

## Tarea Individual 1, Módulo 7

- ¿Cuáles son las bases de datos soportadas por Django y en qué se diferencian?

- PostgreSQL. Base de datos relacional robusta y altamente escalable, con funciones avanzadas. Puede ser un poco más lenta.
- MariaDB. Fork de código abierto de MySQL, compatible con la mayoría de las aplicaciones de MySQL y con características adicionales.
- MySQL. Base de datos relacional popular, conocida por su escalabilidad y amplio soporte de la comunidad.
- Oracle. Base de datos empresarial potente y completa, ampliamente utilizada en entornos de gran escala y de misión crítica.
- SQLite. Base de datos ligera y de fácil configuración que se almacena en un archivo local, adecuada para aplicaciones pequeñas o de desarrollo.
- Y varias otras, soportadas a través de backends third-party

- ¿Qué es una migración en Django y para qué se utiliza?

Una migración es un sistema automatizado para manejar la creación y alteración de bases de datos sin necesidad de escribir comandos ORM, asegurando la sincronía entre los modelos y los esquemas, mientras se evita repetir código de manera tediosa.

- ¿Cuál es la diferencia entre usar consultas SQL y consultas ORM en Django?

Las consultas SQL son comandos que se realizan directamente por el programador sobre la base de datos, y las consultas ORM las hace django según lo definido por el programador en la aplicación.

- ¿Cómo se instalan los paquetes de base de datos en Django y cuál es su importancia?

Utilizando pip u otro gestor de paquetes dentro del entorno de desarrollo. La importancia de hacerlo es que los paquetes y sus versiones se pueden reproducir en otras máquinas manteniendo la compatibilidad.

- ¿Qué ventajas ofrece Django como ORM para la integración con una base de datos?

1. Sintaxis simple. La sintaxis del ORM es más sencilla y fácil de leer, debido a su nivel de abstracción más elevado.
2. Independencia de la base de datos. Al escribir usando ORM, el lenguaje está unificado independiente de la base de datos que utilicemos.
3. Paradigma de la programación orientada a objetos. Con ORM todo nace de un modelo asociado a una clase.
4. Generación automática de queries con ORM, independiente de la operación.
5. Abstracción de los detalles de bajo nivel: El ORM elimina las interacciones de bajo nivel al abstraerlas, lo que permite enfocarse en otros aspectos.
6. Migraciones incorporadas: ORM sincroniza nuestros modelos con la base de datos de forma automática.