

Academia Baile Mambolaye

**Alumno:** Ernesto Vega

**Profesor:** Miguel Rodas

**Tutor:** Juan Rabazzi

**Descripción Temática:**

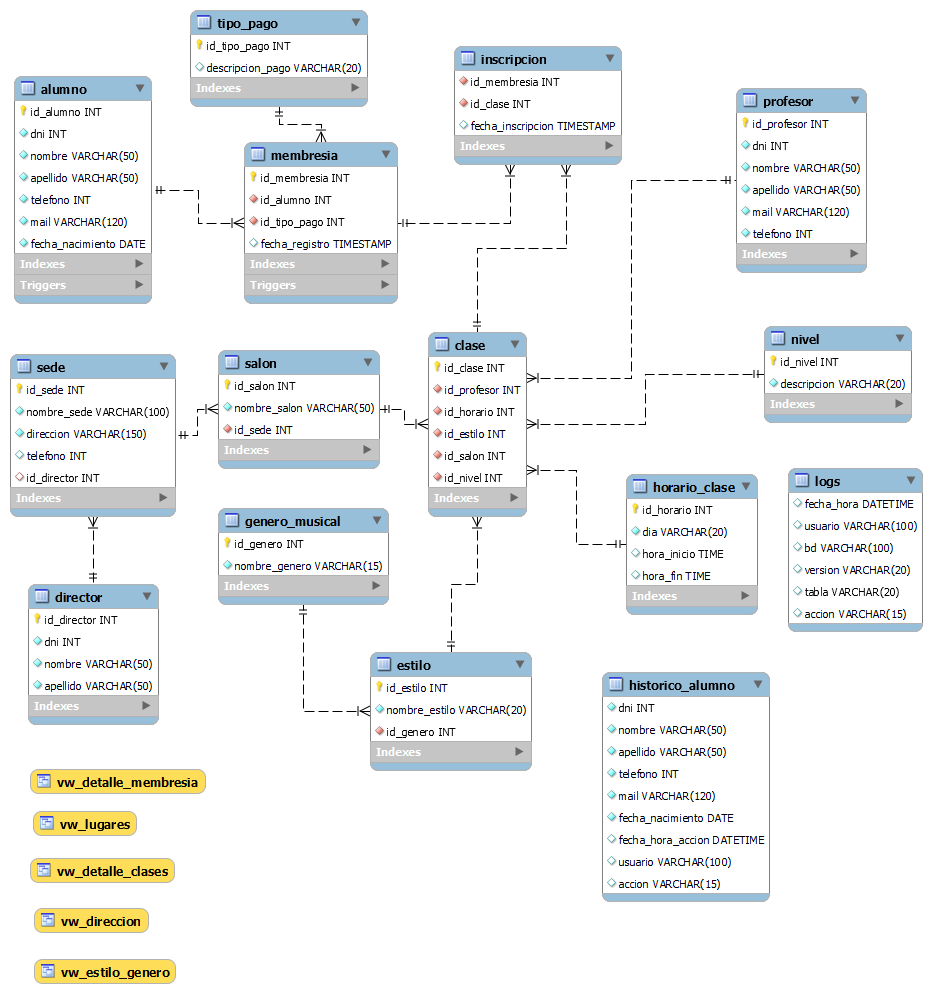
La Academia Mambolaye es una escuela de baile que se dedica a enseñar diversos estilos de baile como Mambo (Salsa on2), Mambo Lady Style, Bachata Estilo Pacheco y próximamente se irán agregando más estilos.

Mambolaye se fundó en Venezuela y actualmente cuenta con un solo espacio de enseñanza en Buenos Aires, con miras de expansión. El nombre de la academia está compuesto por dos palabras **Mambo** que es el estilo de baile y **Laye** que significa calle, es decir, el estilo de baile con un toque propio de la persona que solo se obtiene fuera de salones de baile.

