit()
    query = "DELETE FROM `personas` WHERE `personas`.`rut` = \'" +rut+"\'"
    mycursor.execute(query)
    mydb.commit()

def agregar():
    rut = input("Ingrese RUT ")
    (existe, datos, anio) = buscar(rut)
    if existe:
        print( "RUT:", datos[0][0] )
        print( "Nombre:", datos[0][1] )
        print( "Apellido:", datos[0][2] )
        print( "Direccion:", datos[0][3] )
        print( "Año:", anio )
        print("Estudiante ya existe")
        print("1-Actualizar informacion de estudiante")

```

```

        print("2-Eliminar informacion de estudiante")
        print("Cualquier otra opcion volvera al menu")
        try:
            op = int(input())
        except:
            op = 0
        if op == 1:
            print( "Nombre")
            nombre = input()
            print( "Apellido")
            apellido = input()
            print( "Direccion")
            direccion = input()
            print( "Año")
            anio = input()
            actualizar(rut,nombre,apellido,direccion,anio)
        if op == 2:
            eliminar(rut)
        else:
            print("Estudiante no existe, si desea agregarlo ingrese los datos"
)
            print("De lo contrario presione enter en los demas campos sin ingr
esar informacion")
            print( "Nombre")
            nombre = input()
            print( "Apellido")
            apellido = input()
            print( "Direccion")
            direccion = input()
            print( "Año")
            anio = input()
            if(nombre == "" or apellido == "" or direccion == "" or anio == 0
):
                pass
            else:
                insertar(rut,nombre,direccion,apellido, anio)

def listar():
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", passwor
d="",database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT `personas`.`rut` AS 'RUT', `personas`.`nombre` AS 'Nom
mbre',"

```

```

        query += " `personas`.`apellido` AS 'Apellido', `personas`.`direccion`
        AS 'Direccion', "
        query += "`estudiantes`.`anio` AS 'Anio' FROM `personas` LEFT JOIN `es
        tudiantes` ON `personas`.`rut` = `estudiantes`.`rut`"
        query += "WHERE `estudiantes`.`anio` > 0"
        mycursor.execute(query)
        datos = mycursor.fetchall()
        for elemento in datos:
            print(elemento)
        input("Presione tecla para continuar")

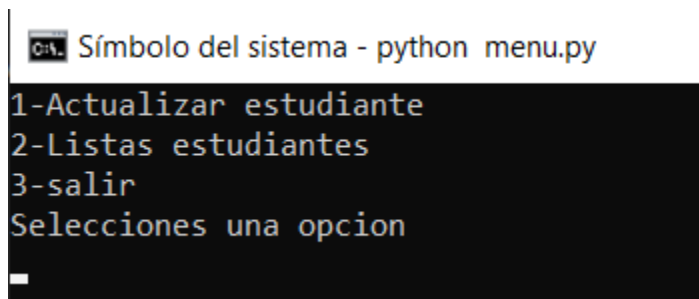
def principal():
    op = 0
    while( op != 3):
        os.system ("cls")
        menu()
        op = int(input())
        if op == 1:
            agregar()
        if op == 2:
            listar()

```

Fuente: Elaboración propia

Figura 4

#### *Menú Estudiante*



```

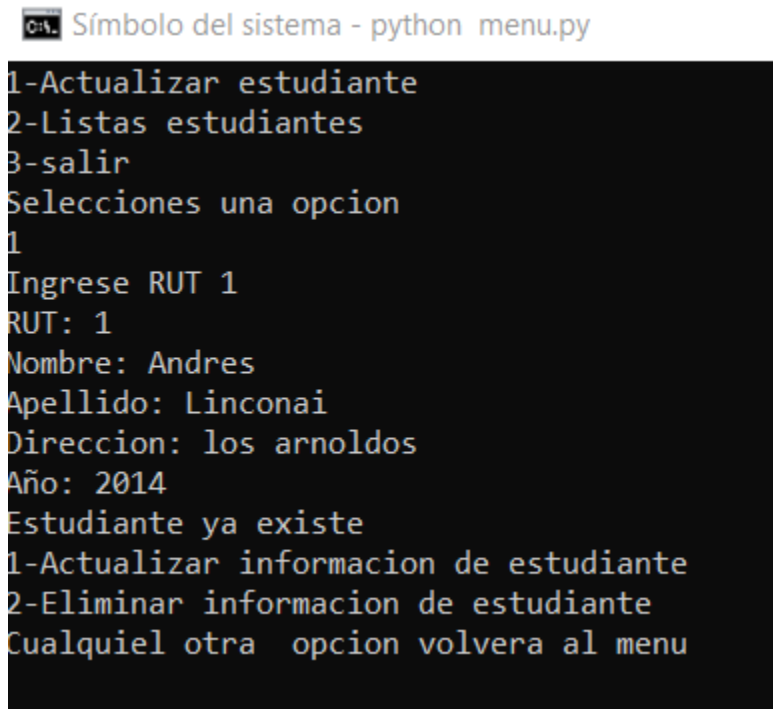
C:\> Símbolo del sistema - python menu.py
1-Actualizar estudiante
2-Listas estudiantes
3-salir
Selecciones una opcion
_

```

Fuente: elaboración propia

Figura 5

*Mantenimiento de estudiante cuando el estudiante existe*



