

MODELO DE DESARROLLO DE SOFTWARE ITERATIVO

PRESENTADO POR:

**YANUARD STEVIN MONTIALEGRE.
JUAN DAVID CERQUERA
JOSUE AUDOR**



¿QUÉ ES?

Un modelo de desarrollo de software es un enfoque o marco de trabajo utilizado para estructurar, planificar y controlar el proceso de desarrollo de un sistema de software.



MODELO DE DESARROLLO DE SOFTWARE Y SUS CARACTERÍSTICA S

1). ESTRUCTURA DEL PROCESO

2). METODOLOGÍA Y TÉCNICAS

3). ROLES Y RESPONSABILIDADES

4). DOCUMENTACIÓN

5). CONTROL Y SEGUIMIENTO

6). ADAPTABILIDAD

7). ENFOQUE EN LA CALIDAD

8). ITERATIVO O SECUENCIAL

9). CLIENTE O USUARIO INVOLUCRADO:

10). ESCALABILIDAD

FUNCIONES CLAVE

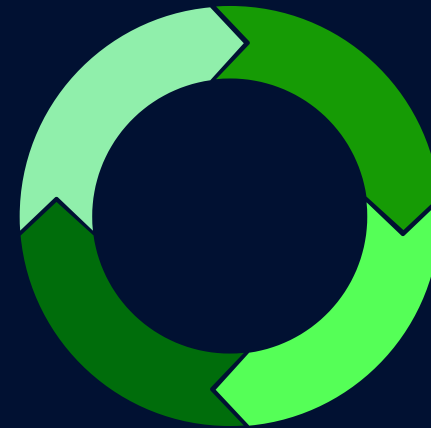
1. Determinar el orden de las fases.
2. Establecer criterios de transición.
3. Definir entradas y salidas.
4. Describir estados y actividades.
5. Servir como base para planificar y desarrollar.

PARA QUÉ SIRVE?

- Define el orden de las fases, criterios de transición, entradas y salidas.
- Describe estados del producto y actividades para transformarlo.

FASES

- Agrupación de actividades relacionadas con un objetivo.
- Comparten un período de tiempo en el proyecto.
 - Pueden incluir bucles de realimentación



CICLO DE VIDA DEL SOFTWARE

QUÉ ES?

Conjunto de fases desde la idea inicial hasta la retirada del software.

ENTREGABLES

- Productos intermedios generados por las fases.
- Pueden ser materiales o inmateriales.
- Evalúan el cumplimiento de requisitos previos.

