

**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**

**CENTRO UNIVERSITARIO UAEM NEZAHUALCÓYOTL**

**MANUAL PARA PRÁCTICAS DE LA UNIDAD DE APRENDIZAJE  
INGENIERÍA DE SOFTWARE**

**“FASE DISEÑO”**



**ELABORARÓN:  
DRA. CARMEN LILIANA RODRÍGUEZ PÁEZ  
DR. RICARDO RICO MOLINA  
DRA. DORICELA GUTIERREZ CRUZ**

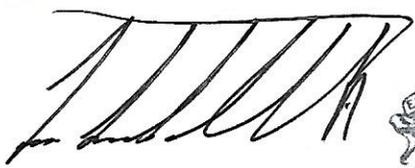
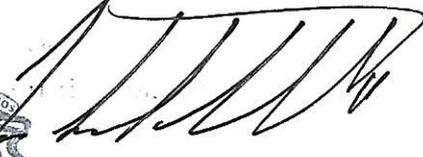
**JULIO 2019**

# MANUAL PARA PRÁCTICAS DE LA UNIDAD DE APRENDIZAJE INGENIERÍA DEL SOFTWARE

## IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

<b>ESPACIO ACADÉMICO:</b> Centro Universitario Nezahualcóyotl							
<b>PROGRAMA EDUCATIVO:</b> LICENCIATURA EN INGENIERÍA EN SISTEMAS INTELIGENTES					<b>Área de docencia:</b> Sustantiva		
Aprobación por los H.H. Consejos Académico y de Gobierno				<b>Fecha:</b> Agosto 2019		Programa elaborado por: Dra. Carmen Liliana Rodríguez Páez	
<b>Nombre de la Unidad de Aprendizaje:</b> INGENIERÍA DEL SOFTWARE					<b>Fecha de elaboración:</b> Agosto 2019		
<b>Clave</b> L40633	<b>Horas de teoría</b> 2	<b>Horas de práctica</b> 1	<b>Total, de horas</b> 3	<b>Créditos</b> 5	<b>Tipo de Unidad de Aprendizaje</b> Curso	<b>Carácter de la Unidad de Aprendizaje</b> Obligatoria	<b>Núcleo de formación</b> Sustantivo
<b>Prerrequisitos</b> Fundamentos de base de datos		<b>Unidad de Aprendizaje Antecedente</b> Ninguna				<b>Unidad de Aprendizaje Consecuente</b> Ninguna	
<b>Programa en los que se imparte:</b> LICENCIATURA EN INGENIERÍA EN SISTEMAS INTELIGENTES							

**EL PRESENTE MANUAL DE PRÁCTICAS HA SIDO AVALADO EN EL MES DE AGOSTO DE 2019 POR:**

  Centro Universitario UAEM Nezahualcóyotl	
 M. EN C. JOSÉ A. CASTILLO JIMÉNEZ SECRETARIO H. CONSEJO DE GOBIERNO H. CONSEJO DE GOBIERNO CENTRO UNIVERSITARIO NEZAHUALCÓYOTL	 M. EN C. JOSÉ A. CASTILLO JIMÉNEZ SECRETARIO H. CONSEJO ACADÉMICO H. CONSEJO ACADÉMICO CENTRO UNIVERSITARIO NEZAHUALCÓYOTL

## ÍNDICE

Directorio UAEM	5
Directorio del Centro UAEM Nezahualcóyotl	6
Ubicación de la asignatura Ingeniería del Software, dentro del programa de la Lic. en Ing. en Sistemas Inteligentes.	7
Secuencia Didáctica	8
Presentación	9
<b>PRÁCTICA 1</b>	<b>12</b>
<b>DISEÑO DEL DICCIONARIO DE DATOS</b>	
Objetivo	12
Introducción	12
Desarrollo	13
Conclusión	15
Bibliografía	15
<b>PRÁCTICA 2</b>	<b>16</b>
<b>DEFINICIÓN DE LA ARQUITECTURA DEL SISTEMA</b>	
Objetivo	16
Introducción	16
Desarrollo	18
Conclusión	19
Bibliografía	19
<b>PRÁCTICA 3</b>	
<b>DISEÑO DE ENTRADAS DEL SISTEMA</b>	<b>20</b>
Objetivo	20
Introducción	20
Desarrollo	22
Conclusión	23
Bibliografía	23
<b>PRÁCTICA 4</b>	<b>24</b>
<b>DISEÑO DE SALIDAS DEL SISTEMA</b>	
Objetivo	24
Introducción	24
Desarrollo	27
Conclusión	28
Bibliografía	28
<b>PRÁCTICA 5</b>	
<b>DIAGRAMA WARNIER-ORR (JEAN DOMINIQUE WARNIER Y KEN)</b>	<b>29</b>
Objetivo	29
Introducción	29
Desarrollo	32
Conclusión	32
Bibliografía	32

<b>PRÁCTICA 6</b>	<b>33</b>
<b>DIAGRAMA HIPO</b>	
Objetivo	33
Introducción	33
Desarrollo	37
Conclusión	37
Bibliografía	37
<b>PRÁCTICA 7</b>	
<b>DIAGRAMA DE FLUJO DE DATOS</b>	<b>38</b>
Objetivo	38
Introducción	38
Desarrollo	42
Conclusión	42
Bibliografía	42
<b>PRÁCTICA 8</b>	<b>43</b>
<b>DISEÑO DEL MODELO CONCEPTUAL CHEN UTILIZANDO UNA HERRAMIENTA CASE</b>	
Objetivo	43
Introducción	43
Desarrollo	46
Conclusión	46
Bibliografía	46
<b>PRÁCTICA 9</b>	
<b>CREACIÓN DEL MODELO ENTIDAD RELACIÓN (ER)</b>	<b>47</b>
Objetivo	47
Introducción	47
Desarrollo	49
Conclusión	49
Bibliografía	49
<b>PRÁCTICA 10</b>	<b>50</b>
<b>GENERACIÓN DE ESPECIFICACIONES DE DESARROLLO DEL SISTEMA DE INFORMACIÓN</b>	
Objetivo	51
Introducción	52
Desarrollo	53
Conclusión	54
Bibliografía	55

# UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

## DIRECTORIO INSTITUCIONAL

Dr. en Edu.

**Alfredo Barrera Baca**

**RECTOR**

M. en E.U. y R.

**Marco Antonio Luma Pichardo**

Secretario de Docencia

M. en C

**Jannet Valero Vilchis**

Secretaria de Rectoría

Dr. en C.I.

**Carlos Eduardo Barrera Díaz**

Secretario de Investigación y Estudios Avanzados

Dr. en A.

**José Édgar Miranda Ortiz**

Secretario de Difusión Cultural

Dra. en Ed.

**Sandra Chávez Marín**

Secretaria de Extensión y Vinculación

M. en Dis.

**Juan Miguel Reyes Viurquez**

Secretario de Administración

M. en L.A.

**María del Pilar Ampudia García**

Secretaria de Cooperación Internacional

Dr. en C.S.

**Luis Raúl Ortiz Ramírez**

Abogado General

Lic. en Com.

**Gastón Pedraza Muñoz**

Director General de Comunicación  
Universitaria

M. en D.F.

Jorge Rogelio Zenteno Domínguez

Encargado del Despacho de la Contraloría  
Universitaria

M. en A.

**José Francisco Mejía Carbajal**

Secretario Particular Adjunto del Rector

M. en E.

**Javier González Martínez**

Secretario de Finanzas

Dr. en C.C.

**José Raymundo Marcial Romero**

Secretario de Planeación y Desarrollo Institucional

Dra. en Dis.

**Mónica Marina Mondragón**

Secretaría de Cultura Física y Deporte

M. en R. I.

**Jorge Bernáldez García**

Secretario Técnico de la Rectoría

M. en A. P.

**Guadalupe Ofelia Santamaría González**

Directora General de Centros Universitarios y  
Unidades Académicas Profesionales

Lic. En Act.

**Angelita Garduño Gómez**

Secretaria particular del Rector

# CENTRO UNIVERSITARIO UAEM NEZAHUALCÓYOTL

## DIRECTORIO

Maestro en Derecho <b>Juan Carlos Medina Huicochea</b>	<b>ENCARGADO DEL DESPACHO DE LA DIRECCIÓN</b>
Maestro en Ciencias <b>José Antonio Castillo Jiménez</b>	Subdirector Académico
Licenciado en Economía <b>Ramón Vital Hernández</b>	Subdirector Administrativo
Doctora en Ciencias Sociales <b>María Luisa Quintero Soto</b>	Coordinadora de Investigación y Estudios Avanzados
Licenciado en Administración de Empresas <b>Víctor Manuel Durán López</b>	Coordinador de Planeación y Desarrollo Institucional
Maestro en Ciencias <b>Cesar Lucio Gutiérrez Ruiz</b>	Coordinador de la Licenciatura en Comercio Internacional
Maestro en S.F. <b>Carlos Anaya Hernández</b>	Coordinadora de la Licenciatura en Educación para la Salud
Doctor en Ingeniería de los Sistemas <b>Ricardo Rico Molina</b>	Coordinador de la Licenciatura en Ingeniería en Sistemas Inteligentes
Maestro en Ciencias <b>Ricardo Pacheco Ruiz</b>	Coordinador de la Licenciatura en Ingeniería en Transporte
Maestro en Ciencias de la Computación <b>Erick Nicolás Cabrera Álvarez</b>	Coordinador de la Licenciatura en Seguridad Ciudadana Mixta
Maestro en Administración <b>José Ramon CS. Garcia Ibarra</b>	Coordinador de la Licenciatura en Seguridad Ciudadana Presencial

Ubicación de la asignatura de Ingeniería del Software, dentro del programa de la Lic. en Ingeniería en Sistemas Inteligentes (Figura 1)

