



UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

**SISTEMAS DE SOFTWARE
ESPECIALIZACIÓN GERENCIA EN INFORMÁTICA
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA**

Vicerrectoría de Educación a Distancia y virtual

2016



El módulo de estudio de la asignatura SISTEMAS DE SOFTWARE es propiedad de la Corporación Universitaria Remington. Las imágenes fueron tomadas de diferentes fuentes que se relacionan en los derechos de autor y las citas en la bibliografía. El contenido del módulo está protegido por las leyes de derechos de autor que rigen al país.

Este material tiene fines educativos y no puede usarse con propósitos económicos o comerciales.

AUTOR

Piedad maría metaute paniagua

Ingeniera de Sistemas, Especialista en Finanzas, Candidata a Magister en Educación y Desarrollo Humano, diplomados en Competencias Pedagógicas, Diseño Curricular, seminarios sobre Formación en Procesos y Técnicas de Investigación, sobre Negociación electrónica, sobre Evaluación por Competencias pruebas ECAES, ICFES, SABER. Con amplia experiencia docente en los diferentes ciclos de educación desde Básica Secundaria, Técnica Tecnológica, Profesional, Especialización en temáticas relacionadas con los sistemas, la computación y la informática, así como experiencia en empresas del sector productivo en lo referente a la construcción de productos de software.

pmetaute@gmail.com

Nota: el autor certificó (de manera verbal o escrita) No haber incurrido en fraude científico, plagio o vicios de autoría; en caso contrario eximió de toda responsabilidad a la Corporación Universitaria Remington, y se declaró como el único responsable.

RESPONSABLES

Jorge Mauricio Sepúlveda Castaño

Decano de la Facultad de Ciencias Básicas e Ingeniería

jsepulveda@uniremington.edu.co

Eduardo Alfredo Castillo Builes

Vicerrector modalidad distancia y virtual

ecastillo@uniremington.edu.co

Francisco Javier Álvarez Gómez

Coordinador CUR-Virtual

falvarez@uniremington.edu.co

GRUPO DE APOYO

Personal de la Unidad CUR-Virtual

EDICIÓN Y MONTAJE

Primera versión. Febrero de 2011.

Segunda versión. Marzo de 2012

Tercera versión. noviembre de 2015

Cuarta versión 2016

Derechos Reservados



Esta obra es publicada bajo la licencia Creative Commons.
Reconocimiento-No Comercial-Compartir Igual 2.5 Colombia.

TABLA DE CONTENIDO

	Pág.
1 MAPA DE LA ASIGNATURA	5
2 UNIDAD 1 El Software y su Clasificación.....	6
2.1.1 RELACIÓN DE CONCEPTOS.....	6
2.1.2 OBJETIVO GENERAL	9
2.1.3 OBJETIVOS ESPECÍFICOS	10
2.2 Los Sistemas Operativos (SO)	10
2.2.2 Ejercicio del tema	15
2.3 Tema 2 Los Lenguajes de Programación	16
2.3.1 En la actualidad los lenguajes de programación se clasifican en:	17
2.3.2 Breve evolución de los lenguajes de programación	17
2.3.3 Ejercicio del tema:	20
2.4 Tema 3 Los motores de Bases de Datos	21
2.4.1 Ejercicio del tema:	30
2.5 Tema 4 Los Sistemas de Aplicación	31
2.5.1 Ejercicio del tema:	32
2.6 Tema 5 Los Sistemas Expertos.....	32
2.6.1 Ejercicio del tema:	33
3 UNIDAD 2 LA INGENIERÍA DEL SOFTWARE (IS).....	34
3.1.1 RELACIÓN DE CONCEPTOS.....	34
3.1.2 OBJETIVO GENERAL	38
3.1.3 OBJETIVOS ESPECÍFICOS	38
3.2 Tema 1 Modelos o Metodologías en la IS	38

3.2.1	Ejercicio del tema:	43
3.3	Tema 2: Los Requerimientos	43
3.3.1	Ejercicio del tema:	47
3.4	Tema 3 Herramientas de Análisis	47
3.4.1	Ejercicio del tema:	53
3.5	Tema 4 Los diseños actuales.....	53
3.5.1	Ejercicio del tema:	56
3.6	Tema 5 Los Riesgos	56
3.6.1	Ejercicio del tema:	58
3.7	Tema 6 Pruebas y Calidad.....	58
3.7.1	Ejercicio del tema:	63
4	UNIDAD 3 Plataformas LMS (Learning Management System o Sistema de Gestión de Aprendizaje)	65
4.1.1	OBJETIVO GENERAL	66
4.1.2	OBJETIVOS ESPECÍFICOS	66
4.2	Tema 1 Utilización, plataformas y aplicaciones complementarias	67
4.2.1	Ejercicio del tema:	69
5	PISTAS DE APRENDIZAJE	70
6	GLOSARIO	71
7	BIBLIOGRAFÍA	72

1 MAPA DE LA ASIGNATURA

SISTEMAS DE SOFTWARE

PROPÓSITO GENERAL DEL MÓDULO

Con el presente módulo, se pretende presentar diferentes herramientas y argumentaciones al Especialista en Gerencia en Informática a través de información que tiene que ver los diferentes tipos de software y sus especificidades, en lo referente a la administración de las máquinas, desarrollo de nuevos productos informáticos, manejo de datos y aplicaciones que tienen que ver con las reglas del negocio así como lo concerniente al software especializado para su elección en el momento que se requiera.

De igual forma se ofrece información suficiente para que éste, el Especialista en Gerencia Informática, tenga un conocimiento claro, sobre los procesos de Ingeniería del Software como factores relevantes para la comprensión de requerimientos, el respectivo análisis, diseño, codificación, pruebas e implementación de un producto informático de calidad.

Y por último se muestra, cómo las herramientas Web han invadido los diferentes contexto organizacionales para el intercambio de información, entre las cuales se abordan plataformas LMS (Learning Management System), la Computación en Nubes (Cloud Computing), entre otras, aptas para el intercambio de información On Line.

OBJETIVO GENERAL

Identificar los diferentes tipos de software, así como los fundamentos de ingeniería de software que se orientan hacia la construcción de aplicativos con calidad, permitiendo la estructuración de conceptos importantes para la adquisición de productos informáticos acorde a las necesidades.

OBJETIVOS ESPECÍFICOS

- Analizar los diferentes tipos de software y su aplicación acorde a las necesidades en cuanto a la administración de las máquinas, desarrollo de nuevos productos informáticos, manejo de datos y aplicaciones que tienen que ver con las reglas del negocio, así como lo concerniente al software especializado para su elección en el momento que se requiera.
- Identificar los principales elementos que giran en torno a los procesos de Ingeniería del Software como factores relevantes para la comprensión de requerimientos, el respectivo análisis, diseño, codificación, pruebas e implementación de un producto informático de calidad.
- Explorar plataformas LMS (Learning Management System), para la adquisición de un conocimiento más claro sobre su utilidad y conformación, así como la manipulación de herramientas en la nube (Cloud Computing) aptas para el intercambio de información On Line.

UNIDAD 1

El Software y su Clasificación

UNIDAD 2

La Ingeniería del Software

UNIDAD 3

Plataformas LMS

2 UNIDAD 1 EL SOFTWARE Y SU CLASIFICACIÓN

En los últimos 30 años, las herramientas informáticas, han cambiado la vida de las sociedades, donde varios protagonistas han dejado y siguen dejando huella, entre ellos la empresa Microsoft y la empresa Apple. Vale la pena observar la película (*Piratas de Silicon Valley*), que muestra los orígenes de estas dos grandes empresas, cuya historia gira alrededor de jóvenes con pensamiento prospectivo, Steve Jobs, Steve Wozniak (creadores de Apple Computer Inc) y Bill Gates, Paul Allen y Steve Ballmer (creadores de MicroSoft), dos de las empresas más importantes en el mundo de la informática.

Al final de la película nos quedamos con dos frases pronunciadas por sus protagonistas:

Steve Jobs: "Somos mejores que vosotros. Nuestro producto es superior."

Bill Gates: "No lo entiendes Steve, eso no importa."

Película Piratas de Silicon Valley



Piratas de Silicon Valley Parte 1 de 3 Latino [Enlace](#)

2.1.1 RELACIÓN DE CONCEPTOS

Para abordar la temática relacionada con los sistemas de software, es importante conocer, a grandes rasgos, la evolución que han sufrido los sistemas computacionales, es así como los computadores tienen sus inicios en los años 40, con la primera máquina electromecánica llamada el ENIAC, ofreciendo su desarrollo más perfeccionado entre los años 60-70, representado por grandes computadoras, especial para grandes empresas (mainframes), donde los usuarios se limitaban al manejo de terminales, sin ninguna función de procesamiento.

A principio de 80 aparecen los computadores de escritorio (Personal Computer), así como el software de escritorio para la realización de trabajos en plantillas de cálculo y procesadores de texto, ya en los años 80 se

conectan redes y servidores, se manejan almacenamientos en discos duros, aparecen los correos electrónicos así como la compartición de archivos y agendas, todo ello en manos de algunas universidades, fuerzas militares y grandes compañías. A principio de los años 90, se maneja el dominio de cliente servidor a través de las redes de comunicación, ya aplicado a diferentes compañías e instituciones.

A mediado de los 90 internet entra con páginas estáticas, las que rápidamente se convirtieron en

Accesos dinámicos e interactivos para diferentes tipos de usuarios, así como la utilización de intranet, extranet sobre la que corren múltiples aplicaciones que permiten el apoyo de los diferentes procesos en las organizaciones.

A principio del 2000, se crea un nuevo concepto en redes inalámbricas, dispositivos móviles, consolas para juegos, utilizados especialmente estos últimos para el entretenimiento, se expande internet como medio de comunicación y de apoyo a todo tipo de actividades personales y profesionales, convirtiéndose dichas herramientas TIC (Tecnologías de la Información y Comunicación) en indispensable en todos los ambientes.

En los primeros años el crédito principal estaba dado especialmente al hardware (máquinas), siendo el software (programas) secundario, el paradigma ha cambiado, las máquinas han bajado de costo y el software se construye a la medida, acorde a los requerimientos y necesidades del cliente, buscando cumplir con estándares de calidad y la satisfacción de dicho cliente.

El software se ha clasificado acorde a su utilidad, por lo tanto aparecen grupos como los sistemas operativos, los lenguajes de programación, el software aplicación, los motores de bases de datos, entre otros con funciones específicas:

Los Sistemas Operativos

Considerado como el software más importante de una computadora, encargado de administrar todos los recursos de hardware y software, sirviendo como puente de comunicación entre el usuario y los recursos de cómputo. Existen varias marcas de Sistemas Operativos, unos más comerciales que otros entre ellos tenemos la línea que ofrece la empresa Microsoft, la de IBM, la de Apple, entre otros, además de sistemas operativos libres es el caso del sistema operativo Linux. Otro tipo de software que vale la pena abordar tiene que ver con los controladores de dispositivos, herramientas de diagnóstico y software de utilidades que sin ser sistemas operativos, ayudan con la gestión de los sistemas de cómputo.

Los Lenguajes de Programación

Conocido como el software especializado para crear productos de software con diversos fines, los cuales se basan en códigos especiales que al ser digitados por el desarrollador se crea el código fuente del programa (que corresponde a instrucciones que se pueden entender desde el lenguaje natural) y al ser compilado por el lenguaje se convierte en código objeto, este último interpretado por la máquina y convertido a lenguaje binario y códigos encriptados para su distribución al cliente.

En la actualidad el desarrollo de software se realiza con la metodología objetual, con el ánimo de reutilizar el código para ser aplicado a otros programas, evitando re codificación y reproceso. Los lenguajes objetuales más utilizados actualmente en la industria del software equivalen a Java, .NET, PHP por cumplir con dichas características objetuales.

Los motores de Bases de Datos

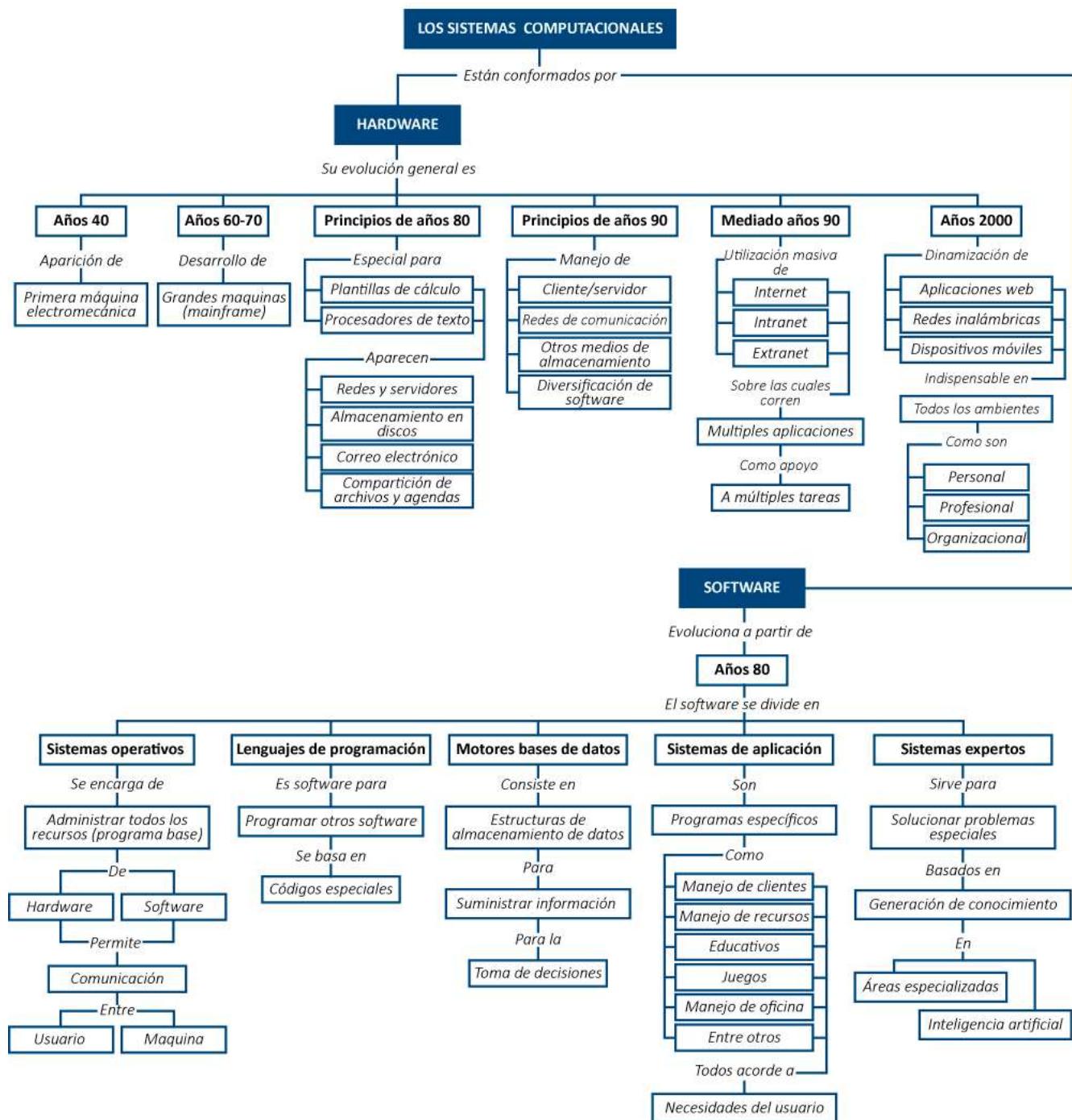
En la actualidad las Bases de Datos se han convertido en los almacenamientos y bodegas de datos (Data Warehouse) para la extracción de la información acorde a las reglas de cada negocio, siendo la información, el activo más importante de cualquier organización. Entre los motores de bases de datos más utilizados en el medio, se tiene Oracle, SQL Server, Informix, My SQL entre otros, que de acuerdo a la necesidad en seguridad, capacidad, costo es elegida por las diferentes organizaciones, instituciones y negocios para la gestión de su información.

Los Sistemas de Aplicación

Los aplicativos de software equivalen a los demás productos de software que cumplen con un fin, ya sean genéricos o diseñados a la medida para necesidades específicas y que apoyan los procesos de las empresas como por ejemplo un software educativo, graficadores, herramientas de oficina, manejo de clientes, manejo de recursos, juegos, entre otros.

Los Sistemas Expertos

Conocido como una aplicación con capacidad para solucionar problemas que tiene que ver con el manejo de conocimientos sobre determinados temas o áreas específicas, dichos sistemas expertos están relacionados con la Inteligencia artificial, tema que será abordado más adelante.



2.1.2 OBJETIVO GENERAL

Analizar los diferentes tipos de software y su aplicación acorde a las necesidades en cuanto a la administración de las máquinas, desarrollo de nuevos productos informáticos, manejo de datos y aplicaciones que tienen que verse con las reglas del negocio, así como lo concerniente al software especializado para su elección en el momento que se requiera.

2.1.3 OBJETIVOS ESPECÍFICOS

- Identificar la función que cumplen los Sistemas Operativos como administrador de los recursos de software y hardware.
- Comprender la función que cumplen los lenguajes de programación en el desarrollo de nuevos productos informáticos, buscando dar solución a necesidades de software.
- Identificar la importancia de las bases de datos, como bodegas de almacenamiento para el suministro de información bajo niveles de seguridad y calidad de dicha información para la toma de decisiones.
- Analizar la importancia que tienen los diferentes aplicativos como herramienta ágil para el eficiente desarrollo de cada una de las actividades personales, profesionales y empresariales.
- Reconocer la importancia de los sistemas expertos como herramientas de apoyo a la gestión del conocimiento, orientados al soporte en situaciones especializadas relacionadas con la inteligencia artificial.

2.2 LOS SISTEMAS OPERATIVOS (SO)

Para comprender mejor la temática de los sistemas operativos, se recomienda observar el video (Historia de los sistemas operativos y Diferencias entre sistemas operativos), donde se explica la evolución que estos han sufrido a través de la historia y su incorporación en los diferentes aparatos electrónicos, así como los diversos sistemas operativos que han sido utilizados y aquellos que se siguen utilizando actualmente.



Historia de los sistemas operativos [Enlace](#)

Basado en el video anterior responder:

¿Quién implementó el primer sistema operativo?

¿Qué es multiusuario?

¿Qué es multiprocesador?

¿Qué es multiprogramación?

¿Qué es un sistema distribuido?

¿Cuáles son las principales diferencias entre los sistemas operativos Windows, Mac, Linux?

El sistema operativo (SO), es el programa base sobre el cual giran los demás programas del computador, sin éste el computador no funciona, ni los aplicativos o programas que se instalen. Un sistema operativo, es un [programa](#) o conjunto de programas que tiene como función gestionar todos los recursos de hardware y software, siendo el primer programa en reconocer la máquina y cada uno de los dispositivos que se encuentran conectados en el momento de iniciar su funcionamiento, actuando de forma privilegiada a comparación de los demás programas que se encuentren instalados en dicho equipo.

El sistema operativo se encarga de la gestión de todos los recursos como son los diferentes procesos, utilización de la memoria, manejo de archivos, procesos de comunicación, seguridad, control de dispositivos, manejo de concurrencia entre otras funciones, así:

2.2.1.1 GESTIÓN DE PROCESOS:

Cada vez que el sistema ejecute un proceso, requiere que se creen los procesos, se destruyan, se reanuden liberando memoria cuando ya no esté en uso, por lo tanto el Sistema Operativo ofrece mecanismos que permitan que dichos procesos se ejecuten de manera adecuada, acorde a su capacidad. Es quien controla los tiempos para que se realicen ciertas tareas, así como la cantidad de recursos que se pueden consumir.