```
Símbolo del sistema - python menu.py
1-Actualizar estudiante
2-Listas estudiantes
3-salir
Selecciones una opcion
1
Ingrese RUT 1
RUT: 1
Nombre: Andres
Apellido: Linconai
Direccion: los arnoldos
Año: 2014
Estudiante ya existe
1-Actualizar informacion de estudiante
2-Eliminar informacion de estudiante
Cualquier otra opcion volvera al menu
```

Fuente: elaboración propia

Al presionar la opción 1 solicitará que reingrese todos los datos del estudiante forma se verá el ingreso tal como se aprecia en la figura 6.

Al ingresar los datos volverá el menú de la figura 4, y en esta ocasión se mostrará la opción 2 que muestra todos los datos de los alumnos existentes tal como aprecia en la aplicación en la figura 7 y para mostrar la modificación de datos se ve también en de phpMyAdmin en la figura 8. En la figura 8 se puede apreciar en forma que la tabla *estudiantes* solo contiene los datos de rut y año ya que los datos generales de personas se encuentran en la tabla *personas*.



Figura 6

*Mantenimiento de estudiante cuando el estudiante existe*

```
Símbolo del sistema - python menu.py
Direccion: los arnoldos
Año: 2014
Estudiante ya existe
1-Actualizar informacion de estudiante
2-Eliminar informacion de estudiante
Cualquier otra opcion volvera al menu
1
Nombre
Jorge
Apellido
Linconao
Direccion
Los alerces
Año
2021
```

Fuente: elaboración propia

Figura 7

*Listado de estudiantes en aplicación de consola*

```
Símbolo del sistema - python menu.py
1-Actualizar estudiante
2-Listas estudiantes
3-salir
Selecciones una opcion
2
('1', 'Jorge', 'Linconao', 'Los alerces', 2021)
('1234678-2', 'ana', 'vaidez', 'las castanas', 2001)
('1234678-3', 'jorge', 'peñaloza', 'las castañas', 3066)
('2', 'andres', 'tres ', 'jorquera', 852)
('3', 'andrea', 'las magnolias', 'carrasco', 2001)
('4', 'Jorge ', 'las', 'Peña', 2110)
Presione tecla para continuar
```

Fuente: elaboración propia

Figura 8

*Listado de estudiantes en phpMyAdmin*

Servidor: 127.0.0.1 » Base de datos: colegio » Tabla: estudiantes

Examinar Estructura SQL Buscar Insertar Exportar

Mostrando filas 0 - 5 (total de 6, La consulta tardó 0,0005 segundos.)

`SELECT * FROM `estudiantes``

☐ Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

+ Opciones

				rut	anio
<input type="checkbox"/>	Editar	Copiar	Borrar	1	2021
<input type="checkbox"/>	Editar	Copiar	Borrar	1234678-2	2001
<input type="checkbox"/>	Editar	Copiar	Borrar	1234678-3	3066
<input type="checkbox"/>	Editar	Copiar	Borrar	2	852
<input type="checkbox"/>	Editar	Copiar	Borrar	3	2001
<input type="checkbox"/>	Editar	Copiar	Borrar	4	2110

Fuente: elaboración propia

Luego de comprobar que la modificación de datos funciona se procede a eliminar un registro ya existente en la base de datos de ana valdes rut 1234678-2 visto en la figura 9, y su comprobación se vera en la figura 10 con phpMyAdmin, en la cual ya no aparece mas el usuario, cabe destacar que la aplicación se cae cuando existen relaciones a otras tablas que contienen registros que se están intentando eliminar. Como información adicional se puede ver la figura 11 la tabla personas, en la cual ya no existe el usuario.

Figura 9

*Vista de usuario antes de eliminación*

```
Símbolo del sistema - python menu.py
1-Actualizar estudiante
2-Listas estudiantes
3-salir
Selecciones una opcion
1
Ingrese RUT 1234678-2
RUT: 1234678-2
Nombre: ana
Apellido: valdez
Direccion: las castañas
Año: 2001
Estudiante ya existe
1-Actualizar informacion de estudiante
2-Eliminar informacion de estudiante
Cualquier otra opcion volvera al menu
```

Fuente: elaboración propia

Figura 10

*Vista de usuario después de eliminación*

+ Opciones					
				rut	anio
<input type="checkbox"/>	✎ Editar	📋 Copiar	🗑 Borrار	1	2021
<input type="checkbox"/>	✎ Editar	📋 Copiar	🗑 Borrار	1234678-3	3066
<input type="checkbox"/>	✎ Editar	📋 Copiar	🗑 Borrار	2	852
<input type="checkbox"/>	✎ Editar	📋 Copiar	🗑 Borrار	3	2001
<input type="checkbox"/>	✎ Editar	📋 Copiar	🗑 Borrار	4	2110

Fuente: elaboración propia

Figura 11

*Vista de usuario después de eliminación*

				rut	nombre	apellido	1	direccion		
<input type="checkbox"/>		Editar		Copiar		Borrar	1234678-4	1	1	1
<input type="checkbox"/>		Editar		Copiar		Borrar	9	1	1	1
<input type="checkbox"/>		Editar		Copiar		Borrar	1234678-3	jorge	andrade	vicuña
<input type="checkbox"/>		Editar		Copiar		Borrar	4	Jorge	las	Peña
<input type="checkbox"/>		Editar		Copiar		Borrar	3	andrea	las magnolias	carrasco
<input type="checkbox"/>		Editar		Copiar		Borrar	1	Jorge	Linconao	Los alerces
<input type="checkbox"/>		Editar		Copiar		Borrar	1234678-1	jorge	peñaloza	las castañas
<input type="checkbox"/>		Editar		Copiar		Borrar	2	andres	tres	jorquera

Fuente: elaboración propia

En relación con la opción 2 de modificación de apoderados de la figura 4, se puede observar su script en código de programación 14 que es casi idéntico a código de programación 13 debido que su relación con la tabla *personas*, su única diferencia es que la tabla *apoderados* tiene los campos *rut* y *rut\_alumno* tal como se aprecia en la figura 12. No se realizaron capturas de pantalla de su funcionamiento por lo explicado anteriormente, pero en el archivo van los códigos de programación para su ejecución.

Código de programación 14

*Nombre archivo: apoderados.sql*