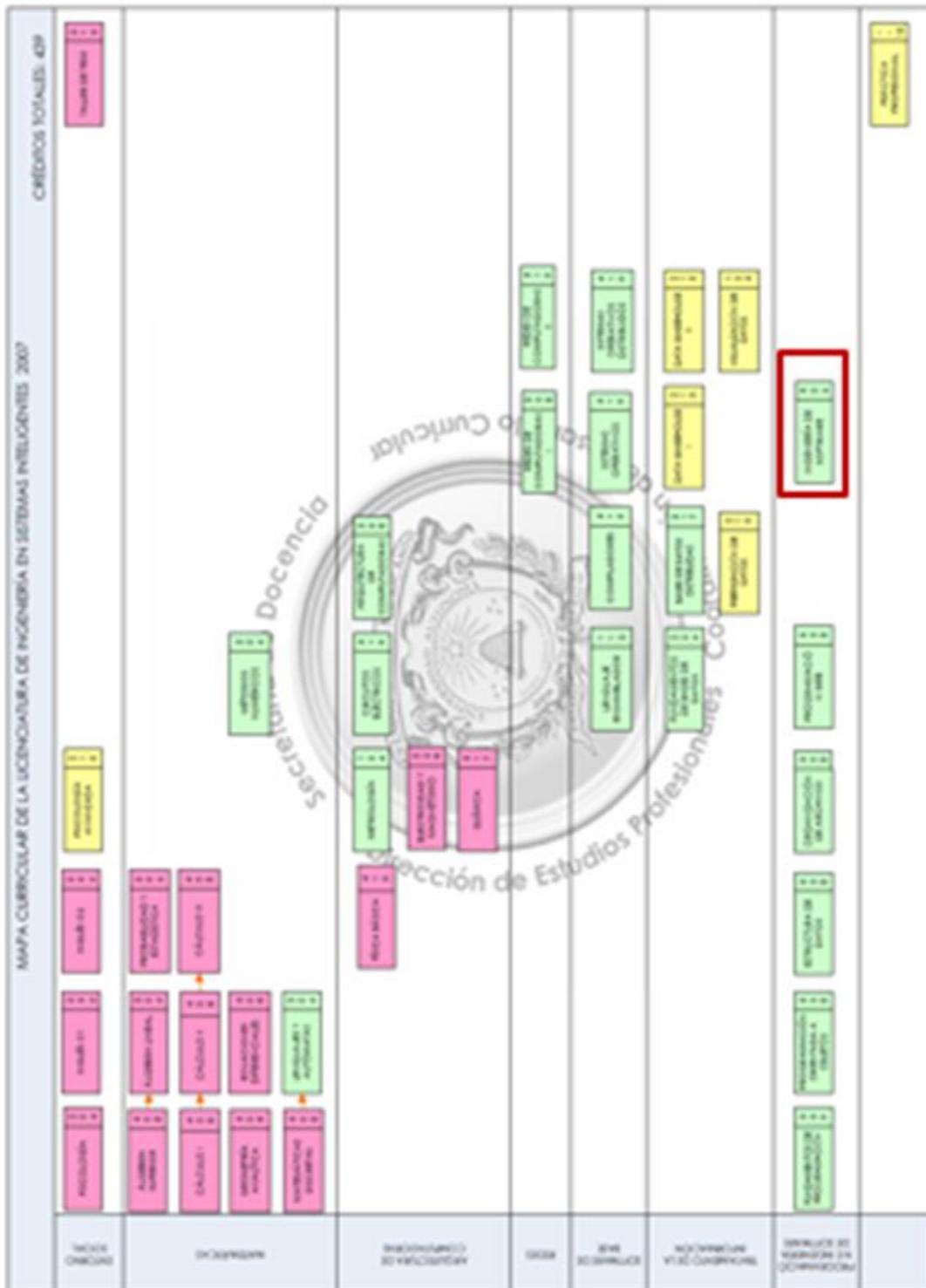


Figura 1. Mapa Curricular ISI-Ing. de Software

## SECUENCIA DIDÀCTICA

**PRÁCTICA 1. DISEÑO DEL DICCIONARIO DE DATOS**

**PRÁCTICA 2. DEFINICIÓN DE LA ARQUITECTURA DEL SISTEMA**

**PRÁCTICA 3. DISEÑO DE ENTRADAS DEL SISTEMA**

**PRÁCTICA 4. DISEÑO DE SALIDAS DEL SISTEMA**

**PRÁCTICA 5. DIAGRAMA WARNIER-ORR (JEAN DOMINIQUE WARNIER Y KEN)**

**PRÁCTICA 6. DIAGRAMA HIPO**

**PRÁCTICA 7. DIAGRAMA DE FLUJO DE DATOS**

**PRÁCTICA 8. DISEÑO DEL MODELO CONCEPTUAL CHEN**

**PRÁCTICA 9. CREACIÓN DEL MODELO RELACIONAL**

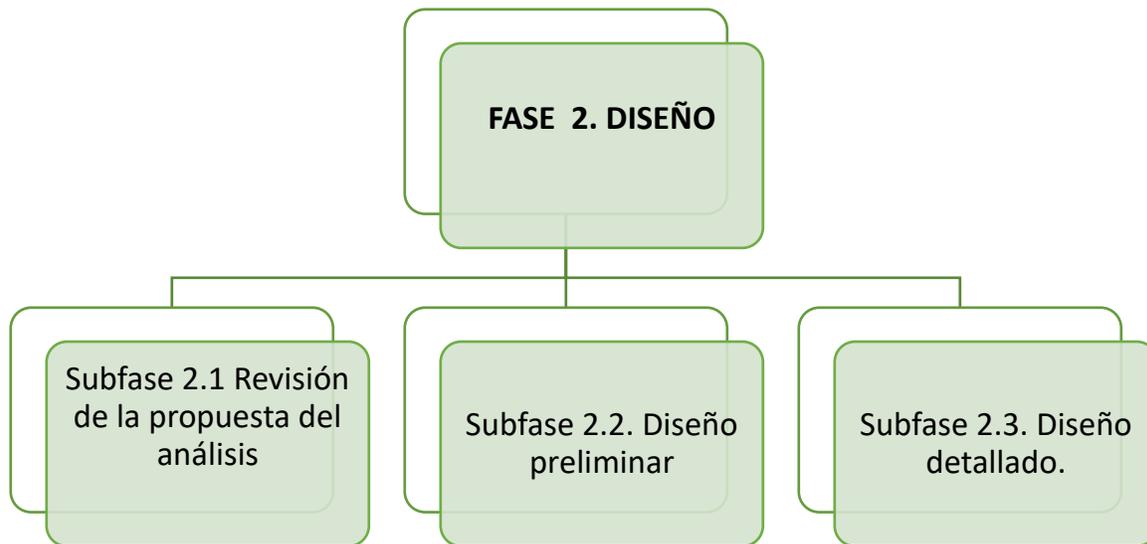
**PRÁCTICA 10. GENERACIÓN DE ESPECIFICACIONES DE DESARROLLO DEL SISTEMA DE INFORMACIÓN**

## PRESENTACIÓN

El objetivo de este Manual de prácticas de la unidad de aprendizaje Ingeniería de Software es servir como guía para la elaboración de la segunda fase “Diseño” del proyecto final del sistema de información para los alumnos del quinto semestre del programa de Licenciatura de Ingeniería en Sistemas Inteligentes, siendo parte fundamental de las prácticas de dicha unidad, en el cual se organiza y coordina cada una de las subfases y sus correspondientes actividades a realizar. Así mismo, se abordará desde una visión general de los principios, características y métodos de diseño del software siendo importante para tener definida una correcta y adecuada arquitectura del sistema, partiendo de las características generales de los principales estilos arquitecturales.

Para este manual se plantea seguir con la Metodología para desarrollar Sistemas de Información Basados en Computadoras (LGS, 2007). La cual consta de cinco fases como son: análisis, diseño, construcción, implementación y operación y mantenimiento.

En la fase de Diseño del Sistema de Información es la etapa donde se plantea una solución, es decir, es una traducción de los requerimientos en formas que los satisfagan. Cuando un diseñador elabora un «diseño» escribe las especificaciones detalladas del nuevo sistema, esto es, se describen sus características: salidas/entradas/controles/. Para la realización de esta se fase se tendrán en cuenta tres subfases (Figura 2): Subfase 2.1 Revisión de la propuesta del análisis, Subfase 2.2. Diseño preliminar y Subfase 2.3. Diseño detallado.



**Figura 2.** Fase de diseño (Elaboración propia, 2019)

Con este manual, se pretende conseguir, además: del nivel didáctico: Servir como información básica que guíe a los alumnos en el proceso de análisis de un sistema de información. A nivel organizativo: Mejorar la organización y coordinación entre las personas involucradas en el desarrollo de proyectos informáticos (alumnos y profesores).

Recordando que durante esta fase del proceso se debe tener especial cuidado, pues de ella depende la calidad y aceptación de los resultados finales. Tenga en cuenta que "un diseño bien realizado y muy completo, representa un 30% de la solución ya obtenida".

# PRÁCTICA 1

## DISEÑO DEL DICCIONARIO DE DATOS

### OBJETIVO

Comprender que es un diccionario de datos y aprender a elaborarlo.

### INTRODUCCIÓN

Una de las herramientas usadas en software son las del análisis estructurado que permiten al ingeniero de software crear modelos de datos, de flujo y de comportamiento en una forma que permite la consistencia y continuidad con facilidad para hacer la revisión, edición y ampliación. Los modelos creados con estas herramientas dan al ingeniero de software la perspectiva de la representación del análisis y lo ayudan a eliminar errores antes de que éstos se propaguen al diseño o, lo que sería peor, a la implementación (Pressman, 2010).

Los diccionarios de datos son herramientas importantes e indispensables en los diagramas de flujo ya que dan precisión sobre los datos que se manejan en el sistema. Evitando así malas interpretaciones o ambigüedades. En ese mismo enfoque se tiene que el diccionario contiene "datos acerca de los datos" es decir, definiciones de otros objetos del sistema, en lugar de simples "datos en bruto" (Date,2001). Por lo anterior, una razón importante para mantener un diccionario de datos es guardar datos ordenados. Esto significa que los datos deben ser consistentes. Si usted guarda datos acerca del sexo de un hombre como "M" en un registro, "Masculino" en un segundo registro y como el número "1" en un tercer registro, los datos no son consistentes. Un diccionario de datos ayudará en este aspecto.

Por lo tanto, un diccionario de datos construido por un conjunto rigurosos de procedimientos que deberá ayudar a cumplir los siguientes objetivos:

- Establecer un glosario de términos.
- Proveer una terminología estándar.
- Definir todos los términos de almacenamiento asociados con un sistema.
- Identificar estructuras de datos disponibles a todos los procesos.
- Proveer una posibilidad de referencias cruzadas.
- Resolver problemas asociados con “ALIAS” (o sinónimos) o acrónimos.
- Proveer un control centralizado de datos, para, por ejemplo, facilitar posibles cambios en el sistema.
- Proveer una guía de referencia para evaluación de diseño.
- Ayudar a minimizar costos de mantenimiento, al tener un repositorio con la información de todas las estructuras de datos.
- Ayudar en la generación de un catálogo completo de datos de pruebas

Las definiciones contenidas en el diccionario de datos se denominan entradas que pueden ser de dos tipos:

- Simples: se definen indicando el tipo de datos que representa o enumerando los posibles valores.
- Compuestas: SE definen indicando elementos que la componen y la forma en la que están combinados.