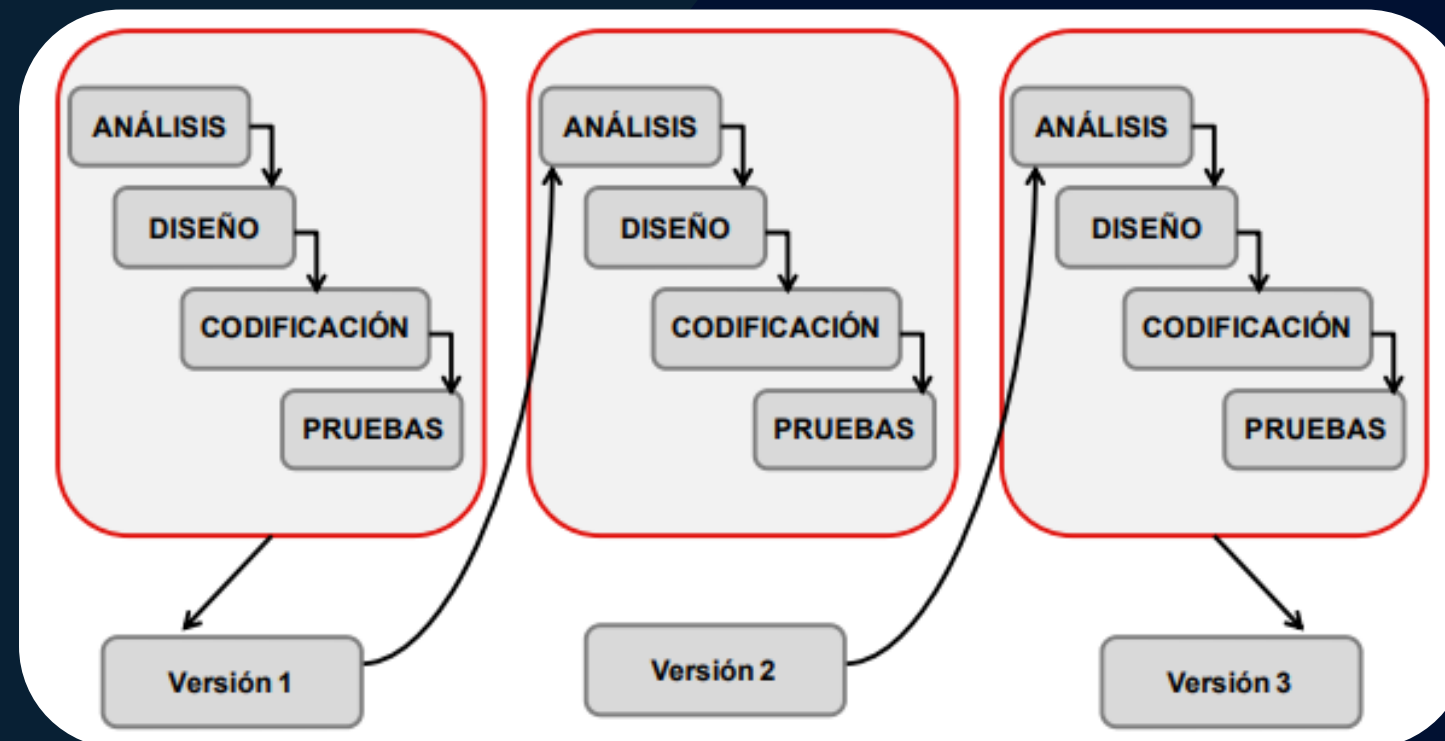
EJEMPLO DE CICLO DE VIDA

1. Requerimientos
2. Diseño
3. Desarrollo
4. Pruebas
5. Despliegue
6. Operación y Mantenimiento
7. Retiro o Reemplazo

CASO PRACTICO

DESARROLLO DE UN SOFTWARE PARA UNA EMPRESA DE SUPERMERCADOS.

En palabras sencillas, el modelo de iteración es un ciclo en cascada que está en continuo cambio, modificación o mejoras con el cliente al terminar cada ciclo. Este modelo se suele utilizar en proyectos en los que los requisitos no están claros por parte del usuario.



CADA VERSIÓN SON MEJORAS O CAMBIOS PRESENTADOS HA Y POR EL CLIENTE PARA SATISFACER LOS REQUISITOS SOLICITADOS Y SOLUCIONAR LAS INQUIETUDES PRESENTES, HASTA DAR CON UNA VERSIÓN CON SOFTWARE FUNCIONAL Y REQUISITOS DEL USUARIO SATISFECHOS.

Ejemplo según nuestro proyecto de un supermercado o ventas en línea y los requisitos son desde el punto de vista del comprador o cliente del supermercado.

VERSIÓN 1

ANÁLISIS/REQUISITOS

- Sistema de registro de usuarios e inicios de sesión.
- Catalogo o lista de artículos en venta.
- Sistema de búsqueda.

DISEÑO

Se diseña la arquitectura y estructura del sistema básico

CODIFICACIÓN

Se codifica el software básico para el aplicativo

PRUEBAS

Se prueba el software básico para garantizar que cumple los requisitos.

VERSIÓN 2

ANÁLISIS/REQUISITOS

- Crear una cuenta de usuario.
- Recuperación de cuenta.
- Guardar productos en un carrito de compra.
- Opciones de entrega a domicilio.