```
import os
import mysql.connector

def menu():
    print("1-Actualizar apoderado")
    print("2-Listas apoderados")
    print("3-salir")
    print("Selecciones una opcion")

def buscar(rut):
```

```

    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="" ,database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `personas` WHERE rut = '"+rut+"'"
    mycursor.execute(query)
    datos1 = mycursor.fetchall()
    filas1 = mycursor.rowcount
    query = "SELECT * FROM `apoderados` WHERE rut = '"+rut+"'"
    mycursor.execute(query)
    datos2 = mycursor.fetchall()
    filas2 = mycursor.rowcount
    if filas2 == 1:
        an = datos2[0][1]
    else:
        an = ""
    if( filas1 == 1):
        return (True, datos1, an)
    else:
        return (False, datos1, an)

def insertar(rut,nombre,apellido, direccion, rut_a):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="" ,database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "INSERT INTO `personas` (`rut`, `nombre`, `apellido`, `direccion`)"
    query += " VALUES ('"+rut+"', '"+nombre+"', '"+apellido+"', '"+direccion+"');"
    mycursor.execute(query)
    mydb.commit()
    query = "INSERT INTO `apoderados` (`rut`, `rut_alumno`) VALUES ('"+rut+"', '"+rut_a+"');"
    print(query)
    mycursor.execute(query)
    mydb.commit()

def actualizar(rut,nombre,apellido, direccion, rut_a):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="" ,database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "UPDATE `personas` SET `nombre` = '"+nombre+"', `apellido` = '"
    query += apellido

```

```

        query += "', `direccion` = '"+direccion+"' WHERE `personas`.`rut` = '"
+rut+"';"
        mycursor.execute(query)
        mydb.commit()
        query = "UPDATE `apoderados` SET `rut_alumno` = '"+rut_a+"' WHERE `apo
derados`.`rut` = '"+rut+"';"
        mycursor.execute(query)
        mydb.commit()

def eliminar(rut):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", passwor
d="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "DELETE FROM `apoderados` WHERE `apoderados`.`rut` = \'"
+rut+"
\'"
    mycursor.execute(query)
    mydb.commit()
    query = "DELETE FROM `personas` WHERE `personas`.`rut` = \'"
+rut+"\'"
    mycursor.execute(query)
    mydb.commit()

def agregar():
    rut = input("Ingrese RUT ")
    (existe, datos, rut_a) = buscar(rut)
    if existe:
        print( "RUT:", datos[0][0] )
        print( "Nombre:", datos[0][1] )
        print( "Apellido:", datos[0][2] )
        print( "Direccion:", datos[0][3] )
        print( "Rut alumno:", rut_a )
        print("persona ya existe")
        print("1-Actualizar informacion de apoderado")
        print("2-Eliminar informacion de apoderado")
        print("Cualquier otra opcion volvera al menu")
        try:
            op = int(input())
        except:
            op = 0
        if op == 1:
            print( "Nombre")
            nombre = input()
            print( "Apellido")
            apellido = input()
            print( "Direccion")
            direccion = input()

```

```

        print( "Rut alumno")
        rut_a = input()
        actualizar(rut,nombre,apellido,direccion,rut_a)
    if op == 2:
        eliminar(rut)
    else:
        print("apoderado no existe, si desea agregarlo ingrese los datos")
        print("De lo contrario presione enter en los demas campos sin ing
esar informacion")
        print( "Nombre")
        nombre = input()
        print( "Apellido")
        apellido = input()
        print( "Direccion")
        direccion = input()
        print( "Rut alumno")
        rut_a = input()
        if(nombre == "" or apellido == "" or direccion == "" or rut_a == 0
):
            pass
        else:
            insertar(rut,nombre,direccion,apellido, rut_a)

def listar():
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", passwor
d="",database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT a.rut, p.nombre, p.apellido, p.direccion, e.rut FROM a
poderados as a "
    query += "LEFT JOIN personas AS p ON a.rut = p.rut LEFT JOIN estudiant
es as e ON e.rut = a.rut_alumno;"
    mycursor.execute(query)
    datos = mycursor.fetchall()
    for elemento in datos:
        print(elemento)
    input("Presione tecla para continuar")

def principal():
    op = 0
    while( op != 3):
        os.system ("cls")
        menu()
        op = int(input())
        if op == 1:

```

```

    agregar()
    if op == 2:
        listar()

```

Fuente: Elaboración propia

Figura 12

*Vista de tabla apoderados*

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0004 segundos.)

`SELECT * FROM `apoderados``

☐ Perfilando [ [Editar en línea](#) ] [ [Editar](#) ] [ [Explicar SQL](#) ] [ [Crear código PHP](#) ] [ [Actualizar](#) ]

☐ Mostrar todo | Número de filas: 25  Filtrar filas:

+ Opciones

← T →

☐ Editar Copiar Borrar 9 2

Fuente: elaboración propia

Los CRUD de Asignaturas y Actividades son idénticos en modelo de tablas y código de programación, por lo cual se explicara con capturas de pantalla solo la opción 3 que es de mantenimiento de asignaturas. Para ver el código fuente de del CRUD de asignaturas puede analizar el código de programación 15 y para analizar el CRUD de actividades se puede revisar a través de código de programación 16.

Código de programación 15

*Nombre archivo: asignaturas.py*