**Descripción de tablas:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **alumno (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_alumno | int | autoinc | x | x |  | Datos de los alumnos de la academia |
| dni | int |  | x |  |  |
| nombre\_alumno | varchar | 50 | x |  |  |
| apellido\_alumno | varchar | 50 | x |  |  |
| telefono | int |  | x |  |  |
| mail | varchar | 120 | x |  |  |
| fecha\_nacimiento | date | 10 | x |  |  |
|  |  |  |  |  |  |  |
| **tipo\_pago (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_tipo\_pago | int | autoinc | x | x |  | Informacion de los metodos de pago |
| descripcion\_pago | varchar | 20 |  |  |  |
|  |  |  |  |  |  |  |
| **Membresia (TT)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_membresia | int | autoinc | x | x |  | Datos de la membresia de cada alumno |
| id\_alumno | int |  | x |  | x |
| id\_tipo\_pago | int |  | x |  | x |
| fecha\_registro | Timestamp |  | x |  | x |
|  |  |  |  |  |  |  |
| **profesor (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NUL** | **PK** | **FK** | **Descripcion** |
| id\_profesor | int | autoinc | x | x |  | Contiene informacion de los profesores de baile |
| dni | int |  | x |  |  |
| nombre | varchar | 50 | x |  |  |
| apellido | varchar | 50 | x |  |  |
| mail | varchar | 120 | x |  |  |
| telefono | int |  | x |  |  |
|  |  |  |  |  |  |  |
| **genero\_musical (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_genero | int | autoinc | x | x |  | Descripcion del genero musical |
| nombre\_genero | varchar | 15 | x |  |  |
|  |  |  |  |  |  |  |
| **estilo (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_estilo | int | autoinc | x | x |  | Estilos de baile relacionados con los generos musicales |
| nombre\_estilo | varchar | 20 | x |  |  |
| id\_genero | int |  | x |  | x |
|  |  |  |  |  |  |  |
| **horario\_clase (TD)** | | | | | | |
| **Campos** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_horario | int | autoinc | x | x |  | Tabla de horarios de clases |
| dia | varchar | 20 | x |  |  |
| hora\_inicio | Time |  |  |  |  |
| hora\_fin | time |  |  |  |  |
|  |  |  |  |  |  |  |
| **director(TD)** | | | | | | |
| **Campos** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_director | int | autoinc | x | x |  | Datos de los directores de las sedes |
| dni | int |  | x |  |  |
| nombre | varchar | 50 | x |  |  |
| apellido | varchar | 50 | x |  |  |
|  |  |  |  |  |  |  |
| **sede (TD)** | | | | | | |
| **Campos** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_sede | int | autoinc | x | x |  | Contiene informacion de la(s) sede(s) de la academia |
| nombre\_sede | varchar | 100 | x |  |  |
| direccion | varchar | 150 | x |  |  |
| telefono | int |  |  |  |  |
| id\_director | int |  |  |  | x |
|  |  |  |  |  |  |  |
| **nivel (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_genero | int | autoinc | x | x |  | Niveles de baile |
| descripcion | varchar | 20 | x |  |  |
|  |  |  |  |  |  |  |
| **salon (TD)** | | | | | | |
| **Campos** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_salon | int | autoinc | x | x |  | Contiene datos de salon de practica |
| nombre\_salon | varchar | 50 | x |  |  |
| id\_sede | int |  | x |  | x |
|  |  |  |  |  |  |  |
| **clase (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_clase | int | autoinc | x | x |  | Contiene informacion sobre las clases que brinda la academia |
| id\_profesor | int |  | x |  | x |
| id\_horario | int |  | x |  | x |
| id\_estilo | int |  | x |  | x |
| id\_salon | int |  | x |  | x |
| id\_nivel | int |  | x |  | x |
|  |  |  |  |  |  |  |
| **inscripcion (TH)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_registro | int | autoinc | x | x |  | Contiene datos de la inscripcion de cada alumno |
| fecha\_registro | datetime | 16 | x |  |  |
| tipo\_pago | varchar | 20 | x |  |  |
| id\_clase | int |  | x |  | x |
| id\_alumno | int |  | x |  | x |
|  |  |  |  |  |  |  |
| **logs (TT)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| fecha\_hora | datetime | autoinc | x | x |  | Contiene informacion sobre las clases que brinda la academia |
| usuario | varchar | 100 | x |  | x |
| bd | varchar | 100 | x |  | x |
| version | varchar | 20 | x |  | x |
| tabla | varchar | 20 | x |  | x |
| accion | varchar | 15 | x |  | x |
|  |  |  |  |  |  |  |
| **historico\_alumnos (TT)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| dni | int |  | x |  |  | Contiene informacion sobre las clases que brinda la academia |
| nombre | varchar | 50 | x |  |  |
| apellido | varchar | 50 | x |  |  |
| telefono | int |  | x |  |  |
| mail | varchar | 120 | x |  |  |
| fecha\_nacimiento | date |  | x |  |  |
| fecha\_hora\_accion | datetime |  | x |  |  |
| usuario | varchar | 100 | x |  |  |
| accion | varchar | 15 | x |  |  |

