

Memoria del Proyecto



Joël Arnaud Carreras

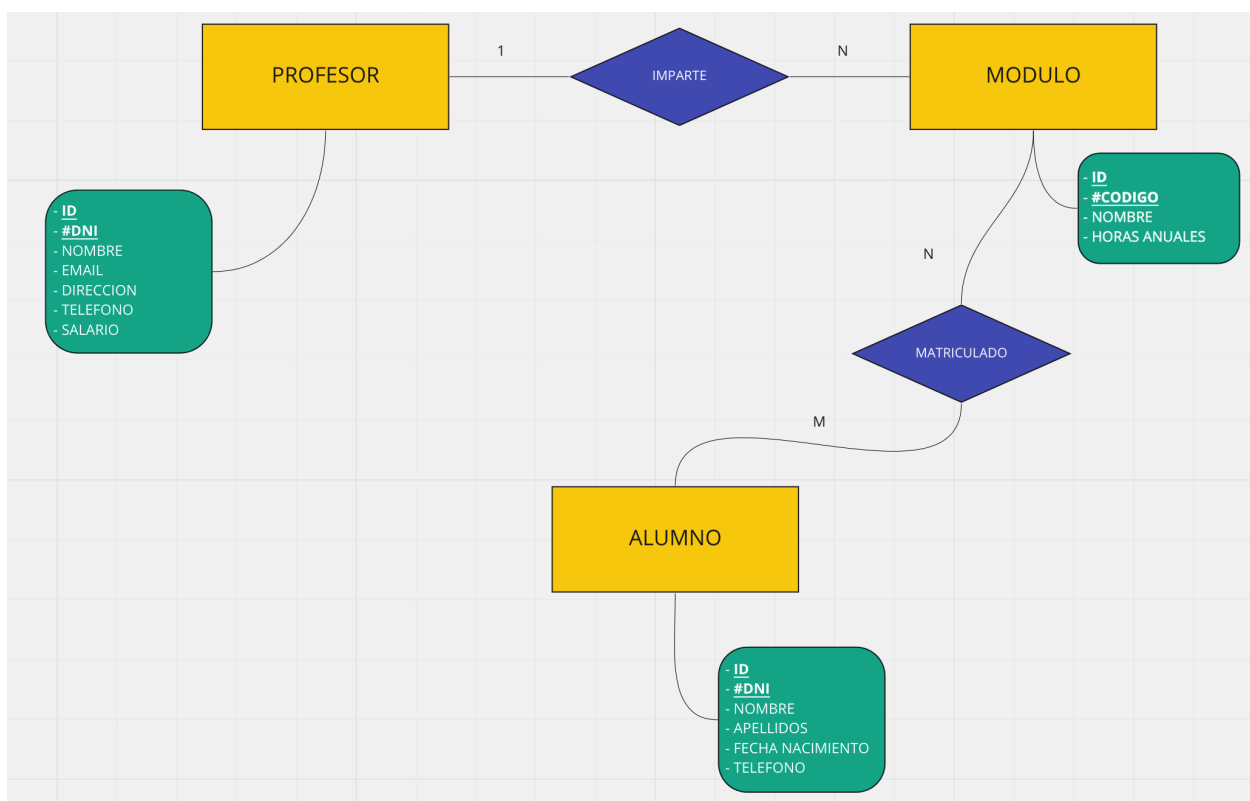
Indice

- Diagrama Entidad-Relación
- Modelo relacional
- SQL Creación de base de datos y tablas
- Consultas SQL usadas en el programa demo
- Muestra de funcionamiento del programa demo
- Tiempo de desarrollo individualizado en horas
- Dificultades encontradas
- Anexo I - Código programa python

Diagrama Entidad-Relación

Se desea diseñar la base de datos de un Centro Docente:

- En la base de datos se desea guardar los datos de los profesores del Centro Docente (DNI, nombre, email, dirección, teléfono y salario).
- Los profesores imparten módulos, y cada módulo tiene un código, un nombre y una cantidad de horas anuales.
- Cada alumno está matriculado en uno o varios módulos.
- De cada alumno se desea guardar el DNI, nombre, apellidos, fecha de nacimiento y teléfonos de contacto.
- Los profesores pueden impartir varios módulos, pero un módulo sólo puede ser impartido por un profesor.



Modelo Relacional

PROFESOR(ID, #DNI, NOMBRE, EMAIL, DIRECCION, TELEFONO, SALARIO_EUROS)

MODULO(ID, #CODIGO, NOMBRE, HORAS_ANUALES, ID_PROFESOR)

ALUMNO(ID, #DNI, NOMBRE, APELLIDOS, FECHA_NACIMIENTO, TELEFONO)

ALUMNO-MATRICULADO-MODULO(ID, ID_ALUMNO, ID_MODULO)

Creación Base de Datos y Tablas

```
CREATE SCHEMA IF NOT EXISTS Proyecto;
```

```
CREATE TABLE PROFESOR
```

```
(  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    DNI CHAR(9),  
    Nombre VARCHAR(70),  
    Email VARCHAR(255),  
    Direccion VARCHAR(255),  
    Telefono VARCHAR(15),  
    Salario_Euros FLOAT  
);
```

```
CREATE TABLE MODULO
```

```
(  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
   Codigo INT,  
    Nombre VARCHAR(70),  
    HORAS_ANUALES INT,  
    ID_PROFESOR INT,  
    FOREIGN KEY (ID_PROFESOR) REFERENCES PROFESOR(ID)  
);
```

```
CREATE TABLE ALUMNO
```

```
(  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    DNI CHAR(9),  
    Nombre VARCHAR(70),  
    Apellidos VARCHAR(70),  
    Fecha_Nacimiento VARCHAR(10),  
    Telefono VARCHAR(15)  
);
```

```
CREATE TABLE ALUMNO_MATRICULADO_MODULO
(
    ID INT PRIMARY KEY AUTO_INCREMENT,
    ID_ALUMNO INT,
    ID_MODULO INT,
    FOREIGN KEY (ID_ALUMNO) REFERENCES ALUMNO(ID),
    FOREIGN KEY (ID_MODULO) REFERENCES MODULO(ID)
);
```

