



Universidad Autónoma de Baja California Sur

BASE DE DATOS AVANZADAS

ENTREGA FINAL DEL PROYECTO

Integrantes:

Aviles Ramos Arisbeth Guadalupe

Briones Herrera Jazziel Abdiel

Castro Castro Martin Alexis

Gibert Toledo Ricardo

Carrera:

Ingeniería en Desarrollo de Software

Maestro:

Mtra. Irma Leticia Palacios López

Fecha de entrega: 01-06-2023



Introducción.

En este proyecto vamos a recolectar un análisis completo sobre la función y responsabilidad de la base de datos que se ha trabajado para el proyecto de software realizado para el gestor de campos Fielder Community.

En ello comenzamos viendo los principales antecedentes para buscar los puntos de entrada y salida en donde podríamos poner a prueba la base de datos, la realización de tablas principales, así como también un modelo entidad-relación donde tendríamos un mejor orden de las bases de datos que se utilizarán para nuestro proyecto, donde todo punto trabajado fue supervisado por el equipo de proyecto, así como por el docente, la actualización de datos al momento de manipular la página utilizando los conocidos disparadores o triggers para la eliminar y también agregar los campos así como también los perfiles de usuarios, una vez realizado la base de datos del proyecto se mostrarán capturas de evidencia en las cuales se apreciará la pagina web en funcionamiento con la base de datos así como capturas de las tablas que se esta utilizando para dicho apartado, sin más que agregar, a continuación mostraremos evidencias, así como el desarrollo de la base de datos, desde su comienzo hasta la base de datos concluida.



Antecedentes:

Nuestra intención es realizar un proyecto de software que será lanzado a principios del mes de junio, inicialmente para plataforma WEB y más adelante para plataforma móvil.

El proyecto consiste en una aplicación que busca almacenar los campos deportivos que se abarcan en el municipio de La Paz Baja California Sur, centrando en distintos deportes como futbol, futbol 7, tenis o basketball.

Buscamos motivar e incitar a la comunidad deportiva a realizar actividades deportivas, evitando buscar el campo adecuado por su cuenta, ayudando a escoger y a organizar eventos donde los mismos usuarios puedan participar alegrarse y divertirse.

Utilizamos 9 tablas ya que nuestro proyecto consiste en gestionar campos deportivos, así como también la realización de eventos deportivos en el municipio de La Paz:

Se necesito principalmente una tabla de campo donde se registran los datos de cada uno de los campos del municipio, así como también de su colonia y el estado del servicio en el que se encuentra el campo esto quiere decir que se podrá ver si está ocupado o no.

La tabla del equipo es muy importante ya que tendrá los datos del nombre del equipo, director técnico del equipo, cantidad de jugadores que tendrá la validación de un límite máximo sobre el equipo, el estado de registro es cuando esta la espera de confirmar el equipo, otro dato seria fecha de registro que es cuando inicio el equipo, y la fecha de salida es cuando se sale el equipo.

Otra tabla importante es la de tabla jugador (usuario) ya que será el encargado de reservar campos, organizar torneos y registrar a su equipo, y unirse a uno, es claro destacar que la fecha de entrada, para saber el dato de cuando se registró y la fecha de salida de saber cuándo se salió.

Otra tabla necesaria es la de torneo oficial, puesto que con esa podremos almacenar los datos del torneo que el jugador registró y de ahí se desprende la tabla de los trofeos ya que cada torneo tendrá uno diferente.

La tabla torneo oficial solamente se basará en las características básicas como el nombre, tipo del torneo, las fechas de inicio y salida. La tabla trofeos oficiales solamente



DASC

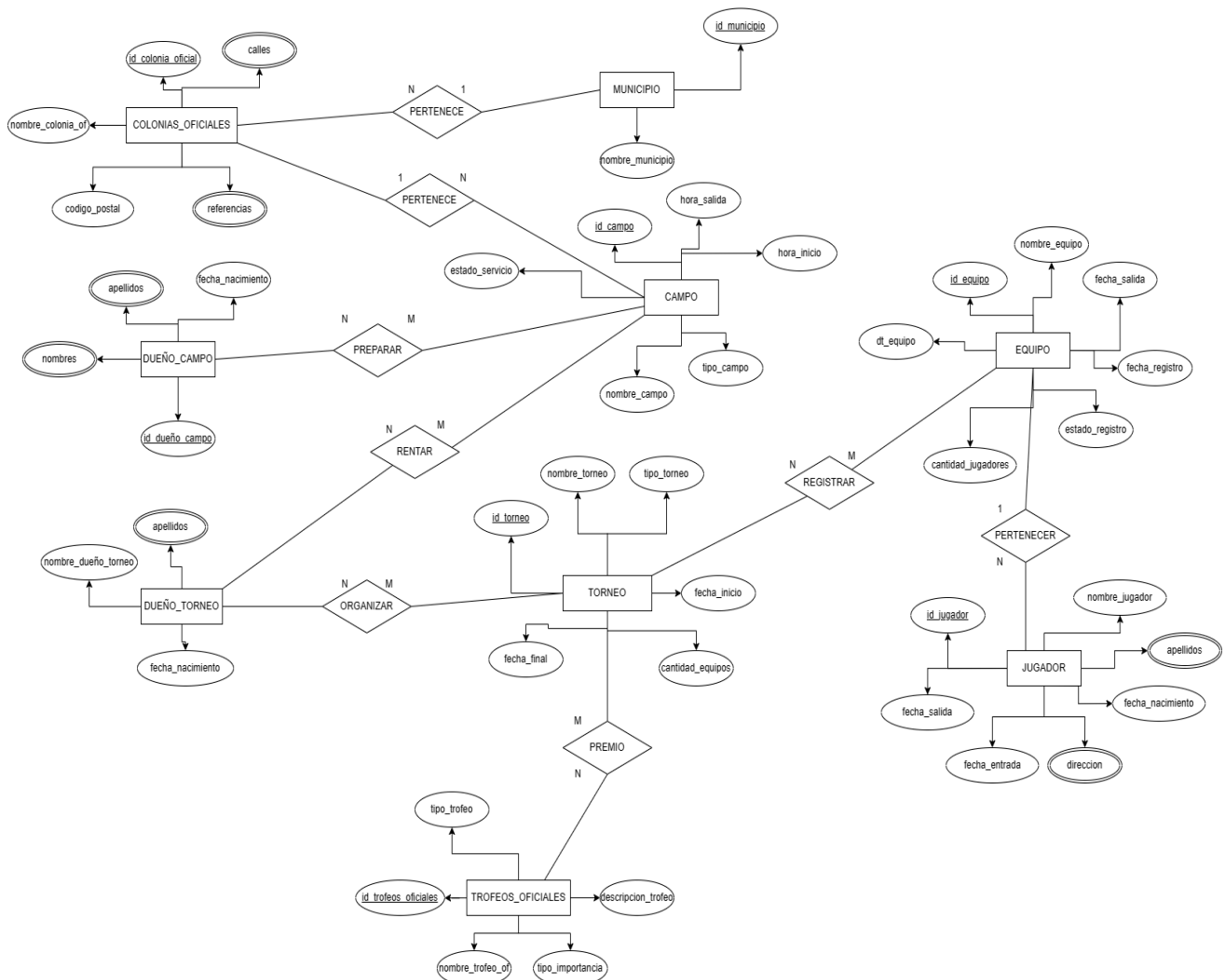
se basará en las características básicas como el nombre, tipo del trofeo y el tipo de importancia que tiene.

La tabla dueño del torneo solamente se basará en quien este encargado de las canchas, obteniendo los siguientes datos de este que sería el nombre, apellidos, fecha de nacimiento.

