SISTEMAS DE INFORMACION

Hasta aquí, se presentaron los conceptos de sistemas y sus características.

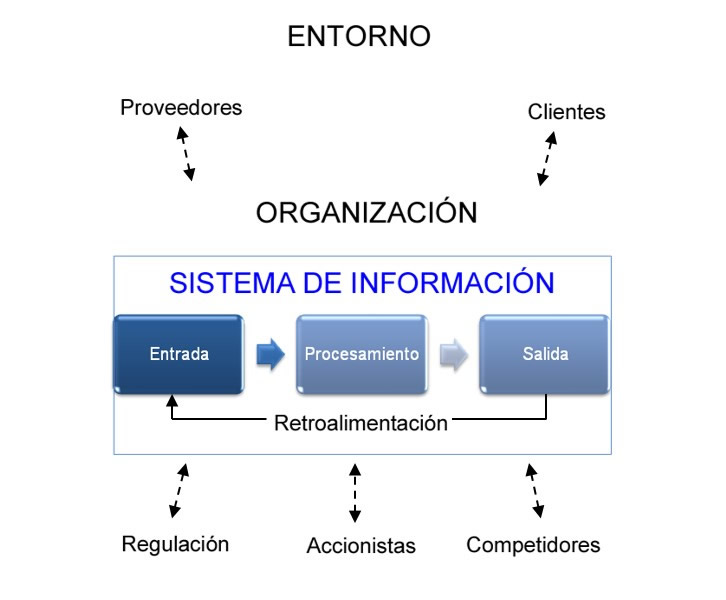
¿Qué sucede si nos referimos a los sistemas de Información?

Recordemos los componentes de un sistema: entrada, proceso, salida y retroalimentación. En los sistemas de información se refiere a:

**Entradas**: la entrada es la actividad que consiste en **recopilar y capturar datos primarios**. (ej. n° de horas trabajadas por los empleados, calificaciones de los estudiantes, escanear un código de barras)

La captura de datos puede ser manual o automatizada, es conveniente realizar la captura de datos en el momento en que se produce el hecho al que está asociado. Procesamiento: supone a conversión de datos en salidas útiles. Esto puede implicar cálculos, realizar comparaciones, almacenamiento de datos para su uso posterior.

**Salida:** La información útil se plasma en una serie de documentos, informes y gráficos, para ser distribuida a la persona adecuada a la organización. En algunos casos la salida de un sistema sirve como entrada de otro.



En el ejemplo de recopilar horas trabajadas por los empleados, el procesamiento sería el cálculo de lo que tiene que cobrar cada uno. La Salida sería el cheque o transferencia de dinero que se gira a la cuenta de cada empleado para el pagarle el sueldo.

Los componentes de un sistema de información:

* Hardware
* Software
* Bases de datos
* Telecomunicaciones, redes e Internet
* Personas ο Procedimientos (recolectar, procesar, distribuir y almacenar)

A lo largo de la materia se verán con más detalle los componentes aquí mencionados. La retroalimentación de la información obtenida en todo este proceso se puede utilizar para realizar ajustes y detectar errores en la captura de datos y/o en su transformación

Modelos para construir sistemas

Para construir un sistema es habitual tener que recurrir a modelos. Por su naturaleza, los modelos son simplificaciones de una realidad.

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.

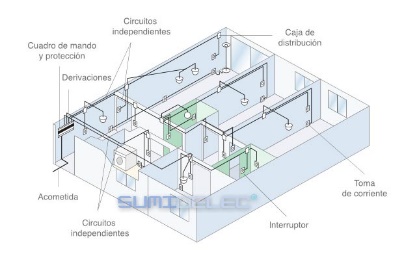
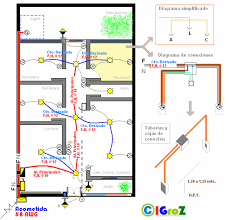
¿Por qué se construyen modelos? ¿Por qué no se construye simplemente el sistema mismo?

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.

Podemos decir que un modelo es una representación de la realidad. Se realizan por medio de **abstracciones**. Cuando mayor semejanza con la realidad tenga un modelo, mejor representará esa realidad la cual se desea significar.

