



# **Preparador Informática**

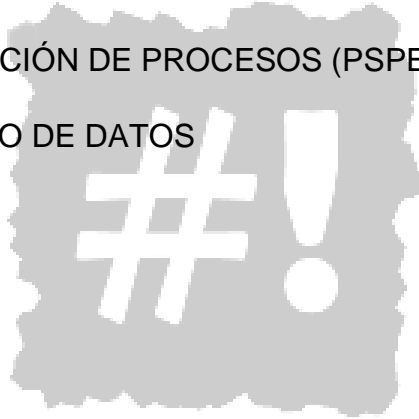
[www.preparadorinformatica.com](http://www.preparadorinformatica.com)

## **TEMA 49. INFORMÁTICA**

**ANÁLISIS DE SISTEMAS:  
MODELIZACIÓN DE TRATAMIENTOS.  
MODELO DE FLUJO DE DATOS Y  
CONTROL. TÉCNICAS DESCRIPTIVAS.  
DOCUMENTACIÓN**

## **TEMA 49 INF: ANÁLISIS DE SISTEMAS: MODELIZACIÓN DE TRATAMIENTOS. MODELO DE FLUJO DE DATOS Y CONTROL. TÉCNICAS DESCRIPTIVAS. DOCUMENTACIÓN**

1. INTRODUCCIÓN
2. ANÁLISIS DE SISTEMAS: MODELIZACIÓN DE TRATAMIENTOS
3. MODELO DE FLUJO DE DATOS
4. MODELO DE FLUJO DE CONTROL
5. TÉCNICAS DESCIRPTIVAS
  - 5.1. ESPECIFICACIÓN DE PROCESOS (PSPEC)
  - 5.2. DICCIONARIO DE DATOS
6. DOCUMENTACIÓN
7. CONCLUSIÓN
8. BIBLIOGRAFÍA



Preparador Informática

## 1. INTRODUCCIÓN

Con el incremento de la complejidad de los programas informáticos, su generalización y la competencia en la industria del software se hizo necesario desarrollar sistemas de calidad que potenciases aspectos como la reutilización, la eficiencia o la documentación del software desarrollado. En definitiva, aplicar criterios de ingeniería al desarrollo de software, utilizando metodologías, herramientas y criterios de desarrollo para mejorar los tiempos del desarrollo y la calidad de los resultados. Esto dio lugar a la “ingeniería del software”.

La **ingeniería de software** es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el **desarrollo de las aplicaciones informáticas (software)**.

La ingeniería de software incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema. La fase de análisis es una de las más importantes en la ingeniería del software, de ahí la importancia que tiene el tema actual. En este tema nos centraremos en esta fase de Análisis de sistemas donde vamos a ver cómo realizar la modelización de tratamientos, centrándonos en el modelo de flujo de datos y control, distintas técnicas descriptivas y conceptos relacionados con la documentación de este proceso.

## 2. ANÁLISIS DE SISTEMAS: MODELIZACIÓN DE TRATAMIENTOS.

Las primeras tareas que se llevan a cabo en cualquier proyecto, independientemente del ciclo de vida utilizado, son las referentes al análisis del sistema. El análisis del software se centra en tres aspectos:

- Identificar y representar los datos que manejar el sistema y sus relaciones.
- Modelar los tratamientos que se realizarán sobre esos datos.
- Analizar las distintas funcionalidades que debe implementar la aplicación.

Existen dos enfoques a la hora de llevar a cabo el análisis de un sistema:

- Análisis estructurado: está orientado a los procesos, ofreciendo una visión del sistema donde se consideran por separado los datos que maneja el sistema y los procesos que los transforman. El refinamiento del sistema



viene dado por la descomposición sucesiva de las distintas funcionalidades del sistema. Este proceso puede llevar a una implementación más directa del sistema, pero frecuentemente ésta es poco flexible y poco tolerante a cambios, un pequeño cambio de funcionalidad puede suponer una reestructuración masiva.

- Análisis orientado a objetos: Ofreciendo una visión del sistema basada en los objetos que lo componen y las interacciones que se establecen entre ellos. Este enfoque da lugar a la construcción de sistemas con alta cohesión y bajo acoplamiento, mucho más flexibles que los anteriores y más tolerantes a cambios, ya que cada funcionalidad se encapsula en el objeto que la ofrece.