2.2.1.2 GESTIÓN DE LA MEMORIA:

Su función principal radica en el manejo sobre la disposición de la memoria, utilización y control de los recursos responsables de dicha utilización, estando en capacidad de decidir qué procesos tendrán prioridad de consumo de memoria, además puede asignar memoria libre y generar alertas cuando exista agotamiento de esta.

La memoria se divide en memoria principal y secundaria, la principal controla el almacén de datos principal de la computadora con accesos compartidos para los diferentes procesos de la Unidad Central de Procesamiento (CPU) y los requerimientos de los diferentes dispositivos de entrada y salida que van conectados al sistema central. Dicha memoria principal es volátil o temporal. En cuanto a la memoria secundaria(disco duro, flexibles, USB, entre otros), su utilidad tiene que ver con el almacenamiento de todos los programas e información de los usuarios, donde el sistema operativo se encarga de gestionar el espacio libre del disco, la asignación del almacenamiento conservando un orden, así como la detección de fallos en dichos almacenamientos.

2.2.1.3 GESTIÓN DE ARCHIVOS:

El sistema operativo se encarga de permitir la manipulación de archivos y carpetas, aplicando sobre estos, operaciones como creación, eliminación, copias, cambiarlos de un lugar a otros, realización de copias de seguridad para la protección de datos, ubicando dicha información en lugares específicos de la memoria secundaria.

2.2.1.4 GESTIÓN DE PROTECCIÓN O SEGURIDAD:

El sistema operativo se encarga de verificar los accesos autorizados y no autorizados al sistema, permite definir los mecanismos para el control de acceso a recursos del sistema a través de contraseñas, llaves, configuración de dispositivos, entre otros.

2.2.1.5 GESTIÓN DE COMUNICACIONES:

Realiza el control de envío y recepción de información a través de dispositivos de red, manteniendo los puntos de comunicación de dichos dispositivos de red entre aplicaciones o programas que se ejecutan tanto localmente como remotamente, buscan con ello que exista transparencia y fiabilidad en la información.

2.2.1.6 GESTIÓN DE CONCURRENCIA:

Cuando el sistema informático tiene muchos programas y opera con un solo procesador, los diferentes procesos comparten recursos y tiempos intercalándose lo que da la sensación de que los procesos son ejecutados simultáneamente. El procesamiento concurrente se da cuando se intercalan y se superponen los procesos, donde se pueden presentar problemas que tienen que ver con la velocidad para las ejecuciones, por lo tanto pueden surgir problemas que tienen que ver con el compartimiento de recursos globales, siendo difícil para el sistema operativo la gestión para la asignación eficiente de recursos que den respuestas adecuadas a las solicitudes del sistema.

Los problemas de concurrencia aparecen tanto para sistemas de único procesado como para aquellos de múltiples procesados, por lo tanto el sistema operativo debe dar continuidad a los distintos procesos activos, pero de igual forma debe asignar y/o retirar recursos a cada proceso como el control de tiempos, de memoria, de archivos, de dispositivos de entrada y/o salida. Otra función que le compete al control de la concurrencia es la protección de los datos y de los recursos físicos contra accesos a otros procesos, evitando su corrupción.

En información encontrada en el link lrmdavid@exa.unne.edu.ar, se explica claramente el proceso de concurrencia, donde se explica que:

Los procesos concurrentes entran en conflicto cuando compiten por el uso del mismo recurso; dos o más procesos necesitan acceder a un recurso durante su ejecución .Cada proceso debe dejar tal y como esté el estado del recurso que utilice, por lo tanto la ejecución de un proceso puede influir en el comportamiento de los procesos que compiten. Por Ej. Si dos procesos desean acceder a un recurso, el sistema operativo le asignará el recurso a uno y el otro tendrá que esperar, es así que cuando hay procesos en competencia, se deben solucionar tres problemas de control: la necesidad de exclusión mutua. Suponiendo que dos procesos quieren acceder a un recurso no compatible. A estos recursos se les llama "recursos críticos" y la parte del programa que los utiliza es la "sección crítica" del programa. Es importante que sólo un programa pueda acceder a su sección crítica en un momento dado, por lo tanto hacer que se cumpla la exclusión mutua provoca un interbloqueo.

Otro problema es la inanición si tres procesos necesitan acceder a un recurso, P1 posee al recurso, luego lo abandona y le concede el acceso al siguiente proceso P2, P1 solicita acceso de nuevo y el sistema operativo concede el acceso a P1 YP2 alternativamente, se puede negar indefinidamente a P3 el acceso al recurso, donde el control de competencia involucra al sistema operativo, porque es el que asigna los recursos.

2.2.1.7 GESTIÓN DE DISPOSITIVOS DE ENTRADA Y SALIDA (E/S):

Se encarga de la gestión de reconocimiento de los diferentes dispositivos (pantalla, teclado, mouse, impresora, scanner, plotter, puertos, entre otros), el control sobre su utilización, planificación de las peticiones de utilización de dichos recursos, detección de conflictos y posible corrección, entre otros controles que se pueden ejercer para que dichos recursos presenten consistencia con los demás recursos de E/S.

Los sistemas operativos poseen herramientas complementaria que le ayudan a gestionar el sistema de cómputo, entre ellos tenemos los drivers que son los controladores del hardware, es decir, interactúan entre el sistema operativo y los dispositivos para que estos se comporten correctamente, otro aliado de los sistemas operativos tiene que ver con software que se encarga de controlar y en muchos casos eliminar aquellos programas que han sido creados para causar daños a otros programas (virus informático), pero de igual forma aparecen otros

programas que reciben el nombre de antivirus, programas que se convierten en espías para tratar de que el sistema trabaje de forma consistente.

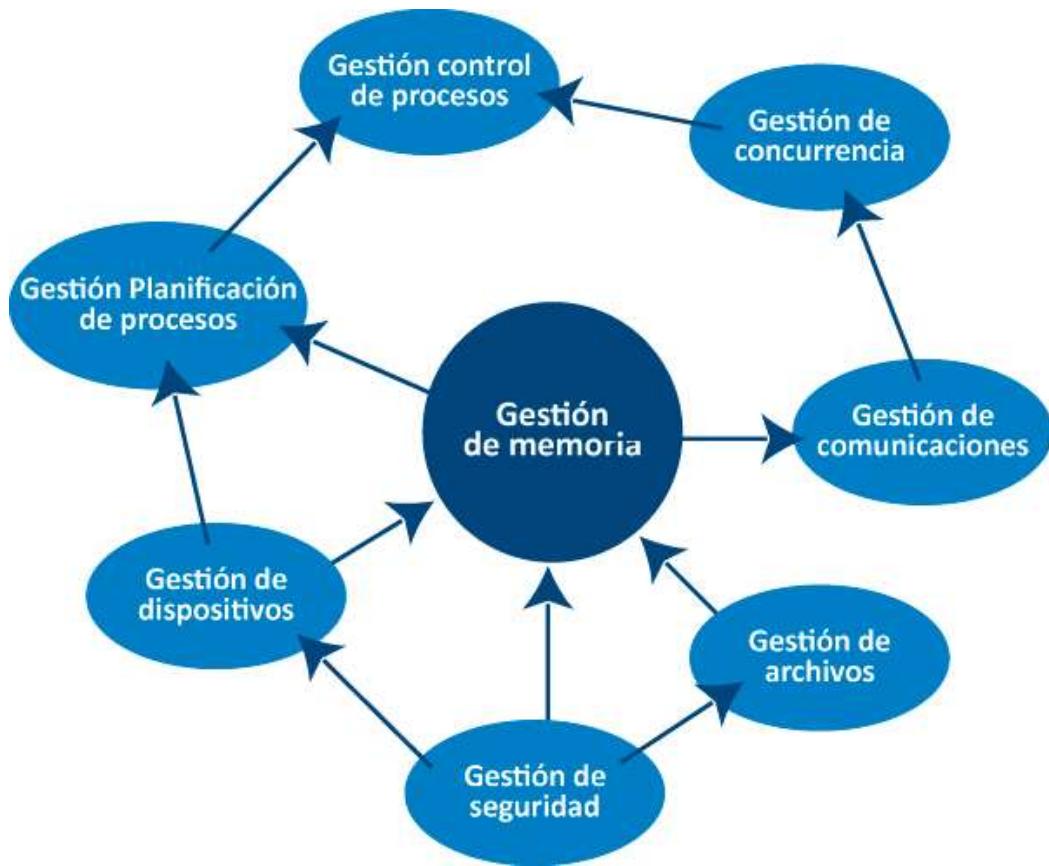
Para ampliar la información sobre los procesos que ejerce el sistema operativo sobre el equipo de cómputo, se recomienda visitar el siguiente link http://es.wikipedia.org/wiki/Sistema_operativo, así como el Video



Los sistemas operativos y la tecnología [Enlace](#)

En la siguiente gráfica se puede observar las principales gestiones que realiza el sistema operativo, donde dichas gestiones se encuentran debidamente relacionadas con el objetivo de administrar todos los recursos de hardware y software, los que se encuentran interrelacionados y trabajan en conjunto, es así como la falla de uno de ellos afecta significativamente todo el sistema.

Gráfica Gestiones del Sistema Operativo



En sus inicios los Sistemas Operativos, eran de uso exclusivo para los equipos de cómputo, pero en la actualidad es normal encontrarlos instalados en múltiples aparatos electrónicos como cámaras, celulares, robots, entre otros.

2.2.2 EJERCICIO DEL TEMA

- 1) Observar la siguiente diapositiva <http://www.slideshare.net/angyespinosa86/capitulo-1-353609>
- 2) Elaborar un mapa conceptual que muestre elementos importantes que sobre el sistema operativo, sus servicios, evolución y componentes.
- 3) Investigar qué función cumple el Núcleo o Kernel del Sistema Operativo.

PISTAS DE APRENDIZAJE**Traer a la memoria:****UN SISTEMA OPERATIVO ES:**

El software básico de una computadora que permite la comunicación entre el resto de programas del computador, los componentes de hardware y el usuario. Sus principales funciones consisten en administrar los recursos físicos, coordinar, organizar archivos en los diferentes almacenamientos. Entre los sistemas operativos más utilizados tenemos DOS, WINDOWS, LINUX, MAC.

2.3 TEMA 2 LOS LENGUAJES DE PROGRAMACIÓN

Un Lenguaje de Programación está conformado por una serie de reglas sintácticas y semánticas que son utilizadas por el programador (persona que conoce el código del lenguaje de programación) y a través de las cuales creará un programa, un subprograma o componente. Dichas instrucciones son conocidas como código fuente. Normalmente los lenguajes de programación de última generación tienen incorporado además del lenguaje de programación, el lenguaje compilador que se encarga de convertir las instrucciones digitadas por el programador en códigos entendibles por la máquina (códigos binarios, compuestos por ceros y unos) para posteriormente ser empaquetados y distribuidos al usuario final como un producto de software que cumple con unos requerimientos del usuario final y que este no podrá modificar.

A continuación se muestra un ejemplo, donde se utiliza seudocódigo, que permite realizar un procedimiento consistente en dar unos valores de entrada al sistema, este los cuenta, y devuelve el promedio de dichos valores que son considerados como válidos. Datos como este son transcritos utilizando las instrucciones propias del lenguaje y para su ejecución deben ser convertidas utilizando el mismo lenguaje de programación en lenguaje de máquina y así el sistema se encarga de arrojar las salidas correspondientes, acorde a los requerimientos del usuario.

```
Procedimiento Media(valor,mínimo,máximo,media,tot_entrada,tot_validos)
    i = 1, tot_entrada = 0, tot_validos = 0, suma = 0 (1)
    Mientras valor[i]<>-999 (2) And tot_entrada<100 (3) Hacer
        tot_entrada = tot_entrada + 1 (4)
        Si valor[i]>=mínimo (5) And valor[i]<=máximo (6) entonces
            tot_validos = tot_validos + 1      (7)           suma = suma
            + valor[i]
        Fin_si
        i = i + 1 (8)
    Fin_Mientras (9)
    Si tot_validos>0 (10) entonces
        media = suma/tot_validos (11)
    Sino
        media = -999 (12)
    Fin_si (13)
```

Fin_Media

2.3.1 EN LA ACTUALIDAD LOS LENGUAJES DE PROGRAMACIÓN SE CLASIFICAN EN:

Lenguaje de bajo nivel: Es el código basado en lenguaje de máquina o sistema binario, muy cercano a la forma como la máquina interpreta dicha información.

Lenguaje de nivel medio: Son los lenguajes que se encuentran construidos de tal forma que sus términos se encuentran entre el lenguaje de máquina y el lenguaje natural o sea aquellas instrucciones que comprende fácilmente el programador.

Lenguaje de alto nivel: Consiste en los lenguajes que están compuestos por elementos del lenguaje natural, es decir el humano, comprensible para los interesados en aprender las diferentes sintaxis para desarrollar productos informáticos. Normalmente dichos lenguajes están escritos en idioma inglés, son conocidos como los lenguajes de cuarta generación (4GL).

2.3.2 BREVE EVOLUCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación al igual que cualquier otra tecnología considerada como práctica, ha sufrido evolución, es así como en la década de los años 40 se inicia el desarrollo de los lenguajes ensambladores que entran a reemplazar los unos y ceros (1,0) considerados como el lenguaje de máquina o código binario, surgiendo representaciones e instrucciones más fáciles de recordar como MOVE, ADD, entre otras y que posteriormente dichas representaciones eran convertidas a instrucciones binarias, de igual forma dichos lenguajes eran difíciles de comprender, ya que se acercaban más al lenguaje de máquina que al humano y eran considerados como lenguajes de bajo nivel.

Entre los años de 1954 y 1957, entra en funcionamiento el lenguaje Fortran por IBM, mostrando su superioridad sobre los lenguajes ensambladores, haciendo más fácil el manejo de las operaciones básicas buscando acercar más la forma de programar al lenguaje humano. Ya en 1958 el Fortran es mejorado, con el lenguaje Algol-58, quien toma como base la sintaxis (instrucciones) del Fortran y mejora su notación (sistema de signos convencionales que se adoptan y utilizan para expresar determinados conceptos de una disciplina).

En 1960, para procesar datos a gran escala surge el lenguaje de programación COBOL que fue utilizado por bancos, compañías y dependencias oficiales, así mismo aparece el lenguaje de programación LISP que únicamente ejecuta funciones, que pueden llamarse a sí mismas cuantas veces sean necesarias, lenguaje especial para inteligencia artificial (Ciencia e ingeniería de hacer máquinas inteligentes). También aparecen otros lenguajes como el FORTH, APL, PL/I, siendo muy complicados de utilizar y por lo tanto no tuvieron mayor fuerza.

Ya a partir de 1964, aparece el BASIC por John Kemey y Thomas Kurtz, quienes lo propusieron como un lenguaje propio para la enseñanza, al igual que el lenguaje PASCAL en 1970 por el profesor Niklaus Wirth, también el lenguaje C con el cual se desarrolla el Sistema Operativo UNIX, así como el PROLOG pensado para desarrollar Inteligencia Artificial.

A partir de los años 80 se trabaja con el Smalltalk, orientado a la programación orientada a objetos (POO), con el SIMULA-67, se aplicó dicho concepto al lenguaje C sacando una nueva versión llamada lenguaje de programación C++.

De SQL surge dBase, y posteriormente el Fox, Clipper, RBASE, Oracle, INFORMIX, etc, al conjunto de estos programas se los denominó (Lenguajes de Cuarta Generación) 4GL. Actualmente los lenguajes se orientan a objetos con componentes reutilizables con contenido multimedia (imagen, video, texto, sonido), que funcionen bajo protocolo internet, especiales para programar dispositivos móviles con lenguajes como Java, lenguajes bajo plataforma .NET, PHP, proporcionando potentes herramientas que permiten la realización de software o aplicaciones ágiles con una gran comprensión por los desarrolladores, acordes a necesidades del contexto.

En el caso de JAVA aparece en 1995 de forma libre que permite crear aplicaciones en todo tipo de equipos de cómputo y con cualquier sistema operativo, siendo uno de los lenguajes de programación.

Respecto a PHP que aparece en 1997, como lenguaje especial para la programación de sitios web dinámicos

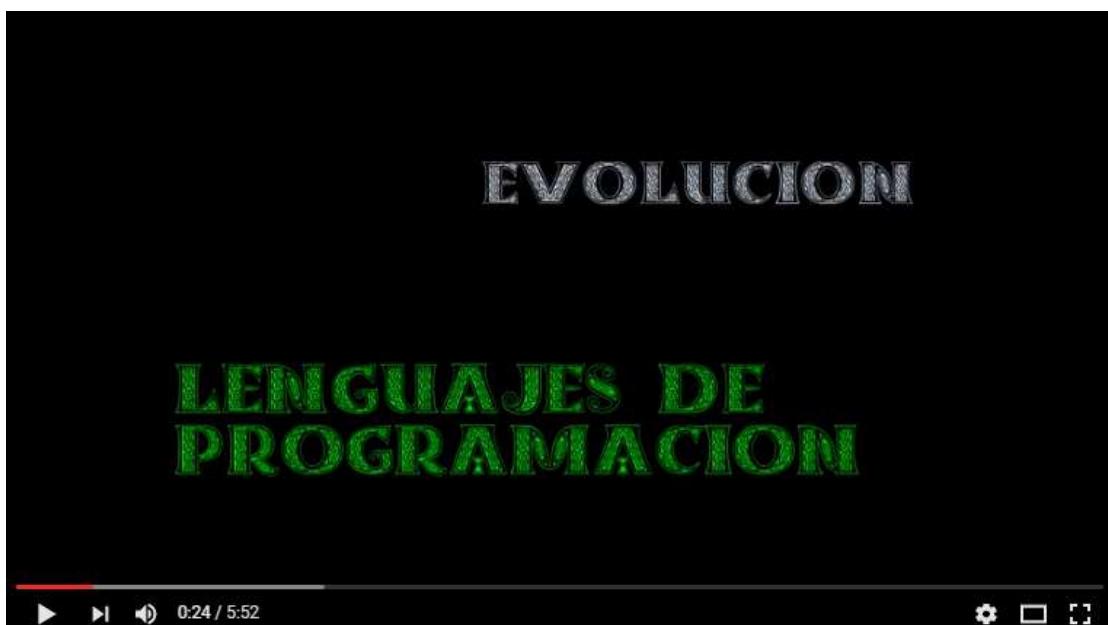
En el año 2001, aparece C#, como evolución del lenguaje C++, pero como software licenciado. La siguiente codificación pertenece al lenguaje de programación C#, donde se puede observar instrucciones propias del lenguaje e instrucciones propias del programador.

```
using System;  
  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Text;  
using System.Windows.Forms;  
namespace Veterinaria  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }  
        private void salirToolStripMenuItem_Click(object sender, EventArgs e)  
        {  
            Application.Exit();  
        }  
        private void coToolStripMenuItem_Click(object sender, EventArgs e)  
        {  
            Raza obra = new Raza();  
            obra.MdiParent = this;  
            obra.WindowState = FormWindowState.Normal;  
            obra.Show();  
        }  
    }  
}
```

```
private void pacienteToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Paciente obra = new Paciente();
    obra.MdiParent = this;
    obra.WindowState = FormWindowState.Normal;
    obra.Show();
}
private void consultaToolStripMenuItem_Click(object sender, EventArgs e)
{
    Consulta obra = new Consulta();
    obra.MdiParent = this;
    obra.WindowState = FormWindowState.Normal;
    obra.Show();
}
private void pacienteToolStripMenuItem_Click(object sender, EventArgs e)
{
    Dueño obra = new Dueño();
    obra.MdiParent = this;
    obra.WindowState = FormWindowState.Normal;
    obra.Show();
}
private void Form1_Load(object sender, EventArgs e)
{
}
private void menuStrip1_ItemClicked(object sender, ToolStripItemClickedEventArgs e)
{
}
}
```

2.3.3 EJERCICIO DEL TEMA:

- 1) Observar el video sobre la evolución de los Lenguajes de Programación en el link



Evolución de los lenguajes de programación [Enlace](#)

- 2) Elaborar un cuadro que posea 3 columnas, donde se debe ubicar año, nombre del lenguaje, características del lenguaje.

