Universidad Autónoma del Estado de México
Facultad de Ingeniería
Ingeniería en Computación

Unidad de aprendizaje: Análisis de sistemas

Tema:
Paradigma estructurado para construir modelos de análisis de sistemas de información en las organizaciones

Mtra. Tania Lilia Chávez Soto

Contenido

- Proyectos de ingeniería de software
- **Paradigma estructurado para construir modelos de análisis de sistemas de información en las organizaciones**
- Paradigma orientado a objetos para construir modelos de análisis de información en las organizaciones
- Uso de herramientas en los proyectos de ingeniería de software y la construcción de modelos
- Modelo de arquitectura orientado a servicios.

Introducción

La ingeniería de software esta fundada sobre el modelo básico de entrada/proceso/salida de un sistema

Estas metodologías se enfocan fundamentalmente en la parte del proceso

Como ejemplos de las metodologías orientadas al proceso tenemos las metodologías de De Marco, Gane y Sarson y posteriormente Yourdon que se basa en la utilización de método descendente de descomposición funcional para definir requisitos del sistema

Introducción

Cuando el analista de sistemas intenta entender los requerimientos de información de los usuarios, debe tener la capacidad de visualizar cómo se mueven los datos en la organización, los procesos o transformaciones que sufren dichos datos y cuáles son los resultados

Especificación de requisitos

Especificación de requisitos

El proceso del estudio de las necesidades de los usuarios para llegar una definición de los requisitos del sistema, de hardware o de software" [IEEE Sd. 610]

Requisito: son las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación.

- La definición de requisitos debe ser el fruto del trabajo conjunto de las partes involucrados en un desarrollo.
  - Los suministradores
  - Los desarrolladores del software
  - Los clientes y usuarios

Especificación de requisitos

- Según el estándar IEEE 1074 consiste en tres grandes actividades:

Definir los requisitos del software

- Tarea interactiva para crear una definición o especificación preliminar de los requisitos que debe cumplir el software a partir de la información recopilada

Definir los requisitos de las interfaces

- Es necesario definir las propiedades que se deben satisfacer para obtener una interacción eficaz con otros elementos del sistema como el usuario, el hardware u otras aplicaciones de software

Integrar los requisitos

- Una vez descritos los requisitos de software y los de sus interfaces se deben revisar. El usuario tiene un papel fundamental en la aprobación o no de los mismos

Especificación de requisitos

- Otra forma de especificar los requisitos:

Extracción	• Los clientes o los futuros usuarios del software descubren, revelan, articulan y comprenden los requisitos que desean
Análisis de requisitos	• Razonamiento sobre los requisitos obtenidos en la etapa anterior, detectando y resolviendo posibles inconsistencias o conflictos, coordinando los requisitos relacionados entre si
Especificación de requisitos	• Redacción o registro de los requisitos. Para este proceso puede recurrirse al lenguaje natural, lenguajes formales, modelos, gráficos, etc.
Validación de requisitos	• El proceso de confirmación por parte de los usuarios o del cliente, de que los requisitos especificados so válidos, consistentes, completos, etc.

Especificación de requisitos del software

- **Especificación**

Documento que define de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento u otras características de un sistema o componente de un sistema

- **Software**

Es el conjunto de programas, procedimientos y documentación asociada a la operación de un sistema informático

Especificación de requisitos del software

Desempeña un papel similar al que representan en la arquitectura los planos que definen el aspecto de una casa.

Características

- Incluir información cierta, es decir, coherente con las necesidades reales del usuario que se desean satisfacer
- Debe comunicar dicha información de forma eficaz, es decir, de tal manera que se pueda comprender perfectamente.

Especificación de requisitos del software

La especificación debe abordar la descripción de lo que hay que desarrollar, no el cómo, el cuándo, etc. se desarrolla el software

Implicaciones

- Describir correctamente todos los requisitos del software, pero sin incluir requisitos innecesarios
- No describir ningún detalle del diseño del software, de su verificación o de la dirección del proyecto, excepto las restricciones impuestas al diseño que influyen en los requisitos

Especificación de requisitos del software

Características de una buena ERS

- No ambigua
- Completa
- Fácil de verificar
- Consistente (coherente)
- Clasificada por importancia
- Fácil de modificar
- Fácil de identificación del origen y consecuencias de cada requisito
- De fácil utilización durante la fase de explotación y mantenimiento

Especificación de requisitos del software

Estructura para la ERS

1. Introducción
    1. Objetivo
    2. Ámbito
    3. Definiciones, siglas y abreviaturas
    4. Referencias
    5. Visión general
2. Descripción general
    1. Perspectivas del producto
    2. Funciones del producto
    3. Características del usuario
    4. Limitaciones generales
    5. Supuestos y dependencias
3. Requisitos específicos
4. Apéndices
5. Índice

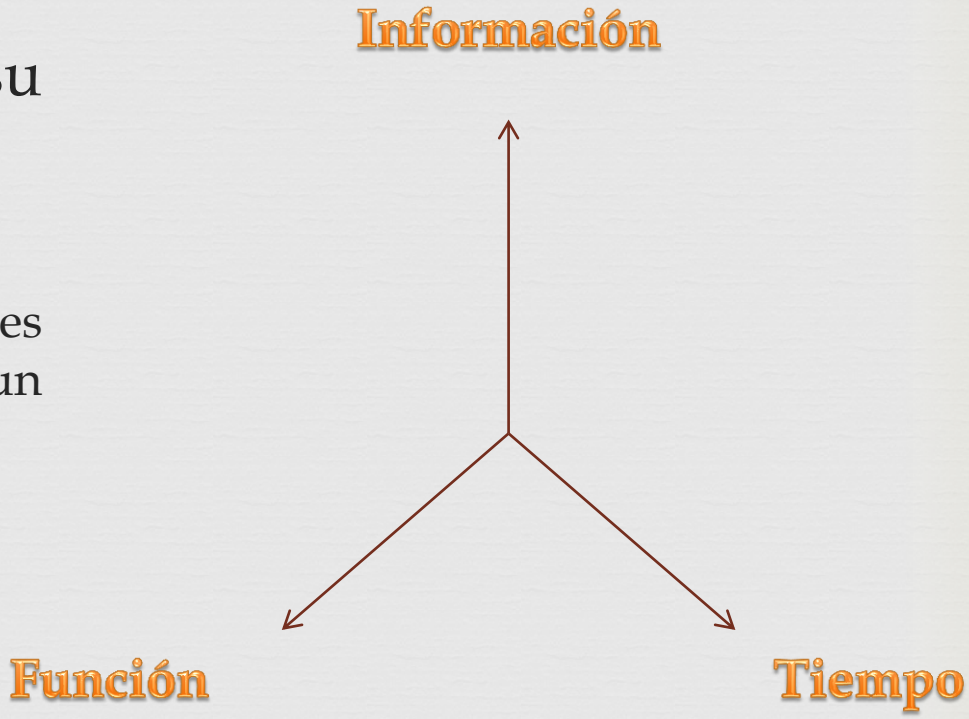
3. Requisitos específicos
    1. Requisitos funcionales
        1. Requisito funcional 1
            1. Introducción
            2. Entradas
            3. Procesamiento
            4. Salidas
        2. Requisito funcional 2
            1. ……
1. Requisitos de interfaz externa
    1. Interfaces de usuario
    2. Interfaces de hardware
    3. Interfaces de software
    4. Interfaces de comunicación
2. Requisitos de ejecución
3. Requisitos de diseño
    1. Acotamiento de estándares
    2. Limitaciones de hardware
4. Ámbitos de calidad
    1. Seguridad
    2. Mantenimiento
    3. …
5. Otros requisitos
    1. Base de datos
    2. Operaciones
    3. Adaptación de situación
    4. …

Especificación de requisitos del software

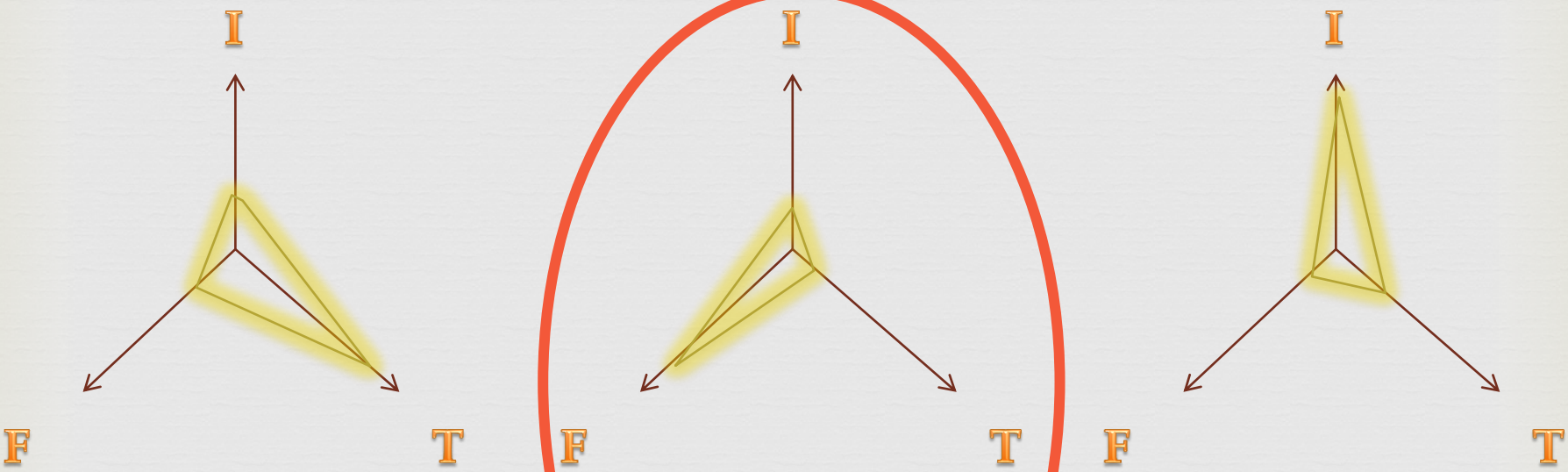
Clasificación según su enfoque de modelización

Yourdon [1993] presenta tres perspectivas para examinar un sistema:

- Función
- Información
- Tiempo



Especificación de requisitos del software



Sistemas de tiempo real

Sistemas de gestión orientados a funciones

Sistemas de gestión orientados a datos

Especificación de requisitos del software

	Información	**Función**	**Tiempo**
Información	Diagramas de entidad relación Diagramas de estructuras de datos Matriz entidad / entidad Diagrama de clases		
Función	Diagrama de flujo de datos Matriz función / entidad Diagrama de clases (nivel de diseño) Diagrama de colaboración	Diagrama de flujo de datos Diagrama de descomposición funcional Diagrama de estructura Diagrama de flujo Diagramas de casos de uso Diagramas de actividad Tarjetas CRC Diagramas de componentes Diagramas de despliegue	
Tiempo	Diagrama de historia y vida de entidad Diagramas de transición de estados Matriz evento / entidad Diagramas de secuencia	Redes de Petri Diagramas de transición de estados Diagramas de actividad Diagramas de secuencia	Lista de eventos Diagramas de transición de estados

Paradigma estructurado

Metodologías estructuradas

Entre los tres tipos de metodologías más empleados están la de Yourdon, la de Tom De Marco y la de Gane-Searson.

Se basan en las siguientes premisas:

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

Metodología de diseño estructurado de Yourdon

Proporciona una forma para diseñar paso a paso sistemas y programas detallados.

