

## **Resumen**

<b>Diagrama de Flujo de Datos</b>	<b>2</b>
Análisis Estructurado De Sistemas	2
Diagrama de Flujo de Datos (DFD)	2
Diagrama de Contexto (DC):	2
Tabla de Eventos (TE):	2
Diagrama de Sistema (Nivel 1)	3
Otros Niveles	4
Diccionario de Datos (DD)	5
Definición de Procesos	5
<b>Diagrama Entidad-Relación</b>	<b>5</b>
Componentes de un DER	5
Entidades	5
Atributos	5
Clave	5
Relaciones	6
<b>Casos de Uso</b>	<b>6</b>
Casos de Uso y UML	6
Actores	6
Casos de Uso	7
Relaciones entre Actores y Casos de Uso	7
Relaciones entre Casos de Uso	7
Relación de Extensión	7
Relación de Inclusión	7
Documentación de Casos de Uso	8
Flujo de Eventos	8
Precondiciones y Postcondiciones	8
Precondiciones	8
Postcondiciones	8
Escenario de un Caso de Uso	9
<b>Requerimientos Funcionales</b>	<b>9</b>
Definición	9
Requerimientos en la Metodología	9
Importancia de los requerimientos para el éxito de un proyecto	9
Tipos de Requerimientos	10
Funcionales	10

# Diagrama de Flujo de Datos

## Análisis Estructurado De Sistemas

El Análisis Estructurado es una metodología que permite establecer **que** es lo que el sistema hace o deberá hacer, sin importar **cómo** lo hace o hará. Determina la idea abstracta que hay siempre detrás de cualquier situación. Permite definir de esa manera, un **modelo lógico** del problema en estudio. Debido a la representación abstracta de la realidad a la cual no estamos acostumbrados, la modificación de la propia mentalidad es un requisito importante para utilizar esta metodología.

## Diagrama de Flujo de Datos (DFD)

El Diagrama de Flujos de Datos es una técnica de análisis estructurado mediante la cual el profesional de sistemas puede **reunir en una representación gráfica, los procesos de datos a lo largo de la organización**. El DFD permite modelar todo tipo de sistemas, concentrándose en las funciones que realiza y la información que intercambia con el contexto. Cuando se realiza un DFD se siguen distintos pasos que van desde un panorama básico del sistema a modelar hasta el nivel de detalle que se necesite para estudiarlo. Estos pasos involucran las siguientes herramientas:

### **Diagrama de Contexto (DC):**

Mediante el DC es posible representar el sistema en estudio de forma tal que de un vistazo se pueda apreciar el sistema y su interacción con el medio conocido. El DC es una instancia inicial de un DFD, es posible hacer un DC y nunca pasar a la segunda instancia (Tabla de Eventos). También se lo conoce como nivel 0. El DC está formado por tres elementos:

- **Proceso:** Función que representa el sistema modelado. Se identifica con el número 0 y el nombre del proceso indica que hace el sistema en forma general.
- **Entidad Externa:** Es todo lo que se relaciona con el sistema en estudio, pero que no opera en el sistema, solo **brinda información** o **recibe la información** de éste. Se le coloca el nombre que representa y una letra que la identifica.
- **Flujo de Datos:** Representan el movimiento de datos o información desde el sistema hacia las entidades externas o viceversa. El nombre indica la información que transportan, **no deben ponerse nombres de formularios físicos ni las palabras DATOS o INFORMACIÓN (Tampoco abreviaciones de las mismas)**.