La tabla dueño campo son los que se encargan de darle algún mantenimiento al campo que tienen asignado donde los datos que tendrá son nombre, apellidos y fecha de nacimiento.

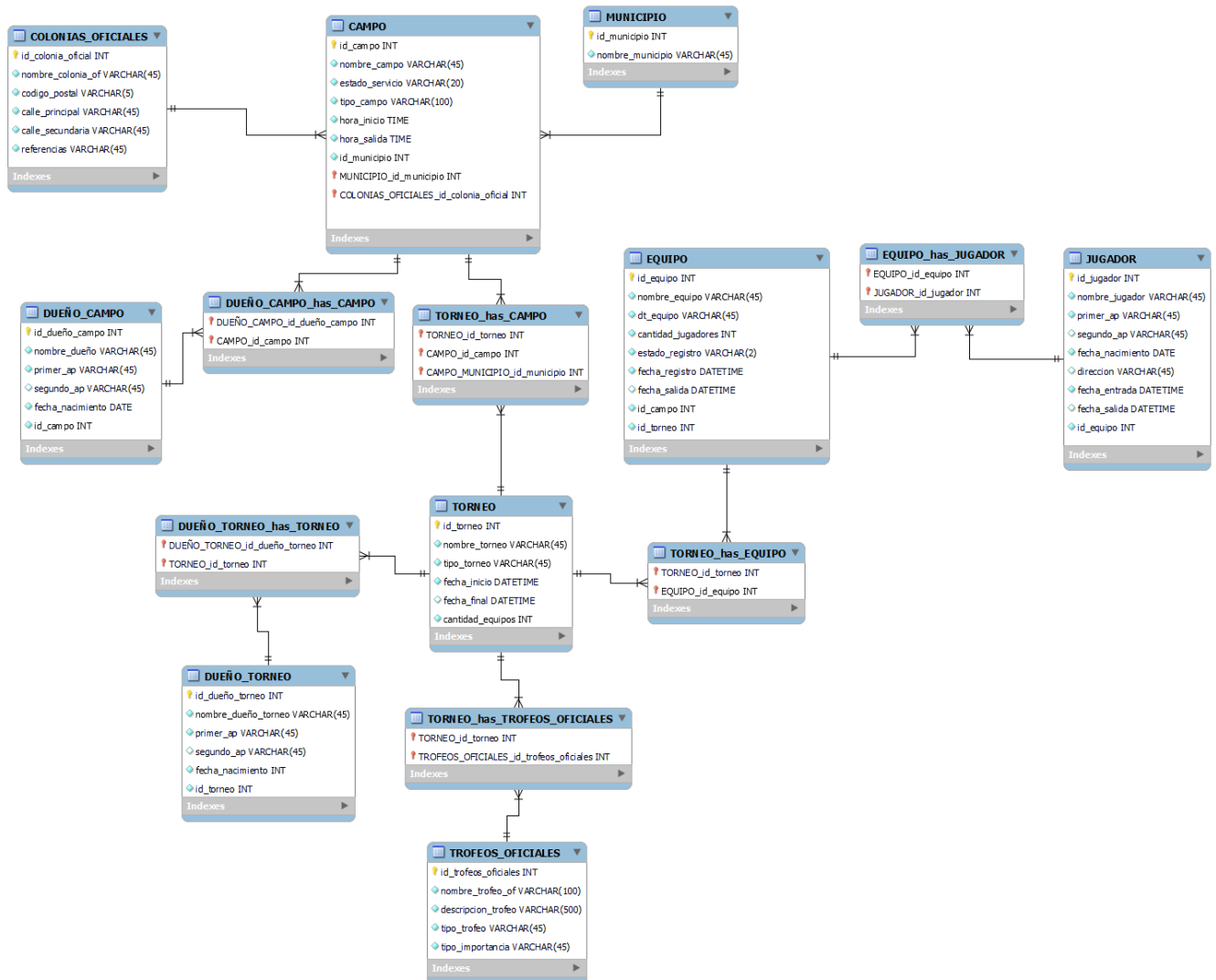
La tabla colonias oficiales son las que están registradas dadas de alta a la página Fielder Community donde se registrara los nombres, código postal, referencias y calles.

La tabla municipio es solamente donde se ubica los campos oficiales.





DASC





DASC

241 • SHOW TABLES;

Result Grid Filter Rows:

Tables_in_feldercommunity
campo
colonias_oficiales
datos_campo
datos_colonias
dueño_torneo
equipo
equipos_torneo
informacion_torneos
jugador
municipio
organizadores_torneos
torneo
trofeos_oficiales

```

15 • CREATE TABLE TROFEOS_OFICIALES(
16     id_trofeo_oficiales INT AUTO_INCREMENT,
17     CONSTRAINT Pk_id_trofeo_oficiales PRIMARY KEY(id_trofeo_oficiales),
18     nombre_trofeo_of VARCHAR(100) NOT NULL,
19     CONSTRAINT Unq_nombre_trofeo_of UNIQUE(nombre_trofeo_of),
20     descripcion_trofeo VARCHAR(500) NOT NULL,
21     tipo_trofeo VARCHAR(100) NOT NULL,
22     CONSTRAINT Chk_tipo_trofeo CHECK (tipo_trofeo IN ("DINERO","TROFEOS","DINERO Y TROFEOS")),
23     tipo_importancia VARCHAR(45) NOT NULL,
24     CONSTRAINT Chk_tipo_importancia CHECK (tipo_importancia IN ("ALTO", "MEDIO", "BAJO"))
25 );

```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

id_trofeo_oficiales	nombre_trofeo_of	descripcion_trofeo	tipo_trofeo	tipo_importancia
1	SPARTACUS	Es un trofeo nunca antes visto, y de un gran valor	DINERO	ALTO
2	TORNATE	Es un trofeo nunca antes visto	TROFEOS	MEDIO
3	CADELABRO	Sin comentarios...	DINERO Y TROFEOS	BAJO
*	NULL	NULL	NULL	NULL



DASC

```
32 • CREATE TABLE COLONIAS_OFICIALES(  
33     id_colonia_oficial INT AUTO_INCREMENT,  
34     CONSTRAINT PK_id_colonia_oficial PRIMARY KEY(id_colonia_oficial),  
35     nombre_colonia VARCHAR(45) NOT NULL,  
36     codigo_postal VARCHAR(5) NOT NULL,  
37     calle_principal VARCHAR(45) NOT NULL,  
38     calle_secundaria VARCHAR(45),  
39     referencias VARCHAR(200),  
40     id_municipio INT,  
41     CONSTRAINT Fk_id_municipio FOREIGN KEY(id_municipio) REFERENCES MUNICIPIO(id_municipio) ON UPDATE CASCADE ON DELETE CASCADE  
42 );  
43 • INSERT INTO COLONIAS_OFICIALES(nombre_colonia,codigo_postal,calle_principal,calle_secundaria,referencias,id_municipio)
```

