Universidad Nacional de la Matanza

Departamento:
Ingeniería e Investigaciones Tecnológicas

Cátedra:

Fundamentos de TIC's

(Tecnologías de la Información y la Comunicación)

e-mail: fundamentos_tics@unlam.edu.ar

JEFE DE CÁTEDRA:
Mg. Daniel A. Giulianelli

UNIDAD NRO. 6

INTRODUCCIÓN A LOS SISTEMAS DE INFORMACIÓN

COLABORACIÓN:
DOCENTES DE LA CÁTEDRA

CICLO LECTIVO:
2009

INTRODUCCIÓN A SISTEMAS DE INFORMACIÓN

INDICE

INTRODUCCIÓN A SISTEMAS DE INFORMACIÓN

INDICE

1. CONCEPTOS GENERALES

El termino "**sistema**" es de uso común. Se habla de sistemas educativos, de sistemas de computación, de sistemas solares, de sistemas de teología, y de muchos otros. Los conceptos de sistemas proveen una infraestructura útil para la descripción y comprensión de muchos fenómenos organizacionales incluyendo las características de los sistemas de información.

Los sistemas pueden ser abstractos o físicos. Un *sistema abstracto,* es una disposición de manera ordenada de las ideas interdependientes o artefactos. Por ejemplo, un sistema de teología es una organización de ideas interdependientes acerca de Dios y las relaciones de los hombres con Dios.

Los *sistemas físicos* serán los que consideraremos de aquí en más simplemente como sistemas.

1.1. DEFINICIÓN DE UN SISTEMA

Un sistema es una colección de componentes interrelacionados que trabajan conjuntamente para cumplir algún objetivo.

La definición de sistemas comprende un amplio rango de sistemas. Por ejemplo, un sistema tan simple como un bolígrafo incluye tres o cuatro componentes de hardware. En contraste, un sistema de control de tráfico aéreo está compuesto de cientos de componentes de software y hardware, además de los usuarios humanos que toman decisiones basadas en la información del sistema.

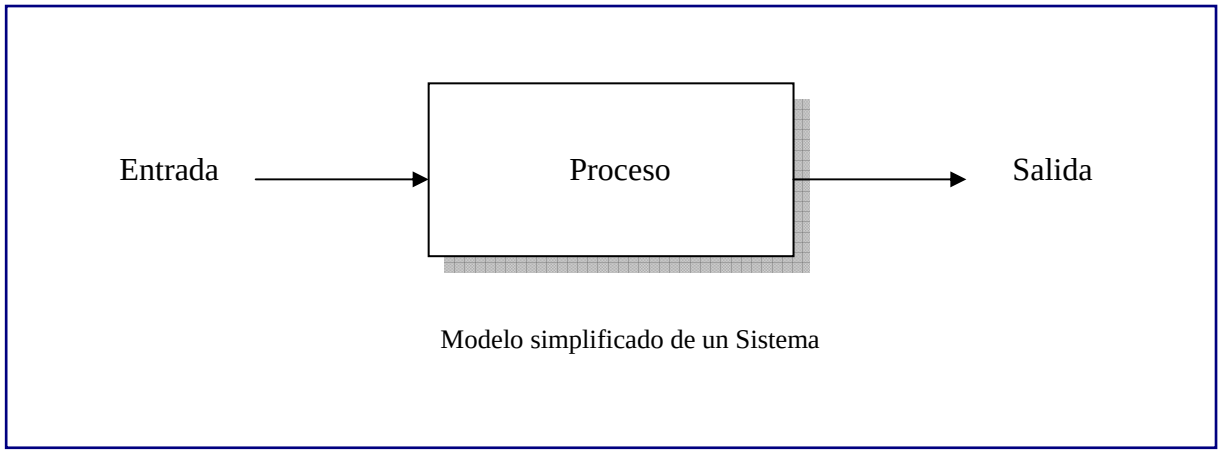
Un sistema físico puede ser mejor definido por medio de ejemplos:

Sistema circulatorio	El corazón y los vasos sanguíneos que mueven la sangre a través del cuerpo.
Sistemas de transporte	El personal, las máquinas y las organizaciones que transportan bienes.
Sistemas de armamentos	El equipo, procedimientos y el personal que hace posible utilizar el armamento.
Sistema escolar	Los edificios, los profesores, los administradores y los textos que funcionan conjuntamente para dar instrucción a los estudiantes.
Sistemas de computación	El equipo que conjuntamente funciona para llevar a cabo el procesamiento basado en el computador.
Sistema de contabilidad	Los registros, las reglas, los procedimientos y el personal que opera para registrar los datos, medir el ingreso y preparar los informes.

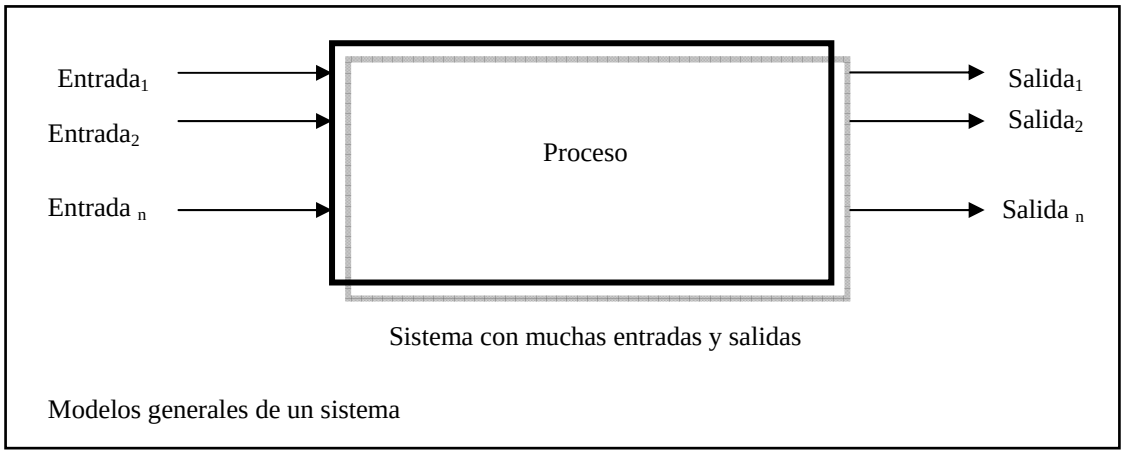
Los ejemplos ilustran cómo un sistema no es un conjunto ensamblado de elementos al azar; consiste en elementos que se pueden identificar cómo pertenecientes a un todo en razón de un propósito, meta u objetivo común. Los sistemas físicos son más que construcciones conceptuales: presentan actividades o comportamientos. Las partes interactúan para lograr un objetivo.

1.2. MODELO GENERAL DE UN SISTEMA

Un modelo general de un sistema físico es la entrada, el proceso y la salida. Esto, por supuesto, es muy simplificado en razón de que un sistema pueda tener varias entradas y salidas.



Modelo simplificado de un Sistema



Sistema con muchas entradas y salidas

Modelos generales de un sistema

Una característica de los sistemas es que las propiedades y el comportamiento de los componentes del sistema están inseparablemente entremezclados. El funcionamiento exitoso de cada componente del sistema depende del funcionamiento de otros componentes. Así, el software sólo puede funcionar si el procesador es operacional. El procesador sólo puede hacer cálculos si el sistema de software que define las operaciones se ha instalado de forma exitosa. La compleja relación entre los componentes de un sistema significa que este último es *más que la simple suma de sus partes.*

Las características que delinean un sistema configuran su ***límite***. El sistema está por dentro de los limites; el *medio ambiente* esta por fuera de los limites. En algunos casos es bastante sencillo de definir lo que es parte de un sistema y qué no lo es; en otros casos, la persona que estudia el sistema, arbitrariamente puede definir los límites. Algunos ejemplos de límites son:

Sistema	**Límites**
Humano	Piel, cabellos, uñas y las partes que están contenidas en el interior forman el sistema.
Automóvil	La carrocería del automóvil más las llantas y todas las partes contenidas dentro de él
Producción	Las máquinas de producción, los inventarios de producción de traba-

	jo en proceso, los empleados de producción, los procedimientos de producción, etc., forman el sistema.

El ejemplo del sistema de producción ilustra el problema del concepto de límite.

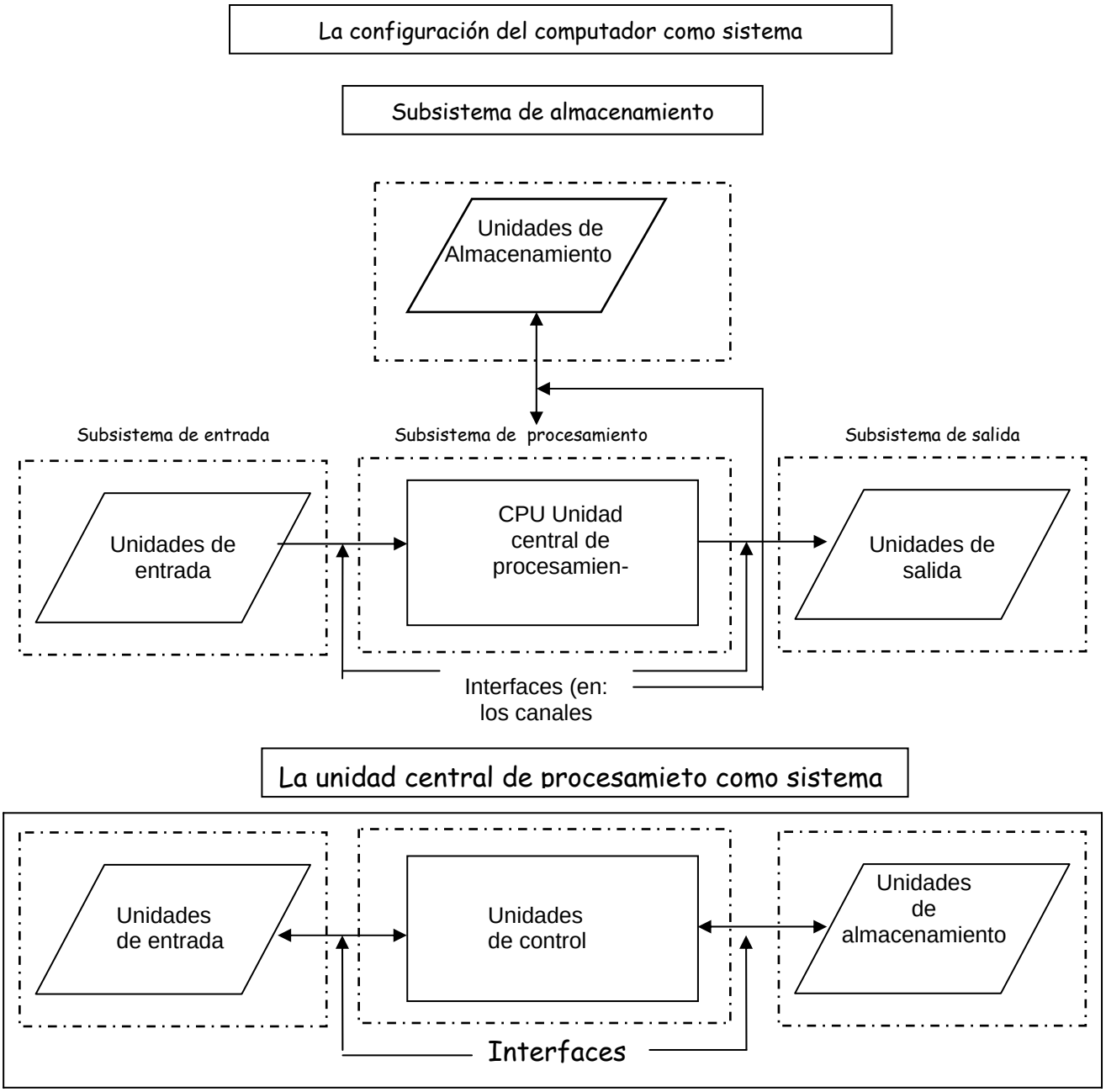
¿El inventario de materia prima está incluido en el sistema de producción? Una definición del sistema de producción puede incluir la materia prima en razón de que es necesaria para el propósito por el cual se estudia; otro uso puede excluirla.

1.3. SUBSISTEMAS

Por lo general, los sistemas son jerárquicos en el sentido de que incluyen otros sistemas. Por ejemplo, un sistema de órdenes y control policial puede incluir un sistema de información geográfico para proporcionar los detalles de la localización de los incidentes. Estos otros sistemas se denominan ***subsistemas***. Una característica de éstos es que se pueden operar por sí solos como sistemas independientes. Por lo tanto, el mismo sistema de información geográfica se puede utilizar en diferentes sistemas. Sin embargo, su comportamiento en un sistema particular depende de la relación con otros subsistemas.

El uso de subsistemas como la construcción por bloques, es básico para analizar y desarrollar los sistemas. Esto requiere la comprensión de los principios que dictaminan la manera como se construyen los sistemas a partir de los subsistemas.

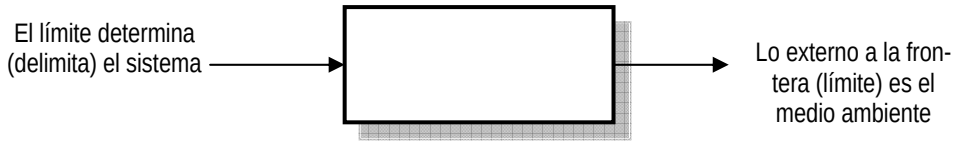
Cada subsistema es delineado por sus límites. Las interconexiones y las interacciones entre los subsistemas se llaman ***interfaces***. Las interfaces ocurren en el límite y toman la forma de entradas y salidas.


La configuración del computador como sistema
Subsistema de almacenamiento
Unidades de Almacenamiento
Subsistema de entrada
Subsistema de procesamiento
Subsistema de salida
Unidades de entrada
CPU Unidad central de procesamien-
Unidades de salida
Interfaces (en: los canales
La unidad central de procesamieto como sistema
Unidades de entrada
Unidades de control
Unidades de almacenamiento
Interfaces

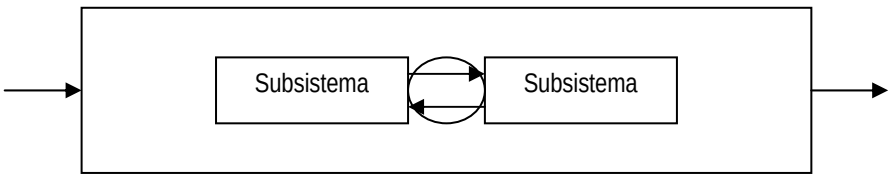
1.3.1. La aplicación como interface.

Un subsistema en el nivel más elemental (entrada, proceso, salida) no se define en cuanto al proceso. A este sistema se le llama una "*caja negra*", ya que las entradas y las salidas se conocen pero no la transformación actual a partir de las primeras sobre las otras. Los principales conceptos de sistemas, límites, interface, subsistema y caja negra se ilustran a continuación.

1.4. CLASES DE SISTEMAS

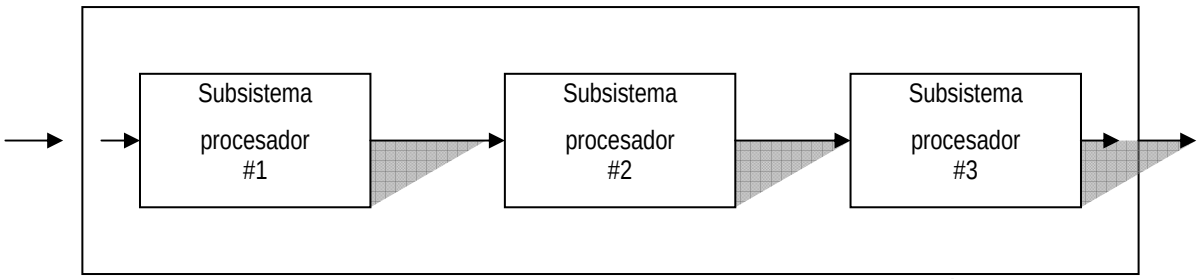


El concepto de frontera



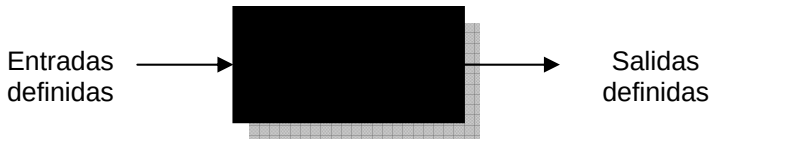
Interfaz-interconexión en la frontera

Interfaz



Caso simple de la operación de un subsistema de manera serial (en serie)

Sistema descompuesto en subsistemas



Procesador (transformación) no definido

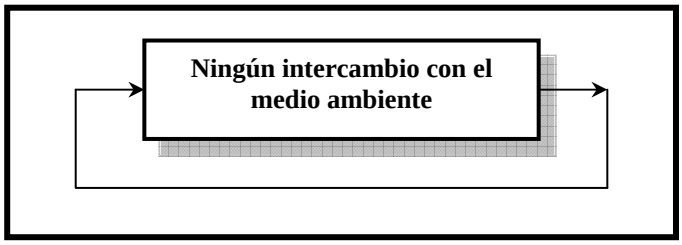
Una caja negra

Aunque muchos fenómenos tan disímiles como un ser humano y un programa de computador se pueden describir en términos de sistemas, son en realidad muy diferentes. Muchas maneras de clasificar los sistemas hacen hincapié en estas diferencias.

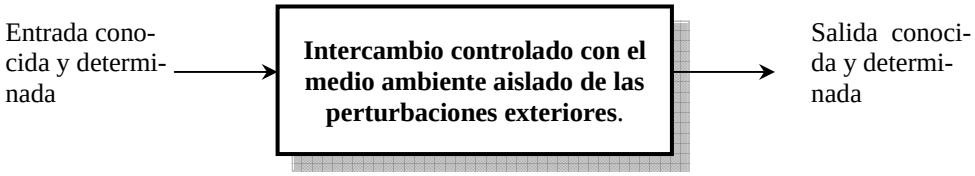
Criterio	Tipo de sistema	Descripción
Según el comportamiento	Determinístico	El sistema opera de una manera muy predecible. La interacción entre las partes se conoce con certeza. Si uno tiene la descripción de un estado del sistema en un momento dado además de una descripción de su operación, el siguiente estado del sistema se puede dar con exactitud, sin error.
	Probabilístico	El sistema se puede definir en términos de comportamiento probable; hay cierto grado de error que siempre está asociado a la predicción de lo que hará este sistema.
Según la cantidad de componentes	Simple	El sistema posee pocos componentes y la relación o interacción entre ellos es sencilla y directa.
	Complejo	El sistema posee muchos elementos estrechamente relacionados e interconectados.
Según los cambios en el tiempo	Estable	El sistema sufre escasos cambios al paso del tiempo.
	Dinámico	El sistema sufre rápidos y constantes cambios al paso del tiempo.
Según la capacidad de modificación	Adaptable	El sistema es capaz de modificarse en respuesta a cambios en el entorno.
	No adaptable	El sistema es incapaz de modificarse en respuesta a cambios en el entorno.
Según el tiempo de existencia	Permanente	El sistema está diseñado para existir durante un período relativamente largo.
	Temporal	El sistema está diseñado para existir durante un período relativamente corto.
Según el intercambio con el medio ambiente	Cerrado	El sistema está definido en lo físico como un sistema que está contenido en sí mismo. No intercambia materiales, información ni energía con su medio ambiente.
	Abierto	El sistema intercambia información, materiales o energía con el medio ambiente incluyendo el azar y entradas no definidas.

Un ejemplo de un sistema cerrado es una reacción química en un contenedor sellado y aislado. Tales sistemas cerrados finalmente terminarán o se convertirán en sistemas desorganizados. Este movimiento hacia el desorden se llama un *incremento de entropía*.

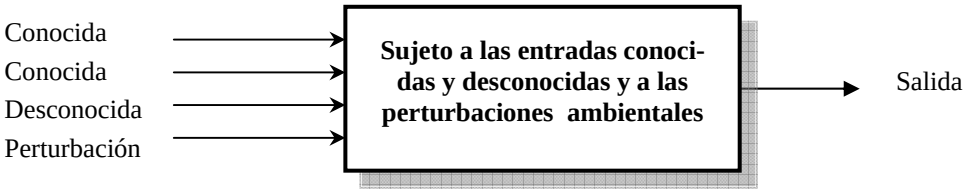
En las organizaciones y en el procesamiento de información, hay sistemas que están relativamente aislados del medio ambiente pero no completamente cerrados en el sentido físico. Estos serán llamados sistemas cerrados; lo que en realidad significa *relativamente* cerrados. Por ejemplo, los sistemas de manufactura con frecuencia son diseñados para minimizar los intercambios no deseados con el medio ambiente fuera del sistema. Un programa de computadora es relativamente cerrado en razón de que acepta solamente entradas previamente definidas, las procesa y provee salidas también previamente definidas. En resumen, el sistema relativamente cerrado es aquel que tiene solamente entradas y salidas controladas y bien definidas. No está expuesto a perturbaciones del exterior al sistema.



Sistema cerrado



Sistema relativamente cerrado



Sistemas abiertos

Concepto de sistema abierto y cerrado

Ejemplo de sistemas abiertos son los que les permiten adaptarse a los cambios de su medio ambiente en tal forma que puedan continuar su existencia. Son "auto-organizados" en el sentido de que modifican su organización en respuesta a las condiciones cambiantes. Los sistemas vivos (células, plantas, hombres, etc.) son sistemas abiertos; intentan mantener el equilibrio por la *homeóstasis*, el proceso de ajuste para conservar la operación del sistema dentro de los límites preestablecidos. Un ejemplo es el cuerpo que mantiene la temperatura dentro de unos límites determinados.

Las organizaciones son sistemas abiertos; una característica crítica de su existencia es su capacidad para adaptarse y afrontar los cambios de la competencia, los cambios del mercado, etc.

Los *sistemas artificiales* son creados, no ocurren en la naturaleza. Las organizaciones, los sistemas de información y los programas de computador son ejemplos de sistemas artificiales. Los sistemas artificiales están diseñados para apoyar los objetivos de los diseñadores y de los usuarios.

1.5. COMPONENTES DE UN SISTEMA INFORMÁTICO

Componente	Características
Hardware	Es todo el elemento electrónico, eléctrico y mecánico (tangible) que compone la computadora. Por ejemplo, el procesador, el monitor, la lectora de DVD, etc.
Software	Es el o los programas (no tangible) que hacen que la computadora funcione como tal más toda la documentación del mismo. Por ejemplo, el sistema operativo, los software de aplicación, los antivirus, etc.
Ser humano	Es quien coordina los dos elementos anteriores. Puede ser experto (programador, diseñador, etc.) o usuario.

