



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



**Dirección General de Educación Tecnológica
Industrial y de Servicios**

Dirección Académica e Innovación Educativa

Subdirección de Innovación Académica

Departamento de Planes, Programas y Superación Académica

Anexos para Aprendizajes Esenciales

Módulo IV Submódulo 1

Ofimática



ANEXOS DEL PRIMER PARCIAL

ANEXO N° 1

Instrucciones: Realiza un mapa conceptual. (act. 3)

TEMA: CONCEPTO Y ORIGEN DE LAS BD Y DE LOS SGBD

Las aplicaciones informáticas de los años sesenta acostumbraban a darse totalmente por lotes (batch) y estaban pensadas para una tarea muy específica relacionada con muy pocas entidades tipo.

Cada aplicación (una o varias cadenas de programas) utilizaba ficheros de movimientos para actualizar (creando una copia nueva) y/o para consultar uno o dos ficheros maestros o, excepcionalmente, más de dos. Cada programa trataba como máximo un fichero maestro, que solía estar sobre cinta magnética y, en consecuencia, se trabajaba con acceso secuencial. Cada vez que se quería añadir una aplicación que requiriera el uso de algunos de los datos necesarios (algo que provocaba redundancia) para evitar que los programas tuviesen que leer muchos ficheros.

A medida que se fueron introduciendo las líneas de comunicación, los terminales y los discos, se fueron escribiendo programas que permitían a varios usuarios consultar los mismos ficheros on-line y de forma simultánea. Más adelante fue surgiendo la necesidad de hacer las actualizaciones también on-line.

El acceso on-line y la utilización eficiente de las interrelaciones exigían estructuras físicas que diesen un acceso rápido, como por ejemplo los índices, las multilistas, las técnicas de *hashing*, etc.

Estos **conjuntos de ficheros interrelacionados**, con estructuras complejas y compartidos por varios procesos de forma simultánea (unos on-line y otros por lotes) recibieron al principio el nombre de *Data Banks*, y después, a inicios de los años setenta, el de *Data Bases*. Aquí los denominamos **bases de datos (BD)**.

El **software de gestión de ficheros** era demasiado elemental para dar satisfacción a todas estas necesidades. Por ejemplo, el tratamiento de las interrelaciones no estaba previsto, no era posible que varios usuarios actualizaran datos simultáneamente, etc. La utilización de estos conjuntos de ficheros por parte de los programas de aplicación era excesivamente compleja, de modo que, especialmente durante la segunda mitad de los años setenta, fue saliendo al mercado *software* más sofisticado: los *Data Base Management Systems*, que aquí denominamos **sistemas de gestión de BD (SGBD)**.

Con todo lo que hemos dicho hasta ahora, podríamos definir el término *BD*; una **base de datos de un SI** es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del SI y de sus interrelaciones. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos.



En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelacionales. La representación será única e integrada, a pesar de que debe permitir utilizations varias y simultaneas.

Lista de cotejo para evaluar mapa conceptual actividad 3

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Lista de Cotejo. Primera actividad		
Indicadores	Si	No
Elabora el mapa conceptual del tema expuesto		
Participa y aporta en la lluvia		
Se mantiene el respeto del derecho de palabra.		
Hace aportes a la discusión.		
Claridad en el lenguaje utilizado.		
Estimula la participación y da retroalimentación.		
Se apega al contenido al responder.		
Contextualiza el contenido del texto.		
Total		



ANEXO N° 2

Instrucciones: Responde cuestionario disponible en actividad 4.

TEMA: MODELOS DE DATOS

Bajo la estructura de las bases de datos se encuentra el modelo de datos: una colección de herramientas conceptuales para describir los datos, sus relaciones, su semántica y las restricciones de consistencia.

Los modelos de datos ofrecen un modo de describir el diseño de las bases de datos diferentes. Los modelos de datos pueden clasificarse en cuatro categorías diferentes:

- **Modelo relacional.** El modelo relacional usa una colección de tablas para representar tanto los datos como sus relaciones. Cada tabla tiene varias columnas, y cada columna tiene un nombre único. El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo dado. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla se corresponden con los atributos del tipo de registro. El modelo de los datos relacional es el modelo de datos más ampliamente usado, y una gran mayoría de sistemas de bases de datos actuales se basan en el modelo relacional.
- El **modelo entidad-relación.** El modelo de datos entidad-relación (E-R) se basa en una percepción del mundo real que consiste en una colección de objetos básicos, denominados *entidades*, y de las *relaciones* entre ellos. Una entidad es una “cosa” u “objeto” del mundo real que es distinguible de otros objetos. El modelo entidad-relación se usa mucho en el diseño de bases de datos.
- **Modelo de datos orientado a objetos.** El modelo de datos orientado a objetos es otro modelo de datos que está recibiendo una atención creciente. El modelo orientado a objetos se puede considerar como una extensión del modelo E-R con los conceptos de la encapsulación, los métodos (funciones) y la identidad de los objetos.



- **Modelo de datos semiestructurados.** El modelo de datos semiestructurado permite la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos. Esto lo diferencia de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos. El lenguaje de marcas extensible (XML, eXtensible Markup Language) se emplea mucho para representar datos semiestructurados.
- El **modelo de datos de red** y el **modelo de datos jerárquico** procedieron cronológicamente al relacional. Estos modelos estuvieron íntimamente ligados a la implementación subyacente y complicaban la tarea del modelado de datos. En consecuencia, se usan muy poco hoy en día, excepto en el código de bases de datos antiguas que sigue estando en servicio en algunos lugares.

Diseño de Base de Datos

El diseño de una BD consiste en definir la estructura de los datos. Para ello se suelen seguir por regla general unas fases en el proceso de diseño, definiendo para ello el modelo conceptual, el lógico y el físico.

Lista de cotejo para evaluar cuestionario actividad 4

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Las respuestas son correctas					
La redacción de las respuestas es clara y hay concordancia					
Entrega en tiempo y forma					
Cuida presentación y ortografía.					
Total					



Instrumento para evaluar actividad 5

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Aplica competencia del submodulo 2, modulo 2.					
Reconoce que tipo de datos requiere					
Obtiene estadística					
Entrega en tiempo y forma					
Total					

**Guía de observación para evaluar metacognición .
Actividad 6**

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Lista de Cotejo. Primera actividad		
Indicadores	Si	No
Reconoce que dificultades se encontró para realizar la actividad		
Puede identificar sus propios conocimientos		
Reconoce que nuevas estrategias podría utilizar		
Total		



ANEXOS DEL SEGUNDO PARCIAL

ANEXO N° 3

Instrucciones: Realiza la lectura y elabora un mapa mental (Act 2)

TEMA: CUALIDADES DEL ANALISTA DE SISTEMAS

A partir de las anteriores descripciones de roles, es fácil deducir que un analista de sistemas exitoso debe poseer un amplio rango de cualidades. Aunque los perfiles pueden variar de un caso específico a otro, hay ciertas cualidades que la mayoría de los analistas de sistemas parecen tener.

Por encima de todo, el analista es un solucionador de problemas: una persona que ve el análisis de los problemas como un reto y se divierte al idear soluciones factibles. Cuando sea necesario, el analista debe tener la capacidad de lidiar de manera sistemática con la situación existente mediante la aplicación habilidosa de herramientas, técnicas y experiencia. El analista también debe ser un comunicador capaz de crear relaciones significativas con otras personas durante periodos extendidos de tiempo. Los analistas de sistemas necesitan ser capaces de comprender las necesidades de los humanos al interactuar con la tecnología, además de que necesitan suficiente experiencia con las computadoras como para programar, comprender las capacidades de las computadoras, deducir los requerimientos de información de los usuarios y comunicar lo que se necesita a los programadores. También deben poseer una sólida ética personal y profesional para poder dar forma a las relaciones con sus clientes.

El analista de sistemas debe ser un individuo disciplinado y motivado, y tener capacidad para coordinar tanto a personas como recursos variados para llevar a cabo los proyectos. El análisis de sistemas es una carrera exigente, pero como compensación siempre está en continua evolución y ofrece nuevos retos.