Existen diferentes modelos, los cuáles se enfocan a diferentes partes de un sistema. Son diferentes vistas de una misma situación. Por ejemplo: si se tiene que representar una casa, un plano de la misma sería un modelo. Para esto podríamos tener diferentes planos que representen distintas vistas. (distintos subsistemas)



*Plano de estructura:* se verá representada la estructura de la casa con elementos tales como: paredes, tabiques, puertas, ventanas, etc. Este plano o modelo le será útil a las personas que construyen la estructura (albañiles, capataz)

*Plano de electricidad:* se representará un modelo de toda la parte eléctrica de la casa. Cables, interruptores, etc. Modelo a realizar o desarrollar por los electricistas.

*Plano de instalaciones sanitarias (Hídrico-sanitarios):* se verán representados los elementos como caños, red de desagüe y drenaje, rejillas, etc. Los plomeros serán los que deberán realizar su trabajo de plomería siguiendo este modelo.

**Modelo: Es una representación de la realidad.**

Genéricamente podemos definir distintos tipos de modelos para representar todo tipo de sistemas a saber:

Modelo de Caja Negra:

Esta representación muestra las entradas y las salidas del sistema que estamos analizando, no se define en cuanto al proceso. Se le denomina “caja negra”, ya que las entradas y las salidas se conocen, pero no la transformación que se produce internamente en el sistema (ver figura a continuación)



Ejemplo: en una máquina expendedora de café, se conocen las entradas: monedas o billetes y elección del tipo de café (dulce, corto, largo, capuchino, etc). Estas serías las Entradas definidas

El proceso que realiza la máquina no se conoce, tal vez se pueda intuir, pero no es relevante para el usuario (persona que interactúa con el sistema) en qué proporción utiliza cada elemento o en qué orden lo realiza. Por eso se dice “caja negra”, no se puede ver lo que hay adentro.

Y las salidas definidas serían: el vaso y la bebida.



Modelo funcional

Existen distintos tipos de modelos funcionales, según la disciplina, según la profundidad con la cual queremos mostrar el funcionamiento del sistema y según el paradigma con el cual estamos modelando. Genéricamente podemos decir que un modelo funcional representa funciones, actividades, transformaciones o procesos que se producen en el sistema. Estamos mirando el interior de nuestro modelo de caja negra

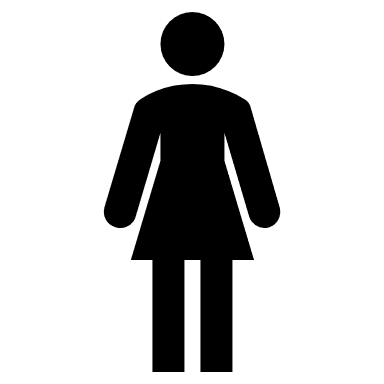
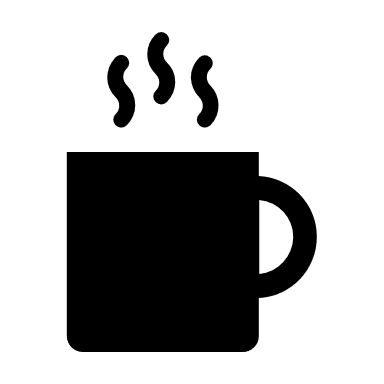
En nuestro ejemplo de la expendedora de café podemos definir las funciones, siempre desde un punto de vista determinado, teniendo en cuenta que cada función puede descomponerse en otras como:

1. Recibir dinero y controlar dinero
2. Recibir orden
3. Armar pedido
4. Preparar pedido
5. Servir bebida

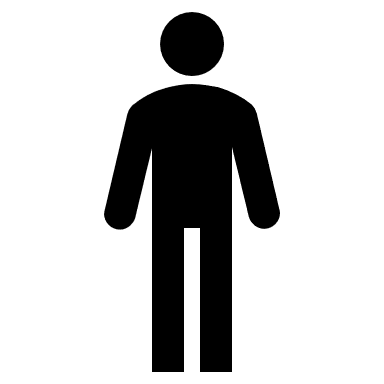
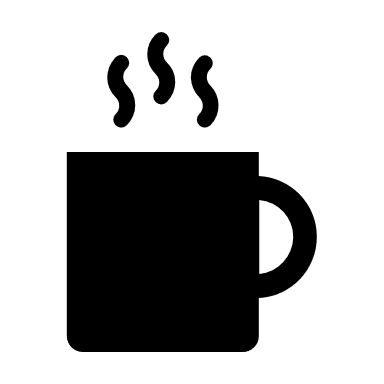
Una representación de este modelo funcional podría ser el siguiente

Preparar pedido

Recibir dinero

Armar pedido orden

 }

Servir bebidas

Recibir orden

Modelo estructural

Este modelo representa las estructuras del sistema. Estas estructuras también son variadas y definidas por el observador, se pueden asociar a las partes que se determinaron en el sistema.

Ejemplos de modelos estructurales: Organigrama, clases, entidades en bases de datos.

Sigamos con nuestro caso de la expendedora de café. Queremos mostrar la estructura o partes de la expendedora.

Expendedora de café

Conexiones eléctricas

Recipientes

Filtros

Productos

Café

Agua

Leche

Azúcar

Procesos o actividades esenciales en todo sistema

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.

Posibles razones para iniciar el CAMBIO

1. Problemas con el sistema existente
2. Aprovechar nuevas oportunidades
3. Competencia creciente
4. Emplear más eficazmente la información
5. Crecimiento organizativo
6. Fusión o adquisición
7. Cambio en el mercado o entorno de negocios
8. …

Etapas del Proceso de Desarrollo de los sistemas en general

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 de la organización cambian en forma importante.

Enumeramos los pasos o etapas comunes o generales, en todos los desarrollos (industriales, civiles, electrónicos, informáticos, mecánicos).

*Aclaramos que cada disciplina tendrá una visión particular de estos procesos adaptados a sus necesidades.*

1) Una primera fase de **identificación del problema** a solucionar como también la investigación de mercados. Se definen las ideas que se llevarán a cabo. Se realiza también el estudio de factibilidad desde distintos puntos de vista (económico, técnico y operativo)

2) **Desarrollo de la solución:** se define que se hará y cómo se hará, se definen materiales, herramientas, recursos (personas, tecnología, tiempos, etc), se utilizan los modelos y se desarrolla el producto

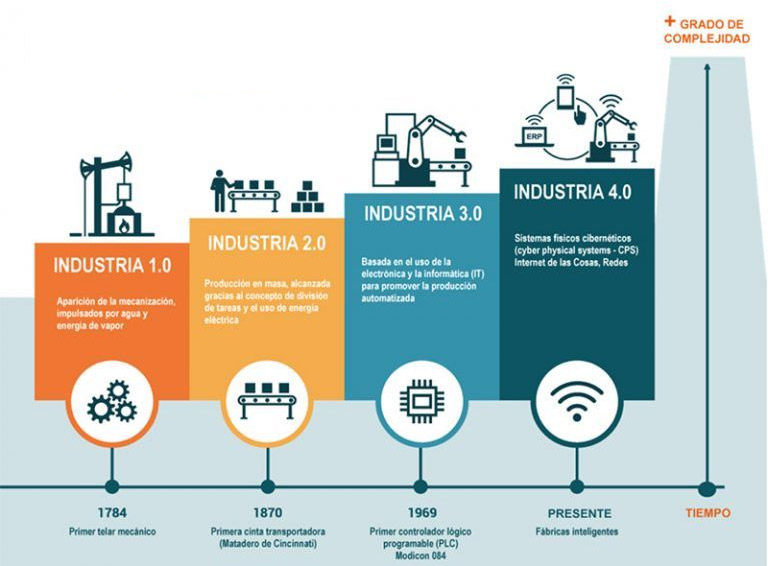
3) **Puesta a punto del sistema:** durante la puesta en marcha de todo sistema, surgirán inconvenientes que deben ser solucionados, se realizan las pruebas.