```

import os
import mysql.connector

def menu():
    print("1-Actualizar asignatura")
    print("2-Listas asignatura")
    print("3-salir")
    print("Selecciones una opcion")

```



```

def buscar(codigo):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `asignatura` WHERE codigo = '"+codigo+"'"
    mycursor.execute(query)
    datos1 = mycursor.fetchall()
    filas1 = mycursor.rowcount
    if( filas1 == 1):
        return (True, datos1)
    else:
        return (False, datos1)

def insertar(codigo,descripcion,planEstudios):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "INSERT INTO `asignatura` (`codigo`, `descripcion`, `planEstudios`)"
    query += " VALUES ('"+codigo+"', '"+descripcion+"', '"+planEstudios+"'"
    query += ");"
    mycursor.execute(query)
    mydb.commit()

def actualizar(codigo,descripcion,planEstudios):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "UPDATE `asignatura` SET `descripcion` = '"+descripcion+"', `planEstudios` = '"+planEstudios"
    query += "' WHERE `asignatura`.`codigo` = '"+codigo+"';"
    mycursor.execute(query)
    mydb.commit()

def eliminar(codigo):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")

```

```

        query = "DELETE FROM `asignatura` WHERE `asignatura`.`codigo` = \''+codigo+'\"
mycursor.execute(query)
mydb.commit()

def agregar():
    codigo = input("Ingrese codigo ")
    (existe, datos) = buscar(codigo)
    if existe:
        print( "Codigo:", datos[0][0] )
        print( "Descripción:", datos[0][1] )
        print( "Plan de estudios:", datos[0][2] )

        print("asignatura ya existe")
        print("1-Actualizar informacion de asignatura")
        print("2-Eliminar informacion de asignatura")
        print("Cualquier otra opcion volvera al menu")
        try:
            op = int(input())
        except:
            op = 0
        if op == 1:
            print( "Descripción")
            descripcion = input()
            print( "Plan de estudios")
            planEstudios = input()

            actualizar(codigo,descripcion,planEstudios)
        if op == 2:
            eliminar(codigo)
        else:
            print("Asignatura no existe, si desea agregarla ingrese los datos"
)
            print("De lo contrario presione enter en los demas campos sin ingresar informacion")
            print( "Descripción")
            descripcion = input()
            print( "Plan de estudios")
            planEstudios = input()
            if(codigo == "" or descripcion == "" or planEstudios == ""):
                pass
            else:
                insertar(codigo,descripcion,planEstudios)

def listar():

```

```

    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="" ,database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `asignatura` WHERE 1;"
    mycursor.execute(query)
    datos = mycursor.fetchall()
    for elemento in datos:
        print(elemento)
    input("Presione tecla para continuar")

def principal():
    op = 0
    while( op != 3):
        os.system ("cls")
        menu()
        op = int(input())
        if op == 1:
            agregar()
        if op == 2:
            listar()

```

Fuente: Elaboración propia

Código de programación 16

*Nombre archivo: actividades.py*

```

import os
import mysql.connector

def menu():
    print("1-Actualizar actividad")
    print("2-Listas actividad")
    print("3-salir")
    print("Selecciones una opcion")

def buscar(codigo):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="" ,database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `actividad` WHERE codigo = '"+codigo+"'"
    mycursor.execute(query)
    datos1 = mycursor.fetchall()
    filas1 = mycursor.rowcount

```

```

        if( filas1 == 1):
            return (True, datos1)
        else:
            return (False, datos1)

def insertar(codigo,descripcion,planEstudios):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "INSERT INTO `actividad` (`codigo`, `descripcion`, `planEstudios`)"
    query += " VALUES ('"+codigo+"', '"+descripcion+"', '"+planEstudios+"'"
    );"
    mycursor.execute(query)
    mydb.commit()

def actualizar(codigo,descripcion,planEstudios):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "UPDATE `actividad` SET `descripcion` = '"+descripcion+"', `planEstudios` = '"+planEstudios"
    query += "' WHERE `actividad`.`codigo` = '"+codigo+"';"
    mycursor.execute(query)
    mydb.commit()

def eliminar(codigo):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "DELETE FROM `actividad` WHERE `actividad`.`codigo` = \'" + codigo + "\'"
    mycursor.execute(query)
    mydb.commit()

def agregar():
    codigo = input("Ingrese codigo ")
    (existe, datos) = buscar(codigo)
    if existe:
        print( "Codigo:", datos[0][0] )
        print( "Descripción:", datos[0][1] )

```