Lista de cotejo para evaluar mapa mental actividad 2

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Enlaza las ideas mediante palabras conectoras					
Utiliza palabras clave					
Se apoya en imágenes					
Cuida presentación y ortografía.					
Total					

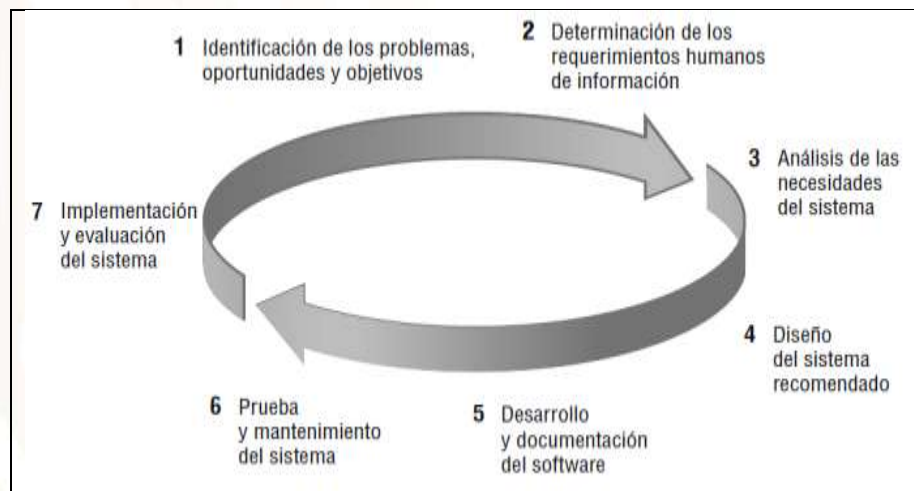
ANEXO N° 4

Instrucciones: Elabora tabla descriptiva (act 3).

TEMA: EL CICLO DE VIDA DEL DESARROLLO DE SISTEMAS

En este capítulo hemos hecho referencia a la metodología sistemática con la que los analistas llevan a cabo el análisis y diseño de los sistemas de información. Gran parte de ello se expresa en lo que conocemos como el ciclo de vida del desarrollo de sistemas (SDLC). El SDLC es una metodología en fases para el análisis y diseño, de acuerdo con la cual los sistemas se desarrollan mejor al utilizar un ciclo específico de actividades del analista y los usuarios.

Los analistas no se han puesto de acuerdo sobre la cantidad de fases que hay en el SDLC, pero por lo general alaban su metodología organizada. En esta imagen se observa que el ciclo está dividido en siete fases:



Aunque cada fase se presenta de manera discreta, en realidad nunca se puede llevar a cabo como un paso separado, sino que varias actividades pueden ocurrir al mismo tiempo, e incluso se pueden repetir.

Incorporación de las consideraciones de la interacción humano-computadora

En años recientes, el estudio de la interacción humano-computadora (HCI) se ha vuelto cada vez más importante para los analistas de sistemas. Aunque la definición sigue evolucionando, los investigadores caracterizan a la HCI como el “aspecto de una computadora que permite las comunicaciones e interacciones entre ella y los humanos. Es el nivel de la computadora que está entre ella y los humanos” (Zhang, Carey, Te’eni & Tremaine, 2005, p. 518). Los analistas que utilizan una metodología HCI se enfocan en las personas en vez del trabajo a realizar o la TI involucrada. Su metodología para un problema es multifacética, ya que analiza los “factores humanos ergonómicos, cognitivos, afectivos y de comportamiento involucrados en las tareas de los usuarios, los procesos de solución de problemas y el contexto de la interacción” (Zhang, Carey, Te’eni & Tremaine, 2005, p. 518). La interacción entre humano y computadora se concentra en las necesidades humanas en vez de enfocarse primero en las necesidades de la organización y del sistema.



Los analistas que adoptan los principios de la HCI examinan una amplia variedad de necesidades en el contexto de los usuarios humanos que interactúan con la tecnología de información para completar sus tareas y resolver problemas.

Aquí también se toman en cuenta los factores físicos o ergonómicos, los cognitivos relacionados con la facilidad de uso, los estéticos, los relacionados con una experiencia de uso agradable, y los aspectos conductuales relacionados con la utilidad del sistema.

La HCI también se considera una metodología centrada en los humanos, que pone a las personas por encima de la estructura o cultura organizacional al crear sistemas. Cuando los analistas emplean la HCI como un lente para filtrar el mundo, su trabajo posee una calidad distinta a la del trabajo de aquellos quienes no poseen esta perspectiva.

La aplicación de los principios de la interacción humano-computadora implica descubrir y resolver las frustraciones que los usuarios experimentan al usar tecnologías de información. Cuando los analistas de sistemas adoptan una metodología HCI, pueden erradicar o minimizar las malas apreciaciones y los errores de diseño que provocan el rechazo de los usuarios hacia los nuevos sistemas o su abandono poco tiempo después de la implementación.

Los investigadores de la HCI observan ventajas al incluir la HCI en cada fase del SDLC. Es una metodología que vale la pena usar y para reflejar esto trataremos de llevar los intereses humanos en forma explícita a cada fase del SDLC. Como estudiante de análisis de sistemas, usted también puede ofrecer una nueva perspectiva al SDLC para identificar las oportunidades que tienen los diseñadores de lidiar con las cuestiones de la HCI y las formas en que los usuarios pueden tener una participación más primordial en cada fase del SDLC.

Fases del ciclo de vida del desarrollo de sistemas

- **Identificación de los problemas, oportunidades y objetivos**

En esta primera fase del ciclo de vida del desarrollo de sistemas, el analista se encarga de identificar correctamente los problemas, las oportunidades y los objetivos. Esta etapa es imprescindible para el éxito del resto del proyecto, ya que a nadie le gusta desperdiciar el tiempo resolviendo un problema mal caracterizado.

Se debe analizar con honestidad lo que está ocurriendo en la empresa. Después, junto con otros miembros de la organización, debe comenzar a señalar los problemas. Las oportunidades residen en las situaciones que el analista cree poder mejorar mediante el uso de sistemas de información computarizados.

Al aprovechar estas oportunidades, la empresa puede obtener una ventaja competitiva o establecer un estándar en la industria. La identificación de los objetivos también es un componente importante de la primera fase. El analista debe descubrir primero qué trata de hacer la empresa; después debe determinar si alguno de los aspectos de las aplicaciones de los sistemas de información puede ayudar a que la empresa logre sus objetivos al enfrentar problemas u oportunidades específicos



Las personas involucradas en la primera fase son los usuarios, los analistas y los administradores de sistemas que coordinan el proyecto. En esta fase las actividades consisten en entrevistar a los encargados de la administración de los usuarios, sintetizar el conocimiento obtenido, estimar el alcance del proyecto y documentar los resultados. El resultado de esta fase es un informe de viabilidad, el cual contiene la definición de un problema y sintetiza los objetivos.

Después, la administración de la empresa debe tomar una decisión en cuanto a proceder o no con el proyecto propuesto. Si el grupo de usuarios no tiene suficientes fondos en su presupuesto o desea hacer frente a problemas que no están relacionados, o si los problemas no requieren un sistema computacional, tal vez se pueda recomendar una solución distinta y el proyecto de sistemas no continúe.

- **Determinación de los requerimientos de información del factor humano**

La siguiente fase es determinar las necesidades de los usuarios involucrados, mediante el uso de varias herramientas, para comprender la forma en que interactúan en el contexto laboral con sus sistemas de información actuales. El analista utilizará métodos interactivos como entrevistas, muestreos e investigación de datos duros, además de los cuestionarios y los métodos discretos, como observar el comportamiento de los encargados al tomar las decisiones y sus entornos de oficina, y los métodos integrales como la creación de prototipos.

Utilizará estos métodos para plantear y responder muchas preguntas relacionadas con la interacción humano-computadora (HCI), incluyendo preguntas tales como: “¿Cuáles son las fortalezas y limitaciones físicas de los usuarios?”, o dicho en otras palabras, “¿qué hay que hacer para que el sistema sea perceptible, legible y seguro?”, “¿cómo puede diseñarse el nuevo sistema para que sea fácil de usar, aprender y recordar?”, “¿cómo puede el sistema ser agradable o incluso divertido de usar?”, “¿cómo puede el sistema apoyar las tareas laborales individuales de un usuario y buscar nuevas formas de hacerlas más productivas?”.