PISTAS DE APRENDIZAJE

Traer a la memoria:



UN LENGUAJE DE PROGRAMACIÓN:

Es un sistema de software que se utiliza para crear otros sistemas de software, se basa en códigos, instrucciones y sintaxis, que permiten traducir instrucciones del desarrollador o programador en códigos binarios o lenguaje de máquina a través de la compilación que tiene que ver con una especie de traductor entre lo que el programador digita y la forma como el computador lo interpreta.

2.4 TEMA 3 LOS MOTORES DE BASES DE DATOS

Los motores de bases de datos o Sistemas de Gestión de Bases de Datos (SGBD), son programas que permiten la gestión de estructuras que almacenan datos, permiten modificarlos en el momento que se requiera, eliminarlos de ser necesario, extraerlos de forma ágil y estructurada, además de la administración de ellos con el fin de garantizar consistencia, seguridad y confiabilidad sobre dichos datos e información.

El SGBD puede gestionar múltiples sitios conectados por una red, buscan ejercer control sobre los datos, donde muchos de estos emplean la arquitectura cliente-servidor, especiales para unir bases de datos que se encuentren físicamente distribuidas en sitios diferentes, es el caso de las sucursales que poseen las empresas ubicadas geográficamente distantes las unas de las otras. Ver video sobre SBBD



Bases de Datos, Sistema de gestión [Enlace](#)

Elaborar un mapa conceptual, donde se muestre el propósito de los SGBD, sus objetivos, sus diferentes niveles de abstracción, las ventajas y desventajas y los productos que permiten gestionar las bases de datos, entre los que se encuentran SQL Server, Oracle, Paradox, entre otros.

Los SGBD por sí solos no presentan mayor utilidad y se consolidan, en el momento que se elaboran estructuras de bases de datos. Antes de abordar las bases de datos, es necesario tener presente los siguientes conceptos que se constituyen elementos importantes para estas:

- Dato: es conocido como la mínima expresión sin sentido propio y que forma parte de las bases de datos, por ejemplo:



Por tanto los datos por sí solos no presentan significados importantes, ni hacen alusión claramente a situaciones específicas, por lo tanto cuando se presentan de esta forma no son prácticos.

- **Campo:** Se considera como el espacio físico de memoria que utiliza el sistema para ubicar el dato y que es controlado por el SGBD y el SO, por ejemplo el campo nombre donde se puede almacenar cualquier nombre.
- **Registro:** Consiste en un conjunto de datos que se encuentran relacionados y que ofrecen significado para ser utilización con fines específicos.

58.551.214	Juan Carlos	Quintero Gallego	3000.000	Administrador
22.456.213	Claudia Patricia	Arboleda Ángel	1200.000	Secretaria
24.897.765	Sara	Hurtado Carvajal	5000.000	Gerenta

En el caso anterior, se tiene claro que en la empresa XX tiene registros con significado, por ejemplo la secretaria se llama Claudia, devenga un salario de un millón doscientos mil pesos (1200.000), es de apellido Arboleda Ángel y tiene una identificación apoyada en un número de cédula, por lo tanto con los datos de este registro se podrían realizar otros análisis acorde a la necesidad.

- **Tabla:** Equivale a un conjunto de registros, que conservan una estructura compuesto por filas y columnas, debidamente organizadas para su mayor entendimiento. En la siguiente tabla se puede realizar lectura por columnas donde aparecen campos y por filas donde aparecen registros, permitiendo tener mayor claridad sobre la información que se contiene allí, por ejemplo podemos observar que existe dentro de la empresa XX, cargos como Administrador, Secretaria y Gerenta, que tienen una identificación propia, así como sus respectivos nombres y apellidos; y devengan un salario. Es de anotar que las columnas son los campos y deben estar identificados por un nombre (el campo cédula, nombres, apellidos, salario, cargo).

CÉDULA	NOMBRES	APELLIDOS	SALARIO	CARGO
58.551.214	Juan Carlos	Quintero Gallego	3000.000	Administrador
22.456.213	Claudia Patricia	Arboleda Ángel	1200.000	Secretaria
24.897.765	Sara	Hurtado Carvajal	5000.000	Gerenta

Teniendo claro los anteriores elementos el dato, el campo, el registro y la tabla, pasaremos a la Base de Datos (BD), que se entiende como un conjunto de tablas (dos o más), interrelacionadas entre sí, que poseen datos pertenecientes a un mismo contexto y que se encuentran almacenados sistemáticamente para su posterior uso, por ejemplo el sistema de matrículas de una universidad se puede considerar una base de datos, el de una biblioteca, el de despacho de buses en una terminal de transporte, entre otros. Hoy se hace impensable el manejo de los diferentes sistemas empresariales, sin las bases de datos que se constituyen en la bodega de datos para realizar cualquier tipo de proceso, convirtiéndose en uno de los activos más valiosos de cualquier institución, empresa u organización. En la siguiente Base de Datos compuesta por las tablas cargo y empleado, se puede observar la relación que existe entre ellas, donde es claro que entre los empleados que tiene la empresa XX, existen dos secretarias que se identifican con el código número 200 y que sus nombres equivalen Claudia Patricia y María Isabel respectivamente. De igual forma se podría sacar cualquier tipo de información a través de las consultas que se realicen a la base de datos. (Observar el campo código, en la tabla empleado)

Tabla cargo

CÓDIGO	CARGO
100	Administrador
200	Secretaria
300	Gerenta
400	Líder de Proyectos

Tabla empleado

CÉDULA	NOMBRES	APELLIDOS	SALARIO	CARGO
58.551.214	Juan Carlos	Quintero Gallego	3000.000	100
22.456.213	Claudia Patricia	Arboleda Ángel	1200.000	200
24.897.765	Sara	Hurtado Carvajal	5000.000	300
45.678321	Roberto	Mosquera Rentería	3500.000	400
21.965.954	María Isabel	Cano Yepes	1200.000	200

Para lograr la estandarización de las bases de datos, en cuanto a su estructura y conformación, es necesario aplicar ciertos modelos que a través de la historia de las bases de datos han sufrido modificaciones y evoluciones significativas, donde hoy para la estructuración de las bases de datos se trabaja normalmente con el modelo relacional, con tendencias al modelo objetual, aunque éste último aún no se ha implementado de forma

adecuada de tal forma que permita la migración de las bases de datos estructurales a las bases de datos objetuales. Por lo tanto nos concentraremos en el modelo relacional para las bases de datos estructurales.

Modelo Relacional: Consiste en una estructura que permite modelar bases de datos, dichas estructuras están compuestas por unos elementos fundamentales que permiten su comprensión como son:

- **La entidad:** Se refiere a cualquier objeto concreto o abstracto que posea significado y características propias, en el caso un objeto abstracto se puede referir a los cargos que sin tener estructura física, se puede extraer información de él otros ejemplo para objetos abstractos pueden ser las carreras profesionales que ofrecen las universidades, las materias o asignaturas, los servicios que presta una empresa entre otros, en lo que tiene que ver con objetos concretos se refiere a aquellos que tienen representación física como los empleados, los estudiantes, los docentes, el hardware, entre otros de los que se pueda extraer datos. La representación de una entidad se hace mediante un rectángulo al cual se le da un nombre en singular. Así:

CARRERA	ESTUDIANTE
Atributos	Atributos

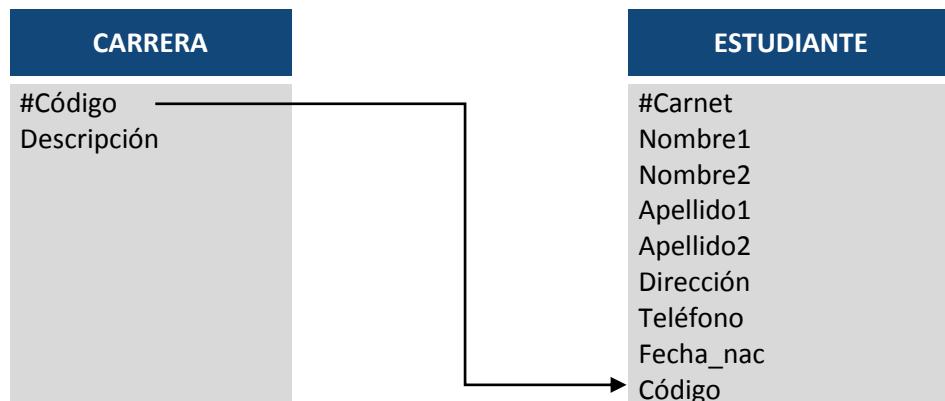
- **Atributos:** Consiste en las característica o elementos que pertenecen a las entidades y dentro de las cuales se pueden ubicar datos, útiles de acuerdo a la necesidad de información así

CARRERA	ESTUDIANTE
Código Descripción	Carnet Nombre1 Nombre2 Apellido1 Apellido2 Dirección Teléfono Fecha_nac

- **Clave primaria o principal:** Consiste en tomar un atributo y convertirlo en un campo especial que no permita repetir datos, ya que se constituye como único y no permitirá que dichos datos sean duplicados, por ejemplo para la entidad estudiante la clave primaria debe ser carnet, ya que los estudiantes dentro de una institución no pueden tener carnet con números repetidos, siendo un código único que identifica a un solo estudiante. Si se estuviese manejando la información de clientes, la clave primaria sería la cédula, ya que se constituye en un dato único. Las claves primarias para efecto del modelo relacional se identifican con el símbolo de número ubicado al lado izquierdo del nombre del atributo (#carnet)

CARRERA	ESTUDIANTE
#Código Descripción	#Carnet Nombre1 Nombre2 Apellido1 Apellido2 Dirección Teléfono Fecha_nac

■ **Clave foránea:** Tiene que ver con una clave primaria o principal, pero ubicada en otra entidad y que sirve para relacionar dos o más entidades, la que recibe el nombre de clave foránea o extranjera. Dicha relación es la que permite en las bases de datos extraer información de estas, acorde a las reglas del negocio. En la siguiente gráfica se puede observar que en la entidad carrera el código es la clave primaria, pero cuando aparece en la entidad estudiante el mismo código, se convierte en clave foránea.



Es de aclarar que la clave primaria en carrera no se puede repetir ya que una carrera no se puede registrarse más de una vez, pero cuando aparece en la entidad estudiante ya se puede repetir porque la misma carrera sí la pueden tomar varios estudiantes. Para comprender mejor el concepto anterior se utilizan las siguientes entidades abiertas que equivaldrían a las tablas ya diseñadas dentro de la base de datos y gestionadas a través del SGBD. En las tablas que se muestran existen tres estudiantes que estudia medicina, uno ingeniería, dos administración, en lo referente a derecho, aún no se han matriculado estudiantes, lo que aclara el concepto de una clave foránea.

Tabla Carrera

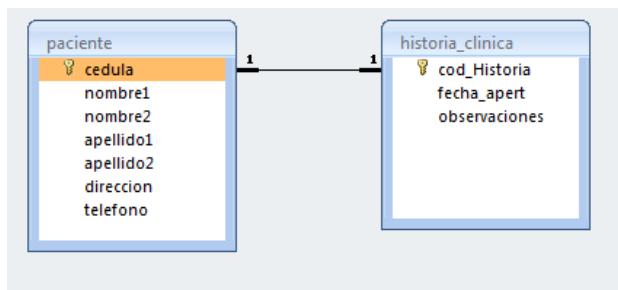
CLAVE PRIMARIA	CÓDIGO	DESCRIPCION
	01	Medicina
	02	Ingeniería
	03	Administración
	04	Derecho

Tabla estudiante

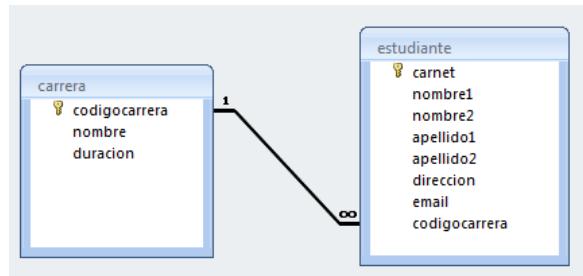
CARNET	NOMBRE1	NOMBRE2	APELLIDO1	...	CODIGO	← CLAVE FORÁNEA
10	Clara	Patricia	Herrera		02	
20	Simón	Eliécer	Cano		01	
30	Raúl	De Jesús	Restrepo		01	
40	Juan	Camilo	Rivera		03	
50	Carlos	Andrés	Castro		03	
60	Roberto		Benavides		01	

Relaciones: Las relaciones en el modelo relacional, tiene que ver con el tipo de asociación entre dos entidades, que de acuerdo a las necesidades del sistema de software son creadas con el fin de que se puedan manejar los datos de forma adecuada. Las relaciones entre las entidades son de tres tipos:

Relación de uno a uno (1:1): Consiste en el tipo de relación, donde un dato solo tiene coincidencia con otro y no tiene la posibilidad de relacionarse de otra forma. Es el caso en una empresa cuando a través de un empleado sólo se puede afiliar una esposa a la EPS (Empresa Promotora de Salud) y esa esposa sólo puede estar afiliado por un esposo, otro ejemplo tiene que ver con las historia clínicas de los pacientes dentro de ese establecimiento, donde un paciente sólo puede tener una historia clínica y esa historia clínica sólo puede pertenecer a un paciente. Así:



Relación de uno a muchos (1:N): Consiste en que los dato se pueden repetir en una entidad, pero en la otra no, su relación se orienta desde la clave primaria hacia la clave foránea, donde en la entidad que posee la clave foránea existe la posibilidad de repetir dichos datos. Así:



En este caso se leería así en una carrea se pueden matricular varios estudiantes, pero un estudiante sólo se puede matricular en una carrera.

Relación de muchos a muchos: (N:M) Cuando se da este tipo de relación, es necesario realizar una entidad de por medio, con el fin de garantizar que los datos tengan la movilidad que se espera para que el SGBD, pueda procesarlos y dar la información que se solicite a través de las consultas que se ejerzan sobre dichos datos.

PISTAS DE APRENDIZAJE

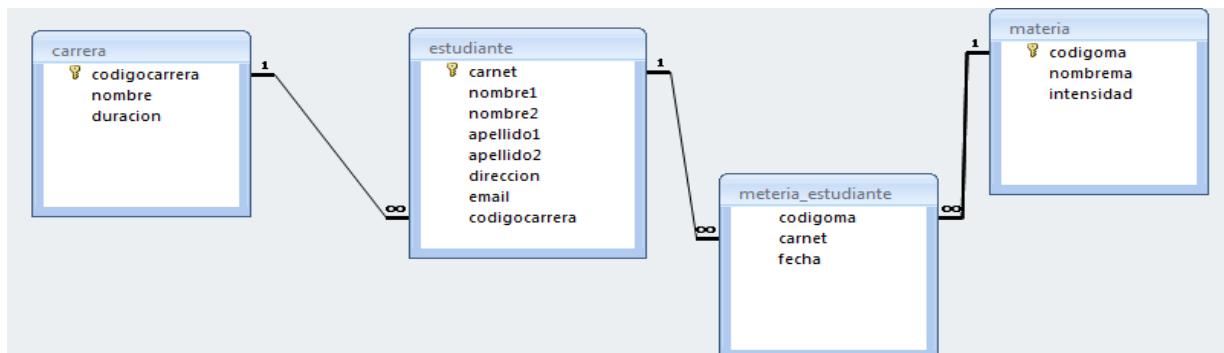
Traer a la memoria:



Recuerde que: en el modelo de datos relacional, una entidad, se convierte en una tabla, ya cuando se crea en el SGBD, un atributo se convertirá en un dato.

Continuando con el ejemplo de la base de datos académica, se puede decir que un estudiante puede tomar varias asignaturas o materias y que cada materia puede ser tomada por varios estudiantes, por lo tanto en este caso existe relación de muchos a muchos y requiere que se elabore una entidad intermedia que en este caso corresponde a la entidad materia_estudiante, así:

Modelo de datos relacional



Visualizando los datos de este modelo relacional, ya diseñando a través del SGBD, que es quien permite su manipulación, se permitirían analizar los datos así:

Se ha registrado en el SGBD, cuatro carreras que no pueden ser repetidas, ya que código de carrera es una clave primaria, pero cuando se visualiza la información en la tabla estudiante, el código de carrera se puede repetir, ya que en esta tabla corresponde a una clave foránea, con tipo de relación de 1:N (uno a muchos), por lo tanto, existen tres estudiantes que pertenecen a Informática, dos a Administración, y uno a Derecho, pero cada estudiante sólo puede pertenecer a una carrera como política de la institución. En el caso de la relación que existe entre estudiante y carrera, sabemos que consiste en una relación de N:M (de muchos a muchos). Al visualizar los datos de la tabla intermedia (materia_estudiante), se puede observar que la estudiante clara matriculó dos asignaturas o materias como son Cálculo e Ingeniería del Software, Carlos tomó Cálculo y Base de Datos, Nicolás tomó Cálculo, Ingeniería del Software y Sistemas de Información, en el caso de Roberto sólo tomó Sistemas de Información.

Tabla carrera

codigocarrera	nombre	duracion
10	Derecho	11 semestres
20	Informática	10
30	Administració	10
40	Educación	10

Tabla estudiante

carnet	nombre1	nombre2	apellido1	apellido2	direccion	email	codigocarre
0001	Clara	Andrea	Osorio	Yepes	Calle 34 56-12	cosorio@gmail.com	10
0002	Carlos	Eliécer	Cano	Restrepo	Carrera 34 65-12	crestrepo@gmail.com	20
0003	Nicolás		Martínez	Mira	Carrera 32 78-23	Nmartinez@hotmail.com	30
0004	Roberto	de J	Gonzalez	Rentería	Carrera 50 46-22	rgonzalez@hotmail.com	20
0005	Isabella	María	Hurtado	Medina	Carrera 77 90-32	lhurtado@gmail.com	20
0006	Jair	José	Rivera	Ossa	Caller 34 21-65	jrivera@hotmail.com	30

Tabla materia

codigoma	nombrema	intensidad
11	Cálculo	64 horas
22	Ingeniería del Software	64 horas
33	Bases de Datos	64 horas
44	Sistemas de Información	48 horas
55	Negocios Internacionales	64 horas
66	Ética	32 horas

Tabla materia estudiante

codigoma	carnet	fecha
11	0001	03/03/12
22	0001	03/03/12
11	0002	04/04/04
33	0002	04/04/12
22	0003	05/04/12
11	0003	05/04/12
44	0003	05/04/12
44	0004	07/04/12

Los modelos de datos, en este caso el modelo relacional, permiten estructurar bases de datos útiles que faciliten la manipulación de la información y que satisfagan las necesidades informacionales de los clientes, pero un diseño de base de datos mal estructurado, puede ser nefasto a la hora de tomar decisiones sobre datos erróneos, de ahí la importancia de realizar una buena estructuración de dichas bases de datos.

Para estandarizar las bases de datos, es necesario tener conocimientos de las formas normales (FN), con las que se busca dar una serie de reglas que permiten el diseño de bases de datos útiles. Existen varias formas normales, pero en la parte práctica, se utilizan principalmente las tres primeras formas normales (1FN, 2FN, 3FN), que se pueden visualizar detalladamente en el siguiente video.



