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

- Django soporta varios tipos de bases de datos, entre las que se incluyen: SQLite, PostgreSQL, MySQL, Oracle y Microsoft SQL Server. Cada una de estas bases de datos tiene sus propias características y ventajas. SQLite es una opción ligera y fácil de usar, ideal para aplicaciones pequeñas. PostgreSQL es una base de datos robusta y escalable, recomendada para aplicaciones de gran tamaño. MySQL es otra opción popular, conocida por su rendimiento y facilidad de uso. Oracle y Microsoft SQL Server son opciones más adecuadas para aplicaciones empresariales que requieren características avanzadas y alta disponibilidad.

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

- Una migración en Django es un proceso que permite modificar la estructura de la base de datos de manera controlada y sin perder los datos existentes. Las migraciones se utilizan para crear, modificar o eliminar tablas, columnas o restricciones en la base de datos. Además, las migraciones también se encargan de mantener un historial de los cambios realizados en la estructura de la base de datos, lo que facilita la colaboración entre desarrolladores y el mantenimiento del proyecto a largo plazo.

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

- La diferencia entre usar consultas SQL y consultas ORM en Django radica en la forma en que se interactúa con la base de datos. Las consultas SQL implican escribir sentencias SQL directamente para realizar operaciones en la base de datos. Por otro lado, las consultas ORM en Django permiten interactuar con la base de datos utilizando el lenguaje de programación Python, a través de modelos y consultas generadas automáticamente. Las consultas ORM ofrecen una abstracción más alta y facilitan la escritura de consultas de manera más legible y mantenible.

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

- Para instalar paquetes de base de datos en Django, se debe especificar el paquete correspondiente en el archivo requirements.txt del proyecto y luego ejecutar el comando pip install -r requirements.txt para instalar todas las dependencias. La importancia de los paquetes de base de datos en Django radica en que permiten la conexión y comunicación con la base de datos elegida, proporcionando las herramientas necesarias para realizar operaciones CRUD (crear, leer, actualizar, eliminar) en la base de datos desde la aplicación Django.

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

- Django ofrece varias ventajas como ORM para la integración con una base de datos. Algunas de estas ventajas incluyen:

- Abstracción de la base de datos: Django proporciona una capa de abstracción que permite interactuar con diferentes bases de datos utilizando el mismo código de aplicación, lo que facilita el cambio de base de datos sin tener que modificar el código.

- Generación automática de consultas: Django genera automáticamente consultas SQL basadas en los modelos y consultas definidas en el código de la aplicación, lo que simplifica la escritura de consultas y reduce la posibilidad de errores.

- Seguridad: Django incluye medidas de seguridad integradas para prevenir ataques de inyección SQL y otros problemas de seguridad comunes en las consultas a la base de datos.

- Mapeo objeto-relacional: Django mapea automáticamente los objetos Python a las tablas de la base de datos, lo que facilita el acceso y manipulación de los datos de la base de datos utilizando objetos y métodos Python familiares.