En este tema nos centraremos en el análisis y modelado estructurado de los tratamientos de información que llevará a cabo el sistema. El análisis estructurado es la técnica que modela el flujo de la información, y el contenido de la misma. A través de notaciones, construye modelos que reflejan la información (control y datos) y el flujo que sigue a lo largo del sistema.

A medida que la información fluye por un sistema informático se transforma: un sistema acepta una serie de entradas heterogéneas que transformará para generar un resultado final. Estas transformaciones pueden ser de complejidades diferentes. Sea cual sea la condición de las entradas y salidas y la complejidad de las transformaciones que estas sufren siempre se podrá crear un modelo de flujo para ellos. Existen diferentes técnicas y diagramas que pueden ser empleados para modelar tanto la información que maneja el sistema como los flujos de datos existentes.

### **3. MODELO DE FLUJO DE DATOS**

Un diagrama de flujo de datos (DFD) traza el flujo de la información para cualquier proceso o sistema. Emplea símbolos definidos, como rectángulos, círculos y flechas, además de etiquetas de texto breves, para mostrar las entradas y salidas de datos, los puntos de almacenamiento y las rutas entre cada destino. Los diagramas de flujo de datos pueden variar desde simples panoramas de procesos incluso trazados a mano, hasta DFD muy detallados y con múltiples niveles que profundizan progresivamente en cómo se manejan los

datos. Se pueden usar para analizar un sistema existente o para modelar uno nuevo. De forma similar a todos los mejores diagramas y gráficos, un DFD puede con frecuencia "decir" visualmente cosas que serían difíciles de explicar en palabras y funcionan para audiencias tanto técnicas como no técnicas, desde desarrolladores hasta directores. Esa es la razón por la que los DFD siguen siendo tan populares después de todos estos años. Aunque funcionan muy bien para software y sistemas de flujo de datos, en la actualidad no se aplican tanto para visualizar software o sistemas interactivos, en tiempo real u orientados a bases de datos.

### Símbolos y notaciones usadas en los DFD

Los sistemas comunes de símbolos llevan el nombre de sus creadores:

- Yourdon-Coad
- Yourdon-DeMarco
- Gane-Sarson

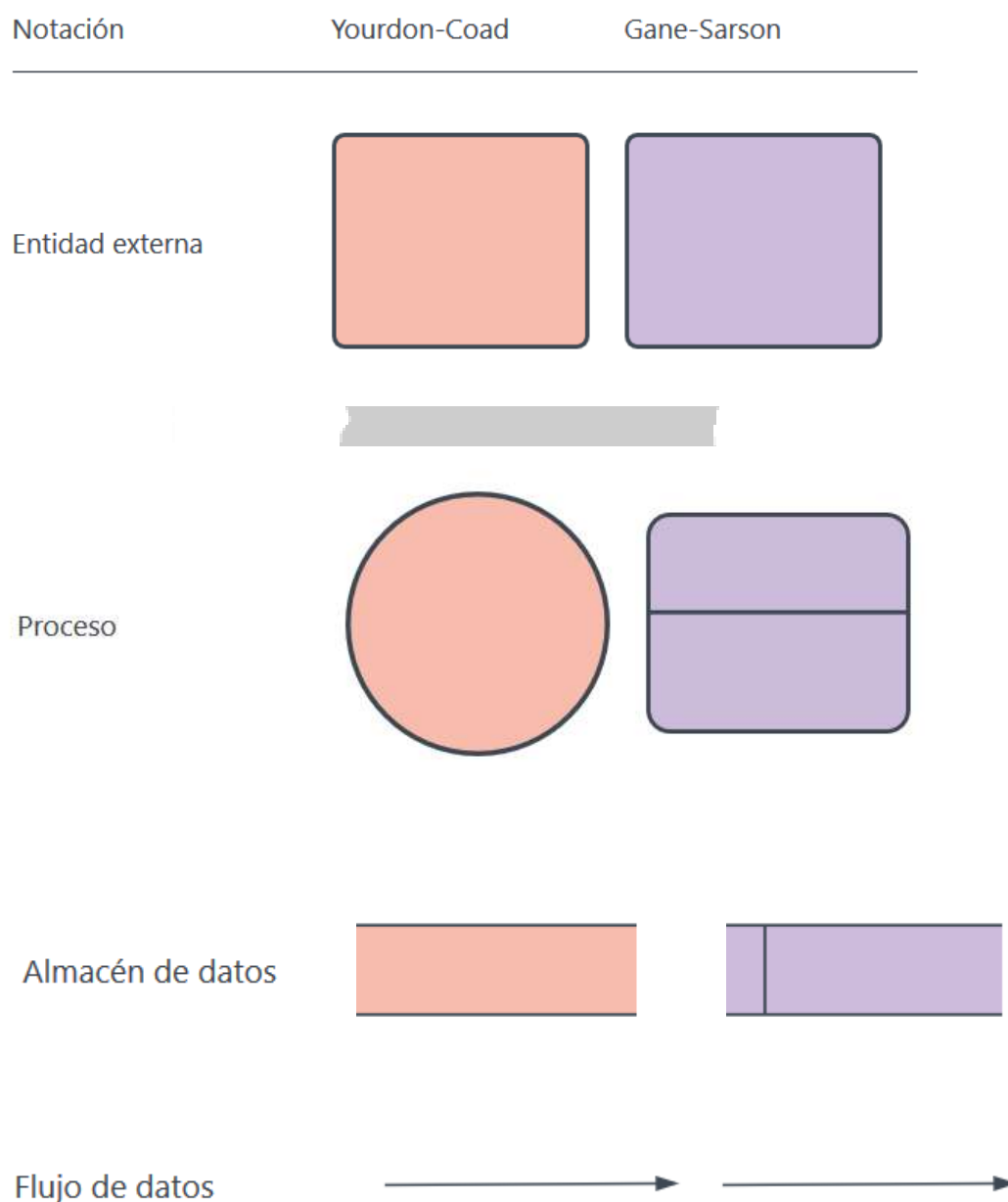
Una diferencia importante en sus símbolos es que Yourdon-Coad y Yourdon-DeMarco usan círculos para procesos, mientras que Gane y Sarson usan rectángulos redondeados, en ocasiones llamados "grageas" (rombos). Hay también otras variaciones de símbolos en uso, por lo que lo importante es ser claro y constante en las figuras y notaciones que se usen.