Entre los pasos que incluye están:

- El análisis
- El diseño
- Medición y mejora de al calidad

Metodologías estructuradas orientadas a procesos

Metodología de De Marco, conformada por 7 pasos:

Estudio del entorno físico actual

- Se plantea un modelo del sistema actual en el que se muestran los procedimientos actuales estén o no automatizados
- El modelo es verificable por el usuario

Derivación del correspondiente modelo lógico actual

- Se obtiene un modelo derivado del anterior, pero sin connotaciones físicas (por ejemplo lugares de la empresa o personas)

Derivación del nuevo modelo lógico

- Se toma en consideración las nuevas necesidades de los usuarios, estableciendo un modelo que describe aquello que hay que hacer, pero no cómo.
- El resultado es una especificación estructurada formada por los Diagramas de Flujo de Datos, Diccionario de Datos y especificación de procesos.

Crear un conjunto de modelos físicos alternativos

- A partir del modelo lógico se establecen varias alternativas de las que se escogerá la más conveniente
- Se analiza el enlace de los procesos con el usuario

Valorar cada opción

- Se estudian los costos y beneficios de los modelos físicos obtenidos en el paso anterior

Seleccionar una opción

- Se selecciona un modelo físico a partir de los datos derivados del paso anterior

Empaquetar la especificación

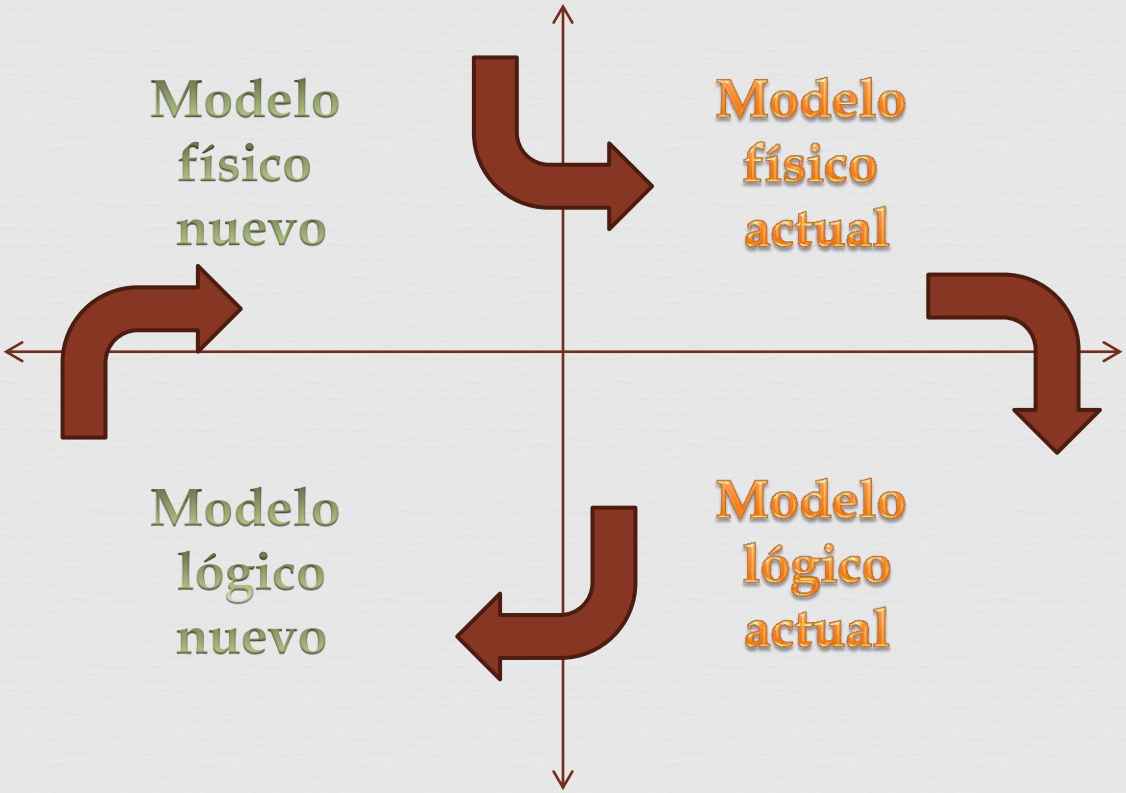
- Se recopila toda la documentación en un documento de especificación

Metodología de análisis de Gane y Searson

- Esta metodología se conforma por cinco apartados:

Paso 1	Paso 2	Paso 3	Paso 4	Paso 5
• Construir un modelo lógico en curso	• Construir un modelo lógico del nuevo sistema. • Construir una especificación estructurada que contenga los diagramas de flujo de datos, un diccionario de datos y las especificaciones de procesos	• Diseñar la base de datos física	• Crear un nuevo modelo físico del sistema	• Empaquetar la especificación en subsistemas

Metodologías estructuradas



Paradigma estructurado

- Herramientas base

Diagrama de Flujo de Datos

Diagrama de Flujo de Datos

Diagrama de Flujo de Datos (DFD)

- Es la técnica más difundida dentro del análisis estructurado
- Permite modelar funciones que debe realizar el sistema y los datos que fluyen entre ellas
- De Marco define esta técnica, y se apoya en otras de descripción textual como el diccionario de datos y la especificación de procesos
- Es un diagrama en forma de red que representa el flujo de datos y las transformaciones que se aplican sobre ellos al moverse de una entrada hasta una salida del sistema
- El sistema se modela mediante un conjunto de DFD nivelados, donde los niveles superiores definen funciones generales y los inferiores definen funciones detalladas

Diagrama de Flujo de Datos

Los diagramas de flujo de datos se catalogan como lógicos o físicos.

- Un diagrama de flujo de datos lógicos se enfoca en el negocio y en el funcionamiento de éste. No se ocupa de la manera en que se construirá el sistema.

- Un diagrama de flujo de datos físico muestra cómo se implementará el sistema, incluyendo el hardware, el software, los archivos y las personas involucradas en el sistema.

Diagrama de Flujo de Datos

Lógico

- Se enfoca en el negocio y en el funcionamiento de este
- No se ocupa de la manera en que se construirá el sistema
- Describe los eventos que ocurren en el negocio y los datos requeridos y producidos por cada evento

Físico

- Muestra como se implementará el sistema, incluyendo el hardware, el software, los archivos y las personas involucradas en el sistema

Diagrama de flujo de datos

Características	DFD Lógico	DFD Físico
Qué describe el modelo	Cómo funciona el negocio	Cómos e implementara el sistema (cómo funciona el sistema actual)
Qué representan los procesos	Las actividades del negocio	Programas, módulos del programa y procedimientos manuales
Qué representan los almacenes de datos	Colecciones de datos independientes de cómo se almacenan	Archivos y bases de datos físicos, archivos manuales
Tipos de almacenes de datos	Muestra almacenes de datos que representan colecciones de datos permanentes	Archivos maestros, archivos de transición. Cualquier proceso que opere en dos momentos diferentes debe conectarse mediante un almacén de datos
Controles del sistema	Muestra los controles del negocio	Muestra controles para validar los datos de entrada, para obtener registro, para asegurar realización exitosa

Diagrama de flujo de datos

Desarrollo de Diagramas de Flujo de Datos

Diagrama de flujo de datos

Desarrollo de Diagramas de Flujo de Datos Lógicos

Ventajas:

- Mejor comunicación con los usuarios
- Sistemas más estables
- Mejor entendimiento de los analistas
- Flexibilidad y mantenimiento
- Eliminación de redundancias y creación más sencilla del modelo físico

Diagrama de flujo de datos

Construcción de Diagramas de Flujo de Datos Físicos

- Aclarar qué procesos son manuales y cuales automatizados
- Describir los procesos con mayor detalle los DFD lógicos
- Distribuir en un orden particular los procesos que se deben realizar
- Identificar los almacenes de daos temporales
- Especificar los nombres reales de archivos y documentos impresos
- Agregar controles para asegurar que los procesos se realicen adecuadamente

Diagrama de flujo de datos

Contenido de los Diagramas de Flujo de Datos Físicos

- Procesos manuales
- Procesos para agregar, eliminar y modificar registros
- Procesos de entrada y verificación de datos
- Procesos de validación para garantizar la precisión de la entrada de datos
- Distribución de los procesos para reorganizar el orden de los registros
- Procesos para producir cada salida única del sistema
- Almacenes de datos intermedios
- Nombres de archivos reales para almacenar datos
- Controles para describir la terminación de tareas o condiciones de error

Diagrama de flujo de datos

Componentes:

- **Procesos:** Representan los componentes funcionales del sistema
- **Almacenes:** Representan datos almacenados o en reposo
- **Entidades externas:** Representan la fuente y/o destino de la información del sistema
- **Flujos de datos:** Representan los datos que fluyen entre las funciones

Diagrama de flujo de datos

Ejemplos:

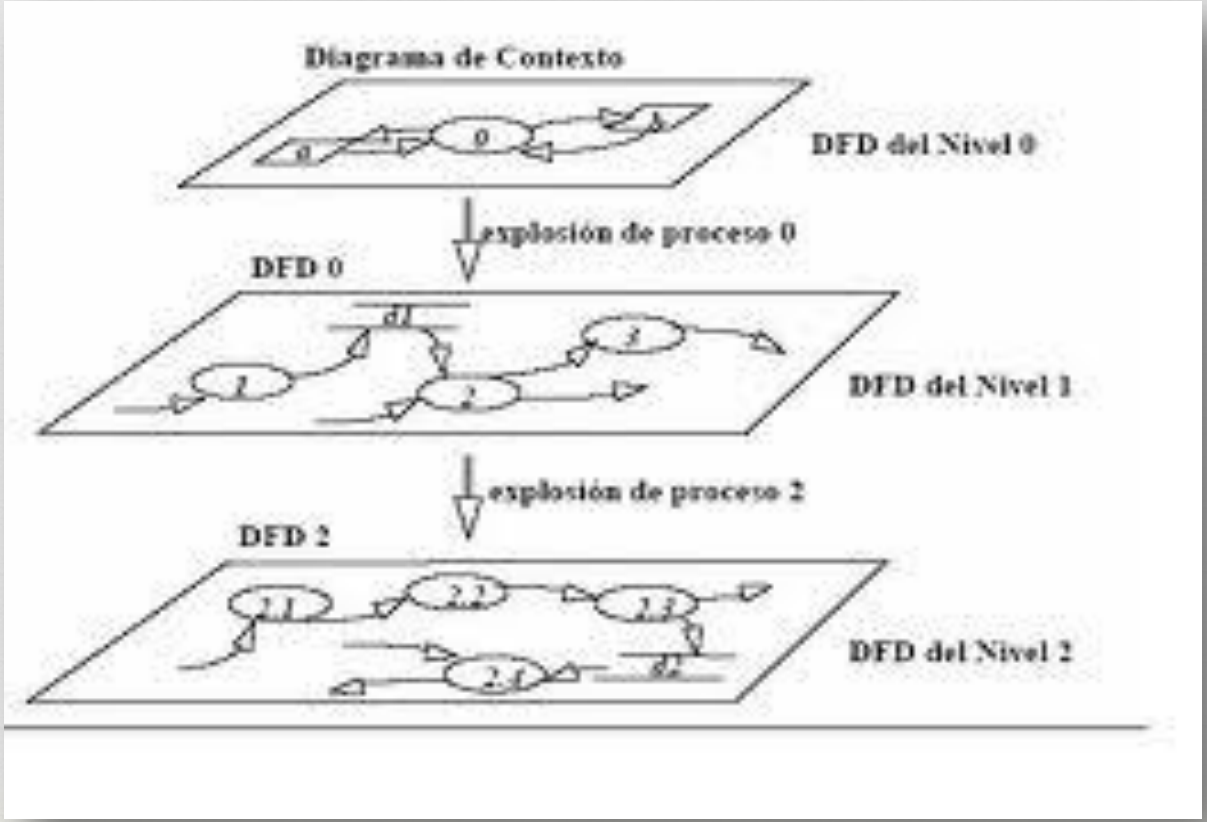
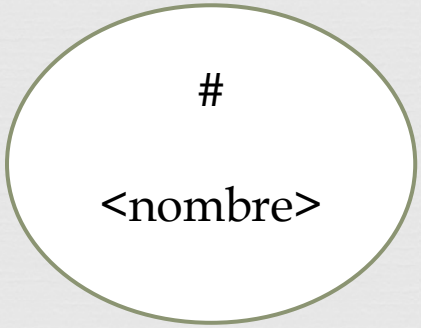