### **Tabla de Eventos (TE):**

Un **evento** es un suceso que provoca que el sistema tenga una reacción y tenga que emitir alguna respuesta. El objetivo de la Tabla de Eventos es **identificar y enumerar los sucesos que hacen reaccionar al sistema de alguna manera**, es decir, que el sistema ante determinada acción externa o interna genere respuestas para el sistema y/o las entidades externas. La tabla de eventos está formada por 5 columnas:

- **Tipo:** Este campo está relacionado con quién o qué inicia el evento, y puede ser tipificado como Externo, Temporal o Desconocido. Es de tipo **Externo** si quien da inicio al evento es un Flujo de Datos proveniente de una Entidad Externa. Es de tipo

**Temporal** si comienza por acción del tiempo. Si se desconoce su activación, lo indicamos como **Desconocido**.

- **Descripción:** Es una breve descripción del evento.
- **Estímulo:** En el caso de un evento externo se indica el nombre del flujo de datos que inicia el evento, el cual debe coincidir con el flujo de datos del Diagrama de Contexto. Debe existir **un solo flujo activador**. En el caso de un evento temporal se coloca una raya ya que el estímulo es el **tiempo**. Si el evento es desconocido se coloca un **signo de pregunta** indicando que se debe consultar quien inicia este evento.
- **Respuesta:** Se indican todos los flujos de datos que genera como salida el evento al terminar. Deben coincidir con flujos de datos de salida del Diagrama de contexto.
- **Proceso Asociado:** Es el nombre de la función o proceso a la que asociamos el evento, se debe colocar un verbo en infinitivo y coincidirá luego con todos los procesos de nivel 1.

## Diagrama de Sistema (Nivel 1)

En el Nivel de Sistema se busca documentar con mayor grado de detalle el sistema. Los flujos de entrada y salida del nivel 0 se mantienen en todos los diagramas siguientes. Sin embargo, el proceso 0 se reemplaza por los procesos asociados en la tabla de eventos. En el nivel 1 van a aparecer nuevos flujos de datos, ya sea los de entrada que en el nivel de contexto se desconocía la procedencia o los de salida que generaba el sistema, pero se desconocía la entidad externa o proceso que los utilizará. Al ampliar el proceso de nivel 0 para representar subprocesos, el analista comienza a completar los detalles de los movimientos de los datos; y se incorpora un nuevo elemento: **Las Demoras**.

Una **demora** indica un depósito de datos, el cual permite la adición y acceso de los datos, es información que tiene diferencia de tiempos entre su generación y su utilización. Su identificador se compone de la letra D seguida por un número. A continuación, lleva un nombre que representa la información que contiene. Los nombres no pueden coincidir con ningún otro elemento del diagrama.

También aparecen en este nivel los flujos de datos que vinculan procesos. Todos los procesos en el nivel 1 deben estar vinculados a través de demoras. Por tal motivo la cantidad de flujos de datos del nivel 1 siempre va a ser mayor a la cantidad de flujos de datos del nivel 0. Con respecto a los procesos, se identificarán con un número, comenzando por 1 ya que el 0 está reservado para el Diagrama de Contexto, y el nombre que será el mismo de la Tabla de Eventos. **Debe indicarse cómo se activa cada uno de ellos**. El **flujo de datos estímulo o activador** debe ser marcado de forma tal de diferenciarlo del resto (Por ejemplo, doble flecha).

La disposición del dibujo debe ser de manera tal que en el centro se dibujen los procesos y demoras del sistema y por fuera quedan las entidades externas. Debo poderse **delimitar con línea punteada** el contexto del sistema.

Las demoras **pertenecen al Sistema**, y cada proceso demorará los datos hasta el momento en que los necesite. Los flujos de salida que van hacia entidades externas **nunca se**

**demoran** ya que si la entidad externa los necesita demorar es ella la que tendrá que hacerlo en su Sistema.

La manera correcta de vincular procesos es que uno **genere** información para la demora y el otro la **utilice**, no es necesario que todos los procesos se vinculan entre sí, pero si levanto imaginariamente un proceso el resto deben estar enganchados a ese a través de demoras.