Usando las reglas o lineamientos para DFD de cualquier convención, los símbolos representan los cuatro componentes de los diagramas de flujo de datos:

1. **Entidad Externa:** Un sistema externo que envía o recibe datos, comunicándose con el sistema que se está diagramando. Son las fuentes y destinos de la información que entra o sale del sistema. Podría ser una organización o persona externas, un sistema de computadoras o un sistema de negocios. También se los conoce como terminadores, fuentes y receptores o actores. Generalmente se los dibuja en los bordes del diagrama.
2. **Proceso:** cualquier proceso que cambia los datos y produce un resultado. Podría realizar cálculos u ordenar datos basados en una lógica o dirigir el

flujo de datos en función de reglas de negocios. Se usa una etiqueta pequeña para describir el proceso, por ejemplo "Enviar pago".

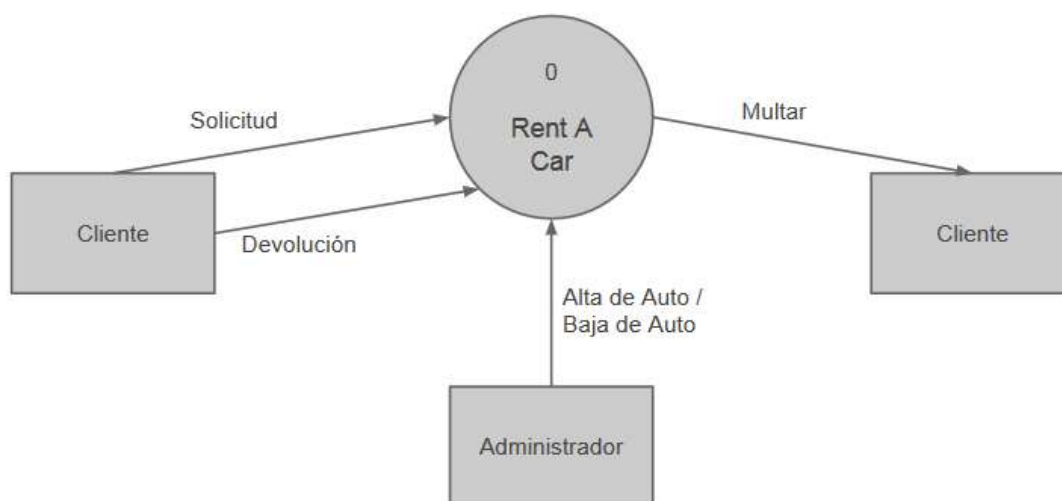
3. **Almacén de datos:** archivos o repositorios que conservan información para uso posterior, p. ej., una tabla de base de datos. Cada almacén de datos recibe una etiqueta simple, p. ej., "Pedidos".
4. **Flujo de datos:** la ruta que los datos toman entre las entidades externas, los procesos y los almacenes de datos. Representa la interfaz entre los otros componentes y se muestra con flechas, generalmente etiquetadas con un nombre de datos corto, como "Detalles de facturación".



