***Lenguaje de programación***  
Se utilizará **Python con el framework Django**, dado que permite un desarrollo ágil de aplicaciones web con estructura Modelo-Vista-Controlador (MVC), integración nativa con bases de datos, seguridad incorporada y escalabilidad.

***Base de datos (BDD)***Se empleará una **base de datos relacional (MySQL)**, ya que el sistema requiere manejar información estructurada (pedidos, choferes, clientes, vehículos, estados). El modelo relacional facilita integridad referencial y consultas complejas para reportes.

***Servidor (Hosting o Cloud, no local)***  
La aplicación será desplegada en **Amazon Web Services (AWS)**, específicamente en una instancia **EC2 con Ubuntu Server**, lo que garantiza disponibilidad y escalabilidad. Además:

* **RDS (Relational Database Service)** para alojar la base de datos MySQL en la nube.
* **S3 (Simple Storage Service)** para almacenamiento de archivos, como evidencia de entrega (fotos o firmas digitales).

***Arquitectura / Infraestructura***  
La arquitectura propuesta es **cliente-servidor en la nube** bajo un esquema de **3 capas**:

1. **Capa de presentación**: interfaz web y versión móvil responsive para clientes, choferes y administradores.
2. **Capa de aplicación**: servidor Django en AWS EC2, donde se gestionan las reglas de negocio (pedidos, asignaciones, estados, notificaciones).
3. **Capa de datos**: MySQL en AWS RDS y archivos en S3.

La infraestructura contempla:

* Balanceo de carga y escalabilidad automática en AWS.
* Seguridad con HTTPS y control de acceso por roles (cliente, chofer, administrador).
* Posibilidad de futuras integraciones con servicios de mapas/GPS vía API (ej: Google Maps).