1.6. CARACTERÍSTICAS IMPORTANTES DE LOS SISTEMAS

Todos los sistemas tienen niveles aceptables de desempeño, denominados ***estándares*** y contra los que se comparan los niveles de desempeño actuales. Siempre deben anotarse las actividades que se encuentran muy por encima o por debajo de los estándares para poder efectuar los ajustes necesarios. La información proporcionada al comparar los resultados con los estándares junto con el proceso de reportar las diferencias a los elementos de control recibe el nombre de ***retroalimentación***.

Los sistemas emplean un modelo de control básico consistente en:

1. Un ***estándar*** para lograr un desempeño aceptable
2. Un método para ***medir*** el desempeño actual
3. Un medio para ***comparar*** el desempeño actual contra el estándar
4. Un método de ***retroalimentación***

Los sistemas que pueden ajustar sus actividades para mantener niveles aceptables continúan funcionando. Aquellos que no lo hacen, tarde o temprano dejan de trabajar.

El concepto de interacción con el medio ambiente, que es lo que caracteriza a los sistemas abiertos, es esencial para el control. Recibir y evaluar la retroalimentación, permite al sistema determinar qué tan bien está operando. En contraste, los sistemas cerrados sostienen su nivel de operación siempre y cuando posean información de control adecuada y no necesiten nada de su medio ambiente. Un objetivo en el diseño de sistemas es construir sistemas que necesiten la menor intervención del medio externo para mantener un desempeño aceptable. Por consiguiente, la autorregulación y el propio ajuste son objetivos de diseño en todos los ambientes de sistemas.

1.7. DESEMPEÑO Y ESTÁNDARES DE SISTEMAS

El desempeño de un sistema puede medirse de varias maneras.

La eficiencia es una medida de lo que se produce dividido lo que se consume; puede ir del 0 al 100 por ciento.

La eficiencia es un término relativo empleado para comparar sistemas. Un motor de gasolina, por ejemplo, es más eficiente que un motor de vapor, pues con un monto equivalente de insumo de

energía (gasolina o carbón), el motor de gasolina produce más energía. El índice de eficiencia de energía de los motores de gasolina es alto en comparación con el de los motores de vapor.

La eficacia es una medida del grado en el que un sistema cumple sus metas.

Se le puede calcular al dividir las metas alcanzadas entre el total de las metas establecidas. Lo mismo que la eficiencia, la eficacia también es un término relativo que sirve para comparar sistemas.

Eficiencia y **eficacia** son objetivos de desempeño fijados en relación con un sistema general. El cumplimiento de estos objetivos supone considerar no sólo la eficiencia y eficacia deseadas, sino también el costo, complejidad y nivel de control que se desean del sistema. El costo comprende tanto los gastos iniciales de un sistema como la totalidad de sus gastos directos permanentes. La complejidad tiene que ver con qué tan complicada es la relación entre los elementos del sistema. El control es la capacidad de un sistema para funcionar dentro del marco de normas predefinidas –tales como políticas, procedimientos y presupuestos –, así como el esfuerzo administrativo requerido para mantener dentro de esos límites el funcionamiento del sistema. El cumplimiento de objetivos definidos de eficiencia y eficacia puede implicar disyuntivas en términos de costo, control y complejidad.

La evaluación del desempeño de un sistema demanda también el empleo de estándares de desempeño.

Un estándar de desempeño de sistemas es un objetivo específico del sistema.

Por ejemplo, un estándar de desempeño de sistemas de un proceso de manufactura podría ser la producción de no más de un 1 por ciento de partes defectuosas. Una vez establecidos los estándares, se mide el desempeño del sistema y se lo compara con el estándar. Las variaciones respecto al estándar son determinantes del desempeño del sistema. El cumplimiento de estándares de desempeño de sistemas también puede imponer disyuntivas en términos de costo, control y complejidad.

1.8. VARIABLES Y PARÁMETROS DE SISTEMAS

Ciertas partes de un sistema son susceptibles de control administrativo directo, mientras que otras no.

Una variable de sistemas es una cantidad o unidad que puede ser controlada por el tomador de decisiones.

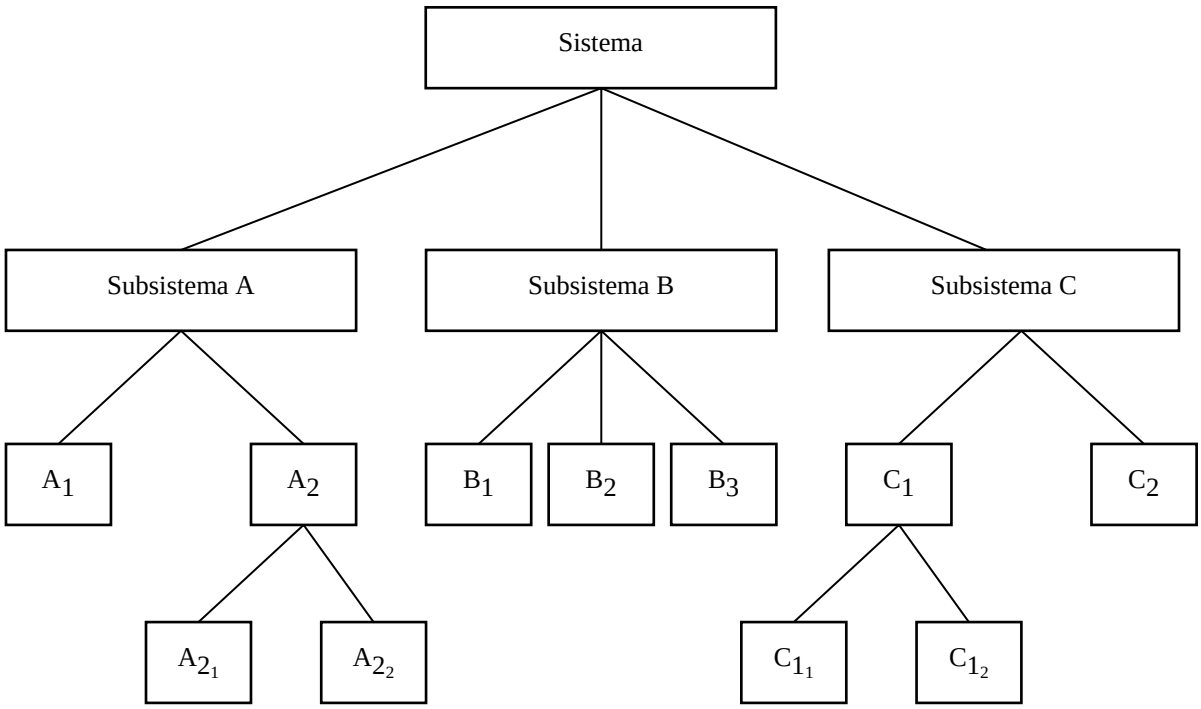
El precio que una compañía fija a su producto es una variable de sistemas, porque puede controlarse.

Un parámetro de sistemas es un valor o cantidad imposible de controlar.

El costo de una materia prima y la cantidad de gramos de un producto químico que habrá de utilizarse en la fabricación de cierto tipo de plástico son ejemplo de cantidad o valor no controlable por los administradores.

1.9. DESCOMPOSICIÓN

Un sistema complejo es difícil de comprender cuando se considera como un todo, por lo tanto, el sistema se descompone o factoriza en subsistemas. Los límites e interfaces están definidos de tal manera que la suma de los subsistemas constituye un sistema completo. Este proceso de descomposición se continúa con los subsistemas que se dividen en subsistemas más pequeños hasta que el más pequeño de los subsistemas tenga un tamaño manejable. Los subsistemas resultantes de este proceso generalmente tienen la forma de estructura jerárquica. En la jerárquía un subsistema es un elemento de un *suprasistema* (el sistema superior a él).



Relaciones jerárquicas de los subsistemas

Un ejemplo de descomposición es la factorización de un sistema de procesamiento de información en subsistemas. Un método para la descomposición podría proceder como sigue:

Sistemas de información dividido en subsistemas tal que:

a. Entradas de ventas y pedidos
b. Inventarios
c. Producción
d. Personal
e. Compras
f. Contabilidad y control
g. Planeación
h. Investigación (inteligencia) del medio ambiente.

La descomposición en subsistemas se usa tanto en el análisis del sistema actual como en el diseño e implementación de un nuevo sistema. En ambos casos el investigador o diseñador debe decidir cómo descomponerlo, por ejemplo, dónde ubicar los límites. Las decisiones dependerán de los obje-

tivos de la descomposición y también de las diferencias personales entre los diseñadores. Esta última se podría minimizar.

El principio general de la descomposición que supone que los objetivos del sistema dictaminan el proceso es la ***cohesión funcional***. Los componentes están considerados como parte del mismo subsistema si desempeñan o están relacionados a la misma función. Como ejemplo, un programa de aplicación al ser dividido en módulos (subsistemas) se dividirá entre las principales funciones del programa. En diseño, la identificación de los subsistemas cohesionados funcionalmente es el primer paso. En consecuencia, los límites necesitan estar claramente especificados, las interfaces simplificadas y establecidas las conexiones apropiadas entre los subsistemas.

1.10. SIMPLIFICACIÓN

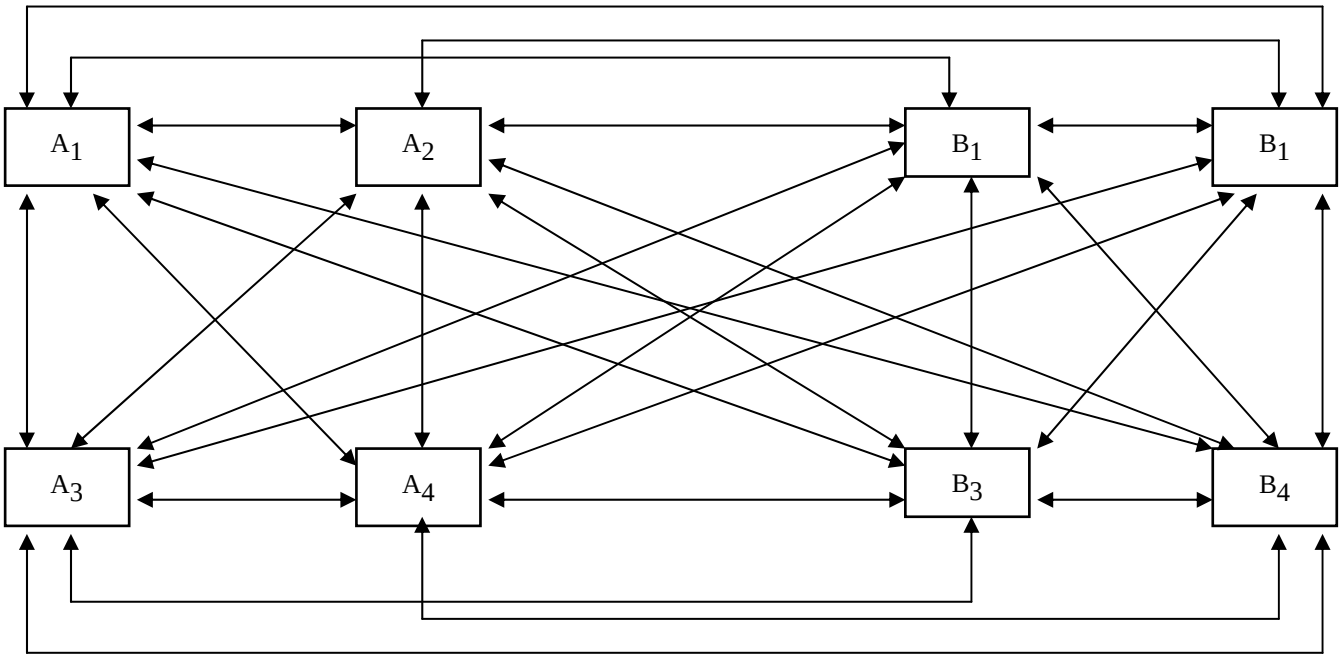
El proceso de descomposición podría conducir a un gran número de interfaces de subsistemas por definir. Por ejemplo, cuatro subsistemas que interactúan todos unos con otros tendrán seis interconexiones: un sistema con 20 subsistemas todos interactuando, tendrá 190 interconexiones.

El número de interconexiones si todos los subsistemas interactúan en general es:

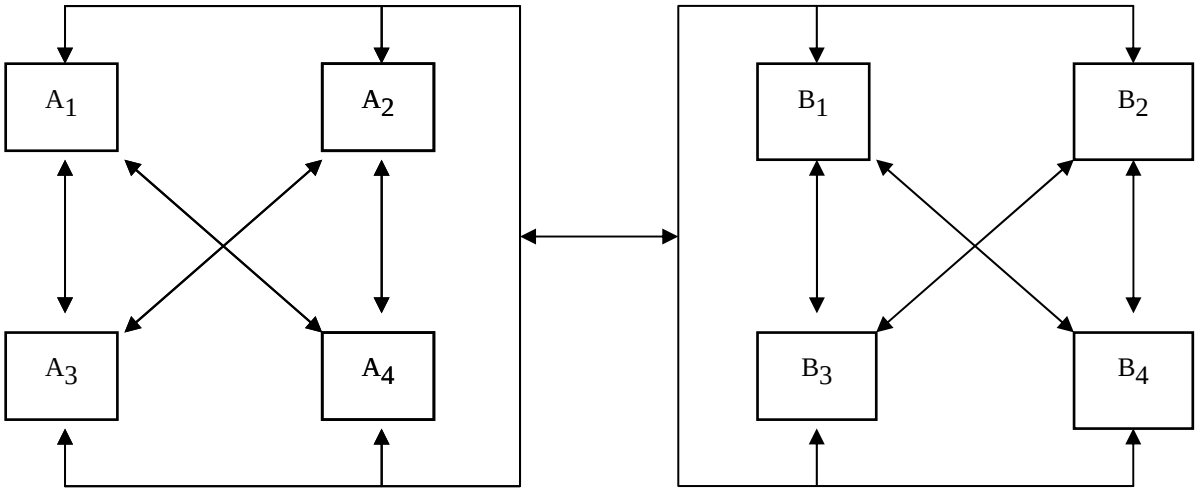
$$\frac{1}{2} n \cdot (n-1)$$

donde n= número de subsistemas.

Cada interconexión es una interfaz potencial para la comunicación entre los subsistemas. Cada interfaz implica una definición de un paso de comunicación.



Todos los subsistemas interconectados



Los subsistemas conectados dentro de un grupo y los grupos interconectados con una interfaz simple

La simplificación es el proceso de organizar los subsistemas de manera tal que se reduzca el número de interconexiones. Algunos métodos de simplificación son:

1. Se establece que las agrupaciones de subsistemas interactúan cada una con la otra, por lo tanto un simple paso de interfaz se define de un grupo hacia otros subsistemas o grupos de subsistemas (ver figura precedente). Un ejemplo es la base de datos a la cual se tiene acceso por varios programas, pero la interconexión se hace solamente a través de la interfaz de la administración de la base de datos.

2. Se establecen los métodos para el desacoplamiento de sistemas de tal manera que la necesidad de la interconexión se reduzca.

1.11. DESACOPLAMIENTO

Si los diferentes subsistemas están conectados de modo muy compacto se requiere entre ellos una coordinación muy exacta. Por ejemplo, si la materia prima entra directamente a producción en el momento en que llega a la fábrica, el sistema de materia prima se puede decir que está fuertemente acoplado. Bajo estas condiciones, las entregas de materia prima (insumos al sistema de producción y salidas provenientes del sistema de materias primas), deben hacerse oportunamente con el fin de evitar demoras en la producción o para prevenir que el material nuevo que llegue demasiado pronto, no tenga lugar donde almacenarse.

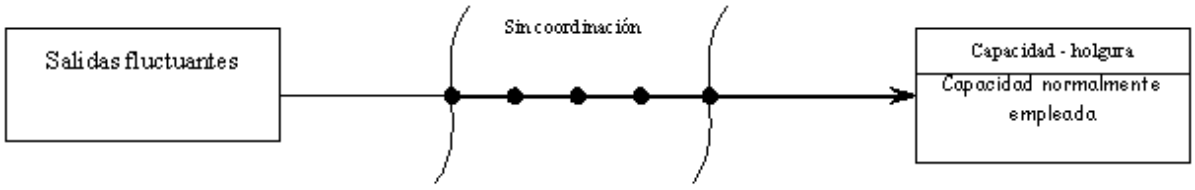
Tales acoplamientos tan compactos plantean una coordinación muy fuerte y exigencias de oportunidad entre los dos sistemas. En razón de que son algo independientes, es difícil hacer que operen de una manera completamente sincronizada, puesto que eventos al azar crean incertidumbre en los tiempos de entrega, y cambian los tiempos esperados de llegada. De la misma manera el proceso de producción puede experimentar demoras al azar o no planeadas. La solución es desacoplar o reducir conexiones de tal manera que los dos sistemas puedan operar en corto plazo con alguna medida de independencia.

Un sistema es altamente eficiente y eficaz cuando tiene una muy alta cohesión funcional y un muy bajo acoplamiento.

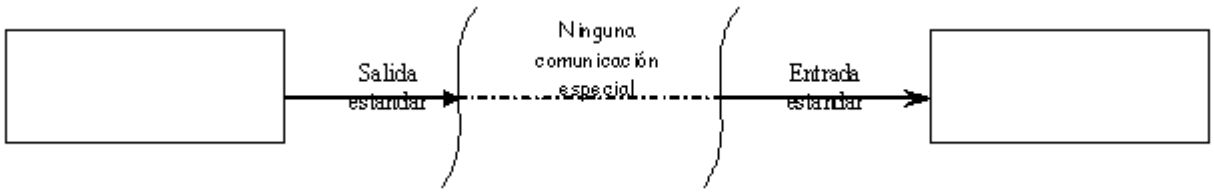
Algunos significados de desacoplamiento son:

1. ***Inventarios, almacenamientos intermedios, o líneas de espera.*** Se crea un almacenamiento de manera tal que no dependa la salida de la entrada del sistema. En el ejemplo de los subsistemas de materias primas y el subsistema de producción, el inventario de materias primas (*stock)* permite a los dos subsistemas operar de alguna manera independiente (en el corto plazo). Las memorias intermedias (*buffers*) de datos se utilizan en algunos sistemas de computación y en algunos sistemas de comunicación para compensar las diferentes relaciones de entradas y salidas de datos.

2. ***Recursos de holgura y flexibles.*** Cuando la salida de algún sistema es la entrada de otro, las existencias de recursos de holgura permiten a los subsistemas que sean algo independientes y aún más, que cada uno responda a las demandas de los otros subsistemas. La capacidad de la organización para responder a las variaciones en la demanda mediante el uso de recursos de holgura se mejora, si la disponibilidad de recursos se puede emplear para diferentes propósitos. Una organización de sistemas de información que utiliza el concepto de combinación de programadores y de analistas de sistemas tiene mas flexibilidad en responder a las variaciones en la demanda entre el análisis y la programación, que una organización con la misma cantidad de personal que utiliza analistas de sistemas solamente para el análisis y el diseño y emplea programadores solamente para la programación. (esta claro que es solo un ejemplo de consideración del problema de selección de trabajos combinados o separados).



3. ***Estándares***. La especificación de las normas, los costos de los estándares y otras normas le permiten a un subsistema planear y organizarse reduciendo la necesidad de comunicarse con otros subsistemas. Si, por ejemplo, el departamento de producción desea diseñar un módulo de procesamiento de datos que incluya bienes terminados y un código estándar de productos que sea utilizado por toda la organización, no tiene necesidad de comunicar y negociar con otros departamentos.



Los problemas de acoplamiento compacto no solamente se derivan de los problemas físicos de la coordinación de los movimientos de los recursos, sino también de los problemas de la comunicación. Los diferentes métodos de desacoplamiento reducen la necesidad de comunicación y permiten a los subsistemas comunicarse sobre bases de excepción. Solamente si el sistema comienza a operar por fuera de ciertos límites hace que los otros subsistemas, con los cuales se interconecta necesiten estar informados. El empleo de mecanismos de desacoplamiento puede por lo tanto ser visto como una alternativa al incremento en las comunicaciones. Esto implica que una mejora en el sistema de información o de comunicación puede aumentar la oportunidad para el acoplamiento compacto y puede reducir la necesidad de mecanismos de descoplamiento.

El proceso de desacoplamiento y el permitir a cada subsistema alguna independencia en el manejo de sus asuntos tiene muchos beneficios, pero no se hace sin costos. Uno de estos es el costo mismo de mecanismo de desacoplamiento (inventarios, almacenamiento intermedio, líneas de espera, recursos de holgura, estándares, etc.). Otros costos parten del hecho de cada subsistema puede actuar de la mejor manera posible como un subsistema, pero la suma de sus acciones puede no ser óptima para su organización. Este es el problema de la suboptimización. Por ejemplo, la producción puede estar organizada para emplear los equipos de producción mediante el planeamiento con varias semanas de anticipación; esto advierte al subsistema de ventas para satisfacer los pedidos urgentes de los clientes.

1.12. TENSIÓN DE SISTEMAS Y CAMBIO DE SISTEMAS

Los sistemas ya sean vivientes o artificiales, los sistemas organizacionales, los sistemas de información o los sistemas de control cambian en razón del esfuerzo de tensión que padecen.

Un esfuerzo de tensión (*stress*) es una fuerza que se transmite por intermedio de un suprasistema al sistema que hace que éste cambie de tal manera que el suprasistema puede lograr mejor sus objetivos.

1.12.1. Clases de tensiones (stress)

Hay dos formas básicas de tensiones que se pueden imponer sobre un sistema en forma separada o concurrente.

1. **Un cambio en el conjunto de objetivos del sistema.** Nuevas metas pueden ser creadas o las antiguas metas se pueden eliminar.

2. **Un cambio deseado en los niveles de ejecución con los objetivos existentes**. El nivel deseado de ejecución puede ser aumentado o disminuido.

1.12.2. Consecuencias de la Tensión

Cuando un suprasistema ejerce tensión sobre un sistema, el sistema cambiará para acomodarse a la tensión o se deteriorará; esto es, decaerá y terminará. Si el sistema no se acomoda al esfuerzo, el suprasistema se aminora, o termina la oferta de materia-energía y de entrada de información. Por ejemplo, en un ambiente de computación, si el sistema de aplicación del computador no satisface los requerimientos del usuario, caerá en desuso. Los usuarios no suministrarán la alimentación al sistema por lo cual los archivos del sistema se desactualizarán, o los datos suministrados contendrán errores en razón del poco interés de los usuarios en mantener la integridad de los datos de entrada.

1.12.3. Proceso de adaptación

Los sistemas se acomodan a la tensión mediante un cambio en la forma que pueden ser cambios estructurales o cambios en los procesos. Por ejemplo, un sistema de computación bajo tensión para un mayor grado de participación de los datos, puede ser cambiado por la instalación de terminales en sitios remotos. Esto es un cambio estructural. Un cambio en el proceso se constituye por las demandas para una mayor eficiencia que se pueda obtener o por el cambio en la forma como se clasifiquen los datos.

