

APÉNDICE A, APARTADO 1

METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS

DOCUMENTACIÓN

1. La necesidad de los diagramas

Los diagramas o representaciones gráficas representan una parte fundamental en el desarrollo de sistemas. En general, se utilizan para definir las especificaciones de un programa y representar el diseño de los sistemas (programas). Proporcionan el proyecto original para la implantación del diseño en código y además son partes esenciales para la documentación del sistema y posteriormente para su mantenimiento.

Los diagramas claros son una parte fundamental en el diseño de sistemas y en el desarrollo de programas durante el *ciclo de vida del software*. Una técnica de diagramación pobre puede provocar la inhibición en el razonamiento, en cambio, una buena puede acelerar el trabajo y aumentar la calidad de los resultados. Por ejemplo, si sólo hay una persona que desarrolla el diseño de un sistema o de un programa, los diagramas ayudarán al razonamiento claro. Cuando hay varias personas que participan en el desarrollo del sistema los diagramas son una poderosa herramienta esencial para la comunicación y el desarrollo del sistema.

Con una técnica de diagramación formal, los desarrolladores tendrán la posibilidad de intercambiar ideas y podrán reunir con precisión los distintos componentes del sistema. Una de las características con que deben contar los diagramas es con el uso de una simbología estandarizada para facilitar la comunicación y evitar las confusiones que puedan conducir a un error dentro del desarrollo del sistema por parte del equipo de trabajo.

En cuanto a la fase de mantenimiento en el ciclo de vida del software, los diagramas son herramientas valiosas para saber por parte del equipo de programación cómo funciona el sistema, además de que permite localizar el origen de los errores y el impacto de las posibles modificaciones que se hagan al sistema. Como se sabe, después de un periodo transcurrido esto es lo que va desgastando al software.

En términos generales, podemos decir que los diagramas forman un lenguaje que permite el modelado del software, debido a que proporcionan una forma de representación concisa y sin ambigüedades. Además, forman una parte fundamental en el ciclo de vida del software; más específicamente en la etapa de documentación del sistema, sin olvidar que son esenciales para el análisis y el diseño de software.

2. Requisitos de documentación

Una buena documentación de un sistema informático, incluyendo los programas, es un componente de lo que se conoce como filosofía estructurada. De lo anterior, una buena documentación cumple con los siguientes puntos:

- ◆ Proporciona una visión de alto nivel del sistema (programa), explicando su propósito y las relaciones entre los diversos componentes (datos y procedimientos).
- ◆ Mejora la comprensión de los sistemas y, en general, del software producido.
- ◆ Describe detalladamente los componentes de los datos y procedimientos que forman el sistema.
- ◆ Es fácil y poco costosa de realizar y actualizar, y existe la posibilidad de producirse automáticamente.
- ◆ Proporciona una forma de representar los requerimientos de diseño de un sistema y del traslado del diseño al código de programa.

3. Importancia de la visión en los diagramas

Las técnicas de diagramación se utilizan para dar una visión de alto nivel y detallada de un sistema. Una de las cualidades es que permiten la localización exacta de un error fácilmente; si una persona deseara determinar en cuál de varios programas se realiza una determinada función, una visión de alto nivel será la mejor ayuda.

Para representar un sistema completo se necesita un diagrama de estructura de datos de alto nivel y otro detallado del sistema. La visión estructurada del sistema identifica los componentes del proceso y las relaciones entre ellos mostrando el flujo de datos, el flujo de control, así como el tiempo y la secuencia de la información. La visión de la estructura de datos describe principalmente las entidades y atributos, así como sus posibles relaciones o asociaciones. La importancia de ambas visiones radica en que los ingenieros pueden ver cómo los datos se derivan o utilizan en un determinado proceso.

Existe varias técnicas de diagramación (estructurada), pero los tres tipos de diagramas fundamentales para representar un sistema informático son:

Diagrama de flujo de datos (DFD): Un diagrama conocido y utilizado durante el análisis para definir los componentes del problema (sistema) y diseñar un primer esbozo de los componentes del programa y del paso de datos entre ellos. Describe la transformación que los datos experimentan en su flujo a través del sistema.

Diagramas de los modelos de datos: Un diagrama utilizado durante el proceso de modelado de los datos para representar los registros y entidades de los datos y las asociaciones lógicas entre ellos.

Diagrama de estructura de árbol: Un diagrama jerárquico creado durante un diseño del sistema para definir la arquitectura global del sistema para mostrar los programas y las relaciones entre ellos.

4. Tipos de metodologías

Entre los tres tipos de metodologías más utilizados están la de Yourdon, De Marco y la de Gane-Searson, las cuales están basadas en los siguientes conceptos:

- ◆ Usan la organización jerarquizada descendente, por medio de la descomposición funcional para definir los requerimientos del sistema.
- ◆ Herramientas gráficas de comunicación y documentación.

4.1. Metodología de diseño estructurado de Yourdon

Esta metodología proporciona una manera para diseñar paso a paso sistemas y programas detallados. Cabe mencionar que unos pasos involucran el análisis, otros el desarrollo del diseño y otros más la medición y la mejora de la calidad del diseño. La principal herramienta generada en el diseño estructurado es el “diagrama de estructura” donde muestra los componentes de procedimientos del programa, su ordenación jerárquica y los datos conectados a ellos.

El diagrama de estructura es un diagrama de árbol o jerárquico que, en términos generales, define la arquitectura global de un programa que muestra los procedimientos y sus interrelaciones. En dicho diagrama se utilizan bloques básicos, como son cajas que representan los componentes de procedimientos y las flechas que muestran como se conectan.

Yourdon en su metodología propone en cuatro pasos el proceso de diseño. A continuación se explicará cada uno.

