[**Proyecto FixApp- Base de Datos 2**](#_gjdgxs)

[Introducción 2](#_30j0zll)

[Tipo de Negocio 2](#_1fob9te)

[Situación Problemática 2](#_3znysh7)

[Objetivo 2](#_2et92p0)

[**Diagrama Entidad-Relación (DER) 3**](#_tyjcwt)

[Tabla cod\_equipo 4](#_1z3ilwp8eg70)

[Tabla contacto 4](#_vs70nqgabclo)

[Tabla estados\_tickets 4](#_p8fvf41sjmnl)

[Tabla historial\_tickets 5](#_9mezjyu3gz4s)

[Tabla laboratorio 5](#_mze76q40s6y5)

[Tabla tickets 5](#_2di6u25admbb)

[Tabla tipo\_servicio 6](#_1tr977rnmk9f)

[Tabla usuario 6](#_hgsm73i3d8m7)

[Tabla tickets\_eliminados 6](#_jzv02ces36yy)

[Documentación del Trigger: 7](#_28gtd78bwg06)

[Código del trigger 8](#_a11z8ayitbto)

[**Justificación de la tecnología: 9**](#_35nkun2)

# Proyecto FixApp- Base de Datos

### Introducción

La base de datos para el proyecto FixApp está diseñada para gestionar de manera eficiente los tickets de soporte técnico en un entorno de laboratorio de informática. Este sistema permite rastrear y administrar los problemas reportados en equipos, así como mantener un historial detallado de cada ticket y las acciones realizadas por los técnicos.

### Tipo de Negocio

Soporte técnico y mantenimiento de equipos de laboratorio.

### Situación Problemática

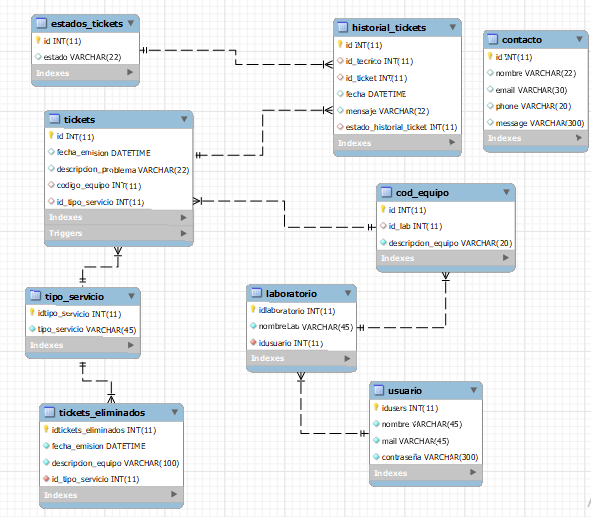
Mejorar la eficiencia en la gestión de tickets de soporte técnico, asegurando que todos los problemas reportados sean atendidos de manera oportuna y que exista un registro claro y completo de cada incidente y su resolución.

### Objetivo

Crear una base de datos que permita:

1. Registrar y gestionar tickets de soporte técnico.
2. Rastrear el estado y la resolución de cada ticket.
3. Mantener un historial detallado de las acciones realizadas sobre cada ticket.
4. Facilitar la asignación de técnicos a los tickets y el seguimiento de su desempeño

## Diagrama Entidad-Relación (DER)



**Descripción de Tablas**

### Tabla cod\_equipo

**Descripción**:  
Tabla que almacena los códigos de los equipos en los laboratorios, incluyendo su descripción y su relación con el laboratorio al que pertenecen.

**Campos**:

* id: Identificador único del código de equipo. (INT, PRIMARY KEY, AUTO\_INCREMENT)
* id\_lab: Identificador del laboratorio al que pertenece el equipo. (INT, FOREIGN KEY)
* descripcion\_equipo: Descripción del equipo. (VARCHAR(20))

### Tabla contacto

**Descripción**:  
Tabla que almacena la información de contacto recibida, como nombre, correo electrónico, teléfono y mensaje enviado por el usuario.

**Campos**:

* id: Identificador único del contacto. (INT, PRIMARY KEY, AUTO\_INCREMENT)
* nombre: Nombre del contacto. (VARCHAR(22))
* email: Correo electrónico del contacto. (VARCHAR(30))
* phone: Teléfono del contacto. (VARCHAR(20))
* message: Mensaje enviado por el contacto. (VARCHAR(300))

### Tabla estados\_tickets

**Descripción**:  
Tabla que almacena los diferentes estados que puede tener un ticket de soporte técnico.

**Campos**:

* id: Identificador único del estado del ticket. (INT, PRIMARY KEY, AUTO\_INCREMENT)
* estado: Descripción del estado del ticket (por ejemplo: abierto, cerrado, en progreso). (VARCHAR(22))

### 

### 

### Tabla historial\_tickets

**Descripción**:  
Tabla que almacena el historial de acciones realizadas sobre los tickets, incluyendo las actualizaciones de estado y las interacciones del técnico.