Es muy improbable que el sistema cambie para acomodar la tensión, pues sería un cambio global en su estructura y proceso. En cambio, aquellos responsables del cambio intentarán ubicarlo por intermedio de limitaciones a los procesos de ajuste solamente, en uno o varios de sus subsistemas.

Como regla general, los subsistemas más próximos a la tensión cambiarán en mayor medida. La proximidad a la tensión es funcional usualmente; el subsistema que realiza la función más semejante a la necesitada para aliviar la tensión, es el sistema más próximo a dicha tensión. Un ejemplo es el caso de una tensión en razón de la actualización no autorizada de un registro, en un sistema en línea; la subdivisión más próxima a la tensión es el subsistema de entrada que tenga la función de dar la autorización.

El concepto de tensión ayuda a explicar algo de la dinámica que hace que los sistemas cambien. El proceso de cambio del sistema de información sigue un modelo conceptual general de adaptación al sistema de tensión (stress).

2. DESARROLLO DE SISTEMAS

2.1. GENERALIDADES

2.1.1. ¿Qué es el software?

Muchas personas asocian el término ***software*** con los programas de computadora. De hecho, ésta es una visión muy restrictiva. El *software* no son sólo programas, sino todos los documentos asociados y la configuración de datos que se necesiten para hacer que estos programas operen de forma correcta.

2.1.2 Proceso Software y Producto Software

Un concepto que debe tenerse claro es la diferencia entre proceso software y producto software.

- **Proceso Software:** Es un conjunto de actividades y resultados asociados que producen un producto de software. Estas actividades son llevadas a cabo por los ingenieros de software. Existen cuatro actividades fundamentales de procesos que son comunes para todos los procesos de software:

Actividades	Características
Especificación del software	La funcionalidad del software y las restricciones sobre su operación deben quedar definidas.
Desarrollo del software	Debe producirse software que cumpla con la especificación.
Validación del software	El software debe validarse para asegurar qué es lo que el cliente requiere.
Evolución del software	El software debe evolucionar para cumplir con los cambios requeridos por el cliente.

- **Producto Software:** Consiste en programas desarrollados y en la documentación asociada. Existen dos tipos de producto de software:

  - *Productos genéricos:* Son sistemas aislados producidos por una organización de desarrollo y que se venden al mercado abierto a cualquier cliente que le sea posible comprarlo. También se los denomina software empaquetados o enlatados. Ejemplos son las bases de datos, los procesadores de texto, herramientas de administración de proyectos, etc.

  - *Productos personalizados:* Son sistemas requeridos por un cliente en particular. Ejemplos son sistemas desarrollados para llevar a cabo procesos de negocios específicos, sistemas de control aéreo, etc.

Los productos software tienen cierto número de atributos asociados que reflejan la calidad de ese software. Estos atributos no están directamente asociados con lo que el software hace. Más bien, reflejan su comportamiento durante su ejecución y en la estructura y organización del programa fuente y en la documentación asociada. Estos atributos, esenciales de un buen software, son:

Características del Producto	Descripción
Mantenibilidad	El software debe escribirse de tal forma que pueda evolucionar para cumplir las necesidades de cambio de los clientes.
Confiabilidad	El software debe tener un gran número de características, incluyendo la fiabilidad, seguridad y protección. No debe causar daños físicos o económicos en el caso de una falla del sistema.
Eficiencia	El software no debe hacer que se malgasten los recursos del sistema, como la memoria y los ciclos de procesamiento.
Usabilidad	El software debe ser fácil de utilizar, sin esfuerzo adicional por parte del usuario para quien está diseñado. Debe tener una interfaz y una documentación apropiada.

2.1.3 Modelado de proceso software

Un modelo de procesos de software es una descripción de un proceso software que se presenta desde una perspectiva particular. Por su naturaleza, los *modelos* son simplificaciones, por lo tanto un *modelo de procesos* del software es una abstracción de un proceso real.

El término "modelo" parece algo formal, pero representa un concepto que se maneja durante la mayor parte de la vida. Ejemplos de modelos son los siguientes:

- *Mapas:* modelos bidimensionales del mundo en que vivimos.
- *Globos terráqueos:* modelos tridimensionales de nuestro mundo.
- *Diagramas de flujo:* representaciones esquemáticas de las decisiones y la secuencia de actividades para llevar a cabo un determinado procedimiento.
- *Dibujos arquitectónicos:* representaciones esquemáticas de un edificio, o de un puente, etc.
- *Partituras musicales:* representaciones gráficas y textuales de notas musicales y tiempos de una pieza musical.

¿Por qué se construyen modelos? ¿Por qué no se construye simplemente el sistema mismo? La respuesta es que se pueden construir modelos de manera tal de enfatizar ciertas propiedades críticas del sistema, mientras que simultáneamente se desprecian otros de sus aspectos. Esto permite comunicarse con el usuario de una manera enfocada, sin distraerse con asuntos y características ajenas al sistema. Y si la comprensión de los requerimientos del usuario no fue la correcta (o de que el usuario cambió de parecer acerca de sus requerimientos), *se pueden hacer cambios en el modelo o desecharlo y hacer uno nuevo, de ser necesario.*

Por esta razón, el analista hace uso de herramientas de modelado para:

- Concentrarse en las propiedades importantes del sistema y al mismo tiempo restar atención a otras menos importantes.
- Discutir cambios y correcciones de los requerimientos del usuario, a bajo costo y con el riesgo mínimo.
- Verificar que el analista comprenda correctamente el ambiente del usuario y que lo haya respaldado con información documental para que los diseñadores de sistemas y los programadores puedan construir el sistema.

2.1.4. Participantes en el desarrollo de sistemas

El desarrollo eficaz de sistemas requiere un esfuerzo de grupo. Este equipo, llamado grupo de desarrollo, se encarga de establecer los objetivos del sistema de información y generar un sistema (software) que satisfaga tales objetivos para la organización. Consta de los siguientes roles:

Participante del grupo	Descripción
Beneficiarios	Son los individuos que obtienen alguna ventaja, en última instancia, ya sea en forma directa o a través del área de la organización a la que representan, del proyecto de desarrollo de sistemas.
Usuarios	Son las personas que interactúan con el sistema en forma regular. Puede tratarse de empleados, administradores, clientes o proveedores.
Analista de Sistemas	Es un profesional especializado en el análisis y diseño de sistemas empresariales.
Programador	Se encarga de modificar o desarrollar programas para satisfacer las necesidades de los usuarios.
Personal de apoyo adicional	Consiste de especialistas técnicos, entre ellos los expertos en bases de datos y telecomunicaciones, ingenieros de *hardware* y representantes de proveedores.

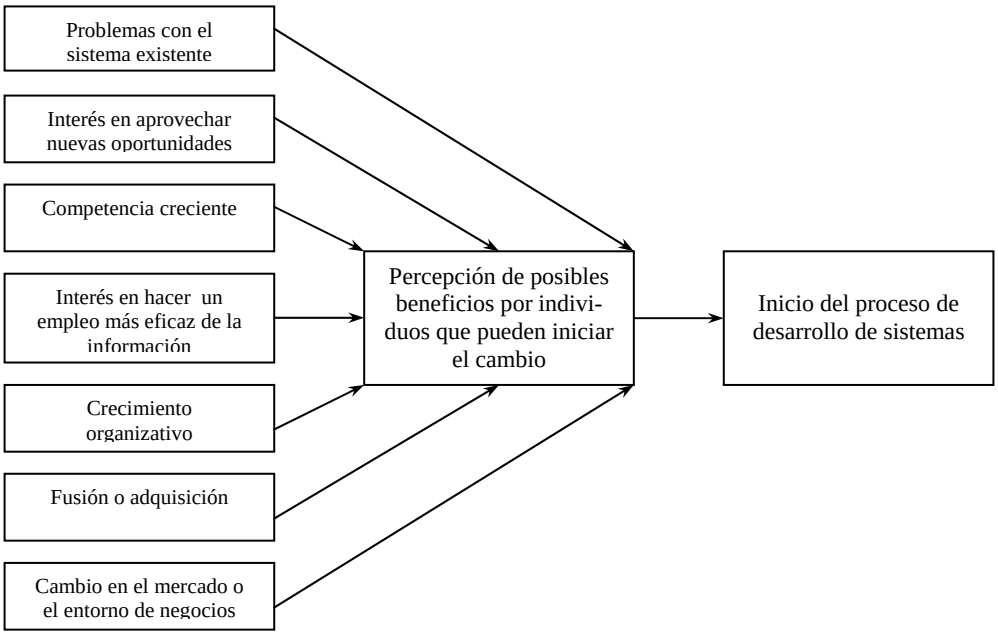
Sin importar la naturaleza específica del proyecto, el desarrollo de sistemas comprende sistemas nuevos o modificados; es decir, *implica cambios*. La administración eficaz de tales cambios obliga a que los miembros del grupo se comuniquen de manera satisfactoria.

2.1.5. Inicio del desarrollo de sistemas

Las actividades de desarrollo de sistemas empiezan cuando un individuo o grupo con la capacidad de iniciar cambios en la organización perciben un posible beneficio de un sistema nuevo o modificado. Ellos tienen interés en el desarrollo del sistema.

No obstante lo anterior, reviste igual importancia la capacidad del individuo o grupo para iniciar el cambio organizacional. Muchas personas no son capaces de iniciarlo. Ello podría deberse no a

que carezcan de motivación para hacerlo, sino a limitaciones de jerarquía, poder, autoridad y posición política en la organización.



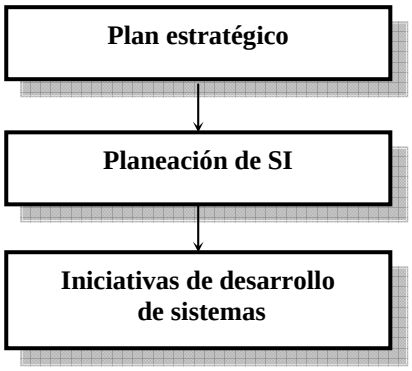
Razones frecuentes para iniciar un proyecto de desarrollo de sistemas

2.1.6. Planeación de sistemas de información

El plan estratégico de una organización contiene tanto los objetivos de la organización misma como una delineación general de los pasos necesarios para alcanzarlos, de modo que dicho plan tiene efecto en el tipo de sistema que necesita una organización. Por ejemplo, en un plan estratégico, podrían identificarse como objetivos de la organización la duplicación de los ingresos por ventas en cinco años, reducción de los gastos administrativos en 20% a tres años, adquisición de por lo menos dos empresas competidoras en un año, o lograr el liderazgo en una categoría de productos determinada. Es frecuente que una parte del plan estratégico incluya lineamientos acerca de cómo lograr los objetivos de la organización.

El término **planeación de sistemas de información** se refiere a la traducción de los objetivos estratégicos y organizacionales en iniciativas de desarrollo de sistemas.

Uno de los beneficios principales de la planeación de SI (Sistemas de Información) es que permite tener un panorama a largo plazo del uso de la tecnología de información en la organización. Otro beneficio es que garantiza el mejor uso de los recursos de sistemas de información.



La planeación de los sistemas de información transforma los objetivos de la organización, delineados en el plan estratégico, en actividades específicas de desarrollo de sistemas.

2.1.7. Desarrollo de una ventaja competitiva.

Muchas compañías buscan proyectos de desarrollo de sistemas que les brinden una ventaja competitiva, como parte de la conversión del plan estratégico corporativo en un plan de sistemas de información. Es común que ello requiera análisis *creativo* y *crítico*.

- El **análisis creativo** requiere investigar nuevos métodos de los problemas existentes. Al analizar los problemas, en formas nuevas o distintas, y usar métodos innovadores para resolverlos, muchas empresas han logrado ventajas competitivas.

- El **análisis crítico** requiere preguntarse, de manera exhaustiva y no prejuiciosa, si los elementos del sistema se relacionan o no en la forma más efectiva y eficaz. Requiere considerar el establecimiento de nuevas relaciones entre dichos elementos y, quizá, introducir nuevos elementos en el sistema. El análisis crítico en el desarrollo de sistemas comprende las acciones siguientes:
  - *Ir más allá de automatizar sistemas manuales.* Muchas organizaciones usan el desarrollo de sistemas simplemente para automatizar los sistemas manuales existentes, lo cual puede hacer que éstos se vuelvan más rápidos y eficaces. Sin embargo, en caso de que el sistema manual existente tenga deficiencias, automatizarlo sólo incrementará el efecto de tales deficiencias. Además, la automatización de sistemas manuales existentes puede hacer que se pierda una variedad de oportunidades, pues se continúa trabajando de la misma manera. El análisis crítico en el desarrollo de sistemas requiere preguntarse la razón de que se trabaje de cierta manera y considerar métodos opcionales.

  - *Plantear preguntas acerca de afirmaciones y supuestos.* Interrogar a los usuarios acerca de sus necesidades y aclarar sus respuestas iniciales puede llevar a mejores sistemas y predicciones más precisas.

  - *Identificar y resolver objetivos y orientaciones en conflicto.* Los diferentes departamentos de una organización pueden tener objetivos y orientaciones distintos. Se deben identificar y resolver estas diferencias antes de desarrollar un nuevo sistema.

2.1.8. Definición de objetivos para el desarrollo de sistemas

El efecto de un sistema particular en la capacidad de una organización para lograr sus objetivos depende del valor real que dicho sistema tenga para la organización.

Los objetivos que definen una organización, a su vez, determinan los que se establecen para un sistema. Un objetivo específico podría consistir en el aviso a los planificadores de mantenimiento de cuándo debe darse mantenimiento preventivo periódico (limpieza y lubricación, por ejemplo) a un equipo en particular. Otro objetivo sería alertarlos cuando las existencias de materiales de limpieza, aceites para lubricación y refacciones disminuyan por debajo de límites preestablecidos. Estos objetivos se lograrían mediante la reposición automática de existencias por intercambio electrónico de datos o informes de excepción.

Sin importar el desarrollo de sistemas específico de que se trate, en dicho proceso hay que definir un sistema con rendimiento y objetivos de costos específicos. El éxito o fracaso de la actividad de desarrollo del sistema se medirá contra tales objetivos.

- *Objetivos de desempeño***.** El grado en que un sistema funciona como se pretende puede medirse a través de objetivos de rendimiento. Es común que el rendimiento de un sistema dependa de factores como los siguientes:

    - *La calidad o utilidad de la salida.*
    - *La calidad o utilidad del formato de la salida.*
    - *La velocidad con que se genera la salida.*

En algunos casos, el 1ogro de los objetivos de desempeño puede medirse con facilidad. En otras ocasiones es más difícil de juzgar en el corto plazo. Sin embargo, es frecuente que estos resultados guarden relación estrecha con los objetivos de la compañía y sean vitales para su éxito a largo plazo.

- *Objetivos de costos***.** Los beneficios del logro de los objetivos de rendimiento deben equilibrarse con los costos relacionados con el sistema, incluidos los siguientes:
    - *Costos de desarrollo*. Deben incluirse todos los costos necesarios para que el sistema funcione.
    - *Costos relacionados con el carácter específico de la aplicación del sistema.* La singularidad de un sistema tiene efecto considerable en su costo.
    - *Inversiones fijas en hardware y equipo relacionado.* Los desarrolladores han de considerar costos como los de computadoras, equipo de red y centros de datos con ambiente controlado en que se operará el equipo.
    - *Costos de operación del sistema.* Éstos incluyen los de personal, *software* e insumos, entre estos últimos la energía eléctrica necesaria para operar el sistema.

2.1.9. Desarrollo de sistemas e Internet

Son cada vez más las compañías que convierten al menos una parte de sus actividades empresariales de modo que se efectúen en Internet, intranets o extranets. Las aplicaciones que se transfieren a Internet incluyen las de venta de productos a clientes, colocación de pedidos con proveedores y acceso de los clientes a información sobre producción, existencias, pedidos o cuentas por cobrar. La tecnología de Internet constituye una plataforma para aplicaciones que permite a las compañías extender su sistema de procesamiento de transacciones (TPS, *transaction processing systems*) más allá de los límites de la organización, a sus clientes, proveedores y socios comerciales. Ello posibilita que las empresas hagan negocios con mucha más rapidez, interactúen con más personas y traten de

mantenerse un paso adelante de sus competidoras.

Es relativamente fácil crear un sitio Web estático que muestre texto e imágenes sencillos. Sin embargo, poner en práctica una aplicación de negocios central dinámica que opere en la Web es mucho más complejo. Tales aplicaciones deben satisfacer necesidades especiales de la empresa. Deben ser escalables para brindar sustento a rendimientos específicos muy variables de transacciones con miles de usuarios potenciales. En teoría, dicha escalabilidad ha de ser posible en forma instantánea cuando resulte necesaria. Asimismo, se requiere que sean confiables y tolerantes a las fallas, con disponibilidad continua y procesamiento exacto de todas las transacciones. Además tienen que integrarse con la infraestructura existente, lo que incluye las bases de datos de clientes y pedidos, aplicaciones existentes y sistemas de planeación de recursos de la empresa. Por último, se requiere que su desarrollo y mantenimiento sean rápidos y sencillos, puesto que las necesidades de la empresa pueden requerir el cambio de las aplicaciones sobre la marcha.

2.1.10. Tendencias en el desarrollo de sistemas y planeación de recursos empresariales

El *software* de planeación de recursos empresariales (ERP, *Enterprise Resource Planning*) ha ido más allá de los procesos empresariales e influye cada vez más en el desarrollo de sistemas. No sólo los planificadores consideran diferentes tipos de sistemas que incluyen *software* de ERP, sino que el *software* de ERP ya instalado está haciendo que los planificadores exploren y desarrollen tipos diferentes de sistemas. Otros usuarios de *software* de ERP pasan ya de simplemente utilizarlo para el funcionamiento de las empresas, a emplearlo en la toma de decisiones de negocios.

Una tendencia importante en el desarrollo de sistemas y el uso de los sistemas de ERP es que las compañías desean permanecer con su proveedor de ERP original (SAP, Oracle, PeopleSoft, etc.), en vez de buscar respuesta en otros sitios a sus necesidades de planeación de producción y almacenamiento de datos o de crear soluciones dentro de la empresa. En su lugar, buscan a su proveedor original de ERP para que les brinde tales soluciones.

Una segunda tendencia es que muchos proveedores de *software* desarrollan en la actualidad *software* que se integra con sus propios paquetes de ERP.

Una tercera e interesante tendencia es el aumento en la cantidad de compañías que se ramifican para brindar consultoría a otras empresas una vez que ponen en marcha con éxito su propio proyecto de ERP.

2.2. PROCESOS DE DESARROLLO

El proceso de desarrollo de sistemas también se denomina ***Ciclo de Vida*** del sistema (SDLC, systems development life cycle), dado que las actividades relacionadas con dicho proceso son continuas. Se diferencia "*Ciclo de Vida*" de "*Proceso de Desarrollo*" porque dentro de distintos procesos de Desarrollo se pueden adoptar el Ciclo de vida que mas útil o adecuado le resulte o un mismo proceso puede utilizar varios Ciclo de Vida..

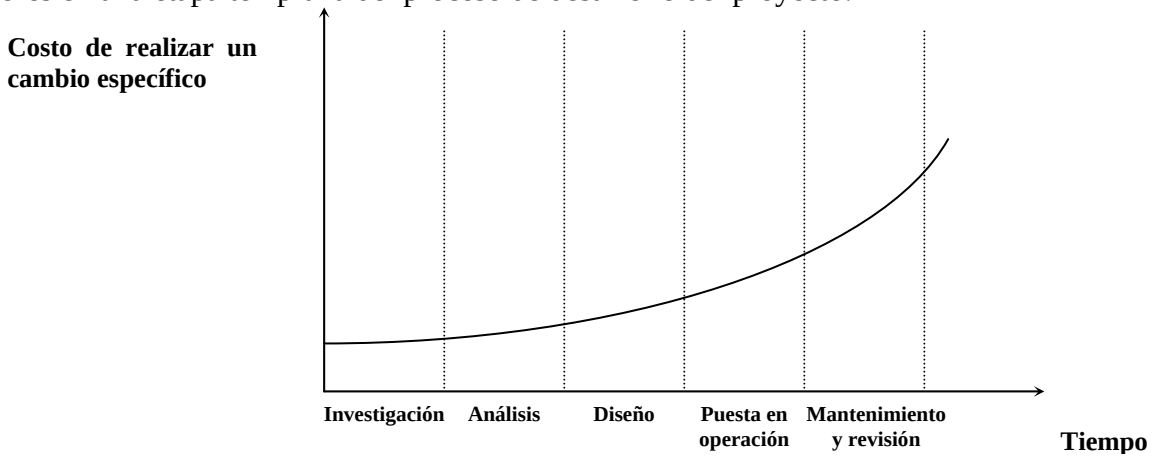
A medida que se crea cada sistema, el proyecto tiene calendarios y fechas límite, hasta que por último se instala y acepta. La vida del sistema continúa con su mantenimiento y revisión. Se inicia un nuevo proyecto si el sistema requiere mejoras significativas, que van más allá del alcance de su mantenimiento, si es necesario reponerlo a causa de una nueva generación de tecnología, o las necesidades del SI (Sistema de Información) de la organización cambian en forma importante.

Los cinco pasos o etapas comunes en todos los desarrollos son:

Etapa	Características
Investigación	En esta etapa se buscan los *requerimientos* del sistema, es decir, se identifican los posibles problemas y oportunidades, y se consideran a la

	luz de los objetivos de la empresa. Esto es, **entender el problema.**
Análisis	Esta fase incluye el estudio de los sistemas y procesos de trabajo existentes para identificar puntos fuertes y débiles, así como oportunidades de mejoramiento. Se detallan las salidas y entradas del sistema, interfaces del usuario y, componentes de hardware, software, base de datos, telecomunicaciones, personal y procedimientos, además de mostrar la relación de todos estos componentes. Esto es, **entender la solución.**
Diseño	Esta etapa es la más importante y quizás la más salteada por los creadores de software improvisados. Implica la creación de modelos estándar para una especificación completa del sistema. Esto es, **planear y elegir la mejor solución.**
Pruebas e Implementación	En esta etapa se realiza el "*testing*" del sistema, esto es, la prueba de cada componente del mismo hasta llegar a una integración total. Consiste en crear o adquirir los diversos componentes del sistema detallados en el diseño, juntarlos y poner en operación el sistema nuevo o modificado. Una tarea importante en esta fase es el adiestramiento de los usuarios. Esto es, **poner en funcionamiento la solución.**
Mantenimiento y revisión	Esta es la etapa más extensa en la vida de un software. Es garantizar la operación del sistema y modificarlo, de modo que continúe cubriendo las necesidades cambiantes de la empresa. Implica el mantenimiento del software, que puede ser: *para reparar fallas del software* (corregir errores inadvertidos durante el proceso), *para adaptar el software a diferentes entornos operativos* (funcionalidades que mejoran la *performance* del sistema) y *para agregar o modificar la funcionalidad del sistema* (actualización del sistema a los nuevos cambios). Esto es, **evaluar la solución.**

Un aspecto básico en el desarrollo de sistemas es que ***entre más avanzado esté el SDLC en el momento de la detección de un error, será más costosa su corrección.*** Una razón para ello es que la identificación de un error en una fase avanzada del SDLC obliga a modificar, hasta cierto punto, las fases previas. Otra razón es que los errores detectados en una etapa más avanzada del SDLC, tienen efecto en más personas. Por ejemplo, un error identificado después de instalar un sistema puede requerir el readiestramiento de los usuarios, toda vez que se cuente con la "solución" del problema. Así pues, los desarrolladores experimentados de sistemas prefieren un método que detecte errores en una etapa temprana del proceso de desarrollo del proyecto.



