

## Proyecto de Sistemas de Bases de Datos I

*BASE DE DATOS DE LA ACADEMIA DE ARTES MARCIALES "XIAO HU YAN"*

**Sistemas de Bases de Datos I  
Segundo Término 2018-2019**

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL  
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

<b>Materia:</b> Sistemas de Bases de Datos 1	<b>Facultad de Ingeniería en Electricidad y Computación</b>
<b>Profesor:</b> Ing. Gustavo Irving Cali Mena	<b>ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL</b>
<b>Última Revisión:</b> Fecha	<b>Pág. 1 de 17</b>

## Índice

Integrantes.....	3
Título del Proyecto.....	3
Objetivo del Proyecto .....	3
Objetivos Específicos.....	3
Descripción General.....	3
Modelo Conceptual.....	5
Diccionario de Datos.....	6
Modelo Lógico.....	10
Flujo de Navegación .....	11
Modelo Físico .....	13
Anexos .....	17

## Integrantes

- ALBAN MAYANCELA HUGO JAVIER
- CEDEÑO VINCES ANDRES SEBASTIAN
- CEVALLOS SALAS CARLOS XAVIER
- INTRIAGO JURADO PAUL MAURICIO
- JARA YUPA ANA BELEN

## Título del Proyecto

*BASE DE DATOS DE LA ACADEMIA DE ARTES MARCIALES "XIAO HU YAN"*  
***Etapas de Desarrollo: Análisis del Sistema***

## Objetivo del Proyecto

Diseñar una base de datos para la academia "XIAO HU YAN" con el sistema MySQL basado en el modelo relacional para la consistencia y accesibilidad de la información de la institución.  
**Etapas de Planificación**

## Objetivos Específicos

- Identificar las entidades, relaciones, atributos, y restricciones de la academia para elaborar la estructura del modelo Conceptual.
- Convertir en tablas todos los tipos de entidades y relaciones descritos en el modelo conceptual para la comprensión del esquema a utilizarse en la base de datos.
- Crear la estructura de la base de datos del sistema usando el lenguaje SQL.

**Etapas de Planificación**

## Descripción General

Crear una base de datos para una academia "XIAO HU YAN" de artes marciales, su administrador consideró necesario el uso tecnológico para solventar ciertas falencias que tiene en su institución.

Los empleados pueden ser profesores o de mantenimiento o de administración. Se quiere registrar, su cédula, su nombre completo, su fecha de nacimiento, su edad, su correo electrónico, su teléfono de contacto y dirección de domicilio.

Actualmente cuenta con cinco profesores cinturones negros. Cada profesor tiene su horario de clases. El horario de clases depende de cada profesor para completar sus 10 horas semanales.

Cada alumno debe dar sus nombres completos, su edad y su cédula de identidad para poder ser matriculado en la academia. Cada alumno tiene que registrar su día de pago de mensualidad, sin embargo, si entrenan como familia es decir más de dos personas existe un descuento en el pago final.

Cada estudiante se registra a las clases de los profesores, para la clase se registra la fecha de la clase, si asistió a clases y los ejercicios realizados.

<b>Materia:</b> Sistemas de Bases de Datos 1	<b>Facultad de Ingeniería en Electricidad y Computación</b>
<b>Profesor:</b> Ing. Gustavo Irving Cali Mena	<b>ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL</b>
<b>Última Revisión:</b> Fecha	<b>Pág. 3 de 17</b>

