

Análisis de sistemas: modelización de tratamientos. Modelo de flujo de datos y control. Técnicas descriptivas. Documentación.

## **TEMA 49**

---

**ABACUS NT**

## ***Índice***

---

### ***1. Introducción***

#### ***1. Análisis de sistemas***

##### **1.1. Aproximaciones al análisis de sistemas**

##### **1.2. Análisis estructurado**

##### **1.3. Niveles de implementación**

- 1.3.1. Nivel conceptual
- 1.3.2. Nivel lógico
- 1.3.3. Nivel físico

#### ***2. Modelo de flujo de datos y control***

##### **2.1. Elementos de un DFD**

##### **2.2. Análisis con DFDs**

- 2.2.1. Identificar entradas y salidas.
- 2.2.2. Rellenar el DFD.
- 2.2.3. Etiquetar los flujos de datos
- 2.2.4. Etiquetar los procesos
- 2.2.5. Consideraciones complementarias

##### **2.3. Descomposición por niveles**

#### ***3. Técnicas descriptivas y documentación.***

##### **3.1. Diagrama de contexto**

##### **3.2. Diccionario de datos**

##### **3.3. Diagramas de Flujo de Datos**

##### **3.4. Especificaciones de los procesos**

- 3.4.1. Diagrama de Estructura
- 3.4.2. Lenguaje de consulta estructurado

#### ***4. Conclusión.***

##### **4.1. Relación con el sistema educativo**

#### ***5. Bibliografía***

## 1. Introducción

### Crisis del software

Con la progresiva implantación de los sistemas informáticos en empresas e instituciones, surge la denominada ***Crisis del Software***.

Este término, acuñado por **Dijkstra** en su libro “The Humble Programmer” -El humilde programador- se trata por primera vez en la primera conferencia de la OTAN en 1968 cuando Friedrich L. Bauer habló por primera vez del conjunto de dificultades o errores ocurridos en la planificación, estimación de los costos, productividad y calidad de un software, a raíz de graves incidentes ocurridos (por ejemplo En abril de 1986 un avión de combate F-16 se estrelló por culpa de un giro descontrolado atribuido a una expresión “if then”).

Tomando como inspiración la sistemática de otras disciplinas de ingeniería, la llegada de metodologías estructuradas de análisis y desarrollo de sistemas propició la salida de esta crisis, y aportó un auténtico enfoque de ingeniería al desarrollo de sistemas informáticos.

A partir de estos momentos hay una evolución en varias líneas o frentes:

- Surgen nuevas técnicas metodológicas como superación del ciclo de vida clásico (técnicas estructuradas).
- Empiezan a surgir nuevas herramientas de desarrollo basadas en el ordenador.
- El hardware evoluciona, aparecen nuevos entornos, máquinas cada vez más pequeñas y a la vez más potentes sobre las que son posibles nuevos sistemas operativos y nuevos entornos de desarrollo basados en las nuevas metodologías y herramientas.

## 1. Análisis de sistemas

### 1.1. Aproximaciones al análisis de sistemas

A la hora de afrontar el análisis de un sistema, es posible distinguir dos aspectos de la misma, **los datos** correspondientes a abstracciones aplicadas a contenidos de información y los tratamientos a **abstracciones operacionales**, es decir a la manipulación de los datos.

Según esto, existen diversos enfoques al analizar un sistema según percibamos los datos como estructurados, objetos, etc., dando lugar a tres aproximaciones distintas:

- Análisis orientado a los tratamientos (flujo de datos)
- Análisis orientado a datos (**estructura de datos o estructurado**)
- Análisis orientado a objetos (visión integral)

En este tema nos centraremos en el **análisis estructurado**, es decir que modeliza el sistema según las estructuras de datos y los tratamientos de que son objeto.

## 1.2. Análisis estructurado

El **Análisis Estructurado** en ingeniería de software y su técnica aliada, **Diseño estructurado**, es un conjunto de métodos orientados a analizar y convertir requisitos de negocio dentro de especificaciones y en última instancia, programas informáticos, configuraciones de hardware y procedimientos manuales relacionados.