EJEMPLO DE NORMALIZACION [Enlace](#)

PISTAS DE APRENDIZAJE

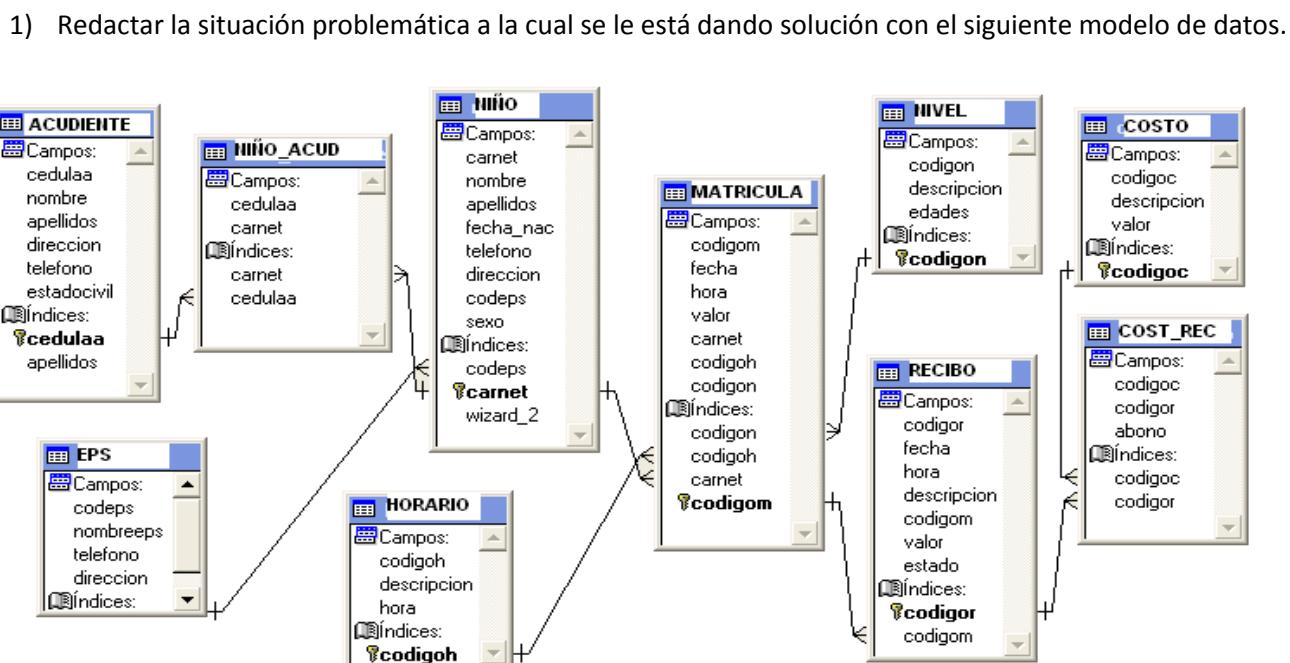
Traer a la memoria:

Los SGBD (sistemas de gestión de bases de datos)

Equivalen a un tipo de software que fue diseñado para administrar, las bases de datos que son diseñadas de acuerdo a necesidades normalmente empresariales. Los SGBD, permiten crear las estructuras de las bases de datos, manejar independencia de datos manteniendo la información consistente, ofrece niveles de seguridad, ofrece facilidad en el proceso de transacciones, de respaldos de información, suministra interfaces para la extracción o consulta de dichos datos.

2.4.1 EJERCICIO DEL TEMA:

Realizar el análisis a la siguiente base de datos así:



Fuente: Elaboración propia.

- 2) ¿Qué formas normales se están violando?
- 3) ¿Qué propondría usted para mejorar el modelo de datos?

2.5 TEMA 4 LOS SISTEMAS DE APLICACIÓN

La mayoría de los software que no pertenecen a los Sistemas Operativos, a los Sistemas de Gestión de Base de Datos, a los Lenguajes de Programación, pertenecen al conjunto de los aplicativos que se elaboran utilizando normalmente los lenguajes de programación y las bases de datos a través de procesos de Ingeniería del Software, buscando con ello satisfacer necesidades de clientes que pueden ser personas naturales, instituciones, empresas, organizaciones y todo aquello que requiera sistemas de software.

Los aplicativos de software, se encuentran presentes en cualquier sector y en los últimos diez años se ha convertido en herramienta de trabajo, ya que agiliza todo tipo de proceso.

En las diferentes organizaciones aparecen sistemas de software que apoyan los diferentes procesos empresariales como:

Los ESS (Sistemas de Apoyo a Ejecutivos) Serán los encargados de apoyar las decisiones a largo plazo que se encuentra a cargo de la dirección general de la empresa y que marcan el norte de esta.

En el caso de los SSG (Sistemas de Soporte Gerencial), son conocidos también dentro de las organizaciones empresariales como SIG (Sistema de Información Gerencial) o en inglés como MIS (Management Information System), refiriéndose al conjunto de métodos que permiten la gestión de la información para la toma de las decisiones más importantes de las organizaciones, con el aval de los respectivos directores.

Estos se encargan de proporcionar a la empresa mecanismos que facilitan la administración de los procesos e información, teniendo presente lo que se requiere, el momento en que se requiere, los usuarios que lo necesitan, la forma en que se requiere, los costos que se pueden generar, la prioridad en la información, la forma cómo se deben ejercer los controles, hasta dónde le compete procesar datos entre otros elementos para convertir los datos en información útil para las reglas del negocio.

Los Sistemas Informáticos SPD (Sistemas de Procesamiento de Datos), se basan en normas administrativas, donde se define qué hacer con los datos almacenados en las diferentes bases de datos y con qué intención deben ser procesados y convertidos en información que soporten cada uno de los procesos organizacionales.

Los SAD (Sistemas de Apoyo a las Decisiones), equivalen a un conjunto de programas y herramientas que permiten obtener oportunamente la información requerida durante el proceso de la toma de decisiones, en un ambiente de incertidumbre con el fin de proporcionar la mayor cantidad de información relevante en el menor tiempo posible, para así decidir lo que más conviene en determinado momento.

Lo que respecta a los SAO (Sistemas de Automatización de Oficina), tienen que ver con aplicaciones destinadas a ayudar con el trabajo diario del administrativo de una empresa u organización, ubicado generalmente sobre herramientas ofimáticas como procesadores de texto, hojas electrónicas, graficadores, diseñador de presentaciones, entre otras.

De igual forma a nivel de diversión aparecen los juegos, que son aplicaciones que poseen un alto contenido multimedia (video, texto, imagen, sonido), permitiendo la interactividad del usuario con la máquina.

En el sector educativo de igual forma aparecen herramientas como enciclopedias, juegos, lecturas interactivas, entre otros que buscan dinamizar los procesos de enseñanza y aprendizaje.

Es de anotar que cada día se presentan nuevas necesidades de software que permitan la automatización de múltiples procesos, todo ello pensado sobre plataformas tipo web, ya que en la última década su uso se ha expandido a todo los sectores.

2.5.1 EJERCICIO DEL TEMA:

Sacar un listado de 10 necesidades de software que se encuentren en su medio y que permitan la agilización de los procesos, garantizando el eficiente manejo de datos y la generación de la información para la toma de decisiones.

PISTAS DE APRENDIZAJE

Traer a la memoria:



EL SOFTWARE DE APLICACIÓN

Puede ser genérico, cuando se elabora un producto de software y este se puede utilizar para varios usuarios, instituciones, empresas u organizaciones que tengan la misma necesidad (ejemplo un software contable) o puede ser hecho a la medida porque las reglas del negocio son diferentes a otras instituciones, empresas u organizaciones.

2.6 TEMA 5 LOS SISTEMAS EXPERTOS

Son sistemas de software que imitan las actividades humanas y permiten solucionar problemas que exigen un gran conocimiento sobre un tema o varios temas. Dichos sistemas actúan bajo bases de conocimiento que han sido almacenadas previamente con datos expertos en áreas específicas. Por lo tanto los Sistemas Expertos son producto de múltiples investigaciones en el campo de la inteligencia artificial, buscándose con ello ser más eficientes y eficaces en la toma de decisiones que requieren información especializada.

Debido a esto en la actualidad se están mezclando diferentes técnicas o aplicaciones aprovechando las ventajas que cada una de estas ofrece para poder tener empresas más seguras. Un ejemplo de estas técnicas sería los agentes que tienen la capacidad de negociar y navegar a través de recursos en línea, (ejemplo las bolsas de valores); y es por eso que en la actualidad juega un papel preponderante en los sistemas expertos.

Los sistemas expertos manejan ventajas como:

- Mayor durabilidad que un humano
- Repetición de tareas, siendo inagotables
- Mayor rapidez al procesar información
- Bajo costo después de ser programado
- Elaboración de trabajos en entornos peligrosos para el ser humano (Ejemplo, desactivar bombas)
- Fiabilidad en los procesos, después de haber sido programado, utilizando controles de calidad.

- Manejo de varios conocimientos y especialidades, entre otros.

Desventajas como:

- Sentido común para tomar otras decisiones, donde estén involucrado los sentimientos humanos.
- Lenguaje natural, para mantener cualquier tipo de conversación
- Capacidad de aprendizaje ilimitado como el ser humano.
- Carecimiento de sentidos.
- Flexibilidad, para el cambio de decisiones cuando sea pertinente hacerlo

Los Sistemas Expertos son artefactos compuestos por hardware y software, que no necesariamente tienen forma humana como los robots. Fueron pensados con el fin de ayudar a tomar decisiones y realizar trabajos programados, con la mínima intervención del hombre.

2.6.1 EJERCICIO DEL TEMA:

- 1) Realizar la siguiente lectura (http://es.wikipedia.org/wiki/Sistema_experto)
- 2) ¿Qué tarea realizan los Sistemas Expertos?
- 3) ¿Qué es Ingeniería del Conocimiento?
- 4) ¿Qué es Gestión del Conocimiento?

PISTAS DE APRENDIZAJE

Traer a la memoria:



UN SISTEMA EXPERTO ES

Un programa que es capaz de manejar problemas que normalmente necesitarían de la intervención humana para su resolución, por lo tanto son capaces de aconsejar, categorizar, analizar, comunicar, consultar, diseñar, diagnosticar, explicar, formar conceptos, interpretar, justificar y planificar.

3 UNIDAD 2 LA INGENIERÍA DEL SOFTWARE (IS)

El software se ha convertido en el corazón e inteligencia de cualquier proceso u objeto que nos rodea, aparece en el control de los medios de transporte, de los diferentes sistemas de comunicación, en los aparatos electrodomésticos, en elementos de ocio, también como apoyo a las diferentes disciplinas, en cada uno de los diferentes tipos de organizaciones dentro de los procesos de producción, de servicios, empresas financieras, instituciones de educación, entre otras, convirtiéndose en herramienta imprescindible para el desarrollo de las diferentes actividades.

Buscando calidad en la construcción de los productos de software, aparece la Ingeniería del Software como disciplina que busca trabajar con proceso y a su vez actividades organizadas tendientes al análisis de requerimientos de usuario, buscando dar solución a problemas, necesidades u oportunidades que deban ser resueltas a través de productos de software, diseños acorde a necesidades, codificaciones, integración de componentes buscando reunir subsistemas y dándole sentido a los macroprocesos y a su vez estos integrados con otros externos, cualificación de producto e instalación de éste, así como la aplicación de diversos métodos que permiten a las empresas de desarrollo de software generar sus propias líneas de desarrollo y la utilización de herramientas adecuadas que tengan como fin primordial satisfacer las necesidad del usuario final.

Para ilustrar la temática, se recomienda observar el video sobre Qué es Ingeniería del Software



Que es la Ingeniería de software? [Enlace](#)

3.1.1 RELACIÓN DE CONCEPTOS

La temática y terminología sobre Ingeniería del Software se empezó a tocar desde 1968, debido a la necesidad de software, ya que el que se desarrollaba se hacía de forma informal y sólo era posible para proyectos grandes y unidos a las potentes máquinas (hardware de computadoras) que se construían en dicha época.

La Ingeniería del software es una disciplina joven, perteneciente al área de la información o ciencias de la computación que busca el suministro de procesos, métodos y herramientas que permitan el desarrollo e implementación de productos informáticos, bajo características de calidad. Algunos de los precursores de dicha disciplina y que pasaron a la historia por su contribución hacia lo que hoy se considera Ingeniería del Software son Alan Turing, Frederick P. Brooks, Jr., Steve McConnell, Barry Boehm, Ivar Jacobson, Grady Booch, Edsger Dijkstra, David Parnas, entre otros.

Algunas definiciones sobre la Ingeniería de Software, abordadas desde el punto de vista de algunos autores y retomada en la tesis de maestro "La Ingeniería de Software: Una Discusión Epistemológica, J. Jesús María Zavala Ruiz", quien plantea dicha definición de la siguiente forma:

"Según la definición del IEEE, citada por [Lewis 1994] "software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según el mismo autor, "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software (Software Engineering) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", que en palabras más llanas, se considera que "la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" [Cota 1994]." Tomado de (Zavala-Ruiz, 2011)

Teniendo en cuenta que el software se construye como cualquier producto y que hasta hace poco se venía trabajando de manera artesanal, es necesario hablar del proceso de Ingeniería del software para su elaboración, donde el mismo autor lo referencia así:

"El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de logra un objetivo, en este caso, la obtención de un producto de software de calidad" [Jacobson 1998]. El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" [Jacobson 1998]" (Zavala-Ruiz, 2011)

Es así como se plantea desde la Ingeniería del Software, abordar temáticas que buscan hacer de los productos informáticos, herramientas prácticas y ágiles para los usuarios finales que en definitiva serán quienes las utilicen en sus diferentes actividades, por lo tanto se abordará temáticas importantes como:

Modelos o Metodologías en la IS

Para el desarrollo de software se han propuesto metodologías, buscando crear productos de software, que satisfagan las necesidades del cliente es así como aparecen metodologías puestas a disposición de los proveedores de dichos productos. Entre algunas de ellas se conoce Cascada, Espiral, Incremental, Prototipos, RUP, XP, entre otras que son acogidas y acomodadas de acuerdo a las líneas bases que dichas empresas adoptan para la construcción de sus productos informáticos.

Los Requerimientos

Se constituye como la etapa de Ingeniería del Software más importante de dicho proceso, ya que se encarga de identificar, documentar, organizar y seguir los requisitos de un sistema y los cambios que se puedan presentar en ellos además de establecer acuerdos entre cliente y empresa de desarrollo.

Herramientas de Análisis

Para la Ingeniería del Software es importante incorporar dentro de su proceso herramientas que permitan especificar los requerimientos del usuario como por ejemplo prototipos, Lenguajes Modeladores (UML), modelos de datos entre otros que conviertan la información inicial abstracto en información más entendible y así convertir esas necesidades del cliente en un producto funcional.

Los Diseños actuales

El diseño es una fase de la Ingeniería del Software que se basa en la especificación de requisitos a raíz del análisis de los requerimientos y permite definir la forma cómo los requisitos se convierten en una estructura entendible para el cliente y/o el usuario.

El diseño debe orientarse hacia la forma de representar la información, la arquitectura para la interfaz, la orientación a objetos, el diseño de los procesos, la parte visual gráfico, entre otros y que a través del código que ofrecen los lenguajes de programación se pueda manipular los datos de las bases de datos volviendo realidad los requerimientos iniciales.

Los Riesgos

En su libro sobre análisis y gestión del riesgo, Robert Charette escribe sobre el riesgo en el sentido en que afecta a los futuros acontecimientos y son consecuencia del pasado y del presente, de tal forma que lo que se vive actualmente es una consecuencia de lo que se hizo o se dejó de hacer en el pasado.

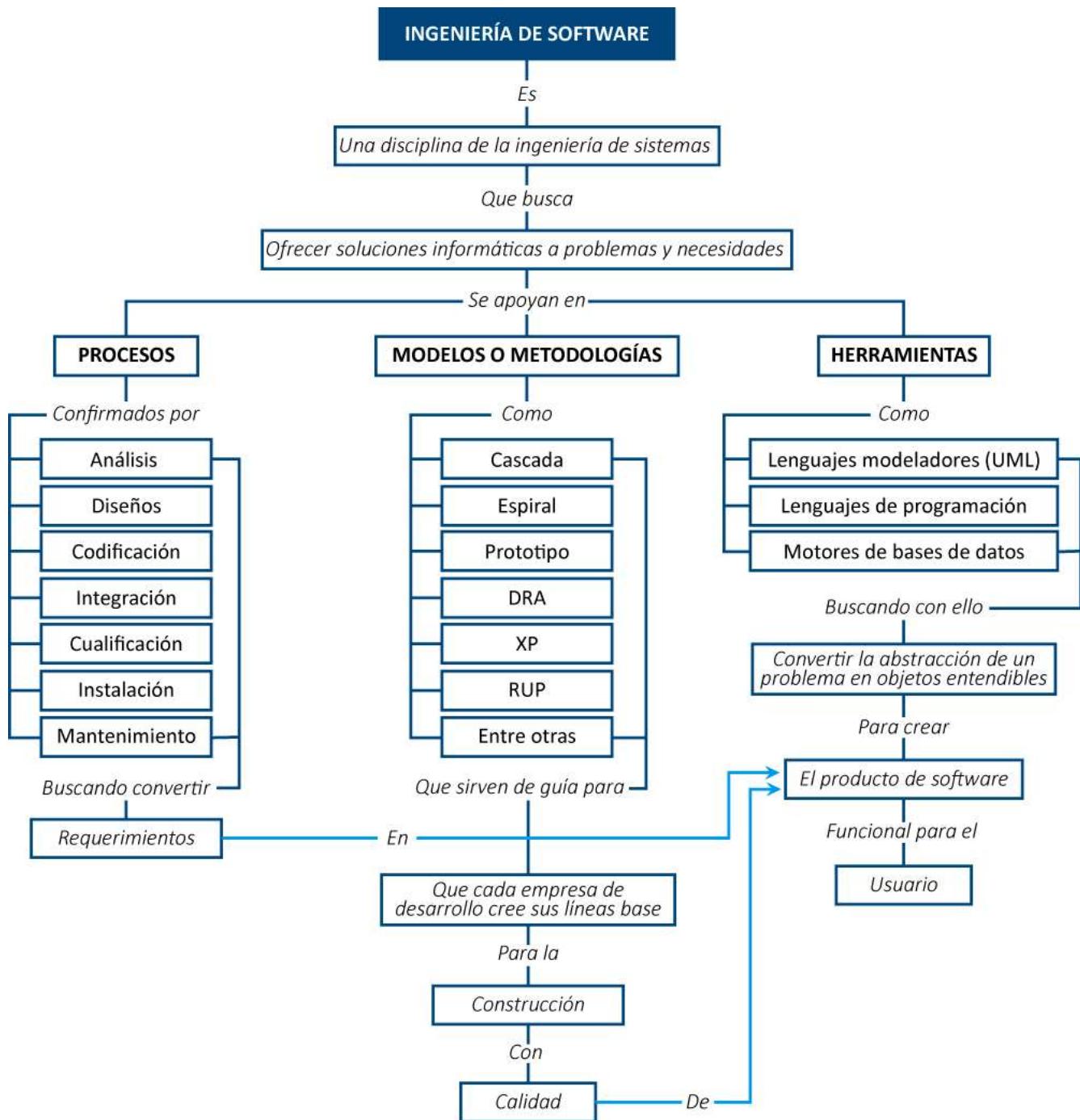
Respecto a los riesgos en la ingeniería del software Charette, se basa en tres pilares fundamentales agrupados en los siguientes interrogantes ¿qué riesgos podrían hacer que nuestro proyecto fracasara? ¿Cómo afectarán los cambios en los requisitos del cliente, en las tecnologías de desarrollo, en los ordenadores a las que van dirigidas, el proyecto y todas las entidades relacionadas con él, al cumplimiento de la planificación temporal y al éxito en general? ¿qué métodos y herramientas deberíamos emplear, cuánta gente debería estar implicada, qué importancia hay que darle a la calidad?. Son interrogantes que se deben tener presente al momento de realizar un plan de riesgos donde se pueda actuar de forma proactiva, evitando la aplicación de correctivos cuando el problema ya está dado.