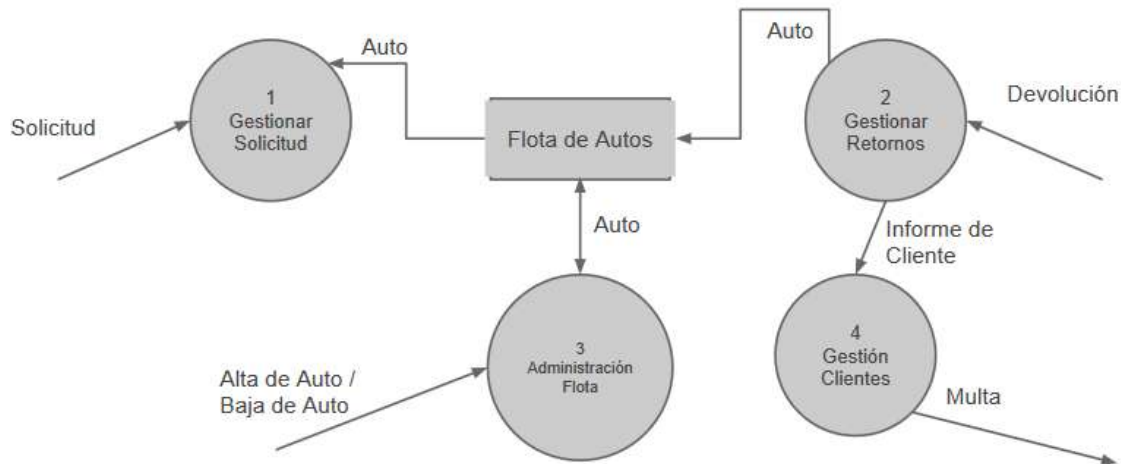
## Niveles y capas del DFD: de los diagramas de contexto al pseudocódigo

Un diagrama de flujo de datos puede profundizar progresivamente en más detalle por medio de niveles y capas, concentrándose en una pieza en particular. Los niveles de un DFD se numeran 0, 1 o 2 y en ocasiones llegan incluso hasta el Nivel 3 o más. El nivel necesario de detalle depende del alcance de lo que estás tratando de lograr.

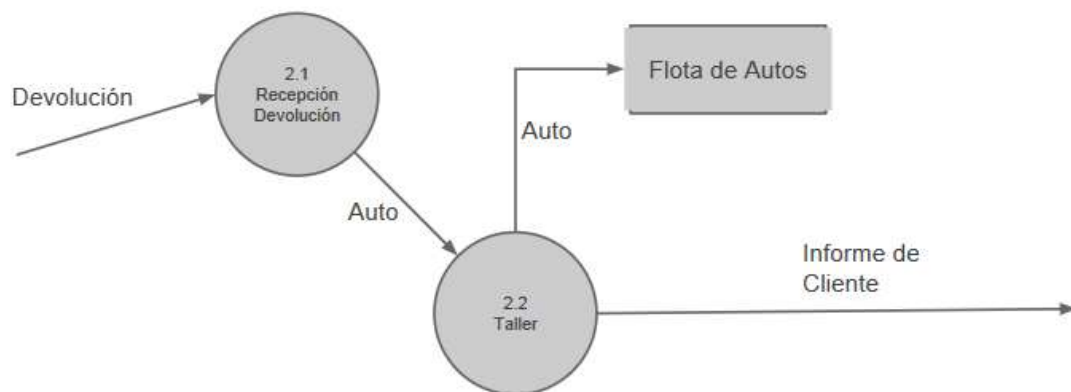
- Al Nivel 0 de un DFD también se lo llama Diagrama de contexto. Es un panorama básico de todo el sistema o proceso que se está analizando o modelando. Está diseñado para ser una vista rápida que muestra el sistema como un único proceso de nivel alto, con su relación con entidades externas. Debe ser entendido fácilmente por una amplia audiencia, incluidas partes interesadas, analistas de negocios, analistas de datos y desarrolladores.



