**CONTENIDO**

1. Introducción. 2
   1. Análisis y diseño estructural. 3
   2. Diagramas de flujos de datos. 3
   3. Descripción de procesos. 7
   4. Diccionario de datos. 10
   5. Sistemas modulares. 13
   6. El proceso de diseño de software: Conceptos y principios
   7. fundamentales de diseño. 14
2. Conclusión 19

**Introducción**

El diseño estructurado es un enfoque disciplinado de la transformación de qué es necesario para el desarrollo de un sistema, a cómo deberá ser hecha la implementación. El análisis de requerimientos del usuario (determinación del qué) debe preceder al diseño y que, al finalizar el diseño se tendrá medios para la implementación de las necesidades del usuario (el cómo), pero no se tendrá implementada la solución al problema.

El diseño estructurado de sistemas se ocupa de la identificación, selección y organización de los módulos y sus relaciones. Se comienza con la especificación resultante del proceso de análisis, se realiza una descomposición del sistema en módulos estructurados en jerarquías, con características tales que permitan la implementación de un sistema que no requiera elevados costos de mantenimiento. La idea original del diseño estructurado fue presentada en la década de los '70, por Larry Constantine, y continuada posteriormente por otros autores: Myers, Yourdon y Stevens.

1.1 Análisis y diseño estructural.

1.1.1.- Análisis estructural.

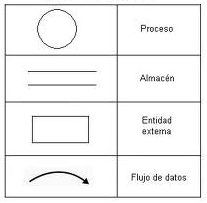
Es un método para el análisis de sistemas manuales o automatizados que conduce al desarrollo de especificaciones para sistemas nuevos. allí, se efectúan modificaciones y mejoras necesarias tanto al procedimiento como al sistema, tomando en cuenta los beneficias que el mismo pueda traer a la organización. el análisis estructural se representa a través de componentes tales como: símbolos gráficos, diccionario de datos, procesos, procedimientos y reglas.

1.1.2.- Diseño estructural.

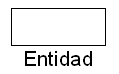
El diseño estructural define las relaciones entre los principales elementos del programa, su principal objetivo es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos. el diseño estructurado se modela a través de diseño de datos, diseño arquitectónico, diseño de interfaz, diseño procedimental.

1.2.- Diagrama de flujos de datos.

Es un modelo lógico-gráfico, que ayuda a representar el funcionamiento de un sistema, este permite incorporar opciones para el depurado de algoritmos. facilitando la localización de errores de ejecución  lógicos más habituales. la representación gráfica se realiza a través de proceso, flujo de datos, entidades externas, almacén. ver la siguiente figura.



1.2.1.- Elementos de los diagramas de flujos de datos (DFDs):

* **Entidad Externa**: Representan entidades ajenas del sistema que aportan o reciben información del mismo puede ser un usuario del sistema, una persona u organización ajena a la empresa u otro sistema que proporcione datos al sistema o que lo reciba de él.

Reglas de construcción (Entidades externa)

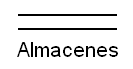
* Cada entidad externa se representa mediante un rectángulo.
* Las entidades externas pueden aparecer varias veces en el diagrama de flujo. se hace con el objetivo de evitar los cruces de los flujos de datos y dar una mayor legibilidad al diagrama
* No se puede representar la comunicación entre entidades externas
* Solo aparece en el diagrama de contexto
* **Proceso:** Representa una función que genera flujo de datos de salida a partir de la transformación del flujo de datos entrada y de, en ocasiones datos locales al proceso.

Reglas de construcción (procesos).

* Cada proceso se representa mediante un círculo.Y un número que indique el proceso.
* No se puede repetir un mismo proceso en el mismo Diagrama de flujo.
* Un proceso no es origen ni final de los datos.
* Un proceso puede transformar un dato en varios.
* Es necesario un proceso entre una entidad externa y un almacén de datos.
* **Almacenes:** Representa depósitos de información dentro del sistema. Es conveniente distinguir las diferentes utilidades que presentan los almacenes de datos.