En la fase de requerimientos del SDLC, el analista se esfuerza por comprender qué información requieren los usuarios para realizar sus trabajos. En este punto el analista examina cómo hacer que el sistema sea útil para las personas involucradas. ¿Cómo puede el sistema ofrecer un mejor apoyo para las tareas individuales que se deben llevar a cabo? ¿Qué nuevas tareas habilita el nuevo sistema que los usuarios no podían realizar sin él? ¿Cómo se puede crear el sistema de manera que extienda las capacidades de un usuario más allá de lo provisto por el sistema anterior? ¿Cómo puede el analista crear un sistema gratificante para los trabajadores?

Las personas involucradas en esta fase son los analistas y los usuarios, por lo general los gerentes y los trabajadores de operaciones. El analista de sistema debe conocer los detalles sobre las funciones del sistema actual: el quién (las personas involucradas), el qué (la actividad de la empresa), el dónde (el entorno en el que se lleva a cabo el trabajo), el cuándo (la coordinación) y el cómo (de qué manera particular se realizan los procedimientos actuales) de la empresa a la que está estudiando. Después, el analista debe preguntar por qué la empresa utiliza el sistema actual. Puede haber buenas razones por las cuales la empresa trabaje con los métodos actuales, razón por la que se deben tener en cuenta al diseñar un nuevo sistema.



Al terminar esta fase, el analista deberá comprender la forma en que los usuarios realizan su trabajo al interactuar con una computadora y deberá empezar a comprender cómo mejorar la utilidad y capacidad de uso del nuevo sistema. También deberá saber cómo funciona la empresa y tener información completa sobre personas, objetivos, datos y procedimientos involucrados.

- **Análisis de las necesidades del sistema**

La siguiente fase involucra el análisis de las necesidades del sistema.

Aquí también hay herramientas y técnicas especiales que ayudan al analista a realizar las determinaciones de los requerimientos. Las herramientas como los diagramas de flujo de datos (DFD) para graficar la entrada, los procesos y la salida de las funciones de la empresa, o los diagramas de actividad o de secuencia para mostrar la secuencia de los eventos, sirven para ilustrar a los sistemas de una manera estructurada y gráfica. A partir de los diagramas de flujo de datos, de secuencia u otros tipos de diagramas se debe desarrollar un diccionario de datos para enlistar todos los elementos de datos utilizados en el sistema, así como sus especificaciones.

Durante esta fase, el analista de sistemas también analiza las decisiones estructuradas llevadas a cabo. Las decisiones estructuradas son aquellas para las que se pueden determinar condiciones, alternativas de condición, acciones y reglas de acción. Hay tres métodos principales para el análisis de las decisiones estructuradas: inglés/español estructurado, tablas de decisión y árboles de decisión.

En este punto del SDLC, el analista de sistemas prepara una propuesta de sistemas en la que sintetiza todo lo que ha averiguado sobre los usuarios, la capacidad de uso y la utilidad de los sistemas actuales; incluye un análisis de costo-beneficio de las alternativas y, si se requiere, hace recomendaciones. Si la administración acepta una de las recomendaciones, el análisis continúa por esa vía. Cada problema de sistemas es único, por lo que nunca hay sólo una solución correcta. La manera en que se formule una recomendación o solución depende de las cualidades individuales y la capacitación profesional de cada analista, y de su interacción con los usuarios en el contexto de su entorno laboral.

- **Diseño del sistema recomendado**

En la fase de diseño del SDLC, el analista de sistemas utiliza la información recolectada antes para realizar el diseño lógico del sistema de información. El analista diseña los procedimientos para ayudar a que los usuarios introduzcan los datos con precisión, de manera que los datos que entren al sistema de información sean los correctos. Además, el analista debe ayudar a que los usuarios completen la entrada de datos efectiva al sistema de información mediante el uso de las técnicas del buen diseño de formularios y páginas Web o pantallas.



Parte del diseño lógico del sistema de información es idear la HCI. La interfaz conecta al usuario con el sistema, por lo que es extremadamente importante. La interfaz del usuario se diseña con ayuda de los usuarios para asegurar que el sistema sea perceptible, legible y seguro, así como atractivo y divertido de usar.

Ejemplos de interfaces de usuario físicas son el teclado (para escribir las preguntas y respuestas), los menús en pantalla (para obtener los comandos de los usuarios) y varios tipos de interfaces gráficas de usuario (GUI) basadas en un ratón o una pantalla táctil.

La fase de diseño también incluye el diseño de bases de datos que almacenarán gran parte de los datos necesarios para los encargados de tomar las decisiones en la organización. Los usuarios se benefician de una base de datos bien organizada que sea lógica para ellos y se corresponda con la forma en que ven su trabajo. En esta fase, el analista también trabaja con los usuarios para diseñar una salida (ya sea en pantalla o impresa) que cumpla con sus necesidades de información.

Por último, el analista debe diseñar controles y procedimientos de respaldo para proteger el sistema y los datos, y para producir paquetes de especificación de programas para los programadores. Cada paquete debe contener los diseños de las entradas y las salidas, las especificaciones de los archivos y los detalles sobre el procesamiento; también puede incluir árboles o tablas de decisión, UML o diagramas de flujo de datos, junto con los nombres y las funciones de cualquier código previamente escrito dentro de la empresa o que utilice código u otras bibliotecas de clases.

- **Desarrollo y documentación del software**

En la quinta fase del SDLC, el analista trabaja con los programadores para desarrollar el software original requerido. Durante ella, el analista desarrolla junto con los usuarios una documentación efectiva para el software, incluyendo manuales de procedimientos, ayuda en línea, sitios Web con preguntas frecuentes (FAQ) y archivos Léame (Read Me) para incluir con el nuevo software. Como los usuarios están involucrados desde el principio, la fase de documentación debe lidiar con las preguntas que hicieron y resolvieron junto con el analista. La documentación indica a los usuarios cómo deben usar el software y qué deben hacer en caso de que ocurran problemas.

Los programadores desempeñan un rol clave en esta fase, ya que diseñan, codifican y eliminan los errores sintácticos de los programas de computadora. Para asegurar la calidad, un programador puede llevar a cabo un recorrido por el diseño o por el código para explicar las porciones complejas del programa a un equipo formado por otros programadores.



- **Prueba y mantenimiento del sistema**

Antes de utilizar el sistema de información, se debe probar. Es mucho menos costoso detectar los problemas antes de entregar el sistema a los usuarios. Una parte del procedimiento de prueba es llevado a cabo por los programadores solos; la otra la realizan junto con los analistas de sistemas. Primero se completa una serie de pruebas para señalar los problemas con datos de muestra y después se utilizan datos reales del sistema actual. A menudo, los planes de prueba se crean en las primeras etapas del SDLC y se refinan a medida que el proyecto progresa.

El mantenimiento del sistema y la documentación de este mantenimiento empieza en esta fase y se lleva a cabo de manera rutinaria durante toda la vida del sistema de información.

Gran parte del trabajo rutinario del programador consiste en el mantenimiento, por lo cual las empresas invierten una gran cantidad de dinero en este proceso. Ciertos procedimientos de mantenimiento, como las actualizaciones de los programas, se pueden llevar a cabo a través del sitio Web del distribuidor.

Muchos de los procedimientos sistemáticos que emplea el analista durante el SDLC pueden ayudar a asegurar que el mantenimiento siempre se mantenga en el nivel mínimo necesario.

- **Implementación y evaluación del sistema**

En esta última fase del desarrollo de sistemas, el analista ayuda a implementar el sistema de información. En esta fase hay que capacitar a los usuarios para operar el sistema. Los distribuidores se encargan de una parte de la capacitación, pero la supervisión de la capacitación es responsabilidad del analista de sistemas. Además, el analista necesita planear una conversión sin problemas del sistema antiguo al nuevo. Este proceso incluye convertir los archivos de los formatos anteriores a los nuevos, o crear una base de datos, instalar equipo y llevar el nuevo sistema a producción.

Instrumento de evaluación para actividad 3

Lista de cotejo para evaluar tabla descriptiva

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Incluye información proporcionada en el video					
Es preciso y conciso en cada descripción					
Entrega en tiempo y forma					
Cuida presentación y ortografía.					
Total					



Instrumento de evaluación para actividad 4
Lista de cotejo para evaluar presentación electrónica

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Selecciona textos e imágenes pertinentes					
La presentación es creativa					
La información clara y proviene de fuentes confiables					
Entrega en tiempo y forma					
Cuida presentación y ortografía.					
Total					

ANEXO N° 5

Instrucciones: Realiza mapa mental (act. 5).