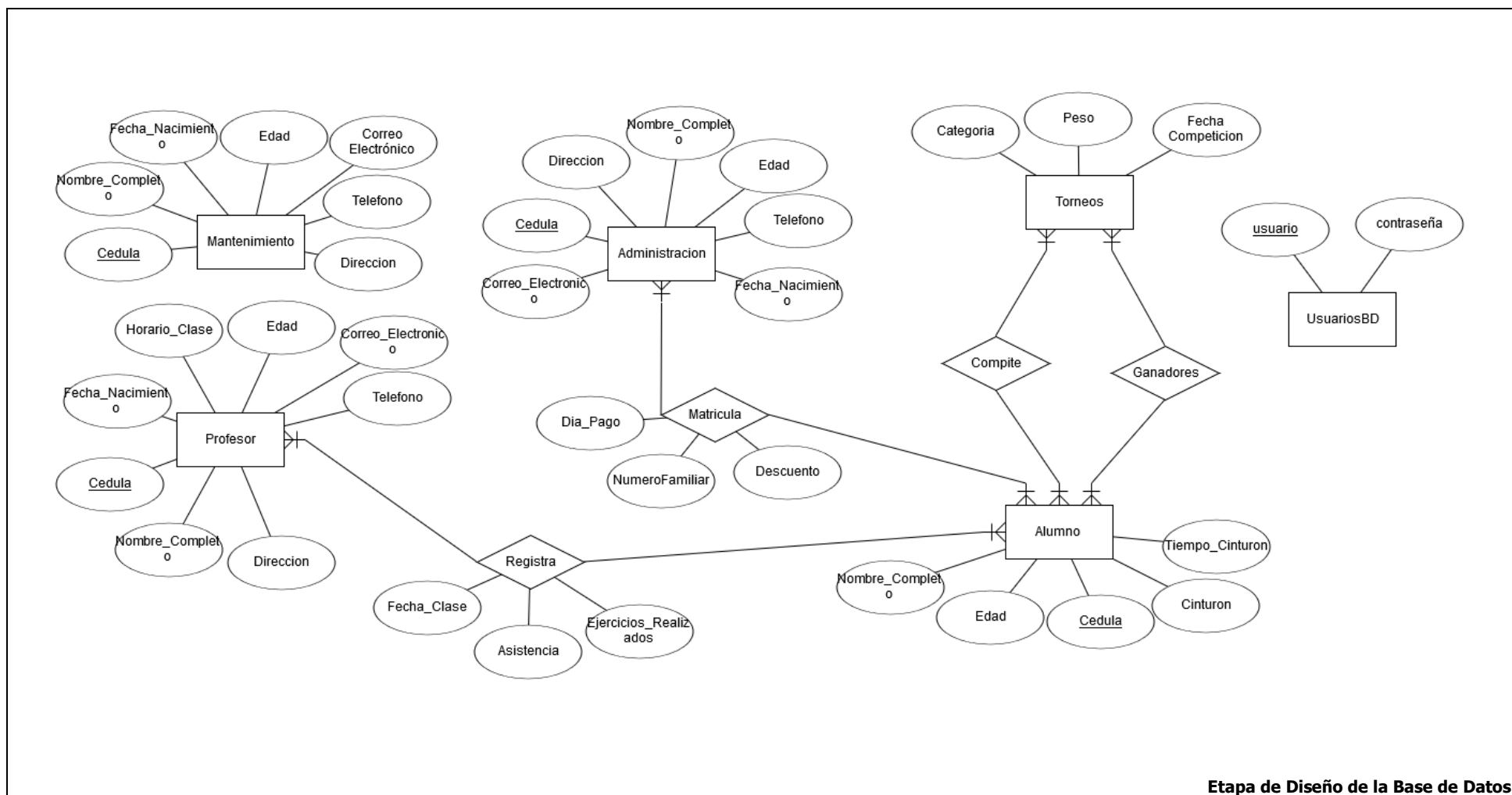
La academia segmenta sus alumnos de acuerdo con su grado de cinturón, siendo el de menor rango blanco, seguido por amarillo, amarillo verde, verde, azul, azul rojo, rojo, rojo negro y negro.

La academia necesita conocer cuánto tiempo lleva un alumno con su cinturón y así conocer cuándo puede presentarse a dar examen. De la misma forma la academia compite en torneos nacionales, obteniendo trofeos, los mismos que tienen categoría, peso, nombre de alumnos competidores y ganadores, además de la fecha de la competición.

Finalmente se cuenta con dos empleados más uno que realiza el mantenimiento y otro que realiza la administración de la academia.

**Etapas de Definición del Sistema**

## Modelo Conceptual



Etapas de Diseño de la Base de Datos

<b>Materia:</b> Sistemas de Bases de Datos 1	<b>Facultad de Ingeniería en Electricidad y Computación</b>
<b>Profesor:</b> Ing. Gustavo Irving Cali Mena	<b>ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL</b>
<b>Última Revisión:</b> Fecha	<b>Pág. 5 de 17</b>

## Diccionario de Datos

<b>Tabla: Profesor</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Cedula	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de un profesor
Nombre_Completo	varchar(30)	Contiene todas las letras del alfabeto	Es el nombre completo del profesor
Fecha_nacimiento	date	fecha	Fecha del día de nacimiento del docente
Cinturon	int	Contiene todos los int posibles	Id del cinturon
Correo_Electronico	varchar(30)	Contiene todas las caracteres posibles	Es el correo electrónico del docente
Telefono	int	Contiene todos los int posibles	Teléfono del docente
Direccion	varchar(30)	Contiene todas las caracteres posibles	Es la dirección del docente