Algunos **consejos** a tener en cuenta son:

- Los procesos deben transformar información. Si no detectamos la transformación y la función solamente consiste en transmitir una información demorada, entonces **NO** es un proceso.
- Todos los procesos **deben ser activados**, o por un flujo activador (doble línea) o por el tiempo.
- Cada evento detectado en la tabla de eventos es un proceso descrito en el Nivel 1.
- Todos los procesos deben relacionarse **con al menos uno** de los demás procesos a través de una o varias demoras.
- Todos los procesos deben tener al menos una **entrada** y al menos una **salida** que represente la transformación de la información.
- Tanto el DC, la TE y el Nivel 1 deben tener **consistencia de información**, es decir, respetar los mismos nombres descritos para cada flujo, entidad o proceso, sino es erróneo.
- Todas las entradas en un Nivel 1 **deben estar demoradas**, excepto el flujo activador de un evento externo, y un flujo de tipo Consulta/Respuesta.
- **Ninguna** salida hacia Entidades Externas debe ir demorada.
- Si una Entidad se repite dentro del gráfico, (por una cuestión de claridad), entonces debe marcarse con una **línea en la esquina superior** en todas las repeticiones correspondientes.
- Si una Demora se repite dentro del gráfico, (por una cuestión de claridad), entonces debe marcarse con una **doble línea** que separe el número y el nombre, en todas las repeticiones correspondientes.
- Todas las demoras **deben tener al menos una entrada y una salida**, si no existe tal situación debe colocarse un flujo de entrada o salida (según corresponda) con un signo de interrogación indicando que se desconoce la entrada/salida de esa demora. Si la demora se repite no hace falta volver a indicar la entrada o la salida.

## Otros Niveles

Cada proceso del Diagrama de Sistemas puede a su vez ser explotado para crear un diagrama más detallado. No es necesario explotar todos los procesos, solo se realizará esta acción en aquellos cuyo nivel de complejidad así lo ameriten. En el nivel 2, la vinculación entre los procesos se realiza a través de un Flujo de Datos que va directo de proceso a proceso. Los procesos que aparecen en el Nivel 2 son numerados usando el número del proceso que se había utilizado en el Diagrama de Sistema y se le agrega un punto decimal y un número único para cada proceso.

## Diccionario de Datos (DD)

El **diccionario de datos** es un listado organizado de todos los datos que pertenecen a un sistema. El objetivo de un diccionario de datos es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades. El DD recopila y coordina términos de datos específicos y confirma lo que cada término significa para las diferentes personas de la organización. Los elementos que definimos en un DD son: **Demoras** y **Flujos de Datos** (Distinguiendo los que pasan por una demora y los que no).

## Definición de Procesos

La definición de procesos, es una herramienta de modelado de sistemas, que **permite definir qué sucede en los procesos** o funciones de un sistema. El objetivo es definir qué debe hacerse para transformar ciertas entradas en ciertas salidas. No hay una única forma de realizar la definición de procesos; existen múltiples herramientas que facilitan esta tarea. Algunas herramientas utilizadas para generar definiciones de procesos son:

- **Lenguaje estructurado:** Se emplea un lenguaje natural limitado en palabras y construcciones, dándole más precisión y claridad, evitando ambigüedades. Definen un algoritmo.
- **Tablas de Decisiones.**
- **Lenguaje narrativo.**
- **Diagramas de flujos.**

## Diagrama Entidad-Relación

Este diagrama permite **modelar la información** que utiliza el sistema y **la relación** que existe entre la misma. Esta herramienta se utiliza en el modelo de datos conceptual, es el primer paso en el proceso de desarrollo de la base de datos.

## Componentes de un DER

### Entidades

Es una cosa significativa acerca de la cual **se necesita conocer y mantener información**. Se representan con un rectángulo. El nombre es un sustantivo en singular y debe ser único.

### Atributos

Son una pieza específica de información de la cual **se necesita tener conocimiento**. Los atributos describen a las entidades. Son las características de las entidades.

### Clave

Uno o más atributos que **permiten identificar una y sola una ocurrencia en una entidad**. La clave debe ser **única** y **mínima**. Se indica con una línea debajo del atributo o atributos que forman parte de la clave.