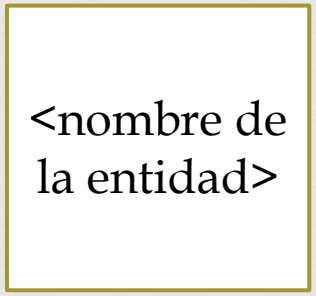
Diagrama de flujo de datos

Notación general:



Proceso

Almacén



Entidad externa

Flujo de datos

Diagrama de flujo de datos

#

<nombre del proceso>

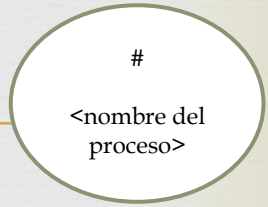
Procesos (funciones o transformaciones):

Transforman los flujos de entrada en uno o varios flujos de datos de salida

Son funciones que tiene que realizar el sistema

Realizan operaciones de transformación "cambio"

Diagrama de flujo de datos



Características de los procesos :

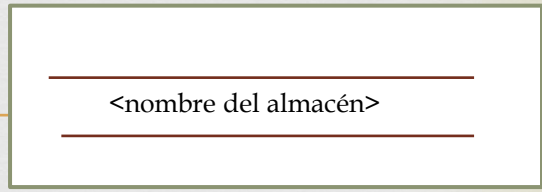
Deben ser lo más representativos posibles de la función que representan

Su referencia debe ser breve, normalmente formada por un verbo seguido de un sustantivo

Incluir el nombre y el numero del proceso. Deben ser únicos en el conjunto de DFD que representa el sistema

El detalle de los procesos se realiza en la descripción de procesos

Diagrama de flujo de datos



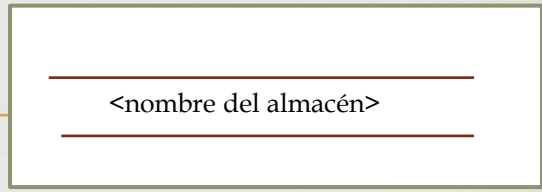
Almacenes de datos:

Representan información del sistema almacenada de forma temporal

Representan datos en reposo

Son depósitos lógicos de almacenamiento y por lo tanto pueden representar cualquier dato temporalmente almacenado independientemente del dispositivo utilizado

Diagrama de flujo de datos



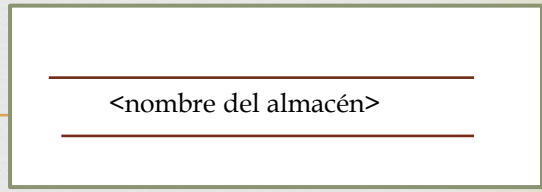
Características de los almacenes :

Todos los almacenes de datos deben llevar un nombre que debe ser lo más representativo posible y no estar asociado a connotaciones físicas

Suele estar en plural para sugerir que se pueden almacenar varios ejemplares o registros de información

Se puede representar varias veces en un DFD si con ello se mejora la legibilidad del diagrama

Diagrama de flujo de datos



Características de los almacenes :

En los DFD nivelados los almacenes se sitúan en el nivel más alto en el que sirve de conexión entre dos o más procesos, donde se representan todos sus accesos. También se incluye en los niveles inferiores

El contenido de los almacenes de define en el diccionario de datos

Diagrama de flujo de datos

<nombre de la entidad>

Entidades externas:

Representa un generador o consumidor de información del sistema y que no pertenece al mismo.

Puede representar un subsistema, una persona, departamento, organización u otra aplicación que proporcione datos al sistema o reciba datos de él

Diagrama de flujo de datos

<nombre de la entidad>

Características de las entidades externas:

Son elementos externos al sistema que está modelando. Los flujos que parten o llegan a ellas definen la interfaz entre el sistema y el mundo exterior

Las relaciones que haya entre las entidades externas no son objeto del estudio del modelo. Por lo que no se representan los posibles flujos entre ellas

Según Yourdon se representan en el diagrama mediante un cuadro en el que se indica el nombre en su interior que debe ser representativo

Las entidades externas se pueden dibujar varias veces en un DFD con objeto de mejorar al legibilidad del mismo. Las entidades duplicadas se marcan con un * (asterisco)

Normalmente solo aparecen en el DFD de mayor nivel (diagrama de contexto)

Diagrama de flujo de datos

<nombre de la entidad>

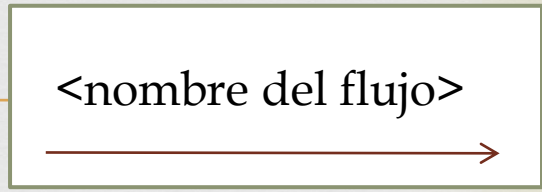
Características de las entidades externas:

Son elementos externos al sistema que está modelando. Los flujos que parten o llegan a ellas definen la interfaz entre el sistema y el mundo exterior

Las relaciones que haya entre las entidades externas no son objeto del estudio del modelo. Por lo que no se representan los posibles flujos entre ellas

Según Yourdon se representan en el diagrama mediante un cuadro en el que se indica el nombre en su interior que debe ser representativo

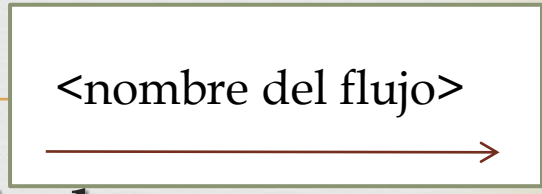
Diagrama de flujo de datos



Flujos de datos:

- Camino a través del cual viajan datos de composición conocida de una parte del sistema a otra
- Representan los datos en movimiento en un momento determinado y con una cardinalidad determinada
- Son el medio de conexión de los restantes componentes del DFD
- Se representan por medio de un arco dirigido, donde la flecha indica la dirección de los datos

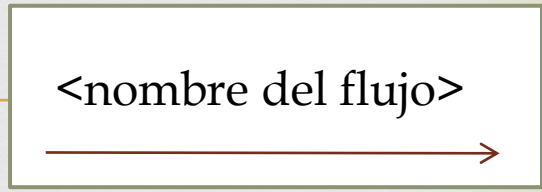
Diagrama de flujo de datos



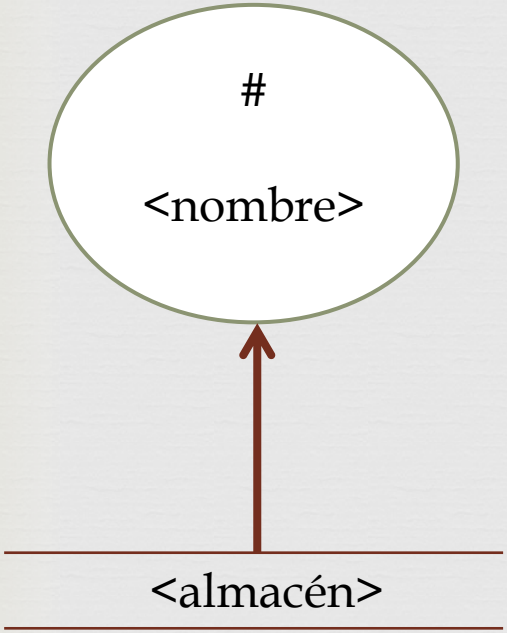
Restricciones de conexión con los flujos de datos:

Destino / fuente	Proceso	Almacén	Entidad externa
Proceso	Si	Si	Si
Almacén	Si	No	No
Entidad externa	Si	No	No

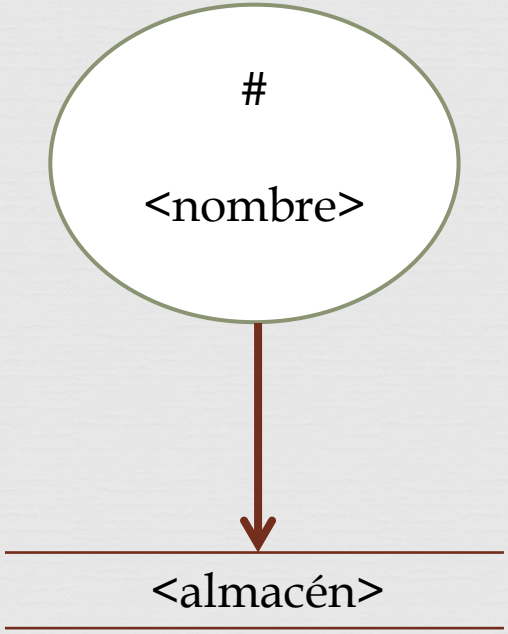
Diagrama de flujo de datos



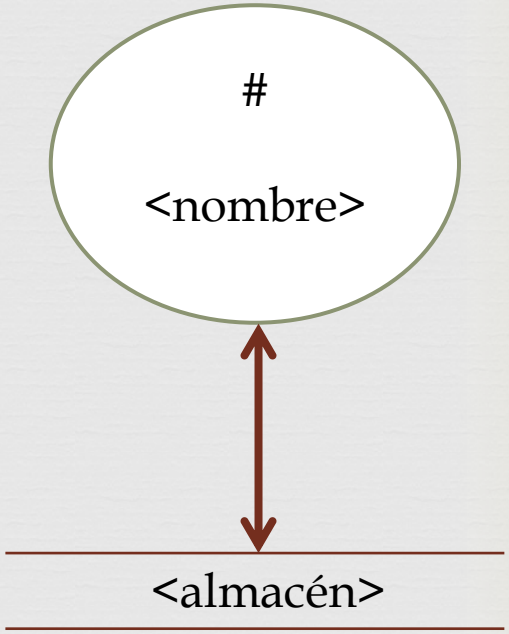
Conexiones entre procesos y flujos de datos:



Flujo de consulta



Flujo de actualización



Flujo de diálogo

Diagrama de flujo de datos

Símbolos empleados por distintos autores:

	Yourdon, De Marco	Gane y Sarson	SSADM Métrica
Flujos de datos			
Procesos			
Almacenes de datos			
Entidades externas			

Diagrama de flujo de datos

Descomposición en niveles de un DFD:

Se sigue una aproximación **descendente** en el que cada nivel proporciona una visión más detallada de una parte definida en el nivel anterior

- Ayuda a construir la especificación de arriba abajo
- Los distintos niveles pueden ir dirigidos a personas diferentes
- Facilita el trabajo de los analistas que pueden trabajar paralelamente modelando funciones independientes del sistema
- Facilita la documentación del sistema

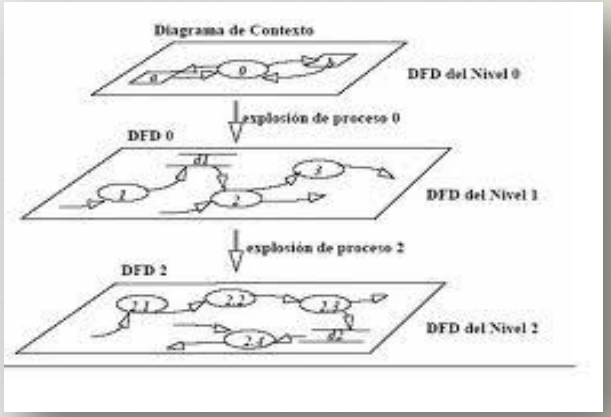


Diagrama de flujo de datos

Construcción del DFD:

Diagrama de contexto

- Único
- Se ubica en la parte superior
- Es muy general

Niveles medios

- Formado por el resto de los diagramas

Funciones primitivas

- Presentes tanto en los niveles intermedios como en los últimos niveles de la jerarquía
- Corresponde con procesos que no se detallan en nuevos DFD