<b>Tabla: Administracion</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Cedula	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de una persona
Correo_Electronico	varchar(30)	Contiene todas las caracteres posibles	Es el correo electrónico del alumno
Direccion	varchar(30)	Contiene todas las caracteres posibles	Es la dirección del alumno registrado
Nombre_Completo	varchar(30)	Contiene todas las letras del alfabeto	Es el nombre completo del alumno registrado
Telefono	int	Contiene todos los int posibles	Teléfono del alumno registrado
Fecha_Nacimiento	date	Fecha	Fecha del día de nacimiento del alumno registrado

<b>Tabla:Alumno</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Cedula	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de una persona
Nombre_Completo	varchar(30)	Contiene todas las letras del alfabeto	Es el nombre completo del alumno que se registra en la academia
Edad	int	Contiene todos los int posibles	Es la edad del alumno que se registra en la academia.
Cinturon	varchar(30)	Color del cinturón	Color del cinturón que porta actualmente del alumno
Tiempo_Cinturon	time	Tiempo con el cinturón	Tiempo que el alumno ha llevado el

<b>Materia:</b> Sistemas de Bases de Datos 1	<b>Facultad de Ingeniería en Electricidad y Computación</b>
<b>Profesor:</b> Ing. Gustavo Irving Cali Mena	<b>ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL</b>
<b>Última Revisión:</b> Fecha	<b>Pág. 6 de 17</b>

			cinturón que porta actualmente
Dia_Pago	Date	Fecha	Fecha para realizar el pago de la inscripción a la academia
<b>Tabla:Torneos</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Categoria	char(30)	Categoría del torneo	Nombre de la categoría
Peso	double	Peso de la categoría	Peso correspondiente a la categoría
Fecha Competicion	date	Fecha	Fecha de realización del torneo
<b>Tabla: Mantenimiento</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Cedula	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de una persona
Nombre_Completo	varchar(30)	Contiene todas las letras del alfabeto	Es el nombre completo de la persona de mantenimiento
Fecha_Nacimiento	date	Fecha	Fecha del día de nacimiento de la persona de mantenimiento
Correo Electronico	varchar(30)	Contiene todas los caracteres posibles	Es el correo electrónico de la persona mantenimiento
Telefono	int	Contiene todos los int posibles	Teléfono de la persona de mantenimiento
Direccion	varchar(30)	Contiene todas los caracteres posibles	Es la dirección de la persona de mantenimiento
<b>Tabla: Cinturon</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Cinturon_ID	int	Contiene todos los int posibles	Es el identificador único de un cinturon
Color	varchar(30)	Contiene todas las letras del alfabeto	Es el tipo de cinturón que tiene cada estudiante

<b>Tabla: Horario_Clase</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Id_Horario	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de un horario de clase
Dia	varchar(30)	Contiene todas las letras del alfabeto	Es el día de la clase
Hora	Varchar(30)	Contiene todos los caracteres posibles	Es la hora en la cual se da la clase
Cedula_Profesor	varchar(30)	Contiene todas los caracteres posibles	Es el identificador único de un profesor
<b>Tabla: Torneos</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Id_Torneo	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de un torneo
Categoria	varchar(30)	Contiene todas las letras del alfabeto	Indica la categoría en el que se encuentra compitiendo el estudiante
Peso	Double	Contiene todos los caracteres posibles	Es el peso establecido para competir en cada una de las categorías
Fecha_Competicion	Date	Fecha	Es la fecha en la cual inicia el torneo
<b>Tabla: Registrar</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Id_Registro	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de un registro
Fecha_Clase	Date	Fecha	Es el día de la clase
Asistencia	Boolean	True o False	Indica si el estudiante si asistió a la clase
Ejercicios_Realizados	varchar(30)	Contiene todas los caracteres posibles	Indica el tipo de ejercicio que se realizo en la clase
Id_Horario	varchar(30)	Contiene todos los posibles caracteres	Es el identificador único de un horario
Cedula_Alumno	Varchar(30)	Contiene todos los posibles caracteres	Es el identificador único de alumno
<b>Tabla: Ganador</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Id_Torneo	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de un ganador
Cedula	varchar(30)	Contiene todas las letras del alfabeto	Es el identificador único de un alumno