id_colonia_oficial	nombre_colonia	codigo_postal	calle_principal	calle_secundaria	referencias	id_municipio
1	Miramar	23085	264 C. Isla	Av. Miramar	Al lado de un secundaria	1
2	SOLIDARIDAD	23083	avenida universidad	geologia	al lado de un gym, la cancha de piso	1

```
48 • CREATE TABLE CAMPO(  
49     id_campo INT AUTO_INCREMENT,  
50     CONSTRAINT Pk_id_campo PRIMARY KEY(id_campo),  
51     nombre_campo VARCHAR(45) NOT NULL,  
52     calificacion FLOAT DEFAULT NULL,  
53     CONSTRAINT Chk_calificacion CHECK(calificacion BETWEEN 1 AND 5),  
54     estado_servicio VARCHAR(50) DEFAULT "LIBRE",  
55     CONSTRAINT Chk_estado_servicio CHECK(estado_servicio IN("LIBRE", "OCUPADO")),  
56     tipo_campo VARCHAR(100) NOT NULL,  
57     CONSTRAINT Chk_tipo_campo CHECK(tipo_campo IN("FUTBOL", "FUTBOL RAPIDO", "TENIS", "BASQUETBOL", "VOLEIBOL")),  
58     hora_inicio TIME NOT NULL,  
59     hora_salida TIME NOT NULL,  
60     id_colonia_of INT,  
61     CONSTRAINT FK_id_colonia_of FOREIGN KEY(id_colonia_of) REFERENCES COLONIAS_OFICIALES(id_colonia_oficial) ON UPDATE CASCADE ON DELETE CASCADE  
62 );
```

id_campo	nombre_campo	calificacion	estado_servicio	tipo_campo	hora_inicio	hora_salida	id_colonia_of
1	MIRAMAR	4.5	LIBRE	FUTBOL	09:00:00	20:00:00	1
2	MIRAMAR	4	LIBRE	FUTBOL RAPIDO	08:00:00	22:00:00	1
3	MIRAMAR	3.4	LIBRE	VOLEIBOL	08:00:00	22:00:00	1
4	SOLI MEZ	2.4	LIBRE	FUTBOL RAPIDO	00:00:00	24:00:00	2

```
71 • CREATE TABLE TORNEO(  
72     id_torneo INT AUTO_INCREMENT,  
73     CONSTRAINT Pk_id_torneo PRIMARY KEY(id_torneo),  
74     nombre_torneo VARCHAR(45) NOT NULL,  
75     CONSTRAINT Unq_nombre_torneo UNIQUE(nombre_torneo),  
76     tipo_torneo VARCHAR(45) NOT NULL,  
77     CONSTRAINT Chk_tipo_torneo CHECK(tipo_torneo IN("FUTBOL", "FUTBOL RAPIDO", "TENIS", "BASQUETBOL", "VOLEIBOL")),  
78     fecha_inicio DATE NOT NULL,  
79     fecha_final DATE DEFAULT NULL,  
80     cantidad_equipos INT NOT NULL,  
81     CONSTRAINT Chk_cantidad_equipos CHECK(cantidad_equipos > 7),  
82     id_campo INT,  
83     CONSTRAINT Fk_id_campo FOREIGN KEY(id_campo) REFERENCES CAMPO(id_campo) ON UPDATE CASCADE ON DELETE CASCADE,  
84     id_trofeo_of INT,  
85     CONSTRAINT Fk_id_trofeo_of FOREIGN KEY(id_trofeo_of) REFERENCES TROFEOS_OFICIALES(id_trofeo_oficiales) ON UPDATE CASCADE ON DELETE CASCADE  
86 );
```

id_torneo	nombre_torneo	tipo_torneo	fecha_inicio	fecha_final	cantidad_equipos	id_campo	id_trofeo_of
1	RELAMPAGO	FUTBOL	2023-06-01	NULL	8	1	1
2	RELAMPAGO II	FUTBOL RAPIDO	2023-05-19	NULL	8	2	2
3	RELAMPAGO VOLI	VOLEIBOL	2023-05-01	NULL	8	3	3



DASC

```
93 CREATE TABLE EQUIPO(  
94     id_equipo INT AUTO_INCREMENT,  
95     CONSTRAINT Pk_id_equipo PRIMARY KEY(id_equipo),  
96     nombre_equipo VARCHAR(45) NOT NULL,  
97     CONSTRAINT Unq_nombre_equipo UNIQUE(nombre_equipo),  
98     dt_equipo VARCHAR(45) NOT NULL,  
99     CONSTRAINT Unq_dt_equipo UNIQUE(dt_equipo),  
100     cantidad_jugadores INT NOT NULL,  
101     CONSTRAINT Chk_cantidad_jugadores CHECK(cantidad_jugadores between 7 and 25),  
102     estado_registro VARCHAR(2) NOT NULL DEFAULT "NO",  
103     CONSTRAINT Chk_estado_registro CHECK(estado_registro IN("SI","NO")),  
104     fecha_registro DATE NOT NULL,  
105     fecha_salida DATE DEFAULT NULL,  
106     id_torneo INT DEFAULT NULL,  
107     CONSTRAINT Fk_id_torneo FOREIGN KEY(id_torneo) REFERENCES TORNEO(id_torneo) ON UPDATE CASCADE ON DELETE CASCADE  
108 );  
109 INSERT INTO EQUIPO(nombre_equipo,dt_equipo,cantidad_jugadores,fecha_registro,id_torneo)
```

Result Grid

	id_equipo	nombre_equipo	dt_equipo	cantidad_jugadores	estado_registro	fecha_registro	fecha_salida	id_torneo
▶	1	PORCINOS	Ibái	10	NO	2023-04-28	NULL	1
	2	GALLOS FC	Jose	15	NO	2023-04-28	NULL	2
	3	MAQUINA VOLI	Arturo	12	NO	2023-04-28	NULL	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
115 CREATE TABLE JUGADOR(  
116     id_jugador INT AUTO_INCREMENT,  
117     CONSTRAINT Pk_id_jugador PRIMARY KEY(id_jugador),  
118     nombre_jugador VARCHAR(45) NOT NULL,  
119     primer_ap VARCHAR(45) NOT NULL,  
120     segundo_ap VARCHAR(45),  
121     fecha_nacimiento DATE NOT NULL,  
122     direccion VARCHAR(45),  
123     email VARCHAR(30) NOT NULL, -- NUEVA IMPLEMENTACION (PRUEBA)  
124     CONSTRAINT Unq_email UNIQUE(email),  
125     password CHAR(100) NOT NULL, -- NUEVA IMPLEMENTACION (PRUEBA)  
126     fecha_entrada DATE NOT NULL,  
127     fecha_salida DATE DEFAULT NULL,  
128     id_equipo INT,  
129     CONSTRAINT Fk_id_equipo FOREIGN KEY(id_equipo) REFERENCES EQUIPO(id_equipo) ON UPDATE CASCADE ON DELETE CASCADE  
130 );  
131 INSERT INTO JUGADOR(nombre_jugador,primer_ap,segundo_ap,fecha_nacimiento,direccion,email,password,fecha_entrada,fecha_salida,id_equipo)
```

Result Grid

	id_jugador	nombre_jugador	primer_ap	segundo_ap	fecha_nacimiento	direccion	email	password	fecha_entrada	fecha_salida	id_equipo
▶	1	Jazziel	Briones	Herrera	2002-04-18	Avenida universidad/geologia y administracion	Jazzielbh@gmail.com	40bd001563089fc35165329ea1ff5c5ecbcbbeef	2023-04-28	NULL	1
	2	Martin	Castro	Castro	2002-06-13	Santa fe	Martin@gmail.com	d435a6cdd786300dff204ee7c2ef942d3e9034e2	2023-04-28	NULL	2
	3	Ricardo	Gibert	Toledo	2002-01-22	Pedregal	Ricardo@gmail.com	f1f836cb4ea6efb2a0b1b99f41ad8b103eff4b59	2023-04-28	NULL	3
	4	David	Talamantes	Castro	1998-02-08	Avenida universidad/geologia y administracion	David@gmail.com	fb644351560d8296fe6da332236b1f8d61b2828a	2023-04-28	NULL	1
	5	Jorge	Ignacio	Perez	1969-01-20	Santa fe	Jorge@gmail.com	8ee51c0aa2c2f4ee2a5b4b7ef95a99db7df1066d7	2023-04-28	NULL	2
	6	Alexis	Gasca	Ruiz	1998-06-30	Pedregal	Alexis@gmail.com	fb644351560d8296fe6da332236b1f8d61b2828a	2023-04-28	NULL	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

JUGADOR 76 x