Diagrama ER



**Script SQL creación Schema:**

*#Habilitar eliminación y actualización en cascada*

*SET sql\_safe\_updates=0;*

*CREATE SCHEMA academia\_mambolaye;*

*USE academia\_mambolaye;*

*-- ALUMNOS*

*CREATE TABLE IF NOT EXISTS alumno(*

*id\_alumno INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*dni INT NOT NULL,*

*nombre VARCHAR(50) NOT NULL,*

*apellido VARCHAR(50) NOT NULL,*

*telefono INT NOT NULL,*

*mail VARCHAR(120) NOT NULL,*

*fecha\_nacimiento DATE NOT NULL*

*);*

*-- TIPO DE PAGO*

*CREATE TABLE IF NOT EXISTS tipo\_pago(*

*id\_tipo\_pago INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*descripcion\_pago VARCHAR(20)*

*);*

*-- MEMBRESIA*

*CREATE TABLE IF NOT EXISTS membresia(*

*id\_membresia INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*id\_alumno INT NOT NULL,*

*id\_tipo\_pago INT NOT NULL,*

*fecha\_registro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,*

*FOREIGN KEY (id\_alumno) REFERENCES academia\_mambolaye.alumno(id\_alumno) ON DELETE CASCADE,*

*FOREIGN KEY (id\_tipo\_pago) REFERENCES academia\_mambolaye.tipo\_pago(id\_tipo\_pago) ON DELETE CASCADE*

*);*

*-- PROFESOR*

*CREATE TABLE IF NOT EXISTS profesor(*

*id\_profesor INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*dni INT NOT NULL,*

*nombre VARCHAR(50) NOT NULL,*

*apellido VARCHAR(50) NOT NULL,*

*mail VARCHAR(120) NOT NULL,*

*telefono INT NOT NULL*

*);*

*-- GENERO MUSICAL*

*CREATE TABLE IF NOT EXISTS genero\_musical(*

*id\_genero INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*nombre\_genero VARCHAR(15) NOT NULL*

*);*

*-- ESTILO DE BAILE*

*CREATE TABLE IF NOT EXISTS estilo(*

*id\_estilo INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*nombre\_estilo VARCHAR(20) NOT NULL,*

*id\_genero INT NOT NULL,*

*FOREIGN KEY (id\_genero) REFERENCES academia\_mambolaye.genero\_musical(id\_genero)*

*);*

*-- HORARIOS DE CLASES*

*CREATE TABLE IF NOT EXISTS horario\_clase(*

*id\_horario INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*dia VARCHAR(20) NOT NULL,*

*hora\_inicio TIME,*

*hora\_fin TIME*

*);*

*-- DIRECTORES ACADEMIA*

*CREATE TABLE IF NOT EXISTS director(*

*id\_director INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*dni INT NOT NULL,*

*nombre VARCHAR(50) NOT NULL,*

*apellido VARCHAR(50) NOT NULL*

*);*

*-- SEDES*

*CREATE TABLE IF NOT EXISTS sede(*

*id\_sede INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*nombre\_sede VARCHAR(100) NOT NULL,*

*direccion VARCHAR(150) NOT NULL,*

*telefono INT,*

*id\_director INT,*

*FOREIGN KEY (id\_director) REFERENCES director(id\_director)*

*);*

*-- NIVEL DE BAILE*

*CREATE TABLE IF NOT EXISTS nivel(*

*id\_nivel INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*descripcion VARCHAR(20) NOT NULL*

*);*

*-- SALONES*

*CREATE TABLE IF NOT EXISTS salon(*

*id\_salon INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*nombre\_salon VARCHAR(50) NOT NULL,*

*id\_sede INT NOT NULL,*

*FOREIGN KEY (id\_sede) REFERENCES academia\_mambolaye.sede(id\_sede)*

*);*

*-- CLASE*

*CREATE TABLE IF NOT EXISTS clase(*

*id\_clase INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,*

*id\_profesor INT NOT NULL,*

*id\_horario INT NOT NULL,*

*id\_estilo INT NOT NULL,*

*id\_salon INT NOT NULL,*

*id\_nivel INT NOT NULL,*

*FOREIGN KEY (id\_profesor) REFERENCES academia\_mambolaye.profesor(id\_profesor),*

*FOREIGN KEY (id\_horario) REFERENCES academia\_mambolaye.horario\_clase(id\_horario),*