4) **Seguimiento del sistema construido:** Las fases no terminan con la entrega al cliente, sino que, muchas veces, es preciso continuar prestando servicios debido a que se necesita mejorar el rendimiento o se solicitan cambios, o se necesita mantenimiento y operaciones en planta.

Etapas explicadas con mayor detalle aplicadas a los sistemas de desarrollo de software

Adelantándonos un poco a los conceptos que se desarrollaran más adelante diremos que: La inteligencia de la **nueva fábrica** es el resultado de la convergencia de las tecnologías de la información, su unión en un “ecosistema digital” con otras tecnologías industriales y el desarrollo de nuevos procesos de organización. Por ello, la informática y sus profesionales serán elementos clave en el escenario de industria 4.0

Explicamos en este punto los conceptos básicos relacionado con el software y la ingeniería del software. (trate de pensar cuál es su equivalente en otras disciplinas -ingeniería civil, industrial. Etc.)



Software- producto y proceso

**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.

**Producto Software:** Consiste en programas desarrollados y en la documentación asociada.

**Diferencia entre Proceso y Producto:** El proceso son las actividades que se realizan y el producto es el resultado de esas actividades.

Ejemplo: La receta de cómo hacer una torta sería el detalle del proceso. La torta es el producto. De la misma manera el Proceso de software describe que actividades deberán hacerse para obtener un producto de software.



Participantes en el Desarrollo de Sistemas software

En el desarrollo de un proyecto participan distintos actores que cumplen diferentes roles. 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. En el grupo de desarrollo de sistemas se pueden enumerar los siguientes roles:

|  |  |
| --- | --- |
| PARTICIPANTES | DESCRIPCIÓN |
| Usuarios  Los usuarios de la Base de Datos | lizlozz2223 | Son las personas que interactuarán con el sistema en forma regular. A quienes estará destinado el software |
| Analista Funcional | Es un profesional encargado de analizar las necesidades funcionales que deberá satisfacer el software. |
| Arquitecto de Software | Es el responsable de la arquitectura del sistema: Define la plataforma en la cual se realizará el desarrollo, la tecnología a utilizar, la estructura interna del proyecto de desarrollo. |
| Líder de Proyecto | Es el encargado de asignar tareas a los programadores y monitorear el accionar de los mismos. |
| Programador | Se encarga de modificar o desarrollar programas para satisfacer las necesidades de los usuarios |
| Tester (Probador) | Se encarga de realizar el testing (testeo ó prueba), es indispensable a fin de asegurar cierto grado de calidad del producto software. Se recomienda que las pruebas sean realizadas por personas distintas a los programadores del software.  Los tipos de pruebas existentes son:  • Caja Negra: No se inspecciona el código fuente el control se realiza con lotes de datos a través de la interface del producto  • Caja Blanca: Se inspecciona el código fuente en búsqueda de defectos |
| Implementador | Pone en funcionamiento el producto final en el destino. Se encarga de: Configuración de equipos, Instalación del Software, etc. |

Etapas en el Desarrollo del Software

Investigación.

En esta etapa se realiza un estudio de factibilidad. Aquí se estima si la realización del software es factible. Si es posible o realizable desde el punto de vista económico y de la tecnología.

Se estiman si las necesidades de los usuarios se pueden realizar o satisfacer con las tecnologías actuales de software y de hardware. El estudio dará como resultado el costo del sistema y de acuerdo a esto si es posible realizarlo con el presupuesto que se tiene.

Este estudio deberá ser rápido y económico, y es de suma importancia para tomar la decisión de seguir con el proyecto (siguientes etapas, análisis, diseño, etc.) o no.

Consiste en realizar un estudio de factibilidad ¿Es posible realizar el sistema solicitado?

Este estudio incluye tres aspectos: Técnico, Político y Económico.

***Estudio de Factibilidad: Estudio que se realiza para decidir si es factible o posible realizar el producto software. Se estudiará desde lo económico y técnico.***

Estas etapas no son exclusivas al desarrollo de un sistema informático, pueden ser aplicadas a otros tipos de sistemas sin requerir modificaciones.