Diagrama de flujo de datos

Construcción del DFD:

- Se comienza con el nivel más alto en la jerarquía (diagrama de contexto)
- Este proceso se descompone en otro DFD (denominado diagrama de sistema) donde se representan las funciones principales o subsistemas
- Se descomponen cada uno de los procesos en nuevos diagramas que representan funciones más simples
- Se procede de la misma manera hasta que todas las funciones son lo "suficientemente detalladas" como para que no sea necesaria la creación de nuevos DFD

Diagrama de flujo de datos

Creación de un Diagrama de Flujo de datos

- Haga una lista de las actividades del negocio y úsela para determinar lo siguiente:
  - Entidades externas
  - Flujos de datos
  - Procesos
  - Almacenes de datos
- Cree un diagrama de contexto que muestre las entidades externas y los flujos de datos desde y hacia el sistema
- Dibuje el diagrama (Muestre procesos, pero que sean generales)
- Cree un diagrama hijo para cada uno de los procesos del diagrama 0
- Revise que no haya errores y asegurese de que sean significativos los nombres asignados a procesos y flujos
- Desarrolle un diagrama de flujo de datos físico a partir del diagrama de flujo de datos lógico
- Distinga entre procesos manuales y automatizados
- Particione el diagrama de flujo de datos físico se propósito de facilitar la programación y la implementación

Diagrama de flujo de datos

Diagrama de contexto:

- Primer diagrama en la jerarquía
- Conocido también como Diagrama de Nivel 0
- Permite delimitar la frontera entre el sistema con el mundo exterior y definir sus interfaces
- Se forma exclusivamente por un proceso que representa una "caja negra" del sistema completo, un conjunto de entidades externas que representen la procedencia y el destino de la información del sistema y un conjunto de flujos de datos de entrada y salida del sistema con el entorno
- Deben representarse todas las entidades externas (según varios autores es el único lugar donde deben aparecer)

Diagrama de flujo de datos

Diagrama de contexto:

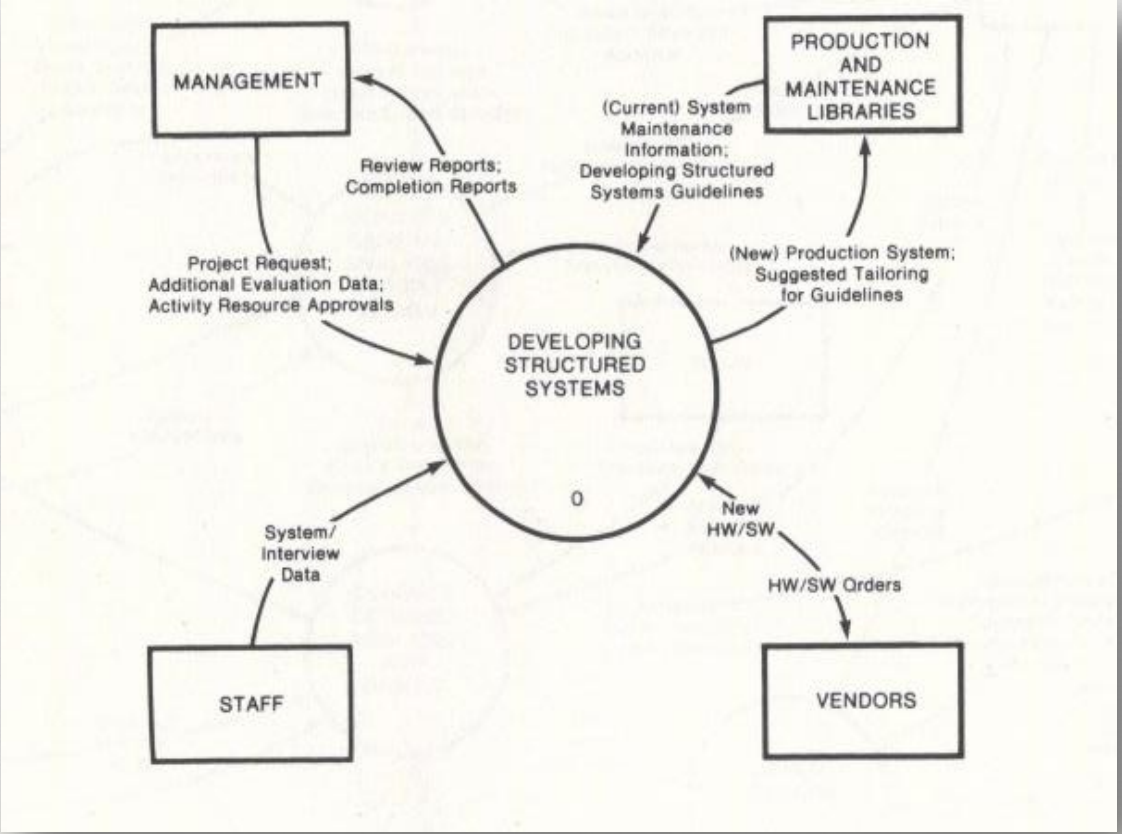


Diagrama de flujo de datos

Diagrama del sistema (Diagrama 0):

- Es el que descompone el diagrama de contexto
- También se le denomina diagrama del sistema, ya que en el se representan las funciones principales que el sistema debe realizar, así como la relación entre ellas
- Las funciones de este diagrama son conceptualmente independientes entre si, lo que facilita la descomposición de cada una por personas distintas.
- Se le denomina diagrama 0 porque se corresponde con la definición del proceso 0. Algunos autores lo denominan también diagrama de nivel 1.

Diagrama de flujo de datos

Diagrama del sistema (Diagrama 0):

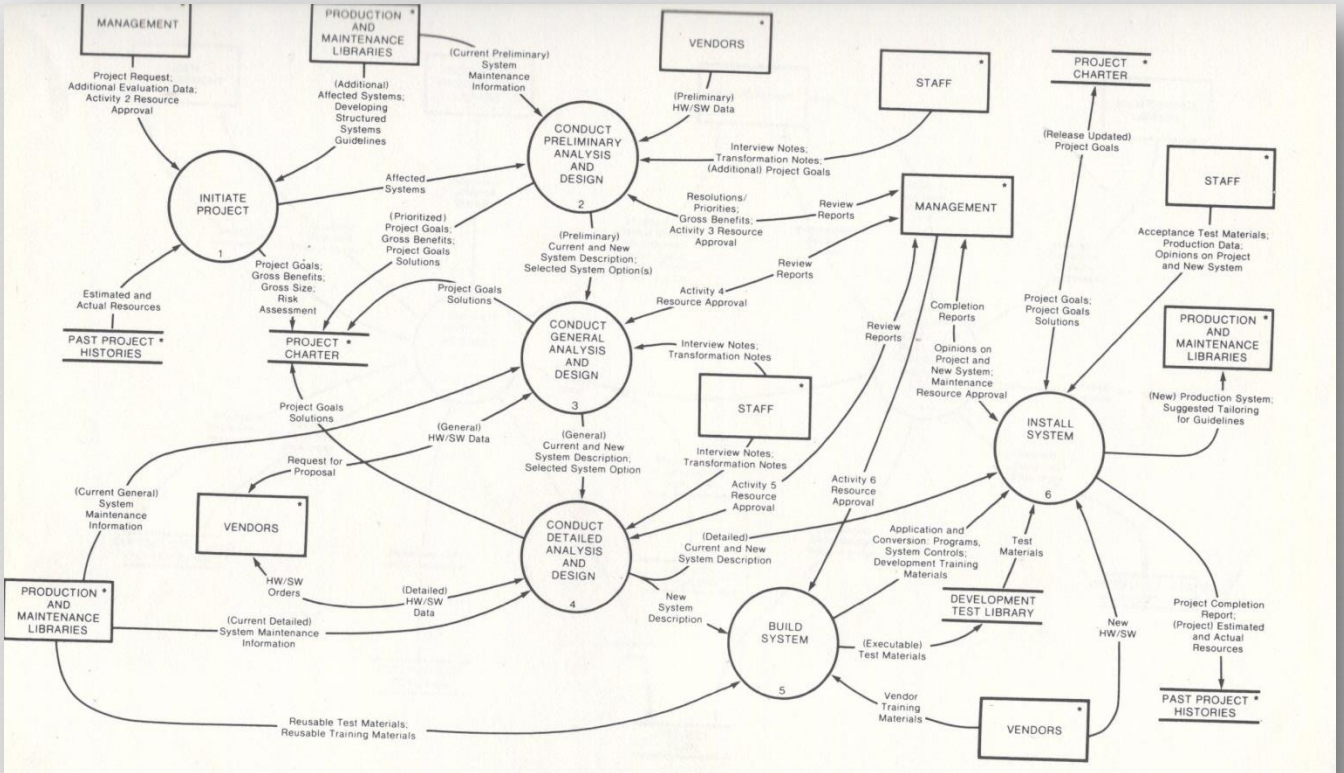


Diagram 0 Developing Structured Systems

Diagrama de flujo de datos

Procesos primitivos:

- También llamadas funciones primitivas
- Son aquellos procesos de un DFD que ya no se descomponen en más diagramas de nivel inferior
- Para cada función primitiva habrá una especificación que la describe

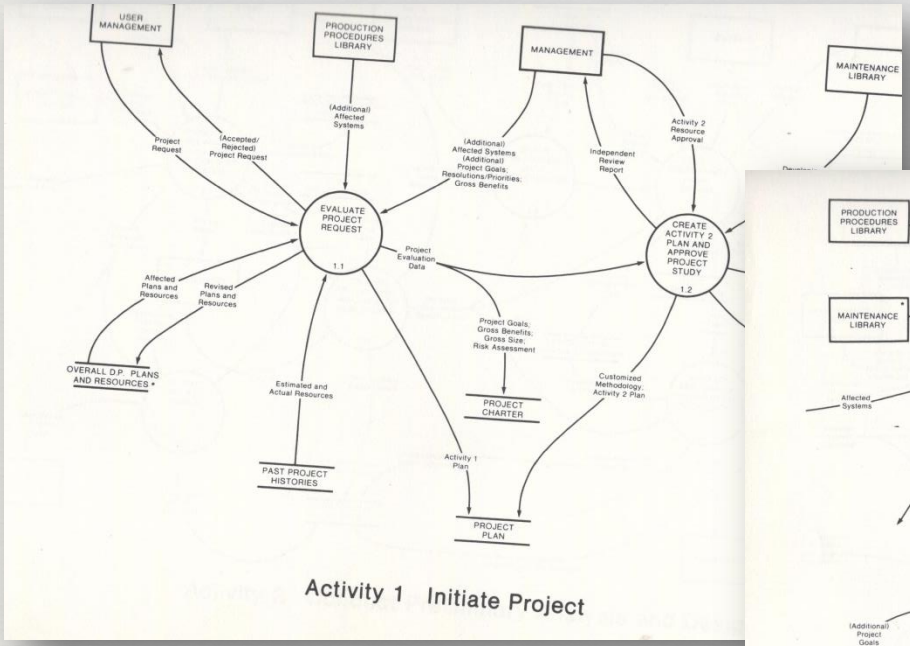
Diagrama de flujo de datos

Descomposición a procesos primitivos:

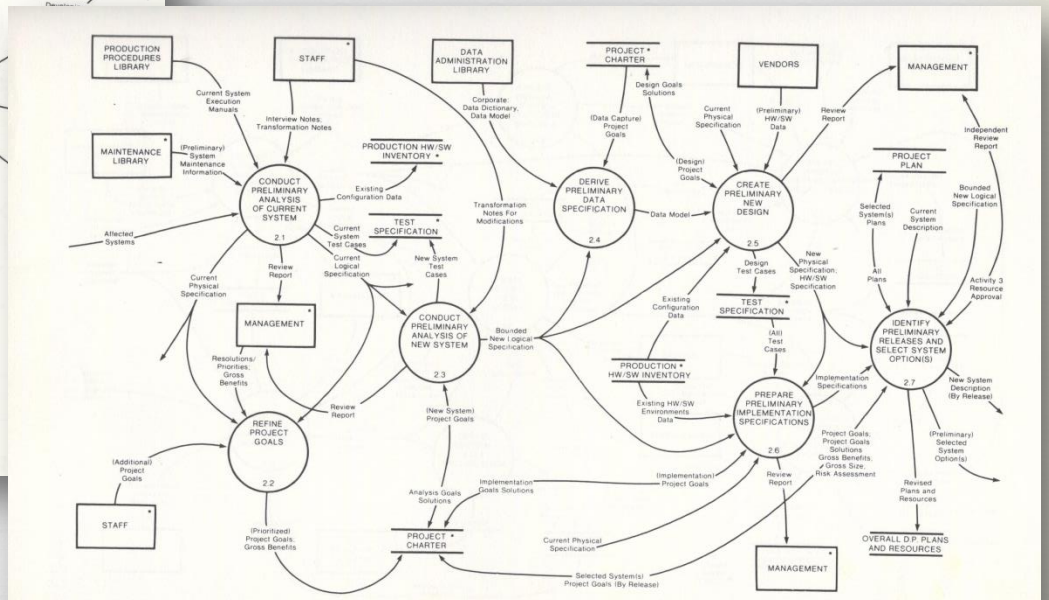
- Cuando un requisito funcional se puede especifica en menos de una página mediante un lenguaje de especificación o pseudocódigo
- Cuando los procesos del diagrama tienen pocos flujos de datos de entrada y salida
- Cuando al descomponer una función de un nivel determinado, se pierde el significado de lo que tiene que hacer esa función

Diagrama de flujo de datos

Procesos primitivos:



Activity 1 Initiate Project



Activity 2 Conduct Preliminary Analysis and Design

Diagrama de flujo de datos

Descomposición a procesos primitivos:

Nivel 0

- Diagrama de contexto

Nivel 1

- Subsistemas

Nivel 2

- Funciones de cada subsistema

Nivel 3

- Sub funciones asociadas a cada uno de los eventos del sistema

Nivel 4

- Procesos necesarios para el tratamiento de cada sub función

Diagrama de flujo de datos

Consistencia entre niveles (balanceo)

La información que entra y sale de un proceso de nivel N, sea consistente con la información que entra y sale del DFD en el que se descompone

- Todos los flujos de datos que entran en un diagrama hijo deben estar representados en el padre por el mismo flujo de datos entrando al proceso
- La salida del diagrama hijo deben ser las mismas salidas del proceso padre asociado

Diagrama de flujo de datos

Consistencia entre niveles (balanceo)

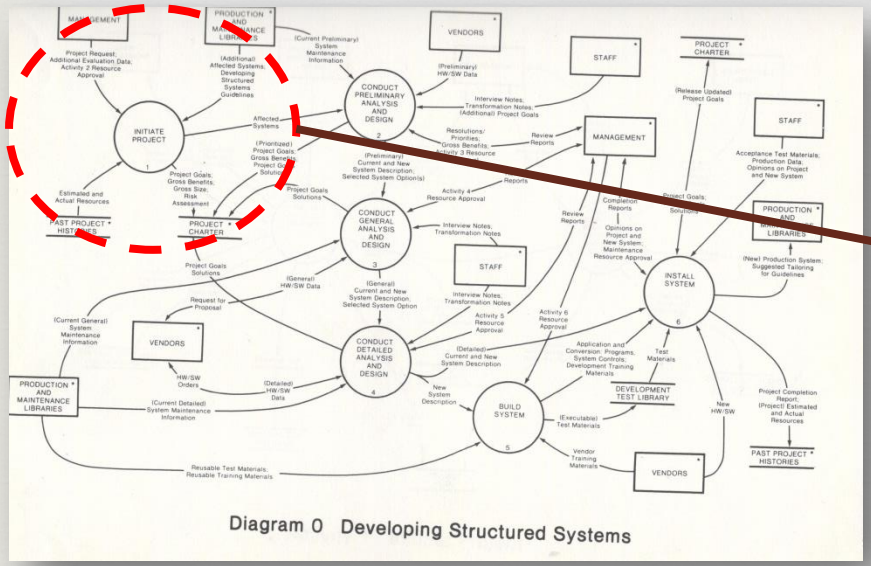
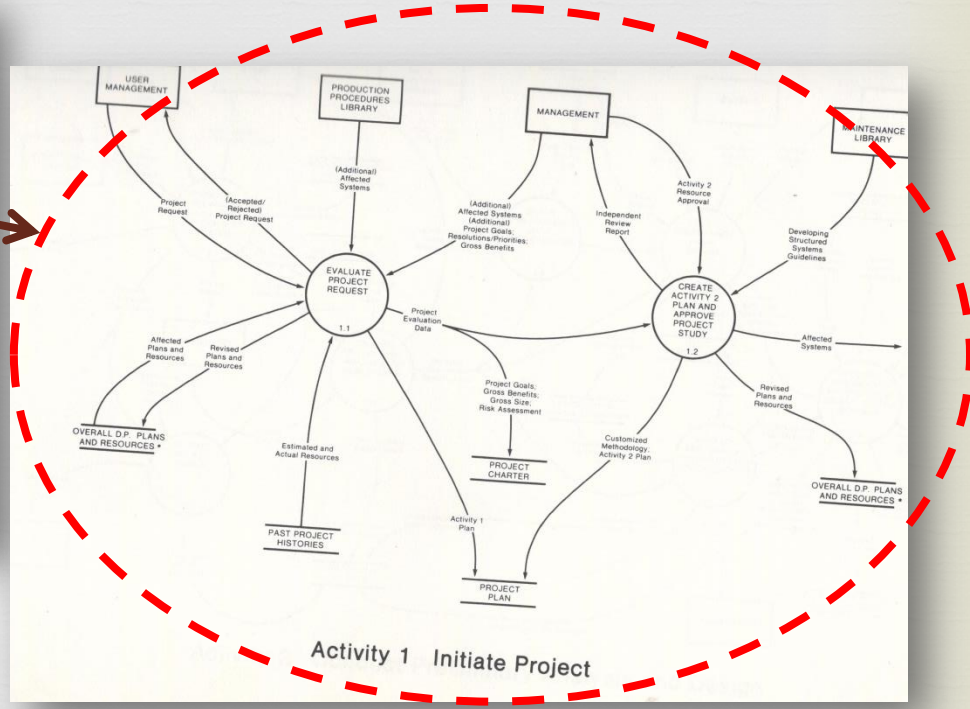


Diagram 0 Developing Structured Systems



Activity 1 Initiate Project

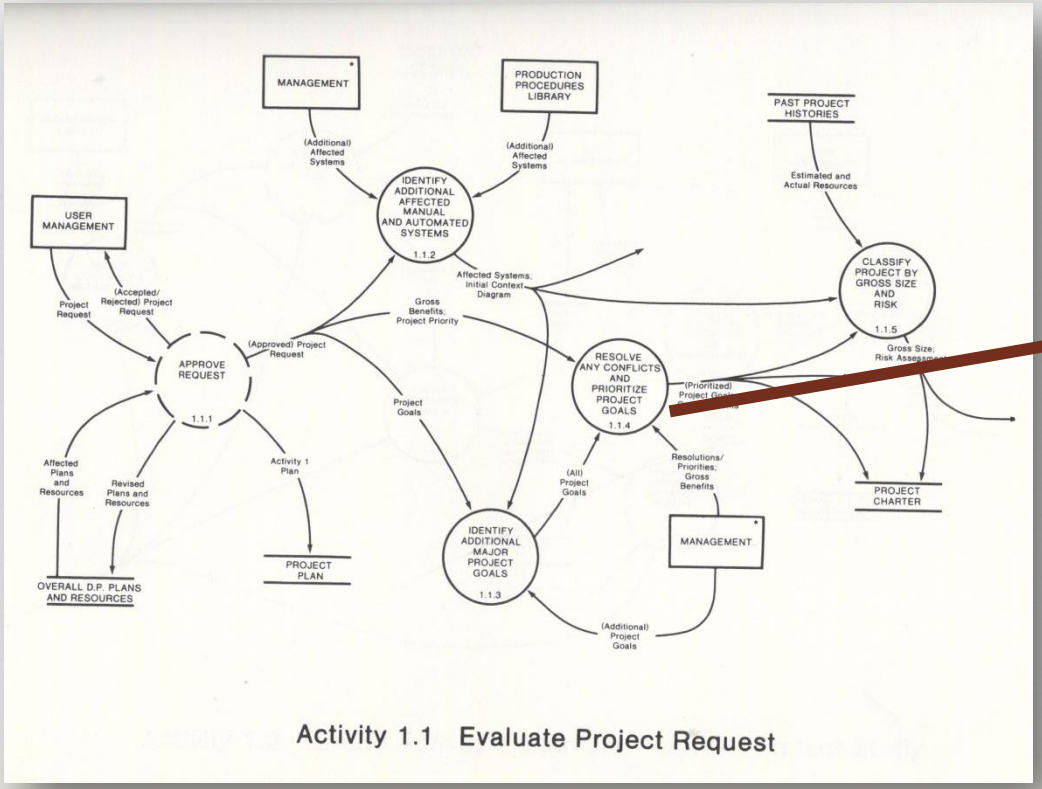
Diagrama de flujo de datos

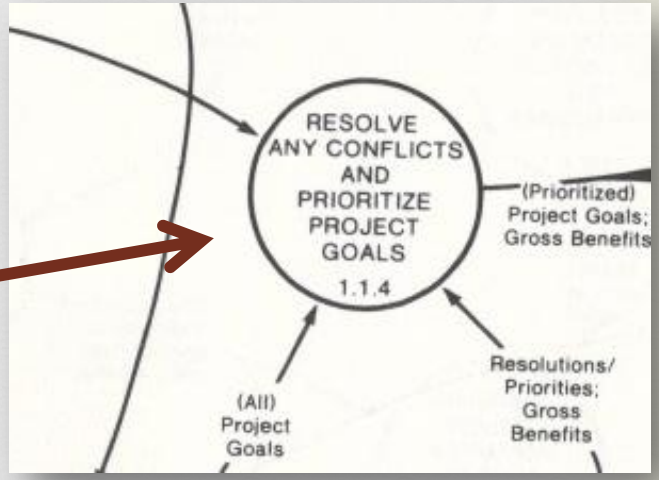
Numeración

- Cada diagrama recibe el número y nombre del proceso que descompone
- El proceso del diagrama de contexto siempre es numerado con cero
- Los procesos del diagrama del sistema se numera por u entero comenzando por 1 y de forma creciente
- En los restantes niveles, los números de los procesos están formados por la concatenación del número del diagrama en el que están más un punto y un numero entero único para identificarlo dentro del diagrama

Diagrama de flujo de datos

Numeración





Actividad I

Una experiencia común que los estudiantes en cada universidad comparten es inscribirse a un curso

A. Dibuje un diagrama de flujo de datos de nivel 1 del movimiento de los datos para matricularse en un curso de la universidad. Use una sola hoja y etiquete claramente cada elemento de datos
B. Amplíe uno de los procesos de su diagrama de flujo de datos original en subprocesos, agregando flujos de datos y almacenes de datos
C. Mencione las partes del proceso de inscripción que están "ocultas" al observador externo y a las cuales ha tenido que hacer suposiciones para completar el diagrama del nivel inferior

Diccionario de Datos

Diccionario de datos

Una vez completados los Diagramas de Flujos de Datos se procede a catalogar toda la información vertida en los diagramas en el Diccionario de Datos

