Fases de desarrollo de software

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con grandes posibilidades de éxito. Esta sistematización indica cómo se divide un proyecto en módulos más pequeños para normalizar cómo se administra el mismo.

Así, una metodología para el desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado.

De esta forma, las etapas del desarrollo de software son las siguientes:

Planificación

Antes de empezar un proyecto de desarrollo de un sistema de información, es necesario hacer ciertas tareas que influirán decisivamente en el éxito del mismo. Dichas tareas son conocidas como el *fuzzy front-end* del proyecto, puesto que no están sujetas a plazos.
Algunas de las tareas de esta fase incluyen actividades como la determinación del ámbito del proyecto, la realización de un estudio de viabilidad, el análisis de los riesgos asociados, la estimación del coste del proyecto, su planificación temporal y la asignación de recursos a las diferentes etapas del proyecto.

Análisis

Por supuesto, hay que averiguar qué es exactamente lo que tiene que hacer el software. Por eso, la etapa de análisis en el ciclo de vida del software corresponde al proceso a través del cual se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer).

Diseño

En esta fase se estudian posibles opciones de implementación para el software que hay que construir, así como decidir la estructura general del mismo. El diseño es una etapa compleja y su proceso debe realizarse de manera iterativa.

Es posible que la solución inicial no sea la más adecuada, por lo que en tal caso hay que refinarla. No obstante, hay catálogos de patrones de diseño muy útiles que recogen errores que otros han cometido para no caer en la misma trampa.

Implementación

En esta fase hay que elegir las herramientas adecuadas, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir. Esta elección dependerá tanto de las decisiones de diseño tomadas como del entorno en el que el software deba funcionar.

Al programar, hay que intentar que el código no sea indescifrable siguiendo distintas pautas como las siguientes:

- Evitar bloques de control no estructurados.
- Identificar correctamente las variables y su alcance.
- Elegir algoritmos y estructuras de datos adecuadas para el problema.
- Mantener la lógica de la aplicación lo más sencilla posible.
- Documentar y comentar adecuadamente el código de los programas.
- Facilitar la interpretación visual del código utilizando reglas de formato de código previamente consensuadas en el equipo de desarrollo.

También hay que tener en cuenta la adquisición de recursos necesarios para que el software funcione, además de desarrollar casos de prueba para comprobar el funcionamiento del mismo según se vaya programando.

Pruebas

Como errar es humano, la fase de pruebas del ciclo de vida del software busca detectar los fallos cometidos en las etapas anteriores para corregirlos. Por supuesto, lo ideal es hacerlo antes de que el usuario final se los encuentre. Se dice que una prueba es un éxito si se detecta algún error.

Instalación o despliegue

La siguiente fase es poner el software en funcionamiento, por lo que hay que planificar el entorno teniendo en cuenta las dependencias existentes entre los diferentes componentes del mismo.

Es posible que haya componentes que funcionen correctamente por separado, pero que al combinarlos provoquen problemas. Por ello, hay que usar combinaciones conocidas que no causen problemas de compatibilidad.

Uso y mantenimiento

Esta es una de las fases más importantes del ciclo de vida de desarrollo del software. Puesto que el software ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres puntos diferenciados:

- Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).
- Adaptarlo a nuevas necesidades (mantenimiento adaptativo).
- Añadirle nuevas funcionalidades (mantenimiento perfectivo).

Aunque suene contradictorio, cuanto mejor es el software más tiempo hay que invertir en su mantenimiento. La principal razón es que se usará más (incluso de formas que no se habían previsto) y, por ende, habrá más propuestas de mejoras.

Teoría del color

¿Qué es la Teoría del color?

Se conoce como Teoría del color a un **conjunto de reglas básicas que rigen la** mezcla **de colores** para conseguir efectos deseados, mediante la combinación de colores o pigmentos. Es un principio de gran importancia en el diseño gráfico, la pintura, la fotografía, la imprenta y la televisión, entre otras áreas visuales.

No existe una única Teoría del color, sin embargo, sino un conjunto de aproximaciones al color y a sus dinámicas. Muchas de ellas forman parte de la historia del arte o de la física (óptica), y poseen diferentes autores.

Por ejemplo, el poeta y científico alemán prerromántico Johann Wolfgang von Goethe (1749-1832) en su libro *Teoría de los colores* de 1810 proponía ya un círculo de color, basándose en los estudios de la materia del propio Isaac Newton. Otro caso conocido es el de Wilhelm Ostwald (1853-1932), químico y filósofo alemán.

Uno de los principales insumos de toda Teoría del color es el círculo cromático. Se trata de una representación circular de todos los colores del espectro visual, organizados de manera tal que los colores contrarios se enfrenten y los colores complementarios estén próximos el uno al otro.

El círculo cromático permite identificar los colores primarios o puros, y aquellos que se consideran derivados, o sea, fruto de la mezcla de colores.

De acuerdo a este tipo de estudios del color, a cada uno se le pueden atribuir distintas propiedades, como son:

- **Matiz.** También llamado “croma”, alude al color en sí mismo, lo que nos permite distinguir un color de otro diferente.
- **Luminosidad.** También llamada “valor”, se refiere a la cantidad de luz presente en el color, o sea, si es más claro o más oscuro, lo que equivale a decir si está más cerca del negro o del blanco.
- **Saturación.** Básicamente se refiere a la pureza del color, o sea, la concentración de gris presente en un color en un momento determinado. Mientras más gris posea, menos puro será y menor será su saturación, viéndose como si estuviera sucio, opaco.