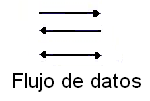
A.- **Almacenes Permanentes:** Es donde se guardan los datos que sirven de referencia de uso del sistema (Base de datos o archivos).

B.- **Almacenes Transitorios:** Es la que soluciona la comunicación asíncrona entre los procesos



Reglas de construcción (Almacenes)

* Cada almacén se representa mediante un rectángulo abierto por los lados.
* Un almacén no puede crear ni destruir datos.
* Aparece en el nivel en que es accedido por dos o más procesos
* El mismo almacén puede aparecer varias veces en un Diagrama de Flujo. se hace con el objeto de evitar los cruces de los flujos de datos y dar una mayor legibilidad al diagrama
* **Flujo de Datos:** Representa la comunicación entre procesos y almacenes; entre procesos y entidades externas; pero nunca habrá comunicación entre entidades externa y almacenes. son los movimientos de información en el sistema.

****

Reglas de construcción (Flujo de datos)

* Cada flujo se representa mediante una línea terminada en flecha en uno o en los dos extremos, debe tener un nombre único. que indique la información que contiene el flujo
* Un flujo no puede crear ni destruir datos.
* Un flujo no es un activador de proceso.
* Un flujo puede ser.
  + De consulta o lectura (flecha hacia el proceso).
  + De actualización o escritura (flecha hacia el almacén).
  + de diálogo, los flujos de diálogo sólo se utilizan en los niveles superiores, en los inferiores se deben descomponer en flujo simple de consulta y actualización.

**1.2.2.- Las reglas de los diagramas de flujo de datos (DFDS):**

* Encontrar todas las entidades externas, para poder encontrar los límites del sistema.
* Elegir nombres con significado, tanto para proceso como también para flujo de datos, almacenes y entidades externas (sin ocupar terminologías técnicas).
* Identificar y enumerar los procesos en una cierta secuencia lógica de ejecución
* Unir los elementos relacionados entre sí.
* Los almacenes de datos, entidades externas y otros procesos deben quedar todos relacionados, es decir, poseer una entrada y una salida para no ser datos sueltos.
* Cada proceso puede ser explotado para convertirse en un DFD por sí mismo.

**1.2.3.- Niveles de los Diagrama de flujos**

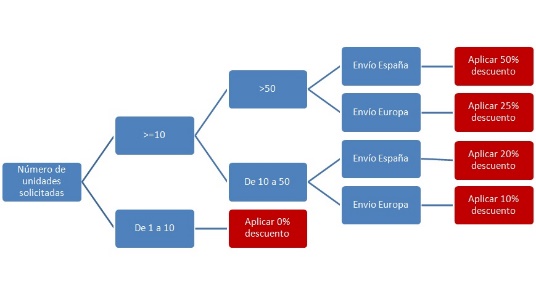
Los diagramas derivados de los procesos principales se clasifican en niveles, los cuales son:

* Nivel 0 (Diagrama de contexto): El diagrama solo modela el proceso principal del problema en cuestión con sus respectivas entidades. Cada proceso debe tener al menos una entrada y una salida de datos.
* Nivel 1 (Diagrama de nivel superior): En este diagrama de nivel superior se plasma todos los procesos que describen al proceso principal. en este nivel aparecen los almacenes. los cuales tienen la capacidad de almacenar o enviar datos para ser usados en distintos procesos.
* Nivel 2 (Diagrama de detalle o expansión) En este diagrama generan procesos provenientes de niveles anteriores.
  1. **Descripción de procesos**

Sirve para describir los procesos a nivel más detallado. Esta especificación comienza desde el nivel más alto del diagrama de flujo de datos, donde los procesos se dividen de manera recursiva, con ayuda de subdiagramas hasta que existen procesos suficientemente pequeños que sean fáciles de implementar. Esta especificación puede ser expresada con lenguajes estructurados, árboles de decisión, tablas de decisión, diagramas de acción, pre y post condiciones.