*FOREIGN KEY (id\_estilo) REFERENCES academia\_mambolaye.estilo(id\_estilo),*

*FOREIGN KEY (id\_salon) REFERENCES academia\_mambolaye.salon(id\_salon),*

*FOREIGN KEY (id\_nivel) REFERENCES academia\_mambolaye.nivel(id\_nivel)*

*);*

*-- INSCRIPCION*

*CREATE TABLE IF NOT EXISTS inscripcion(*

*id\_membresia INT NOT NULL,*

*id\_clase INT NOT NULL,*

*fecha\_inscripcion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,*

*FOREIGN KEY (id\_membresia) REFERENCES academia\_mambolaye.membresia(id\_membresia) ON DELETE CASCADE,*

*FOREIGN KEY (id\_clase) REFERENCES academia\_mambolaye.clase(id\_clase)*

*);*

*#LOGS*

*CREATE TABLE IF NOT EXISTS logs(*

*fecha\_hora DATETIME,*

*usuario VARCHAR(100),*

*bd VARCHAR(100),*

*version VARCHAR(20),*

*tabla VARCHAR(20),*

*accion VARCHAR(15)*

*);*

*#BITACORA\_ALUMNOS*

*CREATE TABLE IF NOT EXISTS historico\_alumno(*

*dni INT NOT NULL,*

*nombre VARCHAR(50) NOT NULL,*

*apellido VARCHAR(50) NOT NULL,*

*telefono INT NOT NULL,*

*mail VARCHAR(120) NOT NULL,*

*fecha\_nacimiento DATE NOT NULL,*

*fecha\_hora\_accion DATETIME,*

*usuario VARCHAR(100),*

*accion VARCHAR(15)*

*);*

**Vistas de la BD**

**vw\_direccion**

**Objetivo**: observar las sedes de la academia y sus respectivos directores

**Tablas**: sede y director

**Script SQL:**

*create view academia\_mambolaye.vw\_direccion as*

*select a.nombre\_sede as Sede*

*,concat(b.nombre,' ', b.apellido) as Director*

*from sede a*

*join director b*

*on (a.id\_director = b.id\_director);*

**vw\_ estilo\_genero**

**Objetivo:** Debido que los estilos de baile pueden variar para un genero musical, mediante esta vista se puede obtener el genero y sus distintos tipos de estilo

**Tablas:** estilo y genero\_musical

**Script SQL:**

*create view academia\_mambolaye.vw\_estilo\_genero as*

*SELECT est.nombre\_estilo as 'Estilo de Baile',*

*gen.nombre\_genero as 'Genero Musical'*

*from estilo est*

*join genero\_musical gen*

*on (est.id\_genero = gen.id\_genero);*

**vw\_detalle\_clase**

**Objetivo**: La tabla clase esta conformada por distintos ID que pertenecen a otras tablas, esto a la vista humana es poco entendible, por ende esta view asocia los distintos ID con su respectiva tablas trayendo la descripción de los mismos y así la información sea mas acorde al usuario.

**Tablas**: clase, estilo, profesor, horario\_clas, nivel y salón

**Script SQL**:

*create view academia\_mambolaye.vw\_detalle\_clases as*

*select cla.id\_clase*

*,est.nombre\_estilo as 'Estilo Baile'*

*,concat(pro.nombre, ' ',pro.apellido) as 'Profesor(a)'*

*,concat(hc.dia, ' ', hc.hora\_inicio, ' ', hc.hora\_fin) as 'Horario'*

*,nv.descripcion 'Nivel'*

*,sl.nombre\_salon 'Salon'*

*from clase cla*

*join estilo est*

*on (cla.id\_estilo = est.id\_estilo)*

*join profesor pro*

*on (cla.id\_profesor = pro.id\_profesor)*

*join horario\_clase hc*

*on (cla.id\_horario = hc.id\_horario)*

*join nivel nv*

*on (cla.id\_nivel = nv.id\_nivel)*

*join salon sl*

*on (cla.id\_salon = sl.id\_salon);*

**vw\_lugares**

**Objetivo:** Observar cuales son los salones que están en cada sede

**Tablas:** salón y sede

**Script SQL:**

*create view academia\_mambolaye.vw\_lugares as*

*select sed.nombre\_sede*

*,sed.direccion*

*,sed.telefono*

*,sal.nombre\_salon as Salon*

*from salon sal*

*join sede sed*

*on (sal.id\_sede = sed.id\_sede);*

**vw\_membresia**

**Objetivo:** Observar toda la información de los alumnos que han obtenido una membresía y también con que método de pago

