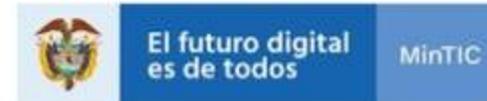


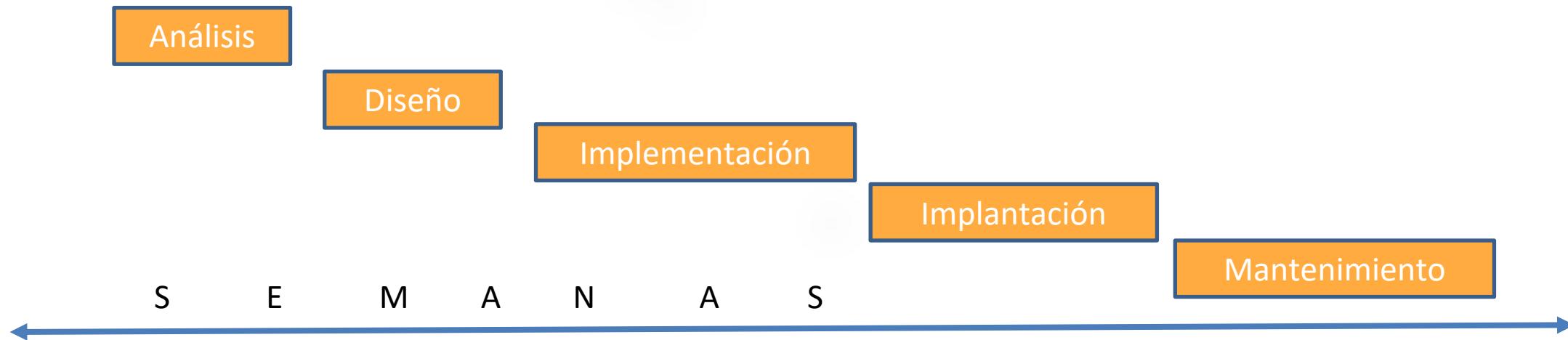
Ciclo de vida del software



¿Que es?

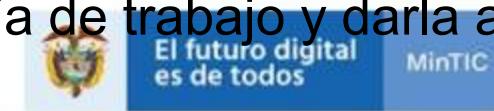
Se llama ciclo de vida porque tiene un comienzo, un desarrollo, un proceso de maduración y un final.

Etapas



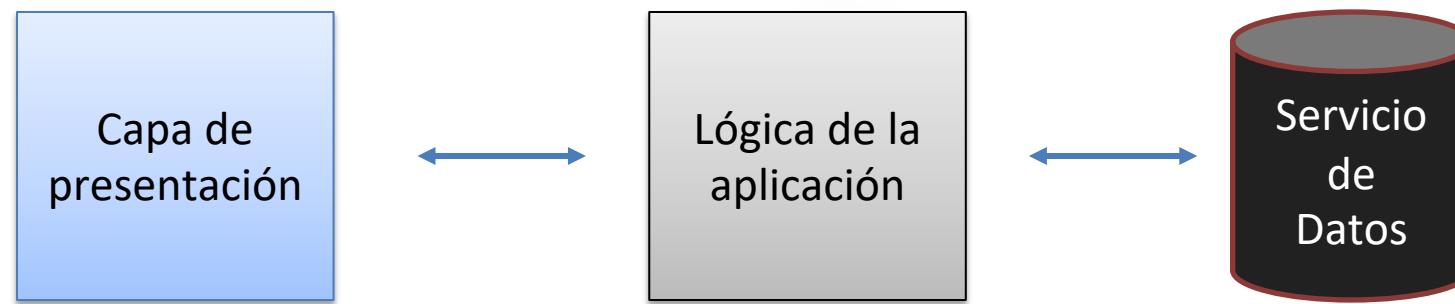
Análisis

- Análisis de riesgos, viabilidad económica, técnica y legal
- Estimación del coste <http://hdl.handle.net/11059/9297>
- Asignación de recursos. Mantener un plan actualizado, no muy detallado ni muy amplio
- Qué abarca y que no abarca el proyecto, qué tiene qué hacer el sistema. Es normal que en esta etapa no se tengan o no sean claros los requerimientos del sistema
- Descripción clara del proyecto
- Uso de modelos: diagramas entidad/relación para indicar cómo se estructuran los datos; flujo de datos para ver cómo se descompone un sistema. Requisitos del usuario
- Desarrollo de prototipos
- Contrato con el cliente
- Realizar entrevistas o encuestas
- Definir la metodología de trabajo y darla a conocer



