
Algunos consejos de trabajo para programar software

inicio de redacción: 2022/Marzo
últimos cambios: 4 de abril de 2022

===== seleccionar una cierta funcionalidad, un caso de uso, algo que los usuarios podrían utilizar; que se pueda completar en un máximo de dos o tres días de trabajo.

Abrir una nueva rama particular de trabajo en el sistema gestor de versiones.

- Trabajar en (pequeños) pasos concretos.
- Acabar completamente cada paso antes de pasar al siguiente. No dejar ningún “fleco suelto” ni nada “para completar más tarde”.
- Escribir tests para el paso, es parte de dejarlo completamente acabado.
- Guardar el paso y sus test en el sistema gestor de versiones.

Refactorizar con frecuencia, cada vez que te des cuenta de que algo merece la pena mejorar.

Escribir nuevos test, cada vez que te des cuenta de que algo más es susceptible de ser comprobado de forma automatizada.

Cuando esté completada toda la funcionalidad, fusionar la rama particular de trabajo en la rama principal del sistema gestor de versiones y subirla al repositorio común.

Tras esto, la rama particular de trabajo puede borrarse.

volver a =====

Cuidar los nombres en pantallas, mensajes, módulos, funciones y variables. Utilizar un vocabulario claro, comprensible y acordado por todos (incluidos los usuarios). {DDD, Domain-Driven-Design}

Escribir código legible por sí mismo, sin necesidad de comentarios para comprenderlo.

Las distintas partes del programa tienen bajo acoplamiento entre ellas, no hay dependencias innecesarias entre ellas y cada parte se encarga de una única tarea concreta. {principios SOLID}

Las distintas partes del software no necesitan saber de la implementación interna de otras partes de las que hacen uso. Ese uso se hace en base a “contratos” (interfaces), definidos según las necesidades funcionales de las partes cliente.

Así, cada parte del software puede evolucionar separadamente, mientras cumpla con los interfaces que se ha comprometido a implementar.

Esos interfaces son como las costuras de un traje. Y, como tales, son los puntos que facilitan los test y los ajustes.

Excepto las funciones de Entrada/Salida (Input/Output) como, por ejemplo, funciones destinadas expresamente a recoger datos o mostrar información en el interfaz de usuario, o funciones destinadas expresamente a leer o escribir en archivos o bases de datos,...

El resto de funciones solo devuelven resultados, resultados que obtienen solo en base a sus parámetros de entrada (inmutables) y a su algoritmo interno. Sin “leer” ni “alterar” ni tener ningún otro “efecto secundario” en nada exterior a ellas. {programación FUNCIONAL}

Además del código fuente, el sistema gestor de versiones almacena también los recursos, los manuales, las configuraciones, los scripts de automatización,... Es decir, todo lo necesario para compilar, chequear y desplegar el software.