* **Lenguaje estructurado:** es un lenguaje formado por un subconjunto de palabras (del idioma elegido) para formar construcciones de la programación estructurada. Representa un típico pseudocódigo de alto nivel que utiliza sentencias en español.

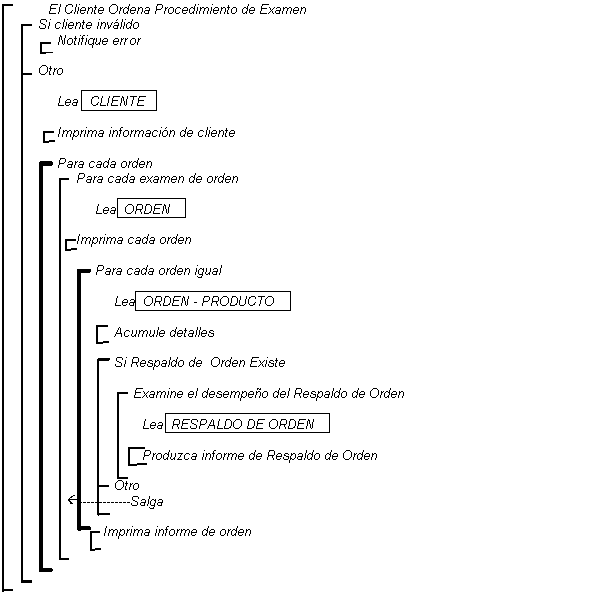
|  |  |
| --- | --- |
| Alternativa | **SI** *condición*  Bloque  **SI NO**  Bloque  **FIN SI** |
| Repetitiva | **MIENTRAS** *condición*  Bloque  **FIN MIENTRAS** |
| **REPETIR**  Bloque  **HASTA** condición |
| Secuencia | Está formada por un conjunto de sentencias (bloque) donde cada una puede ser o una acción sencilla o una estructura de las anteriores. |

* **Árboles de decisión:** es un modelo de una función discreta en la que se determina el valor de una variable y en función de éste se lleva a cabo una acción. Es una representación en forma de árbol que señala los valores de las variables y las acciones tomadas (que dependen del valor de la variable y de las acciones anteriores). Se suele utilizar cuando hay muchas condiciones.

* **Tablas de decisión:** es una de las herramientas más útiles para expresar sin ambigüedad un problema y su solución, ya que en ella se presentan todas las situaciones posibles que rodean el problema junto con las acciones que deberían ser tomadas para cada situación.



* **Diagramas de acción:** Es una técnica de especificación la cual utiliza niveles anidados de corchetes que representan la estructura lógica usada para transformar los datos de entrada en los datos de salida. En la fase de análisis, se preparan de forma general para especificar sólo las normas de transformación de los datos de entrada en datos de salida. Durante el diseño se detallan más esos diagramas.



* **Pre y post condiciones:** se centran más en la relación que deben tener las entradas y salidas del proceso que en su algoritmo. Por un lado se indican las condiciones las cuales se deben cumplir para que el proceso pueda comenzar (precondiciones), así como también las condiciones a lograr cuando el proceso ha concluido (postcondiciones).
  1. **Diccionario de datos.**

Es un conjunto de meta datos que contiene las características lógicas de los datos a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización. El diccionario de datos contiene las definiciones de todos aquellos términos mencionados en el Diagrama de Flujo de Datos, en una especificación del proceso y en el propio diccionario de datos.

* **Notación del diccionario de datos**

Los datos elementales (datos que no pueden ser divididos) Bloque de información básica.

1. **Un nombre:** para distinguir un dato de otro.
2. **Descripción:** indica lo que representa en el sistema.
3. **Alias:** porque un dato puede recibir varios nombres, dependiendo de quién usó este dato.
4. **Longitud:** porque es de importancia saber la cantidad de espacio necesario para cada dato.
5. **Valores de los datos:** porque en algunos de los procesos sólo son permitidos valores muy específicos para los datos. Si los valores de los datos están restringidos a un intervalo específico, esto debe estar en la entrada del diccionario.