TEMA: ETAPAS DEL DISEÑO DE BASES DE DATOS

El diseño de una base de datos no es un proceso sencillo. Habitualmente, la complejidad de la información y la cantidad de requisitos de los sistemas de información hacen que sea complicado. Por este motivo, cuando se diseñan bases de datos es interesante aplicar la vieja estrategias de dividir para vencer.

Por lo tanto, conviene descomponer el proceso del diseño en varias etapas; en cada una se obtiene un resultado intermedio que sirve de punto de partida de la etapa siguiente, y en la ultima etapa se obtiene el resultado deseado. De este modo no hace falta resolver de golpe toda la problemática que plantea el diseño, sino que en cada etapa afronta un solo tipo de subproblema. Así se divide el problema y, al mismo tiempo se simplifica el proceso.

Descompondremos el diseño de bases de datos en tres etapas:

- 1) **Etapas del diseño conceptual:** en esta etapa se obtiene una estructura de la información de la futura BD independiente de la tecnología que hay que emplear. No se tiene en cuenta todavía que tipo de base de datos se utilizara – relacional, orientada a objetos, jerárquica, etc.-; en consecuencia, tampoco se tiene en cuenta con que SGBD ni con que lenguaje concreto se implementara la base de datos.



Así pues, la etapa del diseño conceptual nos permite concentrarnos únicamente en la problemática de la estructuración de la información, sin tener que preocuparnos al mismo tiempo de resolver cuestiones tecnológicas.

El resultado de la etapa del diseño conceptual se expresa mediante algún modelo de datos de alto nivel. Uno de los más empleados es el modelo entidad-relación (*entity-relationship*), que abreviaremos con la sigla ER.

- 2) **Etapa del diseño lógico:** en esta etapa se parte del resultado del diseño conceptual, que se transforma de tal manera que se adapte a la tecnología que se debe emplear. Mas concretamente, es preciso que se ajuste al modelo del SGBD con el que se desea implementar la base de datos. Por ejemplo, si se trata de un SGBD relacional, esta etapa obtendrá un conjunto de relaciones con sus atributos, claves primarias y claves foráneas.

- 3) **Etapa del diseño físico:** en esta etapa se transforma la estructura obtenida en la etapa del diseño lógico, con el objetivo de conseguir una mayor eficiencia; además, se completa con aspectos de implementación física que dependerán del SGBD.

Por ejemplo, si se trata de una base de datos relacional, la transformación de la estructura puede consistir en lo siguiente: tener almacenada alguna relación que sea la combinación de varias relaciones que se han obtenido en la etapa del diseño lógico, partir una relación en varias, añadir algún atributo calculable a una relación, etc. Los aspectos de implementación física que hay que completar consisten normalmente en la elección de estructuras físicas de implementación de las relaciones, la selección del tamaño de las memorias intermedias (*buffers*) o de las paginas, etc.

Lista de cotejo para evaluar mapa mental actividad 5

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Enlaza las ideas mediante palabras conectoras					
Utiliza palabras clave					
Se apoya en imágenes					
Cuida presentación y ortografía.					
Total					



ANEXO N° 6

TEMA: DISEÑO CONCEPTUAL: EL MODELO ER

En este apartado trataremos el diseño conceptual de una base de datos mediante el modelo ER. Lo que es aplicable al diseño de cualquier tipo de base de datos -relacional, jerárquica, etc.-, porque, como ya se ha dicho, en la etapa del diseño conceptual todavía no se tiene en cuenta la tecnología concreta que se utilizara para implementar la base de datos.

El modelo ER es uno de los enfoques de modelización de datos que mas se utiliza actualmente por simplicidad y legibilidad. Su legibilidad se ve favorecida porque proporciona una notación diagramática muy comprensiva. Es una herramienta útil tanto para ayudar al diseñador a reflejar en un modelo conceptual los requisitos del mundo real de interés como para comunicarse con el usuario final sobre el modelo conceptual obtenido y, de este modo, poder verificar si satisface sus requisitos.

¿Qué es un diagrama ER?

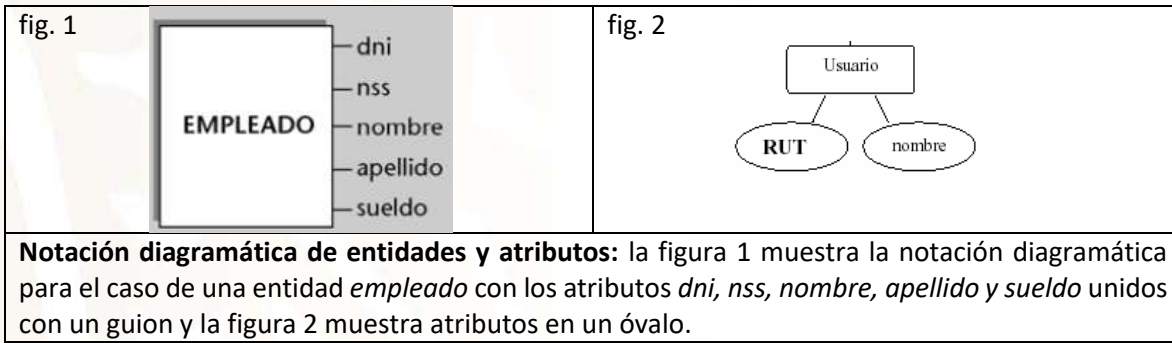
Es un tipo de diagrama de flujo que ilustra como las “entidades”, como personas, objetos o conceptos, se relacionan entre si dentro de un sistema. Los diagramas ER se usan a menudo para diseñar o depurar bases de datos relacionales.

Son un reflejo de la estructura gramatical y emplean entidades como sustantivos y relaciones como verbos.

El termino *entidad* se utiliza tanto para denominar objetos individuales como para hacer referencia a conjuntos de objetos similares de los que nos interesan los mismos atributos; es decir, que, por ejemplo, se utiliza para designar tanto a un empleado concreto de una empresa como al conjunto de todos los empleados de la empresa. Mas concretamente, el termino *entidad* se puede referir a las **instancias u ocurrencias concretas** (empleados concretos) o a **tipos o clases de entidades** (el conjunto de todos los empleados).

El modelo ER proporciona una **notación diagramática** para representar gráficamente las entidades y sus atributos:

- Las **entidades** se representan con un rectángulo. El nombre de la entidad se escribe en mayúsculas dentro del rectángulo.
- Los **atributos** se representan mediante su nombre en minúscula unida con un guion al rectángulo de la entidad a la que pertenecen o también pueden estar representados en óvalos.



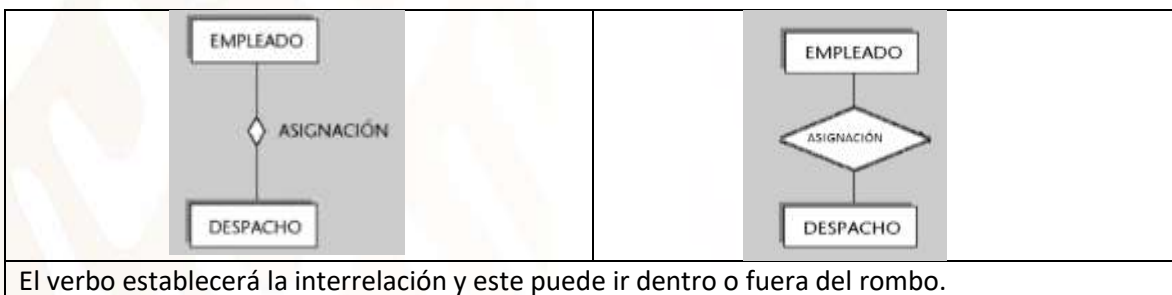
- Se define **interrelación** como una asociación entre entidades. Las interrelaciones se representan en los diagramas del modelo ER mediante un rombo. Junto al rombo se indica el nombre de la interrelación con letras mayúsculas.

Ejemplo de interrelación

Considerando una entidad *empleado* y una entidad *despacho* y supongamos que a los empleados se les asignan despachos donde trabajar. Entonces hay una interrelación entre la entidad *empleado* y la entidad *despacho*.

Esta interrelación, que podríamos denominar *asignación*, asocia a los empleados con los despachos donde trabajan. Estas interrelaciones se representan en los diagramas del modelo ER mediante un rombo.