Supongamos un Ingeniero Civil a quién se lo contrata para realizar un edificio de 15 pisos. Investigar si es Factible (o posible) la propuesta desde el punto de Vista……



***Economico:*** ¿El importe a recibir, permite financiar el proyecto (la obra)? • Materiales requeridos, contratación de personal, etc.

***Politico:*** ¿Hay alguna normativa que impida la realización del proyecto? (la construcción del edificio en la zona)

***Tecnico:*** ¿Existe la tecnología para realizarlo? • Ejemplo: el suelo podría no ser apto para soportar el peso total de la edificación construida. Existen técnicas o materiales para construirlo a pesar de eso?

Analisis

En el análisis se deberán adquirir los requerimientos del sistema a desarrollar. Un requerimiento es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto, servicio o sistema. En esta etapa deberá quedar claro y verificado por el usuario:

¿**Qué** debe hacer el sistema?

¿**Qué** cosas hará el sistema y que cosas no hará?

¿**Qué** usuarios interactúan con el sistema?

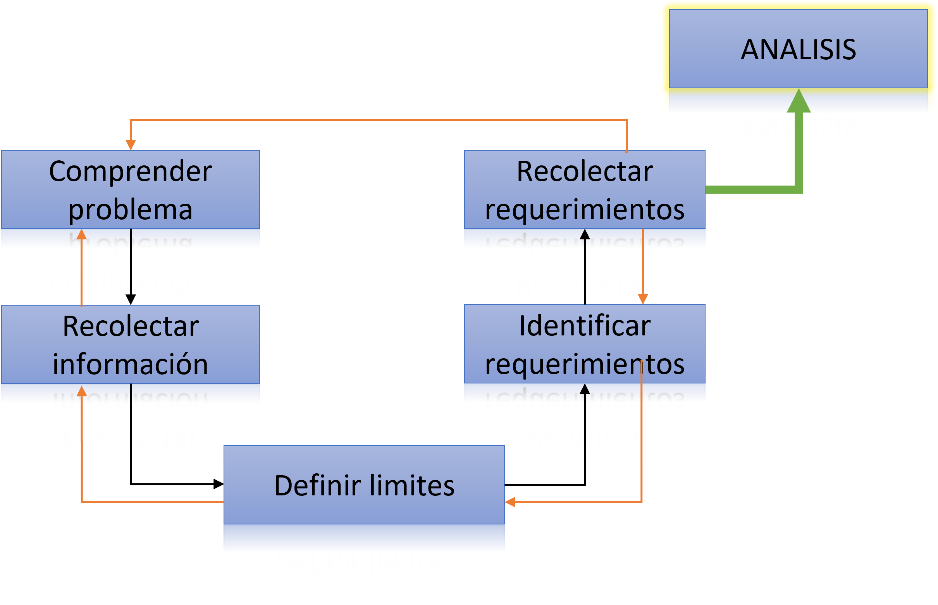
Establecen QUÉ debe hacer el sistema, pero NO CÓMO hacerlo.

A esto se llama **requerimientos**

La obtención de los requerimientos de un sistema comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer por el sistema.

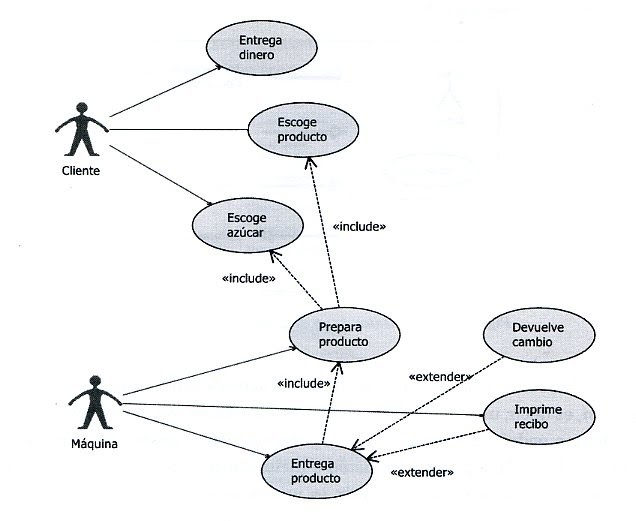
Una colección de requerimientos describe las características o atributos del sistema deseado. Es indispensable la participación de los usuarios y clientes, para la identificación de los requerimientos del sistema.