Entre más tardíos sean los cambios en el SDLC, mayor será su costo

2.2.1. Proceso de desarrollo de software

2.2.1.1. Modelos Tradicionales

Este proceso de desarrollo de sistemas presenta dos alternativas diferenciadas en la etapa de análisis y diseño: *lineal* y *estructurada*.

El desarrollo de sistemas tradicional puede ir desde un pequeño proyecto, como la adquisición de un programa de bajo costo de computadora, hasta un gran proyecto, como la instalación de un sistema con costo de millones de dólares. Es verdad que en este último caso, este desarrollo no es el más óptimo. Los pasos del desarrollo de sistemas tradicional varían de una compañía a otra, si bien muchos métodos incluyen cinco fases: investigación, análisis, diseño, puesta en operación y mantenimiento y revisión.

- ***Características***:
  - Permite un alto grado de control administrativo.
  - Al final de cada fase, se emprende una revisión formal y se toma la decisión de continuar el proyecto, interrumpirlo o quizá repetir algunas tareas de la fase actual.
  - Crea mucha documentación. Si se mantiene actualizada, esta documentación puede ser útil cuando llegue el momento de modificar el sistema.
  - Garantiza que cada requisito del sistema pueda relacionarse con una necesidad de la empresa.
  - Se pueden revisar los productos resultantes para verificar que satisfagan los requisitos del sistema y se ajusten a las normas de la organización.
  - Un problema importante con el proceso de desarrollo tradicional es que el usuario no utiliza la solución hasta que el sistema está casi terminado.
  - Es inflexible, no da cabida a cambios en las necesidades de los usuarios durante el desarrollo.

Las ventajas y desventajas del proceso de desarrollo tradicional se presentan a continuación:

VENTAJAS	DESVENTAJAS
La revisión formal al final de cada fase permite el control administrativo máximo.	Los usuarios reciben un sistema que satisface sus necesidades, desde el punto de vista de los desarrolladores; puede ser que no corresponda a lo que en realidad se necesitaba.
Este método genera documentación considerable del sistema.	La documentación es costosa y su creación requiere tiempo. También es difícil mantenerla actualizada.
La documentación formal garantiza que sea posible vincular los requisitos de sistema con las necesidades específicas de la empresa.	Es frecuente que las necesidades de los usuarios no se expresen o se les interprete en forma correcta.
Genera muchos productos intermedios que se pueden revisar con el fin de indagar si satisfacen o no las necesidades de los usuarios y se ajustan a los estándares.	Los usuarios no pueden revisar fácilmente los productos intermedios y evaluar si un producto específico (por ejemplo, un diagrama de flujo de datos) satisface sus necesidades.

Estructurado

Muchos especialistas en sistemas de información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método de desarrollo del análisis estructurado tiene como finalidad superar esta dificultad por medio de:

1) La división del sistema en componentes.
2) La construcción de un modelo del sistema.

El método incorpora elementos tanto de análisis como de diseño. El ***análisis estructurado*** se concentra en especificar lo que se requiere que haga el sistema o la aplicación. No se establece cómo se cumplirán los requerimientos o la forma en que implantará la aplicación. Más bien permite que las personas observen los elementos lógicos (lo que hará el sistema) separados de los componentes físicos (computadoras, terminales, sistemas de almacenamiento, etc.). Después de esto se puede desarrollar un diseño físico eficiente para la situación donde será utilizado.

*El **diseño estructurado**,* otro elemento del análisis estructurado que emplea la descripción gráfica, se enfoca en el desarrollo de especificaciones del software. **La meta del diseño estructurado es crear programas formados por módulos independientes unos de otros desde el punto de vista funcional.** Este enfoque nos sólo conduce hacia mejores programas sino que facilita el mantenimiento de los mismos cuando surja la necesidad de hacerlo.

El diseño estructurado es una técnica específica para el diseño de programas y no un método de diseño de comprensión. Es decir, no indica nada relacionado con el diseño de archivos o bases de datos, la presentación de entradas o salidas, la secuencia de procesamiento o el hardware que dará soporte a la aplicación. Esta técnica conduce a la especificación de módulos de programa que son funcionalmente independientes.

La herramienta fundamental del diseño estructurado es el diagrama estructurado. Los diagramas estructurados son de naturaleza gráfica y evitan cualquier referencia relacionada con el hardware o detalles físicos. Su finalidad no es mostrar la lógica de los programas (esa es la tarea de los diagramas de flujo solamente).

Los elementos (diagramas) del análisis estructurado, son:

Diagrama	Elementos	Características
Diagrama de Contexto	▪ Burbuja (**sistema**). ▪ Flechas (**interacción con el exterior**). ▪ Rectángulos (**terminadores**)	▪ En una sola burbuja representa todo el sistema. ▪ Las personas, organizaciones y sistemas con los que se comunica el sistema son TERMINADORES. ▪ Enfatiza varias características importantes del sistema: los datos que se reciben y se envían desde el sistema y la frontera del mismo.

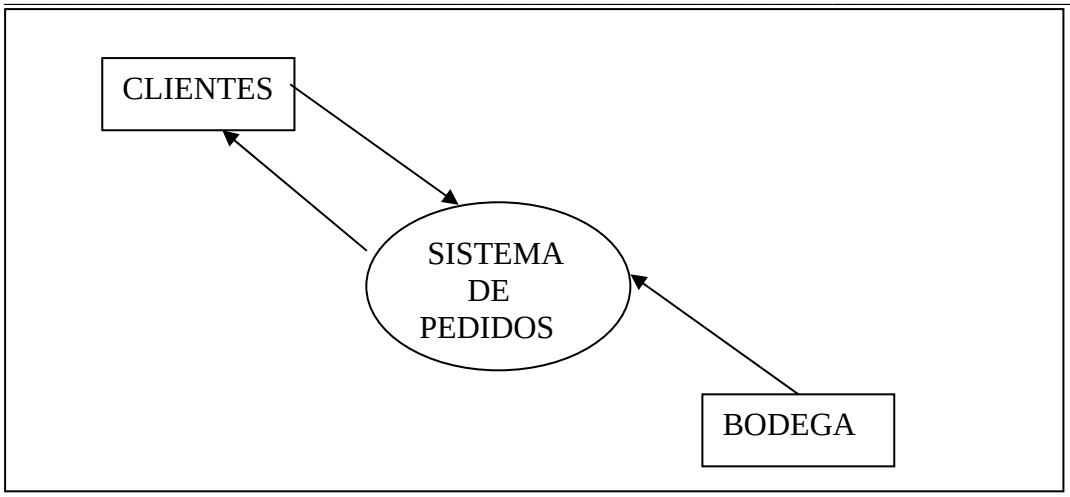


Diagrama	Elementos	Características
Diagrama de Flujo de Datos **(DFD)**	▪ Burbuja (**procesos**). ▪ Flechas curvas (**flujo de datos**). ▪ Rectángulos (**terminadores**). ▪ Líneas paralelas (**agregado de datos**)	▪ Es la herramienta de modelado que se utiliza para describir la transformación de entradas a salidas. Ilustra las ***funciones*** del mismo. ▪ Consiste en: ✓ ***Procesos:*** Las diversas funciones individuales que el sistema lleva a cabo. ✓ ***Agregados de datos:*** Muestran colecciones de datos que el sistema debe recordar por un período de tiempo. ✓ ***Flujos****:* Son las conexiones entre los procesos y representan la información de entrada o salida de los mismos. ✓ ***Terminadores:*** muestran las entidades externas con las que el sistema se comunica.

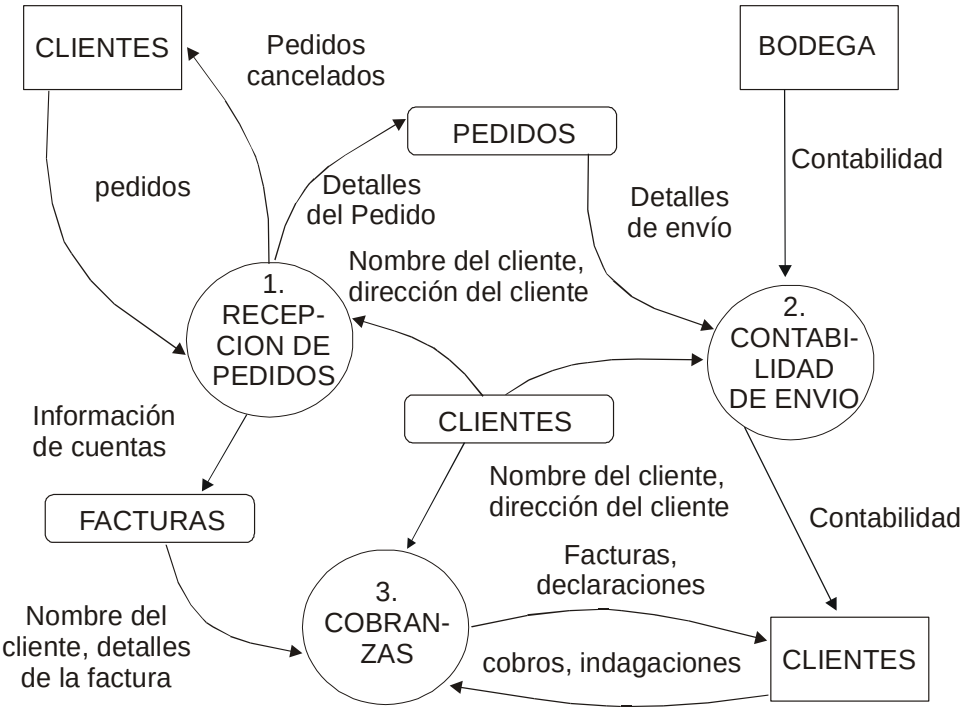


Diagrama	Elementos	Características
Diccionario de Datos	▪ Lenguaje escrito.	▪ Herramienta textual de modelado. ▪ Muestra ***qué*** información se transforma.

Nombre = **Tratamiento de cortesía o título + nombre + apellidos**

Tratamiento de cortesía o título = [Sr. / Srta. / Sra. / Dr. / Prof.]

Nombre = **{carácter válido}**

Apellido = **{carácter válido}**

Carácter válido = [A-Z / a-z / ‘ / - / /]

Diagrama	Elementos	Características
Especificación de Procesos	▪ Lenguaje escrito.	▪ Herramienta textual de modelado ▪ Muestra ***cómo*** la información se transforma.

1. **Si** el monto en dólares de la factura multiplicado por el número e semanas de retraso en el pago rebasa los 10.000 dólares **ENTONCES:**
    a. Proporcionar una fotocopia de la factura al encargado de ventas que llamará al cliente.
    b. Anotar en el reverso de la factura que se le dio una copia al vendedor, junto con la fecha en la que se hizo esto.
    c. Volver a archivar la factura para estudiarla de nuevo dentro de dos semanas.
2. **EN CASO CONTRARIO, SI** se han enviado más de cuatro recordatorios **ENTONCES:**
    a. Dar una copia de la factura al vendedor apropiado para que llame al cliente.
    b. Registrar en el reverso de la factura que una copia ha sido enviada al vendedor, y la fecha en la que se hizo esto.
    c. Volver a archivar la factura para reexaminarla dentro de una semana.
3. **EN CASO CONTRARIO** (la situación aún no ha alcanzado proporciones serias):
    a. Añadir 1 al contador de avisos de moratoria registrado en el inverso de la factura (si no se ha registrado tal contador, escribir: “cuenta vencida de avisos de moratoria = 1”)
    b. Si la factura archivada es ilegible **ENTONCES** mecanografiar una nueva.
    c. Enviar una copia de la factura al cliente, con el sello: “n-ésimo aviso: pago de factura vencido. Favor de remitir inmediatamente”, donde n es el valor de avisos de moratoria.
    d. Registrar en el reverso de la factura la fecha en la que se envió el n-ésimo aviso de moratoria.
    e. Volver a archivar la factura para examinarla dentro de dos semanas.

Diagrama	Elementos	Características
Diagrama de Entidad – Relación **(DER)**	▪ Rectángulo (**tipos de objeto**) ▪ Rombos (**relaciones**)	▪ Modela la relación que existe ***entre*** los agregados o datos. ▪ Consiste en: ✓ ***Tipos de objeto:*** Representa una colección o conjunto de objetos (cosas) del mundo real cuyos miembros juegan algún papel en el desarrollo del sistema. Pueden además ser identificados de manera única y ser descritos por uno o más atributos. ✓ ***Relaciones***: Son la serie de conexiones o asociaciones entre los tipos de objetos que están conectados con la relación por medio de flechas.

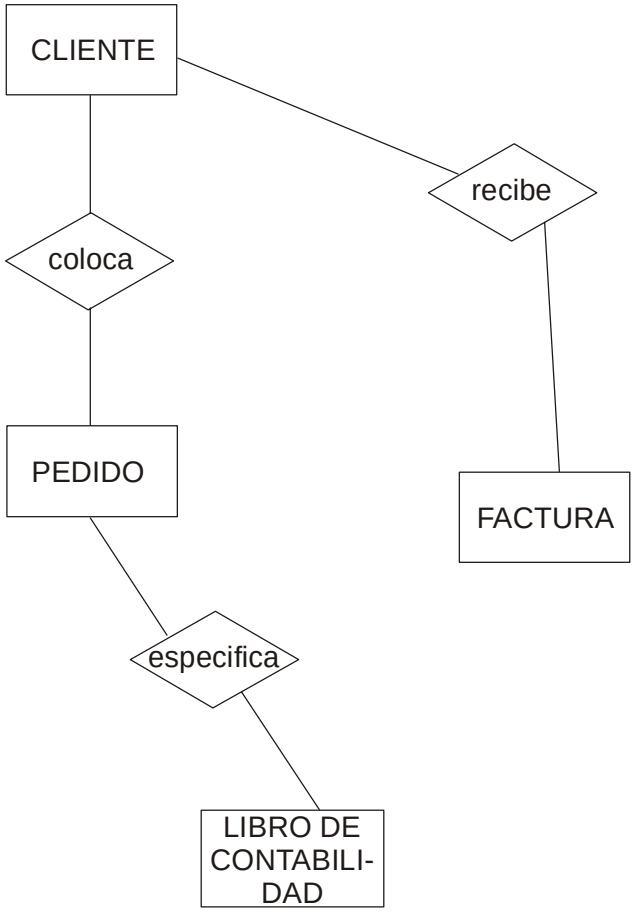


Diagrama	Elementos	Características
Diagrama de Transición de Estados **(STD)**	▪ Rectángulo (**estado**) ▪ Flechas (**cambio de estado**)	▪ Describe el comportamiento del sistema dependiente del ***tiempo***, evaluando las ***condiciones*** con la cual se cambia de estado y las ***acciones*** pertinentes para lograrlo.

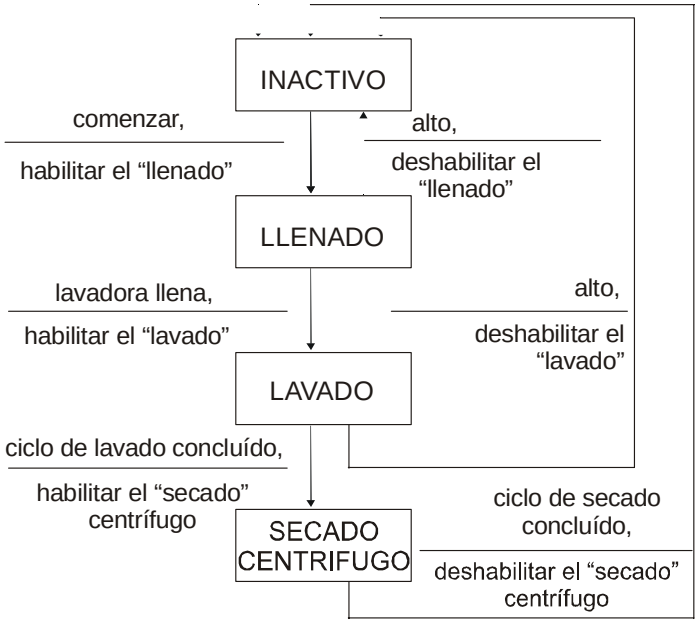
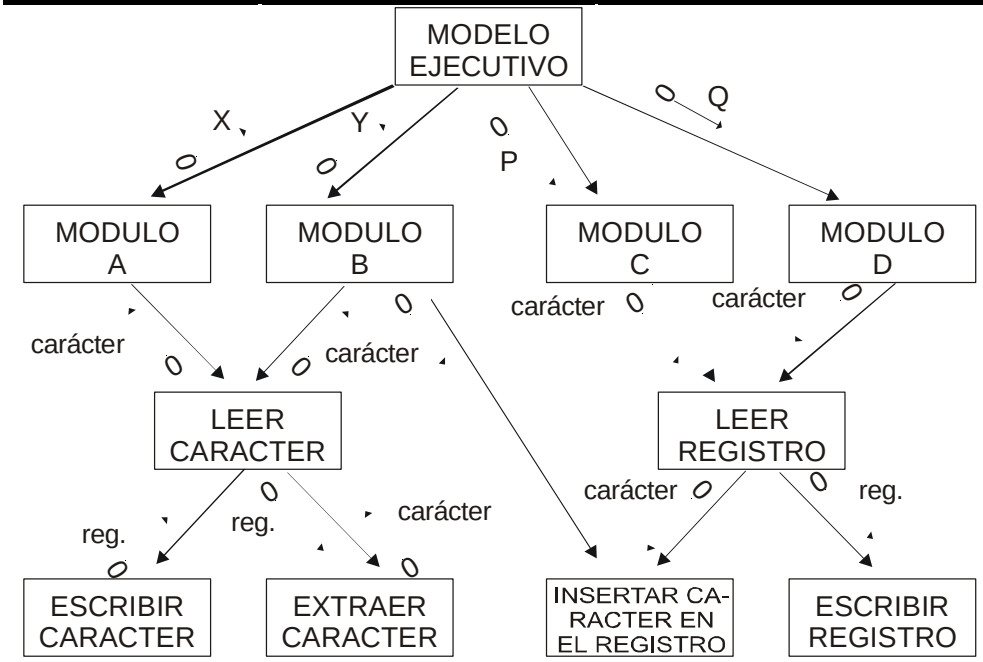


Diagrama	Elementos	Características
Diagrama de Estructuras	▪ Rectángulo (**módulo**) ▪ Flechas (**invocaciones a los módulos**)	▪ Es una herramienta gráfica de modelado comúnmente utilizada para representar la jerarquía de módulos. ▪ Muestra los parámetros de entrada que se le dan a cada módulo invocado, y los parámetros de salida devueltos por cada módulo cuando termina.



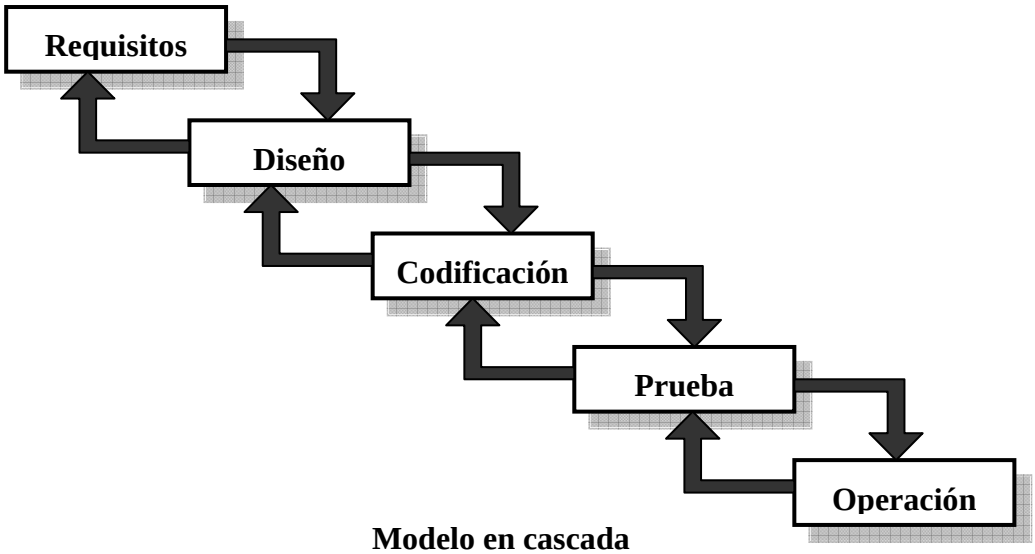
- **Lineal**

El desarrollo de sistemas tradicional puede ir desde un pequeño proyecto, como la adquisición de un programa de bajo costo de computadora, hasta un gran proyecto, como la instalación de un sistema con costo de millones de dólares. Los pasos del desarrollo de sistemas tradicional varían de una compañía a otra, si bien muchos métodos incluyen cinco fases: investigación, análisis, diseño, puesta en operación y mantenimiento y revisión.

- **Ciclos de Vida asociados al desarrollo Tradicional**

  - ✓ **Ciclo de Vida Clásico o en Cascada**

En este modelo la evolución del producto software procede a través de una secuencia ordenada de transiciones de una fase a la siguiente según un orden lineal. Este modelo permite iteraciones durante el desarrollo, ya sea dentro de un mismo estado, ya sea de un estado hacia otro anterior, como muestran las flechas ascendentes. La mayor iteración se produce cuando una vez terminado el desarrollo y cuando se ha visto el software producido, se decide comenzar de nuevo y redefinir los requisitos del usuario.