Pruebas y Calidad

Las pruebas de software (Testing) Son procesos considerados dentro de la Ingeniería del software que buscan verificar los requerimientos funcionales del producto con el fin de identificar fallos antes de la implementación del producto asegurando su calidad.

Con el fin de garantizar la calidad de un producto informático, existen varios tipos de pruebas que las empresas de desarrollo ponen a su consideración dependiendo del tipo de software que se construya. Entre algunos de los tipos de pruebas que se utilizan tenemos pruebas unarias, funcionales, de integración, de validación, del

sistema, de caja blanca, de caja negra de aceptación, de carga, entre otras. Que buscan entregar un producto sin defectos.



3.1.2 OBJETIVO GENERAL

Identificar los principales elementos que giran en torno a los procesos de Ingeniería del Software como factores relevantes para la comprensión de requerimientos, el respectivo análisis, diseño, codificación, pruebas e implementación de un producto informático de calidad.

3.1.3 OBJETIVOS ESPECÍFICOS

- Identificar las principales metodologías o modelos que han sido utilizados y los que se siguen utilizando para la construcción de productos de software.
- Analizar la importancia que tiene la comprensión de los requerimientos en la construcción de productos de software, como aspecto fundamental para que el software sea funcional para el usuario final.
- Identificar herramientas que se utilizan actualmente en la industria del software con el fin de que se obtenga una mejor comprensión de los requerimientos abstractos que solicita el cliente y que deben ser modelados por el usuario informático (proveedor de software).
- Comprender la importancia que tienen los diseños, codificación y estructuración del producto informático, buscando la reutilización de sus componentes a través de metodología objetual.
- Identificar los principales riesgos que ponen en peligro el desarrollo del producto de software, así como su clasificación y tratamiento para minimizar las amenazas.

3.2 TEMA 1 MODELOS O METODOLOGÍAS EN LA IS

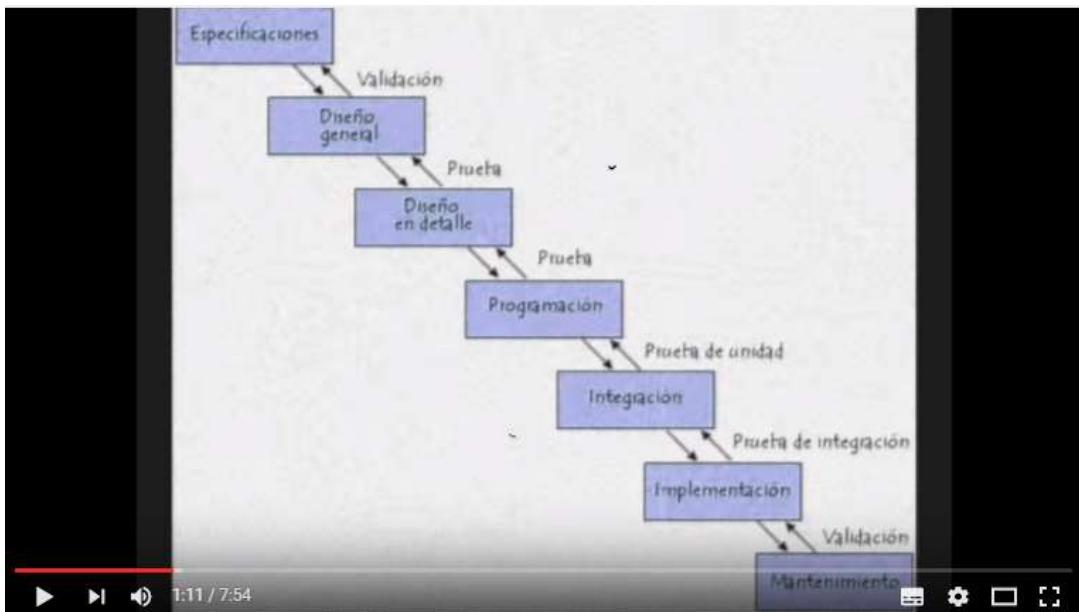
Los modelos y metodología de software, buscan dar elementos o pautas para que las empresas que se dedican a la construcción de productos informáticos, puedan estandarizar sus procesos, buscando con ello generar valor agregado a dichos productos, logrando así que estos sean competitivos y cumplan con las necesidades y requerimientos del cliente.

Los modelos o metodologías, han sufrido evolución, ajustes y mejoras con el pasar del tiempo, donde se ha tratado de mejorarlos, con el fin de ofrecerlos como herramientas prácticas. A continuación se hace alusión a los principales modelos que se han aplicado durante el ciclo de construcción e implementación de dichos productos.

■ **Modelo en cascada**

Fue un modelo propuesto en los años 70 (Winston Royce), cuyos principios se basan en que el proyecto de software se divide en fases (análisis, diseño, desarrollo, pruebas, implementación), que deben ser secuenciales y hasta que no se termine la fase es imposible dar inicio a la siguiente. El desarrollo de dicha metodología, debe basarse en la planificación de todo el proyecto desde principio a fin, estableciendo tiempos, presupuestos y personal, ya que ante un error es difícil devolverse a fases anteriores. Se conoce como cascada haciendo alusión

a una caída de agua. Ver video sobre Modelo Cascada, donde se muestra de forma gráfica, cada una de sus fases, ventajas y desventajas para la construcción de productos de software.

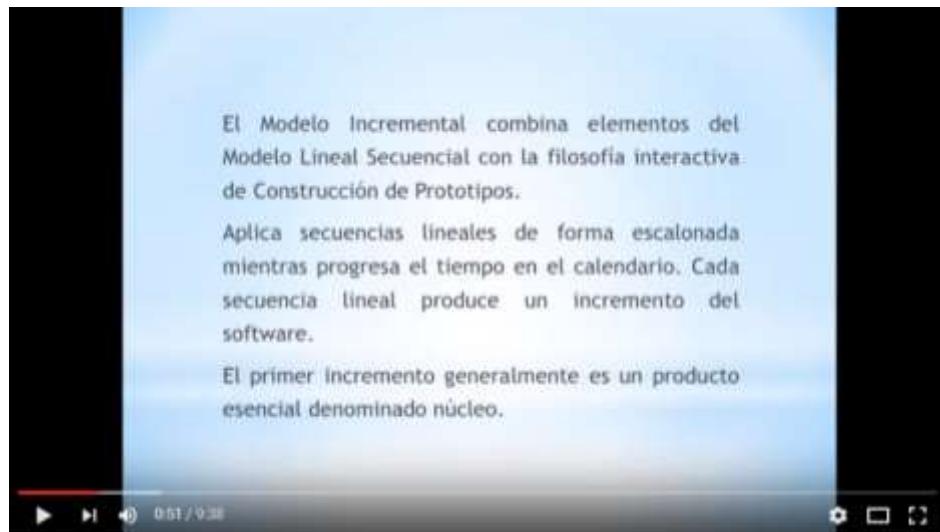


Modelo de Desarrollo en Cascada [Enlace](#)

Es considerado como el primer modelo para la construcción de software, a partir del cual se basan los demás modelos.

Modelo incremental

Es un modelo basado en la metodología anterior que consiste en trabajar por fases utilizando mini_cascada (análisis, diseño, desarrollo, pruebas), revisando el proceso para determinar si se va bien hasta ese momento, donde en la siguiente evolución e incremento de requerimientos, se inicia nuevamente con la cascada. Ver video sobre Modelo incremental, donde se muestra de forma gráfica, cada una de sus fases, ventajas y desventajas para la construcción de productos de software



Modelo incremental - Explicación & Desarrollo [Enlace](#)

■ **Modelo Prototipo**

Consisten en una metodología, permite realizar versiones Demos, con el fin de mostrarle al cliente posiblemente el tipo de producto que se le va a construir. Normalmente un prototipo no es funcional, en el sentido que no cubre requerimientos del software por no ser definitivo. Actualmente es muy utilizado para levantar requerimientos, ya que le da una idea al cliente de lo que posiblemente puede esperar del producto definitivo. Ver video sobre Modelo de Prototipo, donde se muestra de forma gráfica, cada una de sus fases, ventajas y desventajas para la construcción de productos de software.



Modelo de Prototipo [Enlace](#)

■ **Modelo Espiral**

El modelo espiral, significó una gran evolución para dichas metodologías, ya que se incorporan elementos como la evaluación de cada proceso, donde se tienen en cuenta los riesgos que pueden afectar el proceso de ingeniería de software. Dicha metodología divide el proyecto en segmentos y trabaja en forma de espiral, permitiendo la evaluación de dichos riesgos en cada iteración, donde pasa por diferentes fases que corresponden a un mismo ciclo como son: en primer lugar la determinación de los objetivos, alternativas, y desencadenantes de la iteración; en segundo lugar la Evaluación de alternativas que permite identificar y resolver los riesgos; en tercer lugar el desarrollo y verificación de los resultados de la iteración, y en cuarto lugar el plan de la próxima iteración o vuelta del espiral. El conjunto de fases se repite ciclo a ciclo hasta que termine todo el proceso de construcción. Ver video sobre Modelo espiral, donde se muestra de forma gráfica, cada una de sus fases, ventajas y desventajas para la construcción de productos de software.



Modelo Espiral [Enlace](#)

■ **Modelo RUP (Proceso Unificado de Rational)**

Es un modelo recoge elementos de los anteriores, creado finalizando los años 90, muy utilizado actualmente en la creación de productos de software, que contiene un proceso de estructuración orientado especialmente hacia la construcción de productos con orientación a objetos, dando una dinámica diferente a la forma de desarrollar software. El RUP se apoya en el Lenguaje Unificado de Modelado (UML), constituido como la metodología estándar más utilizada en el análisis, implementación, documentación de los sistemas orientados a objetos. El RUP, permite que cada organización adapte su metodología acorde a las necesidades de construcción. Las fases que maneja dicho modelo son:

- **Inicio:** (Documento visión, especificación de requisitos)
- **Elaboración:** (Diagramas de caso de uso)
- **Construcción:** (Documento arquitectura que trabaja con las siguientes vistas:

- **Vista Lógica** (Diagrama de clases, Modelo Relacional para el manejo de datos)
- **Vista de Implementación** (Diagrama de Secuencia, Diagrama de estados, Diagrama de Colaboración)
- **Vista Conceptual** (Modelo de dominio)
- **Vista física** (Mapa de comportamiento a nivel de hardware.)
- **Transición:** (Disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico)

Ver video sobre Modelo RUP, donde se muestra de forma gráfica, cada una de sus fases, ventajas y desventajas para la construcción de productos de software.



Modelo de Referencia RUP [Enlace](#)

■ Modelo XP (Programación Extrema)

Se orienta hacia procesos ágiles para la construcción de productos informáticos, con alta adaptabilidad a la previsibilidad, ya que considera que los cambios durante el proceso son incontrolables en el sentido de que siempre van a aparecer, por lo tanto el modelo permite incorporarlos en cualquier etapa del proceso de ingeniería del software. Este modelo considera la programación por pares lo que significa que para una misma actividad, se debe disponer de dos personas, siendo además indispensable el acompañamiento del cliente durante su desarrollo, considerando que es él quien tiene el conocimiento de cómo se realiza el proceso. Ver video sobre Modelo XP, donde se muestra de forma gráfica, cada una de sus fases, ventajas y desventajas para la construcción de productos de software.



Programacion XP [Enlace](#)

3.2.1 EJERCICIO DEL TEMA:

- 1) Observar cada uno de los videos para desarrollo de software
- 2) Elaborar un esquema, donde se muestre el nombre del modelo, sus principales características, sus ventajas y sus desventajas.
- 3) Argumentar en 20 líneas cuál es el mejor modelo y por qué.

PISTAS DE APRENDIZAJE

Traer a la memoria:



Para la aplicación de modelos o metodologías para desarrollo de software

Cada empresa que se dedica al desarrollo de software, va elaborando sus líneas base, que consiste en organizar modelos o metodología propias, tomando de cada modelo planteando aquellos apartes que permitan apoyar sus procesos, conformando así metodologías que les permitan la construcción de productos de calidad.

3.3 TEMA 2: LOS REQUERIMIENTOS

Todo proceso de ingeniería de software inicia a partir de una necesidad, problema u oportunidad, donde para entender esa parte tan abstracta, es necesario aplicar elementos que ofrecen los modelos o metodologías para

construir software, sin importar el modelo que se aplique, dichas necesidades se deben entender para poder saber qué se desea construir. Para abordar dichas temáticas, se hará alusión a las definiciones que sobre requerimientos e ingeniería de requisitos abordan algunos autores.

La IEEE (Institute of Electrical and Electronics Engineers) define un Requisito de tres formas, así: como una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo, como una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato y como una norma, especificación o documento formal que permita expresar esas necesidades. Respecto a la Ingeniería de requisitos lo define como todas aquellas actividades que se relacionen con la identificación de necesidades de los clientes y/o usuarios, la documentación que se haga de ella, a partir de su identificación, su análisis, verificación, validación con el cliente para realizar ajustes de ser necesario.

Según Roger Pressman “Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente

quiere y cómo interactuarán los usuarios finales con el software”. (Pressman, 2006: 155).

Con el fin de ampliar dicha temática, se sugiere realizar lectura al siguiente artículo (Ingeniería de requisitos base de los procesos de desarrollo. Link

<https://scholar.google.com.co/scholar?q=define+requerimientos+%2Bpressman&hl=es&btnG=Buscar&lr=>

Para ilustrar mejor la temática, se hará alusión a una situación del contexto, en la que se puede identificar fácilmente la importancia de entender los requerimientos.

Situación:

Un centro médico actualmente lleva el control de citas médicas de forma manual por lo que el servicio que se presta se hace lento, algunas veces las citas se confunden quedando mal con los pacientes y realizando reprocesos y pérdida de tiempo.

Por lo tanto se requiere un sistema informático con el fin de controlar dichas citas y saber en determinado momento si los consultorios tienen asignada citas o si están disponibles.

Es de anotar que dentro de los consultorios se atienden tres tipos de citas como son odontología, pediatría, nutrición entre otras. Los pacientes que se atienden son de dos tipos los que van por EPS o por prepagada, estos últimos tienen un descuento especial por utilizar el servicio. Es de aclarar que en las citas es necesario registrar los datos del consultorio, fecha, hora inicial y datos del médico, cada cita tiene una duración de 20 minutos.

La secretaría del centro médico debe realizar las siguientes operaciones:

- Consultar los consultorios que están disponibles en determinado momento.
- Consultar el costo de una cita, de acuerdo al tipo de cita.
- Visualizar el descuento especial para los pacientes que poseen medicina prepagada.
- Consultar por un paciente en especial suministrando cédula, tipo de consulta y número de consultas asignadas.

- Reservar una cita especificando el consultorio, fecha, hora médica.
- Cancelar una cita, especificando el consultorio y la hora.

Por lo tanto se recomienda seguir los siguientes pasos, elaborando con ello un documento que sirva a su vez de sustentación escrita del proceso y que posteriormente se debe validar con el cliente.

■ **Identificar el problema o necesidad**

En el caso del ejemplo anterior el problema consiste en que actualmente se está llevando el control de citas manual, lo que genera reprocesos y por consiguiente pérdida de tiempo y usuarios inconformes.

■ **Realizar un listado de las necesidades**

- Manejo de información de médicos
- Manejo de información de pacientes
- Manejo de información de citas
- Manejo de información de consultorios
- Manejo de información de tipo de consultas
- Manejo de información de EPS
- Manejo de información del tipo de EPS o de Salud

Que a través de la anterior información se pueda:

- Consultar los consultorios que están disponibles en determinado momento.
- Consultar el costo de una cita, de acuerdo al tipo de cita.
- Visualizar el descuento especial para los pacientes que poseen medicina prepagada.
- Consultar por un paciente en especial suministrando cédula, tipo de consulta y número de consultas asignadas.
- Reservar una cita especificando el consultorio, fecha, hora médica.
- Cancelar una cita, especificando el consultorio y la hora.

Es importante no dejar requerimientos por fuera, ya que en etapas avanzadas de proceso de Ingeniería del Software, sería un aspecto a contra para que el sistema tenga calidad. De igual forma es importantísimo contar con el aval del cliente que es quien en realidad conoce el proceso.

■ **Formular objetivos**

Por ejemplo el objetivo general, se podría redactar de la siguiente forma:

Desarrollar un producto informático que permita la gestión sobre el control de citas médicas para el centro médico XYZ, bajo plataforma Web.

Definir el alcance

En la ingeniería de requisitos es importante definir lo que hará el sistema, con el fin de que quede claro lo que no hará, ya que ello evita inconvenientes y malos entendidos con el cliente, por lo tanto debe estar muy claro y registrarse en el documento de requerimientos.

Para mayor claridad, se recomienda definir el alcance en términos de sistemas de información, teniendo en cuenta las entradas, el proceso y la salida, donde las entradas tienen que ver con todo aquello que alimentará al sistema, el proceso con lo que requiere conversión por el sistema, tomando como insumo las entradas y la salida equivale a los informes, consultas y todo aquello que arroje el sistema y obedezca a lo que se pidió en los requerimientos.

Siguiendo con nuestro ejemplo, el alcance se puede organizar de la siguiente forma:

Entradas: Todo lo que sirve para alimentar al sistema

-  Datos de médicos
-  Datos de pacientes
-  Datos de consultorios
-  Datos de tipo de consultas
-  Dato de EPS
-  Datos del tipo de EPS o de Salud

Proceso: Equivale a la parte central del sistema que en este caso se refiere a la gestión de la cita, desde que el cliente la solicita, hasta que se cambia su estado, ya sea cancelada o cumplida.

Salidas: Todo lo que salga del sistema como

-  Consultar los consultorios que están disponibles en determinado momento.
-  Consultar el costo de una cita, de acuerdo al tipo de cita.
-  Visualizar el descuento especial para los pacientes que poseen medicina prepagada.
-  Consultar por un paciente en especial suministrando cédula, tipo de consulta y número de consultas asignadas.
-  Reservar una cita especificando el consultorio, fecha, hora médica.
-  Cancelar una cita, especificando el consultorio y la hora.

Es importante tener presente que para obtener información se debe recurrir a herramientas como entrevistas, encuestas, revisión de documentos, filmaciones para procesos muy difíciles de entender, fotografía, elaboración de gráficos, entre otros que permitan tener claridad sobre lo que desea el cliente.

También se debe tener presente las herramientas de hardware que se requieren para que el sistema funcione óptimamente como el tipo de procesador, capacidad en RAM, en discos duros, red sobre la cual se trabajará, tipo de conexión a internet, licencias para software como Sistemas Operativos, Sistemas de Gestión de Bases De Datos, entre otros, así como personal con el que se cuenta, ya sea informático o no informático, si se requiere capacitar personal, entre otros.

3.3.1 EJERCICIO DEL TEMA:

Usted es contratado para desarrollar un software basado en la situación problemática expuesta, tiene cita mañana a primera hora con el director del centro médico.

Se pide diseñar un formato de entrevista, donde se muestre, hora, fecha, datos del entrevistador, datos del entrevistado, preguntas relevantes que permita tener claridad sobre lo que se debe saber antes de abordar el proceso de requerimientos.

PISTAS DE APRENDIZAJE

Traer a la memoria:



LA INGENIERÍA DE REQUISITOS

Comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de los clientes y/o usuarios. Esta etapa del desarrollo del software debe quedar muy clara, ya que la mayoría de los errores y defectos en el producto de software se derivan de ella.

3.4 TEMA 3 HERRAMIENTAS DE ANÁLISIS

Actualmente las herramientas de análisis, se centran en metodologías que ayudan a interpretar cada uno de los procesos de la Ingeniería del Software. Actualmente se utiliza en la industria del software diferentes herramientas que permiten la elaboración de esquemas, diagramas, estructuración de modelos, herramientas CASE (Software Asistido por Computador) que agilizan todos esos procesos, pero antes que las herramientas se debe tener en cuenta que esta por sí sola no realiza el trabajo, requiriéndose así que se tenga claridad sobre lo que se desea hacer.