## Relaciones

Es una **asociación bidireccional entre dos entidades**. Cada dirección de la relación tiene un verbo que describe cómo es la relación de la entidad con respecto a la otra y un identificador. Existen 2 tipos de relaciones:

- **Asociación:** Cuando el atributo heredado de la relación **no forma parte** del identificador en la entidad destino.
- **Dependencia:** Cuando el atributo heredado de la relación forma parte del identificador en la entidad destino.

Las características de una relación son:

- **Cardinalidad:** Indica el **número máximo de instancias** o elementos de una entidad que pueden asociarse a un elemento de la otra entidad. Puede ser:
  - 1 a 1
  - 1 a Muchos
  - Muchos a Muchos
- **Modalidad:** Indica el **número mínimo de elementos** que participan en una relación. Cada dirección de la relación tiene una modalidad. Esta puede ser:
  - Opcional: Se marca con un 0 y significa que puede no haber ninguna ocurrencia en una entidad para un elemento de la otra.
  - Obligatoria: Se marca con L y significa que debe haber por lo menos 1 ocurrencia en una entidad para un elemento de la otra.

Existe una relación especial de especialización llamada **Relación Tipo/Subtipo**: La entidad se relaciona consigo misma. Cada subtipo hereda los atributos de la entidad tipo pero posee propios. **La clave es la misma**.

## Casos de Uso

### Casos de Uso y UML

UML (Unified Modelling Language), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. UML está relacionado con el paradigma orientado a objetos. Por otro lado, los casos de uso son una **técnica "de redacción"** del conjunto de secuencias de acciones que ejecuta un sistema. Los casos de uso son **una técnica independiente del paradigma que se adopte**, es decir se podrán utilizar tanto en el paradigma estructurado como en el paradigma orientado a objeto. En base a lo antedicho, se puede concluir en que los casos de uso no son una herramienta propia del UML pero dada la utilidad e importancia de los casos de uso, UML los incluye dentro del conjunto de herramientas de modelado, definiendo un estándar de representación gráfica.

### Actores

Un actor **representa un conjunto de roles** que los usuarios de los casos de uso juegan al interactuar con ellos. Este rol o roles es llevado a cabo por una persona, un dispositivo de hardware u otro sistema. La misma persona física puede interpretar varios papeles como actores distintos. El nombre del actor **describe el papel desempeñado**. Los actores son

**externos al sistema.** Es por ello, que al identificar a los actores se está delimitando el sistema.

## Casos de Uso

Son una **descripción de un conjunto de secuencias de acciones** (interacciones con elementos externos al sistema), que ejecuta un sistema **para obtener un resultado que agregue valor**. Un caso de uso es iniciado por un **único elemento externo** al sistema (Actor). A partir de ese momento, el sistema intercambia datos con el entorno. El nombre siempre está expresado **desde el punto de vista del usuario** y no desde el punto de vista del sistema. Se documentan con **texto informal**. Describen tanto lo que hace el usuario como lo que hace el sistema cuando interactúa con él.

## Relaciones entre Actores y Casos de Uso

Los actores pueden conectarse a los casos de uso sólo a través de asociaciones. Una **relación de asociación** indica una relación entre un actor y un caso de uso y la posibilidad que tienen éstos de comunicarse, es decir, enviar y recibir mensajes.

## Relaciones entre Casos de Uso

### **Relación de Extensión**

Una relación de **extensión** (extend) entre casos de uso, significa que uno de ellos **incorpora implícitamente el comportamiento** del otro caso de uso. Se utiliza cuando se desea indicar que mientras se está ejecutando un caso de uso, **se puede utilizar opcionalmente otro caso de uso**, en algún punto de la ejecución del primero. Las características de las extensiones son:

- Representan una parte de la funcionalidad del caso que no siempre ocurre.
- Son un caso de uso en sí mismas.
- No necesariamente provienen de un error o excepción.