Los analistas pueden emplear varias técnicas para obtener los requisitos del sistema. Algunas de esas técnicas son: **las encuestas**, las **entrevistas**, los **cuestionarios** y la **observación**. Técnicas más modernas incluyen los prototipos, y utilizan casos de uso. Cuando sea necesario, el analista empleará una combinación de estos métodos para establecer los requisitos exactos y de ese modo, producir un sistema que resuelva las necesidades del cliente. La figura que se muestra a continuación presenta el proceso de obtención de requerimientos.



Si continuamos con nuestra máquina expendedora de café, mostramos aquí un diagrama que se utiliza para representar los requerimientos del sistema. Si observan el mismo existen los usuarios del sistema, que para este modelo se denominan actores.

Las elipses muestran las funciones que realiza el sistema según el punto de vista desde el cual lo vamos a modelar. Ejemplo de ello sería el usuario que va a consumir el café, y el otro punto de vista que es la máquina.



Diseño

Para diseñar un sistema es habitual el tener que recurrir a modelos. 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.

**Modelo:** Es una representación de la realidad.

**Modelo de Procesos del Software:** es una abstracción de un proceso real.

**Abstracción**: Proceso mental o intelectual que se realiza para representar y seleccionar características y propiedades de determinadas cosas del mundo real. Es una representación mental de la realidad que se desea representar.

**Paradigma**

Esta palabra ya la utilizamos anteriormente.

Recordemos que 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.

Para construir un producto software se pueden utilizar distintos paradigmas. (Para construir una casa se pueden usar distintos paradigmas: Construcción Tradicional (ladrillos, mezcla, etc.) o Construcción en Seco (placas, parantes, etc.)

En la construcción de sistemas se utilizan distintos paradigmas:

Paradigma Estructurado: aquí se enfatizan los procesos separados de los datos

Paradigma Orientado a Objetos. POO: los proceso y los datos están contemplados en el objeto

Paradigma Estructurado

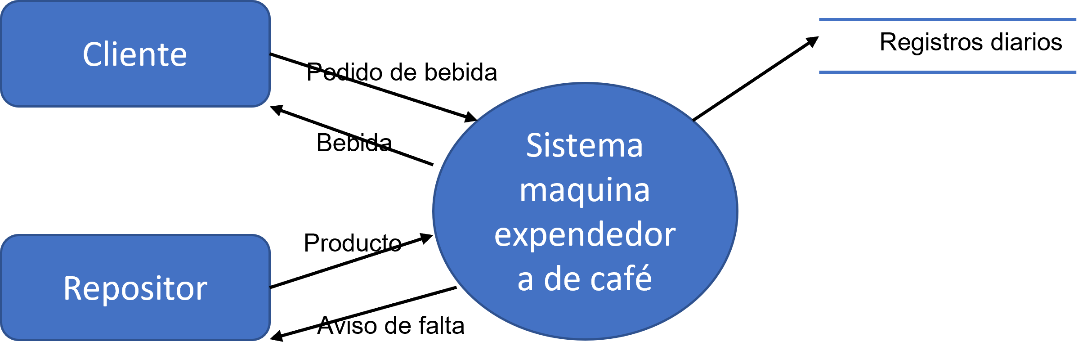
Este paradigma se ocupa de plantear las funciones y procesos que realizará el sistema y la forma en que se llevarán a cabo.

Requiere traducir el dominio (realidad) del problema en una serie de funciones. El analista debe comprender primero el dominio del problema y a continuación documentar las funciones que debe proporcionar el sistema.

Los datos y los procedimientos se representan por separado, ya que lo único que se busca es el procesamiento de unos datos de entrada para obtener otros de salida.

Este diseño continua, en la siguiente etapa (implementación) como programación estructurada.