Pueden estar formadas por entradas simples y/o entradas compuestas.

Un ejemplo de la forma como se representan las entradas compuestas es:

TIPO	FÓRMULA
CONCATENACIÓN	Entrada = componente1+componente2
DISYUNCIÓN	Entrada=[Componente1   Componente2]
OPCIONALIDAD	Entrada=(Componente1)
REPETICIÓN	Entrada={Componente1}

- Concatenación es una entrada está compuesta por dos o más componentes,
- Disyunción una entrada está compuesta por uno de entre varios componentes posibles.
- Opcionalidad supone que un determinado componente puede estar o no en una entrada compuesta.
- Repetición es una entrada, está compuesta por dos o más componentes del mismo tipo.

### Descripción de las Estructuras de Datos

- Normalmente las estructuras de datos se describen usando una notación algebraica. Este método permite al analista producir una vista de los elementos que constituyen la estructura de datos junto con información referente a dichos elementos.
- Ejemplo: el analista indicara si hay muchos elementos iguales en la estructura de datos (un grupo de repetición) o si dos elementos pudieran excluirse mutuamente.

### La notación algebraica usa los siguientes símbolos:

1. Un signo igual (=) significa “este compuesto de”.
2. Un signo de suma(+) significa “y”.
3. Las llaves {} indican elementos repetitivos, también llamados grupos de repetición o tablas.
4. Los corchetes [ ] representan una situación de uno u otro. Se podría representar un elemento u otro, pero no ambos.
5. Los paréntesis ( ) representan un elemento opcional.

En realidad, debe ser posible consultar el diccionario del mismo modo que cualquier otra base de datos, de manera que, por ejemplo, sea posible saber qué programas o usuarios se podrían ver afectados por un cambio propuesto al sistema, la relación que hay entre las bases de datos y los elementos de información a los que se hace referencia (Elmasri, 2005).

La siguiente figura presenta una visión rica de los principales elementos de un diccionario de datos.



**Figura 3.** Diccionario de datos (Garcia, 2003)

Ejemplo para el desarrollo de un diccionario de datos básico:

### Clientes

Columna	Tipo	Nulo	Predeterminado
Tel	int(12)	No	
RFC (Primaria)	varchar(13)	No	
Dom	varchar(60)	No	
Ciudad	text	No	
Nom_clien	text	No	
Codpos	int(5)	No	

### Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	RFC	0	A	No	

## DESARROLLO

En esta actividad, se deberá revisar la información que se presentó en la introducción y complementar con el material de clase, y deberán de realizar el diccionario de datos del diagrama de flujo del proceso escogido en su proyecto y deberán explicar cada paso.

## CONCLUSIONES

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica.

---



---



---



---



---

## BIBLIOGRAFÍA

Date, 2001. Introducción a los sistemas de bases de datos. C. J. Date. 7ª. Edición. Editor Pearson Educación, 2001. ISBN 9684444192, 9789684444195 N.º de páginas 936 páginas.

Elmasri, 2005. Fundamentos de sistemas de bases de datos, R. A. Elmasri, S. B. Navathe. 3ª. Edición. Editor Pearson Educación, 2005. ISBN 8478290516.

García, S., Morales, E. Análisis y Diseño detallado de aplicaciones informáticas de gestión. Paraninfo. 2003.

Pressman, Richard, 2010. INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO Séptima edición. Published by McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020. Copyright © 2010 by The McGraw-Hill Companies, Inc. All rights reserved. 978-0-07-337597-7.

Reporte técnico: Metodología para el desarrollo de un sistema de información maestría en Ciencias en Ingeniería de Sistemas, SEPI, ESIME, IPN, Mexico, 2007, Galindo L.

## PRÁCTICA 2

### DEFINICIÓN DE LA ARQUITECTURA DEL SISTEMA

#### OBJETIVO

Diseñar la arquitectura del sistema es decir de los módulos que permitirán iniciar la fase de construcción.

#### INTRODUCCIÓN

La Arquitectura del Sistema permite ver de una manera gráfica y “de un solo golpe”, la forma como se relacionarán los elementos del Nuevo Sistema de Información; tiene la ventaja, luego entonces, de poder ser entendida sin problemas lo mismo por el experto que por el usuario. Se refiere a “las estructuras [módulos] de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. La cual debe incluir entre otros elementos (Galindo, 2007):

- Definición de niveles de arquitectura
- Identificación de requisitos de diseño y construcción
- Especificación de excepciones
- Especificación de estándares y normas de diseño y construcción
- Identificación de subsistemas de diseño
- Especificación del entorno tecnológico
- Especificación de requisitos de operación y seguridad.

Oro término de *arquitectura* se utiliza mucho sin su connotación tradicional de compilación y existen muchas definiciones de "arquitectura" (en el contexto de sistemas) y "arquitectura del sistema (o sistemas)", por ejemplo:

"Systems architecture: the fundamental and unifying system structure defined in terms of system elements, interfaces, processes, constraints, and behaviors."  
(Arquitectura de sistemas: la estructura del sistema fundamental y de

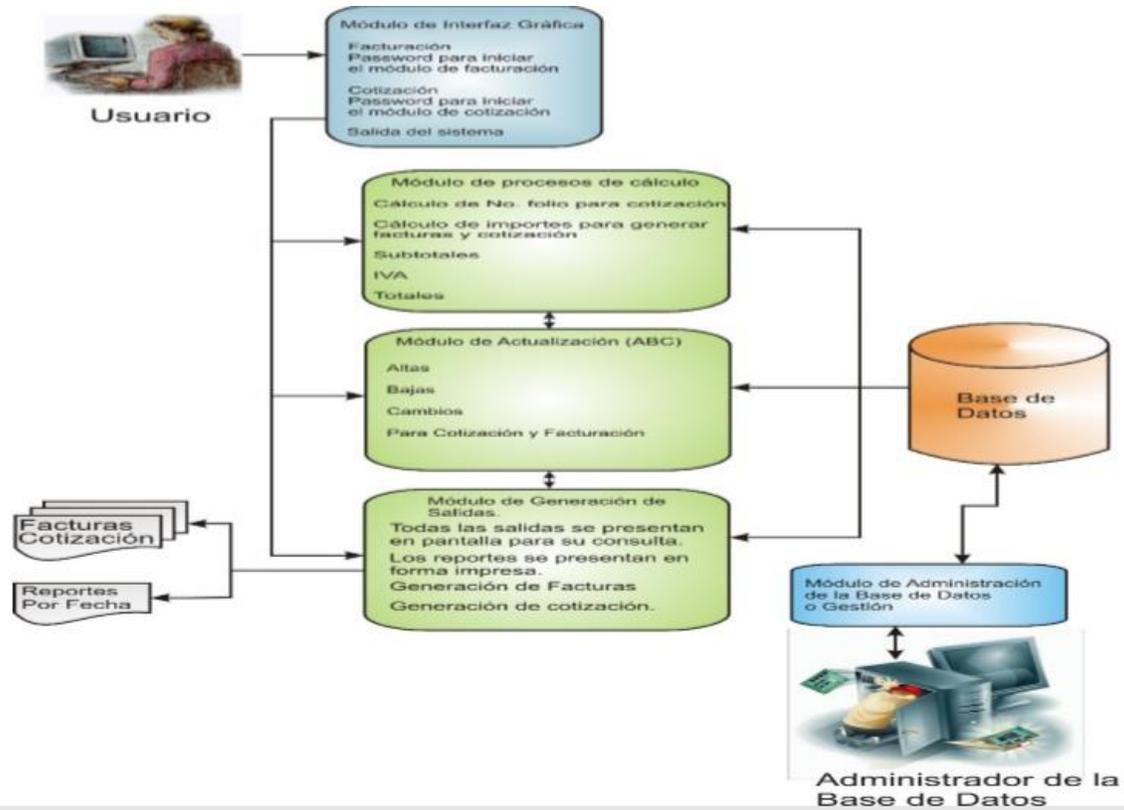
unificación definida en términos de comportamientos, restricciones, procesos, interfaces y elementos del sistema) (Elmasri & Addison, 2007).

### **Definición de arquetipos**

Un arquetipo es una clase o un patrón que representa una abstracción fundamental de importancia crítica para el diseño de una arquitectura para el sistema objetivo. En general, se requiere de un conjunto relativamente pequeño de arquetipos a fin de diseñar sistemas incluso algo complejos. La arquitectura del sistema objetivo está compuesta de estos arquetipos, que representan elementos estables de la arquitectura, pero que son implementadas en muchos modos diferentes con base en el comportamiento del sistema.

Ejemplo:

La Figura 4.0 muestra la Arquitectura Básica de los módulos del Sistema Propuesto: La cual está compuesta por los módulos del sistema, diseño de comunicación entre módulos y la revisión de la interfaz de usuario.



Otro ejemplo se visualiza el modulo de actualización, modulo de calculos, modulo de generación de salidas y le módulo de interfaz gráfica del usuario.

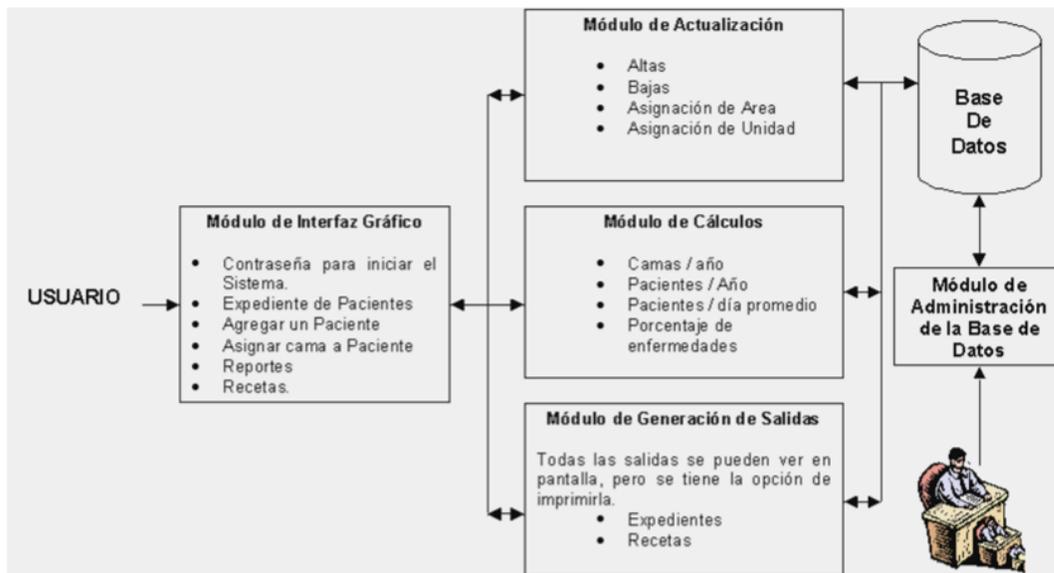


Figura 4. Ejemplo arquitectura del sistema (Elaboración propia, 2019)

## DESARROLLO

Realice el diseño de la arquitectura del sistema de su proyecto. En base a la información dada en clase y complementado con lo expuesto en la introducción.