◆ ***Trazar el diagrama de flujo de datos***

El objetivo es representar el problema de diseño como el flujo de datos a través de un sistema. Un sistema se compone de procesos que transforman a los datos. Estos procesos y los datos que los enlazan forman los cimientos para definir los componentes del programa.

◆ ***Trazar el diagrama de estructura***

En este punto se desea representar el diseño del programa como una jerarquía de componentes de procedimiento. El diagrama de estructura se deriva del diagrama de flujo de datos obtenido previamente. El diseño estructurado proporciona dos estrategias de diseño para guiar la transformación respectiva, las cuales son: los análisis de transformación y los análisis de transacción. Estas dos estrategias nos ayudan a dirigir el diseño jerárquico, así como un proceso paso a paso de transformación por cada estrategia.

Análisis de transformación

Este modelo de flujo de información divide al diagrama de flujo de datos (DFD) en tres partes: la entrada que recibe el nombre de *rama aferente*; el proceso lógico llamado *transformación central*; y la salida, denominada *rama eferente*.

Análisis de transacción

Este modelo se utiliza cuando se diseñan programas con proceso de transacciones. El diagrama de estructura general para un programa con procesos de transacciones se representa en la parte superior por el módulo de la transacción central y en la parte inferior hay varios módulos de transacciones para cada tipo distinto de transacción.

◆ ***Evaluación del diseño***

En este punto la medición de la calidad de diseño es fundamental, para ello se utilizan dos técnicas ya conocidas, como son el *acoplamiento* y la *cohesión*.

El *acoplamiento* mide el grado de independencia entre los componentes de los procedimientos (módulos) en el diagrama de estructura.

La *cohesión* mide la fuerza de las relaciones entre los elementos dentro de un módulo.

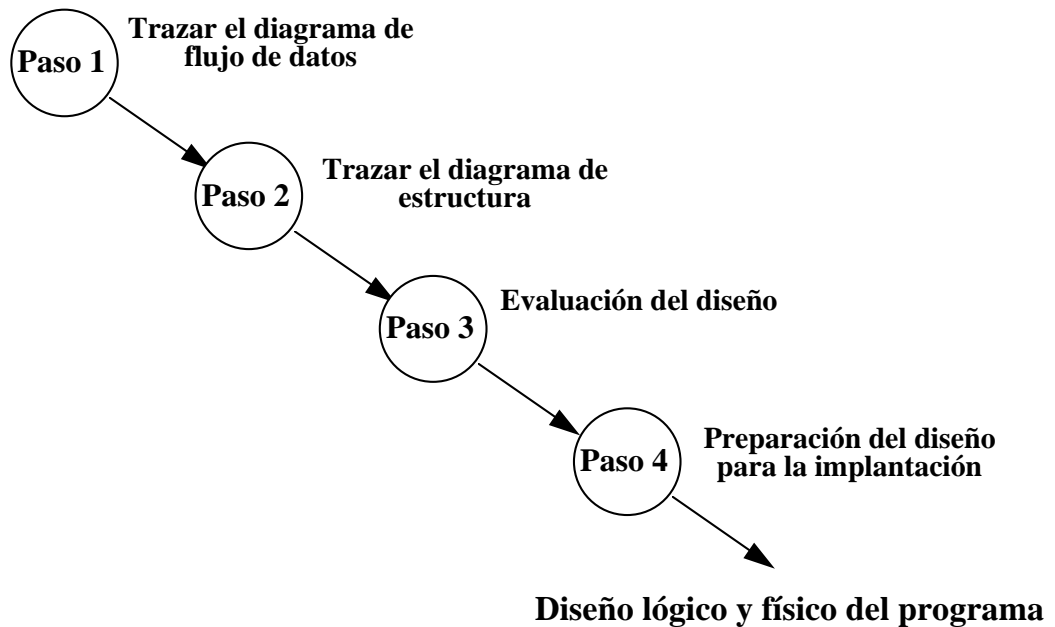
Lo ideal es tener un bajo acoplamiento y un alto grado de cohesión.

◆ ***Preparación del diseño para la implantación***

Esta parte también es conocida como *empaquetar el diseño*. *Empaquetar* es el proceso de dividir el diseño del programa lógico en unidades físicas de implantación llamadas unidades de carga. De hecho es un diseño físico del programa.

En la siguiente figura se muestra los pasos básicos del diseño de Yourdon.

Metodología de diseño estructurado de YOURDON



4.2. Metodología de análisis de DeMarco

La metodología de DeMarco consta de los siguientes siete pasos:

- ◆ Construir el modelo físico.
- ◆ Construir el modelo lógico a partir del modelo físico.
- ◆ Construir un modelo lógico del nuevo sistema. En términos generales, es construir una especificación estructurada que incluye los diagramas de flujo de datos, un diccionario de datos y las especificaciones de los procesos.
- ◆ Crear una familia con los nuevos modelos físicos.
- ◆ Estimar los costos y los tiempos para cada modelo.
- ◆ Seleccionar un modelo.
- ◆ Empaquetar la especificación en subsistemas.

4.3. Metodología de análisis de Gane y Searson

La metodología de Gane y Searson hace referencia a cinco pasos, los cuales se mencionarán a continuación.

- ◆ Construir un modelo lógico en curso.
- ◆ Construir un modelo lógico del nuevo sistema, este punto involucra:
 - Construir una especificación estructurada que contenga los diagramas de flujo de datos, un diccionario de datos y las especificaciones de proceso.
 - Construir un modelo lógico de datos que exprese en tercera forma normal el contenido de los datos almacenados.
- ◆ Diseñar la base de datos física.
- ◆ Crear un nuevo modelo físico del sistema.
- ◆ Empaquetar la especificación en subsistemas.