```
141 CREATE TABLE DUEÑO_TORNEO(  
142     id_dueño_torneo INT AUTO_INCREMENT,  
143     CONSTRAINT Pk_id_dueño_torneo PRIMARY KEY(id_dueño_torneo),  
144     nombre_dueño_torneo VARCHAR(45) NOT NULL,  
145     primer_ap VARCHAR(45),  
146     segundo_ap VARCHAR(45),  
147     fecha_nacimiento DATE NOT NULL,  
148     id_torneo INT,  
149     CONSTRAINT Fk_id_torn FOREIGN KEY(id_torneo) REFERENCES TORNEO(id_torneo) ON UPDATE CASCADE ON DELETE CASCADE  
150 );  
151 INSERT INTO DUEÑO_TORNEO(nombre_dueño_torneo,primer_ap,segundo_ap,fecha_nacimiento,id_torneo)  
152 VALUES("Francisco","Alvarez","Castro","1958-05-20",1)
```

Result Grid

	id_dueño_torneo	nombre_dueño_torneo	primer_ap	segundo_ap	fecha_nacimiento	id_torneo
▶	1	Francisco	Alvarez	Castro	1958-05-20	1
	2	Alvaro	Castro	Trinidad	1978-01-30	2
	3	Eduardo	De la llave	Castro	1989-09-07	3
*	NULL	NULL	NULL	NULL	NULL	NULL



El uso que tendrá esta vista es sencillo ya que solamente mostrara datos del campo, como que calificación tiene, el estado del servicio en el que se encuentre, su nombre y en donde se encuentra, el tipo de campo que es, cuál es su hora de entrada y de salidas, la colonia que es, su código postal, referencias.

```
160 • CREATE VIEW DATOS_CAMPO AS
161 SELECT CA.tipo_campo,CA.calificacion,CA.hora_inicio,CA.hora_salida,CA.estado_servicio,CA.nombre_campo,
162 COL.nombre_colonia,COL.codigo_postal,COL.calle_principal,
163 COL.calle_secundaria,COL.referencias
164 FROM CAMPO AS CA
165 JOIN COLONIAS_OFICIALES AS COL
166 ON COL.id_colonia_oficial = CA.id_colonia_of;
167 -- DROP VIEW DATOS_CAMPO;
168 • SELECT * FROM DATOS_CAMPO;
169 # -----
```

	tipo_campo	calificacion	hora_inicio	hora_salida	estado_servicio	nombre_campo	nombre_colonia	codigo_postal	calle_principal	calle_secundaria	referencias
▶	FUTBOL	4.5	09:00:00	20:00:00	LIBRE	MIRAMAR	Miramar	23085	264 C. Isla	Av. Miramar	Al lado de un secundaria
	FUTBOL RAPIDO	4	08:00:00	22:00:00	LIBRE	MIRAMAR	Miramar	23085	264 C. Isla	Av. Miramar	Al lado de un secundaria
	VOLEIBOL	3.4	08:00:00	22:00:00	LIBRE	MIRAMAR	Miramar	23085	264 C. Isla	Av. Miramar	Al lado de un secundaria
	FUTBOL RAPIDO	2.4	00:00:00	24:00:00	LIBRE	SOLI MEZ	SOLIDARIDAD	23083	avenida universidad	geologia	al lado de un gym, la cancha de piso

La información de torneos se refiere todo lo que conlleva el torneo actual, es decir se mostrarán datos como: el tipo del torneo, que trofeo darán, el nombre del torneo, nombre del campo cuando inicia el torneo y que equipos están en ese torneo.

```
173 • CREATE VIEW INFORMACION_TORNEOS AS
174 SELECT TOR.tipo_torneo,TRO.tipo_trofeo,TOR.nombre_torneo,CA.nombre_campo,TOR.fecha_inicio,
175 EQU.nombre_equipo,EQU.dt_equipo
176 FROM CAMPO AS CA
177 JOIN TROFEOS_OFICIALES AS TRO
178 ON TRO.id_trofeo_oficiales = CA.id_campo
179 JOIN TORNEO AS TOR
180 ON TOR.id_torneo = CA.id_campo
181 JOIN EQUIPO AS EQU
182 ON EQU.id_equipo = TOR.id_torneo
183 JOIN COLONIAS_OFICIALES AS COL
184 ON COL.id_colonia_oficial = CA.id_colonia_of;
185 -- DROP VIEW INFORMACION_TORNEOS;
186 • SELECT * FROM INFORMACION_TORNEOS;
187 # -----
```

	tipo_torneo	tipo_trofeo	nombre_torneo	nombre_campo	fecha_inicio	nombre_equipo	dt_equipo
▶	FUTBOL	DINERO	RELAMPAGO	MIRAMAR	2023-06-01 08:00:00	PORCINOS	Ibai
	FUTBOL RAPIDO	TROFEOS	RELAMPAGO II	MIRAMAR	2023-05-19 10:00:00	GALLOS FC	Jose
	VOLEIBOL	DINERO Y TROFEOS	RELAMPAGO VOLI	MIRAMAR	2023-05-01 14:00:00	MAQUINA VOLI	Arturo



La tabla datos colonias es para saber la información de las colonias para así estar más informado y saber en qué lugar se realizarán los partidos proporciona la siguiente información que sería el nombre de la colonia, su código postal, la calle principal, las referencias, el nombre de los campos, que tipo de campos hay y en que municipio están.

ojo si en una colonia existen 3 campos diferentes, cada campo tendrá su dirección como el ejemplo de abajo, en este caso, los 3 campos se encuentran juntos, entonces su dirección sería la misma, a excepción del tipo de campo del que es.

```
190 • CREATE VIEW DATOS_COLONIAS AS
191 SELECT COL.nombre_colonia,COL.codigo_postal,
192 COL.calle_principal,COL.referencias,CA.nombre_campo,CA.tipo_campo,MUN.nombre_municipio
193 FROM COLONIAS_OFICIALES AS COL
194 JOIN CAMPO AS CA
195 ON COL.id_colonia_oficial = CA.id_colonia_of
196 JOIN MUNICIPIO AS MUN
197 ON MUN.id_municipio = COL.id_municipio;
198 -- DROP VIEW DATOS_COLONIAS;
199 • SELECT * FROM DATOS_COLONIAS;
```

	nombre_colonia	codigo_postal	calle_principal	referencias	nombre_campo	tipo_campo	nombre_municipio
▶	Miramar	23085	264 C. Isla	Al lado de un secundaria	MIRAMAR	FUTBOL	La Paz
	Miramar	23085	264 C. Isla	Al lado de un secundaria	MIRAMAR	FUTBOL RAPIDO	La Paz
	Miramar	23085	264 C. Isla	Al lado de un secundaria	MIRAMAR	VOLEIBOL	La Paz
	SOLIDARIDAD	23083	avenida universidad	al lado de un gym, la cancha de piso	SOLI MEZ	FUTBOL RAPIDO	La Paz

Los organizadores de torneos son para tener un mejor control sobre los torneos que se llevan, la información que proporcionará será la siguiente; el nombre del dueño del torneo, el tipo de torneo que es, el tipo de trofeo, el nombre de los equipos que participan con su cantidad de jugadores con los que cuentan y el nombre del torneo.