La figura a continuación muestra la interrelación *asignación* entre las entidades *empleado* y *despacho*.

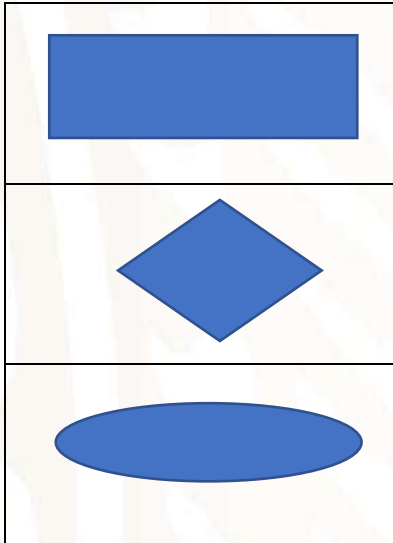




ANEXO N° 6.1

ACTIVIDAD: SIMBOLOGIA DE DIAGRAMAS E-R

Instrucciones: Relaciona con una línea cada símbolo utilizado en el diagrama E-R con su respectiva descripción (act. 6).



Permite reconocer la interacción entre las personas, objetos, etc.

Una propiedad o característica de las personas, objetos, etc.

Se puede definir, como una persona, objeto, concepto u evento, que puede tener datos almacenados acerca de este.

Lista de cotejo para evaluar actividad 6

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Identifica claramente cada representación grafica					
Reconoce que es una entidad, atributo y relación					
Entrega en tiempo y forma					
Es un trabajo limpio.					
Total					



ANEXO N° 6.2

Instrucciones: une y arma correctamente el rompecabezas para formar el diagrama e-r (act. 7).

ACTIVIDAD: DIAGRAMAS E-R

Inciso a:

empleado	cat	codigo	Prof.	pertenece
depto	nombre	nombre	director	

Inciso b:

prov	nombre	color	direccion	suministra
pieza	nombre	ciudad	provincia	

Lista de cotejo para evaluar diagramas e-r actividad 7 y 8

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Reconoce gráficamente que es una entidad					
Reconoce gráficamente que es un atributo					
Reconoce gráficamente que es una relación					
El diagrama esta estructurado coherentemente					
Entrega en tiempo y forma					
Total					



ANEXO N° 7

Instrucciones: Realiza mapa conceptual (act. 9).

TEMA: DISEÑO LOGICO

La etapa de diseño lógico consiste en obtener un esquema lógico a partir del esquema conceptual generado en la etapa anterior. El esquema lógico depende del tipo de base de datos elegido, aunque es independiente de la implementación concreta del sistema de gestión de bases de datos. el modelo lógico recibe el nombre de modelo lógico relacional o, simplemente, modelo relacional.

Este modelo contiene más detalle que el modelo ER conceptual, sin tener en cuenta cómo la información se implementará físicamente en la base de datos.

Los modelos de datos lógicos incluyen todas las entidades y relaciones entre ellas, especifican atributos para cada entidad, especifican claves primarias para cada entidad, especifican claves foráneas, las cuales identifican la relación entre diferentes entidades e involucran la normalización, que es el proceso de eliminación de redundancia en una tabla para que la tabla sea más fácil de modificar.

En este punto ya habremos creado cada tabla, una a una, siguiendo lo identificado en el diagrama E-R y estableciendo índices y demás elementos según las indicaciones de cada uno de los campos. Además, hemos decidido el mejor tipo de datos que podemos aplicar a cada campo (texto, números, fechas... que se almacenan para cada registro).

El objetivo del diseño lógico es convertir los esquemas conceptuales locales en un esquema lógico global que se ajuste al modelo de SGBD sobre el que se vaya a implementar el sistema. Mientras que el objetivo fundamental del diseño conceptual es la compleción y expresividad de los esquemas conceptuales locales, el objetivo del diseño lógico es obtener una representación que use, del modo más eficiente posible, los recursos que el modelo de SGBD posee para estructurar los datos y para modelar las restricciones.

Esto quiere decir que, en esta fase del diseño, se estará realizando una reestructuración en donde se tienen en cuenta los aspectos más ligados con el nivel físico, y que consiste en modificar el esquema obtenido en la fase anterior (en la fase del diseño conceptual) para adaptarlo a las consideraciones de eficiencia. En esta etapa se consideran (debido a la falta de flexibilidad de los SGBD) los aspectos más relacionados con el nivel físico.



Lista de cotejo para evaluar mapa conceptual actividad 9

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Lista de Cotejo. Primera actividad		
Indicadores	Si	No
Elabora el mapa conceptual del tema expuesto		
Participa y aporta en la lluvia		
Se mantiene el respeto del derecho de palabra.		
Hace aportes a la discusión.		
Claridad en el lenguaje utilizado.		
Estimula la participación y da retroalimentación.		
Se apega al contenido al responder.		
Contextualiza el contenido del texto.		
Total		

ANEXO N° 8

Instrucciones: Realiza lista de pasos para convertir un diagrama e-r al relacional (act. 10).

TEMA: TRANSFORMACION DEL MODELO ER AL MODELO RELACIONAL

En este apartado trataremos el diseño lógico de una base de datos relacional. Partiremos del resultado de la etapa del diseño conceptual expresado mediante el modelo ER y veremos como se puede transformar en una estructura de datos del modelo relacional.

Introducción a la transformación de entidades e interrelaciones

Los elementos básicos del modelo ER son las entidades y las interrelaciones:

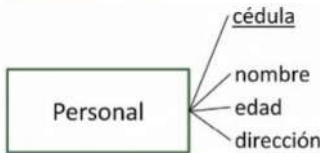
- a) Las entidades, cuando se traducen al modelo relacional, originan relaciones.
- b) Las interrelaciones, en cambio cuando se transforman, pueden dar lugar a claves de alguna relación ya obtenida o pueden dar lugar a nuevas relaciones.

Transformación de entidades

Empezaremos el proceso transformando todas las entidades de un modelo ER adecuadamente. Cada entidad del modelo ER se transforma en una relación del modelo relacional. Los atributos de la entidad serán atributos de la relación y, de forma análoga la clave primaria de la entidad será la clave primaria de la relación

- **Por cada entidad se crea una tabla.**
Por cada atributo simple se crea un atributo en la tabla.
¿Cuál es la clave primaria? Se selecciona uno de los atributos determinantes de la entidad como clave primaria de la tabla.

Ejemplo 1:



Personal (cedula, nombre, edad, dirección)

Tabla: Personal

Cedula	Nombre	edad	Dirección
0256584	José Gómez	24	México
0258789	Erick Diaz	23	Chiapas

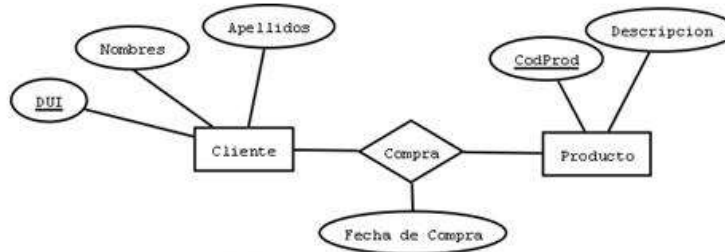
La entidad Personal pasa a ser la tabla y las columnas muestran los atributos (cedula, nombre, edad, dirección)

La transformación de un diagrama ER en un esquema relacional:

- Se transforman en tablas todos los tipos de entidades y relaciones que aparecen en el diagrama ER.
- Se seleccionan las claves primarias para cada una de las tablas de nuestro esquema lógico.
- Se fusionan aquellas tablas que compartan su clave primaria.