Los datos compuestos (datos que pueden ser divididos) Grupo de datos relacionados entre sí.

|  |  |  |
| --- | --- | --- |
| Secuencial | + | Dirección = país+ población+ calle+ número |
| Selección | [ , ] | [D.N.I., Código-País + Número-Pasaporte] |
| Iteración | 1{ }n | Matrícula= Estudiante+1{Asignatura + Convocatoria +Nota}n |
| Opcionales | ( , ) | Domicilio de Envío+ (Domicilio Para Factura) |

|  |  |
| --- | --- |
| **=**  **+**  ( )  { }  n {}m  [| |]  “” | Significa ‘es definido como’ o ‘está hecho de’  Significa ‘y’  Significa que el ítem entre paréntesis es opcional  Significa cero o más de cualquier caso que esté dentro de las llaves o repetición  Significa entre n y m (inclusive)  Significa que uno de los atributos entre las barras está presente  Significa Incluye Literales (valor a utilizar) |

* **Simbología en un diccionario de datos.**

Ejemplo:

* = (igualdad)

Peso\_persona = unidad : kilo

Fecha\_nacimineto = unidad: días

* + y () (positivo y paréntesis)

Direc\_cliente = (direc\_entrega)+ direc\_facturación

* {}(llaves,repetición)

Pedido= nom\_cliente+ direc\_entrega+{artículo}

* [] (corchete, selección)

Sexo= [varón|Hembra]

* **Datos compuestos**

Listado de Datos Compuestos

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Estructura** |
| Cliente | Datos personales o de la empresa que cada cliente proporciona | **=** Nombre+ NIF+[Apellido1 + Apellido 2+Nombre ,Nombre Empresa]+ Dirección+ Teléfono |
| Inf-Cliente | Información del cliente que se genera a partir de los datos que éste proporciona y que se registra en el sistema | **=** Número-Cliente+ Cliente+ Importe-deudor+Fecha-incorporación+Tipo-Cliente+Descuento |

* **Datos elementales, creo que es simbología de diccionario de datos**

Listado de Datos Elementales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Alias** | **Tipo/long** | **Valores** |
| Número-Cliente | Número único y exclusivo que se asigna de forma secuencial a cada cliente cuando se da de alta | Num. | 6 dígitos | Entero |

* **Documentar Almacenes**

Almacenes

**Nombre del Almacén:** Clientes

**Descripción:** Información de cada una de las personas con las cuales la empresa mantiene una relación comercial

**Volumen:** (Medio) 5000

**Dato compuesto:** Inf-Cliente

* **Documentar de Flujo de Datos**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre** | **Id** | **Origen** | **Destino** | **Frecuencia** | **Compuesto** |
| Datos-Cliente | F-10 | E. Externa Cliente | Proceso: Diagrama de Contexto | Entero | Cliente |

* 1. **Sistemas modulares.**

Es uno de los métodos de diseño más flexibles y potentes para mejorar la productividad de un software, en el sistema modular, el problema es divido en módulos partes independientes), donde cada uno del los cuales ejecuta una única función o actividad y se codifican independientemente de otros módulos. Estos módulos individualmente se analizan, codifican y ponen a punto por separado.

De esta manera, en lugar de resolver una tarea compleja y tediosa, se resuelven otras más sencillas y a partir de ellas, llegar a la solución. Esta técnica se usa mucho en programación ya que programar, no es más que resolver problemas, y se le suele llamar diseño descendente, metodología del divide y vencerás o programación top-down.

Es evidente que si esta metodología nos lleva a tratar con sub-problemas. Entonces también tengamos la necesidad de poder crear y trabajar con subprogramas para resolverlos.

* 1. **El Proceso de Diseño de Software: Conceptos y Principios Fundamentales de Diseño.**

El diseño de sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema con suficientes detalles como para permitir su interpretación y realización física.