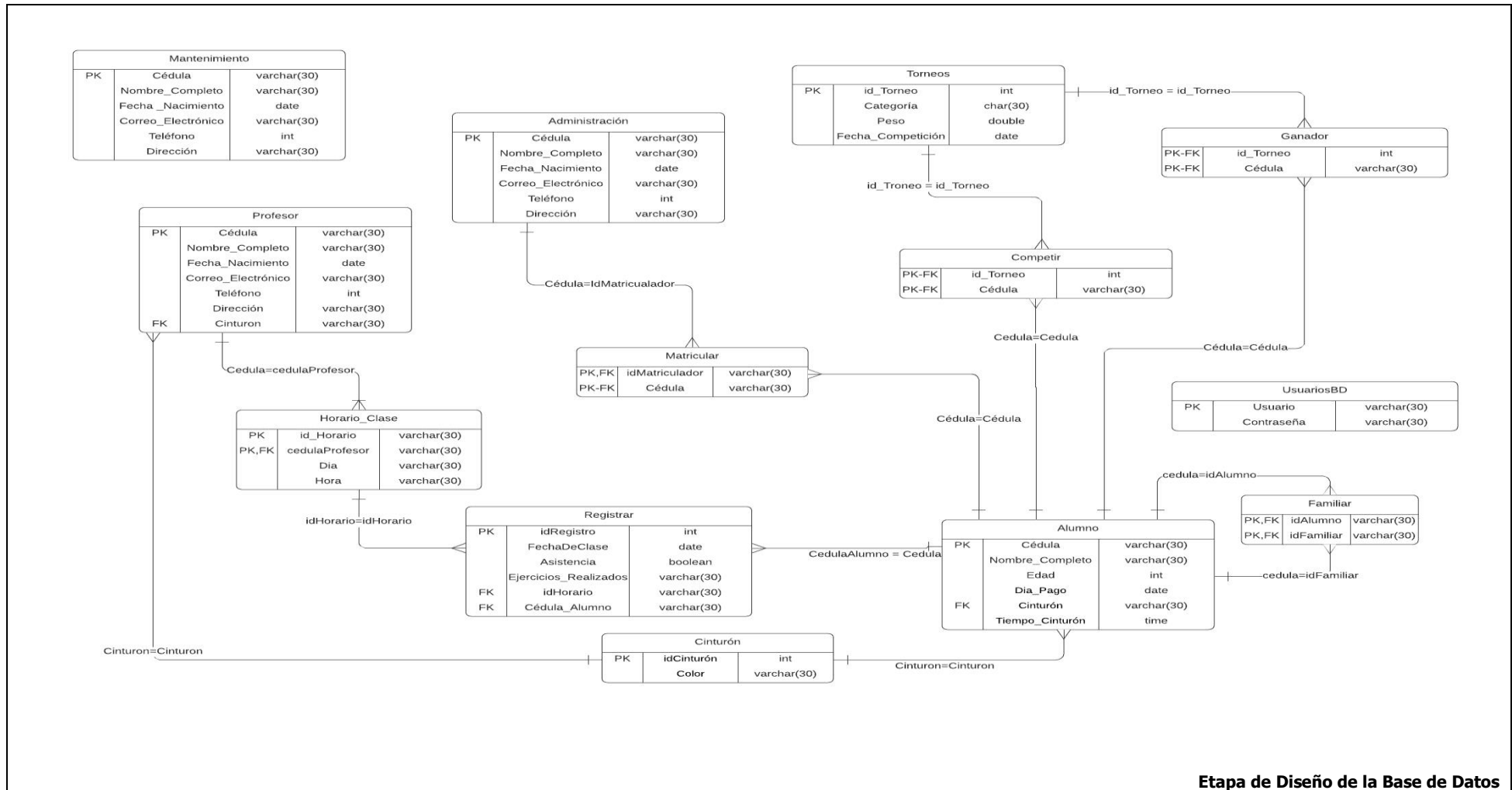
<b>Tabla: Competir</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Id_Torneo	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de un competidor
Cedula	varchar(30)	Contiene todas las letras del alfabeto	Es el identificador único de un alumno

<b>Tabla: Matricular</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
IdMatricular	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de una matricula
Cedula	varchar(30)	Contiene todas las letras del alfabeto	Es el identificador único de un alumno

<b>Tabla: Familiar</b>			
<b>Atributo</b>	<b>Tipo de Dato</b>	<b>Dominio</b>	<b>Descripción</b>
Id_Alumno	varchar(30)	Contiene todos los caracteres posibles	Es el identificador único de un alumno
Id_Familiar	varchar(30)	Contiene todas las letras del alfabeto	Es el identificador único de un familiar