Ejemplo 2:

Supongamos el siguiente modelo entidad-relación para una relación de muchos a muchos:



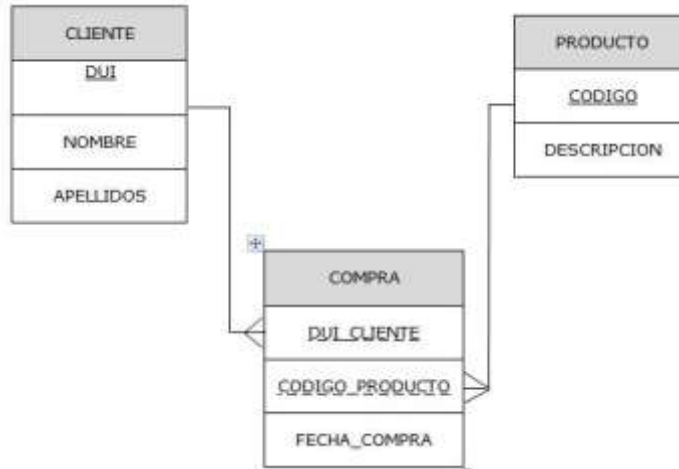


En este caso la relación “compra” se transforma en una nueva tabla cuya clave primaria estará formada por los atributos DUI que es la clave primaria de cliente y CODIGO que es la clave primaria de producto. Además, tendrá como campo fecha compra, ya que este atributo forma parte de la relación.

El modelo relacional quedaría de la siguiente forma:

- CLIENTE (**DUI**, nombre, apellidos)
- PRODUCTO(**código**, descripción)
- COMPRA(**DUI_cliente, código_producto**, fecha_compra)

Y representado en tabla se tendría lo siguiente (sin normalizar):



Instrumento de evaluación actividad 10 y 11

Lista de cotejo para evaluar interpretación del modelo e-r al relacional

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Reconoce las características del modelo relacional					
Identifica cada elemento del modelo relacional					
Muestra un claro nivel de entendimiento de un modelo e-r al relacional					
Se puede interpretar claramente cada esquema					
Total					



ANEXOS DEL TERCER PARCIAL

ANEXO N° 9

Instrucciones: Analiza la lectura responde los cuestionamientos de la actividad 1.

TEMA: ¿QUÉ SON REQUERIMIENTOS?

Se presenta a continuación la definición existente en el glosario de la IEEE de los que es un “Requerimiento”:

- a) “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”. (Std 610.12-1900, IEEE:62)
- b) “Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”. (Std 610.12-1900, IEEE:62)

También Ian, Sommerville presenta una definición acerca de lo que es un “Requerimiento”:

- c) “Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este” (Sommerville, 2005:108).

Analizando las definiciones anteriores, un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

REQUISITO DE COMPONENTE

Equipo y procesador

Windows OS: 1.6 GHz o más rápido, dos núcleos

macOS: Procesador Intel

Memoria

Windows OS: 4 GB de RAM; 2 GB de RAM (32 bits)

macOS: 4 GB de RAM

Disco duro

Windows OS: 4 GB de espacio disponible en disco

macOS: 10 GB de espacio en disco disponible. Disco duro con formato HFS+ (también conocido como Mac OS Extended) o APFS Las actualizaciones podrían requerir de almacenamiento adicional con el tiempo.

Pantalla

Windows OS: Resolución de pantalla de 1280 x 768 (32 bits requiere aceleración de hardware para 4K y superior)

macOS: Resolución de pantalla de 1280 x 800

Gráficos

Windows OS: La aceleración de hardware de gráficos necesita DirectX 9 o versiones posteriores, con WDDM 2.0 o posteriores para Windows 10 (o bien WDDM 1.3 o posteriores para Windows 10 FallCreatorsUpdate).
macOS: Sin requisitos de gráficos.

Sistema operativo

Windows OS: Windows 10, Windows 8.1, Windows Server 2019, Windows Server 2016

macOS: Una de las tres versiones más recientes de macOS. Cuando se publica una nueva versión principal de macOS, la versión de macOS y las dos anteriores

Instalar Office 2016 en un equipo PC

Recuerde que una versión de compra de pago único de Office solo tiene licencia para una instalación.

Según el navegador, haga clic en Ejecutar (en Microsoft Edge o Internet Explorer), Instalar (en Chrome) o Guardar archivo (en Firefox).

Si ve el aviso del Control de cuentas de usuario con el mensaje ¿Quieres permitir que esta aplicación haga cambios en el dispositivo? haga clic en Sí.

Se iniciará la instalación.



La instalación habrá finalizado cuando vea la frase "Ya está listo. Office está instalado" y se reproduzca una animación para mostrarle en qué lugar del equipo podrá encontrar las aplicaciones de Office. Seleccione Cerrar.

Siga las instrucciones en la ventana para buscar las aplicaciones de Office. Por ejemplo, según la versión de Windows, seleccione Inicio y desplácese para buscar la aplicación que desea abrir, como Word o Excel, o escriba el nombre de la aplicación en el cuadro de búsqueda.





Lista de cotejo para evaluar cuestionario actividad 1

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Las respuestas son correctas					
La redacción de las respuestas es clara y hay concordancia					
Entrega en tiempo y forma					
Cuida presentación y ortografía.					
Total					

Instrumento para evaluar actividad 2

Lista de cotejo para evaluar reflexión.

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Lista de Cotejo. Primera actividad		
Indicadores	Si	No
Escribe la reflexión de forma clara y coherente. (2%)		
Muestra un nivel de comprensión bueno en el análisis comparativo del aula de clases tradicional y aula virtual.(2%)		
En el escrito se observan afirmaciones donde se reflejan las semejanzas y diferencias más relevantes de los elementos a comparar (2%)		
Muestra argumento sólidos para defender sus puntos de vista en cuanto los tipos de aulas. (2%)		
El escrito es entregado en tiempo y forma (2%)		
Total		

ANEXO N° 10

Instrucciones: identifica y escribe las reglas para llegar a la Primera Forma Normal. (act 3).

TEMA: NORMALIZACION DE BASES DE DATOS

La normalización es la transformación de las vistas de usuario complejas y del almacén de datos a un juego de estructuras de datos más pequeñas y estables. Además de ser más simples y estables, las estructuras de datos son más fáciles de mantener que otras estructuras de datos (Kendall, 2005).

Primera forma normal (1FN)

La primera regla de normalización se expresa generalmente en forma de dos indicaciones separadas.

1. Todos los atributos, valores almacenados en las columnas, deben ser indivisibles.
2. No deben existir grupos de valores repetidos.

*El valor de una columna debe ser una **entidad atómica**, indivisible excluyendo así las dificultades que podría conllevar el tratamiento de un dato formado en varias partes.*

Supongamos que tienes en una tabla una columna Dirección se utiliza para almacenar la dirección completa, dato que se compondría del nombre de la calle, el número exterior, el número interior (puerta), el código postal, el estado y la capital.

id	Nombre	Dirección	Teléfono	URL
1	Anaya	Jl: Luca	92199932	Anaya.com
2	Pericles	C/Luna # 20-28018 Tlaxcala	99299492	Pericles.com

↓

Calle	Número	Puerta	CP	Población	Provincia
Luna	20		28018	Tlaxcala	Tlaxcala

Figura 1. Tabla con un atributo divisible en varias partes.

Una tabla con esta estructura plantea problemas a la hora de recuperar información. Imagina que necesitas conocer todas las entradas correspondientes a una determinada población, o que quieres buscar a partir del código postal. Al ser la dirección completa una secuencia de caracteres de estructura libre no resultaría nada fácil.

Existirán más columnas, pero cada una de ellas contendrá un valor simple e indivisible que facilitará la realización de las operaciones antes mencionadas.



Instrumento para evaluar actividad 3

Lista de cotejo para reglas de normalización

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Identifica los pasos de la 1 forma normal					
Comprende la importancia de la normalización					
Entrega en tiempo y forma					
Cuida presentación y ortografía.					
Total					

ANEXO N° 11

Instrucciones: Realiza mapa conceptual (act 4).

TEMA: DISEÑO FÍSICO DE BASES DE DATOS

El diseño físico forma parte del proceso general del diseño de una base de datos. En particular, se realiza después del diseño lógico de la misma. En el diseño lógico se genera un conjunto de tablas normalizadas a partir del esquema conceptual creado en la fase de diseño conceptual de la base de datos. El diseño físico empieza en el momento en que dichas tablas son creadas, y se centra en definir los métodos de almacenamiento y acceso a los datos que permitan operar con la base de datos con una eficiencia esperada. Una vez que el diseño físico se ha realizado, el siguiente paso es implementar la base de datos y monitorizarla. Esta tarea de monitorización se encargará de comprobar si la base de datos satisface el rendimiento esperado. De no ser así, se deberán realizar modificaciones para mejorarlo. También puede ser necesario modificar el esquema para ajustar la base de datos a nuevos requerimientos. El responsable de realizar las tareas previamente descritas es el administrador de la base de datos, quien tiene a su disposición diferentes herramientas suministradas por los vendedores de SGBD.