DISEÑO

Se diseñan las nuevas funciones de la aplicación (web, móvil o escritorio).

CODIFICACIÓN

Se codifica el software para las nuevas funciones de la aplicación.

PRUEBAS

Se prueba las nuevas funciones de la aplicación.

VERSIÓN 3

ANÁLISIS FINAL

- Realizar pedidos online.
- Recibir notificaciones sobre ofertas y descuentos.
- Comentarios y opiniones sobre productos.

DISEÑO

Se diseñan las últimas funciones de la aplicación.

CODIFICACIÓN

Se codifica el software para las últimas funciones de la aplicación.

PRUEBAS

Se prueba el software para las últimas funciones de la aplicación.

MODELO DE DESARROLLO DE SOFTWARE ITERATIVO

✓ VENTAJAS

- Mediante este modelo se genera software operativo de forma rápida y en etapas tempranas del ciclo de vida del software.
- Es un modelo más flexible, por lo que se reduce el coste en el cambio de alcance y requisitos.
Es más fácil probar y depurar en una iteración más pequeña.
- Es más fácil gestionar riesgos.
- Cada iteración es un hito gestionado fácilmente

DESVENTAJAS

- Cada fase de una iteración es rígida y no se superponen con otras.
- Pueden surgir problemas referidos a la arquitectura del sistema porque no todos los requisitos se han reunido, ya que se supone que todos ellos se han definido al inicio.

CONCLUSIÓN

LA FUNCIONALIDAD DEL MODELO INTERATIVO ES PRESENTARLE AL USUARIO LAS VERSIONES TERMINADAS Y MOSTRARLAS PARA EL DAR SU APROBADO FINAL O PEDIR CIERTAS MODIFICACIONES EN EL DISEÑO, O REQUISITOS PARA ASÍ PODER RECIBIR O NOSOTROS ENTREGARLE UNA APLICACIÓN SEGÚN SUS GUSTOS Y NECESIDADES, MANTENIENDO LO AL TANTO DEL PROCESO Y NUESTRA COMUNICACIÓN PARA UN TRABAJO EFECTIVO.

RECOMENDACIONES

- [AUDORI]: PERSONALMENTE RECOMENDARÍA ESTE MODELO, CIERTAMENTE TODO DEPENDE DEL TIPO DE PROYECTO QUE SE ABARQUE, PERO EN CASOS DE REQUISITOS INCONCLUSOS O “INESTABLES” RECOMIENDO EL MODELO ITERATIVO, YA QUE PIENSO QUE MANTENER UNA COMUNICACIÓN CON EL CLIENTE ES NECESARIA Y TAMBIÉN PIENSO QUE ES ÚTIL QUE EL CLIENTE ESTÉ AL TANTO DE LO QUE OCURRE ASÍ PUEDE DAR SU OPINIÓN SOBRE EL PROYECTO Y SE PUEDEN AHORRAR CIERTOS INCOMBENIENTES DE ULTIMO MINUTO.
- [YANUARD]: SI LO RECOMENDARIA POR EL ECHO DE SER UN MODELO ITERATIVO QUE SE DESTACA POR SU CAPACIDAD PARA ADAPTARSE A CAMBIOS EN LOS REQUISITOS Y LA PARTICIPACIÓN CONTINUA DEL CLIENTE A LO LARGO DEL PROCESO. ESTO LO CONVIERTE EN UNA OPCIÓN VALIOSA PARA PROYECTOS EN LOS QUE LOS REQUISITOS NO ESTÁN COMPLETAMENTE DEFINIDOS DESDE EL PRINCIPIO
- [CERQUERA]: LO RECOMENDARÍA EN PROYECTOS DONDE LA FLEXIBILIDAD Y ADAPTABILIDAD ES ESENCIAL PARA EL PROYECTO PERO ESTO TIENE QUE SER ANALIZADO DETENIDAMENTE PARA VER QUE MODELO SE AJUSTA MEJOR AL PROYECTO A REALIZAR