**e) Se ha identificado el sistema gestor de datos adecuado a cada sistema ERP-CRM. -**

**Sistema de gestión de datos relacionales.**

Aunque podemos encontrar diferentes tipos de datos de bases de datos, hasta ahora las que hemos usado son de tipo relacional, es decir los datos están representados en tablas. Cada tabla tendrá una clave primaria que permitirá la búsqueda de un dato concreto y será el puente para hacer la relación de este dato con otra información recogida en otras tablas.

Por ejemplo, podemos tener una tabla con los nombres de los clientes. En ella se recogerán los campos de:

* *IdentificadorNombres*
* Nombre
* Apellidos

En otra tabla tendremos la dirección de los clientes y se recogerá

* IdentificadorDirección
* *IdentificadorNombres*
* Calle
* Ciudad
* Código postal

De esta forma, la base de datos relacional nos permite tener una tabla con los datos y a su vez introducir un identificador como clave primaria y foránea que nos mantenga la relación entre una tabla y otra.

Para la consulta de los datos se usa SQL como lenguaje.

Entre los gestores actuales más populares existen:[4](https://es.wikipedia.org/wiki/Base_de_datos_relacional#cite_note-4)​

* [Microsoft SQL Server](https://es.wikipedia.org/wiki/Microsoft_SQL_Server).
* [Oracle](https://es.wikipedia.org/wiki/Oracle).
* [DB2](https://es.wikipedia.org/wiki/DB2).
* [PostgreSQL](https://es.wikipedia.org/wiki/PostgreSQL).
* [MariaDB](https://es.wikipedia.org/wiki/MariaDB)
* [MySQL](https://es.wikipedia.org/wiki/MySQL).

Para este proyecto estamos usando MariaDB que es una alternativa a MySQL de código abierto y gratuita que ofrece soporte para varios tipos de datos mediante múltiples motores de almacenamiento.

Se usa mediante la instalación de XAMPP

Este tipo de bases de datos nos va a ayudar a mantener la integridad de los datos y evitar fallos por duplicidad. No obstante, si proveemos que nuestra aplicación va a experimentar un rápido crecimiento en cuanto a la cantidad de los datos a manejar, quizás nos convendría usar un tipo de base de datos no relacional como MongoDB.

Para la interacción con la base de datos podemos crear pequeños microservicios que nos permitan dividir la aplicación en pequeñas partes y así hacerla mas escalable. Por ejemplo, en nuestra app podemos tener una parte para los datos del cliente, otra para los datos de los trabajadores, otra para el catalogo de productos, para los pedidos, etc.

Dividir la app en microservicios nos permite obtener un desarrollo fullstack sencillo, pudiendo hacer tanto la parte del frontend y todo el tema visual de la app como desarrollar una base de datos sencilla que nos permita acceder de forma útil a la información que se pide desde el frontend.