- El Nivel 1 de un DFD brinda un desglose de piezas más detallado del diagrama a nivel de contexto. Se destacarán las principales funciones que el sistema lleva a cabo, a medida que se desglosa el proceso de alto nivel del diagrama de contexto en sus subprocesos.



- Luego el Nivel 2 del DFD profundiza un paso más hacia partes del Nivel 1. Puede requerir más texto para alcanzar el nivel necesario de detalle acerca del funcionamiento del sistema. Siguiendo con el ejemplo anterior veamos ahora el diagrama de nivel 2 del proceso “Gestionar Retornos”.



- Es posible el avance hacia los Niveles 3, 4 y más, pero ir más allá del Nivel 3 es poco usual. Hacerlo puede crear una complejidad que dificulte comunicar, comparar o modelar de forma efectiva.

Con el uso de capas en el DFD, los niveles en cascada se pueden anidar directamente en el diagrama, lo que proporciona un aspecto más ordenado con fácil acceso a profundizar en más detalle.

Al contar con un DFD con tanto detalle, los desarrolladores y diseñadores pueden usarlo para escribir pseudocódigo, que es una combinación de inglés y de lenguaje de codificación. El pseudocódigo facilita el desarrollo del código real.



#### 4. MODELO DE FLUJO DE CONTROL

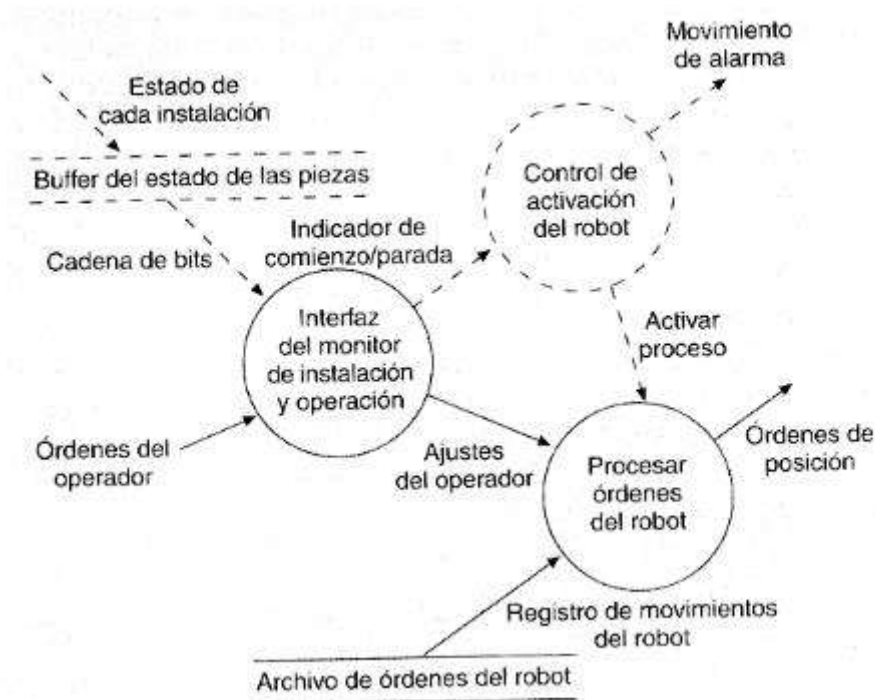
Existe una extensión para los Diagramas de Flujo de Datos denominados Diagramas de Flujo de Control (DFC).

Estos diagramas se utilizan cuando se requiere modelar un sistema en tiempo real como por ejemplo el software para controlar el tráfico de un sistema de vuelo o el software para monitorear el ritmo cardiaco.

En los DFC se ocupan dos símbolos adicionales a los DFD: las burbujas punteadas y las flechas punteadas.

Una burbuja punteada representa un proceso de control que tiene como propósito coordinar a otros procesos que no son de control. Lo único que puede entrar o salir de un proceso de control son flujos de control.

Las flechas punteadas representan flujos de control que tienen como propósito indicarle a otro proceso algún estado o situación que requiera conocer. Cabe destacar que son los únicos flujos que van a otro proceso. Las flechas de entrada al proceso de control indican que el proceso ha terminado o que se ha presentado alguna situación excepcional. Las flechas de salida del proceso de control indican otro proceso que debe iniciar.



