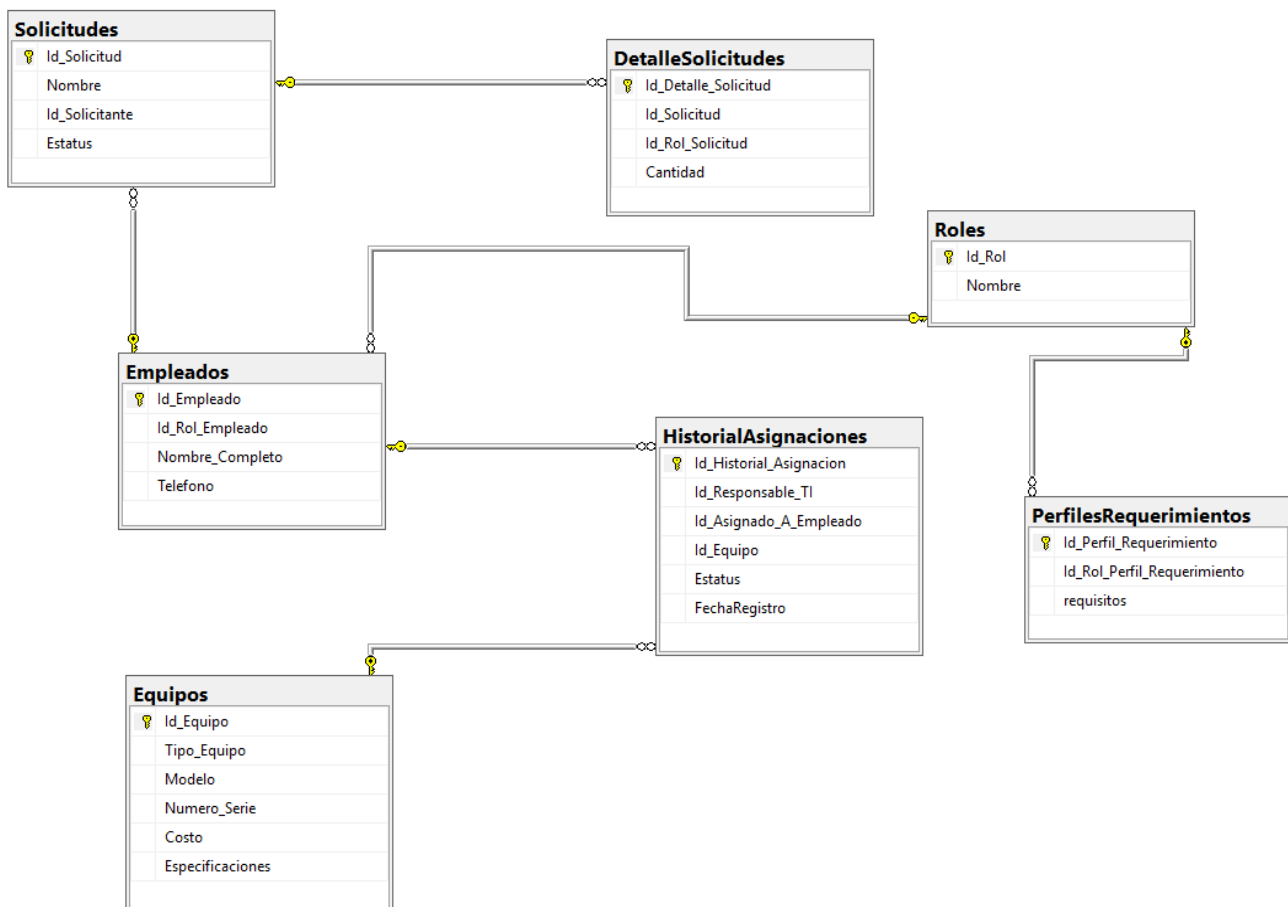


## Arquitectura

Backend: Django con Django Rest Framework(DRF) tiene varias ventajas ya que por estar en Python es muy fácil de aprender, existen muchas librerías para inteligencia artificial que podría ayudar a crear funciones más avanzadas. Su instalación es muy sencilla, hacer cambios también es muy fácil. DRF esta hecho para ser escalable.

Frontend: VUE.JS es una excelente opción para usar con DRF ya que tienen compatibilidad entre ellas. Programar no se vuelve tan tedioso y además es Single Page Applications(SPA). Se pueden crear sistemas dinámicos y útiles, ya que es un framework de "JS"

## Diagrama de Base de Datos



## Algoritmo de Optimización

1. **Declaración de Variables** Se definen variables para manejar conteos, identificadores, nombres de roles, cantidades y tipos de equipo.
2. **Tabla de Prioridades** Se crea la tabla #Prioridades con los roles en orden de prioridad. Se usa para asignar primero a los roles más importantes.
3. **Tabla de Solicitudes** Se crea la tabla #Solicitudes con solicitudes activas del ID proporcionado, junto con su prioridad según el rol.
4. **Tabla de Requerimientos por Perfil** Se crea #Requerimientos\_Perfiles, que contiene los equipos requeridos por cada rol, usando datos en formato JSON desde PerfilesRequerimientos.
5. **Tabla de Historial de Equipos Asignados** Se registra el equipo que actualmente está asignados.
6. **Tabla de Equipos Disponibles** Se crea #Sin\_Asignar con los equipos que no están en uso, ordenados por costo descendente y tipo de equipo.
7. **Tablas de Resultado Final**
  - #Con\_Equipo: Equipos que se lograron asignar a personas por rol.
  - #Sin\_Equipo: Equipos que no se pudieron asignar y muestra a quien no se le asigno.
8. **Ciclo de Asignación** Se recorre cada solicitud, cada persona y cada objeto requerido:
  - Se crea una lista temporal #Lista\_Equipo con los requerimientos del rol.
  - Por cada tipo de equipo, se intenta asignar el equipo disponible más caro.
  - Si se encuentra equipo, se inserta en #Con\_Equipo. Si no, en #Sin\_Equipo.
9. **Resultados y Limpieza** Se muestran los resultados (SELECT) y se eliminan las tablas temporales. La transacción se confirma con COMMIT.
10. **Manejo de Errores** Si ocurre un error, se hace un ROLLBACK y se lanza un mensaje de error con RAISERROR.