Las técnicas de análisis y diseño estructurados son herramientas fundamentales de análisis de sistemas, y desarrolladas a partir de análisis de sistemas clásicos de los años 1960 y 1970

El análisis estructurado, como todos los demás métodos de análisis de requisitos, es una actividad de **construcción de modelos**, en concreto es una técnica de **modelización del flujo y del contenido de la información**.

Mediante una notación que es única del método de análisis estructurado, creamos modelos que reflejan el flujo y el contenido de la información (datos y control); partimos el sistema funcionalmente y, según los distintos comportamientos, establecemos la esencia de lo que se debe construir.

## 1.3. Niveles de implementación

A la hora de afrontar el análisis de un sistema o de una aplicación software, es posible distinguir dos aspectos de la misma, **los datos y los tratamientos** cuyo estudio se viene dividiendo en tres niveles:

### 1.3.1. Nivel conceptual

En el modelo de tratamientos, se utiliza el DFD, una herramienta gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

Adicionalmente se utiliza el diccionario de datos DD para añadir información sobre la aplicación a los gráficos.

### 1.3.2. Nivel lógico

En el nivel lógico, se utiliza un Lenguaje de Descripción de Programas o LDP para especificar el comportamiento de la aplicación. Actualmente el LDP más utilizado es el pseudocódigo que básicamente es un lenguaje de programación de muy alto nivel.

También existen otras técnicas como los ordinogramas o los diagramas de cajas, aunque todas ellas son objeto de estudio en otros temas.

### 1.3.3. Nivel físico

En este nivel se implementan los tratamientos en un lenguaje comprensible por las máquinas donde se van a ejecutar los programas. Este paso puede estar completa o

parcialmente automatizado, generándose el código correspondiente a partir de la descripción efectuada a nivel lógico.

## 2. Modelo de flujo de datos y control

Los DFD es la herramienta de modelado más extendida del análisis estructurado de sistemas. Fue definida por DeMarco a finales de los 70, y permite representar gráficamente un sistema como una red de procesos funcionales, conectados entre sí a través de flujos de datos de información.

### 2.1. Elementos de un DFD

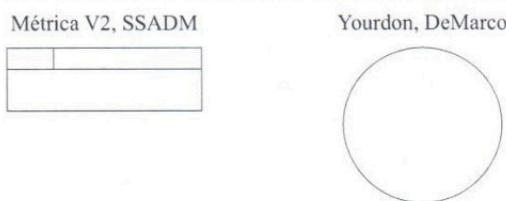
Los componentes de un DFD son cuatro: proceso, flujo de datos, entidad externa y almacenamiento, que se combinan según un conjunto de reglas.

#### Proceso

El primer componente del DFD es el proceso, también conocido como burbuja y representa funciones de transformación de datos, por tanto, reciben flujos de entrada y generan otros de salida obtenidos mediante transformación de los datos de entrada.

Hay que tener en cuenta que un proceso no genera ni destruye datos, sólo los transforma y que entre una entidad externa y un almacén de datos es necesario un proceso.

La representación gráfica es:



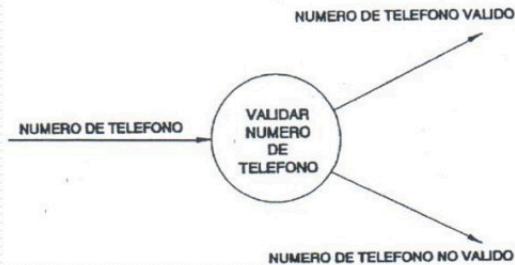
Los procesos se identifican por un número (que depende del nivel de abstracción en el que se encuentre) y por un nombre que debe ser significativo.

#### Flujo de datos

Sirven para conectar procesos, almacenes y entidades, representando la información que fluye entre ellos, así como el sentido de dicho flujo. Gráficamente se identifican con una flecha.

Es importante distinguir la función de los flujos de la de los almacenamientos, así mientras los flujos simbolizan el movimiento de los datos, los almacenamientos representan la información almacenada estáticamente.

A continuación se muestra un ejemplo de utilización de flujos:



En este caso un mismo bloque de información tiene distinto significado según sea tratado como número de teléfono o número de teléfono válido.

