

# Sistema de gestión de incidentes

La siguiente presentación documenta las tareas realizadas para el armado y configuración de una BD que busca manejar incidentes para un equipo de trabajo.

**Jejer Gustavo Javier**

# Objetivo

- ▶ Crear una BD consistente que pueda almacenar datos sobre incidentes que deben trabajar los equipos de TI, como así también la información histórica y una medición de los tiempos (KPI).

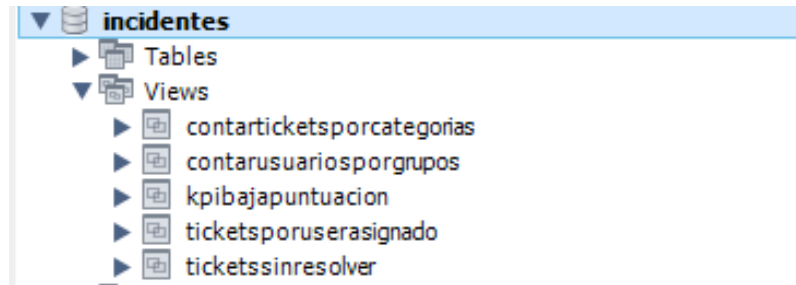
# Situación problemática

- ▶ Actualmente no se cuenta con una herramienta interna que pueda canalizar los tickets que los usuarios envían a TI por mail, por ello surge la búsqueda del desarrollo interno de una APP para la canalización y gestión de incidentes.

# Script de definición de BD

- ▶ Repositorio con todo el código utilizado:  
<https://github.com/Krieger93/IncidentesDB>
- ▶ Script completo de BD:  
[https://github.com/Krieger93/IncidentesDB/blob/master/incidentesDB\\_full\\_script.sql](https://github.com/Krieger93/IncidentesDB/blob/master/incidentesDB_full_script.sql)

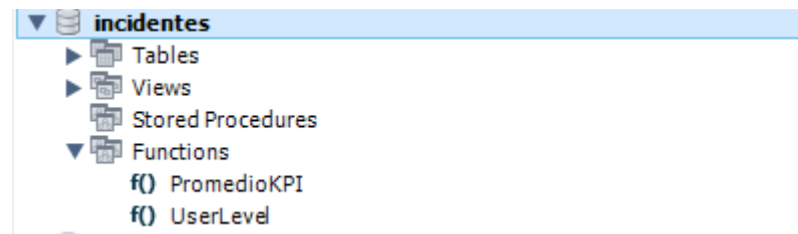
# Vistas



- ▶ **Contar tickets por categoría:** Devuelve el total de tickets que tiene asignada cada categoría. [Tablas: incidentes, categorias]
- ▶ **Contar usuarios por grupos:** Devuelve la cantidad de usuarios que tienen los grupos existentes. [Tablas: usuarios y grupos]
- ▶ **KPI baja puntuación:** Devuelve aquellos incidentes donde el puntaje del usuario es menor a lo esperado. [Tabla: KPIS]
- ▶ **Ticket por usuario asignado:** Join que devuelve los usuarios y sus tickets asignados. [Tablas: incidentes y usuarios]
- ▶ **Tickets sin resolver:** Devuelve el listado de tickets que tengan cualquier estado excepto “Solved”. [Tabla: incidentes]

# Funciones

- ▶ **Promedio KPI:** Devuelve el promedio del puntaje otorgado por cada usuario en los incidentes.
- ▶ **User level:** Ingresando el tipo de nivel del usuario nos devuelve si este es admin o no.



# Stored Procedures

- ▶ **SP\_ordenarinc:** Ordena la tabla incidentes en base a la columna dada y al orden solicitado.
- ▶ **SP\_updateinestado:** Actualiza el estado de un incidente
- ▶ [REPO](#)