**Campos**:

* id: Identificador único del historial de ticket. (INT, PRIMARY KEY, AUTO\_INCREMENT)
* id\_tecnico: Identificador del técnico que realizó la acción. (INT, FOREIGN KEY)
* id\_ticket: Identificador del ticket al que pertenece el historial. (INT, FOREIGN KEY)
* fecha: Fecha y hora en que se realizó la acción. (DATETIME)
* mensaje: Descripción del mensaje o acción realizada. (VARCHAR(22))
* estado\_historial\_ticket: Estado del ticket en ese momento del historial. (INT, FOREIGN KEY)

### Tabla laboratorio

**Descripción**:  
Tabla que almacena la información sobre los laboratorios, incluyendo su nombre y el identificador del usuario responsable.

**Campos**:

* idlaboratorio: Identificador único del laboratorio. (INT, PRIMARY KEY, AUTO\_INCREMENT)
* nombreLab: Nombre del laboratorio. (VARCHAR(45))
* idusuario: Identificador del usuario responsable del laboratorio. (INT, FOREIGN KEY)

### 

### Tabla tickets

**Descripción**:  
Tabla que almacena los tickets de soporte técnico generados por problemas con los equipos. Cada ticket está relacionado con un equipo y un tipo de servicio.

**Campos**:

* id: Identificador único del ticket. (INT, PRIMARY KEY, AUTO\_INCREMENT)
* fecha\_emision: Fecha y hora de emisión del ticket. (DATETIME)
* descripcion\_problema: Descripción del problema reportado. (VARCHAR(22))
* codigo\_equipo: Identificador del equipo relacionado con el ticket. (INT, FOREIGN KEY)
* id\_tipo\_servicio: Identificador del tipo de servicio relacionado con el ticket. (INT, FOREIGN KEY)

### Tabla tipo\_servicio

**Descripción**:  
Tabla que almacena los diferentes tipos de servicios que se pueden proporcionar, como "limpieza" o "arreglo".

**Campos**:

* idtipo\_servicio: Identificador único del tipo de servicio. (INT, PRIMARY KEY, AUTO\_INCREMENT)
* tipo\_servicio: Descripción del tipo de servicio (por ejemplo: "limpieza", "arreglo"). (VARCHAR(45))

**Datos Iniciales**:

* Tipo de servicio "limpieza" (idtipo\_servicio=1)
* Tipo de servicio "arreglo" (idtipo\_servicio=2)

### 

### Tabla usuario

**Descripción**:  
Tabla que almacena la información de los usuarios del sistema, incluyendo su nombre, correo electrónico, y contraseña.

**Campos**:

* idusers: Identificador único del usuario. (INT, PRIMARY KEY, AUTO\_INCREMENT)
* nombre: Nombre del usuario. (VARCHAR(45))
* mail: Correo electrónico del usuario. (VARCHAR(45))
* contraseña: Contraseña cifrada del usuario. (VARCHAR(300))

### Tabla tickets\_eliminados

#### Descripción:

La tabla tickets\_eliminados almacena los registros de los tickets que han sido eliminados del sistema. Cada vez que un ticket se elimina, se guarda una copia de los datos clave (como la fecha de emisión, el código del equipo asociado y el tipo de servicio) en esta tabla para fines de auditoría, trazabilidad o recuperación de información histórica.

#### 

#### 

#### Campos:

1. **idtickets\_eliminados**
   * Tipo: INT(11)
   * Descripción: Identificador único del registro en la tabla tickets\_eliminados. Este campo es la clave primaria de la tabla.
   * Restricciones: NOT NULL, AUTO\_INCREMENT.
2. **fecha\_emision**
   * Tipo: DATETIME
   * Descripción: Fecha y hora en que se emitió originalmente el ticket antes de ser eliminado.
   * Restricciones: NOT NULL.
3. **codigo\_equipo**
   * Tipo: INT(11)
   * Descripción: Identificador del equipo relacionado con el ticket. Es una clave foránea que hace referencia a la tabla cod\_equipo.
   * Restricciones: NOT NULL.
   * Relación: Referencia a cod\_equipo(id).
4. **id\_tipo\_servicio**
   * Tipo: INT(11)
   * Descripción: Identificador del tipo de servicio asociado al ticket. Es una clave foránea que hace referencia a la tabla tipo\_servicio.
   * Restricciones: NOT NULL.
   * Relación: Referencia a tipo\_servicio(idtipo\_servicio).

