

# Desafío Desarrollador GSTI

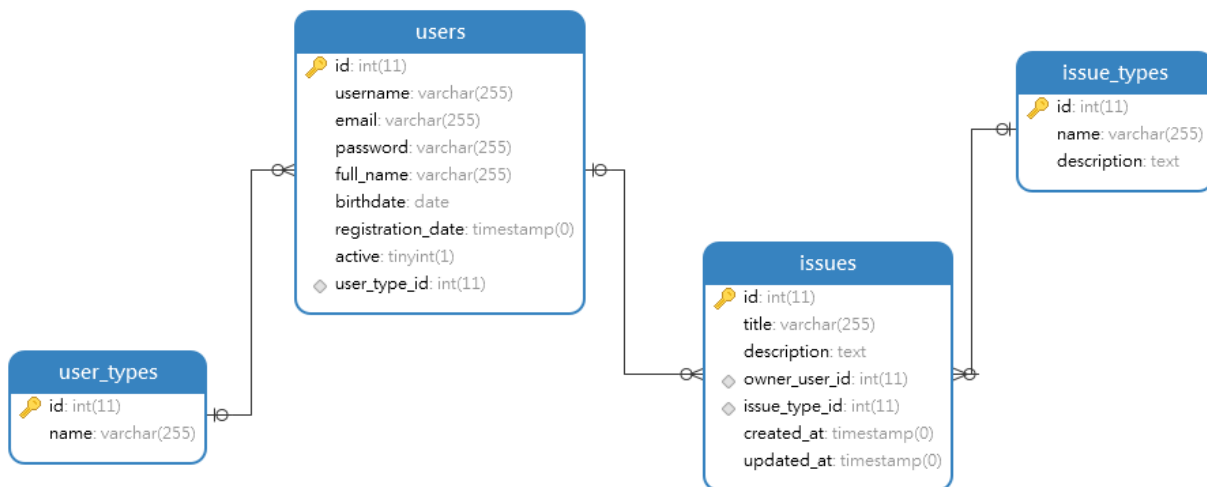
## Contexto:

La empresa **TechResolve Solutions**, especializada en soluciones tecnológicas y soporte, está enfrentando una alta demanda de issues (tickets) que necesitan ser atendidos por su equipo de agentes de soporte técnico. Cada issue generado por los usuarios tiene características diferentes y requiere la intervención de un agente especializado, asignado según el tipo de issue reportado.

En este escenario, la dirección de **TechResolve Solutions** identifica la necesidad de optimizar el proceso de asignación de issues y mejorar la gestión y consulta de los mismos para optimizar los tiempos de respuesta y la eficiencia del equipo de soporte. Para la prueba técnica, no se espera que el postulante cree algún código de automatización integral; más bien, se enfocará en realizar la asignación de issues a agentes a través de consultas SQL o código en PHP, estableciendo así una base sólida para futuras mejoras y automatizaciones en el sistema.

## Desafío 1 - Desarrollo y Consulta de Base de Datos

El Modelo de datos (simplificado) es el siguiente:



### 1.1 Asignar los issues a un agente (assigned) por issue\_type\_id

- Issues: (title, description, owner\_user\_id, issue\_type\_id, created\_at, updated\_at)

Condiciones:

- issue\_type\_id: 1 ➡ assigned\_user\_id: 1
- issue\_type\_id: 2 ➡ assigned\_user\_id: 2
- issue\_type\_id: 3 ➡ assigned\_user\_id: 3
- issue\_type\_id: 4 ➡ assigned\_user\_id: 4
- issue\_type\_id: 5 ➡ assigned\_user\_id: 5

### 1.2 Consultar los tickets que fueron cerrados el ultimo mes.

Condiciones:

- Deberá agregar un campo en la tabla issues

### 1.3 Cantidad de issues asignados, independiente si está abierto o cerrado por cada agente filtrado en la ultima semana

- Agregar información que requiera para construir la consulta según su criterio.

## Desafío 2: Exposición y Consumo de Servicios Web

### 2.1 Exponer servicio (endpoint)

Seleccionar una de las consultas anteriores y exponer un endpoint para consumirlo.

Condiciones:

- Manejo de http status code 400 (Bad Request) y 200 (Ok)
- Exponer con formato json

### 2.2 Consumir el endpoint en un archivo `.js` con `fetch()`

- Usar el archivo `./js/app.js`

### 2.3 Mostrar salida

- Archivo Desafio2.html
- Construir salida en tabla > tbody #issues

## Desafío 3.

### 3.1 Consultar api de GIPHY

Condiciones:

- El postulante puede utilizar las librerías `guzzleHttp` o `curl` para el consumo de la API.
- Se tiene que descomponer la url del endpoint y obtener las urls gifs según un dato de entrada.
- Obtener url original del gif en la respuesta de la API.