```
208 • CREATE VIEW ORGANIZADORES_TORNEOS AS
209 SELECT DUE.nombre_dueño_torneo,TOR.tipo_torneo,TRO.nombre_trofeo_of,
210 TRO.tipo_trofeo,EQU.nombre_equipo,EQU.cantidad_jugadores,TOR.nombre_torneo
211 FROM DUEÑO_TORNEO AS DUE
212 JOIN TORNEO AS TOR
213 ON DUE.id_dueño_torneo = TOR.id_torneo
214 JOIN TROFEOS_OFICIALES AS TRO
215 ON TRO.id_trofeo_oficiales = TOR.id_torneo
216 JOIN EQUIPO AS EQU
217 ON EQU.id_equipo = TOR.id_torneo;
218 • SELECT * FROM ORGANIZADORES_TORNEOS;
219 # -----
220 /*LOS EQUIPOS QUE ESTAN EN UN TORNEO APARECERAN JUNTO CON EL DIRECTOR TECNICO
```

	nombre_dueño_torneo	tipo_torneo	nombre_trofeo_of	tipo_trofeo	nombre_equipo	cantidad_jugadores	nombre_torneo
▶	Francisco	FUTBOL	SPARTACUS	DINERO	PORCINOS	10	RELAMPAGO
	Alvaro	FUTBOL RAPIDO	TORNATE	TROFEOS	GALLOS FC	15	RELAMPAGO II
	Eduardo	VOLEIBOL	CADELABRO	DINERO Y TROFEOS	MAQUINA VOLI	12	RELAMPAGO VOLI



Los equipos que están en un torneo aparecerán junto con el director técnico del equipo, junto con sus jugadores para tener una vista clara al momento de mostrar información.

```
222 • CREATE VIEW EQUIPOS_TORNEO AS
223 SELECT EQU.nombre_equipo, JU.nombre_jugador, EQU.dt_equipo, TOR.nombre_torneo
224 FROM EQUIPO AS EQU
225 JOIN JUGADOR AS JU
226 ON EQU.id_equipo = JU.id_equipo
227 JOIN TORNEO AS TOR
228 ON TOR.id_torneo = EQU.id_equipo;
229 -- DROP VIEW EQUIPOS_TORNEO;
230 • SELECT * FROM EQUIPOS_TORNEO;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
	nombre_equipo	nombre_jugador	dt_equipo
	nombre_torneo		
▶	PORCINOS	Jazziel	Ibai
	PORCINOS	David	Ibai
	GALLOS FC	Martin	Jose
	GALLOS FC	Jorge	Jose
	MAQUINA VOLI	Ricardo	Arturo
	MAQUINA VOLI	Alexis	Arturo



DASC

Procedimiento almacenado: este procedimiento almacenado validara todas las variables, que tengan not null, y los deafault, en este caso principalmente para evitar complicaciones en el futuro, y de tener validadas a la hora de ingresar datos, no haiga ningún problema al respecto evitando problemas mayores al ingresar datos erróneos.

```
32 DELIMITER $$
33 CREATE PROCEDURE ALTA_TROFEOS_OF(in nombre_trofeo_p VARCHAR(100),in descripcion_trofeo_p VARCHAR(500),
34 in tipo_trofeo_p VARCHAR(100), in tipo_importancia_p VARCHAR(45)
35 )
36 BEGIN
37 -- Validar que los parámetros de entrada no sean nulos o vacios
38 IF nombre_trofeo_p IS NULL OR nombre_trofeo_p = '' THEN
39 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El nombre del trofeo es obligatorio.';
40 ELSE IF descripcion_trofeo_p IS NULL OR descripcion_trofeo_p = '' THEN
41 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La descripción del trofeo es obligatoria.';
42 ELSE IF tipo_trofeo_p IS NULL OR tipo_trofeo_p = '' THEN
43 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El tipo de trofeo es obligatorio.';
44 ELSE IF tipo_importancia_p IS NULL OR tipo_importancia_p = '' THEN
45 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El tipo de importancia es obligatorio.';
46 -- Validar que el valor de tipo_trofeo sea uno de los valores permitidos
47 ELSE IF tipo_trofeo_p NOT IN ('DINERO', 'TROFEOS', 'DINERO Y TROFEOS', 'RECONOCIMIENTOS') THEN
48 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El tipo de trofeo debe ser uno de los valores permitidos.';
49 -- Validar que el valor de tipo_importancia sea uno de los valores permitidos
50 ELSE IF tipo_importancia_p NOT IN ('ALTO', 'MEDIO', 'BAJO') THEN
51 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El tipo de importancia debe ser uno de los valores permitidos.';
52 END IF;
53 END IF;
54 END IF;
55 END IF;
56 END IF;
57 END IF;
58 -- Insertar el nuevo registro en la tabla TROFEOS_OFICIALES
59 INSERT INTO TROFEOS_OFICIALES(nombre_trofeo_of, descripcion_trofeo, tipo_trofeo, tipo_importancia)
60 VALUES(nombre_trofeo_p, descripcion_trofeo_p, tipo_trofeo_p, tipo_importancia_p);
61 END $$
62 DELIMITER ;
63 CALL ALTA_TROFEOS_OF("SPARTACUS","Es un trofeo nunca antes visto, y de un gran valor","DINERO","ALTO");
64 CALL ALTA_TROFEOS_OF("TORNATE","Es un trofeo nunca antes visto","TROFEOS","MEDIO");
65 CALL ALTA_TROFEOS_OF("CADELABRO","Sin comentarios...","DINERO Y TROFEOS","BAJO");
66 SELECT * FROM TROFEOS_OFICIALES;
```

PRUEBAS:

✖	56 19:24:43	CALL ALTA_TROFEOS_OF("","Sin comentarios...","DINERO Y TROFEOS","BAJO")	Error Code: 1644. El nombre del trofeo es obligatorio.
✖	57 19:25:47	CALL ALTA_TROFEOS_OF("TORPEDO","", "DINERO Y TROFEOS","BAJO")	Error Code: 1644. La descripción del trofeo es obligatoria.
✖	58 19:26:17	CALL ALTA_TROFEOS_OF("TORPEDO","EL MEJORRR","DINERO Y TROFEOS","BAJO")	Error Code: 1644. El tipo de trofeo es obligatorio.
✖	59 19:26:38	CALL ALTA_TROFEOS_OF("TORPEDO","EL MEJORRR","DINERO Y TROFEOS","BAJO")	Error Code: 1644. El tipo de trofeo debe ser uno de los valores permitidos.
✖	60 19:26:58	CALL ALTA_TROFEOS_OF("TORPEDO","EL MEJORRR","DINERO Y TROFEOS","")	Error Code: 1644. El tipo de importancia es obligatorio.
✖	61 19:27:03	CALL ALTA_TROFEOS_OF("TORPEDO","EL MEJORRR","DINERO Y TROFEOS","")	Error Code: 1644. El tipo de importancia es obligatorio.
✖	62 19:27:17	CALL ALTA_TROFEOS_OF("TORPEDO","EL MEJORRR","DINERO Y TROFEOS","BAJ")	Error Code: 1644. El tipo de importancia debe ser uno de los valores permitidos.



