

Parcial - Entregable

TC1 - Tecnicatura en Redes y Software - Instituto Tecnológico de Informática CETP.

Autores:

Brayan Rivero, Nicolas Gallardi.

Docente:

Martín Viar

Índice

Índice.....	2
Alcance y limitaciones.....	3
Propuesta/Escenario.....	3
Modelo Entidad Relación.....	3
Diagrama.....	4
Pasaje a tablas.....	4
SQL.....	4
Conclusión.....	6
Bibliografía.....	6

Alcance y limitaciones

Se desea modelar un sistema en el cual cada alumno sea capaz de reportar el estado de cada uno de los equipos del laboratorio, si funciona o no, y además poder dejar un comentario sobre el estado del equipo, también interesa un apartado administrativo donde los técnicos puedan observar el listado de equipos, pudiendo filtrar por fecha y estado, agregar equipos, importar usuarios y cambiar el estado de los equipos.

Propuesta/Escenario

Se propone implementar una solución web funcional en el contexto de instituciones educativas que cuenten con laboratorios informáticos. En este escenario, los estudiantes podrán registrar fallos o estados de los equipos de forma sencilla y validada, mientras que los administradores tendrán una herramienta centralizada para el seguimiento de incidencias y mantenimiento. A través de diagnósticos automáticos programados, el sistema facilitará la toma de decisiones técnicas, optimizando el tiempo de respuesta ante fallos y mejorando la disponibilidad de los recursos tecnológicos.

Modelo Entidad Relación

Entidades:

- Estudiante
 - id_estudiante (PK)
 - nombre
 - apellido
 - ci
 - email
- Equipo
 - id_equipo (PK)
 - nombre
- Registro
 - epochtime (PK compuesto)
 - id_equipo (FK a Equipo)
 - id_estudiante (FK a Estudiante)
 - estado
 - comentario

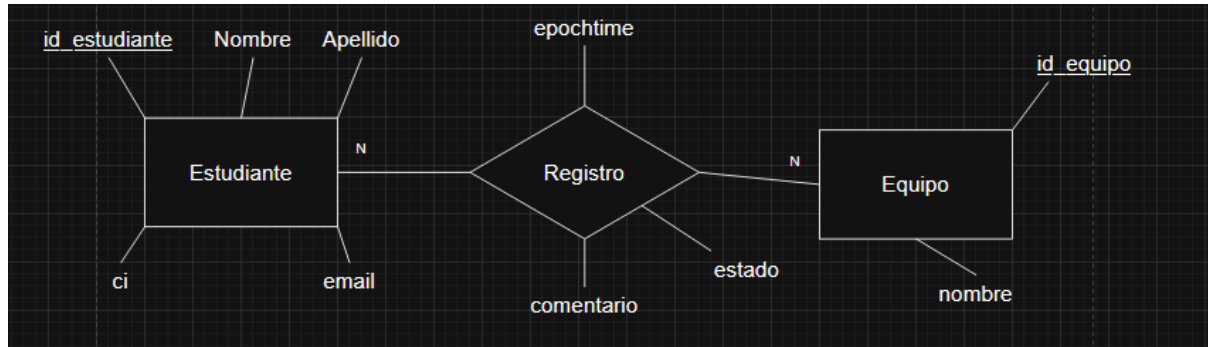
Relaciones:

- Un Estudiante puede realizar múltiples Registros (1:N).
- Un Equipo puede ser registrado múltiples veces por distintos estudiantes (1:N).
- La entidad Registro funciona como entidad débil con clave compuesta (epochtime, id_equipo, id_estudiante) y relaciones N:M resueltas en dos 1:N.

Restricciones:

- Estudiante (1) registra (N) Equipo.
- Equipo (1) registrado (N) Estudiante.

Diagrama



Pasaje a tablas

Equipo (id_equipo, nombre)

Estudiante (id_estudiante, nombre, apellido, ci, email)

Registro (epochtime, id_equipo, id_estudiante, estado, comentario)

SQL

-- Crear la base de datos si no existe y usarla

```
CREATE DATABASE IF NOT EXISTS BDDSGLAB6;
USE BDDSGLAB6;
```

-- Tabla equipo

```
CREATE TABLE IF NOT EXISTS equipo (
    id_equipo INT(3) NOT NULL AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    PRIMARY KEY (id_equipo)
);
```

-- Tabla estudiante

```
CREATE TABLE IF NOT EXISTS estudiante (
    id_estudiante INT(3) NOT NULL,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    ci INT(8) NOT NULL,
    email VARCHAR(50) NOT NULL,
```

```
PRIMARY KEY (id_estudiante)
);

-- Tabla registro
CREATE TABLE IF NOT EXISTS registro (
    epochtime INT(11) NOT NULL,
    id_equipo INT(3) NOT NULL,
    id_estudiante INT(3) NOT NULL,
    estado BOOLEAN NOT NULL,
    comentario VARCHAR(500),
    PRIMARY KEY (epochtime, id_equipo, id_estudiante),
    CONSTRAINT fk_registro_equipo FOREIGN KEY (id_equipo) REFERENCES
equipo(id_equipo),
    CONSTRAINT fk_registro_estudiante FOREIGN KEY (id_estudiante) REFERENCES
estudiante(id_estudiante)
);

-- Inserts para tabla equipo
INSERT INTO equipo (nombre) VALUES
('SGLAB6PC01'),
('SGLAB6PC02'),
('SGLAB6PC01');

-- Inserts para tabla estudiante
INSERT INTO estudiante (id_estudiante, nombre, apellido, ci, email) VALUES
(0, 'Administrador', 'UTU-ITI', 12345679, 'administradorutuiti@gmail.com'),
(1, 'Juan', 'Perez', 12345678, 'juanperez@gmail.com'),
(2, 'Maria', 'Gonzalez', 23456789, 'mariagonzalez@gmail.com'),
(3, 'Luis', 'Rodriguez', 34567890, 'luisrodriguez@gmail.com');

-- Inserts para tabla registro
INSERT INTO registro (epochtime, id_equipo, id_estudiante, estado, comentario) VALUES
(1718401000, 1, 1, TRUE, 'Funciona correctamente'),
(1718402000, 2, 2, FALSE, 'No conecta a Internet'),
(1718403000, 3, 3, TRUE, 'Funciona, no tiene PHP');

-- Crear el usuario para la pag web podemos agregar % luego del @ para que sean de
todas las conexiones
CREATE USER 'UserWeb'@'localhost' IDENTIFIED BY 'LosCosmicos2025';

-- Crear el usuario para los respaldos y conexiones administrativas
CREATE USER 'AdminBD'@'localhost' IDENTIFIED BY 'Admin1234';

GRANT ALL PRIVILEGES ON *.* TO 'AdminBD'@'localhost' WITH GRANT OPTION;

-- Le damos los privilegios justos
GRANT SELECT, INSERT, UPDATE, DELETE ON BDDSGLAB6.* TO
'UserWeb'@'localhost';
```

```
-- Aplicar los cambios  
FLUSH PRIVILEGES;
```

```
SELECT *  
FROM registro  
WHERE FROM_UNIXTIME(epochtime, '%Y-%m-%d') = CURDATE();
```

Conclusión

Se completó correctamente la implementación de la base de datos con sus requisitos.

Bibliografía

Repositorio github: <https://github.com/UTU-ITI/SGLAB>