### Almacenes de datos

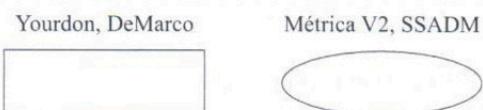
Como su nombre indica, se trata de un depósito de información que puede ser utilizado por los procesos para almacenar y/o recuperar dichos datos y se representa mediante dos líneas paralelas.

Los almacenes de datos se identifican por un nombre representativo de la información que almacenan y como ya se ha comentado, están conectados por flujos a los procesos. Estos flujos pueden ser de entrada de salida o de ambos.

### Entidades externas

Una entidad externa representa un elemento exterior al sistema que aporta o recibe información del mismo, como es el caso de organizaciones o personas que no pertenecen al entorno del sistema.

En cuanto a la representación gráfica existen dos alternativas: la primera se debe a Yourdon y DeMarco y la segunda a la empleada por Métrica v3 y SSADM:



Estas entidades externas pueden conectarse con varios procesos o almacenamientos del sistema diseñado y los flujos que las unen representan la interfaz entre el sistema y el mundo exterior.

## 2.2. Análisis con DFDs

El diagrama de flujo de datos (DFD) es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

Para alcanzar un buen diseño del modelo de flujo de datos de un sistema, es necesario seguir los siguientes pasos:

### 2.2.1. Identificar entradas y salidas

La determinación de entradas y salidas está directamente relacionada con la decisión de cuál será el contexto del estudio, cuyo objetivo es seleccionar un contexto suficientemente grande como para incluir todo lo necesario para el desarrollo, pero lo suficientemente pequeño como para incluir pocos o ningún dato relevante.

Una vez decidida esta frontera se deben buscar los flujos de información que la traspasan y que serán las entradas y salidas a representar en el DFD.

### 2.2.2. Rellenar el DFD

Primero hay que concentrarse en los flujos de datos que atraviesan el sistema. Cuando se identifique un conjunto de información que el usuario trata como una unidad, se representará como un flujo de datos en el DFD. También habrá que colocar procesos allí donde se necesite transformar un flujo de datos en otro, aunque en este momento no es necesario nombrar dichos procesos y se pueden dejar en blanco.

En estos procesos habrá que ver si se puede localizar algún flujo de datos interno que pueda ser utilizado en el proceso, para ello se consultará con el usuario. Si es así, se reemplaza el proceso simple por varios procesos unidos por flujos.

Después aparecerán los archivos en el DFD para representar los distintos depósitos de datos mencionados por el usuario. Será necesario conocer perfectamente su contenido para poder representar los flujos que entran y salen de él.

También hay que estar preparado para modificar la frontera del contexto ya que se pudo haber olvidado una entrada que sea un componente clave para alguno de sus flujos de datos y por tanto, habrá que incluirla. También puede ocurrir lo contrario, suprimiéndose en este caso, los flujos de datos no necesarios.

### 2.2.3. Etiquetar los flujos de datos

Los nombres que se dan a los flujos de datos tienen gran importancia para la legibilidad del DFD y para ello hay que tener presentes algunos conceptos como dar un nombre a todos los flujos y que dicho nombre sea significativo, evitando nombres absurdos como "datos" o "información".

## 2.2.4. Etiquetar los procesos

Siguiendo los pasos del diseño descendente es conveniente trabajar con burbujas en blanco hasta que se hayan asignado todos los nombres a los flujos de datos. Al igual que en el caso anterior, es recomendable utilizar nombres significativos y que consten de un solo verbo, evitando nombres como “proceso” o “gestión”.

## 2.2.5. Consideraciones complementarias

Otros pasos posibles son la documentación del estado actual, ignorar errores triviales y representar flujos de datos y no flujos de control.

## 2.3. Descomposición por niveles

Partiendo del documento de entrevista con los usuarios y responsables de distintas áreas, el analista de sistemas realiza un primer diagrama, identificando en él los elementos del apartado anterior, esto es: procesos, flujos de datos, almacenes de datos y entidades externas.

Este diagrama inicial se denomina **DFD de nivel cero** o diagrama de **contexto** y representa al elemento de software completo como una sola burbuja con datos de entrada y de salida representados por flechas de entrada y de salida, respectivamente.

