- Leonardo Dufrey Mejía Mejía 23648
- **¿Cómo decidimos qué entidades y relaciones incluir?**
  Nos enfocamos en representar lo más esencial del sistema: maquinaria, mantenimientos, técnicos y repuestos. Elegimos estas entidades porque eran las más relevantes para llevar un control claro y ordenado del mantenimiento. Simplificamos algunos detalles para no hacer el modelo demasiado complejo, agregamos un CRUD para cada tabla para practicar y también nos permitía insertar datos y rastrear un poco más.

- **¿Qué tan adecuadas fueron las claves primarias y foráneas?**
  Fueron adecuadas porque permitieron conectar bien las tablas y mantener la integridad de los datos. Gracias a ellas, las relaciones entre mantenimientos, maquinaria y técnicos se manejaron sin errores y las consultas fueron claras y precisas.

- **¿Aplicamos la normalización y cómo nos fue?**
  Sí, aplicamos hasta la tercera forma normal (3FN). Esto nos ayudó a evitar datos duplicados y a tener un modelo más limpio. En algunos casos, pensamos en desnormalizar por rendimiento, pero al final, la normalización fue suficiente para este proyecto.

- **¿Qué restricciones y reglas implementamos en la base de datos?**
  Usamos restricciones como not null y unique y ciertos check con enum y claves foráneas para asegurar que los datos ingresados fueran válidos. Estas reglas evitaron errores y ayudaron a que la base de datos se comportara de forma coherente con las reglas del negocio.

- **¿Ventajas o desventajas del modelo al hacer consultas complejas?**
  El modelo fue bastante útil y flexible. Nos permitió hacer consultas con filtros, uniones y agrupaciones sin mayores problemas. Al estar bien estructurado, las consultas eran claras y el rendimiento fue aceptable. Como problema tuvimos principalmente las fechas al colocar un rango de fechas nos generaba un error que no logramos identificar directamente porque tanto el frontend como el backend tenían el mismo formato de fecha.

- **¿Qué cambiaríamos para escalar a producción?**
  Si el sistema creciera, pensaríamos en optimizar índices, dividir la base en módulos o usar bases de datos más robustas. También revisaríamos el rendimiento con grandes volúmenes de datos y aplicaríamos técnicas de caché o réplicas si fuera necesario.