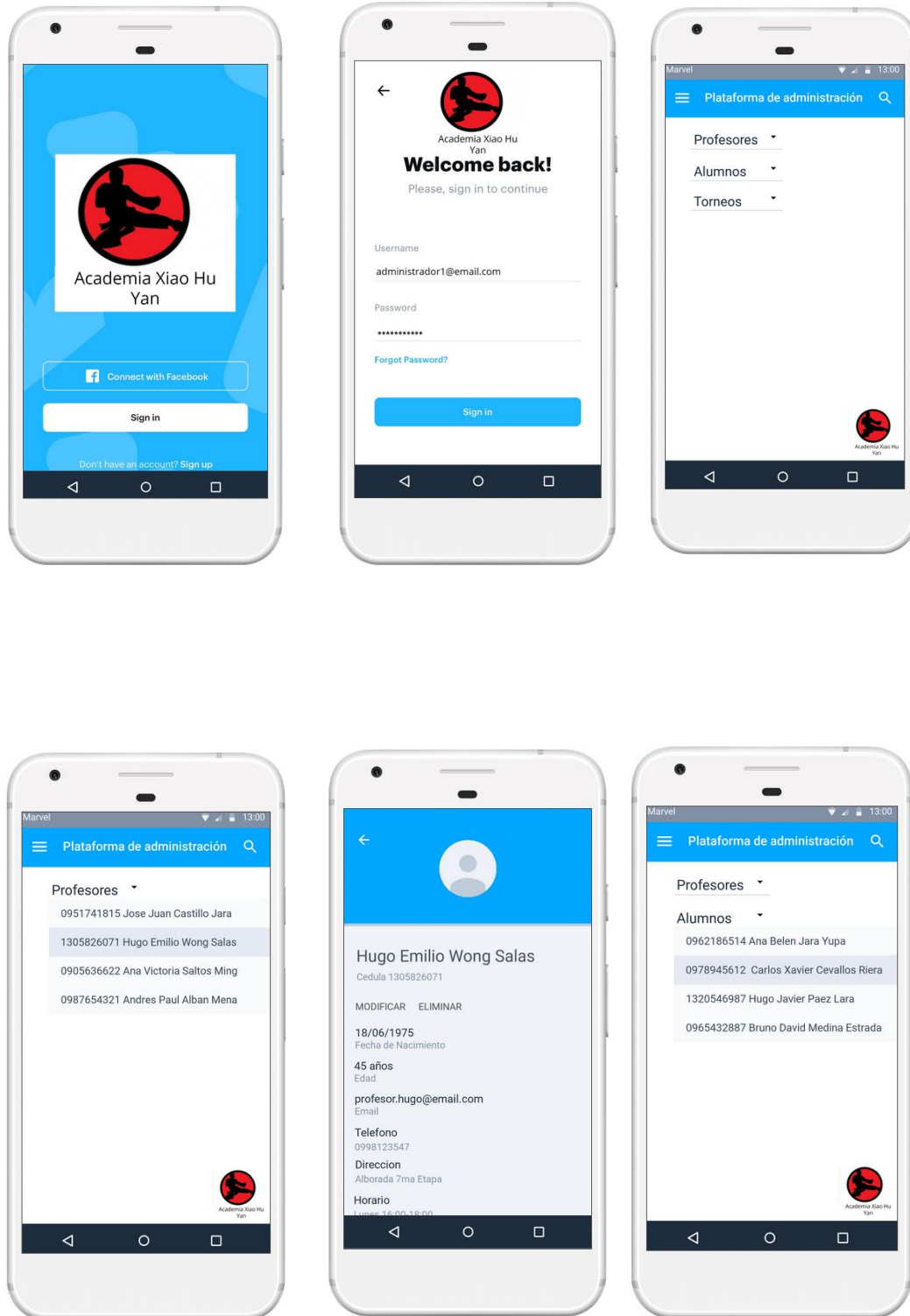
**Etapas de Diseño de la Base de Datos**