A partir de él se pueden definir el resto de niveles. En general, vamos a tener los siguientes:

### Diagrama de Contexto: Nivel 0

En el diagrama de contexto se caracterizan todas las interacciones que realiza un sistema con su entorno (entidades externas), estas pueden ser otros sistemas, sectores internos a la organización, o factores externos a la misma. Se dibuja un solo proceso que representa al sistema en cuestión y se escribe su nombre en dicha burbuja como un sustantivo común más adjetivos. De él solamente parten los flujos de datos que denotan las interrelaciones entre el sistema y sus agentes externos, no admitiéndose otros procesos ni almacenamientos en el dibujo, ya que estos son procesos estructurados y ordenados, además posee una cardinalidad que varía según la función que desempeñe cada diagrama. Resulta de gran utilidad para los niveles posteriores de análisis como herramienta de balanceo y es conocido como el Diagrama de Flujo de Datos DFD de Nivel "0".

### Diagrama de Nivel Superior: Nivel 1

En el diagrama de nivel superior se plasman todos los procesos que describen al proceso principal. En este nivel los procesos no suelen interrelacionarse directamente, sino que entre ellos debe existir algún almacenamiento o entidad externa que los une. Esta regla de construcción sirve como ayuda al analista para contemplar que en un nivel tan elevado de abstracción (DFD Nivel 1) es altamente probable que la información que se maneja requiera ser almacenada en el sistema aunque no esté especificado por un Requisito funcional, siendo en realidad un requisito no-funcional.

## Diagrama de Detalle o Expansión: Nivel 2

En un diagrama de nivel 2 o mayor, comienzan a explotarse las excepciones a los caminos principales de la información dado que aumenta progresivamente el nivel de detalle. De aquí en adelante se permiten los flujos entre procesos.

El DFD (Diagrama De Flujo De Datos) nivel 2 puede considerarse el máximo para ser validado en forma conjunta con el usuario dado que en los niveles posteriores el alto grado de complejidad del diagrama puede resultar de muy difícil lectura para personas ajenas al equipo de sistemas. También se recomienda el diagrama de nivel superior.

A continuación se muestra un ejemplo de DFD:

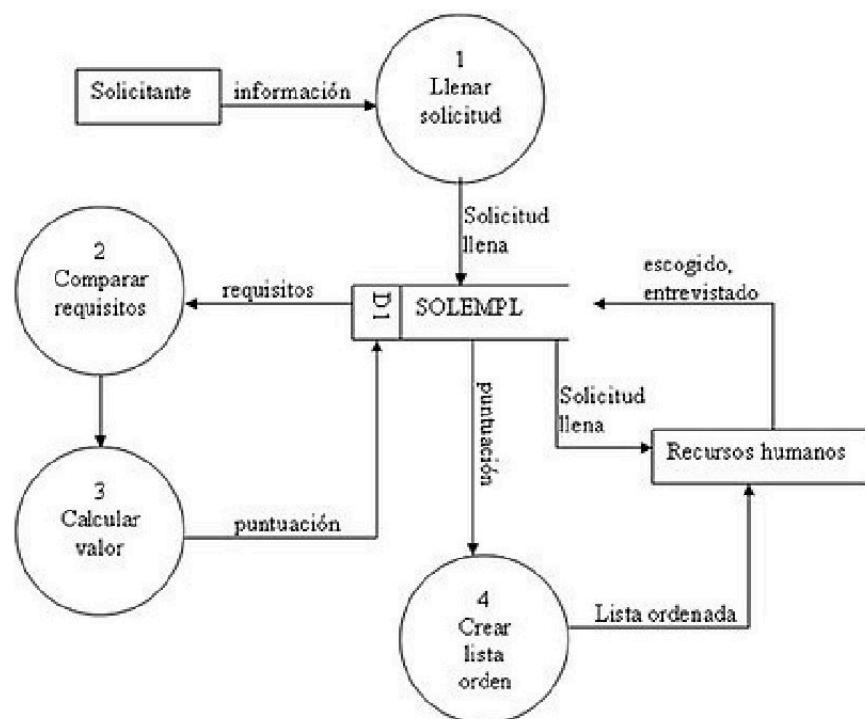


Diagrama de nivel 0.