Una de las herramientas más utilizadas consiste en el UML (Unified Modeling Language), Lenguaje Unificado de Modelado, modelo presentado por Grady Booch, Jim Rumbaugh y Ivar Jacobson, socios del compañía Rational Software en 1995. Actualmente dicho modelo apoya todo tipo de desarrollo de software orientado en objetos en todo lo relacionado con el proceso de ingeniería del software.

El UML cuenta con nueve diagramas divididos en dos grandes grupos, los de estructura y los de comportamientos agrupados así:

Modelado de la estructura

Clases: Muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, cubriendo la vista de diseño estática del sistema.

Objetos: Análogo al diagrama de clases, muestra un conjunto de objetos y sus relaciones, pero a modo de vista instantánea de instancias de una clase en el tiempo.

Componentes: Muestra la organización y dependencias de un conjunto de componentes. Cubren la vista de implementación estática de un sistema. Un componente es un módulo de código, de modo que los diagramas de componentes son los análogos físicos a los diagramas de clases.

Despliegue: Muestra la configuración del hardware del sistema, los nodos de proceso y los componentes empleados por éstos. Cubren la vista de despliegue estática de una arquitectura.

Modelado de comportamiento

Casos de Uso: Muestra un conjunto de casos de uso, los actores implicados y sus relaciones. Son diagramas fundamentales en el modelado y organización del sistema.

Secuencia y colaboración: Son diagramas de interacción, muestran un conjunto de objetos y sus relaciones, así como los mensajes que se intercambian entre ellos. Cubren la vista dinámica del sistema. El diagrama de secuencia resalta la ordenación temporal de los mensajes, mientras que el de colaboración resalta la organización estructural de los objetos, ambos siendo equivalentes o isomorfos.

Estados: Muestra una máquina de estados, con sus estados, transiciones, eventos y actividades. Cubren la vista dinámica de un sistema. Modelan comportamientos reactivos en base a eventos.

Actividades: Tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

Para ampliar la temática, ver archivo en link

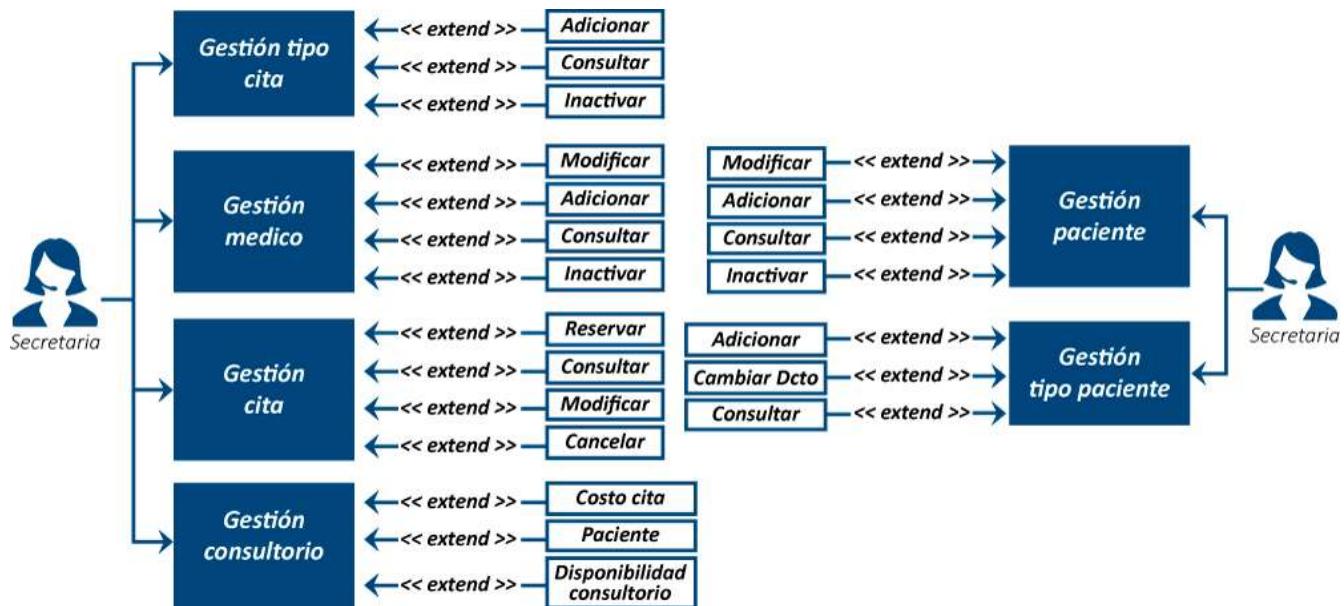
[http://www.slideshare.net/e1da4/diagramas-uml.](http://www.slideshare.net/e1da4/diagramas-uml)

Siguiendo con el ejemplo planteado sobre las citas médicas y para aclarar un poco más el tema se muestran algunos diagramas UML como el de casos de uso, secuencia, clases.

CASOS DE USO CITA MÉDICA



uc GESTIÓN MAESTROS CASOS DE USO 3



En estos diagramas se permite observar la interacción de los actores con el sistema, donde en este caso la secretaría del centro médico, se encargará de realizar todos los procesos, también se puede visualizar que para cada gestión se deben ejecutar acciones por ejemplo en lo referente a médico se podrá adicionar médicos nuevos cuando sean contratados, consultar los ya existentes, modificar sus datos e inactivarlos cuando ya no trabajen para el centro médicos. De igual forma acorde a los requerimientos se podrá manipular datos de entrada, de proceso y de salida. Este tipo de diagrama de casos de uso permite entender mejor lo que realmente se necesita que haga el sistema.

A continuación se muestra una plantilla que permite documentar los casos de uso, donde para cada uso se deberá elaborar una. En el caso del tipo de cita en el uso de consultar, se presenta la siguiente documentación

Plantilla para documentar caso de uso Consultar de Tipo de Cita

Caso de uso: Consultar tipo de cita
Descripción u objetivo: La secretaria se encarga de consultar los tipos de citas que ofrece el centro médico.
Actores: Secretaria
Precondiciones: <ol style="list-style-type: none">1. Debe tener acceso al sistema.2. Debe existir tipos de citas registradas.
Pasos o Flujo normal: <ol style="list-style-type: none">1. La secretaria, digita el código del tipo de cita a consultar2. El sistema verifica la existencia de dicho código3. El sistema muestra toda la información consultada4. El sistema muestra un mensaje.5. Se cierra la ventana
Variaciones o flujo alternativo: Si el paso 2 es falso, va al paso 4
Extensiones:
Pos condiciones: Se muestra el tipo de cita consultado.

A partir de la anterior plantilla se elabora otro tipo de diagrama que tiene que ver con el de secuencia, donde se describe los pasos o flujo normales de la plantilla anterior y tiene que ver con el movimiento de mensajes que son controlados a través del lenguaje de programación.

Diagrama de Secuencia Consultar de Tipo de Cita

En este diagrama interviene el actor que es quien manipula el sistema, la vista que equivale a la interfaz gráfica que utiliza el usuario para manipular el sistema, el control equivale a la codificación que hace el desarrollador a través del lenguaje de programación, y el modelo equivale a la base de datos del software, creada a partir del modelo de datos relacional. Este diagrama envía y devuelve mensajes de lo que está pasando en el proceso y es práctico en el proceso de ingeniería ya que permite tener mayor claridad sobre cómo funcionarán los procesos.

CONSULTAR TIPO CITA

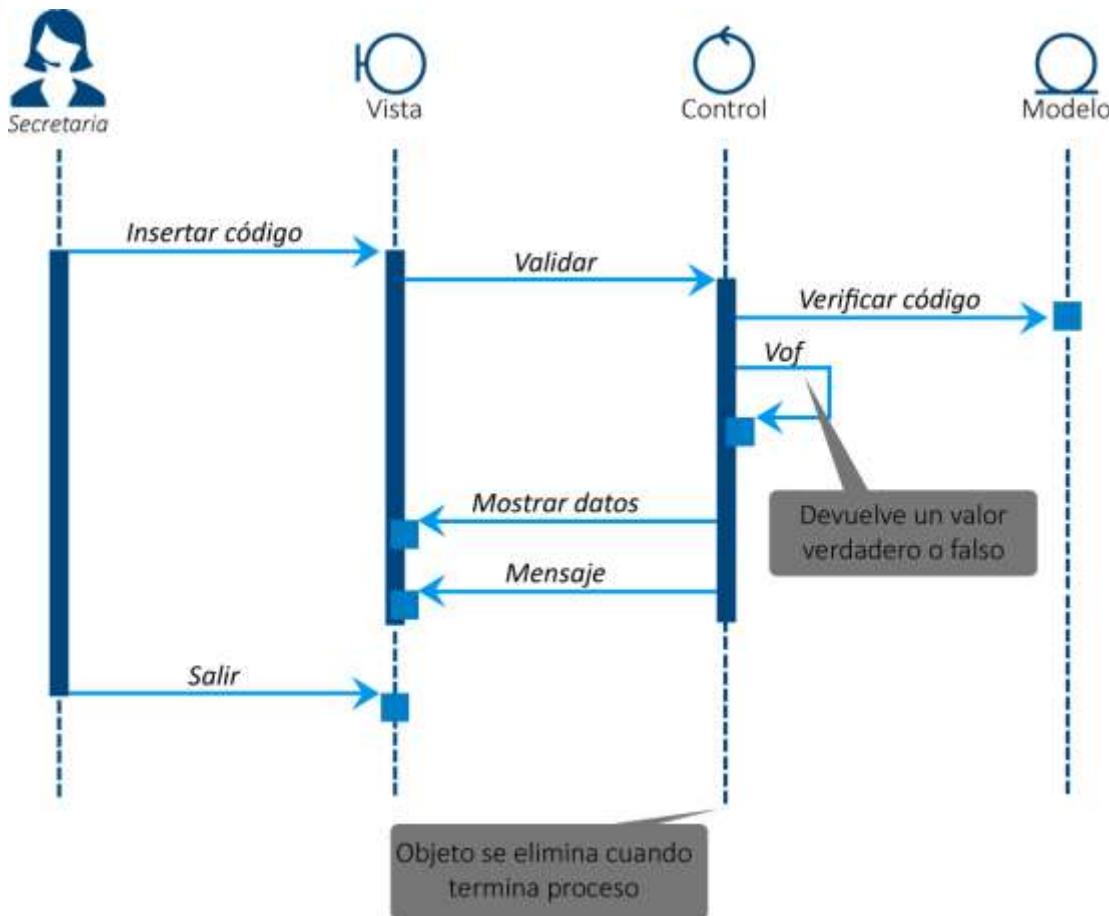
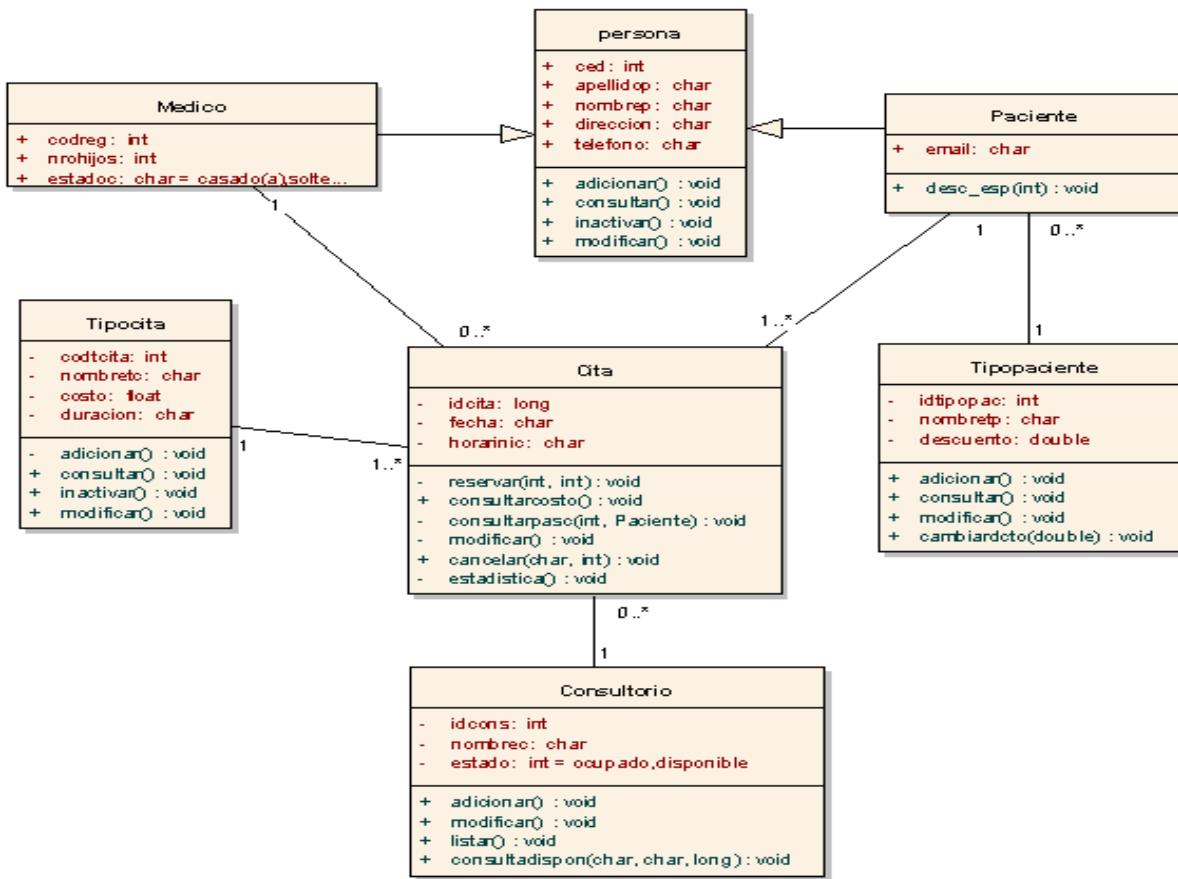


Diagrama de Clases

Este diagrama presenta la estructura que el programador debe tener presente en el momento de codificar con metodología de Programación Orientada a Objetos (POO) utilizando el lenguaje de programación, ya que sirve de guía para que éste estructure la forma como se interrelacionarán los diferentes objetos que intervienen en el software, con el fin de reutilizar código y componentes. De igual forma en este diagrama se puede observar métodos, por ejemplo en la clase tipo de cita aparece, siguiendo el ejemplo el método consultar() que debe ser codificado, teniendo presente el respectivo diagrama de secuencia y el de casos de uso en este caso.

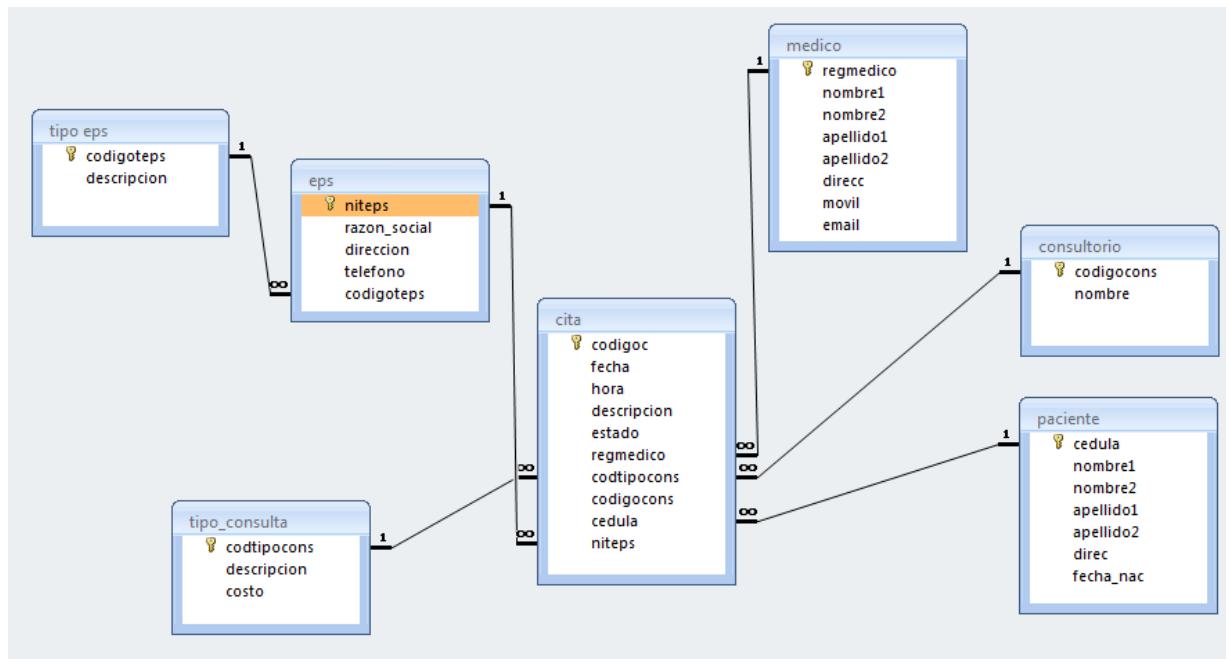


Es importante tener presente que los requerimientos que aparecen al principio del proceso, se deben evolucionar a través de las diferentes herramientas, clarificándolos, pero sin perderlos de vista, y que dichos requerimientos deben ser funcionales.

Base de Datos:

Es importante recordar que el modelo de datos juega un papel muy importante tanto en la etapa de análisis como en la de diseño, siendo una herramienta sumamente importante, ya que permite además convertir los requerimientos en comprensiones más claras, tanto para los que deben entender los requerimientos como para el cliente.

Modelo Relacional base de datos Control citas médicas



3.4.1 EJERCICIO DEL TEMA:

Elaborar un mapa conceptual que muestre cada uno de los diagramas UML, su utilidad en el proceso de Ingeniería del Software.

PISTAS DE APRENDIZAJE

Traer a la memoria:



EL ANÁLISIS DE SOFTWARE

Es una etapa del proceso de ingeniería del software, donde se refinan los requerimientos, utilizando para ello herramientas que faciliten la comprensión de dichas necesidades.

3.5 TEMA 4 LOS DISEÑOS ACTUALES

El diseño de software tiene relación con la forma de convertir esos requerimientos de software tan abstractos, en una representación de software que sea entendible por el usuario y que le sirva de plataforma para poner en ejecución sus requerimientos.

El diseño de software contiene cuatro actividades que son:

Diseño de datos:

Consiste en el diseño de la base de datos, apoyada en el modelo de datos que para nuestro caso consiste en el modelo relacional, como se mostró anteriormente en este mismo módulo. El diseño contiene parte lógica, cuando el modelo se realiza para comprender los requerimientos y parte física, cuando dicho modelo se convierte en la base de datos utilizando el respectivo motor de base de datos.

Diseño arquitectónico:

El objetivo principal del diseño arquitectónico es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos, relaciona la estructura de programas y la estructura de datos y define las interfaces que facilitan el flujo de los datos a lo largo del programa. Una de las arquitecturas más utilizadas actualmente, tiene que ver con la arquitectura de tres capas implementada a partir de la programación orientada a objetos.

Con el diseño arquitectónico se busca relacionar un cierto número de elementos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales que tienen que ver con la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

Diseño Procedimental:

EL diseño procedural se realiza después de que se ha establecido la

Estructura o arquitectura del programa y de los datos, se lleva a cabo con la codificación que permite integrar las demás capas para el desarrollo, teniendo en cuenta las reglas del negocio y la interrelación entre las interfaces y los datos.

