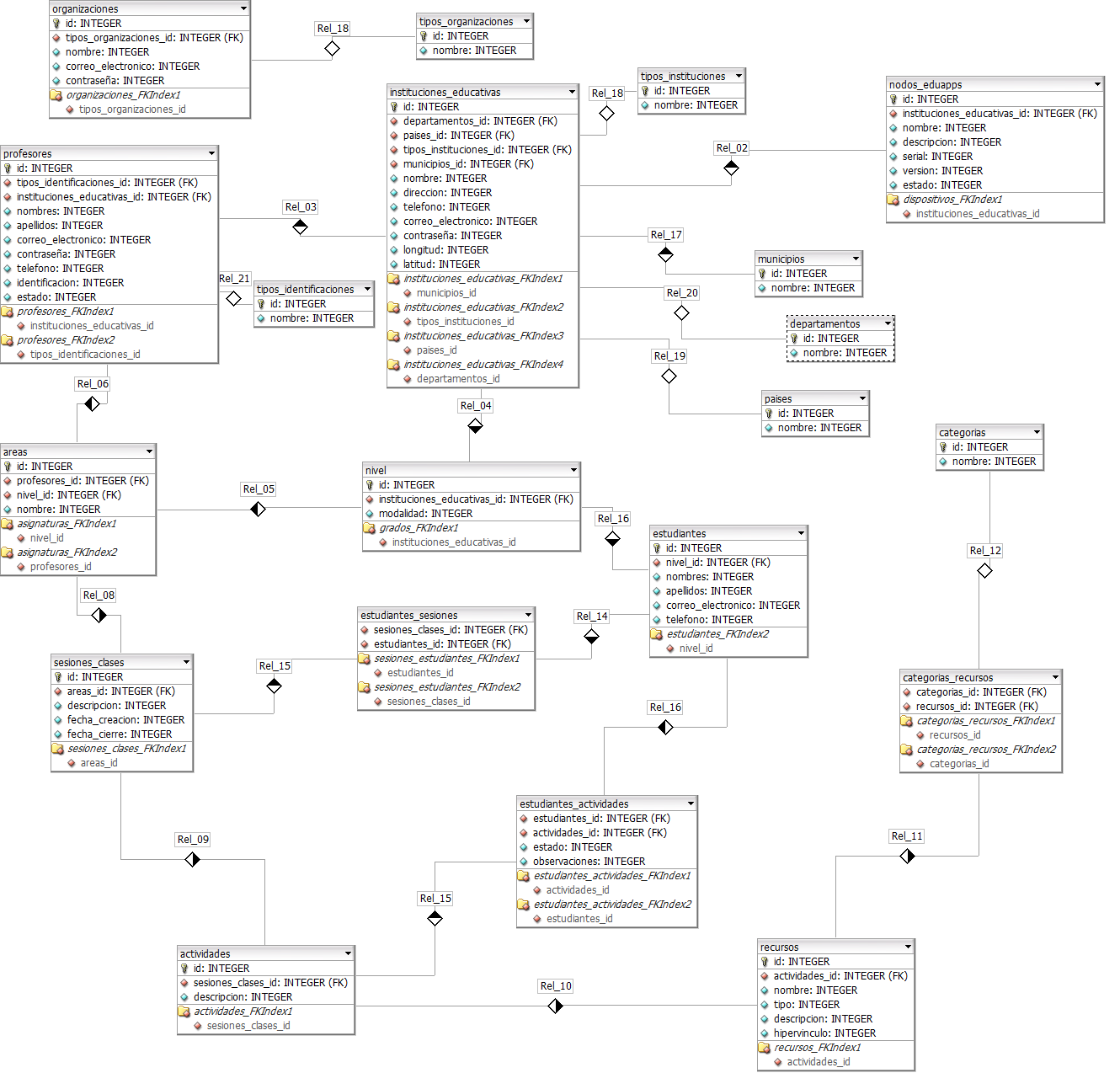
**Juan Camilo Restrepo Velez – 000373886**

**1. Modelo de datos**

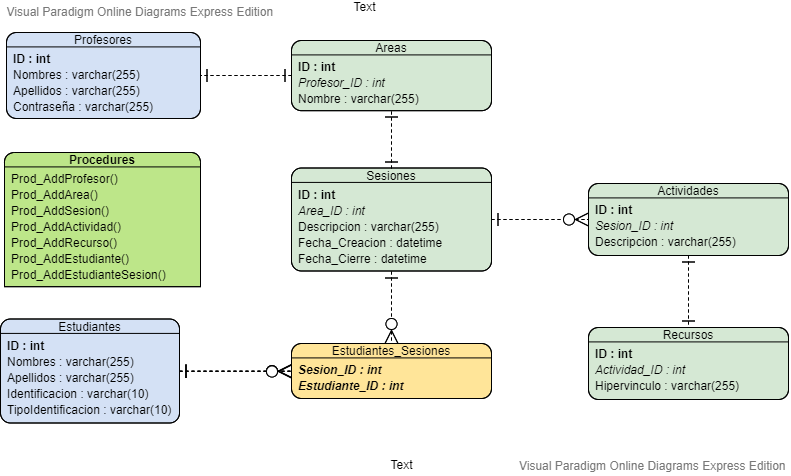
Para definir el modelo de datos que se utilizarán en el nodo Raspberry se retomará el modelo general que estará en el servidor de la plataforma que es el siguiente:



De la cual solo se requerirán las siguientes tablas:

* Profesores y Áreas que se requiere para relacionar con la tabla Sesión
* Sesiones
* Actividades
* Recursos que en el atributo hipervínculo estará la ruta donde se encuentra en el servidor SFTP
* Estudiantes y las tablas para relacionarse con Sesiones

Además, procedimientos almacenados para insertar y obtener, de una forma más sencilla, realizar las acciones que se necesitan. El modelo que queda es el siguiente:



Este modelo será implementado en el servidor de base de datos MariaDB que se instaló en el nodo.

**2. MariaDB en Raspberry Pi 4**

Una vez tenida el servidor de base de datos instalado en el Raspberry se procede a crear la base de datos y las tablas que se necesitarán (apartado 1.). Aunque lo normal sería realizar estas acciones desde uno de los clientes que tiene MariaDB.

sudo mysql -u root -p -h localhost

**Crear un usuario:**

CREATE USER 'admin' @'localhost' IDENTIFIED BY '1234';

**Crear un base de datos:**

CREATE DATABASE EduApps;

Otorgar permisos al usuario previamente creado

GRANT ALL PRIVILEGES ON EduApps.\* TO 'admin'@'localhost';

FLUSH PRIVILEGES;

Reiniciar para que los cambios de apliquen

sudo service mariadb restart

**Crear las tablas**

USE EduApps;

CREATE TABLE Profesores

    (ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

     Nombres VARCHAR(255) NOT NULL,

     Apellido VARCHAR(255) NOT NULL,

     Contraseña VARCHAR(255) NOT NULL);

CREATE TABLE Areas

    (ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY NOT NULL,

     Profesor\_ID INT NOT NULL,

     Nombre VARCHAR(255) NOT NULL,

     CONSTRAINT ‘FKProsefor\_Area’

     FOREIGN KEY (Profesor\_ID) REFERENCES Profesores(ID)

      ON DELETE CASCADE

      ON UPDATE RESTRICT);

CREATE TABLE Sesiones

    (ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

     Area\_ID INT NOT NULL,

     Descripcion VARCHAR(255) NOT NULL,

     Fecha\_Creacion DATETIME NOT NULL,

     Fecha\_Cierre DATETIME NOT NULL,

     CONSTRAINT ‘FKArea\_Sesion’

     FOREIGN KEY (Area\_ID) REFERENCES Areas(ID)

      ON DELETE CASCADE

      ON UPDATE RESTRICT);

CREATE TABLE Actividades

    (ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

     Sesion\_ID INT NOT NULL,

     Descripcion VARCHAR(255) NOT NULL,

     CONSTRAINT ‘FKSesion\_Actividad’

     FOREIGN KEY (Sesion\_ID) REFERENCES Sesiones(ID)

      ON DELETE CASCADE

      ON UPDATE RESTRICT);

CREATE TABLE Recursos

    (ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

     Actividad\_ID INT NOT NULL,

     Hipervinculo VARCHAR(255) NOT NULL,

     CONSTRAINT ‘FKActividad\_Recurso’

     FOREIGN KEY (Actividad\_ID) REFERENCES Actividades(ID)

      ON DELETE CASCADE

      ON UPDATE RESTRICT);

CREATE TABLE Estudiante

    (ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

     Nombres VARCHAR(255) NOT NULL,

     Apellido VARCHAR(255) NOT NULL,

     Identificacion VARCHAR(10) NOT NULL,

     TipoIdentificacion VARCHAR(10) NOT NULL,

     UNIQUE (Identificacion));

CREATE TABLE Estudiantes\_Sesiones

    (Estudiante\_ID INT NOT NULL,

     Sesion\_ID INT NOT NULL,

     PRIMARY KEY (Estudiante\_ID, Sesion\_ID),

     CONSTRAINT ‘FKEstudiante\_Sesion’

     FOREIGN KEY (Estudiante\_ID) REFERENCES Estudiantes(ID)

      ON DELETE CASCADE

      ON UPDATE RESTRICT,

     CONSTRAINT ‘FKSesion\_Estudiante’

     FOREIGN KEY (Sesion\_ID) REFERENCES Sesiones(ID)

      ON DELETE CASCADE

      ON UPDATE RESTRICT);

**Procedimientos almacenados**

DELIMITER $$

CREATE PROCEDURE Prod\_AddProfesor(IN nombre VARCHAR(255), IN apellido VARCHAR(255), IN contraseña VARCHAR(255))

BEGIN

INSERT INTO Profesores VALUES (nombre,apellido,contraseña);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE Prod\_AddArea(IN profesor\_id INT, IN nombre VARCHAR(255))

BEGIN

INSERT INTO Areas VALUES (professor\_id,nombre);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE Prod\_AddSesion(IN area\_id INT, IN descripcion VARCHAR(255), IN fecha\_crea DATETIME, IN fecha\_cier DATETIME)