### **Relación de Inclusión**

Una relación de **inclusión** (include), entre casos de uso, significa que un caso de uso **incorpora siempre el comportamiento de otro caso de uso**. La relación de inclusión se usa para **evitar describir el mismo flujo de eventos repetidas veces**, poniendo el comportamiento común en un caso de uso aparte (que será incluido por un caso de uso base). También se utiliza cuando se descompone un caso de uso largo (o complejo) en subunidades para **mejorar la comprensión**. Las características de las inclusiones son:

- Aparecen como funcionalidad común, luego de haber especificado varios casos de uso.
- Los casos usados son casos de uso en sí mismos.
- El caso es usado siempre que el caso que lo usa es ejecutado. Esto marca la diferencia con las extensiones, que son opcionales.

La principal **diferencia** que existe entre las relaciones Include y Extend, es que la relación extend cuenta con **una condición a ser evaluada**. En cambio, la relación include se define como **la inclusión de un Caso de Uso en otro**, pero no precisa de una condición.

## Documentación de Casos de Uso

Una vez que se identifican los casos de uso, se comienzan a documentar sus pasos. Este documento se crea para cada caso de uso, detallando lo que el sistema debe proporcionar al actor cuando el caso de uso es ejecutado. La documentación de casos de uso podría consistir en lo siguiente:

- Nombre.
- Descripción.
- Describir cómo comienza el caso de uso y cómo termina.
- Realizar un flujo normal de eventos.
- Realizar un flujo alterno de eventos.
- Detallar las excepciones al flujo de eventos.
- Precondiciones.
- Postcondiciones.

## Flujo de Eventos

Un caso de uso está compuesto por un flujo principal y uno o más flujos alternativos:

- **Flujo Principal:** Es la ejecución del curso normal del caso de uso.
- **Flujo Alternativo:** Durante la ejecución de un caso de uso, suelen aparecer errores o excepciones al curso principal. Esas desviaciones del curso normal del caso de uso se llaman alternativas y representan un error o excepción en el curso normal del caso de uso.

## Precondiciones y Postcondiciones

### **Precondiciones**

Las precondiciones, **reflejan el estado en el que debe estar el sistema y su entorno para que pueda comenzar la ejecución del caso de uso**. Las precondiciones establecen lo que **debe cumplirse antes de comenzar** un caso de uso. Estas no se prueban en el caso de uso, sino que **se asumen como verdaderas**, generalmente, implica que un caso de uso anterior se haya completado exitosamente. Las precondiciones indican suposiciones importantes que deben cumplirse.

### **Postcondiciones**

Las postcondiciones, reflejan **el estado en el que queda el sistema y su entorno luego de la ejecución del caso de uso**. Establecen qué debe cumplirse cuando el caso de uso se completa con éxito.



## Escenario de un Caso de Uso

Un escenario es una **instancia** de un caso de uso, es decir, un camino concreto que puede tomar un caso de uso. Los escenarios **no contienen condiciones**, ya que describen una de las posibles instancias del caso de uso. Todos los escenarios de un caso de uso comienzan igual, pero pueden terminar de muchas maneras diferentes. No se deben mostrar sólo instancias (escenarios) exitosas del caso de uso, también se deben mostrar algunas en las que falla. Cada uno de los escenarios definidos, **van a servir como casos de prueba** a la hora de realizar testing (test funcional) del sistema. El desarrollo de un caso de uso **es un conjunto de todos los escenarios posibles que puede presentar** el caso de uso.

## Requerimientos Funcionales

### Definición

Un **requerimiento (o requisito)** es una **necesidad** o solicitud cuyo objetivo es **resolver un problema**. Los requisitos son condiciones, características o capacidades que debe tener un sistema (Requerimientos del Sistema) para satisfacer las necesidades del usuario (Requerimientos del Usuario).