Tabla normalizada

Una tabla está normalizada cuando representa un concepto único del mundo real. Si el diseño conceptual de la base de datos es correcto, las tablas resultantes en el diseño lógico de la misma cumplen esta condición. Una tabla no normalizada padece redundancias, es decir, repeticiones de los datos que serían evitables.

Conceptualmente podemos definir el **diseño físico** de una base de datos como un proceso que, a partir de su diseño lógico y de información sobre su uso esperado, crea una configuración física de las bases de datos adaptada al entorno donde se alojara y que permita el almacenamiento y la explotación de los datos de la base de datos con un rendimiento adecuado.



A continuación vamos a estudiar a detalle esta definición para entender a fondo el concepto de diseño físico:

- a) **Entradas:** el diseño físico parte de la siguiente información:
 - **El esquema lógico:** es el resultado del diseño lógico de la base de datos y contiene la lista de tablas necesarias para almacenar la información relevante, incluyendo la menos, sus columnas, claves primarias y foráneas.
 - **Información sobre el uso esperado de la base de datos:** estimación sobre los volúmenes de datos y transacciones que el sistema deberá procesar. También deberá contener una estimación sobre las operaciones SQL que se utilizarán, sobre qué datos, con qué frecuencia, y la calidad de servicio esperada.
- b) **Salida:** se tomará un conjunto de decisiones sobre las estructuras físicas más adecuadas a utilizar, generando una configuración física de la base de datos. Esta configuración podrá estar compuesta por una combinación adecuada de los espacios de almacenaje, un conjunto de índices para mejorar el rendimiento de las consultas, un particionamiento de tablas adecuado, un conjunto de vistas materializadas y otros elementos adicionales, como por ejemplo disparadores que regulen reglas de negocio complejas o procedimientos almacenados en la base de datos.
- c) **Adaptado al entorno:** el diseño físico puede verse influido por el SGBD donde se implemente la base de datos, los dispositivos de almacenamiento no volátil donde se guarden los datos de la base de datos, y el entorno hardware donde se aloje el SGBD.
 - **SGBD:** el paso de un diseño lógico a físico requiere un conocimiento profundo del SGBD donde se vaya a implementar la base de datos. En particular, entre otros, se deberá tener un conocimiento de los siguientes elementos:
 - Soporte ofrecido a la integridad referencial.
 - Tipos de índices disponibles.
 - Tipos de datos disponibles
 - Tipos de restricciones de integridad disponibles
 - Parámetros de configuración que puedan afectar el rendimiento, como por ejemplo el tamaño de página y la gestión de datos y concurrencia utilizados.
 - Construcciones SQL disponibles de soporte al diseño físico.
 - Particularidades del SGBD utilizado (y en consecuencia del lenguaje SQL) en relación a la definición de elementos relacionados con el diseño físico de la base de datos.
- d) **Que permita el almacenamiento y explotación de los datos con un rendimiento adecuado:** el objetivo del diseño físico es obtener un buen rendimiento de la base de datos en un entorno real. Con rendimiento nos referimos básicamente al tiempo de respuesta a operaciones de consulta o actualización, a la carga de transacciones a procesar y a la disponibilidad de la base de datos.



Elementos de diseño físico

Los elementos ofrecidos por el SGBD para afinar el modelo físico de la base de datos son principalmente los siguientes:

- ✓ Espacios para tablas
- ✓ Índices
- ✓ Vistas materializadas
- ✓ Particiones

Lista de cotejo para evaluar mapa conceptual actividad 4

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Lista de Cotejo. Primera actividad		
Indicadores	Si	No
Elabora el mapa conceptual del tema expuesto		
Participa y aporta en la lluvia		
Se mantiene el respeto del derecho de palabra.		
Hace aportes a la discusión.		
Claridad en el lenguaje utilizado.		
Estimula la participación y da retroalimentación.		
Se apega al contenido al responder.		
Contextualiza el contenido del texto.		
Total		



Instrumento para evaluar actividad 5

Lista de cotejo para evaluar definición de conceptos

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Aspecto a observar	Cumple			Observaciones	%
	SI	NO	NA		
Se expresa de manera coherente en cada respuesta					
Utiliza expresiones propias					
Entrega en tiempo y forma					
Cuida presentación y ortografía.					
Total					

ANEXO N° 12

Instrucciones: Realiza un mapa conceptual (act. 6).

TEMA: ¿QUÉ ES UN GESTOR DE DATOS Y PARA QUÉ SIRVE?

Un Sistema Gestor de Base de Datos (SGBD) o DGBA (Data Base Management System) es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos. Los gestores de base de datos o gestores de datos hacen posible administrar todo acceso a la base de datos ya que tienen el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones.

¿Qué es el gestor de datos?

El gestor de datos es un sistema de software invisible para el usuario final, compuesto por un lenguaje de definición de datos, un lenguaje de manipulación y de consulta, que puede trabajar a distintos niveles.

Entre sus funciones se encuentran la de permitir a los usuarios de negocio almacenar la información, modificar datos y acceder a los activos de conocimiento de la empresa. El gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes.

A su vez, el SGBD puede entenderse como una colección de datos relacionados entre sí, estructurados y organizados dentro del ecosistema conformado por ese conjunto de programas que acceden a ellos y facilitan su gestión.



Frente al anterior sistema de gestión de datos y archivos, que hay que recordar que se trata de un conjunto de programas que definen y trabajan sus propios datos; el acceso a los datos es independiente de los programas que los gestionan. Esta característica, en la práctica, supone una ventaja bastante importante de cara a tratar grandes volúmenes de información.

¿Qué permiten los gestores de base de datos?

Básicamente, podría simplificarse diciendo que el gestor de base de datos controla cualquier operación ejecutada por el usuario contra la base de datos. Para poder desarrollar esta función, es habitual que se necesiten emplear herramientas específicas, como sistemas de búsqueda y de generación de informes, además de distintas aplicaciones.

Lista de cotejo para evaluar mapa conceptual actividad 6.

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Lista de Cotejo. Primera actividad		
Indicadores	Si	No
Elabora el mapa conceptual del tema expuesto		
Participa y aporta en la lluvia		
Se mantiene el respeto del derecho de palabra.		
Hace aportes a la discusión.		
Claridad en el lenguaje utilizado.		
Estimula la participación y da retroalimentación.		
Se apega al contenido al responder.		
Contextualiza el contenido del texto.		
Total		



ANEXO N° 13

Instrucciones: Elabora un mapa conceptual

TEMA: CONCEPTOS BÁSICOS SOBRE BASES DE DATOS

¿Qué es una base de datos?

Una base de datos es una herramienta para recopilar y organizar información. Las bases de datos pueden almacenar información sobre personas, productos, pedidos u otras cosas. Muchas bases de datos comienzan como una lista en una hoja de cálculo o en un programa de procesamiento de texto. A medida que la lista aumenta su tamaño, empiezan a aparecer redundancias e inconsistencias en los datos. Cada vez es más difícil comprender los datos en forma de lista y los métodos de búsqueda o extracción de subconjuntos de datos para revisión son limitados. Una vez que estos problemas comienzan a aparecer, una buena idea es transferir los datos a una base de datos creada con un sistema de administración de bases de datos (DBMS), como Access.

Una base de datos computarizada es un contenedor de objetos. Una base de datos puede contener más de una tabla. Por ejemplo, una base de datos de Access almacena sus tablas en un solo archivo, junto con otros objetos como formularios, informes, macros y módulos. Las bases de datos creadas en el formato Access 2007 (que también usan Access 2016, Access 2013 y Access 2010) tienen la extensión de archivo .accdb y las bases de datos creadas en formatos anteriores de Access tienen la extensión de archivo .mdb. Puede usar Access 2016, Access 2013, Access 2010 o Access 2007 para crear archivos en formatos de archivo anteriores (por ejemplo, Access 2000 y Access 2002-2003).

Con Access, puede: Agregar nuevos datos a una base de datos, como un nuevo artículo en un inventario, Modificar datos existentes en la base de datos, por ejemplo, cambiar la ubicación actual de un artículo, Eliminar información, por ejemplo, si un artículo se vende o se descarta, Organizar y ver los datos de diferentes formas, Compartir los datos con otras personas mediante informes, correo electrónico, intranet o Internet.

