**Ciclos de Vida de Software:**

**Planning, Analysis, Design, Implementation, Testing, Maintenance**

**Planning**:  
En esta etapa se definen los objetivos, alcances y recursos del proyecto mediante reuniones con stakeholders. Se establece un cronograma inicial y se priorizan las funcionalidades clave para garantizar que el desarrollo se alinee con las metas estratégicas del negocio.

**Analysis**:  
Se validan los requisitos funcionales y no funcionales del software mediante entrevistas, encuestas o talleres con usuarios finales. También se documentan casos de uso, reglas de negocio y restricciones técnicas para asegurar una comprensión clara de las necesidades del sistema.

**Design**:  
Se diseña la arquitectura del software, definiendo módulos, componentes y patrones de diseño (como MVC o microservicios). Además, se crean diagramas UML, wireframes o prototipos para visualizar la estructura y facilitar la comunicación entre equipos técnicos y no técnicos.

**Implementation**:  
Aquí se traduce el diseño en código usando lenguajes y frameworks específicos (como Python/Django o JavaScript/React). Los equipos siguen prácticas como revisiones de código y pair programming para garantizar calidad y coherencia en la base de código.

**Testing**:  
Se ejecutan pruebas unitarias, de integración y de usuario para detectar errores o inconsistencias. Herramientas como Selenium, JUnit o pruebas manuales aseguran que cada funcionalidad cumpla con los requisitos antes del despliegue.

**Maintenance**:  
Incluye corrección de errores post-lanzamiento, actualizaciones de seguridad y optimización del rendimiento. También implica adaptar el software a nuevos requisitos del usuario o cambios tecnológicos para prolongar su vida útil.