DASC

Procedimiento almacenado: este procedimiento almacenado servirá para validar las variables necesarias para insertar, que serían todas las variables que se mostraran a continuación, principalmente esto servirá para evitar problemas futuros a la hora de insertar datos erróneos, si todo esta correcto se lanzará un mensaje de registro exitoso.

```
DELIMITER $$
CREATE PROCEDURE ALTA_CAMPO(
    IN nombre_campo_p VARCHAR(45), IN calificacion_p FLOAT, IN estado_servicio_p VARCHAR(50),
    IN tipo_campo_p VARCHAR(100), IN hora_inicio_p TIME, IN hora_salida_p TIME, IN id_colonia_of_p INT)
BEGIN
    -- Validar el nombre del campo
    IF nombre_campo_p IS NULL OR nombre_campo_p = '' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El nombre del campo es obligatorio.';
    ELSEIF LENGTH(nombre_campo_p) > 45 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El nombre del campo no puede ser mayor a 45 caracteres.';
    END IF;

    -- Validar la calificación del campo
    IF calificacion_p IS NOT NULL AND (calificacion_p < 1 OR calificacion_p > 5) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La calificación del campo debe estar entre 1 y 5.';
    END IF;

    -- Validar el estado del servicio
    IF estado_servicio_p NOT IN ('LIBRE', 'OCUPADO') THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El estado del servicio debe ser LIBRE o OCUPADO.';
    END IF;

    -- Validar el tipo de campo
    IF tipo_campo_p NOT IN ('FUTBOL', 'FUTBOL RAPIDO', 'TENIS', 'BASQUETBOL', 'VOLEIBOL') THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El tipo de campo debe ser FUTBOL, FUTBOL RAPIDO, TENIS, BASQUETBOL o VOLEIBOL.';
    END IF;

    -- Validar la hora de inicio
    IF hora_inicio_p IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La hora de inicio es obligatoria.';
    END IF;

    -- Validar la hora de salida (si se proporciona)
    IF hora_salida_p IS NOT NULL AND hora_salida_p <= hora_inicio_p THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La hora de salida debe ser mayor que la hora de inicio.';
    END IF;

    -- Validar la colonia oficial
    IF NOT EXISTS (SELECT 1 FROM COLONIAS_OFICIALES WHERE id_colonia_oficial = id_colonia_of_p) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La colonia oficial proporcionada no existe.';
    END IF;

    -- Insertar el registro en la tabla si no hay errores
    INSERT INTO CAMPO (nombre_campo, calificacion, estado_servicio, tipo_campo, hora_inicio, hora_salida, id_colonia_of)
    VALUES (nombre_campo_p, calificacion_p, estado_servicio_p, tipo_campo_p, hora_inicio_p, hora_salida_p, id_colonia_of_p);
    SELECT "Registro insertado correctamente." AS mensaje;
END $$
DELIMITER ;

DROP PROCEDURE ALTA_CAMPO;
CALL ALTA_CAMPO("MIRAMAR",1,"LIBRE","FUTBOL","09:00:00",NULL,1);
CALL ALTA_CAMPO("MIRAMAR",4,NULL,"FUTBOL RAPIDO","08:00:00",NULL,6);
CALL ALTA_CAMPO("MIRAMAR",3,4,"", "VOLEIBOL","08:00:00",NULL,3);
CALL ALTA_CAMPO("SOL 1 ME 2",2,4,"LIBRE","FUTBOL RAPIDO","08:00:00","",2);
CALL ALTA_CAMPO("SOL 1 ME 2",1,4,"", "BASQUETBOL","08:00:00","",2);
```




DASC

PRUEBAS:

2	20:10:48	CALL ALTA_CAMPO('','4.5','','FUTBOL','09:00:00',NULL,1)	Error Code: 1644. El nombre del campo es obligatorio.
3	20:10:54	CALL ALTA_CAMPO(NULL,4.5','','FUTBOL','09:00:00',NULL,1)	Error Code: 1644. El nombre del campo es obligatorio.
4	20:11:02	CALL ALTA_CAMPO('MIRAMAR',0','','FUTBOL','09:00:00',NULL,1)	Error Code: 1644. La calificación del campo debe estar entre 1 y 5.
5	20:11:12	CALL ALTA_CAMPO('MIRAMAR',1','','FUTBOL','09:00:00',NULL,1)	Error Code: 1644. El estado del servicio debe ser LIBRE o OCUPADO.
6	20:11:19	CALL ALTA_CAMPO('MIRAMAR',1,'LIBRE','','09:00:00',NULL,1)	Error Code: 1644. El tipo de campo debe ser FUTBOL, FUTBOL RAPIDO, TENIS, BASQUETBOL o VOLEIBOL.
7	20:11:28	CALL ALTA_CAMPO('MIRAMAR',1,'LIBRE','FUTBOL','09:00:00',NULL,1)	1 row(s) returned
8	20:11:42	CALL ALTA_CAMPO('MIRAMAR',4,NULL,'FUTBOL RAPIDO','08:00:00','00:00:00',0)	Error Code: 1644. La hora de salida debe ser mayor que la hora de inicio.
9	20:12:02	CALL ALTA_CAMPO('MIRAMAR',4,NULL,'FUTBOL RAPIDO','08:00:00','','0)	Error Code: 1644. La hora de salida debe ser mayor que la hora de inicio.
10	20:12:08	CALL ALTA_CAMPO('MIRAMAR',4,NULL,'FUTBOL RAPIDO','08:00:00',NULL,0)	Error Code: 1644. La colonia oficial proporcionada no existe.

Procedimiento almacenado: en este procedure, se validan las entradas de nombre del equipo, director técnico y cantidad de jugadores, y se verifica la existencia del torneo especificado en el parámetro id_torneo_p. si alguna de estas validaciones falla, se envía un mensaje de error. si todas las validaciones pasan, se inserta el nuevo equipo en la tabla equipo.

```

DELIMITER %%
CREATE PROCEDURE ALTA_EQUIPO (IN nombre_equipo_p VARCHAR(45),IN dt_equipo_p VARCHAR(45),
IN cantidad_jugadores_p INT,IN id_torneo_p INT)
BEGIN
    DECLARE estado_registro_aux VARCHAR(2);
    DECLARE torneo_existe_aux INT;
    SELECT id_torneo INTO torneo_existe_aux FROM TORNEO WHERE id_torneo = id_torneo_p;
    -- Validar nombre de equipo
    IF nombre_equipo_p IS NULL OR nombre_equipo_p = '' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: El nombre del equipo no puede ser nulo o vacío';
    END IF;
    -- Validar director técnico del equipo
    IF dt_equipo_p IS NULL OR dt_equipo_p = '' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: El nombre del director técnico no puede ser nulo o vacío';
    END IF;
    -- Validar cantidad de jugadores del equipo
    IF cantidad_jugadores_p IS NULL OR (cantidad_jugadores_p < 7 OR cantidad_jugadores_p > 24) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: La cantidad de jugadores debe estar entre 7 y 24';
    END IF;
    -- Verificar existencia del torneo
    IF torneo_existe_aux IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: El torneo especificado no existe';
    END IF;
    -- Insertar nuevo equipo
    INSERT INTO EQUIPO (nombre_equipo, dt_equipo, cantidad_jugadores, fecha_registro, id_torneo)
    VALUES (nombre_equipo_p, dt_equipo_p, cantidad_jugadores_p, CURDATE(), id_torneo_p);
END %%
DELIMITER ;

```

PRUEBAS:

43	20:39:02	CALL ALTA_EQUIPO('','J',7,2)	Error Code: 1644. Error: El nombre del equipo no puede ser nulo o vacío
44	20:39:06	CALL ALTA_EQUIPO('GALLOS FC','J',7,2)	1 row(s) affected
45	20:39:11	CALL ALTA_EQUIPO('GALLOS FC','','7,2)	Error Code: 1644. Error: El nombre del director técnico no puede ser nulo o vacío
46	20:39:21	CALL ALTA_EQUIPO('GALLOS FC','JJ',1,2)	Error Code: 1644. Error: La cantidad de jugadores debe estar entre 7 y 24
47	20:39:29	CALL ALTA_EQUIPO('GALLOS FC','JJ',10,0)	Error Code: 1644. Error: El torneo especificado no existe
48	20:39:33	CALL ALTA_EQUIPO('GALLOS FC','JJ',10,1)	Error Code: 1062. Duplicate entry 'GALLOS FC' for key 'equipo.Unq_nombre_equipo'



DASC

```
369  /*ESTA FUNCION SERVIRA PARA SABER LA CANTIDAD DE CAMPOS
370  QUE EXISTEN EN UNA COLONIA DE MANERA OFICIAL, REGISTRADOS
371  EN NUESTRA PLATAFORMA PARA ASI LLEVAR UN REGISTRO
372  Y ESTAR MAS ORGANIZADOS, INGRESANDO EL
373  NOMBRE DEL CAMPO*/
374  DELIMITER $$
375  ● CREATE FUNCTION CANTIDAD_CAMPOS(nombre_campo_p VARCHAR(45))
376  RETURNS VARCHAR(45)
377  DETERMINISTIC
378  BEGIN
379  DECLARE cantidad_aux INT;
380  SELECT COUNT(*) INTO cantidad_aux FROM CAMPO
381  WHERE nombre_campo = nombre_campo_p;
382  RETURN cantidad_aux;
383  END $$
384  DELIMITER ;
385  -- +DROP FUNCTION CANTIDAD_CAMPOS;
386  SELECT CANTIDAD_CAMPOS("MIRAMAR") AS "EL TOTAL DE CAMPOS OFICIALES ES DE: ";
387  "
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



EL TOTAL DE CAMPOS OFICIALES ES
DE:

3



DASC

```
387 # -----
388 /*ESTA FUNCION ES PARA SABER CON EXACTITUD LA
389 CANTIDAD DE EQUIPOS QUE ESTAN REGISTRADOS
390 POR TORNEO PARA LLEVAR UN MEJOR FUNCIONAMIENTO
391 A LA HORA DE ORGANIZAR LOS DATOS A FUTUROS.
392 INGRESAND EL ID DEL TORNEO*/
393 DELIMITER $$
394 ● CREATE FUNCTION CANTIDAD_EQUIPOS_TORNEO(ID_torneo_p VARCHAR(45))
395 RETURNS INT
396 DETERMINISTIC
397 BEGIN
398 DECLARE cantidad_aux INT;
399 DECLARE nombre_aux VARCHAR(45);
400 SELECT COUNT(equipo.id_torneo) INTO cantidad_aux FROM EQUIPO where ID_TORNEO = ID_TORNEO_P;
401 RETURN cantidad_aux;
402 END $$
403 DELIMITER ;
404 ● SELECT NOMBRE_TORNEO, CANTIDAD_EQUIPOS_TORNEO(ID_TORNEO) FROM TORNEO;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	NOMBRE_TORNEO	CANTIDAD_EQUIPOS_TORNEO(ID_TORNEO)
▶	RELAMPAGO	3
	RELAMPAGO II	0
	RELAMPAGO VOLI	0

```
410 /*ESTA FUNCION SERVIRA PARA SOLICITAR LA CALIFICACION(ESTRELLAS)
411 QUE LLEVA CADA CAMPO REGISTRO DE MANERA OFICIAL EN
412 NUESTRA PAGINA, DONDE EVALUAREMOS LA OPINION O LA
413 ACEPTACION DE LOS USUARIOS HACIA LOS CAMPOS DEPORTIVOS
414 INGRESANDO EL ID DEL CAMPO, PARA SABER SU RESPECTIVA EVALUACION (CALIFICACION)*/
415 DELIMITER $$
416 ● CREATE FUNCTION CAMPO_CALIFICACION(id_campo_p INT)
417 RETURNS FLOAT
418 DETERMINISTIC
419 BEGIN
420 DECLARE calif_aux FLOAT;
421 SELECT calificacion INTO calif_aux FROM CAMPO
422 WHERE id_campo = id_campo_p;
423 return calif_aux;
424 END $$
425 DELIMITER ;
426 ● -- DROP FUNCTION CAMPO_CALIFICACION;
427 SELECT CAMPO_CALIFICACION(2) AS "LA CALIFICACION DE ESTE CAMPO ES DE: ";
428 SELECT * FROM CAMPO;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

▶	LA CALIFICACION DE ESTE CAMPO ES DE:
	4



DELIMITER ;

Result Grid									Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
	id_equipo	nombre_equipo	dt_equipo	cantidad_jugadores	estado_registro	fecha_registro	fecha_salida	id_torneo								
▶	1	PORCINOS	Ibai	7	INSCRITO	2023-05-25	NULL	1								
	2	GALLOS FC	JJ	10	INSCRITO	2023-05-25	NULL	1								
	3	MAQUINA VOLI	JOSE	12	INSCRITO	2023-05-25	NULL	1								
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL								



DASC

Datos al actualizar la tabla torneo.

```
460 • -- SET SQL_SAFE_UPDATES = 1;
461 UPDATE TORNEO SET fecha_final = '2023-07-24' WHERE id_torneo = 1;
462 • SELECT * FROM TORNEO;
463 • SELECT * FROM EQUIPO;
```

[illegible]

TRIGGER_ACT_CAMPO:

```
/*ESTE TRIGGER TENDRA LA FUNCION DE ACTUALIZAR LA COLUMNA ESTADO SERVICIO
DE LA TABLA CAMPO A OCUPADO, AL MOMENTO DE INSERTAR UN VALOR EN LA TABLA TORNEO
ESPECIFICANDO QUE CAMPOS SE JUGARA LOS TORNEOS, ASI PONIENDOLOS EN OCUPADO*/
DELIMITER $$
```

```
CREATE TRIGGER TRIGGER_ACT_CAMPO
AFTER INSERT ON TORNEO
FOR EACH ROW
BEGIN
    UPDATE CAMPO
    SET estado_servicio = 'OCUPADO'
    WHERE id_campo = NEW.id_campo;
END $$
DELIMITER ;
```

Datos sin actualizar aún.

```
477 • SELECT * FROM TORNEO;
478 • SELECT * FROM CAMPO;
479 -- INSERT INTO TORNEO(nombre torneo,tipo torneo,fecha inicio,cantidad equipos,id campo,ic
```

[illegible]



Datos al insertar a la tabla torneo.

```
77 • SELECT * FROM TORNEO;  
78 • SELECT * FROM CAMPO;  
79 • INSERT INTO TORNEO(nombre_torneo, tipo_torneo, fecha_inicio, cantidad_equipos, id_campo, id_trofeo_of)  
80 VALUES("CARP", "FUTBOL", "2023-07-09", 10, 1, 1);
```

result Grid							
Filter Rows:							
Edit: Export/Import: Wrap Cell Content:							
id_campo	nombre_campo	calificacion	estado_servicio	tipo_campo	hora_inicio	hora_salida	id_colonia_of
1	MIRAMAR	1	OCUPADO	FUTBOL	09:00:00	NULL	1
2	MIRAMAR	4	LIBRE	FUTBOL RAPIDO	08:00:00	NULL	1
3	MIRAMAR	3.4	LIBRE	VOLEIBOL	08:00:00	NULL	1
4	SOLI MEZ	2.4	LIBRE	FUTBOL RAPIDO	08:00:00	NULL	2
5	SOLI MEZ 2	1.4	LIBRE	BASQUETBOL	08:00:00	NULL	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Obviamente si se quiere ver más la diferencia, se tendría que colocar el trigger antes de que se crean los datos de la tabla torneo.



Anexos. (Capturas del diseño web del sistema)

Información de usuario antes de insertar uno nuevo.