```

        print( "Plan de estudios:", datos[0][2] )

    print("Actividad ya existe")
    print("1-Actualizar informacion de actividad")
    print("2-Eliminar informacion de actividad")
    print("Cualquier otra opcion volvera al menu")
    try:
        op = int(input())
    except:
        op = 0
    if op == 1:
        print( "Descripción")
        descripcion = input()
        print( "Plan de estudios")
        planEstudios = input()

        actualizar(codigo,descripcion,planEstudios)
    if op == 2:
        eliminar(codigo)
    else:
        print("Actividad no existe, si desea agregarla ingrese los datos")
        print("De lo contrario presione enter en los demas campos sin ingresar informacion")
        print( "Descripción")
        descripcion = input()
        print( "Plan de estudios")
        planEstudios = input()
        if(codigo == "" or descripcion == "" or planEstudios == ""):
            pass
        else:
            insertar(codigo,descripcion,planEstudios)

def listar():
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `actividad` WHERE 1;"
    mycursor.execute(query)
    datos = mycursor.fetchall()
    for elemento in datos:
        print(elemento)
    input("Presione tecla para continuar")

def principal():

```

```

op = 0
while( op != 3):
    os.system("cls")
    menu()
    op = int(input())
    if op == 1:
        agregar()
    if op == 2:
        listar()

```


Fuente: Elaboración propia

Cuando se selecciona la opción 1 del submenú de asignaturas de la figura 13 se ingresa directamente a la mantención de esos datos. Y cuando se selecciona la opción 2 se puede observar el listado de asignaturas existentes en el sistema tal como se ve en la figura 14.

Figura 13

*Submenú de asignatura*

---

 Símbolo del sistema - python menu.py

```

1-Actualizar asignatura
2-Listas asignatura
3-salir
Selecciones una opcion
_

```

Fuente: elaboración propia

Cuando se selecciona la opción se solicita el código de asignatura, si esta existe permite modificar o eliminarla del sistema y si esta no existe permite agregar la asignatura.

Por ejemplo, al ingresar el código “3” que no existe se puede ver la pantalla de la figura 15 que indica la asignatura no existe, y se puede agregar si se ingresa la información necesaria.

Una vez insertada la información de la figura 15, se pueden hacer los cambios en la base en la tabla *asignatura* tal como se aprecia en la figura 16.

Figura 14

*Listado de asignaturas existentes*

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0006 segundos.)

```
SELECT * FROM `asignatura`
```

☐ Perfilando [ [Editar en línea](#) ] [ [Editar](#) ] [ [Explicar SQL](#) ] [ [Crear código PHP](#) ] [ [Actualizar](#) ]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:  Sort by key: Ninguna

+ Opciones

				codigo	descripcion	planEstudios			
<input type="checkbox"/>		Editar		Copiar		Borrar	1	ingles	verbos
<input type="checkbox"/>		Editar		Copiar		Borrar	2	matematica	funciones

☐ Seleccionar todo Para los elementos que están marcados: Ed

Símbolo del sistema - python menu.py

```
1-Actualizar asignatura
2-Listas asignatura
3-salir
Selecciones una opcion
2
(1, 'ingles', 'verbos')
(2, 'matematica', 'funciones')
Presione tecla para continuar
```

Fuente: elaboración propia

Figura 15

*Vista de ingreso de asignatura*

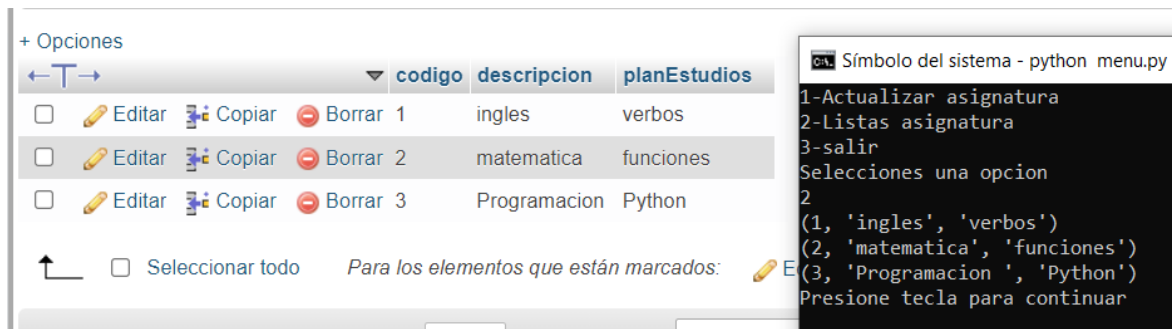
Símbolo del sistema - python menu.py