El Diccionario de Datos (DD) es una aplicación especializada acerca de los datos (metadatos)

Recopila y coordina términos de datos específicos y confirma lo que cada término significa para las diferentes personas de la organización

Diccionario de datos

Es una lista organizada de los datos utilizados por el sistema que gráficamente están representados por los flujos de datos y almacenes presentes sobre el conjunto de DFD

Se crea con los DFD durante el análisis del sistema

Las entradas son realizadas cada vez que se identifica un elemento y pueden ser de tres tipos:

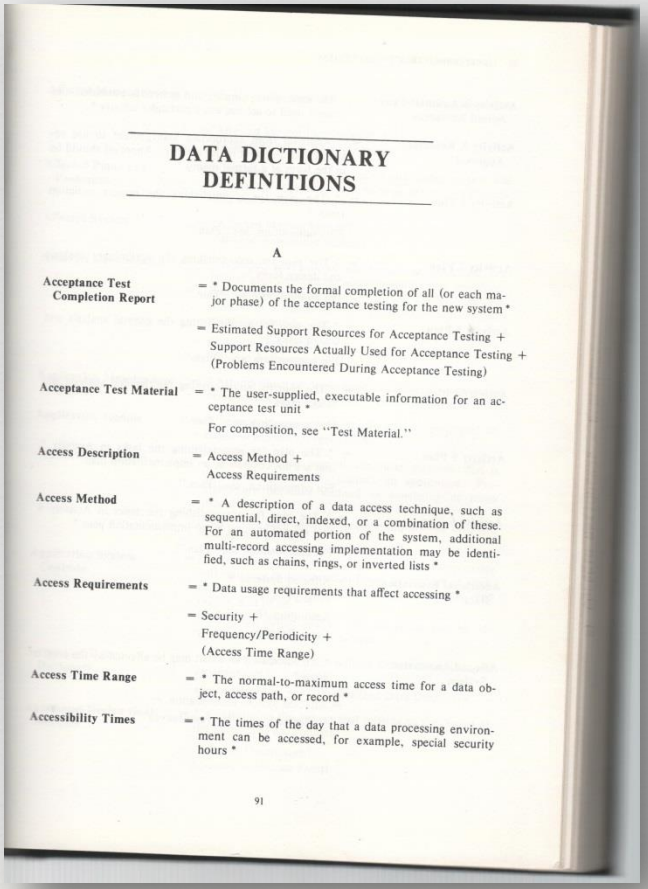
- Flujos de datos
- Almacenes
- Datos elementales

Diccionario de datos

Objetivo

- Validar la integridad y exactitud del diagrama de flujo de datos
- Proporcionar un punto de partida para desarrollar pantallas e informes
- Determinar el contenido de los datos almacenados en archivos
- Desarrollar la lógica para los procesos del diagrama de flujo de datos

Diccionario de datos



DATA DICTIONARY DEFINITIONS

A

Acceptance Test Completion Report = * Documents the formal completion of all (or each major phase) of the acceptance testing for the new system *

= Estimated Support Resources for Acceptance Testing +
Support Resources Actually Used for Acceptance Testing +
(Problems Encountered During Acceptance Testing)

Acceptance Test Material = * The user-supplied, executable information for an acceptance test unit *

For composition, see "Test Material."

Access Description = Access Method +
Access Requirements

Access Method = * A description of a data access technique, such as sequential, direct, indexed, or a combination of these. For an automated portion of the system, additional multi-record accessing implementation may be identified, such as chains, rings, or inverted lists *

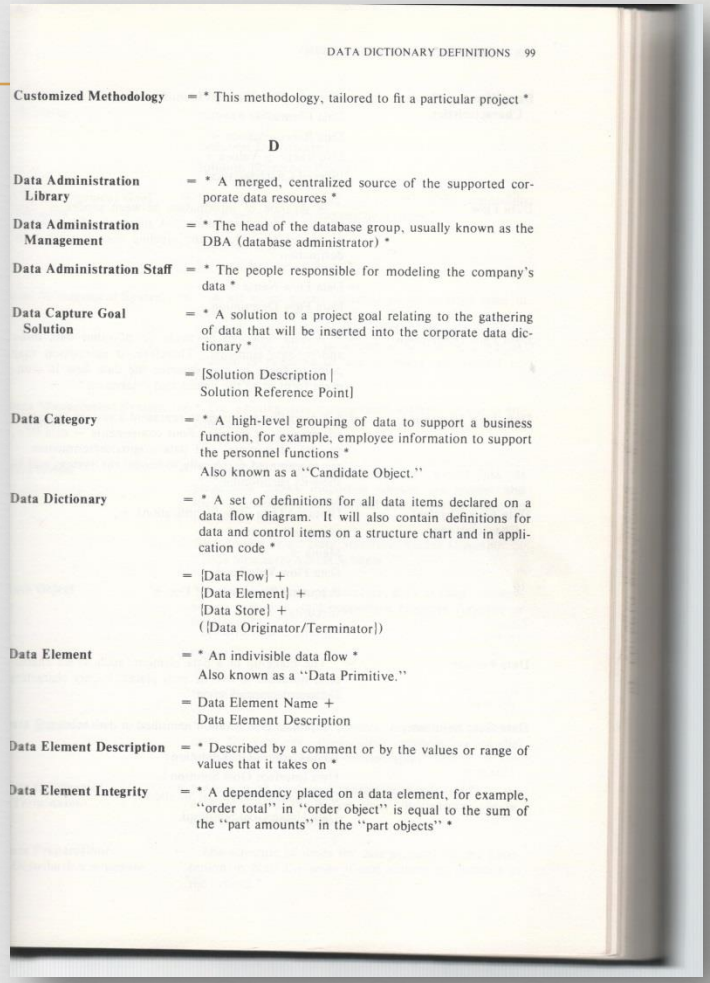
Access Requirements = * Data usage requirements that affect accessing *

= Security +
Frequency/Periodicity +
(Access Time Range)

Access Time Range = * The normal-to-maximum access time for a data object, access path, or record *

Accessibility Times = * The times of the day that a data processing environment can be accessed, for example, special security hours *

91



Customized Methodology = * This methodology, tailored to fit a particular project *

D

Data Administration Library = * A merged, centralized source of the supported corporate data resources *

Data Administration Management = * The head of the database group, usually known as the DBA (database administrator) *

Data Administration Staff = * The people responsible for modeling the company's data *

Data Capture Goal Solution = * A solution to a project goal relating to the gathering of data that will be inserted into the corporate data dictionary *

= {Solution Description |
Solution Reference Point}

Data Category = * A high-level grouping of data to support a business function, for example, employee information to support the personnel functions *
Also known as a "Candidate Object."

Data Dictionary = * A set of definitions for all data items declared on a data flow diagram. It will also contain definitions for data and control items on a structure chart and in application code *

= {Data Flow} +
{Data Element} +
{Data Store} +
({Data Originator/Terminator})

Data Element = * An indivisible data flow *
Also known as a "Data Primitive."

= Data Element Name +
Data Element Description

Data Element Description = * Described by a comment or by the values or range of values that it takes on *

Data Element Integrity = * A dependency placed on a data element, for example, "order total" in "order object" is equal to the sum of the "part amounts" in the "part objects" *

Diccionario de datos

Flujos de datos

Sigue una aproximación “top-down”

Los componentes son definidos a su vez mediante componentes más detallados y así se continua hasta ya no poderlos dividir más

A=B + C

B=B1+B2+B3

C=C1+C2

Diccionario de datos

Flujos de datos

Símbolo	Nombre	Significado
=	Composición	Compuesto de o equivalente a
+	Inclusión	Agregado y
[ ]	Selección	Selección de una de las opciones encerradas entre corchetes y separadas por el símbolo |
{ }	Iteración	Iteraciones del componente encerrado entre llaves
( )	Opción	El componente encerrado es opcional (puede estar presente o ausente)
texto	Comentario	Comentario aclarativo de una entrada en el DD
@	Identificador	Campo o conjunto de campos que identifican cada ocurrencia en el almacén

Diccionario de datos

Usos de los símbolos en el Diccionario de Datos

Tarjeta_bibl = num + apellidos + nombre + tipo_tarjeta

Composición e inclusión

Socio= nombre + domicilio +[RFC | IFE]

Selección

Prestamos={ libros }
Libros=Clasificación +título + autor
Prestamos=1 { libros } 5

Iteración

Tarjeta_bibl= num + apellidos + nombre +
tipo_tarjeta + (num_telefono)

Opción

Diccionario de datos

Almacenes

Se definen como entidades repetitivas de datos y/o grupos de datos

Se lista de forma detallada la composición de cada repositorio colocando el nombre de cada uno de los elementos que lo integran

Diccionario de datos

Información a registrar en los almacenes

- ID del almacén de datos
- Nombre del almacén de datos
- Alias para el archivo (si existe)
- Breve descripción del almacén
- Tipo de archivo (manual o automatizado)
- Si es automatizado colocar formato del archivo
- Número máximo y promedio de registros en el archivo
- % de crecimiento
- Nombre del conjunto de datos que especifica el nombre del archivo
- Estructura de datos (debe estar en DD)

Diccionario de datos

Alías

- Empleados cuando se modelan datos que se nombran de forma distinta pero se refieren exactamente a lo mismo
- Es un sinónimo de una entrada del DD ya sea un flujo de datos o elemento de datos
- No se fomenta mucho su utilización porque se crean redundancias en la especificación estructurada

Petición_libro=tarjeta_biblioteca+ficha_libro

Petición_libro=Petición_de_prestamo

Diccionario de datos

Datos elementales

Cada elemento de datos se debe definir una vez en el DD

Información a registrar

- ID del elemento
- Nombre del elemento
- Alias
- Descripción breve
- Determinar si el elemento es base o derivado
- Longitud del elemento
- Tipo de datos
- Formatos de entrada y salida
- Criterios de validación
- Valores predeterminados
- Observaciones

Diccionario de datos

Uso del Diccionario de Datos

- El diccionario ideal es automatizado, interactivo, en línea y evolutivo
- Para maximizar su potencial, el DD se debe vincular a varios programas de sistemas para que con cualquier modificación siempre este actualizado
- Se debe ver como una actividad paralela al análisis y diseño de sistemas

Actividad II

Tomando como base los diagramas planteados en la actividad I

A. Genere por lo menos 10 entradas en el Diccionario de Datos, incluyendo:
   I. Almacenes
   II. Flujos de Datos
   III. Descripción de datos primitivos

Especificación de procesos

Especificación de procesos

- También llamada mini especificación
- Técnica que define el procedimiento que realiza un proceso primitivo
- Describe de una manera más o menos formal cómo se obtienen los flujos de datos de salida a partir de los flujos de datos de entrada, más información local del proceso

Especificación de procesos

Descripción de procedimiento

Process name: APPROVE REQUEST
Process reference number: 1.1.1
Process guided by: General department plan

Process notes: If the project request comes directly from a user without being reviewed by data processing management, verify the following:

- that your department is the appropriate one to act on this request
- that the requestor is authorized to initiate a project
- that the contents of the request are clear and complete (as declared in the data dictionary definition)

The most common problem encountered is that the need for the project is stated as a solution, "I want an on-line system to store and retrieve customer records," instead of as a business objective, "I want to improve customer relations with faster turnaround on inquiries." In analysis, you may find the turnaround problems were not due to storing and retrieving speed, but instead were due to unreadable reports. The on-line solution may just give *faster* unreadable reports. Analysis will identify the problem; therefore, finding a solution will best be left until after analysis.

After verifying the request, identify the following:

- whether the request conflicts with an existing request, duplicates an existing request, or conflicts with any restrictions, such as federal regulations or company policy
- whether further evaluation should be deferred, for example, whether the product that the request supports is likely to be phased out, or whether the project is an extension to an existing system that is going to be rewritten or that is in the process of development (which means the request can be accommodated only in the next release)

If the request is not rejected or deferred and if you have the resources available for an initial study, update the overall data processing plans and resources accordingly and develop a project plan. This plan should identify only the scheduling of resources for initial project evaluation.