Fuente: [Wikipedia](#)

## 3. Técnicas descriptivas y documentación.

El análisis estructurado se aprovecha de la ocultación de información a través del análisis de descomposición sucesiva (TOP-DOWN). Esto permite que la atención se centre en los detalles pertinentes y evita la confusión de mirar los detalles irrelevantes. Como el nivel de detalle aumenta, se reduce la amplitud de la información. El resultado del análisis

estructurado es un conjunto relacionado de diagramas, gráficas, descripciones de procesos, y las definiciones de datos. Ellos describen las transformaciones que deben llevarse a cabo y los datos necesarios para cumplir con el Requisito funcional de un sistema.

El enfoque del análisis estructurado desarrolla perspectivas tanto en los objetos del proceso y los datos de los objetos.

El enfoque de DeMarco consta de los siguientes objetos:

1. **Diagrama de contexto de sistema**
2. **Diagrama de flujo de datos**
3. **Diccionario de datos**
4. **Especificaciones de los procesos**

Con el diagrama de flujo de datos (DFDs) se dirigen gráficos. Los arcos representan los datos, y los nodos (círculos o burbujas) representan procesos que transforman los datos. Un proceso se puede descomponer además a un DFD más detallado que muestra los subprocessos y los flujos de datos dentro de ella. Los subprocessos pueden a su vez descomponerse aún más con otro conjunto de DFD hasta que sus funciones se pueden entender fácilmente. Las funciones primitivas son procesos que no necesitan ser descompuestas aún más. Estas funciones primitivas se describen por una especificación de proceso (o mini-spec).

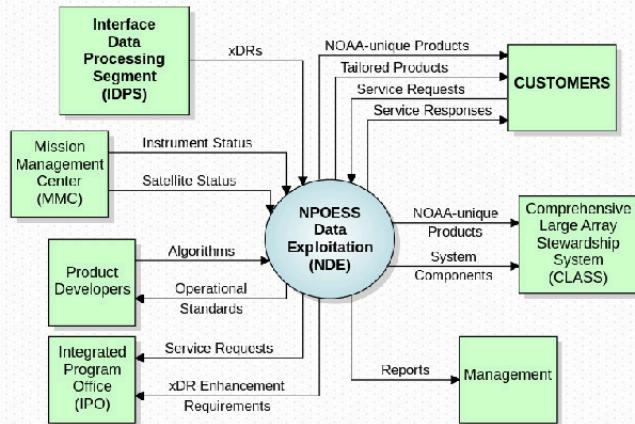
La especificación del proceso puede consistir en **pseudocódigo**, **diagramas de flujo**, o **Inglés estructurado**. En el modelo de DFD la estructura del sistema es como una red de procesos interconectados compuestos de funciones primitivas.

El diccionario de datos es un conjunto de entradas (definiciones) de los flujos de datos, elementos de datos, archivos y bases de datos. Los diccionarios de datos se dividen de una manera de arriba hacia abajo. Estos pueden ser referenciados en otras entradas del diccionario de datos y en los diagramas de flujo de datos.

### 3.1. **Diagrama de contexto**

Los Diagrama de contexto de sistema son diagramas que representan los usuarios que fuera de un sistema podrían interactuar con él.

Este diagrama es la vista de más alto nivel de un sistema, similar al diagrama de bloques, que muestra, posiblemente un sistema de software basado en entradas y salidas además de sus factores externos.



Ejemplo de un diagrama de contexto

De NDE - NDE Context Diagram.jpg (redrawn in SVG), Dominio público, <https://commons.wikimedia.org/w/index.php?curid=39317877>

Este tipo de diagrama según Kossiakoff (2003) por lo general "muestra el sistema en el centro, sin detalles de su estructura interior, rodeado de todos sus sistemas de interacción, medio ambiente y actividades. El objetivo de un diagrama de contexto del sistema es centrar la atención en factores y eventos que deben ser considerados en el desarrollo de un conjunto completo de las propuestas del sistema y las limitaciones externas".

El diagrama de contexto del sistema está relacionado con el Diagrama de flujo de datos, y muestran las interacciones entre un sistema y otros usuarios con los que el sistema está diseñado para hacer frente. Un diagrama de contexto del sistema puede ser útil para comprender el contexto en el cual el sistema será parte de la ingeniería de software.