Consultas SQL usadas en el programa

SELECT * FROM MODULO; → Esta consulta ha sido utilizada para convertir el código del módulo en ID del mismo y para comprobar si existía dicho módulo.

SELECT * FROM ALUMNO; → Esta consulta ha sido utilizada para convertir el DNI del alumno en ID del mismo y para comprobar si existía dicho alumno.

SELECT * FROM edad_alumnos LIMIT %s, 5; → Se ha utilizado para mostrar en paginación a los alumnos registrados.

INSERT INTO ALUMNO(DNI, Nombre, Apellidos, Fecha_Nacimiento, Telefono) VALUES (%s, %s, %s, %s, %s); → Se ha utilizado para introducir nuevos alumnos.

INSERT INTO ALUMNO_MATRICULADO_MODULO (ID_ALUMNO, ID_MODULO) VALUES(%s, %s); → Se ha utilizado para “matricular” alumnos a módulos.

DELETE FROM ALUMNO WHERE ID=%s; → Utilizado para eliminar alumnos.

SELECT * FROM lista_profesores_modulos; → Utilizado para listar a los profesores y sus respectivos módulos que imparten.

SELECT ceil(COUNT(*) / 5) FROM edad_alumnos; → Utilizado para el paginamiento del listado de alumnos.

Muestra funcionamiento del programa

Bienvenid@ a FlahseandoMuros2000

Elige una opción

Opción 1: Registrar nuevo alumno

Opción 2: Matricular alumno a módulo

Opción 3: Eliminar alumno

Opción 4: Listar módulos, profesor y numero alumnos

Opción 5: Listar alumnos del centro

Opción 6: Salir

Seleccione opción:


Registrar nuevo alumno

Introduzca su DNI: 



Introduzca su nombre: *Joel*

Introduzca su apellido: *Arnaud Carreras*

Introduzca la fecha de nacimiento (año-mes-día): *1999-11-27*

Introduzca su número de teléfono: 

Alumno añadido correctamente.

61		Joel	Arnaud C...	1999-11-27	
----	---	------	-------------	------------	---

Matricular alumno a módulo

DNI del Alumno: [REDACTED]

Código del módulo: 71975

Alumno matriculado correctamente.

61	61 →	6 →
----	------	-----

Eliminar Alumno

DNI del Alumno: [REDACTED]

DNI del alumno encontrado.

Alumno eliminado correctamente.

Listar módulos profesores y alumnos

```
('Emma', 'Ingrid Mann', 1)
('Vernon', 'Brynn Shaw', 10)
('Mufutau', 'Gannon Kaufman', 9)
('Fay', 'Daphne Rhodes', 6)
('Lev', 'Alfonso Lara', 2)
('Brenden', None, 12)
('Emerson', None, 7)
('Jamalia', None, 7)
('Adrienne', None, 4)
('Lawrence', None, 2)
```

Listar alumnos

```
(1, '53545144U', 'Dara', 'Bray', '2014-10-15', 7, '+34 609259696')
(2, '27656677C', 'Nigel', 'Miranda', '2015-02-06', 7, '+34 648872152')
(3, '66727362Q', 'Josiah', 'Lara', '2011-03-26', 11, '+34 662655922')
(4, '76635972J', 'Brody', 'Cochran', '2016-01-09', 6, '+34 641415227')
(5, '37643333V', 'Ciara', 'Holt', '2016-05-19', 6, '+34 627755976')
```

D - AVANZAR

A - RETROCEDER

S - SALIR

Página 1 de 12

Listar alumnos

```
(16, '44856198H', 'Vladimir', 'Woods', '2014-03-27', 8, '+34 641247713')
(17, '99767766D', 'Aquila', 'Brooks', '2012-02-15', 10, '+34 645109227')
(18, '23575657Y', 'Farrah', 'Griffin', '2016-06-09', 5, '+34 696621541')
(19, '81857599V', 'Yoshi', 'Travis', '2010-12-29', 11, '+34 605441266')
(20, '79122584H', 'Reed', 'Booth', '2014-05-16', 8, '+34 608181733')
```

D - AVANZAR

A - RETROCEDER

S - SALIR

Página 4 de 12

Dificultades encontradas

Algunas de las dificultades encontradas podrían ser el funcionamiento del código Python enlazado con SQL, es decir, nosotros estamos acostumbrados a trabajar con los mismos datos/variables de un mismo archivo, pero por ejemplo, si queríamos solo coger las ID's de los alumnos no sabíamos muy bien cómo hacerlo. Podríamos decir que nos ha faltado un poco de explicación del funcionamiento del mismo, pero nos hemos buscado la vida y consideramos que al final lo hemos sacado hacia adelante ya que no éramos los únicos que no sabían programar Python relacionado con SQL.

Anexo I - Código programa Python

<https://sourceb.in/jGlj5OWImQ>