## CONCLUSIONES

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

---

---

---

---

---

## BIBLIOGRAFÍA

“Fundamentals of Database Systems 5th Ed.”, Elmasri & Navathe, Addison Wesley, 2007. (cap. 2)

Reporte técnico: Metodología para el desarrollo de un sistema de información maestría en Ciencias en Ingeniería de Sistemas, SEPI, ESIME, IPN, Mexico, 2007, Galindo L.

## PRÁCTICA 3

### DISEÑO DE ENTRADAS DEL SISTEMA

#### OBJETIVO

Elaborar el diseño de entrada que tendrá el sistema.

#### INTRODUCCIÓN

El diseño de las Entradas es la liga que une, en primera instancia, al Sistema de Información con el mundo de los usuarios. Algunos aspectos del diseño de las entradas variarán dependiendo de si, el sistema estará orientado a operar por lotes (batch) o en línea (textual) o por interfaz gráfica de usuario. La importancia de las entradas se desprende de una regla básica en la Informática: “Si entra basura al sistema, saldrá basura”. En consecuencia, es crucial el realizar un diseño adecuado de éstas (Galindo, 2007).

Objetivo del Diseño de Entrada: Consiste en el desarrollo de especificaciones y procedimientos para la preparación de datos, la realización de los procesos necesarios para poner los datos de transacción en una forma utilizable para su procesamiento así como la entrada de los datos se logra al instruir a la computadora para que lea ya sea documentos escritos, impresos ó por personas que los escriben directamente al sistema (<https://www.monografias.com/trabajos7/resi/resi.shtml#enra>).

Existen cinco objetivos que controlan la cantidad de entrada requerida, a evitar los retrasos, controlar los errores y mantener la sencillez de los pasos necesarios, estos son:

- Control de la Calidad de Entrada
- Evitar los Retrasos
- Evitar los errores en los datos
- Evitar los pasos adicionales
- Mantener la Sencillez del Proceso

### Control de la Calidad de Entrada:

Existen varias razones por las cuales un buen diseñador debe controlar la cantidad de datos en la entrada:

- Las Operaciones de preparación y entrada dependen de las personas dado que los costos de mano de obra son altos y la preparación de ingreso de los datos también lo son.
- La fase de entrada puede ser un proceso lento que toma mucho más tiempo que el que necesitan las computadoras para realizar sus tareas.

Otra de las finalidades de hacer estas entradas es para mantener la sencillez del Proceso:

El sistema mejor diseñado se ajusta a las personas que lo utilizarán y al mismo tiempo proporcionarán métodos para el control de los errores, la simplicidad funciona y es aceptada por cualquier usuario. Cuesta trabajo que los usuarios acepten sistemas complejos o confusos y que no exista ninguna garantía para el éxito al instalar un sistema complejo y que domine.

La importancia de las entradas se desprende de una regla básica en la Informática: “Si entra basura al sistema, saldrá basura”. En consecuencia, es crucial el realizar un diseño adecuado de éstas. La figura 5 muestran un ejemplo del diseño de entradas de un sistema de información en base a los requerimientos mencionados en párrafo anterior (Kendall & Kendall, 2006).

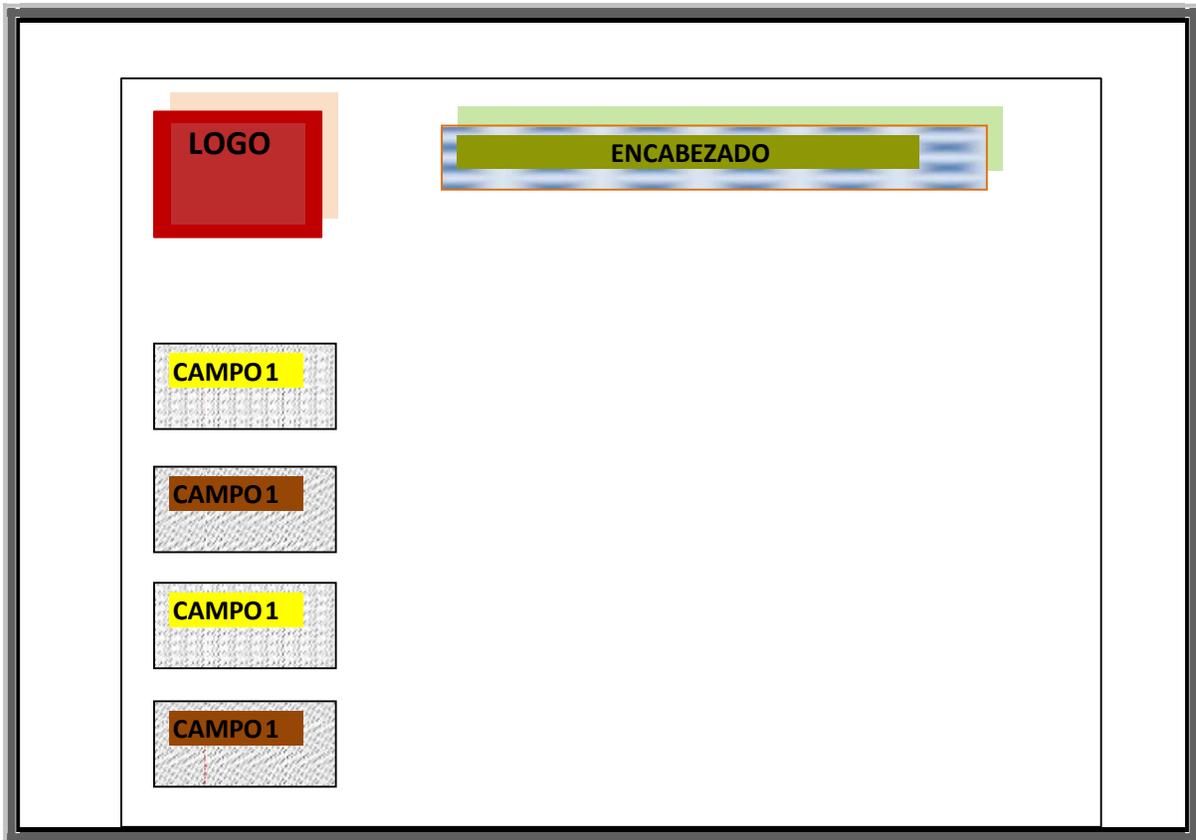


Figura 5. Ejemplo de entrada del sistema

## DESARROLLO

En esta fase se elaboran entradas que deberá tener el sistema, en base lo indicado en la introducción y lo complementan con lo visto en clase.

## CONCLUSIONES

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica:

---



---



---



---

## BIBLIOGRAFÍA

Kendall & Kendall, Entradas efectivas, sesion 12, Ana Mercedes Cáceres ,Año 2006

James A. Senn, Análisis y Diseño de Sistemas de Información, Segunda Edición, Mc Graw Hill, consultado el 17 de julio de 2019 en <http://www.monografias.com>.

Reporte técnico: Metodología para el desarrollo de un sistema de información maestría en Ciencias en Ingeniería de Sistemas, SEPI, ESIME, IPN, México, 2007, Galindo L.

## PRÁCTICA 4

### DISEÑO DE SALIDAS DEL SISTEMA

#### OBJETIVO

Diseñar las salidas del sistema propuesto.

#### INTRODUCCIÓN

El término "salida" se aplica a cualquier información producida por un sistema, ya sea impresa, desplegada o verbal. Cuando los analistas diseñan la salida, seleccionan métodos para representar la información y crean documentos, informes u otros formatos que contienen información producida por el sistema (<https://www.monografias.com/trabajos7/resi/resi.shtml#enra>).

Los métodos de salida varían a lo largo de los sistemas. Para algunos, como un informe de inventarios de la cantidad de mercancía, el sistema del computador, bajo el control del programa, nada más consulta los datos que se tienen a mano en el almacenamiento, y los ensambla en una forma que sea presentable. Otra salida puede requerir procesamiento sustancial, antes de que esté disponible para utilizarlo.

Los analistas deben decidir cuándo imprimir, desplegar o presentar su salida en forma audible. La salida impresa puede utilizar papel en blanco o formas preimpresas, la salida visual puede utilizar una o múltiples pantallas para desplegar información. El diseño de las salidas del sistema es importante debido a que, para los usuarios, en la primera característica de un Sistema de Información en que se fija, son las salidas, ya que si no es de calidad puede pensar, que todo es sistema no es necesario, bueno o adecuado y, por tanto, evite su utilización o incluso ocasione errores, generando que el sistema falle.

Y el concepto Salida se aplica a cualquier tipo de información producida por un Sistema Informático o computacional, y puede ser: impreso, desplegado en la pantalla, verbal, multimedia, etc.

El objetivo principal durante el diseño de salida de la computadora es la información que será presentada a las personas, puede afirmarse que la salida de la computadora es para las personas, es por esto que no se aborda la forma en que los datos se mueven entre los procesos o entre los almacenamientos de datos.

### ***Tipos de salida***

La salida del sistema puede ser:

- un reporte
- un documento
- un mensaje

De acuerdo con las circunstancias y los contenidos, la salida puede ser impresa o presentada en una pantalla, el contenido de la pantalla tiene su origen en las siguientes fuentes:

- Recuperación de un Dispositivo de Almacenamiento.
- Transmisión desde un Proceso o Actividad del Sistema.
- Directamente desde una Fuente de Entrada.

Por otra parte, Galindo (2007) en su reporte señala que, para obtener un diseño de salida, es necesario realizar y responde cinco preguntas que le indican a los ingenieros de software tener claro la salida que se está diseñando, por ejemplo:

1. ¿Quién recibirá la salida?
2. ¿Cuál es su uso planeado?
3. ¿Cuánto detalle se necesita?
4. ¿Cuándo y con qué frecuencia se necesitan las salidas?
5. ¿Por cuales medios?