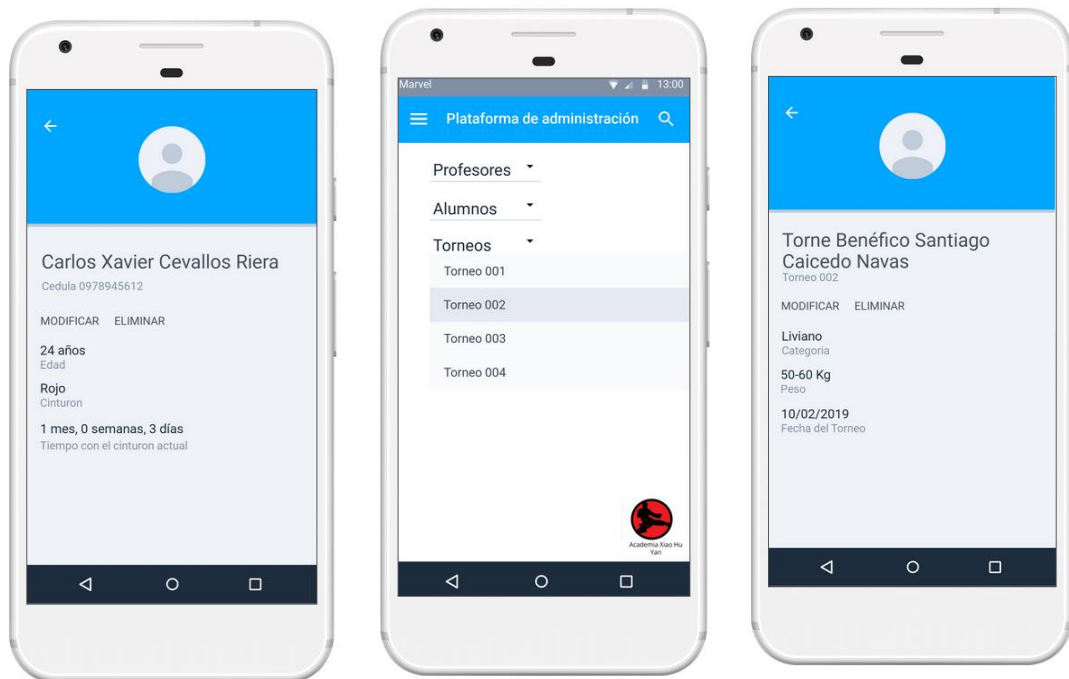
## Modelo Lógico



**Etapas de Diseño de la Base de Datos**

## Flujo de Navegación





**Etapas de Diseño y Prototipo de la Aplicación**

## Modelo Físico

```

CREATE DATABASE PROYECTOBD;
USE PROYECTOBD;

CREATE TABLE USUARIOSBD
(
USUARIO VARCHAR(30) PRIMARY KEY,
CONTRASEÑA VARCHAR(30)
);

CREATE TABLE MANTENIMIENTO
(
CEDULA VARCHAR(30) PRIMARY KEY,
NOMBRE_COMPLETO VARCHAR(30),
FECHA_NACIMIENTO DATE,
CORREO_ELECTRONICO VARCHAR(30),
TELEFONO INT NULL,
DIRECCION VARCHAR(30)
);

CREATE TABLE ADMINISTRACION
(
CEDULA VARCHAR(30) PRIMARY KEY,
NOMBRE_COMPLETO VARCHAR(30),
FECHA_NACIMIENTO DATE,
CORREO_ELECTRONICO VARCHAR(30),
TELEFONO INT NULL,
DIRECCION VARCHAR(30)
);

CREATE TABLE CINTURON
(
CINTURON_ID INT PRIMARY KEY,
COLOR VARCHAR(30)
);

CREATE TABLE PROFESOR
(
CEDULA VARCHAR(30) PRIMARY KEY,
NOMBRE_COMPLETO VARCHAR(30),
FECHA_NACIMIENTO DATE,
CORREO_ELECTRONICO VARCHAR(30),
TELEFONO INT,
DIRECCION VARCHAR(30),
CINTURON INT,
FOREIGN KEY (CINTURON) REFERENCES CINTURON (CINTURON_ID)
);

CREATE TABLE HORARIO_CLASE
(
ID_HORARIO VARCHAR(30) PRIMARY KEY,
DIA VARCHAR(30),
HORA VARCHAR(30),
CEDULA_PROFESOR VARCHAR(30),
foreign key (CEDULA_PROFESOR) REFERENCES PROFESOR (CEDULA)
);

CREATE TABLE ALUMNO
(
CEDULA VARCHAR(30) PRIMARY KEY,
NOMBRE_COMPLETO VARCHAR(70),
EDAD INT,
Dia_Pago DATE,
CINTURON INT,
TIEMPO_CINTURON VARCHAR(30),
FOREIGN KEY (CINTURON) REFERENCES CINTURON (CINTURON_ID)
);