```
1-Actualizar asignatura
2-Listas asignatura
3-salir
Selecciones una opcion
1
Ingrese codigo 3
Asignatura no existe, si desea agregarla ingrese los datos
De lo contrario presione enter en los demas campos sin ingresar informacion
Descripción
Programacion
Plan de estudios
Python
```

Fuente: elaboración propia

Figura 16

*Vista de asignaturas*

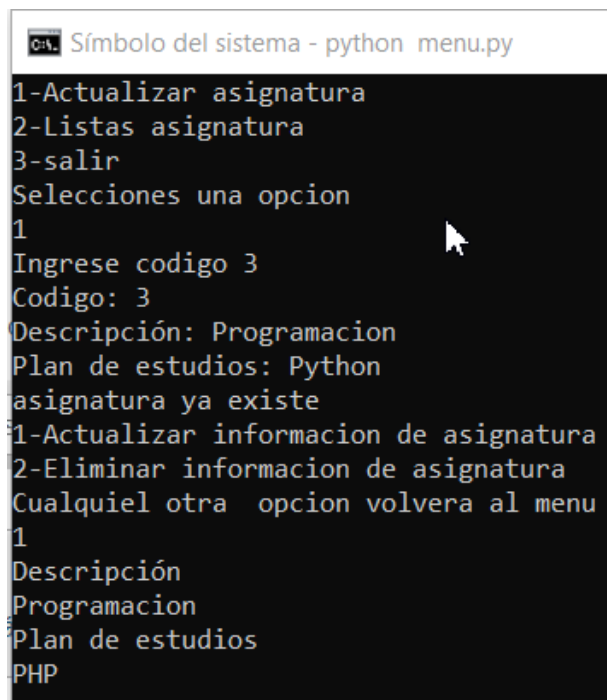


Fuente: elaboración propia

Posteriormente al ingresar nuevamente el código “3”, se puede apreciar la opción de modificar la información para lo cual se ingresan los datos de la figura 17, que se puede comprobar su modificación en la figura 18.

Figura 17

*Vista de modificación de asignaturas*



Fuente: elaboración propia



Figura 18

*Vista de modificación de asignaturas*

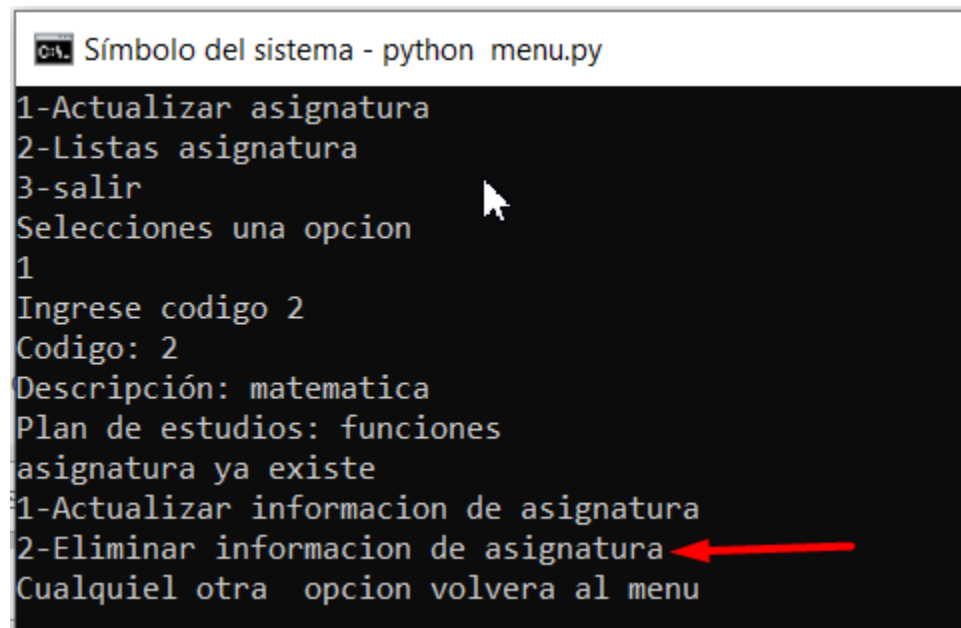


Fuente: elaboración propia

Luego se procede a eliminar la asignatura de matemáticas vista en la figura 19, y se comprueba su eliminación con la figura 20.

Figura 19

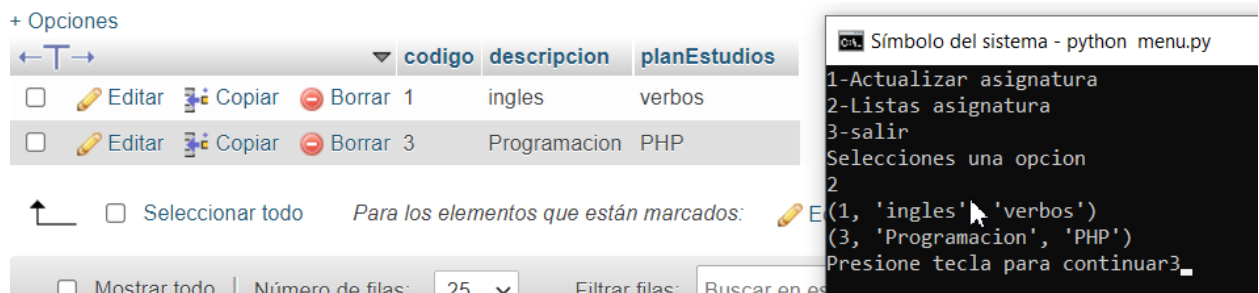
*Vista de eliminación de asignaturas*



Fuente: elaboración propia

Figura 20

*Vista de eliminación de asignaturas*



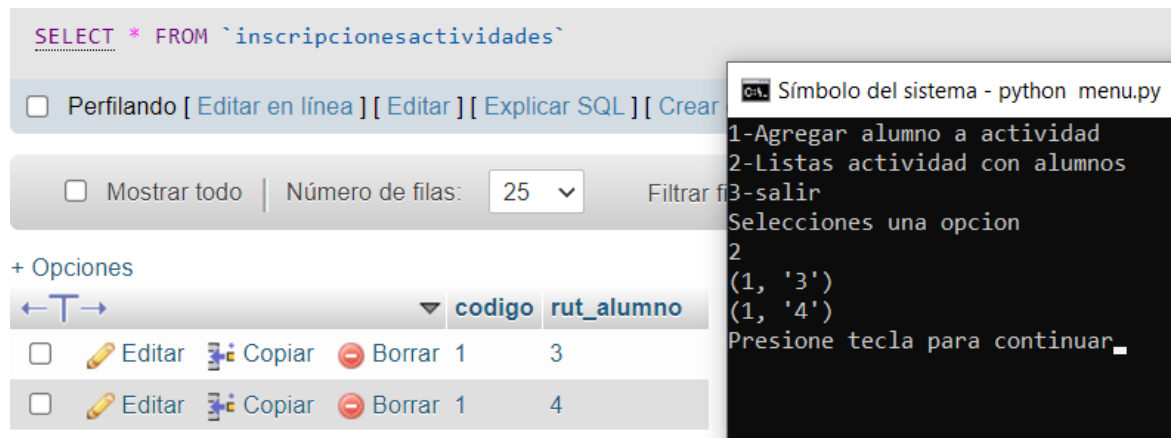
Fuente: elaboración propia

Ya con esta sección del programa explicada, se procede a explicar la forma de inscribir asignaturas, la sección de actividades solo se mostrarán las tablas que muestran sus datos ya que esta funciona de forma idéntica a asignaturas. Para ver que funciona correctamente la sección de ingreso de alumnos a actividades se puede ver la figura 21.

En código de programación 17 se puede ver el script de inscripción de actividades y en código de programación 18 se puede ver el script de inscripción de asignaturas.

Figura 21

*Vista de tabla actividades*



Fuente: elaboración propia

## Código de programación 17

Nombre archivo: *addActividades.py*