Partes de una base de datos de Access

Las secciones siguientes son breves descripciones de las partes de una base de datos de Access típica.

Tablas: Una tabla de base de datos es similar en apariencia a una hoja de cálculo en cuanto a que los datos se almacenan en filas y columnas. Por ende, es bastante fácil importar una hoja de cálculo en una tabla de base de datos. La principal diferencia entre almacenar los datos en una hoja de cálculo y almacenarlos en una base de datos es la forma en la que están organizados los datos.



Para aprovechar al máximo la flexibilidad de una base de datos, los datos deben organizarse en tablas para que no se produzcan redundancias. Por ejemplo, si quiere almacenar información sobre los empleados, cada empleado debe especificarse solo una vez en la tabla que está configurada para los datos de los empleados. Los datos sobre los productos se almacenarán en su propia tabla y los datos sobre las sucursales se almacenarán en otra tabla. Este proceso se denomina normalización.

Cada fila de una tabla se denomina registro. En los registros se almacena información. Cada registro está formado por uno o varios campos. Los campos equivalen a las columnas de la tabla. Por ejemplo, puede tener una tabla llamada "Empleados" donde cada registro (fila) contiene información sobre un empleado distinto y cada campo (columna) contiene otro tipo de información como nombre, apellido, dirección, etc. Los campos deben designarse como un determinado tipo de datos, ya sea texto, fecha u hora, número o algún otro tipo.

Otra forma de describir los registros y los campos es imaginar un catálogo de tarjetas antiguo de una biblioteca. Cada tarjeta del archivador corresponde a un registro de la base de datos. Cada dato de una tarjeta individual (autor, título, etc.) equivale a un campo de la base de datos.

Formularios: formularios permiten crear una interfaz de usuario en la que puede escribir y modificar datos. Los formularios a menudo contienen botones de comandos y otros controles que realizan distintas tareas. Puede crear una base de datos sin usar formularios con tan solo modificar los datos en las hojas de datos de la tabla. Sin embargo, la mayoría de los usuarios de bases de datos prefieren usar formularios para ver, escribir y modificar datos en las tablas.

Puede programar botones de comandos para determinar qué datos aparecen en el formulario, abrir otros formularios o informes, o ejecutar otras tareas. Por ejemplo, puede tener un formulario llamado "Formulario de cliente" en el que trabaja con los datos de los clientes.

El formulario de cliente puede tener un botón que abra un formulario de pedido en el que puede especificar un pedido nuevo para el cliente.

Los formularios también le permiten controlar de qué manera otros usuarios interactúan con los datos de la base de datos. Por ejemplo, puede crear un formulario que muestre solo determinados campos y permita que se realicen únicamente ciertas operaciones. Esto ayuda a proteger los datos y a asegurarse de que los datos se especifican correctamente.

Informes: Los informes se usan para dar formato a los datos, resumirlos y presentarlos. Por lo general, un informe responde a una pregunta específica como: "¿Cuánto dinero recibimos de cada cliente este año?" o "¿En qué ciudades residen nuestros clientes?". A cada informe se le puede dar formato para presentar la información de la manera más legible posible.

Se puede ejecutar un informe por vez y siempre se reflejan los datos actuales de la base de datos. Generalmente, se les da formato a los informes para imprimirlos, pero también pueden verse en pantalla, exportarse a otro programa o enviarse como datos adjuntos en un correo electrónico.



Consultas: Las consultas pueden realizar diversas funciones en una base de datos. La función más común es recuperar datos específicos de las tablas. Los datos que quiere ver generalmente están distribuidos en varias tablas y las consultas le permiten verlos en una única hoja de datos. Además, debido a que muchas veces no quiere ver todos los registros a la vez, las consultas le permiten agregar criterios para "filtrar" los datos y obtener solo los registros que quiere.

Ciertas consultas son "actualizables", es decir, puede modificar los datos de las tablas subyacentes mediante la hoja de datos de la consulta. Si está trabajando en una consulta actualizable, recuerde que los cambios se realizan en realidad en las tablas, no solo en la hoja de datos de la consulta.

Hay dos variedades básicas de consultas: consultas de selección y consultas de acciones. Una consulta de selección simplemente recupera los datos y los pone a disposición para su uso. Puede ver los resultados de la consulta en la pantalla, imprimirlos o copiarlos al portapapeles. O bien, puede usar el resultado de la consulta como un origen de registro para un formulario o un informe.

Una consulta de acción, tal como el nombre lo indica, realiza una tarea con los datos. Las consultas de acción se pueden usar para crear tablas nuevas, agregar datos a las tablas existentes, o actualizar o eliminar datos.

Macros: Las macros en Access pueden considerarse un lenguaje de programación simplificado que puede usar para agregar funciones a la base de datos. Por ejemplo, puede adjuntar una macro a un botón de comando en un formulario para que la macro se ejecute cada vez que se hace clic en ese botón. Las macros contienen acciones que ejecutan tareas, como abrir un informe, ejecutar una consulta o cerrar la base de datos. La mayoría de las operaciones de la base de datos que realiza manualmente se pueden automatizar mediante el uso de macros, por lo que se convierten en dispositivos que permiten ahorrar mucho tiempo.

Módulos: Los módulos, como las macros, son objetos que puede usar para agregar funciones a la base de datos. Mientras que las macros se crean en Access mediante la elección de una lista de acciones de macro, los módulos se escriben en el lenguaje de programación Visual Basic para Aplicaciones (VBA). Un módulo es una colección de declaraciones, instrucciones y procedimientos que se almacenan juntos como una unidad. Un módulo puede ser un módulo de clase o un módulo estándar. Los módulos de clase se adjuntan a formularios o informes y, por lo general, contienen procedimientos que son específicos para el formulario o el informe al que están adjuntos. Los módulos estándar contienen procedimientos generales que no están asociados a ningún otro objeto. Los módulos estándar aparecen en Módulos en el panel de navegación mientras que los módulos de clase no.



Lista de cotejo para evaluar mapa conceptual actividad 7.

NOMBRE DEL ALUMNO	GRADO Y GRUPO
	Sub Módulo: I Diseña bases de datos ofimáticas..
MAESTRO:	FECHA:

Lista de Cotejo. Primera actividad		
Indicadores	Si	No
Elabora el mapa conceptual del tema expuesto		
Participa y aporta en la lluvia		
Se mantiene el respeto del derecho de palabra.		
Hace aportes a la discusión.		
Claridad en el lenguaje utilizado.		
Estimula la participación y da retroalimentación.		
Se apega al contenido al responder.		
Contextualiza el contenido del texto.		
Total		



BIBLIOGRAFIA

Silberschatz, Korth y Sudarshan, *Fundamentos de bases de datos 5ª edición* Universidad de Yale, 2006. <https://docer.com.ar/doc/vv010>. También disponible en versión impresa y CD-ROM.

Pares, Castillas, Costal, Ginesta, Martín y Pérez. *Bases de Datos*, Universitat Oberta de Catalunya, 2005. https://www.academia.edu/36084231/Rafael_Camps_Par%C3%A9 Software_libre_U_Formaci%C3%B3n_d_e_Posgrado_Bases_de_datos_71Z799014MO

Chiquito Alor, *Diseñando una Base de Datos*, Centro de Estudios Tecnológicos industrial y de servicios N° 79, 2017. <https://view.genial.ly/59af7e2f072c1e1c904cf41c/interactive-content-diseno-de-bd>

Microsoft Soporte, "Conceptos básicos de base de datos", 2021. <https://support.microsoft.com/es-es/office/conceptos-b%C3%A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204>

Burgués Illa, *Diseño lógico de bases de datos*, Universitat Oberta de Catalunya, 2016. <http://uoc.gitlab.io/2016/DiseBBDD/3.pdf>

Conesa y Rodríguez, *Diseño físico de bases de datos*, , Universitat Oberta de Catalunya, s/f. http://openaccess.uoc.edu/webapps/o2/bitstream/10609/77645/5/Bases%20de%20datos%3B%20conceptos%20b%C3%A1sicos%2C%20dise%C3%B1o%20f%C3%ADsico%20y%20rendimiento_M%C3%B3dulo%201_Dise%C3%B1o%20f%C3%ADsico%20de%20bases%20de%20datos.pdf