

INTRODUCCIÓN

MANTENIBILIDAD

Daniel Blanco Calviño

EL COSTE DEL SOFTWARE

- El **desarrollo inicial** hasta una primera versión **es sólo una fracción del coste**.
 - Investigar problemas.
 - Solucionar bugs.
 - Mantener los sistemas funcionando.
 - Migraciones.
 - Añadir o modificar funcionalidades.

EL COSTE DEL SOFTWARE

- El **desarrollo inicial** hasta una primera versión **es sólo una fracción del coste**.
 - Investigar problemas.
 - Solucionar bugs.
 - Mantener los sistemas funcionando.
 - Migraciones.
 - Añadir o modificar funcionalidades.
- Si queremos que nuestro producto prospere **debemos hacerlo sencillo de mantener**.
 - En el peor de los casos, no hacerlo puede resultar en el abandono del producto.

CARACTERÍSTICAS DE UN SISTEMA MANTENIBLE

- Observabilidad.
- Simplicidad.
- Extensibilidad.

CARACTERÍSTICAS DE UN SISTEMA MANTENIBLE

- **Observabilidad.**
 - Facilitar el trabajo del equipo de operaciones.
- **Simplicidad.**
 - Facilitar a los nuevos ingenieros el entender y aportar a nuestro sistema.
- **Extensibilidad.**
 - Facilitar futuros cambios en nuestro sistema.

OBSERVABILIDAD

- Dependiendo de la empresa, **el trabajo de operaciones puede ser llevado por el equipo de desarrollo o por un equipo diferente.**

OBSERVABILIDAD

- Dependiendo de la empresa, **el trabajo de operaciones puede ser llevado por el equipo de desarrollo o por un equipo diferente.**
- Las **responsabilidades** son las mismas:
 - **Monitorizar** la salud del sistema y **restaurarlo** lo antes posible si sucede algún problema.
 - **Investigar la razón de los problemas** (fallos, degradación del rendimiento)
 - Mantener el software y las plataformas donde se despliega el mismo actualizadas.
 - **Anticipar futuros problemas** que pueden ocurrir.
 - **Tareas de mantenimiento** complejas, como migraciones a plataformas o servicios de terceros distintos.

HACIENDO LA VIDA MÁS SENCILLA AL EQUIPO DE OPS.

- Dar visibilidad a la información interna del sistema con **buena monitorización**.
- Ofrecer soporte para la automatización y la integración del código.
- **Evitar** lo máximo posible **depender de máquinas individuales**.
- **Crear buena documentación** sobre el sistema.
- Dar opción a los administradores de modificar el estado del sistema.
- Intentar que nuestro sistema sea lo más **predecible** posible, **minimizando las sorpresas**.

SIMPLICIDAD

- Según el código crece, la calidad del mismo tiende a empeorar.
- Es importante mantener el **código lo más simple y limpio posible**.
 - Los nuevos integrantes tardarán más en empezar a aportar valor.
 - El equipo en general disfrutará menos trabajando.
 - Aumenta la probabilidad de crear nuevos **bugs**.