```
import os
import mysql.connector

def menu():
    print("1-Agregar alumno a actividad")
    print("2-Listas actividad con alumnos")
    print("3-salir")
    print("Selecciones una opcion")

def buscar(codigo, rut):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `inscripcionesactividades` WHERE codigo = '"+codigo+"' AND rut_alumno = '"+rut+"'"
    mycursor.execute(query)
    datos1 = mycursor.fetchall()
    filas1 = mycursor.rowcount
    if( filas1 == 1):
        return (True, datos1)
    else:
        return (False, datos1)

def insertar(codigo,rut_alumno):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "INSERT INTO `inscripcionesactividades` (`codigo`, `rut_alumno`)"
    query += " VALUES ('"+codigo+"', '"+rut_alumno+"');"
    mycursor.execute(query)
    mydb.commit()

def agregar():
    codigo = input("Ingrese codigo ")
    rut = input("Ingrese RUT alumno ")
    (existe, datos) = buscar(codigo, rut)
    if existe:
        print( "Codigo:", datos[0][0] )
```

```

        print( "RUT alumno:", datos[0][1] )
        print("Alumno ya esta agregado")
        print("Cualquier otra opcion volvera al menu")
        input()

    else:
        insertar(codigo,rut)
        print( "Agregando alumno")
        input()

def listar():
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password=" ", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `inscripcionesactividades` WHERE 1;"
    mycursor.execute(query)
    datos = mycursor.fetchall()
    for elemento in datos:
        print(elemento)
    input("Presione tecla para continuar")

def principal():
    op = 0
    while( op != 3):
        os.system ("cls")
        menu()
        op = int(input())
        if op == 1:
            agregar()
        if op == 2:
            listar()

```

Fuente: Elaboración propia

Código de programación 18

*Nombre archivo: addAsignaturas.py*

```

import os
import mysql.connector

def menu():
    print("1-Actualizar asignatura")

```

```

    print("2-Listas asignatura")
    print("3-salir")
    print("Selecciones una opcion")

def buscar(codigo):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `asignatura` WHERE codigo = '"+codigo+"'"
    mycursor.execute(query)
    datos1 = mycursor.fetchall()
    filas1 = mycursor.rowcount
    if( filas1 == 1):
        return (True, datos1)
    else:
        return (False, datos1)

def insertar(codigo,descripcion,planEstudios):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "INSERT INTO `asignatura` (`codigo`, `descripcion`, `planEstudios`)"
    query += " VALUES ('"+codigo+"', '"+descripcion+"', '"+planEstudios+"'"
    query += ");"
    mycursor.execute(query)
    mydb.commit()

def actualizar(codigo,descripcion,planEstudios):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "UPDATE `asignatura` SET `descripcion` = '"+descripcion+"', `planEstudios` = '"+planEstudios"
    query += "' WHERE `asignatura`.`codigo` = '"+codigo+"';"
    mycursor.execute(query)
    mydb.commit()

def eliminar(codigo):
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )

```

```

mycursor = mydb.cursor()
mycursor.execute("USE colegio;")
query = "DELETE FROM `asignatura` WHERE `asignatura`.`codigo` = \''+co
digo+'\'
mycursor.execute(query)
mydb.commit()

def agregar():
    codigo = input("Ingrese codigo ")
    (existe, datos) = buscar(codigo)
    if existe:
        print( "Codigo:", datos[0][0] )
        print( "Descripción:", datos[0][1] )
        print( "Plan de estudios:", datos[0][2] )

        print("asignatura ya existe")
        print("1-Actualizar informacion de asignatura")
        print("2-Eliminar informacion de asignatura")
        print("Cualquier otra opcion volvera al menu")
        try:
            op = int(input())
        except:
            op = 0
        if op == 1:
            print( "Descripción")
            descripcion = input()
            print( "Plan de estudios")
            planEstudios = input()

            actualizar(codigo,descripcion,planEstudios)
        if op == 2:
            eliminar(codigo)
        else:
            print("Asignatura no existe, si desea agregarla ingrese los datos"
)
            print("De lo contrario presione enter en los demas campos sin ingr
esar informacion")
            print( "Descripción")
            descripcion = input()
            print( "Plan de estudios")
            planEstudios = input()
            if(codigo == "" or descripcion == "" or planEstudios == ""):
                pass
            else:
                insertar(codigo,descripcion,planEstudios)

```