### Documentación del Trigger:

#### Nombre del Trigger:

before\_delete\_tickets

#### Tipo de Trigger:

BEFORE DELETE

#### Tabla Asociada:

tickets

#### Descripción del Trigger:

Este trigger se ejecuta antes de que un registro de la tabla tickets sea eliminado. Su propósito es realizar una copia de los datos relevantes del ticket que está siendo eliminado y almacenarlos en la tabla tickets\_eliminados para mantener un historial de los tickets eliminados. En lugar de almacenar el identificador del equipo asociado con el ticket (codigo\_equipo), el trigger obtiene la descripción del equipo desde la tabla cod\_equipo y la guarda en la tabla tickets\_eliminados.

#### 

#### 

#### 

#### Funcionalidad del Trigger:

**Extracción de la descripción del equipo**:  
Cuando un ticket es eliminado de la tabla tickets, se extrae el valor de codigo\_equipo de la fila eliminada. Este valor es utilizado para buscar la descripción del equipo en la tabla cod\_equipo, a través de una consulta SELECT.

**Inserción de datos en la tabla tickets\_eliminados**:  
Los siguientes valores se insertan en la tabla tickets\_eliminados:

* + fecha\_emision: Fecha y hora en que se generó el ticket (tomado de la fila eliminada).
  + descripcion\_equipo: Descripción del equipo (obtenida de la tabla cod\_equipo mediante el codigo\_equipo del ticket).
  + id\_tipo\_servicio: Tipo de servicio asociado al ticket (tomado de la fila eliminada).

## Código del trigger

DELIMITER //

CREATE TRIGGER before\_delete\_tickets

BEFORE DELETE

ON tickets

FOR EACH ROW

BEGIN

DECLARE descripcion\_equipos VARCHAR(20);

-- Get the descripcion\_equipo from the cod\_equipo table

SELECT descripcion\_equipo

INTO descripcion\_equipos

FROM cod\_equipo

WHERE id = OLD.codigo\_equipo;

-- Insert the record into tickets\_eliminados

INSERT INTO tickets\_eliminados (fecha\_emision, descripcion\_equipo, id\_tipo\_servicio)

VALUES (OLD.fecha\_emision, descripcion\_equipos, OLD.id\_tipo\_servicio);

END;

//

DELIMITER ;

#### 

#### 

#### Descripción de los Elementos del Trigger:

**DELIMITER // y DELIMITER ;**:  
Estos comandos cambian el delimitador para poder definir el trigger sin que el sistema interprete el punto y coma (;) como final de la declaración. El delimitador se cambia a // para la creación del trigger y se restablece a ; después de finalizar.

**DECLARE descripcion\_equipos VARCHAR(20)**:  
Se declara una variable local llamada descripcion\_equipos de tipo VARCHAR(20) que almacenará la descripción del equipo que se obtiene de la tabla cod\_equipo.

**SELECT descripcion\_equipo INTO descripcion\_equipos**:  
Se realiza una consulta SELECT a la tabla cod\_equipo utilizando el valor de OLD.codigo\_equipo (el identificador del equipo del ticket que se está eliminando) para obtener la descripción del equipo. El resultado se almacena en la variable descripcion\_equipos.

**INSERT INTO tickets\_eliminados (fecha\_emision, descripcion\_equipo, id\_tipo\_servicio)**:  
Los valores de fecha\_emision, descripcion\_equipos y id\_tipo\_servicio se insertan en la tabla tickets\_eliminados. Estos valores corresponden a los datos de la fila del ticket que está siendo eliminado.

**OLD**:  
La palabra clave OLD se usa en triggers de tipo BEFORE DELETE para referirse a los valores de los campos de la fila antes de que se eliminen.

## Justificación de la tecnología:

Decidimos utilizar Sql debido a que consideramos que nuestra información sería más manejable si se encontraba estructurada, además , necesitábamos tener la capacidad de relacionar mucha información de manera sencilla.

Otra de las razones de nuestra elección es el gran soporte y comunidad que tiene MySql, siendo que esta es la base de datos más grande y usada del mundo.