### 3.2. Diccionario de datos

Un diccionario de la base de datos contiene una lista de todos los archivos de la base de datos, el número de registros en cada archivo, y los nombres y tipos de cada campo de datos. El diccionario de datos contiene la descripción lógica y la documentación de la información que se guarda en los almacenes de información, de los datos que transportan cada uno de los flujos de datos y de los procesos que aparecen en el modelo gráfico.

Es decir, es una lista organizada de todos los datos del sistema con definiciones precisas y rigurosas para que tanto el analista como el usuario tengan una extensa relación de todas las entradas, salidas, almacenes de datos y cálculos intermedios.

Aunque el formato del diccionario varía entre las distintas herramientas, la mayoría contiene la siguiente información:

- Nombre. Se asigna a cada dato elemental un nombre significativo y único.
- Descripción. Se proporciona una descripción concisa y clara indicando de manera breve lo que representa.
- Alias. En ocasiones, un mismo dato elemental es conocido por nombres diferentes según el usuario o del departamento que lo utilice.

- Longitud. Indica el número de dígitos que ocupará sin tener en cuenta su almacenamiento interno.
- Valores de datos. Para algunos datos y procesos sólo son permitidos determinados valores, en este caso es necesario registrar los valores posibles en el diccionario de datos.

Los datos compuestos pueden estar formados por una secuencia de datos, selección entre varias alternativas, repetición de datos o datos opcionales.

Ejemplos:

- Nombre = TituloCortesia+NombrePila+Apellido
- TituloCortesia = [ “Sr” | “Sra” ]
- NombrePila = {Carácter}
- Apellido = {Carácter}
- Carácter =[ “A”-“Z”|“a”-“z” ]

En general, a cada flujo y almacén de datos se le asigna un dato compuesto. En el caso del almacén de datos es importante registrar el volumen previsto, que será necesario tener en cuenta para definir la capacidad de los archivos del sistema.

Para documentar cada uno de los procesos de más bajo nivel, será necesario dar una descripción que permita obtener una completa comprensión del sistema y de sus funciones, por tanto, definen qué debe hacerse para transformar las entradas de un proceso primitivo en salidas.

Para estas descripciones puede utilizarse el lenguaje natural o bien establecer ciertos estándares semejantes a algoritmos de alto nivel, detallando la secuencia de los pasos o acciones que han de llevarse a cabo para realizar una determinada función, así como las validaciones que el sistema realizará para los procesos, los pasos que un posible usuario habrá de realizar, etc.

La especificación de un proceso se realiza de forma que se pueda verificar tanto por el analista como por el usuario, por ello, se hace necesario utilizar un lenguaje claro, preciso y no ambiguo, siendo el más característico el pseudocódigo.

Los diccionarios de datos son de gran utilidad en sistemas donde se manipulan grandes volúmenes de información, pero incluso en pequeños sistemas un analista no podrá mantener toda la información sobre los datos a manipular sin anotarla y especificar sus características en este documento.

En la actualidad, existen diccionarios automatizados y herramientas CASE que permiten validar automáticamente la consistencia del diccionario de datos y los DFD.

### 3.3. Diagramas de Flujo de Datos

El DFD está diseñado para mostrar cómo un sistema se divide en porciones más pequeñas y para resaltar el flujo de datos entre las partes. Este diagrama de flujo de datos de nivel de contexto se "explotó" para mostrar más detalles del sistema que se está modelando.

Los diagramas de flujo de datos (DFDs) son una de las tres perspectivas esenciales de Método de análisis y diseño de sistemas estructurados (SSADM).

### 3.4. Especificaciones de los procesos

#### 3.4.1. Diagrama de Estructura

Un diagrama de estructura (SC) es un gráfico que muestra la distribución del sistema de configuración de los niveles más bajos y manejables. Esta tabla se usa en la programación estructurada para organizar los módulos de programa en una estructura de árbol. Cada módulo está representado por una caja que contiene el nombre del módulo. La estructura de árbol visualiza las relaciones entre los módulos.