```

def listar():
    mydb = mysql.connector.connect( host="127.0.0.1", user="root", password="", database="test" )
    mycursor = mydb.cursor()
    mycursor.execute("USE colegio;")
    query = "SELECT * FROM `asignatura` WHERE 1;"
    mycursor.execute(query)
    datos = mycursor.fetchall()
    for elemento in datos:
        print(elemento)
    input("Presione tecla para continuar")

def principal():
    op = 0
    while( op != 3):
        os.system("cls")
        menu()
        op = int(input())
        if op == 1:
            agregar()
        if op == 2:
            listar()

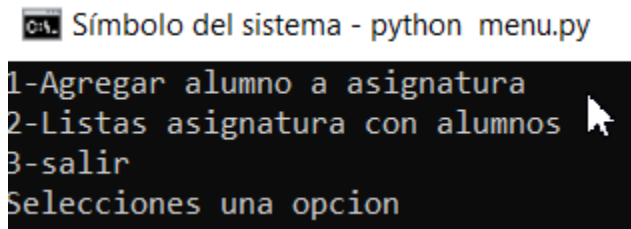
```

Fuente: Elaboración propia

En la figura 22 se puede observar el submenú de asignaturas, en la cual se puede ver que tiene las opciones de mantenimiento de datos y lista de alumnos por asignatura tal como se ve en la figura 23.

Figura 22

*Vista de submenú de asignaturas*



```

Símbolo del sistema - python menu.py
1-Agregar alumno a asignatura
2-Listas asignatura con alumnos
3-salir
Selecciones una opcion

```

Fuente: elaboración propia

Figura 23

*Listado de alumnos inscritos en alguna asignatura*

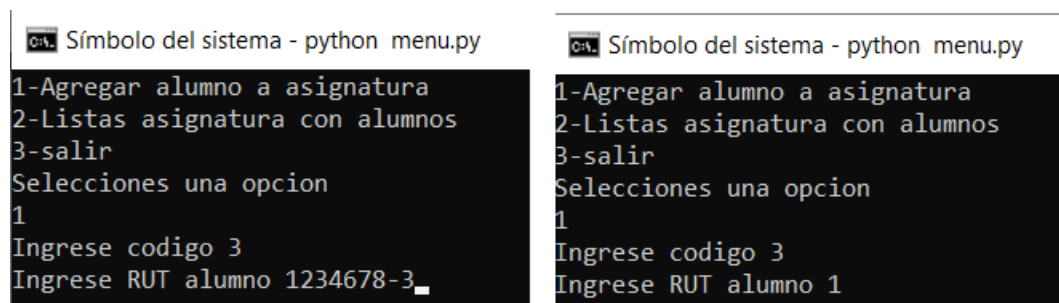


Fuente: elaboración propia

Al entrar al menú para agregar alumno a asignatura, se solicitan los datos de código de asignatura y RUT del estudiante, por lo cual se agregaran 2 alumnos al curso de programación vistos en la figura 24.

Figura 24

*Alumnos ingresados a Programacion*



Fuente: elaboración propia

En la figura 25 se pueden ver los alumnos inscritos en el curso de programación, y para comprobar su existencia en la base de datos se puede usar la sentencia y resultado vistos en la figura 26.



Figura 25

*Alumnos ingresados a programación*

The screenshot shows a web application interface with a SQL query editor at the top containing the query: `SELECT * FROM `inscripcionesasignaturas``. Below the editor are buttons for 'Perfilando', 'Editar en línea', 'Editar', 'Explicar SQL', 'Crear código PHP', and 'Actualizar'. A table below shows the results of the query with columns 'codigo' and 'rut\_alumno'. The table has four rows, with the last two rows highlighted in red. To the right of the table is a terminal window showing a command prompt with the text: '1-Agregar alumno a asignatura', '2-Listas asignatura con alumnos', '3-salir', 'Selecciones una opcion', '2', '(1, '1')', '(1, '4')', '(3, '1')', '(3, '1234678-3')', and 'Presione tecla para continuar'.

	codigo	rut_alumno
<input type="checkbox"/>	1	1
<input type="checkbox"/>	4	4
<input type="checkbox"/>	3	1
<input type="checkbox"/>	3	1234678-3

Fuente: elaboración propia

Figura 26

*Alumnos ingresados en programación*

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab is active, displaying a SQL query that joins the 'inscripcionesasignaturas' table with the 'personas' table. The query is: `SELECT inscripcionesasignaturas.codigo as codigo, inscripcionesasignaturas.rut_alumno as rut, personas.nombre, personas.apellido, asignatura.descripcion FROM inscripcionesasignaturas, personas, asignatura WHERE personas.rut = inscripcionesasignaturas.rut_alumno AND asignatura.codigo = inscripcionesasignaturas.codigo`. Below the query is the 'Result Grid' showing the results of the query. The grid has five columns: 'codigo', 'rut', 'nombre', 'apellido', and 'descripcion'. The results are as follows:

codigo	rut	nombre	apellido	descripcion
1	1	Jorge	Linconao	ingles
1	4	Jorge	las	ingles
3	1	Jorge	Linconao	Programacion
3	1234678-3	jorge	andrade	Programacion

Fuente: elaboración propia

## Bibliografía

- IACC. (2020). Bases de Datos con Python y MySQL. En *Programación avanzada 2. Semana 7*.  
IACC. (2021). Tarea semana 7.