```

<b>Materia:</b> Sistemas de Bases de Datos 1	<b>Facultad de Ingeniería en Electricidad y Computación</b>
<b>Profesor:</b> Ing. Gustavo Irving Cali Mena	<b>ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL</b>
<b>Última Revisión:</b> Fecha	<b>Pág. 13 de 17</b>

```

CREATE TABLE TORNEOS
(
ID_TORNEO VARCHAR(30) PRIMARY KEY,
CATEGORIA VARCHAR(30),
PESO DOUBLE,
FECHA_COMPETICION DATE
);

CREATE TABLE REGISTRAR
(
ID_REGISTRO VARCHAR(30) PRIMARY KEY,
FECHA_CLASE DATE,
ASISTENCIA BOOLEAN,
EJERCICIOS_REALIZADOS VARCHAR(30),
ID_HORARIO VARCHAR(30),
CEDULA_ALUMNO VARCHAR(30),
foreign key (ID_HORARIO) REFERENCES HORARIO_CLASE (ID_HORARIO),
foreign key (CEDULA_ALUMNO) REFERENCES ALUMNO (CEDULA)
);

CREATE TABLE GANADOR
(
ID_TORNEO VARCHAR(30),
CEDULA VARCHAR(30),
PRIMARY KEY(ID_TORNEO, CEDULA),
foreign key (ID_TORNEO) REFERENCES TORNEOS (ID_TORNEO),
foreign key (CEDULA) REFERENCES ALUMNO (CEDULA)
);

CREATE TABLE COMPETIR
(
ID_TORNEO VARCHAR(30),
CEDULA VARCHAR(30),
PRIMARY KEY(ID_TORNEO, CEDULA),
foreign key (ID_TORNEO) REFERENCES TORNEOS (ID_TORNEO),
foreign key (CEDULA) REFERENCES ALUMNO (CEDULA)
);

CREATE TABLE MATRICULAR
(
IDMATRICULAR VARCHAR(30),
CEDULA VARCHAR(30),
PRIMARY KEY(IDMATRICULAR, CEDULA),
foreign key (IDMATRICULAR) REFERENCES ADMINISTRACION (CEDULA),
foreign key (CEDULA) REFERENCES ALUMNO (CEDULA)
);

CREATE TABLE FAMILIAR
(
ID_ALUMNO VARCHAR(30) ,
ID_FAMILIAR VARCHAR(30) ,
PRIMARY KEY(ID_ALUMNO , ID_FAMILIAR),
foreign key (ID_ALUMNO) REFERENCES ALUMNO (CEDULA),
foreign key (ID_FAMILIAR) REFERENCES ALUMNO (CEDULA)
);

-- Consultas
-- 1.Mostrar la cedula, nombre, categoria e id del torneo en los que compitieron
-- los alumnos con edad mayor a o igual a 14
SELECT DISTINCT a.CEDULA, a.NOMBRE_COMPLETO, t.ID_TORNEO, t.CATEGORIA
FROM Alumno a, Competir c, Torneos t
WHERE (a.CEDULA=C.CEDULA) AND (t.ID_TORNEO=c.ID_TORNEO) AND a.EDAD>=14;

-- 2. Mostrar los alumnos y su color de cinturon que se registraron a las clases de agosto
SELECT c.CEDULA, c.NOMBRE_COMPLETO, r.FECHA_CLASE, ci.Color
FROM registrar r , alumno c, cinturon ci
WHERE c.CEDULA=r.CEDULA_ALUMNO AND c.cinturon=ci.cinturon_id AND r.FECHA_CLASE like '%-
08-%';

```

```
-- 3. Mostrar la cantidad de alumnos que han faltado por dia
SELECT HC.DIA, count(HC.DIA) as CantFaltas
FROM registrar r, alumno c, horario_clase hc
WHERE c.CEDULA=r.CEDULA_ALUMNO AND hc.ID_HORARIO=r.ID_HORARIO AND r.ASISTENCIA=
false
GROUP BY hc.DIA;

-- 4. Mostrar los alumnos que asistieron a clases entre el mes de
-- Agosto y Septiembre que tengan un familiar en la academia
select a.nombre_completo
from alumno a, familiar f, registrar r
where a.cedula = f.id_alumno and a.cedula = r.cedula_alumno and r.asistencia = true
AND r.fecha_clase between '2020-08-01' and '2020-09-30'
UNION
select a.nombre_completo
from alumno a, familiar f, registrar r
where a.cedula = f.id_familiar and a.cedula = r.cedula_alumno and r.asistencia = true
AND r.fecha_clase between '2020-08-01' and '2020-09-30';

-- Views
create view Tiempo_Cinturon as
select a.CEDULA, a.NOMBRE_COMPLETO, a.TIEMPO_CINTURON, c.COLOR
from alumno a, cinturon c
where a.CINTURON= c.CINTURON_ID
order by NOMBRE_COMPLETO;

create view Horarios_Profesores as
select p.CEDULA, p.NOMBRE_COMPLETO, hc.DIA, hc.HORA
from profesor p, horario_clase hc
where p.CEDULA = hc.CEDULA_PROFESOR;

create view Alumnos_Ganadores as
select a.CEDULA, a.NOMBRE_COMPLETO, a.EDAD, t.CATEGORIA, c.COLOR
from alumno a, ganador g, torneos t, cinturon c
where a.CEDULA=g.CEDULA AND g.ID_TORNEO=t.ID_TORNEO AND a.CINTURON=c.CINTURON_ID
group by a.CEDULA;

create view Familiares as
select a.CEDULA, a.NOMBRE_COMPLETO, f.cedulaFamiliar, f.nombre
from alumno a, (select a.NOMBRE_COMPLETO as nombre, a.CEDULA as cedulaFamiliar, f.ID_FAMILIAR
from alumno a, familiar f
where f.ID_ALUMNO = a.CEDULA) f
where f.ID_FAMILIAR = a.CEDULA;

-- SPS
-- 1. Ver horario de un solo profesor
delimiter //
create procedure verHorarioProfesor( IN CedulaProfesor varchar(30))
begin
select *
from horario_clase
where CEDULA_PROFESOR = CedulaProfesor;
end //
delimiter ;
-- call verHorarioProfesor('0951741815');

-- 2. Ver el color del cinturon y el tiempo que tiene un estudiante especificando su cedula.
delimiter //
create procedure verCinturon( IN IdAlumno varchar(30), OUT color varchar(30), OUT tiempo
varchar(30))
begin
select c.COLOR,a.TIEMPO_CINTURON into color,tiempo
from alumno a, cinturon c
where a.CINTURON= c.CINTURON_ID and a.CEDULA=IdAlumno;
end //
delimiter ;
-- call verCinturon('0988589801',@color,@tiempo);
-- select @color,@tiempo;
```