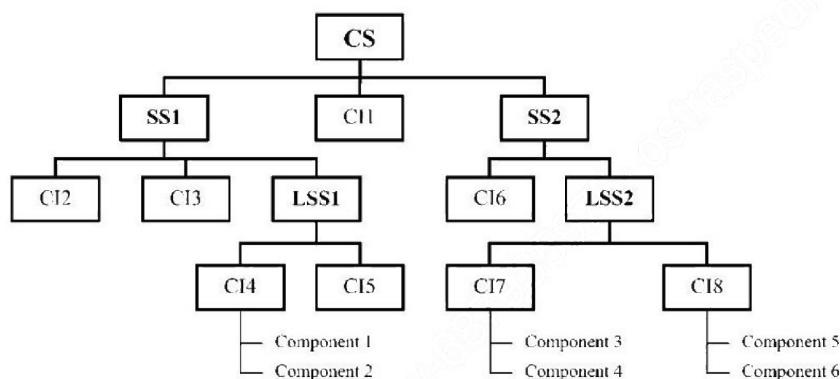


Diagrama de estructura.

De irs.gov - <https://www.irs.gov/irm/part2/47349004.html>, Dominio público, <https://commons.wikimedia.org/w/index.php?curid=5200516>

En análisis estructurado los diagramas de estructura se utilizan para especificar el diseño de alto nivel, o la arquitectura de un programa de computadora. Como una herramienta de diseño, ayudan al programador a dividir y conquistar un problema de software grande, es decir, de forma recursiva romper un problema en partes que son lo suficientemente pequeños para ser entendido por un cerebro humano. El proceso es llamado **Top-down y bottom-up**, o de descomposición funcional. Los programadores usan un diagrama de estructura para construir un programa de una manera similar a cómo un arquitecto utiliza un plano para construir una casa. En la etapa de diseño, el diagrama se dibuja y se utiliza como una manera para que el cliente y los diferentes diseñadores de software puedan comunicarse. Durante la construcción real del programa (aplicación), al gráfico se le refiere continuamente como **el plan maestro**.

#### 3.4.2. Lenguaje de consulta estructurado

El lenguaje de consulta estructurado (SQL) es un lenguaje estandarizado para la consulta de la información de una base de datos. SQL se introdujo por primera vez como un sistema de bases de datos comerciales en 1979 y desde entonces ha sido el lenguaje de consulta favorita para los sistemas de gestión de base de datos que se ejecutan en sistemas grandes y medianos. Cada vez más, sin embargo, SQL está siendo apoyado por los sistemas de bases de datos de PC, ya que soporta las bases de datos distribuidas (véase la definición

de base de datos distribuida). Esto permite que varios usuarios en una red informática puedan acceder a la misma base de datos al mismo tiempo. Aunque hay diferentes dialectos de SQL, sin embargo es lo más parecido a un lenguaje de consulta estándar que existe en la actualidad

## 4. Conclusión.

Un proyecto software es un sistema de pasos a realizar bien de forma secuencial o simultáneos que implica a componentes hardware, software, comunicaciones y personas, enfocado a obtener una serie de resultados deseables sobre un sistema de información.

Si no se han planificado bien los recursos previamente, será muy difícil que se pueda llegar a destino. Se deben detectar y organizar todas las tareas. A cada tarea se le debe asignar recursos materiales y humanos para que pueda ser ejecutada en un determinado periodo, considerando siempre un uso eficiente de los recursos.

Los elementos diagramas y elementos gráficos vistos en este tema contribuyen en gran medida a alcanzar dicho objetivo.

### 4.1. Relación con el sistema educativo

- GS – DAW - DAM –Entornos de Desarrollo
- GS – DAM – Sistemas de Gestión Empresarial

## 5. Bibliografía

- Pressman, R. S. **Ingeniería del Software. Un enfoque práctico**, 3<sup>a</sup> edición. Ed. McGraw-Hill, 2000.
- Sommerville, I.: **Ingeniería de Software**. 6<sup>a</sup> Edición. Addison-Wesley Iberoamericana, 2002.
- Piattini, M. y otros: **Análisis y diseño detallado de Aplicaciones Informáticas de Gestión**. RA-MA, 2003.
- Cerrada, J. A.: **Introducción a la Ingeniería del Software**. 1<sup>a</sup> Edición de 2000. Editorial Centro de Estudios Ramón Areces, S.A.
- [https://es.wikipedia.org/wiki/An%C3%A1lisis\\_estructurado](https://es.wikipedia.org/wiki/An%C3%A1lisis_estructurado)