Inform the project requestor (user management) of the outcome of this process.

Process name: PREPARE ACTIVITY 3 DEVELOPMENT PLAN (BY RELEASE)
Process reference number: 2.7.3
Process guided by: Activity 2 plan

Process notes: Having completed all of the development activities for analysis and design at a preliminary level and having identified releases for the new system, you are in a good position to prepare the development plan for the general-level study. Decide whether you will conduct the general-level study using just the first release or whether the complete system will be needed. Your decision will depend on the system's size and on the available time and resources, but you may not be the decision maker in this case. It may be the user management's decision. Therefore, you may wish to develop a plan for each option and release for presentation to management.

Having completed the preliminary study, you have five factors to help you develop the Activity 3 plan:

1. familiarity with the system
2. deliverables and processes already partly identified in the methodology
3. preliminary current system description and preliminary new system description, with releases defined and with the areas to study identified in more detail
4. figures for estimated and actual resources used in Activity 2
5. results from any blitz conducted during the preliminary study that are used to identify general-level detail required

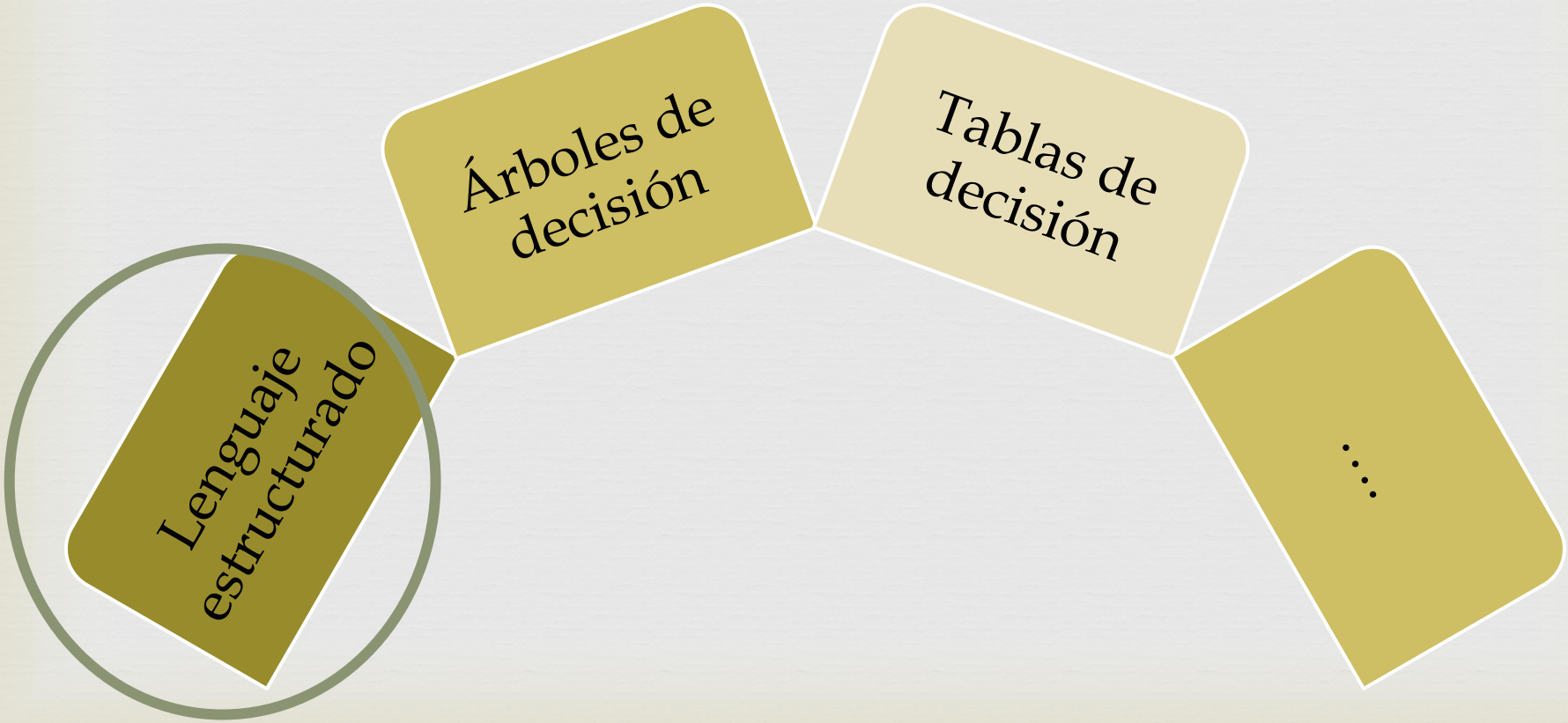
Using these five factors, prepare the development plan for Activity 3. Include all tasks, resources, dependencies, and information about schedules needed to conduct the general-level analysis and design.

The plan should be as detailed as possible for Activity 3, but for cost/benefit analysis and feasibility purposes, you should estimate the plans needed for each release for Activities 4 through 6, excluding those releases already covered in the implementation plans. The plan may include the use of manpower resources from outside the project team, such as a database group, the user, and the operations staff (confirm their availability from management).

Note: Even though this is a managerial task because the agents are the project manager and management, the best source for identification of the effort involved in Activity 3 is the project team.

Especificación de procesos

Descripción de procedimiento



Especificación de procesos

Usando lenguaje estructurado

```
IF
    IF
        IF
        ELSE
        ENDIF
    ELSE IF
        IF
        ELSE
        ENDIF
    ELSE
        IF
        ELSE
        ENDIF
    ENDIF
ELSE

ENDIF
```

Estructura secuencial

- Bloque de acciones en el cual no ocurren bifurcaciones

Estructura de decisión

- Solo cuando la condición es verdadera complete las instrucciones de otra manera pase al ELSE

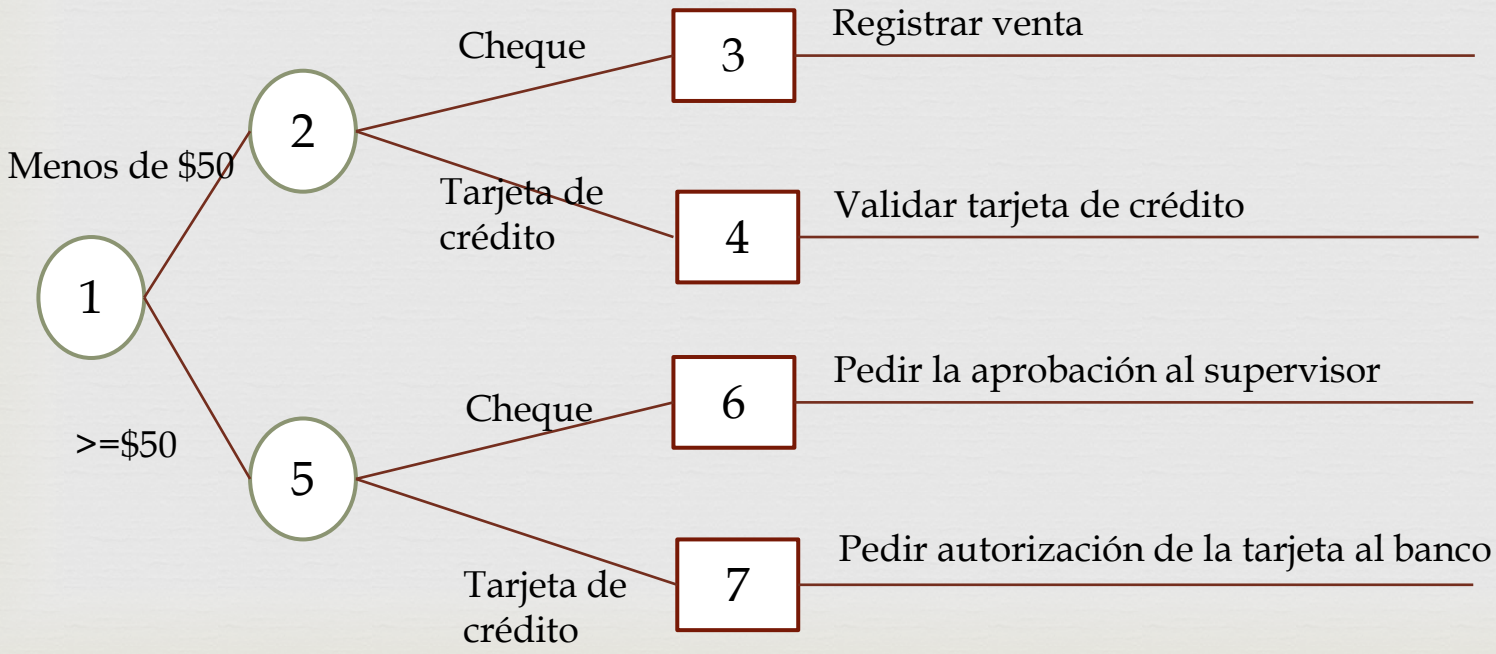
Estructura de iteración

- Bloques de instrucciones que se repiten hasta que se completan

Especificación de procesos

Usando árboles de decisión

Usados cuando ocurre una bifurcación compleja en un proceso de decisión estructurada. También son útiles cuando es necesario mantener una cadena de decisiones en una secuencia particular



Especificación de procesos

Usando tablas de decisión

Tabla de filas y columnas en cuatro cuadrantes

Especificación de procesos

Usando tablas de decisión

Condiciones y acciones	1	2	3	4
Menor de $50	S	S	N	N
Paga con cheque	S	N	S	N
Usa tarjeta de crédito	N	S	N	S
Registra una venta	X			
Valida tarjeta de crédito		X		
Busca aprobación al supervisor			X	
Pedir autorización de la tarjeta al banco				X

Especificación de procesos

Metas para producir especificaciones de proceso

Reducir ambigüedad	Obtener descripción precisa	Validar el diseño del sistema
• Obliga al analista a prender acerca del funcionamiento de un proceso	• Identificar claramente lo que se esta realizando, se incluye en un paquete de especificaciones para el programador	• Incluye garantizar que un proceso tenga todo el flujo de datos de entrada necesario para producir la salida • Todas las entradas y salidas deben representarse en el DFD

Especificación de procesos

Procesos que no requieren especificación

- Procesos que representan entrada o salida física (requieren sólo lógica simple)
- Validación de datos simple, la cual es bastante fácil de realizar
- Procesos que incluyen código pre escrito, generalmente se incluyen en un sistema como subprograma y funciones

Especificación de procesos

Elementos que deben ser incluidos

- Número del proceso
- Nombre del proceso
- Descripción breve de lo que realiza el proceso
- Lista de flujos de datos de entrada
- Flujos de datos de salida
- Indicación del tipo de proceso (por lote, en línea o manual)
- Nombre de los subprogramas (en caso de ser código pre escrito)
- Descripción de la lógica del proceso que indique las políticas y reglas del negocio en lenguaje cotidiano, no en pseudocódigo
- Mencionar cualquier problema a resolver partes incompletas de la lógica u otras consideraciones

Especificación de procesos

Process name: APPROVE REQUEST
Process reference number: 1.1.1
Process guided by: General department plan

Process notes: If the project request comes directly from a user without being reviewed by data processing management, verify the following:

- that your department is the appropriate one to act on this request
- that the requestor is authorized to initiate a project
- that the contents of the request are clear and complete (as declared in the data dictionary definition)

The most common problem encountered is that the need for the project is stated as a solution, "I want an on-line system to store and retrieve customer records," instead of as a business objective, "I want to improve customer relations with faster turnaround on inquiries." In analysis, you may find the turnaround problems were not due to storing and retrieving speed, but instead were due to unreadable reports. The on-line solution may just give *faster* unreadable reports. Analysis will identify the problem; therefore, finding a solution will best be left until after analysis.