El principio fundamental para el diseño de software se basa en:

a.- transformar el modelo de dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el software.

b.- Diseñar los datos que se relacionan entre cada uno de los elementos estructurales del programa.

c.-Describir arquitectónicamente la relación entre cada uno de los elementos estructurales del programa.

d.- Diseñar las interfaces que permitan la comunicación del software consigo mismo, con los sistemas que operan junto con él y con los operadores y usuarios que lo emplean.

e.- Diseñar los procedimientos que implica transformar elementos estructurales de la arquitectura del programa.

**Ejemplo de Diagrama de Flujo de Datos.**

**Gestión de Mantenimientos de Aviones.**

Se debe obtener el análisis mediante un diagrama de flujo de datos de una empresa aeronáutica que desea mecanizar la gestión de mantenimiento de los aviones de acuerdo el siguiente procedimiento.

* El cliente realiza una petición de presupuesto de reparación de un avión, señalando las diferencias observadas.
* El cliente recibe un presupuesto que se realiza teniendo en cuenta los costos de los componentes. Dicho presupuesto se guarda.
* Si el cliente entrega el documento de aceptación del presupuesto, este es aceptado. Se actualiza el archivo de presupuesto y se hace llegar al avión a los hangares de la compañía emitiéndose un albarán de depósito y se comienza la reparación. Al equipo de reparación se le pasa una orden de trabajo.
* Una vez terminado el trabajo de reparación pasa una nota de trabajo inmediatamente después de recibir la nota se activa la emisión de una factura que es entregada al cliente.
* Un equipo de presupuesto se encarga de actualizar periódicamente los componentes.

**Nivel 0- DFD de Contexto.**

El DFD de contexto de este sistema tiene tres entidades externas, la entidad externa cliente identificada en los cuatro primeros puntos y que tiene cinco flujos de datos con el proceso Gestión Mantenimiento.

* El primero es Petición Presupuesto, identificado junto a la entidad externa en el punto 1. Donde dice: “el cliente realiza una petición de presupuesto de reparación de aviones señalando las deficiencias observadas”.
* El segundo es Aceptación Presupuesto, punto 2. “el cliente recibe un presupuesto”.
* En el punto 3 se observa dos flujos, uno de entrada. Aceptación Presupuesto (presupuesto inicial).Y otros hacia el cliente, Albarán Deposito.
* Y por último, el punto 4 se identifica el flujo de salida Factura.

Otra entidad externa es el flujo de Reparación, que recibe del sistema la información referente a una Orden de Trabajo y proporciona al mismo los detalles de una reparación a través de una Nota de Trabajo.

Y la tercera entidad externa es Equipo de Presupuesto, que proporciona el sistema la información al Precio de Componentes, siendo esta el único flujo que intercambia el sistema.

**Nota:**

Es importante comprobar que existe flujo tanto de entrada como de salida entre el proceso y las entidades externas. Si no el diagrama tendrá una solución errónea del sistema, ya que solo seria fuente o sumidero de información y no habrá realizado ninguna trasformación de la misma que es su fundamento.

EQUIPO REPARACION

CLIENTE

EQUIPO PRESUPUESTO

CLIENTE

NOTA DE TRABAJO

FACTURA

ALBARAN DEPÓSITO

ORDEN DE TRABAJO

PETICION PRESUPUESTO

ACEPTACION PRESUPUESTO

PRECIO COMPONENTE

GESTIONAR MANTENIMIENTO

**Nivel 1- DFD de Sistema.**

El DFD de sistema de este ejercicio presenta tres procesos principales, Gestionar Presupuesto (proceso 1), que incluye todo lo relacionado con la regeneración, emisión y aceptación de un presupuesto. Actualización de Componentes (proceso 2), cuya funcionalidad es permitir al Equipo de Presupuesto actualizar periódicamente el precio de los componentes, y Emitir Factura (proceso 3), que se encarga de emitir las facturas.

El proceso Emitir Factura (3), utiliza los datos en los flujos Presupuesto Actualizado y Nota de Trabajo, el flujo de salida es la información referente a la propia Factura.