**Tablas:** alumno, membresia y tipo\_pago

**Script SQL:**

*create view academia\_mambolaye.vw\_detalle\_membresia as*

*select al.\**

*,pay.descripcion\_pago*

*from alumno al*

*join membresia mem*

*on (mem.id\_alumno = al.id\_alumno)*

*join tipo\_pago pay*

*on (mem.id\_tipo\_pago = pay.id\_tipo\_pago)*

*order by al.id\_alumno asc;*

**Funciones**

**cant\_alumnos\_x\_clase**

**Objetivo:** Contabilizar cuantos alumnos están inscritos en una clase especifica indicando el id de la clase. Se ingresa un entero que corresponde al ID de la clase a consultar y la función realiza un select count de la tabla inscripción.

**Script SQL:**

*DELIMITER $$*

*CREATE DEFINER=`root`@`localhost` FUNCTION `cant\_alumnos\_x\_clase`(p\_id\_clase INT) RETURNS int*

*DETERMINISTIC*

*BEGIN*

*DECLARE resultado INT;*

*set resultado = (select count(\*) from inscripcion where id\_clase=p\_id\_clase);*

*RETURN resultado;*

*END$$*

*DELIMITER ;*

**ult\_alumno\_x\_clase**

**Objetivo:** Retornar el nombre del último alumno que se registró a una clase en especifica. Se ingresa un entero que corresponde al ID de la clase a consultar y la función retornar una cadena de texto (varchar) con el nombre de la persona.

**Script SQL:**

*DELIMITER $$*

*CREATE DEFINER=`root`@`localhost` FUNCTION `ult\_alumno\_x\_clase`(p\_id\_clase INT) RETURNS varchar(255)*

*READS SQL DATA*

*BEGIN*

*DECLARE resultado varchar(255);*

*set resultado = (select max(al.nombre) from inscripcion ins join membresia mem on (ins.id\_membresia = mem.id\_membresia)*

*join alumno al on (al.id\_alumno = mem.id\_alumno) join clase cla on (ins.id\_clase = cla.id\_clase) where ins.id\_clase = p\_id\_clase);*

*RETURN resultado;*

*END$$*

*DELIMITER ;*

**Stored Procedures**

**sp\_orden\_x\_campo**

**Objetivo:** Mostrar datos ordenados de una tabla indicada por el usuario, esto permitirá al usuario que trabaje directamente en la BD ahorrar tiempo y código de sentencia ya que al llamar al SP solo deberá ingresar el nombre de la tabla, la columna y si es en orden ascendente o descendente.

**Script SQL:**

*drop procedure if exists `sp\_orden\_x\_campo`;*

*DELIMITER $$*

*CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_orden\_x\_campo`(IN p\_table VARCHAR(100), IN p\_field VARCHAR(100), IN p\_order INT)*

*BEGIN*

*IF p\_field <> '' THEN*

*SET @field = CONCAT(' ORDER BY ', p\_field);*

*ELSE*

*SET @field = CONCAT(' ORDER BY 1');*

*END IF;*

*IF p\_order = 1 THEN*

*SET @ord = CONCAT(' ASC');*

*ELSE*

*SET @ord = CONCAT(' DESC');*

*END IF;*

*SET @sentence = CONCAT('SELECT \* FROM academia\_mambolaye.',p\_table, @field, @ord);*

*PREPARE sentenceSQL FROM @sentence;*

*EXECUTE sentenceSQL;*

*DEALLOCATE PREPARE sentenceSQL;*

*END$$*

*DELIMITER ;*

**sp\_delete\_alumno**

**Objetivo:** Eliminar alumno enviando por parámetro el DNI, el SP ayuda a evitar la posibilidad de eliminar registros por equivocación y omisión de la clausula where. Cuenta con una validación de longitud por si el usuario no ingresa por completo la secuencia de números que conforman un DNI.

**Script SQL:**

*drop procedure if exists `sp\_delete\_alumno`;*

*DELIMITER $$*

*CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_delete\_alumno`(IN p\_dni INT)*

*BEGIN*

*IF length(p\_dni) <> 8 then*

*SELECT 'ERROR: no se pudo eliminar el registro indicado';*

*ELSE*

*DELETE FROM academia\_mambolaye.alumno WHERE dni = p\_dni;*

*END IF;*

*END$$*

*DELIMITER ;*