BEGIN

INSERT INTO Sesiones VALUES (área\_id, descripcion,fecha\_crea,fecha\_cier);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE Prod\_AddActividad(IN sesion\_id INT, IN descripcion VARCHAR(255))

BEGIN

INSERT INTO Actividades VALUES (sesión\_id, descripcion);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE Prod\_AddRecurso(IN actividad\_id INT, IN path VARCHAR(255))

BEGIN

INSERT INTO Recursos VALUES (actividad\_id,path);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE Prod\_AddEstudiante(IN nombre VARCHAR(255), IN apellido VARCHAR(255), IN identificacion VARCHAR(10), IN tipo VARCHAR(10))

BEGIN

INSERT INTO Estudiante VALUES(nombre,apellido,identificacion,tipo);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE Prod\_AddEstudianteSesion(IN estudiante\_id INT, IN sesión\_id INT)

BEGIN

INSERT INTO Estudiantes\_Sesiones VALUES(estudiante\_id,sesión\_id);

END$$

DELIMITER ;

**Funciones**

DELIMITER $$

CREATE FUNCTION Prod\_RdEstudianteSesion(@estudiante\_id INT)

RETURNS TABLE

AS

RETURN

(

SELECT Sesion\_ID

FROM Estudiantes\_Sesiones

WHERE estudiante\_id = Estudiante\_ID;

);

GO$$

DELIMITER ;

DELIMITER $$

CREATE FUNCTION Prod\_RdActividadSesion(@sesion\_id INT)

RETURNS TABLE

AS

RETURN

(

SELECT Actividad\_ID

FROM Actividades

WHERE sesion\_id = Sesion\_ID;

);

GO$$

DELIMITER ;

DELIMITER $$

CREATE FUNCTION Prod\_RdRecursoActividad(@actividad\_id INT)

RETURNS TABLE

AS

RETURN

(

SELECT Hipervinculo

FROM Recursos

WHERE actividad \_id = Actividad \_ID;

);

GO$$

DELIMITER ;

**Insertar datos iniciales**

Se reutilizan los procedimientos almacenados

CALL Prod\_AddProfesor(‘Oscar’,‘Sanchez’,‘1234’);

CALL Prod\_AddProfesor(‘Wilder’,‘Valencia’,‘1234’);

CALL Prod\_AddArea(1,‘Admin’);

CALL Prod\_AddArea(2,‘Admin’);

CALL Prod\_AddSesion(1, ‘Sesion creada por Oscar’,‘2020-05-1’,‘2020-06-1’);

CALL Prod\_AddSesion(2, ‘Sesion creada por Wilder’,‘2020-05-1’,‘2020-06-1’);

CALL Prod\_AddActividad(1, ‘Actividad 1 de la sesion de Oscar’);

CALL Prod\_AddActividad(1, ‘Actividad 2 de la sesion de Oscar’);

CALL Prod\_AddActividad(2, ‘Actividad 1 de la sesion de Wilder’);

CALL Prod\_AddRecursos(1,‘/home/pi/Oscar/sesion1/actividad1/archivos.zip’);

CALL Prod\_AddRecursos( 2,‘/home/pi/Oscar/sesion1/actividad2/archivos.zip’);

CALL Prod\_AddRecursos( 3,‘/home/pi/Wilder/sesion1/actividad1/archivos.zip’);

CALL Prod\_AddEstudiante(‘Estudiante’,‘Estudiante’,‘1234’, ‘TI’);

CALL Prod\_AddEstudianteSesion(1,1);

CALL Prod\_AddEstudianteSesion(1,2);

**Nota:** el LogIn se realiza con el ID y la contraseña

**Referencias**

Llama, L. Cómo instalar MariaDB en Raspberry Pi. Luis Llamas. Retrieved 5 April 2020, from <https://www.luisllamas.es/como-instalar-mariadb-en-raspberry-pi>

Llama, L. Cómo instalar MySQL en Raspberry Pi. Luis Llamas. Retrieved 5 April 2020, from <https://www.luisllamas.es/como-instalar-mysql-en-raspberry-pi/>

Foreign Keys. MariaDB KnowledgeBase. Retrieved 5 April 2020, from <https://mariadb.com/kb/en/foreign-keys/>

Claves Foráneas MariaDB — documentación de Apuntes Informatica - 1.0.0. Apuntes-snicoper.readthedocs.io. Retrieved 5 April 2020, from <https://apuntes-snicoper.readthedocs.io/es/latest/programacion/mariadb/claves_foraneas_mariadb.html>

INSERT. MariaDB KnowledgeBase. Retrieved 5 April 2020, from <https://mariadb.com/kb/en/insert/>

Cómo crear y usar un procedimiento almacenado MySQL. Neoguias. Retrieved 6 April 2020, from <https://www.neoguias.com/procedimientos-almacenados-mysql/>

CREATE FUNCTION (Transact-SQL) - SQL Server. Docs.microsoft.com. Retrieved 6 April 2020, from <https://docs.microsoft.com/es-es/sql/t-sql/statements/create-function-transact-sql?view=sql-server-ver15>