## 5. TÉCNICAS DESCRIPTIVAS.

Los diagramas de flujo y de control permiten describir el funcionamiento del sistema, pero no describen el funcionamiento interno de los procesos que son activados como resultado de una transición, ni tampoco los datos que se intercambian entre los procesos. Para ello se utilizan dos tipos de representaciones: la especificación de procesos (PSPEC) y el diccionario de datos.

### 5.1. ESPECIFICACIÓN DE PROCESOS (PSPEC).

La especificación del proceso (PSPEC) se utiliza para describir todos los procesos del modelo de flujo que aparecen en un nivel. El contenido de la especificación del proceso incluye el texto narrativo, una descripción del lenguaje de diseño del programa, del algoritmo del proceso, ecuaciones matemáticas, tablas o diagramas de actividad UML. Por norma general cada burbuja del diagrama de flujo vendrá acompañada por su PSPEC que defina su comportamiento.

### 5.2. DICCIONARIO DE DATOS

Un diccionario de datos incluirá una lista organizada de todos los elementos de datos manejados por el sistema, que se irá refinando a lo largo de todas las fases de diseño del sistema. Las definiciones deben ser rigurosas y precisas, ya que debe facilitar tanto a los diseñadores como a los desarrolladores la comprensión de cada uno de los datos manejados por el sistema.

El diccionario de datos contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenidos y organización, así como los procesos donde se emplea los datos y los sitios donde se necesita el acceso inmediato a la información. Servirán para identificar los requerimientos de las bases de datos durante el diseño del sistema.

## 6. DOCUMENTACIÓN

La documentación de software es una etapa transversal a todas las demás etapas del diseño: se iniciará en los primeros pasos del ciclo de vida y continuará hasta la entrega final del producto.

Los documentos creados en esta fase es conveniente que se lleven a cabo con la colaboración del cliente y que sean aprobados por éste antes de proceder al desarrollo del proyecto ya que el objetivo fundamental de esta fase es fundamentalmente acordar con el cliente el alcance del proyecto.

Cada fase, proyecto, infraestructura y organización tendrá sus propios procedimientos de documentación, pero hay una serie de manuales y documentos que son comunes a las diferentes etapas de los ciclos de vida.

- Enunciado del alcance: Descripción de alto nivel de las funciones y objetivos del proyecto.
- Documento de requisitos: Aquí se identificarán todos los requisitos que debe cumplir la aplicación y permitirá realizar un seguimiento de los mismos para asegurar que todos ellos se consideren y se cumplan.
- Documento funcional: En este documento se incluirán diversos diagramas creados en la fase de análisis como diagrama de flujo de datos, de casos de uso, etc.
- Modelo conceptual de datos del sistema: Normalmente se realiza con el modelo entidad relación o el modelo entidad relación extendido.
- Plan de riesgos: Incluirá una relación de los riesgos junto con los análisis cualitativos y cuantitativos de los riesgos identificados, y un plan de contingencia para los riesgos más relevantes.
- Diccionario de datos: El diccionario de datos es la “base de datos” del sistema: en él se registrará toda la información manejada por el sistema junto con su significado, relaciones, restricciones y cualquier otra información relevante acerca de cada uno de los datos. Este diccionario de datos es un documento que se irá refinando a lo largo del proceso de desarrollo del software

## 7. CONCLUSIÓN

Una de las fases más importantes en la ingeniería del software es el análisis. Si esta fase se realiza adecuadamente tenemos grandes opciones de que nuestro software cumpla correctamente con los requerimientos del cliente y por lo tanto tengamos un software exitoso.

En esto radica la gran importancia de este tema ya que es imprescindible dominar y realizar adecuadamente la modelización de tratamientos, realizar adecuadamente el modelo de flujo de datos y el modelo de flujo de control, ser capaz de utilizar las técnicas descriptivas más importantes para ello y documentar adecuadamente la fase para que en fases posteriores podamos avanzar en el ciclo de desarrollo del software de manera adecuada.

## 8. BIBLIOGRAFÍA

- Pressman, Roger. ***Ingeniería del software: un enfoque práctico.*** Editorial McGraw-Hill
- Piattini, Mario G. y otros. ***Análisis y diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software.*** Editorial Ra-Ma.
- Barlow y Bentley, ***Análisis y diseño de sistemas de información.*** Editorial McGraw-Hill
- [https://www.tutorialspoint.com/es/software\\_engineering/software\\_development\\_life\\_cycle.htm](https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm)
- <https://cs.uns.edu.ar/~mll/modelos/downloads/Teoria/2018-Modelos-de-Software-Clase-6-Flujo-de-Datos.pdf>