## Lineamientos para el diseño de reportes impresos

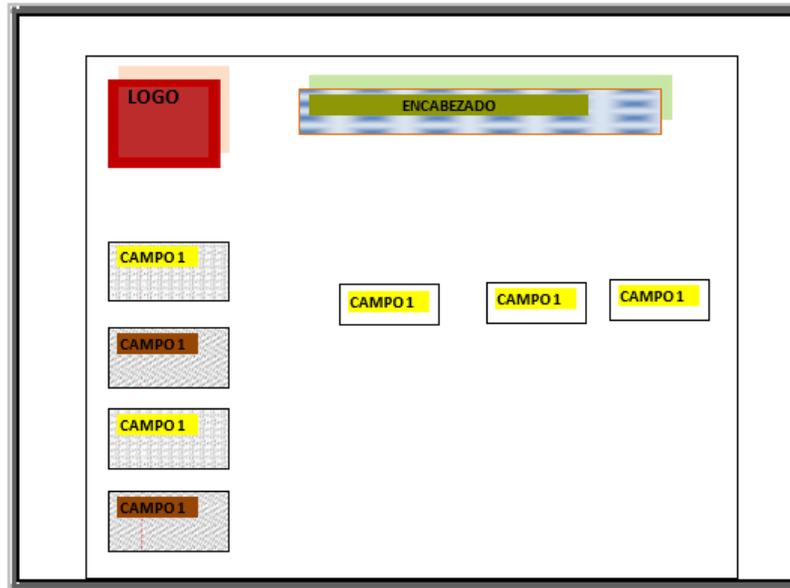
Convenciones para el diseño de reporte: Cómo indicar el tipo de dato, cómo mostrar el tamaño exacto y cómo presentar la manera de indicar continuaciones en las formas de diseño consecutivas. Calidad de papel, tipo y tamaño: Para hacer que la forma diseñada sea clara y estéticamente agradable. Formas de salida especiales: Por ejemplo: Envío de documentos de retorno a los clientes.

Consideraciones de diseño: Para que se proporcione al usuario la información necesaria en un formato legible. Atributos funcionales: Encabezado o título, número de página, fecha de preparación, etc. Atributos estilísticos/estéticos: Los reportes deben estar organizados siguiendo la forma en que los ojos ven. Utilización de márgenes, espaciado, codificación por colores, logotipos organizacionales, etc. (Kendall & Kendall, 2006).

### La salida de pantalla difiere de la salida impresa.

- ✓ La de pantalla es efímera.
- ✓ Está más enfocado hacia el usuario.
- ✓ Está disponible en momentos más flexibles
- ✓ No es portátil en la misma forma.
- ✓ Y a veces no puede ser cambiado por medio de interacción directa.

Ahora, se anexa un ejemplo de diseño de salida en pantalla e impreso.



The wireframe shows a form titled "MEMBRETE LABORATORIO" with the following sections:

MEXICO, DF. A \_\_\_\_\_ DE \_\_\_\_\_ DE 2001

**CLIENTE:** \_\_\_\_\_  
**DOMICILIO FISCAL:** \_\_\_\_\_

**CIUDAD:** \_\_\_\_\_ **COL:** \_\_\_\_\_  
**RFC:** \_\_\_\_\_ **TEL:** \_\_\_\_\_

**Código de Identificación:** \_\_\_\_\_

CANTIDAD	CONCEPTO	PRECIO	IMPORTE

**IMPORTE CON LETRA:** \_\_\_\_\_

**EMPRESA:** \_\_\_\_\_  
**DIRECCIÓN DE LA EMPRESA:** \_\_\_\_\_  
**ATENCIÓN:** \_\_\_\_\_  
**FECHA DE RECEPCIÓN:** \_\_\_\_\_  
**FECHA DE ENTREGA:** \_\_\_\_\_  
**CARACTERÍSTICAS DE LA MUESTRA:** \_\_\_\_\_

**NORMAS APLICADAS:** \_\_\_\_\_

**EQUIPO UTILIZADO:** \_\_\_\_\_

**TEMPERATURA:** \_\_\_\_\_  
**HUMEDAD RELATIVA:** \_\_\_\_\_

PROHIBIDA LA REPRODUCCIÓN PARCIAL O TOTAL. LOS RESULTADOS QUE SE DAN SON DE CARÁCTER CONFIDENCIAL.

Figura 6. Ejemplo de salida del sistema en pantalla e impresa

## DESARROLLO

Elaborar el diseño de las salidas que tendrá el sistema de acuerdo con las respuestas obtenidas en cada una de las cinco preguntas. Recordando que la salida en su mayoría es impresa o mostrada en pantalla no omite incluir: títulos y encabezados, líneas de detalle y valores de datos, acumulados o totales o resúmenes, entre otros aspectos que considere el cliente.

## CONCLUSIONES

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica:

---

---

---

---

## BIBLIOGRAFÍA

James A. Senn, Análisis y Diseño de Sistemas de Información, Segunda Edición, Mc Graw Hill, consultado el 17 de julio de 2019 en <http://www.monografias.com>.

Reporte técnico: Metodología para el desarrollo de un sistema de información maestría en Ciencias en Ingeniería de Sistemas, SEPI, ESIME, IPN, Mexico, 2007, Galindo L.

Kendall & Kendall , Salidas efectivas, sesion 11, Ana Mercedes Cáceres ,Año 2006.

## **PRÁCTICA 5**

### **DIAGRAMA WARNIER-ORR (JEAN DOMINIQUE WARNIER Y KEN)**

#### **OBJETIVO**

Comprender el uso y forma de aplicación del diagrama Warnier-Orr, para aplicarlo a su proyecto final.

#### **INTRODUCCIÓN**

Este diagrama permite la descripción de los procedimientos y los datos de la organización y fueron desarrollados inicialmente en Francia, por Jean-Dominique Warnier y Kenneth Orr. Estos diagramas permiten diseñar un programa identificando primero su salida, para trabajar hacia atrás y poder determinar las combinaciones de pasos y entradas para producirlas. La ventaja de estos diagramas consiste en su apariencia simple y sencilla de entender. En ellos se agrupan los procesos y datos de un nivel a otro. Los sencillos métodos gráficos usados en los diagramas de Warnier/Orr hacen evidentes los niveles en un sistema y más claros los movimientos de los datos en dichos niveles.

#### **Consideraciones sobre el proceso de diseño**

- 1) Evaluar las características de la estructura de datos.
- 2) Representar los datos en términos de formas elementales tales como secuencia, selección, y repetición.
- 3) Transformar la representación de la estructura de datos en una jerarquía de control para el software.
- 4) Refinar la jerarquía del software utilizando los criterios definidos como parte de un método.
- 5) Finalmente, desarrollar la descripción procedimental del software.

## Elementos Básicos

Los diagramas de Warnier/Orr muestran los procesos y la secuencia en que se realizan. Cada proceso se define de una manera jerárquica; es decir, consta de conjuntos de subprocesos que lo definen, en cada nivel, el proceso se muestra en una llave que agrupa a sus componentes. Puesto que un proceso puede tener muchos subprocesos distintos, un diagrama de Warnier/Orr usa un conjunto de llaves para mostrar cada nivel del sistema (Barranco, 2001).

### Uso de diagramas de warnier/orr

La capacidad de mostrar la relación entre procesos y pasos de un proceso no es exclusiva de los diagramas de Warnier/Orr, así como tampoco lo es el uso de la iteración, selección de alternativas o el tratamiento de casos individuales. Tanto los diagramas de flujo estructurado y los métodos del español estructurado logran eso también. Sin embargo, el enfoque que se usa para desarrollar las definiciones de un sistema por medio de estos diagramas es distinto y se adapta y se adaptan bien a los que se usan en el diseño de sistemas lógicos.

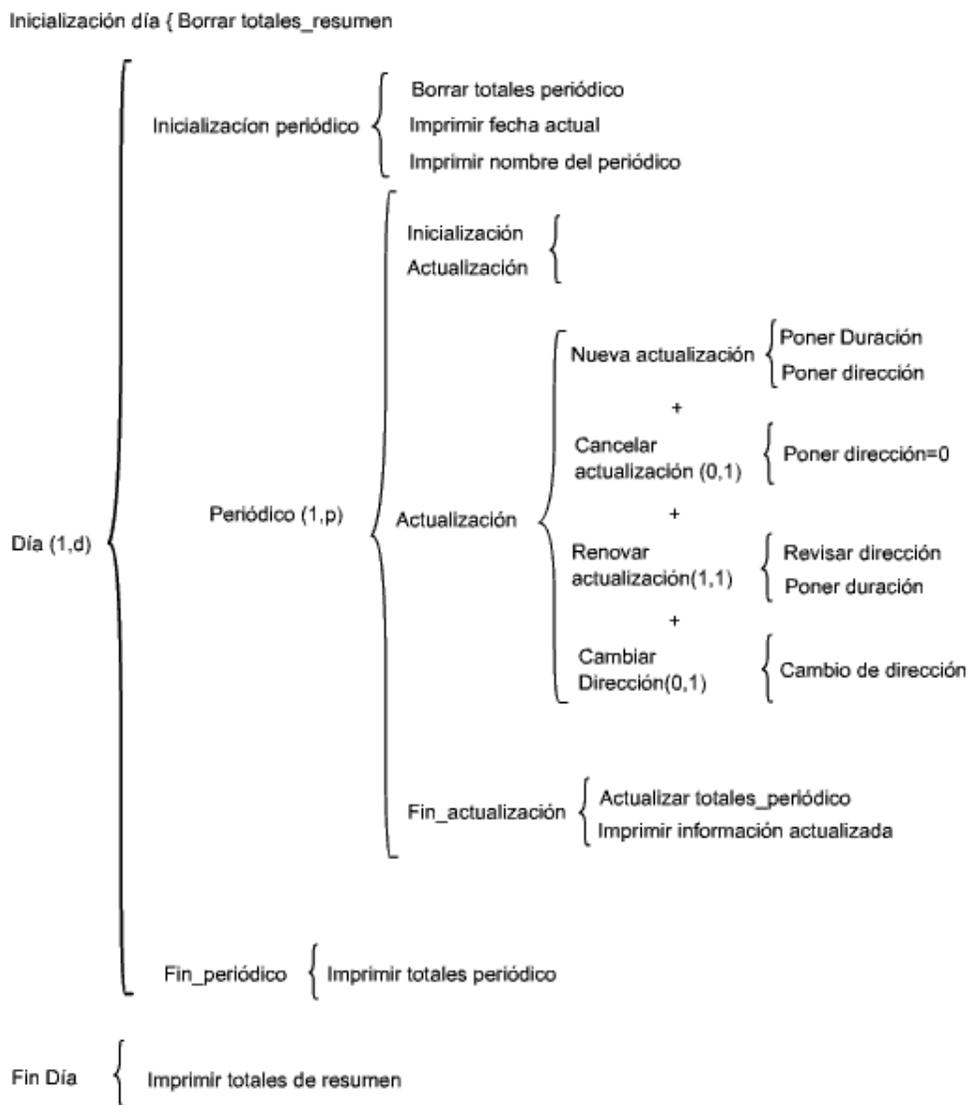
Para desarrollar un diagrama de Warnier/Orr, el analista trabaja hacia atrás, empezando con la salida del sistema y usando un análisis orientado hacia la salida. En el papel el desarrollo se mueve de izquierda a derecha. En primer lugar, se definen la salida o resultados esperados del procedimiento. En el nivel siguiente, mostrado mediante la inclusión por medio de una llave, se definen los pasos necesarios para producir la salida. A su vez, cada paso se define un poco más. Las llaves adicionales agrupan los procesos requeridos para producir el resultado en el siguiente nivel.