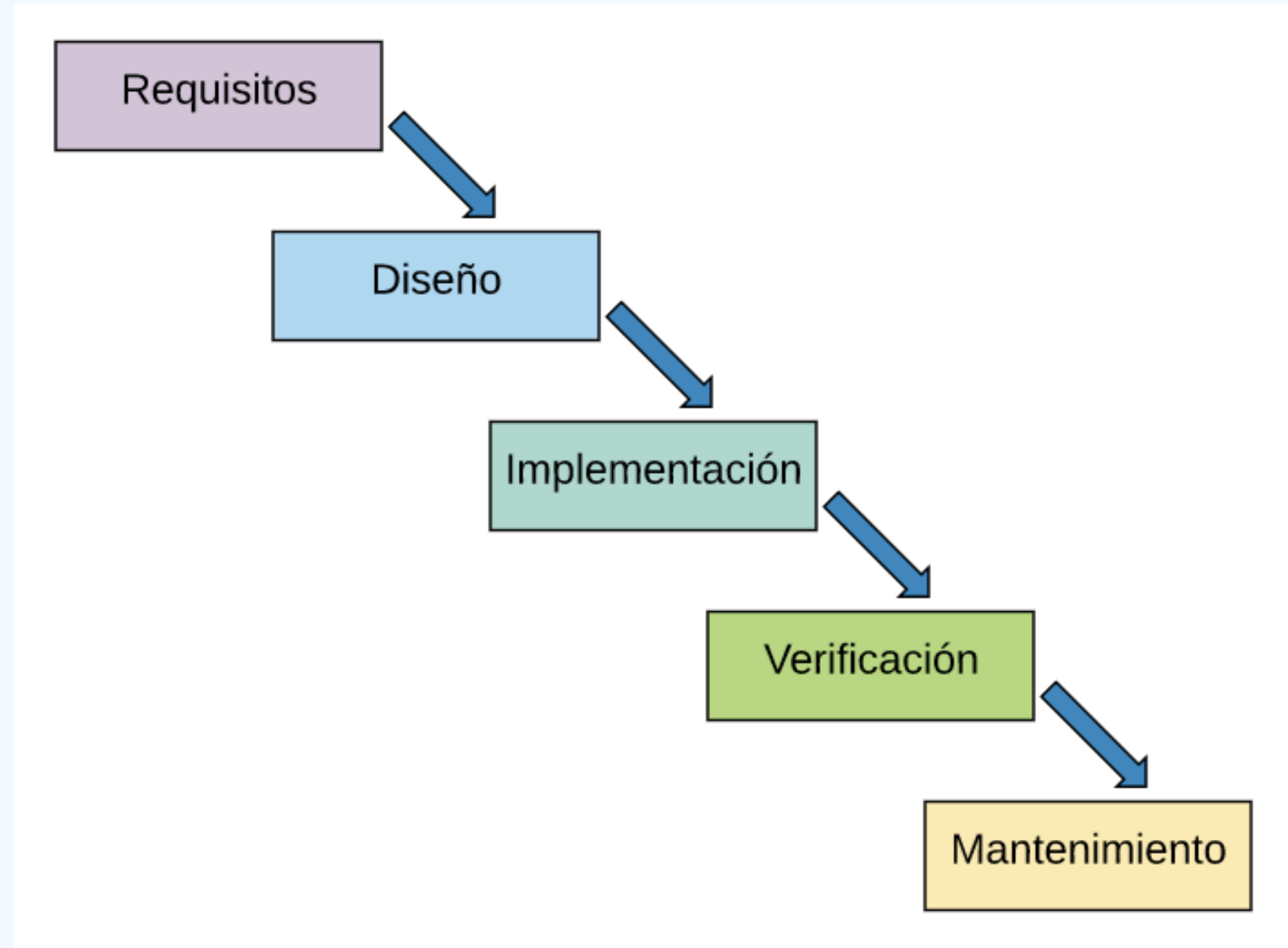
SIMPLICIDAD

- Según el código crece, la calidad del mismo tiende a empeorar.
- Es importante mantener el **código lo más simple y limpio posible**.
 - Los nuevos integrantes tardarán más en empezar a aportar valor.
 - El equipo en general disfrutará menos trabajando.
 - Aumenta la probabilidad de crear nuevos **bugs**.
- **Problemas habituales:**
 - Módulos muy acoplados.
 - Dependencias innecesarias.
 - Nombres de variables, métodos etc. inconsistentes.
 - Acciones inesperadas.

EXTENSIBILIDAD

- **El cambio en el software es frecuente e inevitable.**
 - Debemos adaptarnos, no luchar contra él.
- Formas de hacer nuestro sistema extensible:
 - Desde el punto de vista del **código, manteniéndolo simple y limpio.**
 - Desde el punto de vista de los **procesos de la organización.**

CASCADA



AGILE