Diseño de interfaces:

En la parte de interfaz, se busca que ésta por si sola le permita al usuario realizar una comprensión casi natural de su manejo, por lo tanto debe cumplir con características de amigabilidad, facilidad de uso, agradable a la vista, debe dar la sensación de limpieza, entre otras características que permitan confortabilidad a los usuarios. Ver presentación sobre diseños de interfaces (<http://www.slideshare.net/marti1125/diseo-de-interfaces-2073360>).

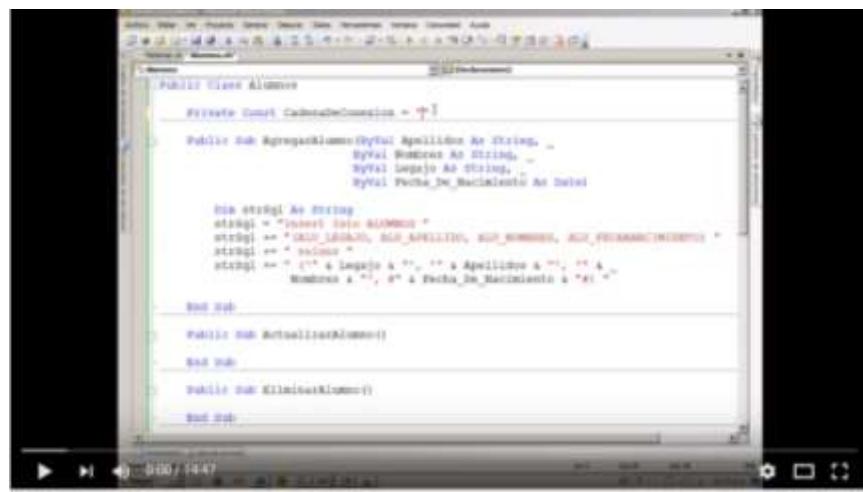
Los siguientes videos ilustran el diseño en tres capas relacionadas con el desarrollo de software orientado a objetos, haciendo alusión a la capa que tiene que ver con los datos, la que tiene que ver con las reglas del negocio y los respectivos controladores que permiten comunicar los datos con la capa de la vista o interfaz gráfica para el usuarios. Ver los siguientes videos:



Arquitectura de Software. Programación en 3 capas Parte 1 [Enlace](#)



Arquitectura de Software. Programación en 3 capas Parte 2 [Enlace](#)

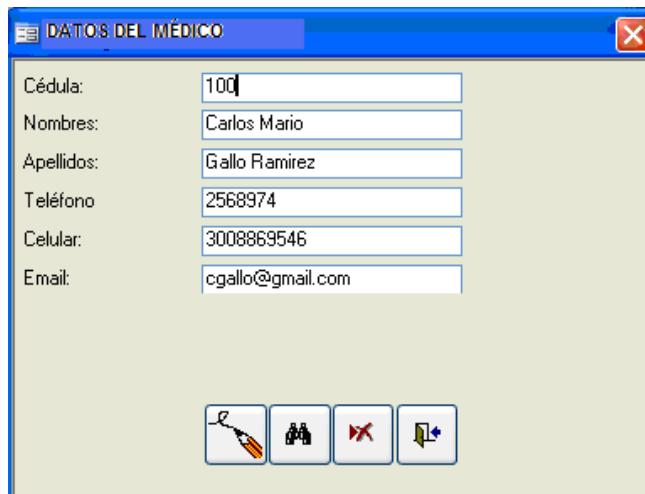


```
private static Connection Conexion = null;
public static AgregarAlumno(String Apellido, String Nombre, String Legajo, Date Fecha_Nacimiento) {
    String strsql = "Insert into ALUMNO"
        + "(ALU_APELLIDO, ALU_NOMBRE, ALU_LEGJO, ALU_FECHANACIMIENTO"
        + " values"
        + " (" + Legajo + ", " + Apellido + ", " + Nombre + ", " + Fecha_Nacimiento + ")";
}
public static ActualizarAlumno() {
}
public static EliminarAlumno() {
}
```

Arquitectura de Software. Programación en 3 capas Parte 3 [Enlace](#)

Según el ejercicio presentado sobre las citas médicas, se muestra una interfaz gráfica que en este caso permite manipular los datos del médico a través de la interfaz que se presenta, seguida de reglas que permiten validar el tipo de datos que se ingrese, además de unos controladores que hacen relación con los datos donde finalmente se tiene almacenada la información, en este caso del médico para su posterior utilización.

Diseño vista o interfaz médica



Fuente: Elaboración propia.

3.5.1 EJERCICIO DEL TEMA:

- Realizar la lectura del siguiente documento <http://indalog.ual.es/mtorres/LP/FundamentosDiseno.pdf>
- Sacar tres conclusiones sobre las características principales que deben cumplir los diseños de software.

PISTAS DE APRENDIZAJE

Traer a la memoria:



EL DISEÑO DE SOFTWARE

Es una etapa del proceso de ingeniería del software, que permite convertir los requerimientos abstractos y difíciles de comprender, en herramientas a las cuales se accede utilizando diseños como interfaces agradables que permiten integrar datos para que el sistema los procese y devolver información en forma de informes y consultas.

3.6 TEMA 5 LOS RIESGOS

Según (Menéndez, 2005) el riesgo de software, está relacionado con dos características que son la incertidumbre y las pérdidas que conducen a consecuencias no deseadas. De acuerdo a lo anterior el software

es un producto que se construye bajo altos niveles de incertidumbre, ya que se parte de necesidades intangibles que incluso pueden varias a medida que se desarrolla el producto o después de implementado, por lo que pueden aparecer otro tipo de requerimientos, teniendo en cuenta que las organizaciones son sistemas abiertos en constante evolución. En el caso de pérdidas puede ocurrir por ejemplo por el olvido de requerimientos, por el cambio de otros o solicitud de nuevos, por problemas con el equipo de desarrollo, por reestructuración de contratos con el cliente, entre otros donde se ve involucrado recursos humanos y físicos que normalmente repercute en el incremento de tiempos y recursos económicos.

Por lo tanto es necesario al inicio de cualquier proyecto elaborar una proyección donde se prevea el riesgos que puede presentarse al inicio del proyecto, durante su desarrollo e incluso después de ser implementado.

El riesgo se debe clasificar de acuerdo a su nivel de afectación al proyecto que puede ser (alto, medio o bajo), de acuerdo al tipo de riesgo que puede ser (hacia el proyecto, hacia factores técnicos o riesgos relacionados con el negocio donde se aplicará el producto informático), es necesario planificar los correctivos que se deben aplicar cuando este se presente, con el fin de evitar pérdidas que pueden ser nefastas para dicho proyecto.

Los riesgos del proyecto: Amenazan el desarrollo del proyecto, influyen para que el proyecto se retrase, pueden generar aumento de costos por aumento de recursos.

Los riesgos Técnicos: Tienen repercusión en la calidad del producto de software, puede generar retrasos en los procesos, ya que pueden presentarse problemas en las etapas del proceso de ingeniería del software como dificultades en requerimientos, en diseños, en interfaces, en pruebas, en implementación del producto, en su mantenimiento, en técnicas y herramientas anticuadas, tecnologías de punta desconocidas para los especialistas del proyecto, entre otros.

Los riesgo hacia el negocio: Se presentan normalmente cuando el producto informático se desarrolla y no es aceptado por los usuarios, ya sea porque no lo necesitan, porque se resisten al cambio o porque no se les tuvo en cuenta para la construcción de este, otro tiene que ver con que el producto de software no obedece a necesidades de la empresa, por cambio de personal, por que no se tiene presupuesto para costearlo, entre otros.

Los riesgos deben identificarse con el fin de buscar alternativas de prevención o de abordaje en caso de que se presenten con la posibilidad de visualizar el tratamiento que se le debe dar.

En la siguiente tabla se hace alusión a un ejemplo sobre la tabulación de posibles riesgos que se pueden presentar durante el proceso de ingeniería de software que tiene relación desde el inicio del proyecto, hasta que el software sale del mercado.

Tipo de Riesgo	Riesgo	Nivel (Alto, medio, bajo)	Possibles soluciones
Proyecto	Renuncia de desarrollador	Medio	Banco de hojas de vida.
Técnico	Virus Informático	Alto	Instalación de antivirus durante el proceso de desarrollo y en la

			implementación del software.
	Incendio	Alto	Copias de seguridad de las bases de datos en lugares diferentes al de donde funciona el sistema central.
Negocio	Cambio de requerimientos por parte del cliente	Alto	Reorganización contrato, para hacer ajuste a alcance, número de horas, costos.

3.6.1 EJERCICIO DEL TEMA:

Realizar lectura sobre proyección del riesgo

(http://www.wikilearning.com/curso_gratis/gestion_de_riesgos_en_ingenieria_del_software-proyeccion_del_riesgo/3620-12).

PISTAS DE APRENDIZAJE

Traer a la memoria:



UN RIESGO INFORMÁTICO

Es aquel evento que se debe prever, en caso de que llegue a suceder, ya que si el riesgo es alto puede atentar contra el desarrollo del proyecto de software, generando pérdidas económicas.

3.7 TEMA 6 PRUEBAS Y CALIDAD

Pruebas de Software:

Las pruebas del software son actividades que se encuentran dentro del proceso de Ingeniería del Software y tienen como fin encontrar errores, antes de que se avance a otros procesos o se entregue el producto al usuario final, ya que en este caso no sería un error, sino un defecto, poniendo en duda la calidad de dicho producto. Una prueba de software es exitosa, cuando se encuentran errores.

Las pruebas de software deben cumplir con características como: alta probabilidad de encontrar un error, no debe ser redundante, no debería ser ni demasiado sencilla ni demasiado compleja

Los productos de software se deben probar de dos formas, utilizando pruebas de caja blanca y pruebas de caja negra, donde con ambas pruebas se busca asegurar la calidad en los productos de software:

La prueba de caja negra: se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información (por ejemplo si ingreso las reservas de citas, en cualquier momento puedo sacar el listado de los pacientes que pidieron cita el mes pasado). Este tipo de prueba la puede realizar cualquier persona, teniendo claro los requerimientos del software.

La prueba de caja blanca se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. Este tipo de prueba la realizan los desarrolladores o personal experto en pruebas, ya que requiere preparación en temas relacionados con desarrollo de software.

Una de las metodologías propuestas por Pressman, para realizar pruebas de software de caja blanca tiene que ver con las pruebas del camino básico que consiste en un método que permite al diseñador de caso de prueba obtener una medida de la complejidad lógica de un diseño procedural y usar esa medida para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

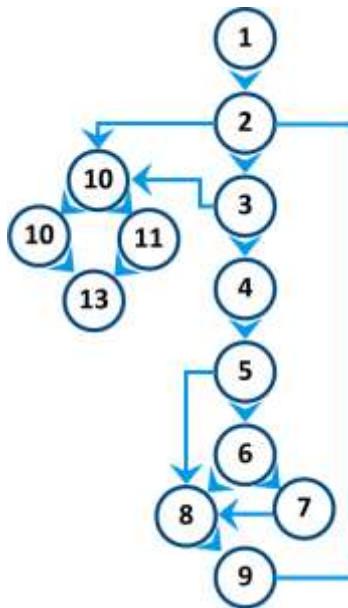
Pasos para aplicar pruebas de caja blanca utilizando el método del camino básico:

Paso 1. Tomar el algoritmo o codificación realizada a través del lenguaje de programación

PROCEDIMIENTO MEDIA: Calcula la media de máximo 100 números que se encuentran en unos límites. También calcula el total de entradas y el total de números válidos.

```
Procedimiento Media(valor,mínimo,máximo,media,tot_entrada,tot_validos)
    i = 1, tot_entrada = 0, tot_validos = 0, suma = 0 (1)
    Mientras valor[i]<>-999 (2) And tot_entrada<100 (3) Hacer
        tot_entrada = tot_entrada + 1 (4)
        Si valor[i]>=mínimo (5) And valor[i]<=máximo (6) entonces
            tot_validos = tot_validos + 1 (7)
            suma = suma + valor[i]
        Fin_si
        i = i + 1 (8)
    Fin_Mientras (9)
    Si tot_validos>0 (10) entonces
        media = suma/tot_validos (11)
    Sino
        media = -999 (12)
    Fin_si (13)Fin_Media
```

Paso 2. Elaborar un diagrama de grafos, donde se transcriba el código



Paso 3. Determinar la complejidad ciclomática del grafo: Existen tres forma de hacerlo y todas deben dar el mismo resultado, se aconseja aplicarlas todas con el fin de verificar el resultado. Dicho resultado permitirá determinar cuántas pruebas se le deben hacer al sistema para estar seguro de que se chequearon todas las posibles opciones.

El número de caminos a buscar es igual a la complejidad ciclomática, la cual está basada en la teoría de grafos y nos da una métrica del software demasiado útil. La complejidad ciclomática se puede calcular de 3 formas:

1. El número de regiones del grafo coincide con la complejidad ciclomática
2. La complejidad ciclomática se define como: $V(G) = A - N + 2$

A: Nº de aristas, N: Nº de nodos

3. La complejidad ciclomática también se define así: $V(G) = P + 1$

P: Nº de nodos predicados

Según lo anterior para cada forma, da como resultado:

1. $V(G) = 6$ regiones
2. $V(G) = 17$ aristas – 13 nodos +2 = 6
3. $V(G) = 5$ nodos predicado + 1 = 6

Lo que significa que al programa se le deben hacer mínimo 6 pruebas para buscar posibles errores.

Paso 4. Determinar un conjunto básico de caminos linealmente independientes:

Camino 1: 1,2,10,11,13

Camino 2: 1,2,10,12,13

Camino 3: 1,2,3,10,11,13

Camino 4: 1,2,3,4,5,8,9,2,...

Camino 5: 1,2,3,4,5,6,8,9,2,...

Camino 6: 1,2,3,4,5,6,7,8,9,2,...

Los puntos suspensivos indican que cualquier camino del resto de la estructura de control es aceptable.

Los nodos 2, 3, 5, 6 y 10 son nodos predicados.

Paso 5 Preparar los casos de prueba que forzarán la ejecución de cada camino del conjunto básico.

Caso de prueba del camino 1:

Valor[k] = entrada válida con $k < i$

Valor[i] = -999 con $i \leq 100$

Resultados esperados: media correcta sobre los k valores y totales adecuados.

Nota: el camino 1 no se puede probar por sí solo; debe ser probado como parte de las pruebas de los caminos 4, 5 y 6.

Caso de prueba del camino 2:

Valor[1] = -999

Resultados esperados: media = -999; otros totales con sus valores iniciales.

Caso de prueba del camino 3:

Intento de procesar 101 valores o más. Los primeros 100 valores deben ser válidos.

Resultados esperados: media correcta sobre los k valores y totales adecuados.

Caso de prueba del camino 4:

Valor[i] = entrada válida donde $i < 100$

Valor[k] < mínimo, para $k < i$

Resultados esperados: media correcta sobre los k valores y totales adecuados.

Caso de prueba del camino 5:

Valor[i] = entrada válida donde $i < 100$

Valor[k] > máximo, para $k \leq i$

Resultados esperados: media correcta sobre los n valores y totales adecuados.

Caso de prueba del camino 6:

Valor[i] = entrada válida donde $i < 100$

Resultados esperados: media correcta sobre los k valores y totales adecuados.

Se ejecuta cada caso de prueba y se comparan los resultados obtenidos con los esperados. Una vez terminados los casos de prueba, el responsable podrá estar seguro de que todas las sentencias del programa se han ejecutado por lo menos una vez.

El propósito de una prueba de software consiste en que dichas pruebas deben estar enfocadas a encontrar y documentar errores existentes, sugerir mejoras según la calidad percibida, validar que los requisitos estén implementados correctamente y completamente.

Las pruebas son realizadas en diferentes estados o niveles de esfuerzo. Estos niveles se distinguen generalmente por el rol de quien las ejecuta y las técnicas utilizadas. Ellas son:

Pruebas Unitarias:

Deben ser implementadas y ejecutadas por el desarrollador.

Se utiliza la técnica de la caja blanca.

Se realizan en las primeras iteraciones de la construcción.

Enfocadas a analizar el control de flujo y datos.

Pruebas de integración:

Deben ser planeadas, implementadas y ejecutadas por el equipo de desarrollo.

Se utiliza la técnica de pruebas ascendentes y descendentes.

Se realizan al final de un ciclo de construcción.

Su objetivo es encontrar fallas al integrar varios componentes.

Pruebas del sistema:

Deben ser planeadas, implementadas y ejecutadas por un equipo independiente llamado SQA.

Se utilizan técnicas de caja negra.

Se realizan durante la fase de construcción.

Su objetivo es encontrar defectos y analizar que la implementación del software está acorde a sus especificaciones.

Pruebas de aceptación:

Deben ser ejecutadas por el usuario final.

Se utiliza la técnica de la caja negra.

Se realizan al final de la fase de transición.

Su objetivo es asegurar que el software está listo y puede ser usado por los usuarios finales.

Las pruebas de software, buscan que los productos de software tengan calidad.

Calidad de Software

“La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. (IEEE, Std. 610-1990).

Es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad del producto informático.

3.7.1 EJERCICIO DEL TEMA:

Teniendo en cuenta los elementos dados para aplicar pruebas de caja blanca.

Aplicar prueba del camino básico, al siguiente algoritmo

1. Dibujar el grafo de flujo.
2. Determinar la complejidad ciclomática del grafo de flujo.
3. Determinar el conjunto de caminos linealmente independientes.
4. Preparar los casos de prueba para cada camino. De acuerdo al vector dado, en caso de buscar.

-Los datos (40, 20, 15).

Índices

1	2	3	4	5
50	40	10	5	20

BUCAR(VEC,N,DATO)

I=1,POS=0

MIENTRAS (I<=N)HAGA

SI(VEC[I]=DATO

 POS=I

 I=N+1

 SINO

 I=I+1

 FIN-SI

FIN-MIENTRAS

SI (POS<>0) ENTONCES

 IMPRIMIR "SI EXISTE"

 SINO

 IMPRIMIR "NO EXISTE"

 FIN-SI

REGRESE

PISTAS DE APRENDIZAJE

Traer a la memoria:



UNA PRUEBA DE SOFTWARE ES EXITOSA CUANDO

Se encuentra el mayor número de errores en cualquiera de las fases de construcción del producto de software.