### Requerimientos en la Metodología

Teniendo en cuenta los pasos de la metodología de sistemas, definimos que:

- En la etapa de **RELEVAMIENTO** se relevan, capturan, escuchan, observan, detectan y documentan las necesidades del usuario (requerimientos del usuario).
- En la etapa de **ANÁLISIS DE REQUISITOS** se definirán los Requerimientos (requerimientos del sistema) que deberá tener el sistema para cumplir con todas esas necesidades (requerimientos del usuario).

### Importancia de los requerimientos para el éxito de un proyecto

Estadísticamente, la mitad de los proyectos fallan por causas que directamente o indirectamente están motivadas por malos procesos de ingeniería de requerimientos. En la mayoría de los proyectos de software, se comienza a programar demasiado pronto, concentrando el esfuerzo en producir código para obtener rápidos resultados. La principal consecuencia de los problemas asociados con los requerimientos es el rehacer algo que está hecho: **Retrabajo**.

Las **características** de un Requerimiento correctamente definido son:

- **Completo:** Contiene toda la información necesaria y no necesita ser expandido en otro ni dividido.
- **No ambiguo:** Debe tener una y sólo una interpretación.
- **Verificable:** Debe ser demostrado o probado su cumplimiento.
- **Consistente:** No deberá contradecir otros requerimientos.

Las ventajas de los **Requerimientos** correctos son:

- Identificación de riesgos de manera temprana. Atacar más rápido al problema.
- Análisis y Desarrollo más claro y rápido.
- Bases para un buen diseño.
- Definición clara de casos de prueba para realizar un Testing claro, completo y rápido.

## Tipos de Requerimientos

### Funcionales

Los **Requerimientos Funcionales** describen las funcionalidades del sistema, es decir **lo que el sistema debe hacer**, su comportamiento específico. Describen las transformaciones que el sistema realiza **sobre las entradas para producir salidas**. Deben estar redactados de tal forma que sean comprensibles para usuarios sin conocimientos técnicos avanzados. Definen qué debe hacer un sistema.

### No Funcionales

Los **Requerimientos No Funcionales** son los atributos o características que definen el cómo el sistema realizará el trabajo. Pueden considerarse como las restricciones planteadas al sistema respecto a cómo los requerimientos funcionales son implementados. Definen **cómo** debe ser un sistema. Se clasifican en:

- **De Producto o Calidad:** Limites o restricciones sobre el comportamiento del sistema.
  - Mantenibilidad: Significa que puede cambiar sin generar un gran impacto.
  - Usabilidad: Amigable para el usuario, respecto a la navegabilidad de las pantallas.
  - Performance: Eficiencia esperada, es decir buen desempeño del frente a la demanda esperada.
  - Disponibilidad: La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días X 24 horas.
  - Seguridad: La solución debe reflejar patrones de seguridad teniendo en cuenta la alta sensibilidad de la información que maneja de acuerdo a las especificaciones funcionales dadas y a las políticas, normas y estándares de seguridad requeridas
- **Organizacionales:** Se derivan de las políticas y procedimientos de la organización como por ejemplo estándares de procesos o requerimientos de implementación:
  - De entorno u organizacionales: Procedimientos operativos que describen cómo será usado el sistema dentro del contexto de la organización.
  - De diseño, desarrollo e implementación: Lenguaje de programación a usar, estándares de codificación, patrones (y antipatrones) de diseño y programación, herramientas para gestionar el desarrollo de software, entorno de desarrollo de software (ambiente de desarrollo), entorno de pruebas de software (ambiente de pruebas), entre otros aspectos.
- **Externos:**
  - Regulatorios: Leyes y reglamentos que establecen que debe hacer el sistema y cómo debe hacerlo para cumplirlas

- Éticos: Requerimientos que aseguran que el sistema será aceptable para el usuario, público en general y se adapta a las costumbres de la sociedad en la que se desenvuelve o a la que presta servicios.
- Legislativos: Características que debe cumplir el sistema para cumplir con la ley, por ejemplo en el área de contabilidad (normas contables y estándares financieros), requerimientos de seguridad industrial (para sistemas críticos), entre otros aspectos.