```

250 # INSERT INTO JUGADOR(nickname,nombre_jugador,primer_ap,segundo_ap,fecha_nacimiento,email,password,fecha_entrada)
251 SELECT * FROM USUARIO;
252 -- DELETE FROM USUARIO WHERE id_usuario = 5;
253
254 -- DROP TABLE USUARIO;
  
```

id_usuario	nickname	nombre_usuario	primer_ap	segundo_ap	fecha_nacimiento	email	password	fecha_entrada	fecha_salida
1	MartinGG	Martin	Castro	Castro	2002-01-01	Martin@gmail.com	\$2b\$10\$4w38hQ75vYxvdaRPA9hge4dePwrXu...	2023-05-25	NULL
2	JazzielGod	Jazziel	Briones	Herrera	2002-01-01	Jazzielbh@gmail.com	\$2b\$10\$koLUZfJg3gE6i7aqDUlt4QrTvFv.CPSM...	2023-05-25	NULL
3	a	a	a	a	2002-01-01	ja@gmail.com	\$2b\$10\$Z3DZrLlBicKoVrBwEwkOwJa0Se0Hm...	2023-05-25	NULL
4	Jazzielgof	Jazziel	Briones	Herrera	2002-01-01	prueba@gmail.com	\$2b\$10\$7JMqE3w7MkhANSBjqGCqO.N8/bhNIFI...	2023-05-29	NULL
5	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Registrarse

REGISTRAR

¿Ya tienes una cuenta?

Inicia sesión y comienza a gestionar tus eventos deportivos

INICIAR SESIÓN



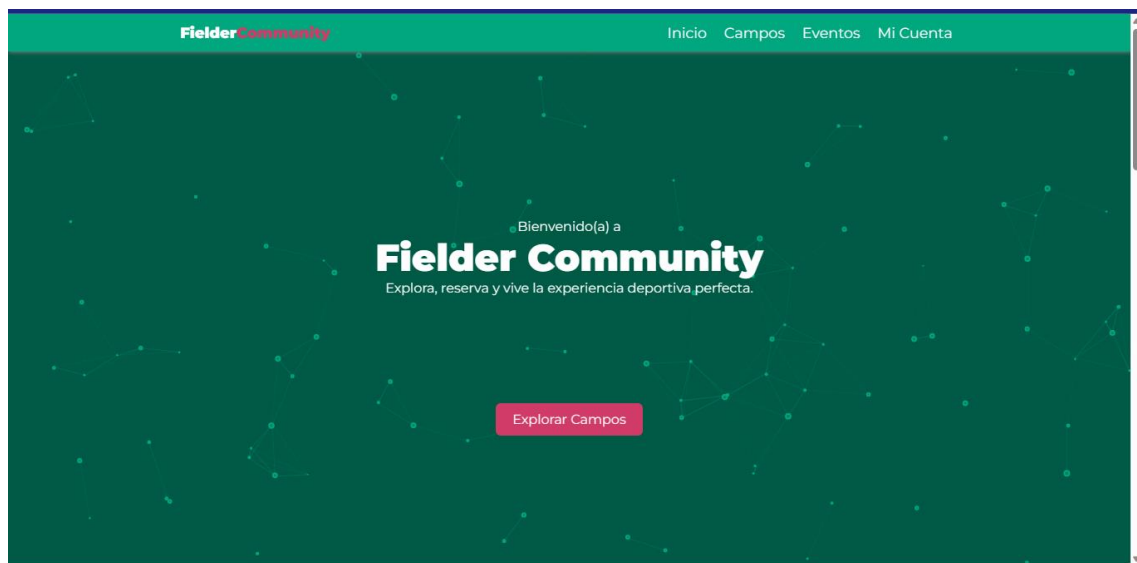
Información después de la ejecución.

```
250      # INSERT INTO JUGADOR(nickname,nombre_jugador,primer_ap,segundo_ap,fecha_nacimiento,email,password,fecha_entrada)
251 •    SELECT * FROM USUARIO;
252      -- DELETE FROM USUARIO WHERE id_usuario = 5;
253
254      -- DROP TABLE USUARIO;
```

Result Grid

	id_usuario	nickname	nombre_usuario	primer_ap	segundo_ap	fecha_nacimiento	email	password	fecha
▶	1	MartinGG	Martin	Castro	Castro	2002-01-01	Martin@gmail.com	\$2b\$10\$4w38hQ75vYxvdaFPYA9hge4dePwrXu...	2023-01-01
	2	JazzielGod	Jazziel	Briones	Herrera	2002-01-01	Jazzielbh@gmail.com	\$2b\$10\$koLUZfJg3gE6i7aqDIUt4OrTvFv.CPSM...	2023-01-01
	3	a	a	a	a	2002-01-01	ja@gmail.com	\$2b\$10\$Z3DZrTLIBicKovrBwEwkOw/a0Se0HMr...	2023-01-01
	4	jazzielgof	Jazziel	Briones	Herrera	2002-01-01	prueba@gmail.com	\$2b\$10\$JMQeJw7MkhAN5BjqGCqO.N8/bHN1Fi...	2023-01-01
	6	Prof_Irma	Irma	Lopez		2002-01-01	Irma@alu.uabcs.mx	\$2b\$10\$Wu2ArosnJaLeAY4n.cdpOz9rJbd2CW...	2023-01-01
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

USUARIO 4 x





BOASC

FielderCommunity[Inicio](#) [Campos](#) [Eventos](#) [Mi Cuenta](#)

Campos

 [Buscar](#)

Camp Nou
Disponible
Blvd. Abasolo, entre Calle Lima y Calle C
Calificación
[Reservar Ahora](#)
[Más Información](#)

Santiago Bernabeu
No Disponible
Blvd. Abasolo, entre Calle Lima y Calle C

[Reservar Ahora](#)
[Más Información](#)

Estadio Azteca
Disponible
Blvd. Abasolo, entre Calle Lima y Calle C
*
[Reservar Ahora](#)
[Más Información](#)

FielderCommunity[Inicio](#) [Campos](#) [Eventos](#) [Mi Cuenta](#)

Eventos

Torneo Coca Cola
Estadio UABCS
[Ver Bases](#)

Trofeo del barrio
Estadio Maracanã
[Ver Bases](#)

Miramar Tournament
Camp Nou
[Ver Bases](#)

Tenis tournament
Estadio VegasFC
[Ver Bases](#)

FielderCommunity

fieldercommunity@gmail.com
612-132-0728
612-132-0728
612-132-0728
612-132-0728
Carretera al sur KM 5.5, La Paz, Baja California Sur, México
Todos los derechos reservados a Fielder Community 2023©



DASC

Bienvenido a Fielder Community

Bienvenido(a) al mejor gestor de campos y torneos deportivos para la comunidad.

REGISTRARSE

Iniciar sesión

ENTRAR

FielderCommunity

Inicio Campos Eventos Mi Cuenta

Bienvenido(a) a

Fielder Community

Explora, reserva y vive la experiencia deportiva perfecta.

Explorar Campos



Conclusión.

Con esto, concluimos que la web de Fielder Community dará muchas oportunidades hacia la comunidad para la creación de eventos o torneos en la localidad, para que así se pueda impulsar el deporte.

Se usaron distintas formas y procedimientos para lograr esta meta, así que con esto logramos facilitar lo explicado.