After verifying the request, identify the following:

- whether the request conflicts with an existing request, duplicates an existing request, or conflicts with any restrictions, such as federal regulations or company policy
- whether further evaluation should be deferred, for example, whether the product that the request supports is likely to be phased out, or whether the project is an extension to an existing system that is going to be rewritten or that is in the process of development (which means the request can be accommodated only in the next release)

If the request is not rejected or deferred and if you have the resources available for an initial study, update the overall data processing plans and resources accordingly and develop a project plan. This plan should identify only the scheduling of resources for initial project evaluation.

Inform the project requestor (user management) of the outcome of this process.

Actividad III

Tomando como base los diagramas planteados en la actividad I

A. Realice la descripción de procesos del diagrama del último nivel, utilice por lo menos tres técnicas distintas para realizar dicha especificación:

   I. Pseudocódigo
   II. Matriz / Árbol de decisión
   III. Texto libre

Especificación estructurada

Especificación estructurada

Especificación estructurada

Comprobaciones a realizar sobre la especificación estructurada

Yourdon indica que es conveniente realizar la revisión en base a cuatro aspectos:

- Compleción (Si los modelos son completos)
- Integridad (Si no existen contradicciones o inconsistencias)
- Exactitud (Si los modelos cumplen los requisitos del usuario)
- Calidad (Se revisa el estilo, legibilidad y facilidad de mantenimiento de los modelos)

Especificación estructurada

Lista de comprobación sugerida

Calidad

- Todos los componentes tienen nombre
- Todos los procesos tienen numero
- Todos los procesos primitivos tienen una especificación de proceso asociada
- Todos los flujos están definidos en el DD
- Todos los elementos de datos están definidos

Especificación estructurada

Lista de comprobación sugerida

Integridad

- Hay elementos definidos en el DFD no incluidos en el DD
- Los almacenes de datos representados en los DFD están definidos en el DD
- Los elementos de datos referenciados en las especificaciones de proceso están definidos en el dD
- Los flujos de datos de entrada y salida de un proceso primitivo se corresponden con las entradas y salidas de la especificación de proceso
- Hay errores de balanceo
- Hay procesos que tienen solo entradas o solo salidas
- Por cada proceso se cumple la regla de conservación de datos
- Hay flujos de entrada superfluos al proceso

Especificación estructurada

Lista de comprobación sugerida

Integridad

- Hay flujos de control o flujos de datos como activadores de procesos
- Los procesos pueden generar los flujos de salida a partir de los de entrada más información local a proceso
- Hay pérdida de información en los procesos
- Hay almacenes sólo con entradas o sólo con salidas
- Hay conexiones incorrectas entre los elementos del DFD
- Hay almacenes locales
- Es correcta la dirección de las flechas de los DFD
- Existen redes desconectadas

Especificación estructurada

Lista de comprobación sugerida

Exactitud

- Cada requisito funciona del usuario tiene asociado uno o más procesos primitivos en los DFD

Calidad

- El diagrama es claro
- Hay nombres de componentes con poca significacia
- Hay muchos flujos de entrada y salida en procesos primitivos

Actividad IV

Tomando como insumos las herramientas desarrolladas en las actividades I, II y III realice la verificación de la especificación estructurada considerando los aspectos que Yourdon propone:

- Compleción
- Integridad
- Exactitud
- Calidad

Especificación estructurada

Otra forma de verificación

Sample Table of Contents for a MANUAL SYSTEM MODEL

	Current/New Environment*		
■ **Physicalized Leveled Data Flow Diagram(s):** Levels developed incrementally	P	G	D
■ **Physicalized Data Dictionary:**			
□ For each major data flow (documents/transaction streams):			
○ Data flow name	P		
○ Data flow description (optional)	P		
○ Physical data flow ID (if different from name)	P		
○ Media	P		
○ Data flow volume	P		
○ Frequency/periodicity of use	P		
○ Security (if applicable to whole stream)	P		
○ Special characteristics (optional)	P		
□ For each component data flow (documents/transactions):			
○ Data flow name	P		
○ Data flow description (optional)		G	
○ Physical data flow ID (if different from name)		G	
○ Sample/layout (optional) (at detailed level in new environment)		G	
○ Media (if different from whole stream)		G	
○ Data flow volume		G	
○ Frequency/periodicity (if different from whole stream)		G	
○ Security (optional)		G	
○ Special characteristics (optional)		G	
□ For each data element (fields/attributes):			
○ Data element name		G	
○ Data element description			D
○ Physical data element ID (optional)			D
○ Data format			D
○ Data representation			D
○ List/range of values			D
○ Special characteristics (optional)			D
□ For each major data store (files):			
○ Data store name	P		
○ Data store description	P		
○ Physical store ID (optional)	P		
○ Sample/layout (optional) (at detailed level in new environment)	P		
○ Media	P		
○ Organization	P		
○ Frequency/periodicity of use (if not governed by data flow)	P		
○ Access management system (optional)	P		
○ Min-max store volume (estimate)	P		
○ Security (if applicable to whole store)	P		
○ Special characteristics (optional)	P		

*P = Preliminary, G = General, D = Detailed

	Current/New Environment		
□ A data model (data structure diagram) (if available in current environment) or for each record/document: (See "For each component data flow," above)	P		
□ For each data field: (See "For each data element," above)			
■ **Physicalized Process Descriptions:**			
□ For each major process (for example, departments):			
○ Process name	P		
○ Process notes (optional)	P		
○ Yearly operating costs (estimate)	P		
○ Special characteristics (optional)	P		
□ For each component process (for example, sections/groups):			
○ Process name	P		
○ Process notes (optional)		G	
○ Yearly operating costs (estimate)		G	
○ Special characteristics (optional)		G	
□ For each individual process (procedures/tasks):			
○ Process name		G	
○ Process specification			D
○ Doer of the process (his/her name or a title) (optional)			D
○ Skill level required			D
○ Frequency/periodicity of use (if not driven by data flow)			D
○ Yearly operating cost			D
○ Special characteristics (optional)			D
■ **Organizational Hierarchy** (optional) developed incrementally	P	G	D

Especificación estructurada

Sample Table of Contents for an AUTOMATED SYSTEM MODEL

	Current Environment*			New Environment*		
	P	G	D	P	G	D
■ **Design Charts:**						
□ Different types of design charts						
■ **Automated Data Dictionary:**						
□ For each major data stream (transaction/report groups):						
○ Stream name	P			P		
○ Stream description (optional)	P			P		
○ Media	P			P		
○ Stream volume	P			P		
○ Frequency/periodicity of use	P			P		
○ Security (if applicable to whole stream)	P			P		
○ Special characteristics (optional)	P			P		
□ For each stream component (transactions/reports):						
○ Transaction ID	P				G	
○ Transaction name (if different from ID)	P			P		
○ Sample/layout (optional)		G				D
○ Media (if different from whole stream)		G			G	
○ Transaction volume		G			G	
○ Frequency/periodicity (if different from whole stream)		G			G	
○ Security (optional)		G			G	
○ Special characteristics (optional)		G			G	
□ For each data element (fields/attributes):						
○ Data element name		G			G	
○ Data element description			D			D
○ Data format			D			D
○ Data representation			D			D
○ List/range of values			D			D
○ Special characteristics (optional)			D			D
□ For each data store (files and intermediate files):						
○ Data store ID	P		D			
○ Data store name (if different from ID)	P			P	G	D
○ Data store description	P			P	G	D
○ Sample/layout	P					
○ Media	P			P	G	D
○ Organization	P					D
○ Min-max store volume	P			P	G	D
○ Access method	P				G	D
○ Security (if applicable to whole store)	P			P	G	D
○ Frequency/periodicity (if applicable to whole store)	P			P	G	D
○ Special characteristics (optional)	P			P	G	D

* P = Preliminary, G = General, D = Detailed

	Current Environment			New Environment		
	P			P	G	D
□ A data model (logical and/or physical, if available) or for each record/document:						
○ (See "For each stream component," above, reading transactions as records)						
□ For each data field:						
○ (See "For each data element," above)						
■ **Automated Process Descriptions:**						
□ For each automated job (batch/on-line, front-end processing, and so on):						
○ Job name	P			P		
○ Job description	P			P	G	
○ Job size/complexity (estimate)	P			P	G	
○ Frequency/periodicity (if not governed by data)	P			P	G	
○ Executing processor type (if applicable to whole job)	P			P	G	
○ Yearly operating costs (estimate)	P			P	G	
○ Special characteristics (optional)	P			P	G	
□ For each program:						
○ Program ID	P					D
○ Program name	P					D
○ Program description (optional in new environment)		G				
○ Frequency/periodicity of use (if not governed by data)		G				D
○ Executing processor type (if different from whole job)		G				D
○ Source language		G				D
○ Program size/complexity		G				D
○ Cost per execution		G				D
○ Special characteristics (optional)		G				D
□ For each module (routines):						
○ Module ID		G				D
○ Module name		G			G	D
○ Module description			D		G	D
○ Frequency of use (if not governed by data)			D			D
○ Pseudocode (for current environment — actual code listing)			D			D
○ Source language (if not same) for whole program			D		G	D
○ Special characteristics (optional)				D		D
■ **System flowchart**	P					D

Especificación estructurada

Sample Table of Contents for a LOGICAL SPECIFICATION

	*Current/New Environment**		
■ **Logical Leveled Data Flow Diagram(s):**	P	G	D
□ Levels developed incrementally			
■ **Logical Data Dictionary:**			
□ For each major data flow (originally documents, transactions, and report streams or time-delayed stores):			
○ Data flow name	P		
○ Data flow description (optional)	P		
○ Cross reference to current physical definition	P		
□ For each component data flow (originally documents, transactions, reports, or records):			
○ Data flow name	P		
○ Data flow description (optional)		G	
○ Cross reference to current physical definition		G	
□ For each data element:			
○ Data element name		G	
○ Data element description			D
○ Cross reference to current physical definition			D
■ **Data Model** (optional in manual and automated logical models):			
○ Data categories	P		
○ Informal data structure/access diagram		G	
○ Formal data structure/access diagram			D
□ For each data object (or group) (originally records/documents):			
○ Data object name	P		
○ Data object description		G	
○ Cross reference to current physical definition		G	
□ For each data attribute (originally fields or elements) (See "For each data element," above)			
■ **Logical Data Store** (for manual and automated logical models):			
□ For each (necessary) data store:			
○ Data store name	P		
○ Data store description	P		
○ Cross reference to current physical definition	P		
□ For each data store component (originally records/documents): (See "For each component data flow," above)			
□ For each data element (originally fields): (See "For each data element," above)			

* P = Preliminary, G = General, D = Detailed

	Current/New Environment		
■ **Logical Process Descriptions:**			
□ For each major function (originally departments or automated jobs):			
○ Process name	P		
○ Process notes (optional)	P		
○ Cross reference to current physical process(es)	P		
□ For each component function (originally sections/groups or programs):			
○ Process name	P		
○ Process notes (optional)		G	
○ Cross reference to current physical process(es)		G	
□ For each functional primitive (originally procedures/tasks or modules/routines):			
○ Process name		G	
○ Process specification			D
○ Cross reference to current physical process(es)			D

Bibliografía

- Dickinson, Brian (1980). Developing Structured System . A methodology using structured techniques.. Yourdon Press.

- Kendall K. y Kendall J.(2005) Análisis y diseño de sistemas. 6ª edición. Pearson/Prentice Hall.

- Piattini M, Calvo- Manzano J, Cervera J y Fernández L. (2004) Análisis y diseño de aplicaciones informáticas de gestión. Una perspectiva de ingeniería del software. Alfaomega-RaMa.

- Pressman, R. (2005) Ingeniería del software. 6ª ed. McGraw-Hill.