Los diagramas de Warnier/Orr ofrecen a los expertos en sistemas algunas ventajas distintivas. Son simples en apariencia y fáciles de entender. Aun así, son poderosas herramientas de diseño. Tienen la ventaja de mostrar agrupaciones de procesos y los

datos que deben transferirse de nivel a nivel. Además, la secuencia del trabajo hacia atrás garantiza que el sistema estará orientado hacia el resultado (Coolieng, 2013)

### Símbolos del diagrama warnier

En este tipo de diagramación (Figura 7) no es tan visual como los demás diagramas utilizados, pues las llaves son los únicos símbolos que se utilizan, más otras anotaciones



### Figura 7. Ejemplo Diagrama Warnier-Orr

#### DESARROLLO

Desarrollar una descripción teórica y realizar el diagrama Warnier-Orr de su proyecto siguiendo los siguientes pasos:

1. Definir la salida o resultado esperado del procedimiento.
2. Detallar el nivel siguiente mostrado por la llave se muestran los pasos para producir la salida.
3. Agrupar entre llaves los procesos para el siguiente nivel.

#### CONCLUSIONES

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica:

---

---

---

---

#### BIBLIOGRAFÍA

J. E. Cooling (2013). Software Design for Real-time Systems consultado en <https://books.google.com.mx/books?id=1Vb0BwAAQBAJ> el 17 de julio de 2019.

Jesús Barranco de Areba (2001). Metodología del análisis estructurado de sistemas. Edición ilustrada. Editor Univ Pontifica Comillas, ISBN 8484680436, 9788484680437 N.º de páginas 540 páginas.

## PRÁCTICA 6 DIAGRAMA HIPO

### OBJETIVO

Diseñar la tabla visual HIPO de su proyecto.

### INTRODUCCIÓN

(En inglés, Hierarchy-Input-Process-Output) fueron desarrollados por IBM como esquemas de representación para un desarrollo jerárquico de arriba a abajo y como una ayuda de documentación para productos comercializados. Un conjunto de diagramas HIPO contiene una tabla visual de contenido, un conjunto de diagramas generales y un conjunto de diagramas de detalles (Kendall & Kendall, 2005).

La teoría en la que HIPO se basa en lo fácil de perder la pista de la función deseada de un sistema o componente de un sistema grande. La razón por lo cual es difícil es comparar los sistemas existentes contra sus especificaciones originales, por lo cual pueden ocurrir fallas incluso en los sistemas técnicamente bien formulados.

El usuario desde el punto de vista, la función puede extenderse a varios módulos, por lo tanto, el interés del analista es entender, describir y documentar los módulos y su interacción de forma que se obtenga el detalle suficiente, no perdiendo en panorama general.

El diagrama HIPO son representaciones graficas del sistema, en vez de prosa o Narrativa (Figura 8).

Características:

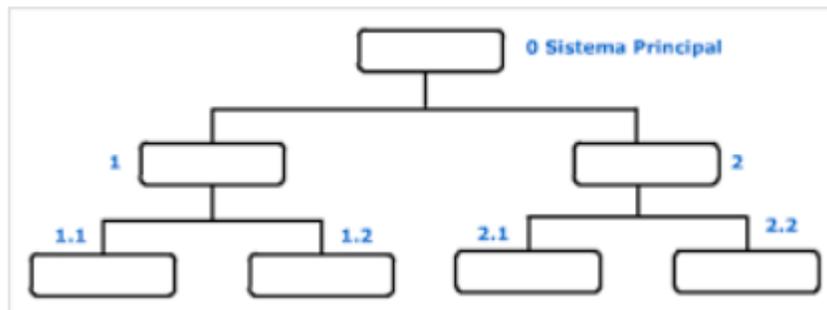
1. Utiliza una serie de diagrama y funciones de un sistema de producto para mostrar el insumo de la tecina HIPO.
2. Tabla de contenido visual, diagrama detallado y diagrama general, son tres

clases de diagrama HIPO.

3. El contenido visual del nivel superior de la tabla del diagrama HIPO.
4. Se describe las entradas del diagrama general, los procesos y las salidas de los componentes principales del sistema.
5. El diagrama detallado provee de la información necesaria para atender cuales son las entradas, procesos llevados a cabo y las salidas de un componente utilizable y eficiente.

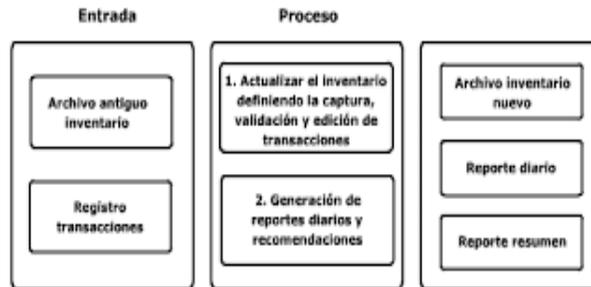
Por otra parte, estos diagramas se pueden representar de acuerdo a Yourdon (1993) en tres puntos:

1. Como una tabla visual de contenido es el directorio del conjunto de diagramas en el paquete; consta de un directorio con estructura de árbol (o de gráfica), un resumen de los contenidos de cada diagrama general, y una explicación de los símbolos utilizados.

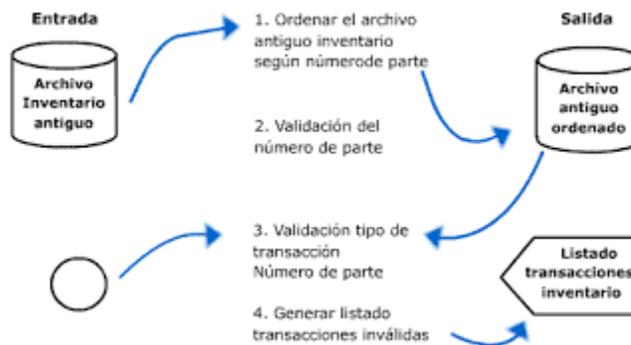


0.	_____
	Descripción _____
1.	_____
	Descripción _____
2.	_____
	Descripción _____
	.....

2. Los diagramas generales especifican los procesos de un sistema en forma funcional; cada diagrama describe las entradas, los pasos de proceso y las salidas para la función en cuestión; un diagrama general puede indicar la localización de los diagramas de detalles subordinados necesarios.



3. Los diagramas de detalle permiten crear para cada módulo la realización de un diagrama funcional. Por ejemplo, validar transacciones.



**Figura 8.** Ejemplo Diagrama HIPO

## DESARROLLO

Elaborar el diagrama HIPO del sistema, siguiendo las cinco características acá explicadas, también se puede apoyar en las ayudas de clase.

## **CONCLUSIONES**

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica:

---

---

---

---

## **BIBLIOGRAFÍA**

Kendall, Keneth E., Análisis y diseño de sistemas, México, Prentice Hall, 2005.

Yourdon E., Análisis estructurado moderno, Hispanoamericana, México, Prentice Hall, Prentice Hall, 1993

## **PRÁCTICA 7 DIAGRAMA DE FLUJO DE DATOS**

### **OBJETIVO**

Crear el diagrama de flujo de datos del proceso que van a apoyar con el sistema de información.

### **INTRODUCCIÓN**

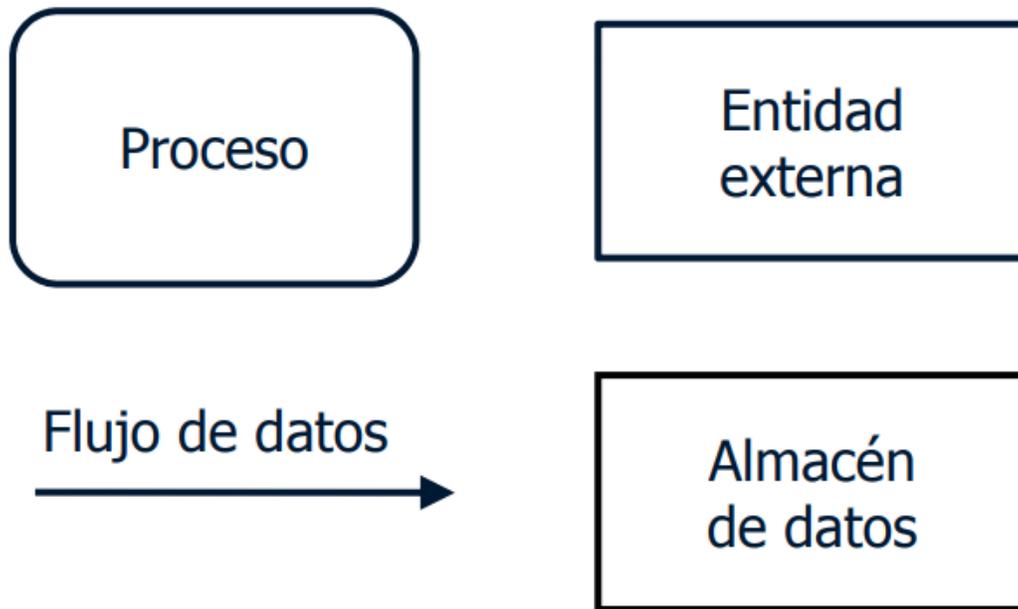
Los diagramas de flujo de datos (DFD) muestran los procesos de transformación de los datos en el sistema o proceso. En forma gráfica de un sistema que ilustra cómo fluyen los datos a través de distintos procesos (Córdoba, 2003).

Los (DFD) se realizan se realizan a distintos niveles de abstracción, detallando procesos concretos que aparecen como elementos simples en DFD de nivel superior. de nivel superior.