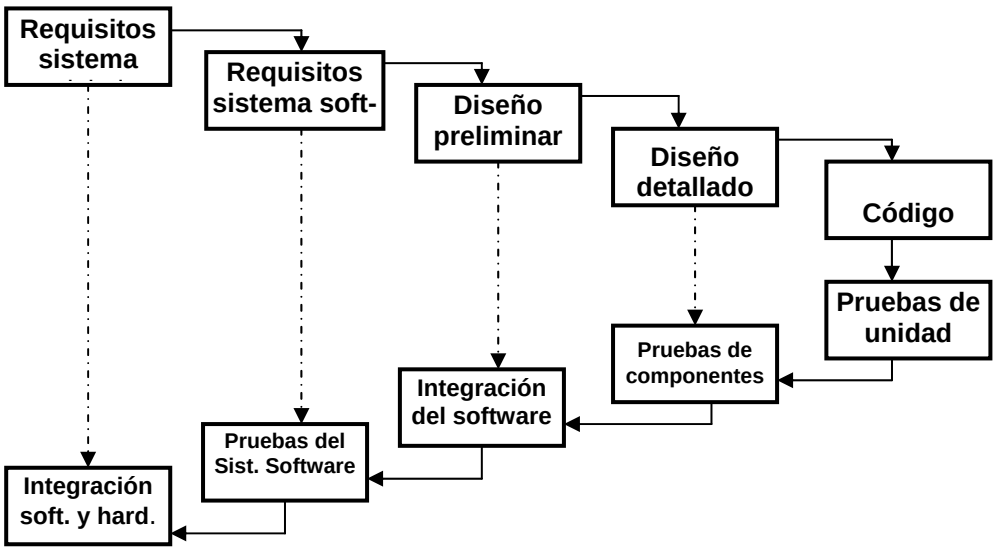
Modelo en cascada

Las principales ventajas y desventajas son:

Ventajas	Desventajas
Las etapas están organizadas de un modo lógico. Es decir, una etapa no puede llevarse a cabo hasta que se hayan tomado ciertas decisiones de más alto nivel, debe esperar hasta que esas decisiones estén tomadas.	El modelo en cascada asume que los requisitos de un sistema pueden ser congelados antes de comenzar el diseño. Esto, para sistemas totalmente nuevos, es poco realista.
Cada etapa incluye cierto proceso de revisión, y se necesita una aceptación del producto antes de que la salida de la etapa pueda usarse. Este ciclo de vida está organizado de modo que se pase el menor número de errores de una etapa a la siguiente.	Congelar los requisitos requiere seleccionar el hardware. La terminación de un gran proyecto puede llevar años. Dada la velocidad de obsolescencia de la tecnología es bastante probable que el software final utilice un hardware obsoleto.
El ciclo es iterativo. A pesar de que el flujo básico es de arriba hacia abajo, el ciclo de vida en cascada reconoce que los problemas encontrados en etapas inferiores afectan a las decisiones de las etapas superiores.	El punto más negro del ciclo de vida en cascada es que envía al cliente el primer producto solamente después de que se ha consumido el 99 % de los recursos para el desarrollo.

- ✓ **Ciclo de vida en Cascada alternativo o refinado**

Enfatiza la validación de los productos, y de algún modo el proceso de composición existente en la construcción de sistemas software.



Visión alternativa del ciclo de vida en cascada

El proceso de análisis o descomposición subyacente en la línea superior del modelo precedente consiste en los requisitos del sistema global que se dividen en requisitos del hardware y requisitos del software. El ensamblaje del producto final fluye justo en sentido contrario, se realiza dentro de un proceso de síntesis o composición.

2.2.1.2. Modelos de desarrollo por Fases

El enfoque de desarrollo por fases es una forma de reducir la repetición del trabajo en el proceso de desarrollo y proporcionar a los clientes algunas oportunidades para retrasar las decisiones en los requerimientos detallados hasta que adquieran cierta experiencia con el sistema.

- **Ciclos de Vida asociados al desarrollo por fases**

  - ✓ **Desarrollo Incremental o Iterativo**

En un proceso de desarrollo incremental, los clientes identifican, de forma somera, los servicios que proveerá el sistema. Entonces, se definen varios incrementos en donde cada uno proporciona un subconjunto de funcionalidad al sistema.

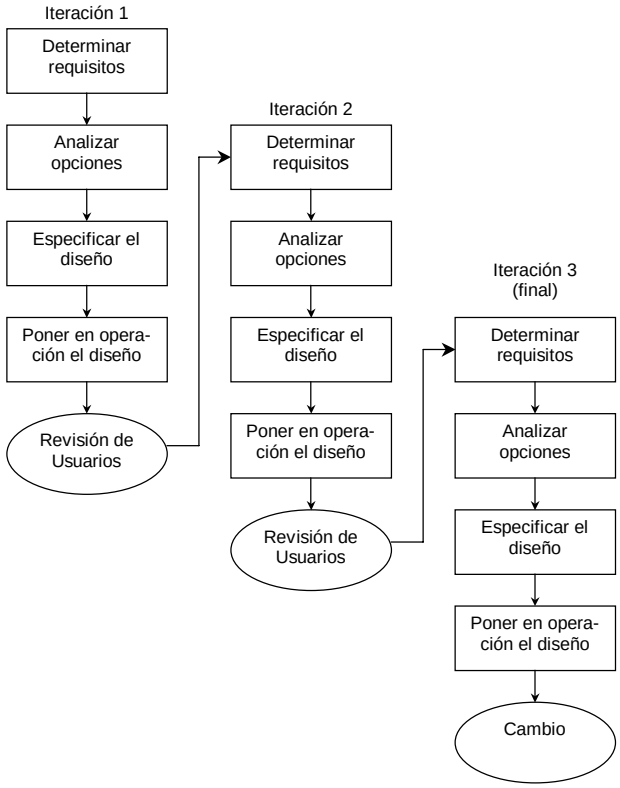
Una vez que un incremento se completa y entrega, los clientes pueden ponerlo en servicio. Esto significa utilizar parte de la funcionalidad del sistema. También experimentan con el sistema, lo cual les ayuda a clarificar sus requerimientos para los incrementos posteriores y para las últimas versiones del incremento actual.

No es necesario utilizar el mismo ciclo de vida para cada incremento. Se puede utilizar un ciclo de vida en cascada si los servicios en un incremento tienen una especificación bien definida. Si la especificación no es clara, se puede utilizar un ciclo de vida evolutivo.

La esencia de los procesos iterativos o incrementales es que la especificación se desarrolla junto con el software. Sin embargo, esto crea conflictos con el modelo de obtención de muchas organizaciones donde la especificación completa del sistema es parte del contrato para el desarrollo del mismo.

Las principales ventaja y desventajas son:

Ventajas	Desventajas
Los clientes no tienen que esperar hasta que el sistema completo se entregue para sacar provecho del él. El primer incremento satisface los requerimientos más críticos de tal forma que el software está disponible para su uso inmediato.	Los incrementos deben ser relativamente pequeños (no más de 20.000 líneas de código) y cada uno debe entregar alguna funcionalidad al sistema.
Los clientes pueden utilizar los incrementos iniciales como un prototipo para obtener experiencia sobre los requerimientos del los incrementos posteriores del sistema.	Puede ser difícil adaptar los requerimientos del cliente a incrementos de tamaño apropiado.
Existe un bajo riesgo de fallar en el proyecto total. Aunque algunos problemas se pueden encontrar en algunos incrementos, lo normal es que el sistema se entregue de forma satisfactoria al cliente.	Muchos de los sistemas requieren un conjunto de recursos que se utilizan en diferentes partes del sistema.
Puesto que los servicios de alta prioridad se entregan primero y los incrementos posteriores se integran a ellos, es inevitable que los servicios más importantes del sistema sean a los que les haga más pruebas.	Es difícil identificar los recursos comunes que requieren todos los incrementos.



✓ Desarrollo Rápido de Aplicaciones (RAD. Rappid aplication Development)

En el desarrollo rápido de aplicaciones (**RAD**) se usan herramientas, técnicas y métodos diseñados para **acelerar** el desarrollo de aplicaciones. Sus características son:

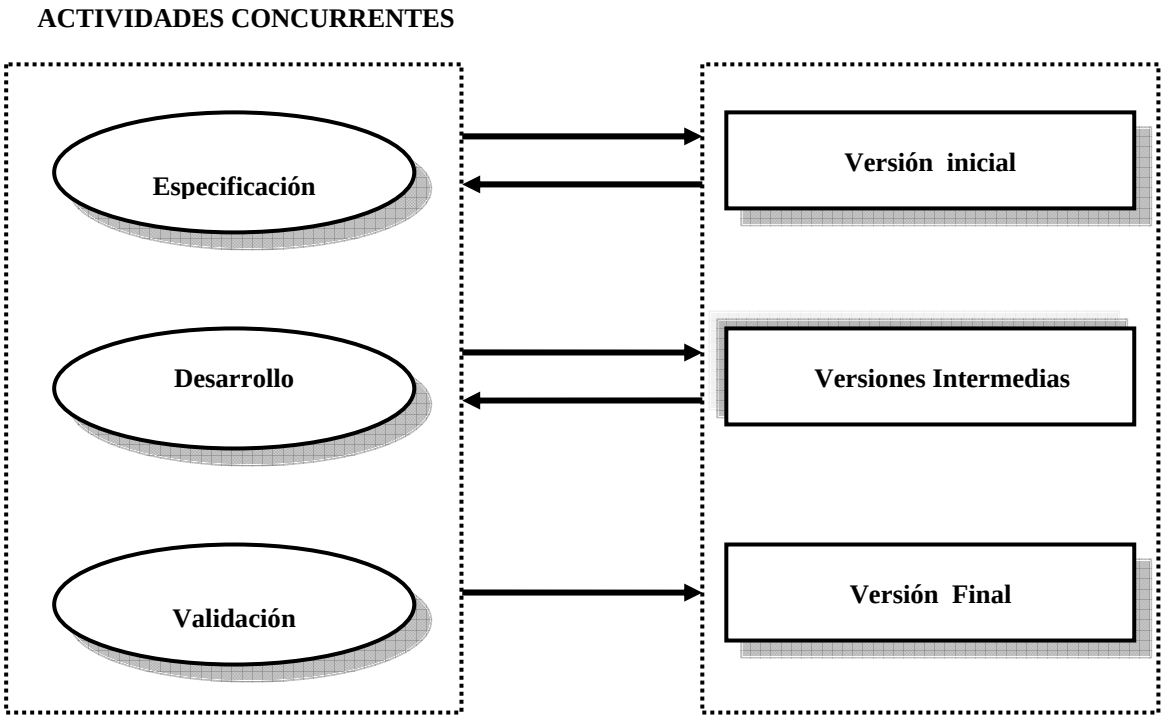
- Disminuye la documentación en papel.
- Automatiza la generación del código fuente del programa.
- Facilita la participación de los usuarios en las actividades de diseño y desarrollo.
- Se desarrollan sistemas completos en menos de seis meses.
- El objetivo final es acelerar el proceso.
- Las aplicaciones entran a la etapa de producción con mucha mayor prontitud que cuando se utilizan otros métodos.
- Usuarios y desarrolladores trabajan en conjunción como grupo mientras dura un proyecto.
- Promueve una toma de riesgos y de decisiones de grupo sanas, lo que genera mejores sistemas con fechas de entrega a menor plazo.
- Si el sistema en su totalidad es demasiado grande para completarlo en menos de seis meses, se puede dividir en subsistemas y entregar uno a la vez.
- Genera menor desperdicio de recursos, pues aun cuando exista un error grave en el sistema, sólo tiene que recrearse un subsistema.
- En general, es idóneo para sistemas de información administrativa y de apoyo a decisiones, y lo es menos para aplicaciones de procesamiento de transacciones.
- La participación de los beneficiarios y usuarios es mucho mayor que con otros métodos.
- Ello puede resultar problemático si los usuarios también tienen que desempeñar sus funciones habituales en la empresa.
- En virtud de la inversión de tiempo completo y las extremadas fechas límite de programas, el método RAD es de alta presión, y puede originar fácilmente agotamiento en los empleados.

Las ventajas y desventajas de este sistema son los siguientes:

Ventajas	Desventajas
Para proyectos idóneos, con este método se pone en operación una aplicación con más prontitud que con cualquier otro.	Este forma de trabajo intensa puede agotar a los desarrolladores del sistema y a otros participantes en el proyecto.
La documentación se genera como producto secundario de las tareas de realización del proyecto.	Este método requiere analistas de sistemas y usuarios conocedores de las herramientas de desarrollo y técnicas de RAD.
El RAD obliga al trabajo de grupo y a la interacción constante de los usuarios y beneficiarios.	El RAD absorbe un porcentaje de tiempo de los beneficiarios y usuarios mayor que con cualquier otro método.

2.2.1.3. Modelos de desarrollo Evolutivos

Estos modelos se basan en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios de los usuarios y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado. Más que tener actividades separadas de especificación, desarrollo y validación, éstas se llevan a cabo concurrentemente, y tienen retroalimentación rápida a lo largo del proceso.



Desarrollo Evolutivo

Ciclos de Vida asociados al desarrollo evolutivo

✓ Prototipos

Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y, de forma general, enterarse más acerca del problema y las posibles soluciones.

Los tipos de prototipos pueden ser:

Criterio	Tipo de Prototipo	Descripción
Según la funcionalidad	Operativo	Permite el acceso a archivos de datos reales y la edición de los datos de entrada, realiza los cálculos y comparaciones necesarios y produce una salida real.
	No Operativo	Es un modelo, posiblemente a escala. Se pueden desarrollar con mucho más rapidez que los operativos.
Según el modelo de uso	Desechable	Se usa para ayudar al cliente a identificar los requisitos de un nuevo sistema. El usuario, mediante el uso del prototipo, descubrirá esos aspectos o requisitos no captados. Todos los elementos del prototipo serán posteriormente desechados.
	Maqueta	Aporta al usuario ejemplo visual de entradas y salidas. La diferencia con el anterior es que en los prototipos desechables se utilizan datos reales, mientras que las maquetas son formatos encadenados de entrada y salida con datos simples estáticos.
	Evolutivo	Es un modelo de trabajo del sistema propuesto, fácilmente modificable y ampliable, que aporta a los usuarios una representación física de las partes claves del sistema antes de la implementación. Una vez definidos todos los requisitos, el prototipo evolucionará hacia el sistema final.

El ciclo de vida clásico queda modificado de la siguiente manera por la introducción del uso de prototipos:

1. Análisis preliminar y especificación de requisitos
2. Diseño, desarrollo e implementación del prototipo
3. Prueba del prototipo
4. Refinamiento iterativo del prototipo
5. Refinamiento de las especificaciones de requisitos
6. Diseño e implementación del sistema final

Las ventajas y desventajas del uso de prototipos son:

Ventajas	Desventajas
Los usuarios pueden probar el sistema y proporcionar retroalimentación constructiva durante su desarrollo.	Cada iteración se basa en la iteración previa y depura de manera adicional la solución. Ello dificulta rechazar la solución inicial por inade-

	cuada y empezar de nuevo. Así pues, la solución final sólo es incrementalmente mejor que la primera.
Puede producirse un prototipo operativo en semanas.	No se efectúan revisiones formales al final de cada fase. De tal suerte, es muy difícil controlar el alcance del prototipo y el proyecto parecería nunca terminar.
Los usuarios adoptan una actitud más positiva hacia la puesta en operación del sistema, a medida que observan cómo surge una solución que satisface sus necesidades.	Es frecuente que se carezca de documentación del sistema o que ésta sea incompleta, pues el método principal se centra en el desarrollo de los prototipos.
El uso de prototipos permite la detección temprana de errores y omisiones.	Pueden omitirse aspectos de respaldo, recuperación, rendimiento y seguridad del sistema, en la prisa por desarrollar el prototipo.

Para poder crear *prototipos rápidos* hay disponibles tres clases genéricas de métodos y herramientas:

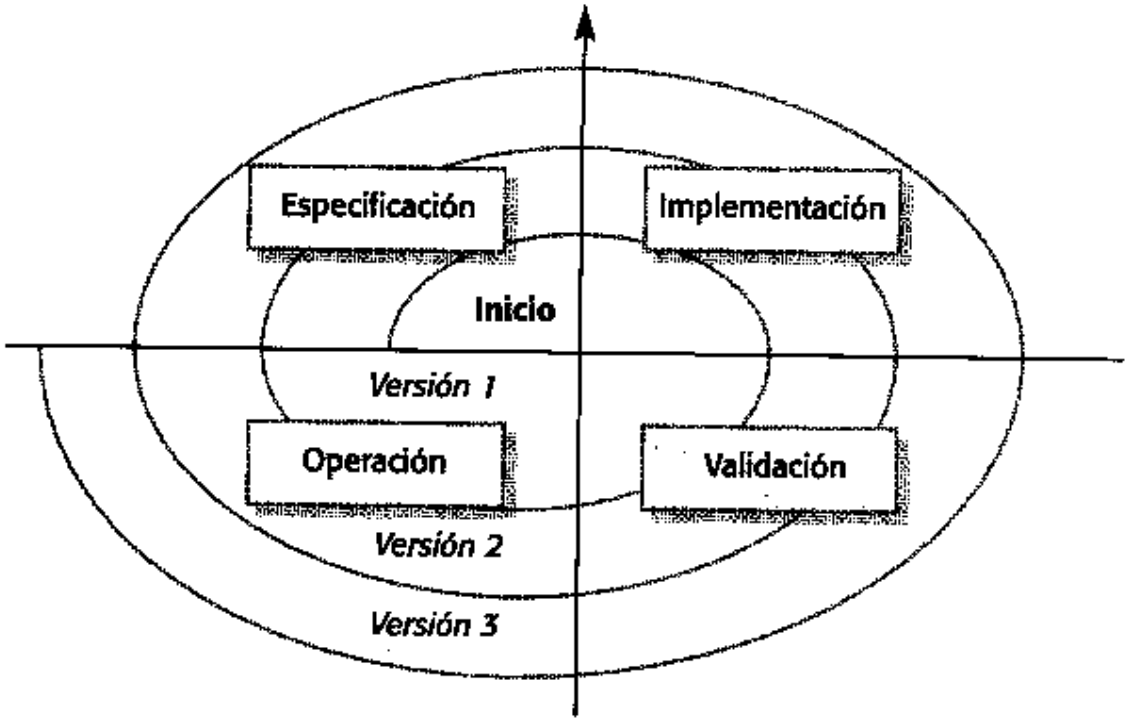
Método y herramienta	Características
Técnicas de Cuarta Generación (4GT)	Comprenden una amplia gama de lenguajes de consulta e informes de bases de datos, generadores de programas y aplicaciones y de otros lenguajes no procedimentales de muy alto nivel.
Componentes de software reutilizable	Significa ensamblar, más que construir, el prototipo mediante un conjunto de componentes software existente. La combinación de prototipos con la reutilización de componentes de programa sólo funcionará si se desarrolla un sistema bibliotecario de manera que los componentes existentes estén catalogados y puedan recogerse.
Especificaciones formales y entornos para prototipos.	Se han desarrollado varios lenguajes formales de especificación y herramientas como sustitutos de las técnicas de especificación con lenguaje natural.

- ✓ **Espiral**

Este ciclo más que representar el proceso del software como una sucesión de estados como una sucesión de actividades con retrospectiva de una actividad a otra, se representa como una espiral. Cada ciclo de la espiral se divide en cuatro sectores:

Sector	Descripción
Definición de Objetivos	En esta fase se definen los objetivos específicos. Se identifican las restricciones del proceso y del producto. Se identifican los ***riesgos*** del proyecto. Dependiendo de estos ***riesgos,*** se planean estrategias

	alternativas.
Evaluación y reducción de riesgos	Se lleva a cabo un análisis detallado de cada uno de los ***riesgos*** del proyecto. Se definen los pasos para reducir dichos ***riesgos.***
Desarrollo y validación	Después de la evaluación de ***riesgos,*** se elige un ciclo o varios para el desarrollo del sistema.
Planeación	El proyecto se revisa y se toma la decisión de si se debe continuar con un ciclo posterior de la espiral.



La diferencia importante entre el modelo en espiral y los otros modelos es ***la consideración explicita del riesgo*** en el modelo en espiral. Los ***riesgos*** conducen a problemas como calendarización y excesos en los costos, por lo tanto, la disminución de ***riesgos*** es una actividad muy importante en la administración del proyecto.

Las ventajas de este modelo son:

- Su rango de opciones permite utilizar los modelos del proceso de construcción de software tradicionales dentro del ciclo de vida.
- Su orientación al ***riesgo*** evita muchas dificultades.
- Presta atención a las opciones que permiten la reutilización de software existente.
- Se centra en la eliminación de errores y alternativas poco atractivas.
- No establece una diferenciación entre desarrollo de software y mantenimiento del sistema.
- Proporciona un marco estable para desarrollos integrados hardware-software.

2.2.2. Desarrollo Orientado a Objetos

Se vive en un mundo de objetos. Estos objetos existen en la naturaleza, en entidades hechas por el hombre, en los negocios y en los productos que usamos. Ellos pueden ser clasificados, descritos,

organizados, combinados, manipulados y creados. Por esto no es sorprendente que se proponga una visión orientada a objetos para la creación de software de computadora, una abstracción que modela el mundo de forma tal que nos ayuda a entenderlo y gobernarlo mejor.

La idea principal, al hablar de abstracción, es la de moverse en distintos niveles de complejidad, lo cual dará distintos niveles de abstracción. Es así como a medida que se vayan incorporando detalles más finos, se estará ingresando a niveles de **mayor complejidad** y al mismo tiempo se tendrá un nivel de **abstracción menor**. La ***abstracción de datos*** permite no preocuparse de los detalles no esenciales. Existe en casi todos los lenguajes de programación. Las estructuras de datos y los tipos de datos son un ejemplo de abstracción. Los procedimientos y funciones son otro ejemplo.

El paradigma orientado a objetos

Como primera medida, es importante saber qué es un paradigma. Los paradigmas son un marco o perspectiva bajo la cual se analizan los problemas y se trata de resolverlos. Se puede ver a un paradigma como un modelo que sirve de norma, brinda marcos de referencia que dice qué es lo que se puede hacer y con qué elementos se cuenta.

El paradigma **orientado a objetos** (**OO**) brinda un marco para la construcción de programas y establece a los **objetos** como únicos elementos del paradigma, es decir, que todo es un **objeto** o puede ser representado como tal.