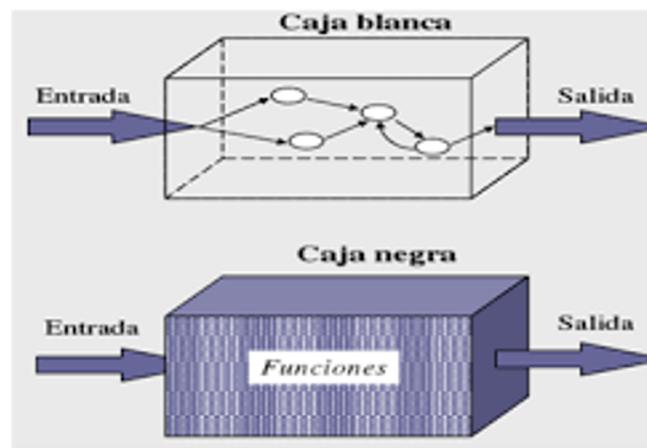
Diseño:

- Diseño de las aplicaciones y de la base de datos
- Divide y vencerás. División de un sistema en capas



Implementación:

- Selección de herramientas, lenguaje de programación, escribir un código claro, documentar, licencias de uso, casos de pruebas.
- Control de versiones
- Pruebas (el objetivo es buscar defectos)



a. Técnicas de Diseño de Casos de Pruebas



Mision
TIC 2022

Implantación

Conjunto de archivos entre los cuales están audios, imágenes, videos, documentos de texto entre otros.

Mantenimiento

- Correctivo: eliminar defectos
- Adaptativo: nuevo entorno
- Perfectivo: nueva funcionalidad o mejoras
(esto implica repetir el ciclo de vida)



Modelo DRA

- Desarrollo Rápido de Aplicaciones
- Ciclo de desarrollo extremadamente corto
- Construcción basado en componentes

Características del Modelo

- Debido a que el software o aplicación se requiere lo más pronto posible no existe una especificación del sistema detallada.
- El software no se desarrolla y utiliza en su totalidad, sino en una serie de incrementos, donde en cada incremento se incluyen nuevas funcionalidades al sistema.
- A menudo se desarrollan las interfaces de usuario del sistema utilizando un sistema de desarrollo interactivo que permite que el diseño de la interfaz se cree rápidamente dibujando y colocando iconos en la interfaz.
- Las funciones secundarias son eliminadas como sea necesario para cumplir con el calendario.

Fases

- **Modelado de Gestión**

El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesó?.

- **Modelado de Datos**

El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.

- **Modelado de Procesos**

Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.

Fases

- **Generación de Aplicaciones**

El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario).

- **Pruebas de Entrega**

- Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas.
- Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

Modelo en cascada o lineal secuencial

Ventajas

- Existe documentación en cada fase
- Hay estabilidad de la operación
- Se facilita la gestión del desarrollo, pues sus fases son conocidas por los desarrolladores
- Modelo y planificación fácil y sencillos.
- Los usuarios lo pueden comprender fácilmente.

Modelo en cascada o lineal secuencial

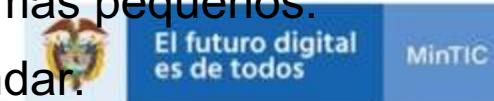
Desventajas

- Una fase no inicia hasta que termina la anterior
- Los requisitos se deben establecer al inicio del proceso
- El tiempo de espera para que el usuario pueda ver resultados es largo
- Los errores en la etapa de análisis son costosos de eliminar pues estos se propagan a las otras fases.
- Los recursos de mantenimiento son altos.
- El proceso de creación del software tarda mucho tiempo ya que debe pasar por el proceso de prueba y hasta que el software no esté completo no se opera.
- Esto es la base para que funcione bien.
- Cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo.

Desarrollo rápido de aplicaciones DRA

Ventajas

- Es muy rápido.
- Permite trabajar en él a varias personas a la vez
- Los entregables pueden ser fácilmente trasladados a otra plataforma.
- El desarrollo se realiza a un nivel de abstracción mayor.
- Entrega temprana al cliente.
- Compromiso del cliente con el sistema.
- Mayor flexibilidad.
- Menor codificación manual.
- Mayor involucramiento de los usuarios.
- Posiblemente menos fallas.
- Posiblemente menor costo.
- Ciclos de desarrollo más pequeños.
- Interfaz gráfica estándar.



Desarrollo rápido de aplicaciones DRA

Desventajas

- Para proyectos grandes aunque por escalas, el DRA requiere recursos humanos suficientes como para crear el número correcto de equipos DRA.
- Requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado
- No todos los tipos de aplicaciones son apropiados para DRA. Si un sistema no se puede modular adecuadamente. La construcción de los componentes necesarios para DRA será problemático.
- No es adecuado cuando los riesgos técnicos son altos. Esto ocurre cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el nuevo software requiere un alto grado de interoperabilidad con programas de computadora ya existentes.
- Enfatiza el desarrollo de componentes de programas reutilizables. La reutilización es la piedra angular de las tecnologías de objetos, y se encuentra en el modelo de proceso de ensamblaje.

Referencias

- De acuerdo al artículo "del manifiesto ágil sus valores y principios
<https://revistas.utp.edu.co/index.php/revistaciencia/article/download/5643/3047>
- <https://modelosdesoftware.webnode.es/dra/>