### **Elementos de un DFD:**

- Entidades externas, terminadores o elementos del entorno: Fuentes o sumideros de información. Diagramas de flujo de datos Emiten o reciben la información que fluye a través de las interfaces externas del sistema (las interfaces externas del sistema.
- Flujos de datos: Indican el flujo de información a través del sistema.
- Procesos o actividades: Transforman la información que les llega a través de Diagramas de flujo de datos los flujos de datos de entrada en la información que sale a través de los flujos de datos de salida.
- Almacenes de datos y ficheros: Lugares donde se guardan los datos para su procesamiento posterior.

Al ser una técnica grafica su fundamento es el uso de símbolos iconográficos, cuyo formato se basa en la notación de Gane&Sarson (Bello, 2000):

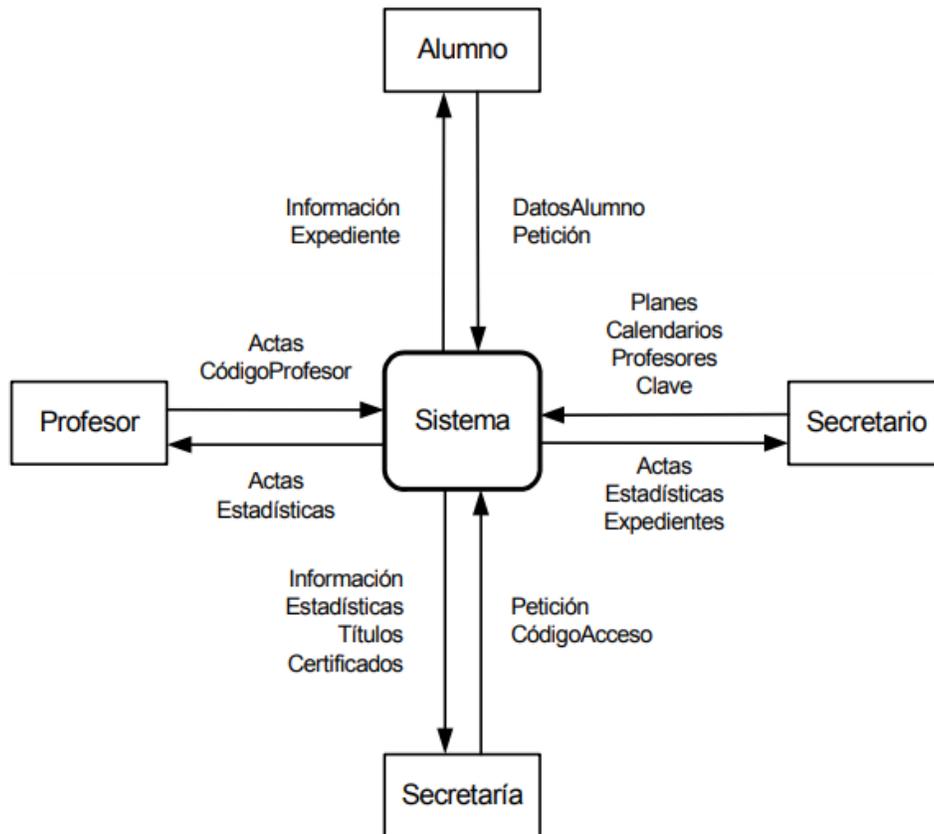


**Figura 9.** Notación para diagrama de flujos de procesos

#### Elaboración de DFD:

- Cada elemento tiene asociado un nombre unívoco a modo de etiqueta.
- Procesos y ficheros no pueden poseer sólo flujos de Diagramas de flujo de datos entrada (ni sólo de salida)
- Los flujos no pueden incluir información de control.
- Los flujos de datos pueden converger o divergir.
- Las entradas y salidas netas de un DFD deben coincidir con los flujos de entrada y salida del proceso al que corresponde en el nivel superior.
- Se desarrolla de lo general a lo particular se empieza por determinar las actividades más generales del área en estudio, posteriormente, se empieza a profundizar y recopilar más información específica y detallada, para ello se realiza preguntas específicas a un nivel de detalle cada vez mayor, elaborando diagramas más particulares o "finos".

En la siguiente figura, se presenta un diagrama de flujo de datos elemental:



**Figura 10.** Diagrama de flujo de datos contexto

Acercas del diagrama de contexto: Contiene un único proceso que representa al sistema completo, todos los terminadores (entidades externas que interactúan con el sistema) y todos los flujos de datos que conectan un sistema con su entorno.



**Figura 11.** DFD Nivel 0. del sistema actual (Elaboración propia, 2019)

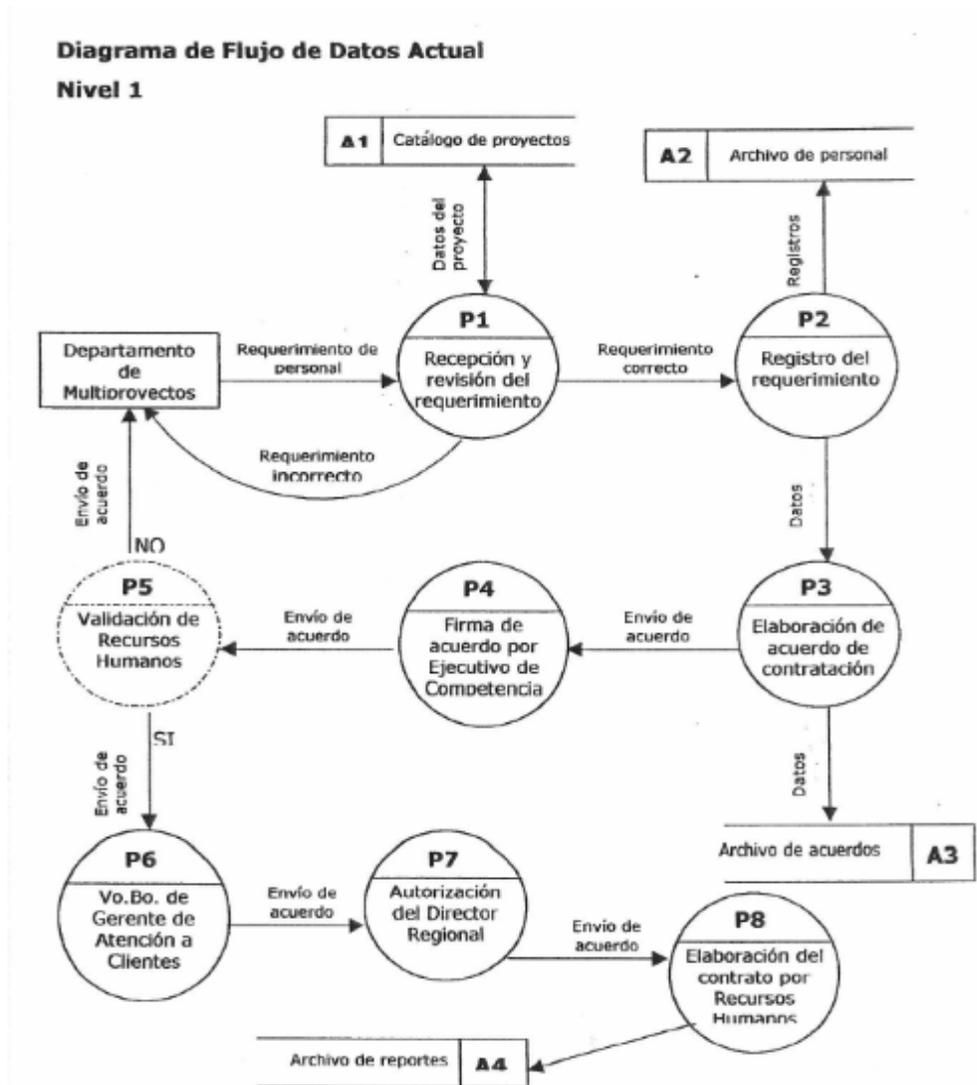


Figura 12. DFD Nivel 1 del sistema actual (Elaboración propia, 2019)

En este diagrama de nivel 1. Se desglosa cada uno de los procesos que representa al sistema completo, todos los terminadores (entidades externas que interactúan con el sistema) y todos los flujos de datos que conectan un sistema con su entorno.

## DESARROLLO

Favor en base a la literatura y a lo visto en clase desarrolle el diagrama de flujo de datos de nivel 0 y nivel 1 para los procesos del sistema actual.

## CONCLUSIONES

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica:

---

---

---

---

## BIBLIOGRAFÍA

Bello Gonzáles, Rabel. *Algoritmos y diagramas de flujo*. (5ta edición 2000)

Córdova Teodoro. *Diagrama de flujo de datos*. Disponible en: <http://www.monogramfias.com/trabajos-pdf/diagrama-de-flujo-shtml-18k-añoiii.no.31.junio.2003>

Ruiz Víctor. *Aprender a crear diagramas de flujo*. Disponible en <http://www.mis-algoritmos.com/2006/10/7/aprendaa-crear-diagramas-de-flujo/69k>.

## PRÁCTICA 8

### DISEÑO DEL MODELO CONCEPTUAL CHEN UTILIZANDO UNA HERRAMIENTA CASE

#### OBJETIVO

Realizar el diagrama del modelo conceptual de su proyecto final empleando la notación CHEN y las herramientas CASE.

#### INTRODUCCIÓN

El modelo conceptual más utilizado para el diseño de bases de datos fue introducido por Peter Chen en 1976. Es uno de los estilos de modelado más antiguos. Este modelo está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. La mayoría de los softwares de modelado actuales han eliminado esta notación de sus opciones tristemente (Elmasri y Navathe, 2002).

#### Simbología o nomenclatura básica según notación Chen:

Entidad: Es un objeto real o abstracto de interés, sobre el que se recoge información y se representa gráficamente mediante un rectángulo y su nombre aparece en el interior en mayúsculas. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual. Generalmente se expresa con sustantivos (De Miguel *et al.*, 1999).



Entidad

Relación Es una asociación, vinculación o correspondencia entre entidades. Se representa gráficamente con un rombo etiquetado. Generalmente representadas por verbos.



Relación

Una Relación queda caracterizada por tres propiedades:

- ✓ Nombre. Debe de tener un nombre que la identifique unívocamente.
- ✓ Grado. Número de tipos de entidad sobre las que se realiza la asociación. Generalmente binaria.
- ✓ Tipo de Correspondencia. Número máximo de ejemplares de cada tipo de entidad que pueden intervenir en un ejemplar del tipo de relación. A esta propiedad también se le denomina cardinalidad.

Atributo: Es una propiedad o característica asociada a una determinada entidad o relación y por lo tanto común a todos los ejemplares La representación gráfica utilizada es por medio de una elipse etiquetada en letra en minúsculas (Connolly *et al.*, 1998).