Elementos básicos de la Programación Orientada a Objetos (POO)

✓ Clases

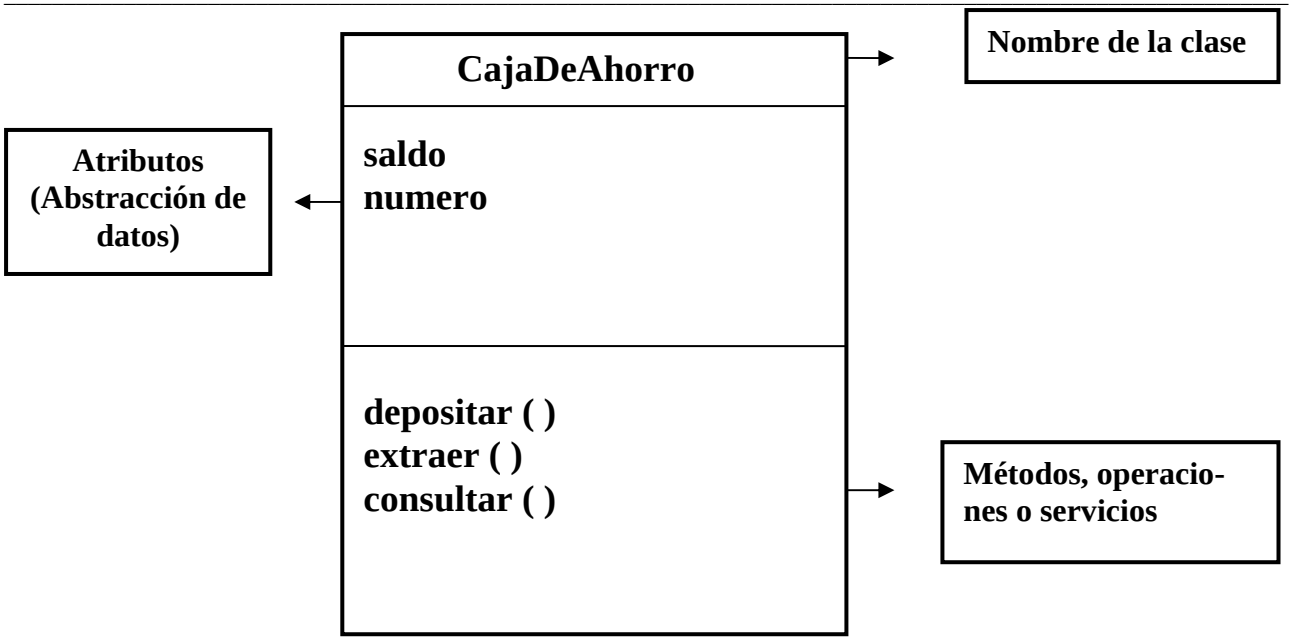
Una clase permite describir en un solo lugar el comportamiento genérico de un conjunto de **objetos**. Las clases permiten pensar en un nivel de abstracción más alto. Una clase puede pensarse como un **molde** para un tipo específico de **objeto**. Molde en el mismo sentido que al referirse al mecanismo de fabricación, es decir, entendiendo a la clase como un medio para definir la forma de los **objetos.**

Una clase es responsable de crear otros **objetos**, a los que se los llama **instancias.** Las clases se comportan como fábricas, dado que crean nuevos **objetos** a partir de un molde. Entonces se tiene que las clases cumplen tres roles:

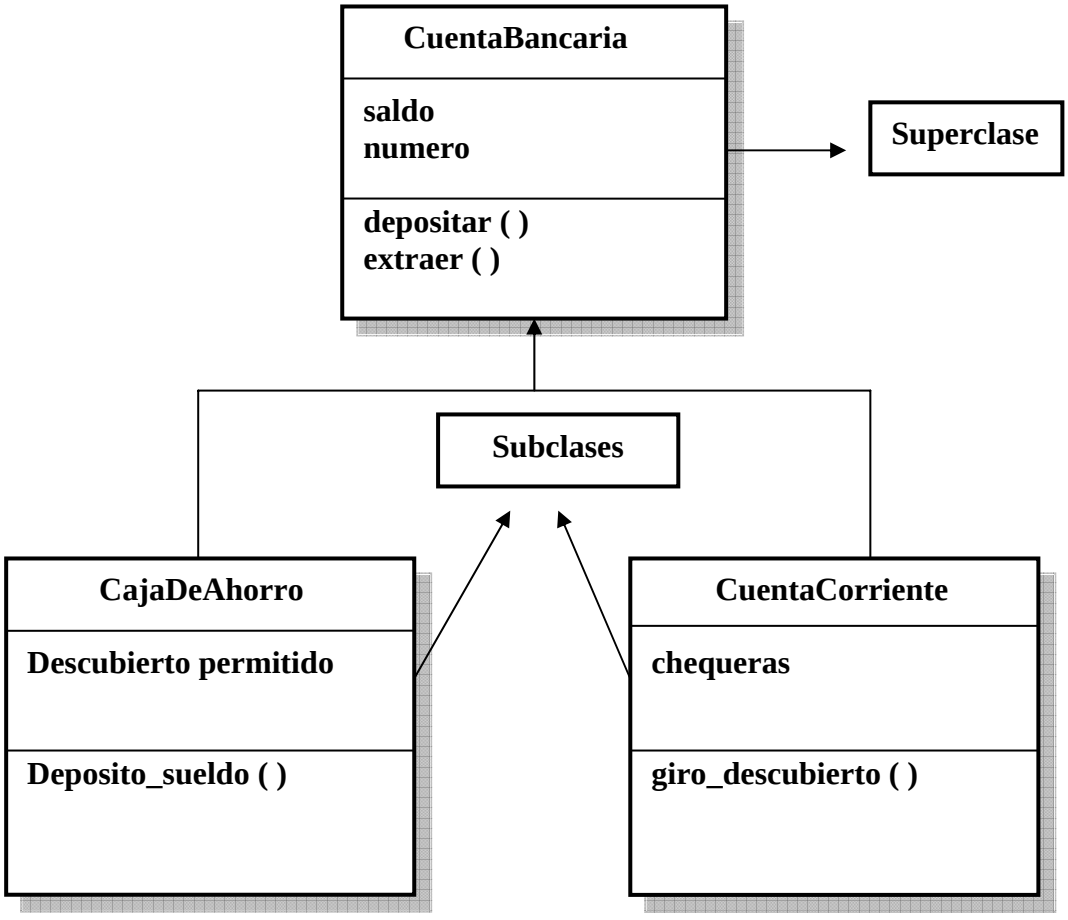
- Agrupan el comportamiento común de las **instancias (objetos).**
- Definen las formas de estas **instancias.**
- Crean **objetos** que serán esas **instancias.**

En un sistema basado en clases todo **objeto** debe ser **instancia** de alguna clase. Pero dado que todo es un **objeto** en el paradigma, entonces ***las clases también son objetos.***

Las **abstracciones de datos (atributos)** que describen la clase están encerradas por una "muralla" de abstracciones procedimentales (llamadas **operaciones, métodos o servicios**) capaces de manipular los datos de alguna manera. La única forma de alcanzar los atributos (y operar sobre ellos) es ir a través de alguno de los métodos que forman la muralla. Por lo tanto, la clase encapsula datos (dentro de la muralla) y el proceso que manipula los datos (los métodos que componen la muralla). Esto posibilita el ocultamiento de información y reduce el impacto de efectos colaterales asociados a cambios. Como estos métodos tienden a manipular un número limitado de atributos, esto es una cohesión, y como la comunicación ocurre sólo a través de los métodos que encierra la "muralla", la clase tiende a un acoplamiento con otros elementos del sistema. Todas las características del diseño conducen a software de alta calidad.



Una ***superclase*** es una colección de clases y una ***subclase*** es una instancia de una clase. Estas definiciones implican la existencia de una ***jerarquía de clases*** en la cual los atributos y operaciones de la superclase son heredados por subclases que pueden añadir, cada una de ellas, atributos "privados" y métodos.



✓ Objetos

Un **objeto** es una abstracción que representa a una entidad del dominio del problema (perteneciente a la realidad) ya sea tangible o no. Un **objeto** captura la esencia de dicha entidad en un determinado contexto. Los **objetos** son los elementos computacionales que se utilizan para construir programas, son los elementos primarios y únicos del paradigma. Es decir, ***todo es un objeto.***

Un **objeto** presenta tres características fundamentales:

- *Comportamiento bien determinado:* dicta **qué** hace o **qué** es capaz de hacer.
- *Presenta una implementación para ese comportamiento:* define **cómo** hace el **objeto** lo que sabe hacer.
- *Posee una identidad:* permite *distinguir* un **objeto** de otro.

Los **objetos** son **instancias** de una clase. Cuando creamos una **instancia** tenemos que especificar la clase a partir de la cual se creará. Esta acción de crear un objeto a partir de una clase se llama **instanciar**.

Por ejemplo, un **objeto** de la clase fracción es por ejemplo 3/5. El concepto o definición de fracción sería la clase, pero cuando ya estamos hablando de una fracción en concreto 4/7, 8/1000 o cualquier otra, la llamamos **objeto**.

✓ Atributos

Los atributos están asociados a clases y **objetos**. Las entidades de la vida real están a menudo descritas con palabras que indican características estables. La mayoría de los objetos físicos tienen características tales como forma, peso, color y tipo de material. Las personas tienen características incluyendo fecha de nacimiento, padres, nombres y color de los ojos. Los atributos van a tomar distintos valores a través de la vida del objeto. Una característica puede verse como una relación binaria entre una clase y cierto dominio.

La relación binaria anteriormente señalada implica que un atributo puede tomar un valor definido por un dominio enumerado. En la mayoría de los casos, un dominio es simplemente un conjunto de valores específicos.

Por ejemplo, la *clase* Automotores tiene un atributo *color*. El dominio de valores de *color* es (blanco, negro, plata, gris, azul, rojo, amarillo, verde). En situaciones más complejas, el dominio puede ser un conjunto de clases: un atributo *motor* que abarca los siguientes valores de dominio; (valor, 15 opción económica, 16 válvulas opción deportiva, 24 válvulas opción deportiva, 32 válvulas opción de lujo). Cada una de las opciones indicadas tiene un conjunto de atributos específicos de ella.

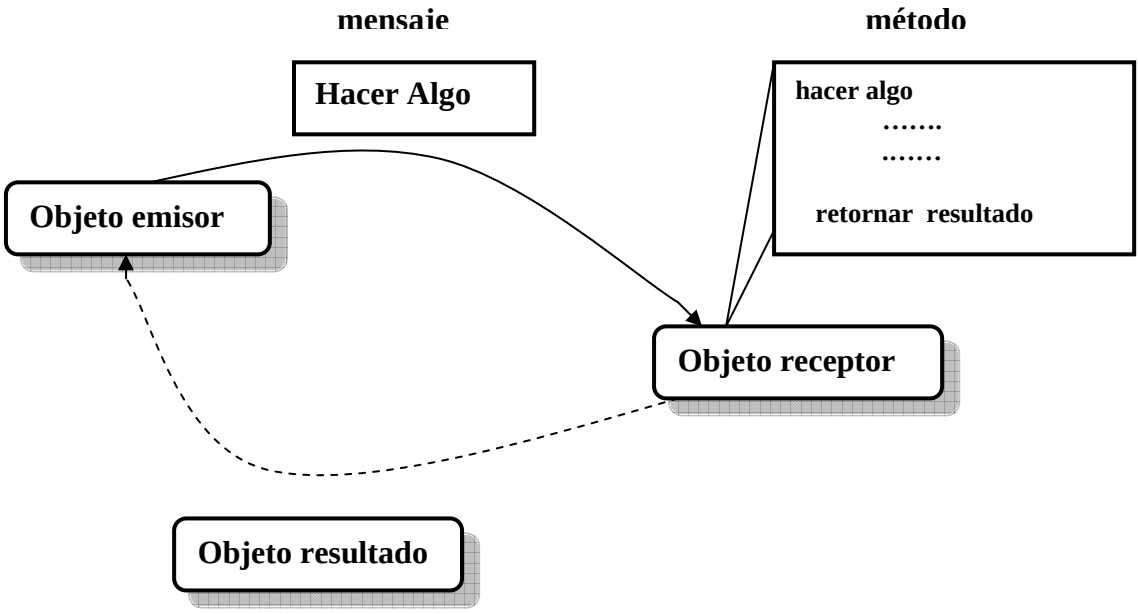
✓ Métodos, operaciones o servicios

Un **objeto** encapsula datos (representados como una colección de *atributos*) y los algoritmos que procesan estos datos. Estos algoritmos son llamados *operaciones, métodos o servicios* y pueden ser vistos como módulos en un sentido convencional. Cada una de las operaciones encapsuladas por un objeto proporciona una representación de uno de los comportamientos del objeto. Por ejemplo, la operación *DeterminarColor* para el objeto **automóvil** (**instancia** de la clase Automotores) extraerá el color almacenado en el atributo *color*. La consecuencia de la existencia de esta operación es que la clase Automotores ha sido diseñada para recibir un estímulo que requiere el color de una instancia particular de una clase.

- ✓ **Mensajes**

Una colaboración consiste en el envío y la recepción de un *mensaje* en la que se ven involucrados dos **objetos.** Uno de ellos será el encargado de enviar el *mensaje*, es decir, cumplirá el rol de emisor, mientras que el restante estará cumpliendo el rol de receptor del *mensaje.*

Al enviarle un *mensaje* al **objeto,** éste responde activando el *método* asociado al *mensaje*, siempre y cuando éste exista. Un *mensaje* es enviado a un **objeto** por otro **objeto** y como resultado del envío de un *mensaje* siempre se retorna un **objeto.**



Una operación dentro de un **objeto** emisor genera un mensaje de la forma:

Mensaje: [destino, operación, parámetros]

donde *destino* define el **objeto** receptor (el cual es estimulado por el mensaje), *operación* se refiere al método que recibe el mensaje y *parámetros* proporciona información requerida para el éxito de la operación.

Características del Desarrollo Orientado a Objetos

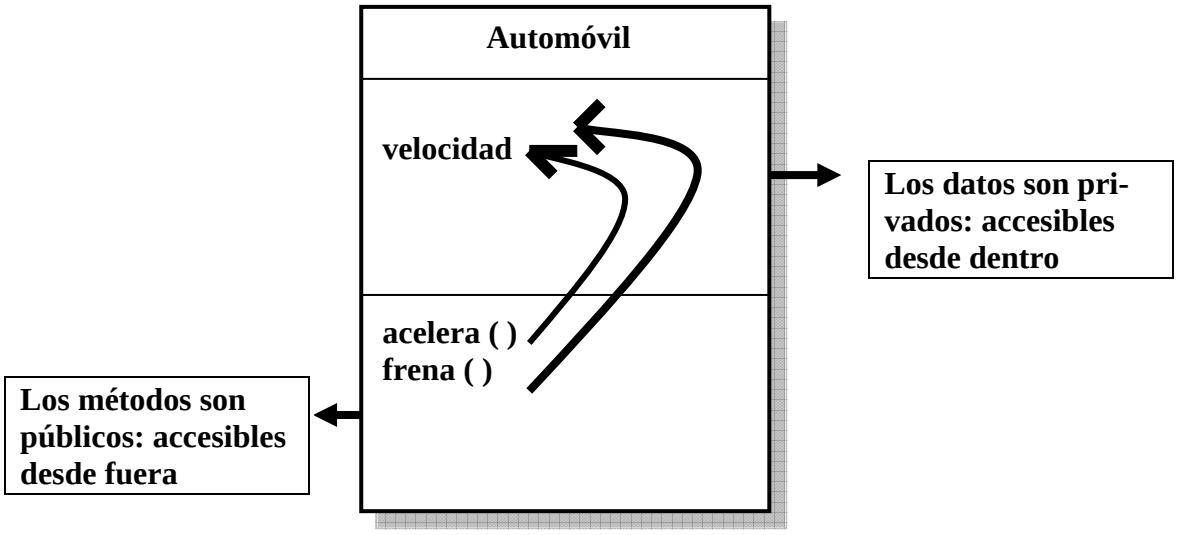
- ✓ **Encapsulamiento**

El *encapsulamiento* también llamado "ocultamiento de la información", asegura que los **objetos** no pueden cambiar el estado interno de otros **objetos** de manera inesperada. Solamente los propios métodos internos del **objeto** pueden acceder a su estado. Cada tipo de **objeto** expone una interfaz a otros **objetos** que especifica cómo otros **objetos** pueden interactuar con él. Algunos lenguajes permiten un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción.

Los **objetos** deben comunicarse solamente a través de su propio protocolo de modo que la responsabilidad en administrar el comportamiento sea sólo del **objeto** que implementa dicho comportamiento.

La utilidad del *encapsulamiento* es facilitar el manejo de la complejidad, ya que las Clases se ven como cajas negras donde sólo se conoce el comportamiento pero no los detalles internos, y esto es conveniente porque ***sólo interesará conocer qué hace la Clase pero no será necesario saber cómo lo hace.***

Esta característica es la que denota la capacidad del **objeto** de responder a peticiones a través de sus métodos sin la necesidad de exponer los medios utilizados para llegar a brindar estos resultados.

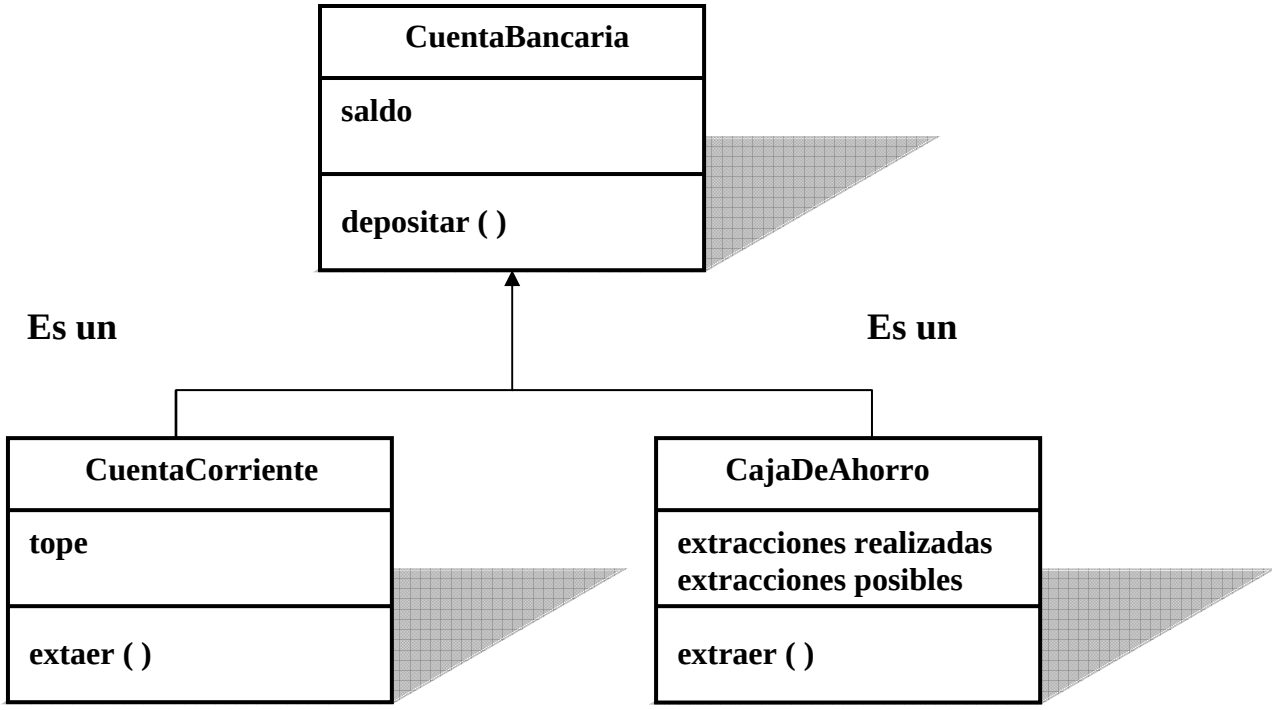


- ✓ **Herencia**

La *herencia* es uno de los conceptos más cruciales en la **Programación Orientada a Objetos**. La *herencia* básicamente consiste en que una clase puede hacer heredar sus atributos y métodos. Esto significa que una subclase, aparte de los atributos y métodos propios, tiene incorporados los atributos y métodos ***heredados*** de la superclase. De esta manera se crea una jerarquía de herencia.

La relación "*es un*" significa que la clase hija (o heredera), es, además, lo mismo que su padre. Es decir, un auto "*es un*" transporte, un caballo "*es un*" animal, etc. Cualquier cambio en los datos u operaciones contenidas dentro de una superclase se hereda inmediatamente por todas las subclases que se derivan de la superclase. Debido a esto, la jerarquía de clases se convierte en un mecanismo a través del cual los cambios (a altos niveles) pueden propagarse inmediatamente a través de todo el sistema.

En algunos casos, es tentador heredar algunos atributos y operaciones de una clase y otros de otra clase. Esta acción se llama *herencia múltiple* y es controvertida. En general, la herencia múltiple complica la jerarquía de clases, dificulta el entendimiento del código y crea problemas potenciales en el control de la configuración. Como las secuencias de herencia múltiple son más difíciles de seguir, los cambios en la definición de una clase que reside en la parte superior de la jerarquía pueden tener impactos no deseados originalmente en las clases definidas en zonas inferiores de la arquitectura.



- ✓ **Polimorfismo**

Se dice que dos o más **objetos** son *polimórficos* respecto de un mensaje, si todos ellos pueden responder ese mensaje, aún cuando cada uno lo haga de un modo diferente. *En* **programación orientada a objetos** *se denomina* ***polimorfismo*** *a la capacidad que tienen los objetos de una* clase *de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación.*

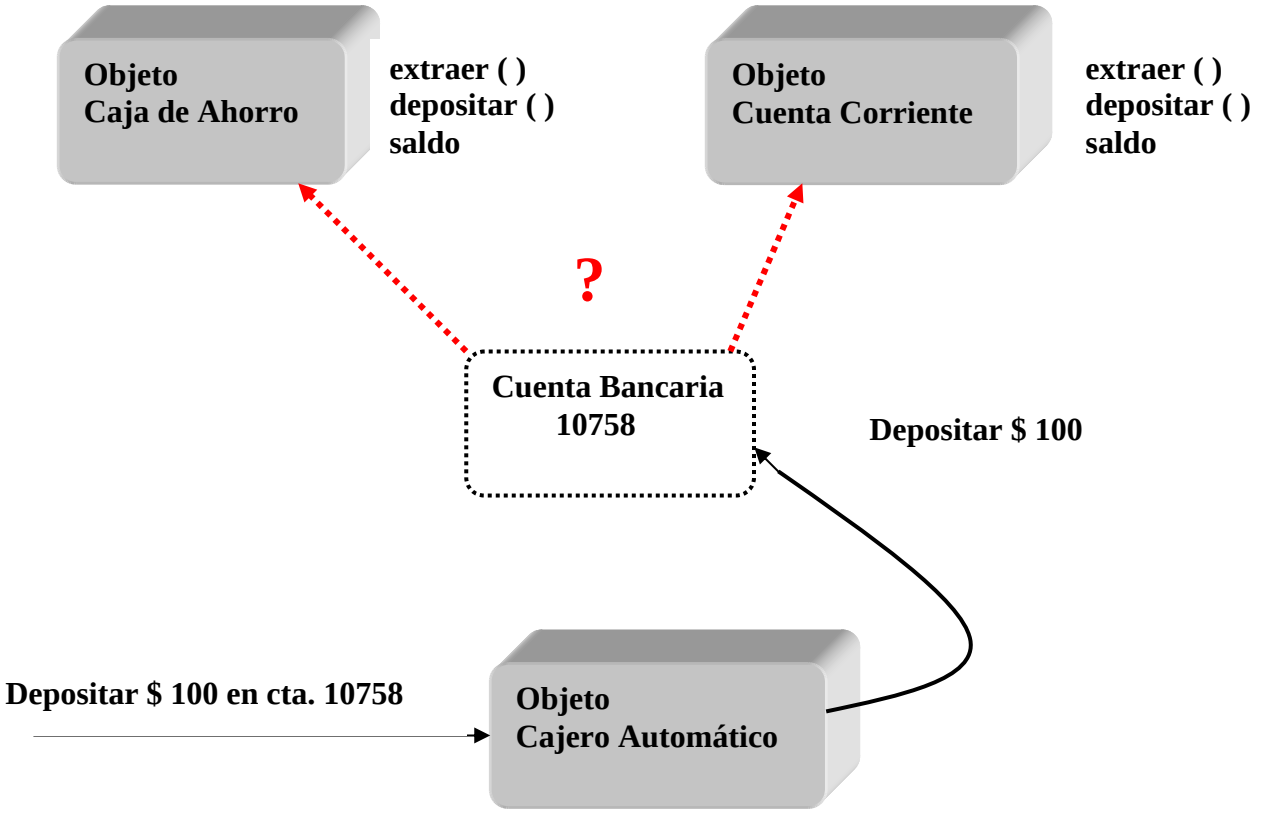
Por ejemplo, si se tienen distintos instrumentos de música (guitarra, flauta y piano), todos ellos saben tocar notas musicales, pero cada uno de ellos interpretará las notas a su manera, generando distintas melodías. Es decir, en términos técnicos, todos los instrumentos entienden el mismo mensaje (tocarNota ( )) y cada uno de ellos proveerá una implementación diferente para responder a ese mensaje, o sea, implementarán métodos distintos. De esta manera, cualquier otro **objeto** podrá enviarle el mensaje tocarNota ( ) a distintos instrumentos, sin preocuparse sobre la naturaleza de cada uno de ellos, simplemente confiará en que cada instrumento sabrá interpretar la nota.