<b>Materia:</b> Sistemas de Bases de Datos 1	<b>Facultad de Ingeniería en Electricidad y Computación</b>
<b>Profesor:</b> Ing. Gustavo Irving Cali Mena	<b>ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL</b>
<b>Última Revisión:</b> Fecha	<b>Pág. 15 de 17</b>

```
-- 3. Elimiar el registro de un alumno en la tabla alumno
delimiter //
create procedure eliminarAlumno( IN cedula varchar(30))
begin
delete from alumno where CEDULA=cedula;
end //
delimiter ;
-- call eliminarAlumno('0956009820');
```

```
-- 4. Mostrar los alumnos de acuerdo a la edad especificada
delimiter //
create procedure edadAlumnos(IN edad int)
begin
select *
from alumno a
where a.EDAD = edad;
end //
delimiter ;
-- call edadAlumnos(20);
```

```
-- Triggers
-- 1. Colocar la asistencia a falso antes de INSERT un registro
delimiter |
create trigger asistenciaInicial before insert on registrar
for each row
begin
set NEW.ASISTENCIA= false ;
end |
delimiter ;
```

```
-- 2. Eliminar los registro de una hora de clase cuando se actualiza su hora.
delimiter |
create trigger eliminarRegistro1 after update on horario_clase
for each row
begin
delete from registrar r where r.ID_HORARIO= NEW.ID_HORARIO;
end |
delimiter ;
```

```
-- 3. Actualizar el tiempo cuando cambia de cinturon
delimiter |
create trigger actualizarTiempo before update on alumno
for each row
begin
if (NEW.CINTURON != OLD.CINTURON) then set NEW.TIEMPO_CINTURON='0 meses';
end if;
```

```
end |
delimiter ;
```

```
-- 4. Como nueva regla, si se agrega nuevos torneos cantonales, estos serán con un peso de 35 kg.
delimiter |
create trigger setPesoCantonal before insert on torneos
for each row
begin
if(new.CATEGORIA = 'Cantonal' ) then set NEW.PESO = 35;
end if;
end |
delimiter ;
```

Etapas de implementación de la Base de datos



## Anexos

### EXTRA

Creación del respaldo de la base de datos: en formato .SQL usando comandos.

```
mysql> quit
Bye
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -h localhost -u root -p proyectobd > C:\Users\hugui\Desktop\RESPALDO-BASE_DATOS\proyectobd.sql
Enter password: *****
C:\Program Files\MySQL\MySQL Server 8.0\bin>
```

### EXTRA

LOGIN REALIZADO EN LA APP:

The screenshot shows a web application interface for 'Academia Xiao Hu Yan'. At the top center is a logo featuring a red circle with a black silhouette of a person in a martial arts pose, set against a blue background. Below the logo, the text 'Academia Xiao Hu Yan' is displayed. On the left side, there is a login section with two input fields: 'USUARIO:' and 'CONTRASEÑA:'. Below these fields is a 'Conectar' button. To the right of the login section, there is a section titled 'Tabla Alumno' which contains several input fields for student information: 'CÉDULA:', 'NOMBRE COMPLETO:', 'EDAD:', 'DÍA DE PAGO:', 'CINTURÓN:', and 'TIEMPO DE CINTURÓN:'. Below these fields are buttons for 'Delete', 'Insert', 'Update', and 'Limpiar'. There are also labels for 'EstadoConexión', 'EstadoConsultar', 'EstadoDelete', and 'EstadoInsert'.