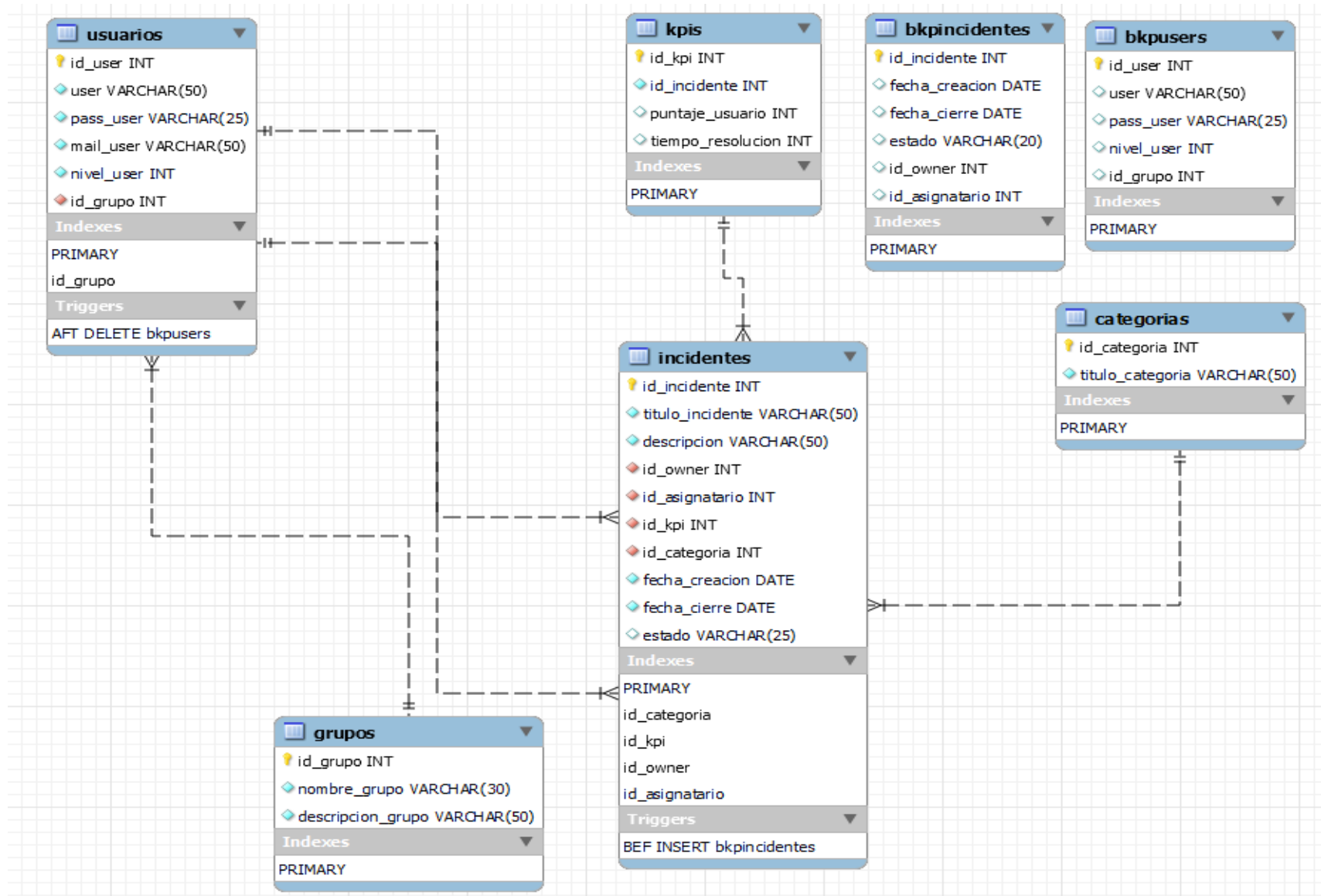
```
SP_ordenarinc

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `SP_ordenarinc`(IN campo VARCHAR(30), IN orden VARCHAR(4))
2 BEGIN
3     -- DECLARE clausula varchar(50);
4     DECLARE clausula varchar(50);
5     -- @clausula ES LA CLAUSULA FINAL A EJECUTAR, EN LA QUE SE ENCADENAN LA SENTENCIA SQL Y LAS VARIABLES
6     SET @clausula = concat('SELECT * FROM incidentes ORDER BY ', campo, ' ', orden);
7     -- SE PREPARA, EJECUTA Y CIERRA LA SENTENCIA
8     PREPARE ejecutar FROM @clausula;
9     EXECUTE ejecutar;
10    DEALLOCATE PREPARE ejecutar;
11 END
```

```
SP_updateinestado

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `SP_updateinestado`(IN i_id INT, IN i_estado varchar(25))
2 BEGIN
3     UPDATE incidentes
4     SET
5     estado = i_estado WHERE id_incidente = i_id;
6 END
```

# Estructura de Base de Datos





# Herramientas y tecnologías utilizadas

- ▶ Se implementará una interfaz grafica basada en la web (HTML, CSS, Javascript), la BD será almacenada en un nodo independiente a la interfaz con MYSQL allí instalado.