Como consecuencia del *polimorfismo*, entonces se tiene que un **objeto no necesita saber a quién** le está enviando un mensaje, sólo espera que el objeto receptor del mensaje, sepa resolverlo de alguna manera.

Para entender mejor este concepto y su utilidad, considérese el siguiente ejemplo:

En un banco se tiene dos clases de cuentas bancarias: cajas de ahorro y cuentas corrientes. En las cuentas bancarias se puede depositar y extraer un monto de dinero y además consultar su saldo. Un usuario del banco puede acceder a su cuenta por medio de un cajero automático, para realizar cualquiera de esas operaciones sabiendo el número de cuenta con la cual quiere operar. Supongamos que queremos depositar $ 100 en la cuenta número 10758, lo que se hace es enviarle el mensaje depositar $ 100 en Cuenta 10758 al cajero automático. Este objeto sabrá de alguna manera obtener dicha cuenta y depositará el dinero.

En ningún momento se necesitará saber la naturaleza de la cuenta en la cual se va a depositar (si es de ahorro o corriente), sólo bastará con esperar que sepa depositar.



Las ventajas del *polimorfismo* son:

- Permite generar código más genérico.
- Se piensa a nivel protocolo y se olvida la implementación.
- Reduce el conocimiento entre **objetos**.
- Reduce la interdependencia entre **objetos**, logrando el desacoplamiento.
- No hay que preocuparse en detalles de implementación.
- Se pueden intercambiar los **objetos** a bajo costo, dinámicamente y sin desestabilizar el sistema.
- Favorece la reutilización.
- Permite el uso de un mismo tipo de *objeto* en dominios de problema o modelos diferentes.
- Trae múltiples ventajas, sobre todo a mediano y largo plazo.

2.3. DOCUMENTACIÓN

Un sistema de software no servirá de mucho a menos que las personas puedan aprender a usarlo y mantenerlo. Por tanto, ***la documentación es una parte importante del software*** y, a su vez, ***su desarrollo es un tema importante en la ingeniería de software***.

Tipo de Documentación	Propósito	Características
Documentación del usuario	Ser leída por el usuario	▪ Tiende a no ser técnica ▪ Hace que el paquete sea accesible ▪ Adopta la forma de manual: una parte presenta una introducción a las funciones más utilizadas, una sección de cómo instalar el software y una sección que describe los detalles de cada función del software ▪ Está disponible en forma de libro, pero en muchos casos se proporciona como un archivo almacenado en el mismo medio que el software
Documentación del sistema	Describir el software en sí para poder mantener el sistema en una etapa posterior de su ciclo de vida	▪ Es más técnica que la del usuario ▪ En el pasado, consistía en los programas fuente finales y algunas explicaciones a grandes rasgos ▪ Esta documentación informal simplemente ya no es aceptable para los grandes sistemas de software de la actualidad ▪ Actualmente la documentación de sistemas comienza con el desarrollo de las especificaciones originales del sistema y continúa durante todo el ciclo de vida del software ▪ Consiste en todos los documentos que se prepararon durante el desarrollo del software

Algo de primordial importancia es la versión fuente de todos los programas del sistema. Es importante que dichos programas se presenten en un formato comprensible, y es por esto que los ingenieros de software recomiendan el empleo de lenguajes de programación de alto nivel bien diseñados, el uso de enunciados de comentario para anotar los programas y un diseño modular que permita presentar cada módulo como una unidad coherente.

El hecho de que el desarrollo de la documentación del sistema sea un proceso continuo origina un conflicto entre las metas de la ingeniería de software y la naturaleza humana. El problema en este caso es la tentación de realizar sobre la marcha cambios en el diseño del sistema sin regresar a los documentos de diseño iniciales y actualizados. En consecuencia, es grande la probabilidad de que estos documentos sean incorrectos y su empleo en la documentación final sea engañoso.

2.4. OBTENCIÓN DE REQUERIMIENTOS DEL SISTEMA

Obtener información quizá no sea el aspecto más interesante de un estudio de sistemas. Dar la consideración apropiada para desarrollar entrevistas, formular cuestionarios y observar comportamientos es trabajo, un trabajo desafiante.

Los distintos tipos de obtención de datos se detallan a continuación.

✓ La encuesta

El método de *encuesta*, consiste en una combinación de técnicas que se han desarrollado en la investigación de diferentes y variadas ciencias. Esta combinación incluye procedimientos como la entrevista, el cuestionario, la formación de escalas, el análisis estadístico de datos, entre otros.

La *encuesta* no es un método específico, se aplica al estudio e investigación en muchos campos del conocimiento, y es empleada en aquellos casos en los que la información que se necesita, no puede hallarse u obtenerse a través de otras fuentes.

Este método tiene la particularidad de depender de que las personas elegidas para ser entrevistadas ***puedan*** (no se requiera métodos especiales de medición), y ***quieran*** (no incrimine o reprima) ***proporcionar la información*** que les es requerida; no obstante si el entrevistador procede con habilidad y tacto la generalidad de los casos indica una gran tolerancia a las preguntas realizadas aún en las que posean carácter personal.

Las *encuestas* varían fundamentalmente en su alcance, diseño y contenido, debiéndose determinar dichas características a partir de los objetivos básicos.

El universo de la *encuesta* permite definir el tipo de población sobre el que se realizará la encuesta. Pueden ser universos de las encuestas la población nacional, áreas geográficas dentro de un país si la encuesta requiriera un análisis regional especial, o la representación de provincias, distritos o ciudades específicas.

Los tipos de *encuesta* son:

Tipo de encuesta	Características
Sección transversal no ponderada	▪ Se efectúa una única vez en cada investigación ▪ Permite determinar las características de una población en un momento particular
Sección transversal ponderada	▪ Constituye una variación del anterior ▪ Supone una sobreestimación deliberada de algún subgrupo del universo que tiene una importancia particular para los objetivos de la encuesta
Muestras contrastantes	▪ Consiste en obtener muestras de subgrupos que están en posiciones extremas respecto de la variable más importante ▪ Existe el peligro de suponer que una diferencia entre extremos refleja una relación lineal
Secciones transversales sucesivas	▪ Utilizado para estudios del cambio ▪ Requieren mediciones en diferentes puntos del tiempo ▪ Consiste en la repetición de la muestra en la misma población
Entrevistas reiteradas	▪ Se utiliza cuando es necesario observar las actividades o actitudes de los mismos individuos a través de un período determinado de tiempo ▪ Pueden llamarse también "paneles"

Para la realización de una encuesta se deberá planificar una serie de actividades que abarcan desde el planeamiento hasta la elaboración del informe final.

Las diferentes actividades a realizar son las siguientes:

- ***Establecimiento de objetivos generales***. Definición en términos amplios del área general y alcance del proyecto.
- ***Objetivos específicos:*** Enunciado de todos los datos que deben reunirse y las hipótesis que deben verificarse a través de la encuesta.

- ***Muestra*** En este punto se tomarán dos decisiones: Universo de la encuesta y en segundo término el tamaño y diseño de la muestra que debe extraerse.
- ***Cuestionario*** Se determina el medio por el que se tomará contacto con la muestra (entrevista personal, telefónica o por correo) y la preparación del cuestionario.
- ***Trabajo de campo*** Si las entrevistas serán personalmente se deberá instruir a los encuestadores tanto en los procedimientos generales de la entrevistas, como en los aspectos particulares de la encuesta. Debe proporcionárseles un manual de instrucciones que explique los objetivos del estudio y el significado de cada pregunta.
- ***Análisis de contenido*** Los datos obtenidos en la encuesta deberán ser transcriptos cuidadosamente en tablas de modo sencillo.
- ***Análisis e informe*** Los datos finalmente obtenidos y analizados permitirán la generación de un informe con los resultados de la encuesta, incluyendo gráficos o tablas comparativas que permitan validar las afirmaciones del informe.

✓ La entrevista

La forma más usual para realizar una encuesta, suele ser la entrevista en vivo, frente a frente con la persona especializada, si bien existen otros métodos que suelen ayudar en casos en los que este encuentro sea imposible, como por ejemplo el envío de un cuestionario escrito que el encuestado deberá completar y devolver.

En el planeamiento de una *entrevista* los pasos principales de la preparación son:

- *Establecer los objetivos de la entrevista:*
- *Decidir a quien entrevistar*
- *Preparar a la persona a entrevistar*
- *Decidir los tipos de preguntas y la estructura.*

Los tipos de preguntas de una *entrevista* pueden ser:

Tipo de pregunta	Características	Ventajas	Desventajas
Abiertas	Preguntas del tipo "¿Qué piensa acerca de ...?" o "¿Explíqueme cómo ...?"	▪ Comodidad del entrevistado ▪ Permiten detectar a través del vocabulario del entrevistado su nivel de educación, actitudes, etc. ▪ Detalles en abundancia ▪ Permite espontaneidad	▪ Puede llevar a perder el control de la entrevista ▪ Obtener demasiados detalles superfluos ▪ Dar la impresión de no tener objetivos para la entrevista
Cerradas	Enunciados de preguntas que dejan espacio para respuestas concretas en un número finito, o con opciones a elegir entre varias o con opciones de elección por verdadero/falso	▪ Ahorro de tiempo ▪ Facilitación de la comparación de las entrevistas ▪ Mantenimiento del control de la entrevista	▪ Pueden llevar a aburrir al entrevistado ▪ No llegar a grandes detalles

Si bien puede parecer una tarea sencilla, la realización de entrevistas encierran ciertas dificultades que se tendrán que tener en cuenta para la realización exitosa de la misma.

Algunos puntos importantes a tener en cuenta son:

- *Entrevistar a las personas correctas y en el momento correcto.*
- *Hacer las preguntas correctas*.
- Crear una atmósfera agradable y de confianza.

Las entrevistas pueden variar en su estructura de acuerdo con el tipo de estudio que se está realizando, y pueden básicamente ser:

- *Entrevistas no estructuradas:* se van realizando preguntas dentro del contexto general de una conversación
- *Entrevistas estructuradas:* la formulación de las preguntas tendrá un carácter más metódico

Dentro de las *entrevistas* estructuradas existen fundamentalmente tres tipos de estructuras:

- ***Embudo:*** comienzan con preguntas de tipo general y a medida que se va avanzando en la entrevista se pasa a preguntas de tipo más específicas.
- ***Pirámide:*** de preguntas específicas pasan a preguntas generales.
- ***Diamante o rombo:*** combinan las dos anteriores.

Existen algunas recomendaciones de cómo formular las preguntas:

- ✓ Usar el cuestionario de manera informal.
- ✓ Las preguntas deben ser formuladas exactamente como están redactadas en el cuestionario y una sola vez.
- ✓ Las preguntas deben ser formuladas en el mismo orden en que están presentadas en el formulario.
- ✓ Dar a la persona entrevistada el tiempo suficiente para pensar en sus respuestas.
- ✓ No dar por respondida una pregunta con respuestas que se derivan de otras.
- ✓ Es conveniente usar frases de transición (Bueno..., Veamos..., ahora bien, ¿le parece que sigamos con...?).
- ✓ Debe dejarse constancia escrita de los cambios introducidos eventualmente en el cuestionario.
- ✓ Hacer breves comentarios que ayuden a mantener la comunicación.
- ✓ Si la entrevista es no estructurada, se debe preparar un *esquema o una relación de preguntas*.

Ventajas de las entrevistas	Desventajas de las entrevistas
Es una técnica eficaz para obtener datos relevantes	Limitaciones de la expresión oral
La información obtenida es susceptible de cuantificación y tratamiento estadístico para una más rigurosa elaboración de los datos recogidos	Otorgar igual validez a todas las respuestas prescindiendo de quien responde
	Carácter estático de la realidad que capta la entrevista

- ✓ **Cuestionarios**

Todas las consideraciones realizadas para la entrevista son válidas para los cuestionarios enviados por correo. La diferencia fundamental entre éstos es que en la segunda modalidad (Cuestionario enviado por correo), no existe la presencia del encuestador lo que deriva en una serie de ventajas y desventajas.

El envío postal del cuestionario requiere ciertas precauciones: la distribución debe ser hecha en conjunto y dentro de un lapso relativamente breve. Se debe indicar el plazo para que sea devuelto.

Su utilización depende de la índole de la investigación:

a) *Según el tipo de información que se necesita:* Cuando se necesitan cuerpos muy extensos de datos, cuando se necesita profundidad en la investigación, NO CONVIENE USAR este procedimiento
b) *Tipo de encuestado al que llega el cuestionario* (alfabetización)
c) *Accesibilidad de los que responden* Este procedimiento es muy útil cuando los encuestados están muy espaciados geográficamente.
d) *Precisión de las hipótesis* lo que supone una considerable cantidad de trabajo exploratorio y de pruebas previas a la utilización del instrumento, para reducirlo al número de preguntas absolutamente necesarias.

Ventajas de cuestionarios	Desventaja de cuestionarios
▪ Menor costo	▪ Alto riesgo de cuestionarios no respondidos
▪ Se pueden abarcar mayores áreas geográficas	▪ Exclusión de personas que no saben leer y escribir
▪ Menor tiempo para llegar a la misma cantidad de personas	▪ Imposibilidad de ayuda complementaria
▪ Mayor libertad en las respuestas	▪ Imposibilidad de pedir aclaraciones
▪ Menor riesgo de distorsiones provenientes de la influencia del encuestador	▪ Recepción tardía de muchos cuestionarios
▪ Permite mayor reflexión en los sujetos encuestados	

✓ Observación

La observación permite al analista ganar información que no se puede obtener por otras técnicas. Por medio de la *observación* el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades. Este método es más útil cuando el analista necesita observar, por un lado, la forma en que se manejan los documentos y se llevan a cabo los procesos y, por otro, si se siguen todos los pasos especificados. Los observadores experimentados saben qué buscar y cómo evaluar la significancia de lo que observan.

3. LAS ORGANIZACIONES COMO SISTEMAS

La definición de las organizaciones como sistemas implica la consideración de circunstancias que no deben ser desconocidas si se trata de operar sobre ellas. Resulta necesario comprender que si el subsistema técnico de una organización se realiza cambios, los otros subsistemas tendrán que ajustarse en función de esos cambios.

3.1 ORGANIZACIONES

Una ***organización*** es un sistema social integrado por *individuos y grupos* que, bajo una determinada estructura y dentro de un contexto al que controlan parcialmente, desarrollan actividades aplicando *recursos* para lograr determinados *objetivos* (valores comunes). Es un sistema abierto, es decir se relaciona con el contexto y, por lo común, es frecuentemente influido por éste.

Básicamente tres elementos son los que permiten caracterizar el fenómeno organizacional:

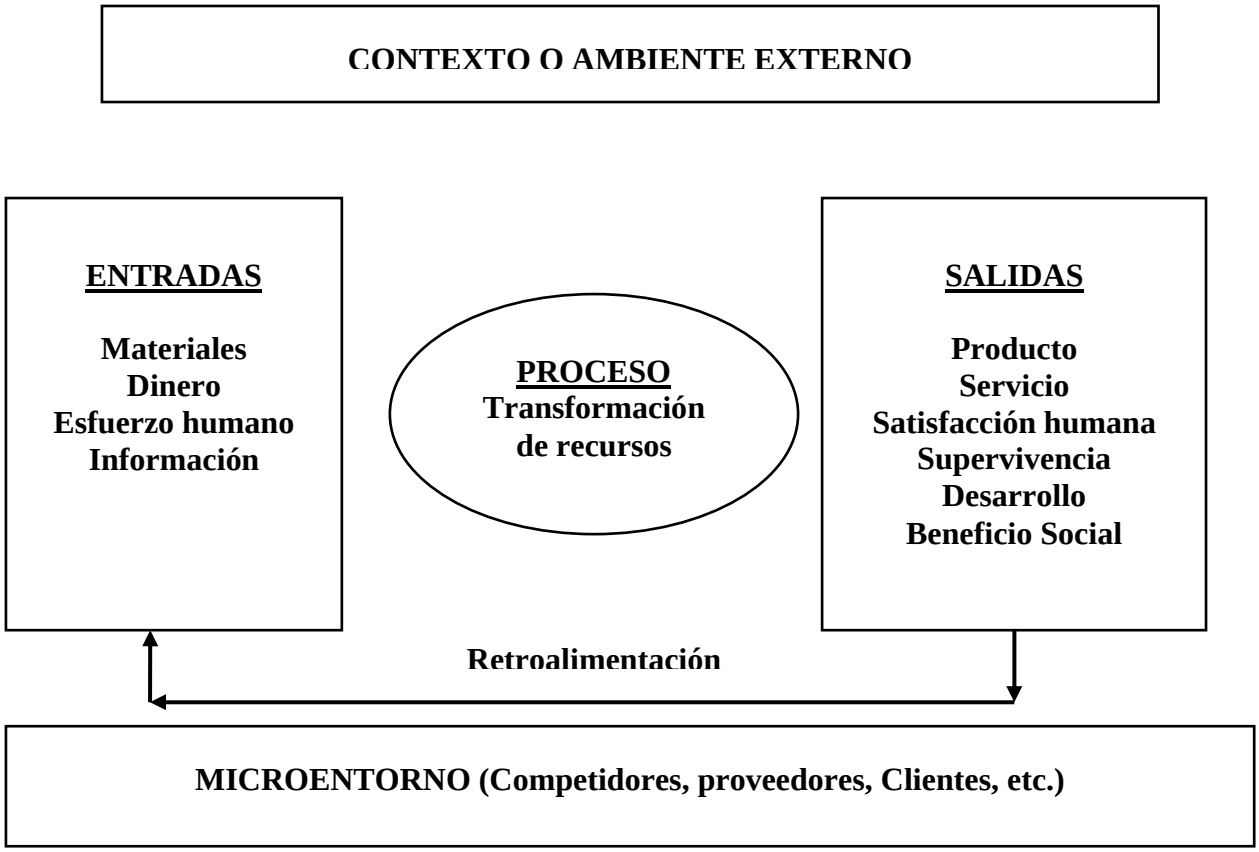
- Los valores comunes (objetivos, llamados también metas, fines o propósitos).
- Los recursos.
- Los agentes.

Ninguno de estos elementos debe faltar para que exista una organización. Un grupo humano que desarrolla actividades y tiene recursos para hacerlo, pero no sabe para qué lo hace, es errático y por lo tanto no constituye una organización. Si sólo consideramos agentes y objetivos de que éstos quieren cumplir, pero no tienen medios para hacerlo, no estamos en presencia de una organización. Y finalmente, aunque existan recursos y objetivos a cumplir, sin grupos ni individuos que desarrollen actividades que permitan traducir los primeros en los segundos, tampoco se estaría ante una organización.

A partir de la constante interacción con el medio ambiente, la organización desarrolla cierta capacidad adaptativa con respecto a los cambios que se producen en aquél, operando en condiciones estables o de equilibrio dinámico, al tiempo que prosigue con su continuo flujo de entradas, transformación y salidas. Por lo tanto, la retroalimentación es fundamental para mantener al sistema dentro de ese equilibrio dinámico.

Si la organización perdiera la capacidad de adaptarse a los cambios del ambiente perdiendo su condición estable, derivaría en un proceso por el cual comenzaría desorganizándose y descomponiéndose, hasta llegar a desaparecer. Por el contrario, una organización como sistema abierto, desarrolla una tendencia que la lleva a avanzar hacia una mayor diferenciación y complejidad, tratando de arribar a un nivel más alto de sistematización. Esta mayor complejidad crea una mayor especialización y diferenciación entre subsistemas.

Las organizaciones consideradas como sistemas, tienen otra característica particular, la **equifinalidad**, que significa que pueden lograr sus objetivos organizacionales de más de una manera. En los sistemas mecánicos, existe una relación directa de causa y efecto entre las condiciones de un estado inicial y un estado final. En las organizaciones al no existir esta correspondencia, el administrador puede utilizar toda una gama de elementos para transformarlos de las más diversas maneras a fin de obtener los resultados buscados.



La Organización como Sistema

3.1.1 Actividades de la organización

Todas las organizaciones realizan diversas actividades a fin de que en la optimización de su desempeño, pueda arribar a los objetivos deseados. Esquemáticamente puede establecerse una jerarquía de niveles organizativos:

- ✓ ***Personal Operativo:*** Está en la base del modelo. Procesan transacciones, participan en las tareas administrativas de la organización y se vinculan con las tareas primarias relacionadas con producir un producto o brindar un servicio.

- ✓ ***Administración Operativa:*** La actividad predominante es la supervisión del personal operativo, por lo que se requiere una conexión muy estrecha con ellos.