En la figura siguiente se muestra un modelo que utiliza este paradigma para representar el flujo de actividades que el sistema tendrá con los usuarios. Donde el circulo representa el sistema. Las flechas el flujo de actividades Los rectángulos las entidades que interactúan El rectángulo sin rellenar representa los lugares que se deben proveer para guardar información



Paradigma Orientado a Objetos (POO)

En este paradigma se representa la realidad mediante 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. 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 que no sean esenciales.

El paradigma orientado a objetos (POO) 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, considerando al mundo como una colección de objetos significativos que colaboran para obtener un comportamiento de alto nivel. Los objetos, se generar a partir de CLASES.

Implementación

En la etapa de implementación, se programa, y esta programación se realiza de acuerdo al paradigma que se utilizó en la etapa anterior.

En la programación estructurada sólo se escriben funciones que procesan datos, los cuales se declara previamente a la construcción de la función. La idea principal de esta forma de programación es separar las partes complejas del programa en módulos. De esta manera se tiene un diseño modular compuestos por módulos independientes que pueden comunicarse entre sí. Ejemplos de lenguaje utilizado es el C.

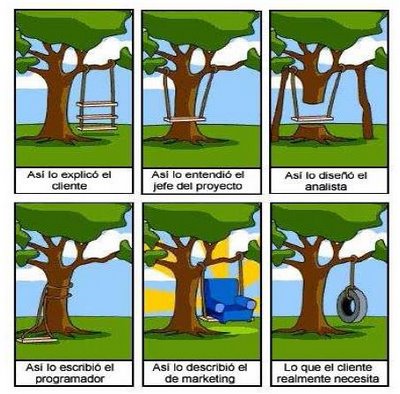
Si se utiliza el paradigma orientado a objetos se utilizan lenguajes tales como JAVA y C++. La programación orientada a objetos o POO (OOP según sus siglas en inglés) usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. En la POO se definen clases que reúnen a los distintos objetos que interactúan entre sí. Las funciones del sistema las proveen los objetos, los que conocen sus datos (llamados atributos), y cada objeto juega un rol en la solución del problema. Cada objeto proporciona un servicio o realiza una acción que es posteriormente utilizada por otros miembros de la comunidad.

Pruebas (Testing).

En esta etapa se realiza las pruebas ó “testing” del sistema, esto es, la prueba de cada componente del mismo hasta llegar a una integración total.

Esta etapa consiste en validar el desarrollo realizado. El objetivo principal del testing es determinar si los productos que surgen como consecuencia de cada una de las etapas anteriormente vistas, satisfacen con los requisitos especificados.

Cabe recordar que se entiende por Producto software: “A todos los programas desarrollados y la documentación asociada”.



Las pruebas tienen dos momentos:

• Planificar las pruebas: en etapas tempranas del proyecto de desarrollo.

• Realizar las pruebas: en etapa de prueba.

En la etapa de la planificación de las pruebas se pueden identificar las siguientes tareas:

• Realizar el plan de pruebas

• Hacer una lista para verificar requerimientos.

• Diseñar los Casos de Prueba.

El objetivo de las pruebas es descubrir errores.

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

**Mantener:**

•Reparar fallas

•Corregir errores inadvertidos

**Actualizar:**

•Adaptar el Software a diferentes entornos

•funcionalidades que mejoran la performance del sistema

**Mejorar el Sistema:**

•Actualizar el Sistema a Nuevos Cambios

•Agregar o Modificar Funcionalidades del Sistema

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*.

En este apunte no se ha incluido a la documentación en una etapa particular del proceso de desarrollo de un software porque se considera que deberá irse construyendo a medida que se construye el software (sobre todo la documentación técnica).

Tipos de documentación

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 técnica ó Documentación del sistema** (describir el software en sí para poder mantener el sistema en una etapa posterior de su ciclo de vida):

Es una documentación técnica en la cual se asientan los requisitos del usuario, los distintos modelos que un analista haya realizado, comentarios sobre la opción de diseño elegida, explicación de los distintos componentes desarrollados, etc.

En el pasado, consistía en los programas fuentes finales y algunas explicaciones a grandes rasgos. Esta documentación informal simplemente ya no es aceptable para los grandes sistemas de software de la actualidad.