4 UNIDAD 3 PLATAFORMAS LMS (LEARNING MANAGEMENT SYSTEM O SISTEMA DE GESTIÓN DE APRENDIZAJE)

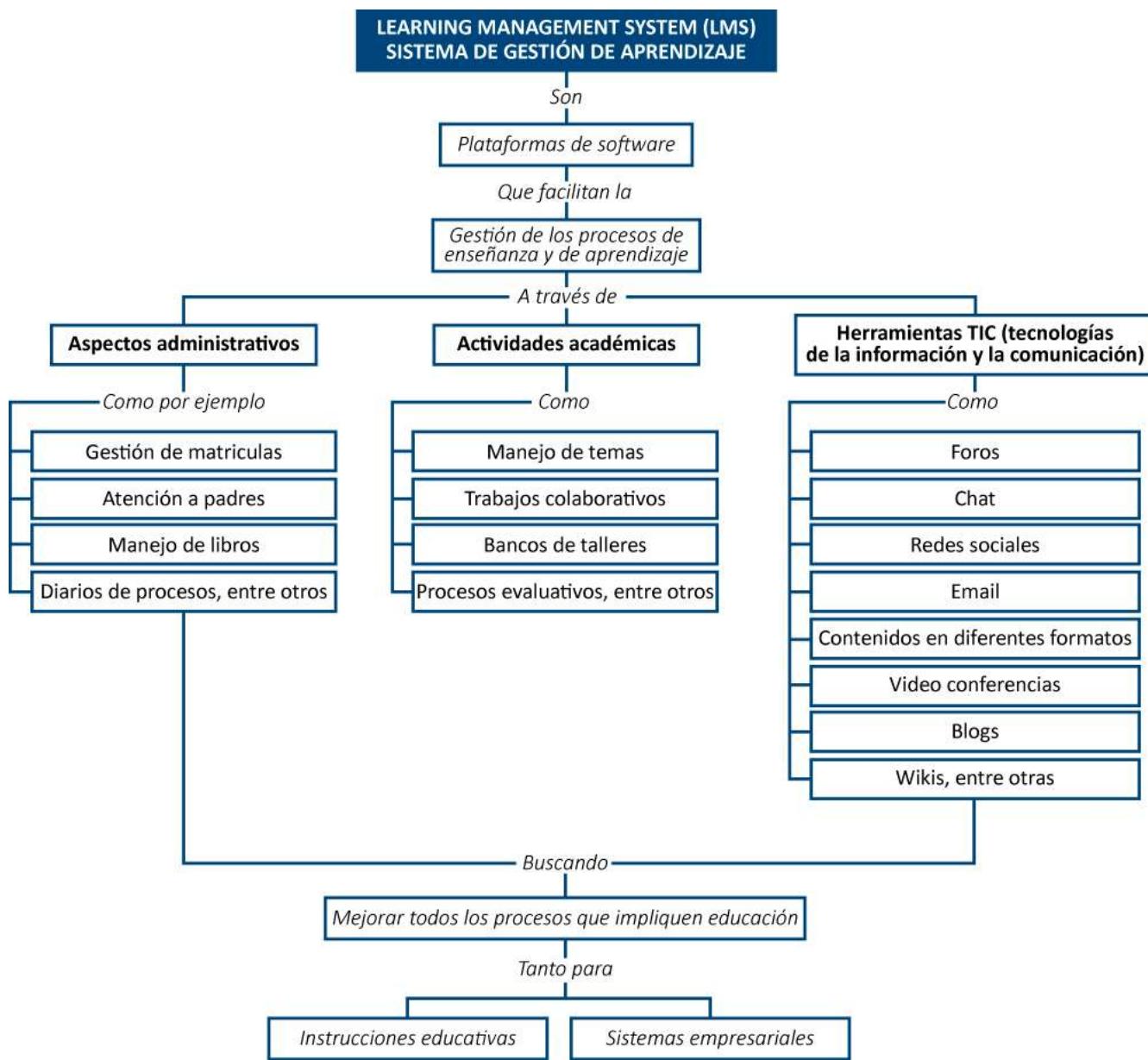
La web se ha constituido en el invento más utilizado en los últimos años y ha transversalizado todos los espacios públicos y privados de la sociedad, convirtiéndose en una herramienta necesaria sobre la cual fluctúa información que se encuentra al servicio de diversos usuarios.

Los Learning Management System o Sistema de Gestión de Aprendizaje, son plataformas basadas en la web, que se orientan a la gestión de contenidos para el aprendizaje a distancia. En el campo educativo se viene utilizando dichas plataformas que permiten la gestión del aprendizaje cuyas funciones principales son la administración, la distribución y el control de las actividades de formación no presencial (virtual) tan normales hoy en una institución u organización y que ha dinamizado los ambientes virtuales de aprendizaje.

Es un programa que ofrece herramientas para organizar materiales y actividades académicas como la gestión de matrículas, realizar seguimientos a cada uno de los procesos de enseñanza y aprendizaje como el manejo de procesos de evaluaciones, bancos de talleres, suministro de herramientas de comunicación on line, puede usarse como plataforma virtual de educación o como complemento a la educación presencial.

Existen diferentes plataformas para cursos web, especiales para el diseño de clases, donde los docentes pueden configurar dicho software para subir sus cursos y demás actividades académicas e interactuar con los estudiantes de forma virtual a través de herramientas como tableros de discusión o foros, sistemas de correos electrónicos, conversaciones On line (chats), contenido en formato de páginas web, archivos PDF entre otros. Algunas de las plataformas más utilizadas actualmente en Colombia son WebCT (Web Course Tools), Moodle, entre otras que vienen tomando fuerza por su facilidad de uso y su utilización como software libre.

Otras herramientas que permiten la dinamización de los procesos educativos, tienen que ver con los blogs, las wikis, computación en nubes, entre otras que le permiten al docente personalizar espacios virtuales con temas de interés, actividades propuestas y material seleccionado para que los usuarios accedan de forma rápida a la información y puedan interactuar a través de trabajos colaborativos fortaleciendo sus procesos de enseñanza y aprendizaje. Aspectos que apenas se vienen incorporando a la vida académica y empresarial.



4.1.1 OBJETIVO GENERAL

Explorar plataformas LMS (Learning Management System), para la adquisición de un conocimiento más claro sobre su utilidad y conformación, así como la manipulación de herramientas en la nube (Cloud Computing) aptas para el intercambio de información On Line.

4.1.2 OBJETIVOS ESPECÍFICOS

Comprender la importancia que tiene la plataforma LMS, para aspectos relacionados con los procesos de enseñanza y aprendizaje, prácticos en la actualidad, tanto para el sector empresarial como el educativo;

realizando comparaciones sobre plataformas LMS, con el fin de que se obtenga claridad sobre dichos productos informáticos y su elección en el momento que se requiera, adquiriendo, además, conocimiento sobre herramientas que se ofrecen en la web y que permiten visionar los campos de acción sobre los cuales se pueden aplicar, acortando distancias, minimizando recursos, fomentando trabajos colaborativos y generando conocimiento.

4.2 TEMA 1 UTILIZACIÓN, PLATAFORMAS Y APLICACIONES COMPLEMENTARIAS

En artículo publicado sobre los LMS (Learning Management System) por la revista de Universidad y sociedad del conocimiento, comenta que:

"Los sistemas de comunicación pueden ser síncronos o asíncronos. Los sistemas síncronos son aquellos que generan comunicación entre usuarios en tiempo real, como podrían ser los chats o las videoconferencias. Los sistemas asíncronos no generan comunicación a tiempo real, pero ofrecen la posibilidad de que las aportaciones de los usuarios queden grabadas.

El correo electrónico y los foros son algunas de las herramientas que usan este tipo de comunicación. Las plataformas de e-learning son el software de servidor que se ocupa principalmente de la gestión de usuarios, cursos y de la gestión de servicios de comunicación.

Los contenidos o courseware es el material de aprendizaje que se pone a disposición del estudiante. Los contenidos pueden estar en varios formatos, en función de su adecuación a la materia tratada en línea con elementos multimedia e interactivos que permiten que el usuario adelante por el contenido lo que aprende." (Boneu, 2007)

Las herramientas LMS, ofrecen una serie de elementos que permiten el apoyo de las actividades académicas como son:

Los foros: Estos permiten el intercambio de mensajes e información entre varios usuarios y obedecen a una programación de temáticas y tiempo establecido para ello.

E-portafolio: consiste en un portafolio digital que permite hacer seguimiento a los estudiantes sobre los diferentes trabajos y participaciones que estos realizan en las diferentes actividades.

Manejo de archivos: Permite enviar y recibir documentos tipo texto, imagen, videos entre formatos.

Chat: Permite el intercambio de mensajes on line.

Correo electrónico: permite el envío y recibo de información a través de diferentes formatos como tipo imagen, texto, datos. Actualmente los servicios de correos electrónicos traen incorporado dentro de la misma cuenta otros servicios como el chat, foro, entre otros.

Video conferencias: Consiste en herramientas que permiten la participación directa de los participantes, donde el receptor también se convierte en emisor, siendo una comunicación sincrónica.

Tableros compartidos: Herramienta donde el docente utiliza un tablero virtual y al mismo tiempo se recibe participación por parte de los estudiantes.

Blogs: Son estructuras colgadas en la red, donde que se organizan con temáticas específicas, con posibilidad de que los receptores realicen su anotaciones u observaciones ante determinadas temáticas planteadas. Normalmente el docente publica información de la asignatura que está orientando.

Webblogs: Son herramientas que se pueden abordar de forma individual o grupal, donde se pide a los estudiantes realizar participaciones de forma periódica, ya sea de forma individual o colectiva.

Wikis: Consiste en herramientas que ofrecen mayor interactividad que los blogs, ya que ofrecen mas espacios de estructuración dentro de la página a través de pestañas que permiten separar temáticas y trabajar con elementos multimedia (imagen, texto, movimiento y sonido). Es una herramienta muy útil para trabajos colaborativos con los estudiantes.

Dichos sistemas LMS, permiten estructuración del curso, manejo de planillas, personalización del entorno a través de diferentes configuraciones y presentación del entorno, manejo de perfiles de usuarios protegido por contraseña, entre otras herramientas que permite dinamizar los diferentes procesos de enseñanza y aprendizaje. Para ampliar el tema realizar lectura del artículo. Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos.(<http://redalyc.uaemex.mx/redalyc/pdf/780/78040109.pdf>).

Redes sociales: Son perfiles que se crean a través de la utilización de sitios web, donde los usuarios intercambian información con amigos personales, profesionales y tratan temas de interés, es el caso de Facebook, Twitter, entre otros. En la actualidad dicha herramienta se emplea normalmente para el entretenimiento con posibilidad de ser aplicada en otros aspectos como el educativo.

En el ambiente académico las herramientas TIC, están orientadas al trabajo colaborativo, donde los estudiantes interactúan entre ellos, buscando responder por actividades académicas que les permita desarrollar sus competencias. Es un campo que aún falta explotarlo para fortalecer los procesos de enseñanza y aprendizaje.

La Cloud Computing (La computación en nubes): La web a partir de los años 90 se constituyó en la mejor herramientas para establecer cualquier tipo de comunicación transacción trabajo, entre otras, siendo un recurso en constante descubrimiento y explotación por las diversas actividades que se pueden derivar de ella, es el caso del concepto Cloud Computing, que aunque no es un concepto nuevo toma fuerza, ya que se interrelacionan varios componentes tecnológicos para la realización de procesos donde una tarea se resuelve con una amplia variedad de tecnologías , incluidas entre ellas las computadoras , las redes, los servidores, y la Internet asociada a estas aplicaciones que son orientadas con diversos fines. En la video conferencia Cloud Computing, expuesta por el Ingeniero Mauricio Sepúlveda en 2011, se puede observar claramente información concerniente a dicha temática donde se habla ampliamente de su utilidad, componentes, herramientas que la apoyan libres y licenciadas, de tal forma que contextualiza suficientemente en el tema en mención.



Cloud Computing [Enlace](#)

4.2.1 EJERCICIO DEL TEMA:

- Qué es Moodle y qué ventajas ofrece.
- Qué es WebCT y qué ventajas ofrece.
- Qué ventajas ofrece la Cloud Computing.

PISTAS DE APRENDIZAJE

Traer a la memoria:



LOS LMS (LEARNING MANAGEMENT SYSTEM)

Son plataformas que permiten estructurar ambientes de aprendizaje, especial para cursos virtuales o para el apoyo a la educación presencial, con la utilización de diferentes herramientas TIC (tecnologías para la información y la comunidad).

5 PISTAS DE APRENDIZAJE

Los sistemas de software:

Se dividen de forma general en:

- Sistemas operativos, que se encargan de administrar todos los recursos de hardware y de software y que un computador sin este software no funciona
- Lenguajes de programación, es un tipo de software especial para desarrollar con el otro software y/o aplicaciones. Que la programación que se utiliza actualmente es la orientada a objetos.
- SGBD, es software que permite toda la administración de los datos de las bases de datos y que el principal activo de toda organización actualmente son los datos.
- Software a aplicación, equivale a todos los otros programas que cumplen con funciones específicas

La ingeniería del software:

Es una rama de la ingeniería informática, que se encarga de todo el proceso que tiene que ver con el desarrollo de productos informáticos, desde la fase de requerimientos, hasta que el software es retirado del mercado y que contiene unas fases que tienen que ver con los requerimientos, el análisis, el diseño, el desarrollo o codificación, las pruebas, los riesgos que pueden afectar el desarrollo del producto, la implementación, pero que todo ello debe apoyarse en herramientas que permite modelar todo el proceso, así como su debida documentación, buscando que el software debe ser de calidad.

Las plataformas LMS (learning management system o sistemas de gestión de aprendizaje)

Son herramientas muy útiles para la administración de los procesos académicos y que permiten ser configuradas para adaptarlas a las necesidades de las instituciones educativas, sirviendo como herramienta dinamizadora del proceso enseñanza y aprendizaje, ya que actualmente se requiere cambiar las metodologías pedagógicas, convirtiéndose en una excelente oportunidad para hacer ajustes que permitan la incorporación de las TIC (tecnologías de la información y comunicación) como mediadoras de dicho proceso

6 GLOSARIO

Driver: Software utilizado para la configuración de dispositivos del computador como tarjetas, sonido, cámara, entre otros.

Extranet: Red corporativa de las empresas, con acceso restringido a través de internet a información propia de la compañía

Hardware: Dispositivos físicos de un computador, toda la parte concreta que se puede tocar.

Línea Base: Pasos o secuencia definido por una empresa para aplicar a los procesos, que sólo se pueden cambiar por quienes tienen la facultad para ello.

LOO: Lenguaje de Programación Orientado a Objetos.

Mainframe: [computadora](#) grande, potente y costosa usada principalmente por una gran empresa para el procesamiento de una gran cantidad de datos.

Personal Computer: Computador personal.

Software: Equivale a los programas del computador, parte intangible-

SO: Sistema Operativo, principal programa de un equipo de cómputo que permite gestionar todos los recursos informáticos.

SGBD: Sistema de Gestión de Base de Datos, que tiene como función administrar los recursos y todo el proceso de la base de datos.

SQA: Sistema de Aseguramiento a la calidad.

UML: Lenguaje de Modelamiento Unificado, que se utiliza para procesos de Ingeniería del Software en lo relacionado con la programación orientada a objetos.

Virus: Programas mal intencionados, que se construyeron para atacar los sistemas informáticos.

7 BIBLIOGRAFÍA

Libros físicos

- Jacobson, Ivar; Booch, Grady; Rumbaugh, James (2000) (en Español). El Proceso Unificado de Desarrollo de Software. Pearson Addison-Wesley.
- Pressman, Roger S. (2003) (en español). Ingeniería del Software, un enfoque Práctico (Quinta edición). Mc Graw Hill. [ISBN 84-481-3214-9](#).
- Pressman, Roger S. 2006, "Ingeniería del Software: Un enfoque práctico", Sexta edición, México DF, Editorial McGraw Hill
- IEEE. (1990). IEEE Standard Glossary of Software Engineering Terminology.
- IEEE/ANSI Standard 610.12-1990: IEEE

Fuentes digitales o electrónicas

Libros Virtuales:

- Boneu Josep. (2007). Artículo. Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos. Consultado el 08 de abril de 2012
[web:<http://redalyc.uaemex.mx/redalyc/pdf/780/78040109.pdf>](http://redalyc.uaemex.mx/redalyc/pdf/780/78040109.pdf)
- Colombia, Hurtado Carmona Douglas. (2011) Teoría General de Sistemas. Un enfoque hacia la Ingeniería de Sistemas. Segunda edición, ISBN: 978-1-257-78193-5. Recuperado 21 de febrero de 2012. Web: <http://www.acis.org.co/archivosAcis/72-392-1-PB.pdf>

Documentos virtuales:

- Concurrencia en un sistema. Consultado 06 de abril de 2012. Web:lrmdavid@exa.unne.edu.ar
- Concurrencia. Consultado 07 de abril de 2012. Sitio Web:
<http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/CONCEM02.htm>
- Diseño de Software. Consultado 07 de abril de 2012.
Web:<http://www.iidia.com.ar/rgm/comunicaciones/c-icie99-ingenieriasoftwareeducativo.pdf>
- Diseño de Software. Consultado 07 de abril de 2012.
Web:<http://indalog.ual.es/mtorres/LP/FundamentosDiseno.pdf>
- Jacobson, I. 1998. "Applying UML in The Unified Process" Presentación. Rational Software. Consultado el 26 de febrero de 2012. Presentación disponible en <http://www.rational.com/uml> como UMLconf.zip
- Lacalle Sáez Alberto, (febrero, 2011) Diseño de Software. Recuperado el 25 de febrero de 2012, del sitio web: <http://albertolacalle.com/disenio.htm>
- Menéndez-Barzanallana Rafael (2005). Gestión de Riesgos. Recuperado el 26 de febrero de 2012, del sitio Web:
http://www.wikilearning.com/curso_gratis/gestion_de_riesgos_en_ingenieria_del_software/3620
- Metaute Paniagua Piedad María. (Febrero de 2012). Módulo sobre Sistemas de Software.
- Salinas Duarte Andrés Ernesto (2007). Obstáculos en la gestión de proyectos en tecnologías de información y comunicación -TICS y posibles soluciones. Recuperado el 26 de febrero de 2012, del sitio Web: http://www.acis.org.co/fileadmin/Articulos/Andres_Salinas.pdf

- Sistemas Operativos. Consultado 06 de abril de 2012.
Web:(http://es.wikipedia.org/wiki/Sistema_operativo)
- Sistema Expertos. Consultado 07 de abril de 2012. Sitio
Web:http://es.wikipedia.org/wiki/Sistema_experto)
- Software. (Febrero de 2012),. Recuperado el 25 de febrero de 2012, del sitio Web: http
- UML. Consultado 06 de abril de 2012. Web:<http://www.slideshare.net/e1da4/diagramas-uml>.
- Zabala Ruiz Jesús. Qué es la Ingeniería del Software. (2011). Artículo de Tesis de Maestría Metropolitan Autonomous University – Iztapalapa, Mexico. Consultado 05 de abril de 2012.
Web:<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>

Videos

- Los sistemas operativos y Diferencias entre sistemas operativos. Consultado 07 de abril de 2012. Web: (http://www.youtube.com/watch?v=o_gEkCuKn5Q&feature=related)
- Sepúlveda Mauricio (2011) Video Cloud Computing, Consultado 07 de abril de 2012, web:
http://www.youtube.com/watch?v=lV_t7LI9m5o
- Sistemas Operativos y la Tecnología. . Consultado 07 de abril de 2012. Web:
(<http://www.youtube.com/watch?v=BWHWoAlrsTA>)
- Evolución de los Lenguajes de Programación . Consultado 07 de abril de 2012. Web:
<http://www.youtube.com/watch?v=qFSUiDGh6RE>
- SGBD . Consultado 07 de abril de 2012. Web: <http://www.youtube.com/watch?v=3R7uwrGTGIQ>
- Formas Normales . Consultado 07 de abril de 2012. Web: <http://www.youtube.com/watch?v=-HajWU4pDLM&feature=related>
- Qué es Ingeniería del Software . Consultado 07 de abril de 2012. Web:
<http://www.youtube.com/watch?v=YFin8nNnARA>
- Concurrencia en un sistemas . Consultado 07 de abril de 2012. Web: lrmdavid@exa.unne.edu.ar
- Sistemas Operativos . Consultado 07 de abril de 2012. Web:
http://es.wikipedia.org/wiki/Sistema_operativo
- Cascada . Consultado 07 de abril de 2012. Web:
http://www.youtube.com/watch?v=BWI66_zLtKE&feature=related
- Incremental . Consultado 07 de abril de 2012. Web:
<http://www.youtube.com/watch?v=zN1paO8PCCw&feature=related>
- Espiral . Consultado 07 de abril de 2012. Web:
<http://www.youtube.com/watch?v=AfOIH31YUYU&feature=related>
- Prototipo . Consultado 07 de abril de 2012. Web:
<http://www.youtube.com/watch?v=cvv0Zj1CPZg&feature=related>
- RUP . Consultado 07 de abril de 2012. Web: <http://www.youtube.com/watch?v=b71uy8aBHHU>
- RUP . Consultado 07 de abril de 2012. Web: <http://www.youtube.com/watch?v=WboyNd-C1u4>
- XP (Programación Extrema) . Consultado 07 de abril de 2012.
Web:<http://www.youtube.com/watch?v=Wt3b1tDN4C8>