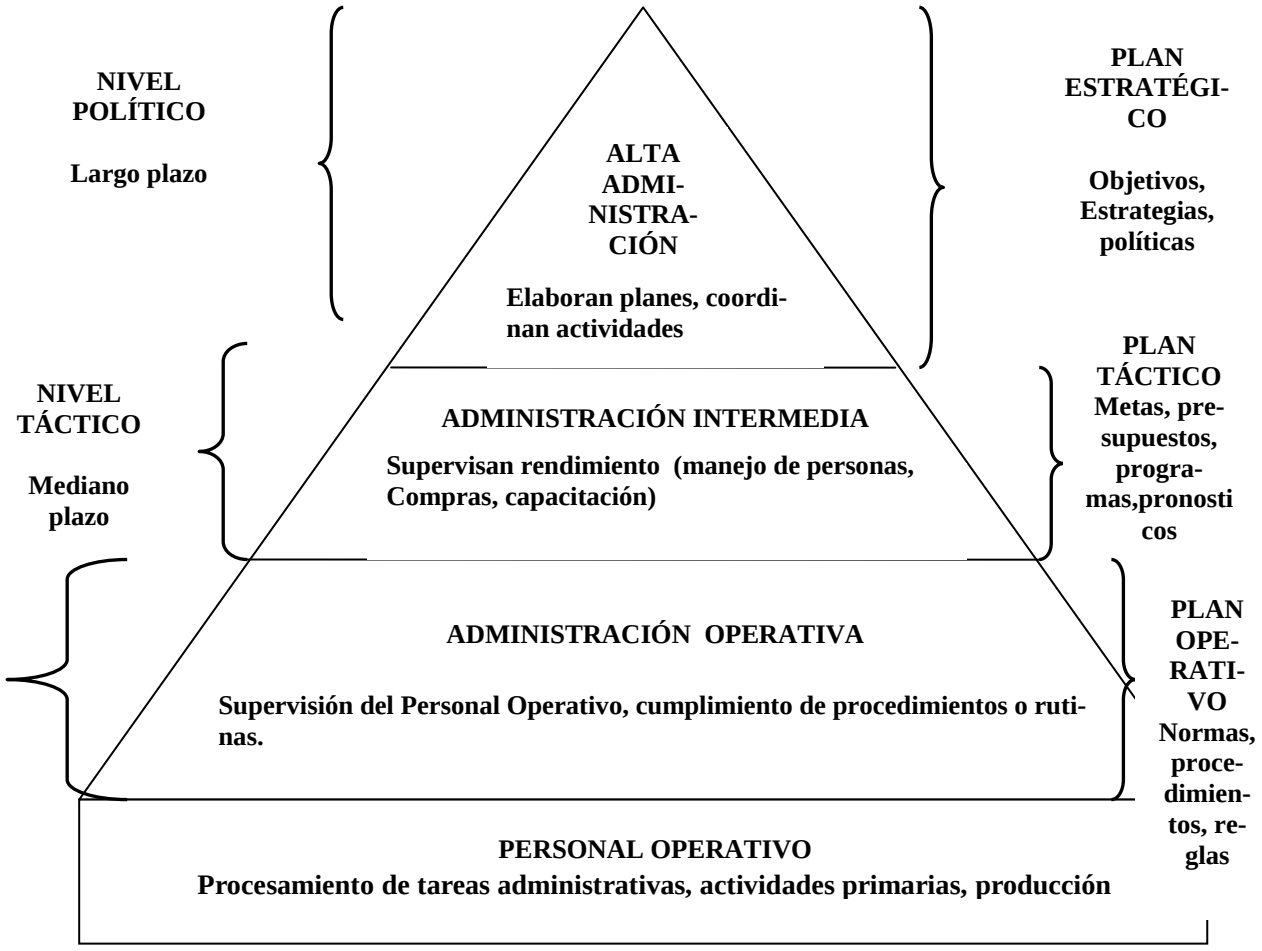
- ✓ ***Administración Intermedia:*** Se ocupa de supervisar el rendimiento de la organización y controlar las actividades que llevan a ésta hacia el logro de sus objetivos.

- ✓ ***Alta Administración:*** Realiza tareas básicamente de planeación y formulación estratégica. Su tarea se orienta hacia el futuro de la organización y dirige un grupo pequeño de funcionarios que realizan los planes de la empresa.

3.1.2. La administración de las organizaciones

Las actividades administrativas en una organización pueden verse en los siguientes tres niveles:

NIVEL	DEFINICIÓN
Planeamiento Estratégico	Determinación de objetivos organizacionales: fijación de políticas sobre los recursos y elementos, delimitación de las estrategias y criterios generales que permitan planear el curso de la organización.
Control Administrativo	Proceso de obtención de recursos, establecimiento de planes y procesos limitados por los objetivos del Planeamiento Estratégico.
Control Operacional	Programación y control de tareas y transacciones, buscando que éstas se realicen en forma eficaz y eficiente dentro de las restricciones presupuestarias.



3.1.3. La jerarquía de las funciones administrativas

La información puede caracterizarse de diversas formas, según resulten más adecuadas para un problema de decisión. En tanto el planeamiento estratégico requiere una información mayormente basada en fuente externa, los otros dos niveles se basan en información generada por la propia organización.

Además, los tres niveles se diferencian por los diferentes tipos de decisiones asociadas a cada uno de ellos.

3.2 EMPRESAS

Una ***empresa*** es aquella organización que se dedica a los negocios. Desarrollan actividades económicas a partir de ciertos recursos (humanos, materiales, energéticos, financieros, informáticos, etc.) que aplican a procesos de producción de bienes y / o prestación de servicios y comercializan para satisfacer las necesidades de los consumidores.

Si bien los objetivos de una empresa suelen ser muchos y variados, el fin económico que se traduce en buscar la rentabilidad del patrimonio a través de la obtención de utilidades, suele hallarse en la mayoría de los casos (con excepciones como por ejemplo: las empresas estatales que procuran fines sociales).

Al desarrollar la gestión económica, la empresa debe enfrentar a otras que buscan los mismos propósitos y en los mismos mercados (conjunto de personas que compran o que podrían comprar un producto), generándose así una competencia, que es cada vez más aguda.

Las empresas pueden clasificarse:

Característica	Tipo de empresa	
De acuerdo al ámbito geográfico en que actúan	Locales	
	Nacionales	
	Multinacionales	
En función a su tamaño	Grandes	
	Medianas	
	Pequeñas	
En relación a su propietario	Privadas	Unipersonales
		Familiares
		Sociedades anónimas
		Grupos o Corporaciones
	Públicas	

3.2.1 ESTRUCTURA ORGANIZATIVA

Las organizaciones se sustentan en dos procesos:

- ✓ ***Delegación:*** Es el proceso por el cual un miembro de la organización transfiere una o más funciones a otro miembro.

- ✓ ***Departamentalización:*** Consiste en agrupar tareas o funciones en conjuntos homogéneos, especializados para el cumplimiento de cierto tipo de actividades. Generalmente adopta la forma de gerencia, departamentos, secciones, etc.

Toda empresa cuenta en forma explícita o implícita, con un cierto conjunto de jerarquías y atribuciones asignadas a los miembros de la misma. Por ese motivo, toda organización cuenta con una estructura que puede ser:

- ✓ ***Formal:*** En general suele estar representada por el *organigrama* (representación gráfica de esta estructura formal, que muestra las relaciones existentes entre las partes que componen la empresa).

- ✓ ***Informal:*** Es la distribución real de la empresa, generalmente en aquellas pequeñas o familiares, que no tienen estructura predeterminada alguna.

3.2.1.1 Organigrama

Los organigramas son una representación gráfica de la estructura formal de la organización en un momento determinado. Si bien no brindan una información completa de los aspectos formales e informales de la organización, constituyen una primera aproximación al conocimiento de los mismos.

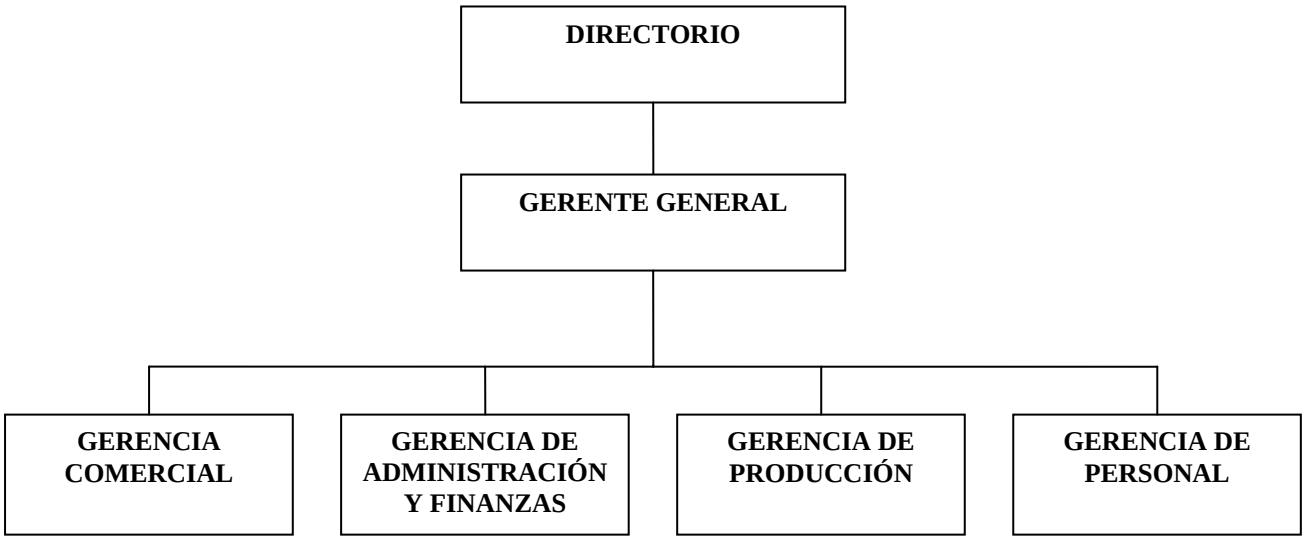
Los Organigramas señalan la vinculación que existe a lo largo de las líneas de autoridad principales. Dejan expresados:

- ✓ La división de funciones.
- ✓ Los niveles jerárquicos.
- ✓ Las líneas de autoridad y responsabilidad.
- ✓ Los canales formales de comunicación.
- ✓ La naturaleza lineal o staff**(1)** [1]del departamento.
- ✓ Los jefes de cada grupo.
- ✓ Las relaciones existentes entre los diversos puestos de la empresa y en cada departamento o sección.
- ✓ Los organigramas deben ser, ante todo, muy claros.
- ✓ Los organigramas deben contener nombres de funciones y no de personas.

3.2.1.2 Tipos de organigrama

- ***Organigrama Vertical***

En este organigrama cada puesto subordinado a otro se representa por cuadros en un nivel inferior, ligados a aquel por líneas que representan la comunicación de responsabilidad y autoridad.

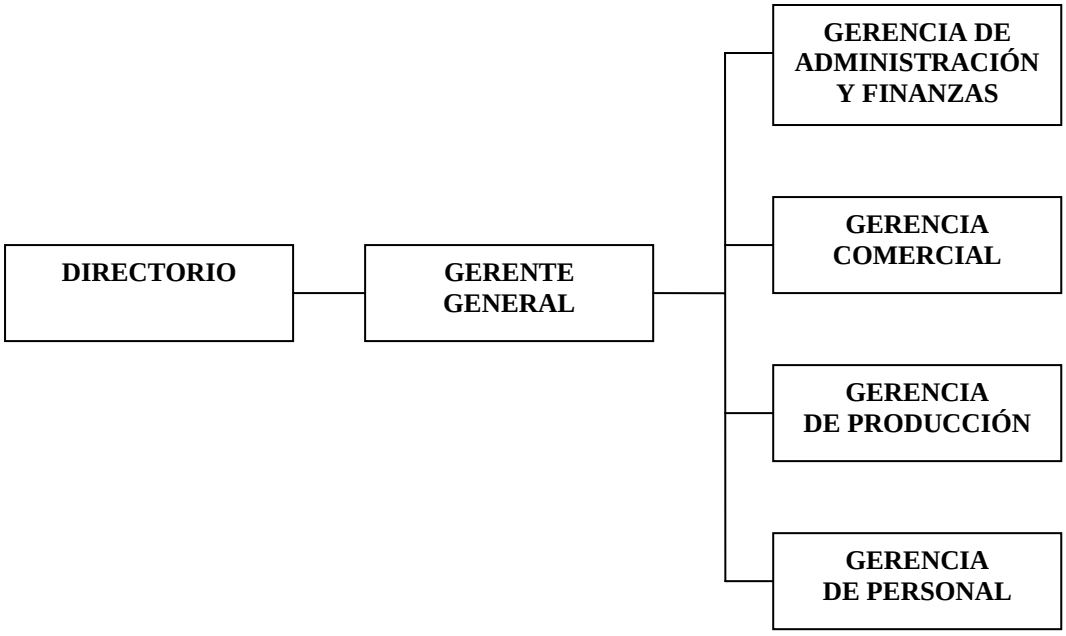


VENTAJAS	DESVENTAJAS
Son los más usados y fácilmente comprensibles.	Se produce el llamado "efecto de triangulación", ya que después de dos niveles, es muy difícil indicar los puestos inferiores.
Indican en forma objetiva las jerarquías del personal.	

- ***Organigrama Horizontal***

Es similar al anterior, sólo que comenzando del nivel máximo jerárquico a la izquierda y haciéndose los demás niveles sucesivamente hacia la derecha.

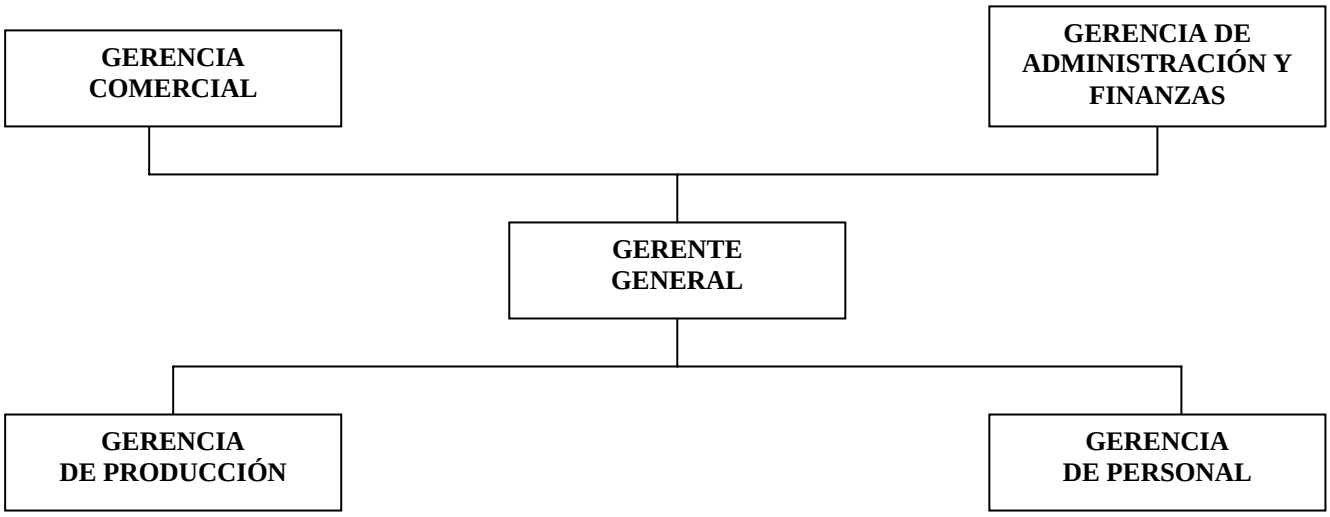
[1] La tarea ***staff*** se caracteriza por el asesoramiento y la consulta, y la influencia en lugar del ejercicio de la autoridad.



VENTAJAS	DESVENTAJAS
Siguen la forma normal en que se acostumbra a leer.	Son poco usados en la práctica.
Disminuye en forma notable el efecto de triangulación.	
Indican mejor la longitud de los niveles por donde pasa la autoridad formal.	

- ***Organigrama Circular***

Formado por un cuadro central, que corresponde a la autoridad máxima en la empresa, a cuyo derredor se constituyen los niveles de la organización.



VENTAJAS	DESVENTAJAS
Señalan muy bien la importancia de los niveles jerárquicos.	Resultan confusos y difíciles de leer, ya que no permiten colocar con facilidad niveles donde hay un solo funcionario y que fuerzan demasiado los niveles.
Disminuyen la idea del estatus más alto o más bajo.	
Permiten colocar mayor número de puestos en el mismo nivel.	

3.2.2 PRINCIPIOS BÁSICOS DE LAS ORGANIZACIONES

Son pautas que determinan formalmente las reglas a las que deben ajustarse las entidades en cuanto de las funciones y responsabilidades. Un organigrama podría denunciar el no cumplimiento de alguno de estos principios:

- ***1) Unidad de mando:*** Cada sector debe responder a un "único superior". Por ejemplo, un sector no puede estar dependiendo jerárquicamente de dos o más sectores.

- ***2) Definición precisa de los niveles jerárquicos:*** La ubicación jerárquica de un sector debe estar definida. Esto puede acarrear problemas y conflictos varios al no tener claro que sector depende de que sector.

- ***3) Separación de funciones:*** Funciones homogéneas deben ser parte de un mismo sector. Cuando una serie de funciones impliquen una responsabilidad patrimonial, deben pertenecer a diferentes sectores. Violar este principio puede crear además condición incompatible desde el punto de vista interno, para tareas de control sobre si misma o control por oposición de intereses.

- ***4) Precisión en la determinación de funciones de línea y de asesoramiento:*** Las funciones de asesoramiento como su nombre lo indica no forman parte de la rutina de trabajo diaria. La existencia de cargos "adscriptos" puede crear confusión.

- ***5) Alcance de control:*** Dependiendo del tipo de tarea a desarrollar cada responsable tiene un número ideal de colaboradores a controlar. Cuanto menos especializada sea la tarea menor será el número de puestos a controlar. En funciones operativas de línea se considera un supervisor por cada tres subordinados.

3.3 SISTEMA INFORMÁTICO

Un sistema o aplicación informática es el conjunto de personas, máquinas y procedimientos que se utilizan para llevar a cabo una tarea informática o proceso de datos. Son típicos sistemas informáticos los de gestión comercial, de control de stock, de ventas, gestión de personal, etc.

3.3.1 PERSONAL INFORMÁTICO

La clasificación que se presenta a continuación no puede ser rigurosa ni exhaustiva debido a los muchos matices que se dan en el personal de informática, dependiendo del tipo de empresa y el volumen de las aplicaciones que se realicen:

Puesto Informático	Características
Director de Informática	▪ Es el responsable máximo ▪ Se encarga de la selección, elección, estructura y dirección del personal y equipos. ▪ Coordina trabajos y controla presupuestos. ▪ Se relaciona con la dirección de la empresa ▪ Estudia las necesidades de mecanización de trabajos.
Jefe de Área de Desarrollo	▪ Es el responsable de la creación y desarrollo de los nuevos sistemas y aplicaciones. ▪ Coordina y distribuye el personal a su cargo entre los distintos proyectos. ▪ Dependen de él los jefes de proyecto, analistas y programadores.
Jefe de Área de Explotación	▪ Es el responsable de la explotación de las aplicaciones y del funcionamiento del sistema. ▪ Planifica los trabajos que deben llevarse a cabo. ▪ Estima los costos. ▪ Establece medidas de seguridad. ▪ Tiene a su cargo a los operadores y grabadores de datos.
Jefe de Proyectos	▪ Su misión consiste en la dirección de un proyecto informático a partir de las necesidades del usuario hasta su explotación. ▪ Controla el desarrollo del proyecto. ▪ Asegura el perfecto funcionamiento de la aplicación una vez terminada. ▪ Estima los recursos y el tiempo necesario para la puesta en marcha del proyecto.
Técnico de Sistemas	▪ Su misión fundamental es el conocimiento profundo de los equipos y del sistema operativo. ▪ Asesora al director de informática y a los jefes de área. ▪ Asesora también a los analistas y programadores. ▪ Impone restricciones de seguridad al personal informático y a los usuarios.
Administrador de la Base de Datos	▪ Es el gestor de la base de datos del sistema. ▪ Se encarga de facilitar el uso de la base al personal informático. ▪ Asesora sobre la base de datos a los jefes de área, jefes de proyecto y analistas.
Administrador del Sistema	▪ Controla en un determinado sistema, los permisos, prioridades y privilegios del personal informático y de los usuarios.
Analistas	▪ Confecciona el análisis de las aplicaciones a desarrollar.

	▪ Ayuda a los programadores en la puesta a punto de los sistemas. ▪ Pueden ser funcionales u orgánicos, según el tipo de análisis que realicen. ▪ Pueden ser de sistemas o aplicaciones.
Programadores	▪ Diseñan el diagrama o pseudocódigo. ▪ Codifican según el lenguaje elegido. ▪ Se encargan de la puesta a punto del sistema. ▪ Se encargan de la documentación dirigida al usuario. ▪ Pueden ser programadores de sistemas o aplicaciones.
Operadores	▪ Se encargan del funcionamiento y operación directa del sistema. Ejecutan los procesos, preparan los soportes, periféricos y material necesario.
Grabadores de Datos	▪ Cargan los datos al sistema. ▪ Dependen del jefe de área de explotación.

Bibliografía.

J. P. Tremblay / P. G. Sorenson; "An Introduction to Data Structures whit applications"; Editorial Mc. Graw Hill.
George Beekman; "Computación & Informática Hoy"; Editorial Addison Wesley.
H. N. Laden /T. R. Gildersleeve; "Diseño de Sistemas de Computación"; Editorial Limusa.
Glenn Brookshear; "Introducción a las Ciencias de la Computación"; Editorial Addison Wesley.
Mario D. Albarracín. E. Alcalde Lancharo García López; "Introducción a la Informática"; Editorial Mc Graw Hill.
Pressman Roger S; "Ingeniería de Software"; Editorial Mc Graw Hill
Yourdon, Edward; "Análisis Estructurado Moderno"; Editorial Prentice Hall
Klein Miguel Jorge; "Cursogramas Técnicas y casos"; Editorial Macchi
IEEE; Std. 1074-1989; "Standard Software Life Cycle. Processes"
Neighbors, J.; The Draco; "Approach to Constructing Software from Reusable Components"
IEEE; "Modelo de ciclo de vida por ensamblaje de componentes reutilizables"
Basili, V.R.-Tumer A.J.; "Iterative Enhancement: A practical technique for Software Development" ; IEEE Modelo de ciclo de vida de refinamiento sucesivo
Bauer, F.L.; "Programming as an Evolutionary Process. Computer Society"; Modelo de ciclo de vida de transformación continua
Böehm, B.W.; "Prototyping vs. Specifying: A Multi-project Experiment. Conf. Soft. Engr."; Modelo de ciclo de vida en cascada y prototipado
Lehman, M.M.A.; "Further Model of Coherent Programming Processes. IEEE Computer Society"; Modelo de ciclo de vida como una transformación continua
Tully, C.; "Software Development Models. IEEE Computer Society"; Modelo de ciclo de vida con emision gradual
Ian Sommerville, "Software Engineering, Six Edition"; Editorial Pearson Educational Limited.
Cátedra Sistemas de Információn; "Operaciones típicas de una empresa"; Editorial Club de Estudio
Lardent Alberto, Gómez Echaren Manuel, Loro Alberto; "Técnicas de Organización, Sistemas y Métodos"; Editorial Club de Estudio
Long, Larry; "Introducción a las computadoras y al procesamiento de información"; Editorial Prentice may
L. Alfonso Ureña López, Antonio Miguel Sánchez Solana, María Teresa Martín Valdivia, José Miguel Mantas Ruiz; "Fundamentos de informática"; Editorial Alfaomega
Narváez, Jorge Luis; "Planificación de organizaciones"; Editorial Universidad Nacional de La Matanza, Editorial Prometeo libros.