Los almacenes utilizados por este sistema para guardar los datos que van a ser posteriormente utilizados, son el almacén Componentes y Presupuesto, en el primero se guardan la información referente a los componentes y sus precios, que será utilizada por el proceso 1. Gestionar Presupuesto cuando se tenga que elaborar un presupuesto y que es actualizar por el proceso 2. Actualización Componentes y en el segundo se guardan los datos refuentes a los presupuesto pedidos y aceptación para ser utilizados por el proceso 3. Emitir Factura cuando se tenga que generar una.

Es importante recordad que los flujos representados en los DFDs no se consideran activadoras de de programa, esto es, aunque cuando se recibe una Nota de Trabajo se genera una Factura, no se pretende indicar con las flechas que representa a la Nota de Trabajo, si no que por ahí va circulando una información que será utilizada por el proceso 3. Emitir Factura, y no que ese proceso comience su ejecución al llegar la información, aunque en la realidad así suceda.

FACTURA

ORDEN DE TRABAJO

PETICION PRESUPUESTO

ACEPTACION PRESUPUESTO

ALBARAN DEPÓSITO

GESTIONAR PRESUPUESTO

EMITIR FACTURA

PRESUPUESTO INICIAL

D1 COMPONENTES

NUEVO COMPONENTE

D1 PRESUPÚESTO

PRESUPUESTO ACTUALIZADO

PRECIO COMPONENTE

ACTUALIZARCIONCOMPONENTE

1

2

3

PRESUPUESTO

NOTA DE TRABAJO

COMPONENTE

**Nivel 2- DFD Extendido.**

El proceso 1 Gestionar Presupuesto, puede ser tratado con más detalle en un nuevo diagrama se puede descomponer con un mayor nivel de detalle en los procesos.

* Generar Presupuesto (1,1), que se encarga de automatizar todo el proceso generado hasta que el presupuesto es aceptado por el cliente , por el cliente, Trata con 4 flujos, el de Petición de Presupuesto, el que introduce en el proceso 1.1 los datos necesarios para generar dicho presupuesto extraídos del almacén Componentes, que aparecerá, por lo tanto, en este diagrama de salida con la información del Presupuesto Inicial que se entrega al cliente y el de salida que guarda en el almacén Presupuesto. La información relativa al Presupuesto Encargado
* Tratar Presupuesto (1,2), Automatizar la gestión de un presupuesto estregado con el flujo de entrada, Aceptación Presupuesto y tres flujos de salida. Una hacia el almacén Presupuestos, el flujo llamado Presupuesto Actualizado y de hacia el exterior del sistema, Orden de Trabajo Albarán Deposito.

TRATAR PRESUPUESTO

GESTIONAR PRESUPUESTO

1.1

1.2

PRESUPUESTO INICIAL

ALBARAN DEPÓSITO

ORDEN DE TRABAJO

ACEPTACION PRESUPUESTO

PETICION PRESUPUESTO

D1 COMPONENTES

COMPONENTE

D1 PRESUPÚESTO

PRESUPUESTO ACTUALIZADO

PRESUPUESTO ENCARGADO

**Conclusión**

A lo largo de la elaboración de este trabajo se pudo concluir que el análisis y diseño estructurado conllevan tanto al desarrollo de especificaciones para sistemas nuevos como a la definición de la relación existente entre los principales elementos del programa.

Este se sustenta de varios elementos que ayudan a lo nombrado anteriormente como lo son:

* Diagrama de flujos de datos, el cual posee cierta cantidad de elementos característicos.
* Descripción de procesos, los cuales sirven para describir los procesos a nivel más detallado.
* Diccionario de datos, el cual ayuda a la documentación de lo que vaya a hacer
* Sistemas modulares, esta metodología es uno de los métodos más flexibles y potentes para mejorar la productividad de un software llevando un problema grande a varios sub-problemas permitiendo trabajar de una manera más rápida y sencilla.