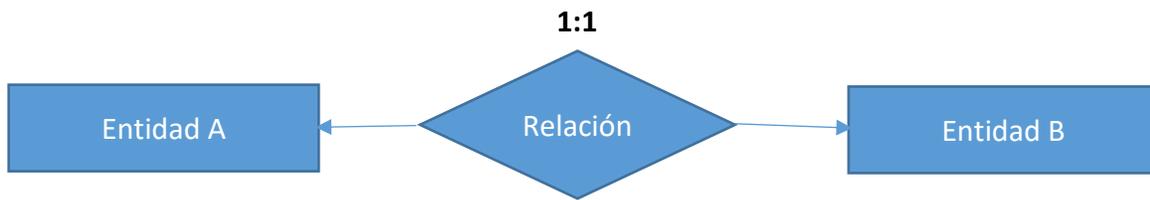


Tipos de atributos:

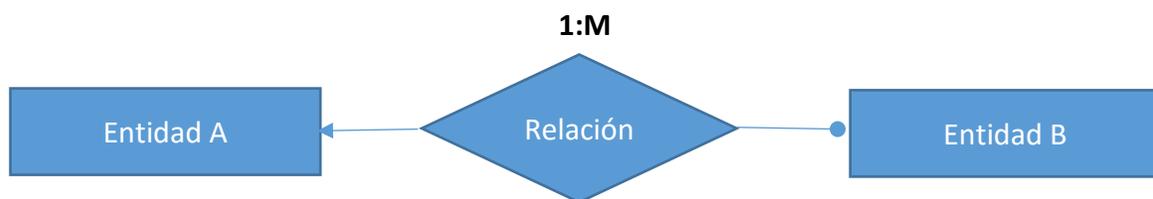
- ✓ Normal: Tiene un solo valor.
- ✓ Compuestos: Se dividen en otros atributos
- ✓ Multivalorados: Tiene un conjunto de valores para una entidad concreta. Se representa con doble elipse.
- ✓ Derivados: Cuando un valor puede calcularse u obtenerse a partir de otro. Se representa con una elipse con línea discontinua.
- ✓ Opcionales: Son usados cuando es posible desconocer el valor del atributo para cierta entidad o no se tiene un valor aplicable. Se representa con una línea discontinua entre la entidad y el atributo

Cardinalidad: Número de ejemplares de una entidad asociadas a otro ejemplar de una entidad o de la misma. Para una relación binaria (grado = 2), existen tres posibles tipos de correspondencia:

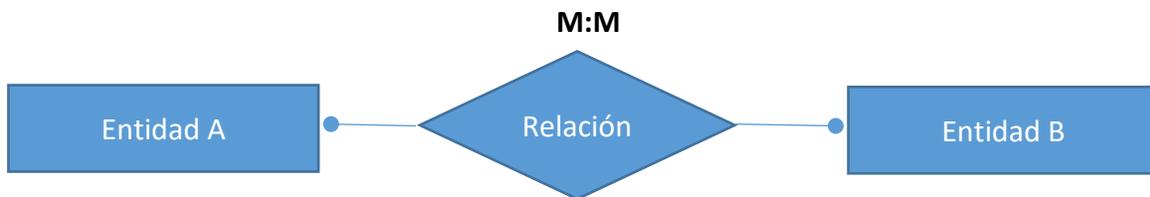
Cardinalidad uno a uno. Un ejemplar de la Entidad A se asocia con un ejemplar de una Entidad B y viceversa.



Cardinalidad uno a muchos. Un ejemplar de la Entidad A se puede asociar con muchos ejemplares de una Entidad B. y un ejemplar de la Entidad B se asocia con un solo ejemplar de la Entidad A.



Cardinalidad muchos a muchos. Un ejemplar de la Entidad A se puede asociar con muchos ejemplares de una Entidad B y viceversa.



Hay dos tipos de entidades: fuertes y débiles. Una entidad fuerte es una entidad que tiene existencia propia y tiene una clave primaria. Una entidad débil es una entidad que no tiene suficientes atributos para formar una clave primaria. Una entidad débil no puede existir sin su entidad fuerte. Se representa gráficamente por dos rectángulos concéntricos, la relación de asociación se representa con un doble rombo, su cardinalidad es 1:M.

Clave primaria. Es un atributo o conjunto de atributos que identifican en forma única a una entidad. Se representa subrayando el nombre del atributo.



Clave débil o discriminante. Es un atributo en una entidad débil que la identifica junto con la clave primaria de la entidad fuerte. Se representa subrayando en forma discontinua el atributo.



Clave candidata o alternativa. Es un atributo que puede ser clave primaria, pero no fue elegida como tal. Sin embargo, es importante considerarla.

Ejemplo de notación para atributos:

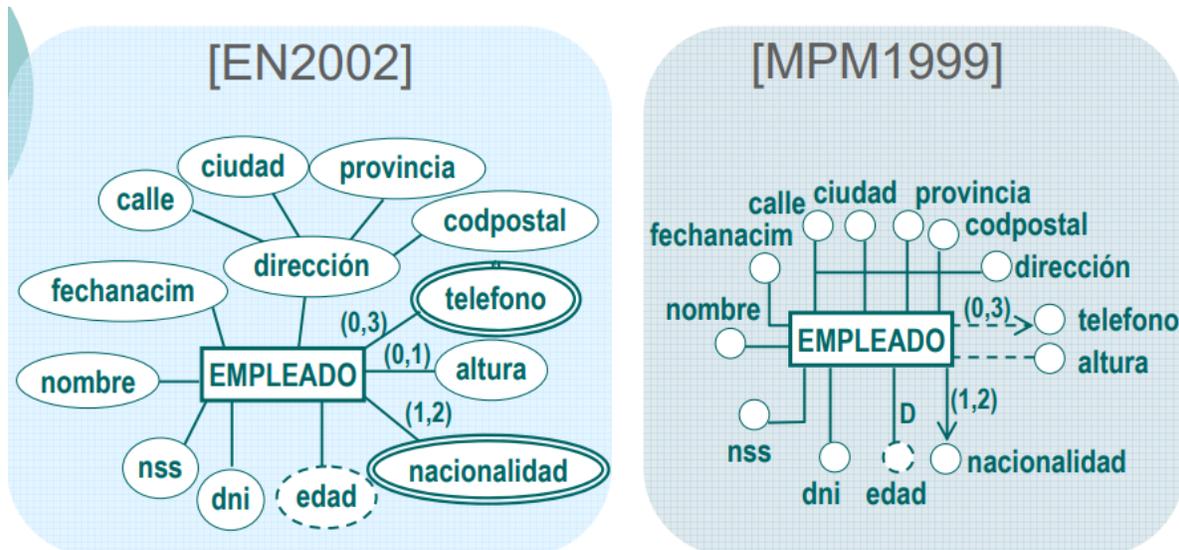


Figura 13. Ejemplo de notación de atributos

## DESARROLLO

En base a las indicaciones elabore su modelo conceptual, enfatizando en cada uno de los elementos del diagrama usando la herramienta CASE seleccionada, identificando en cada texto las entidades, atributos, relaciones y llaves primarias.

## **CONCLUSIONES**

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica:

---

---

---

---

## **BIBLIOGRAFÍA**

Elmasri, R.; Navathe, S.B. (2002). Fundamentos de Sistemas de Bases de Datos. 3ª ed. Addison-Wesley.

De Miguel, A.; Piattini, M.; Marcos, E. (1999). Diseño de bases de datos relacionales. Ra-Ma.

Connolly, T.; Begg C.; Strachan, A. (1998). Database Systems: A Practical Approach to Design, Implementation and Management. 2nd ed. Addison- Wesley.

## PRÁCTICA 9 CREACIÓN DE MODELOS RELACIONALES

### OBJETIVO

Crear el modelo de datos relacionales empleando herramientas CASE a partir de un diagrama ER.

### INTRODUCCIÓN

El modelo de datos relacional organiza y representa los datos en forma de tablas o relaciones (Date, 2001):

Una base de datos relacional es una colección de relaciones [tablas] es una colección de relaciones [tablas].

Representación lógica	Representación física	Modelo Relacional
Tabla	Archivo secuencial	Relación
Fila	Registro	Tupla
Columna	Campo	Atributo

El concepto de relación: Tuplas, atributos y dominios (Ramez *et al.*, 2007).

id_trabajador	nombre	tarifa_hr	tipo_de_oficio	id_supv
1235	F. Aguilera	12,50	Electricista	1311
1412	A. Calvo	13,75	Fontanero	1540
2920	N. Marín	10,00	Carpintero	null
3231	O. Pons	17,40	Albañil	null
1540	J.M. Medina	11,75	Fontanero	null
1311	J.C. Cubero	15,50	Electricista	null
3001	D. Sánchez	8,20	Albañil	3231

**Atributo (A<sub>i</sub>):** Elemento susceptible de tomar valores (cada una de las columnas de la tabla).

**Dominio (D<sub>i</sub>):** Conjunto de valores que puede tomar un atributo (se considera finito).

Tupla: Cada uno de los elementos que contiene una instancia de la relación (filas).

Para obtener un modelo relacional a partir de un modelo Entidad-Relación se tienen varias reglas (Thomas & Carolyn, 2005):

1. Se transforman en tablas todos los tipos de entidades y relaciones que aparecen en el diagrama E/R
  2. Se seleccionan las claves primarias para cada una de las tablas de nuestro esquema lógico.
  3. Se fusionan aquellas tablas que compartan su clave primaria.
1. Toda entidad fuerte se transforma en una relación
    - Se conservan los atributos y la llave primaria
    - En llaves candidatas se establece restricción de unicidad
    - Atributos compuestos se colocan en forma individual
    - Atributos multivalorados se crea una nueva relación propagando la llave primaria de la relación como foránea a la nueva relación.
    - Atributos derivados o calculados se establecen como atributos calculados
    - Se establecen restricciones sobre atributos

Simbología a utilizar

PRIMARY KEY (PK)  
 UNIQUE (U)  
 FOREIGN KEY (FK)  
 CHECK (CK)  
 Calculados (C)  
 Discriminantes (D)  
 No obligatorio (N)

Ejemplo:

PROFESOR={cveProf(PK),nomProf,apPat,apMat(N),rfcProf(U),edad(C)\*CS1}  
 EMAILPROF={cveProf(PK,FK),email(PK)}

\*CS1 Se obtiene del rfc del profesor

2. Entidades Débiles: Se crea la relación conservando todos sus atributos y se propaga la llave principal de la entidad fuerte de la que depende. La llave principal se formará con la llave primaria de la entidad fuerte y el discriminante de la entidad débil.

### 3. Relaciones:

1:1 La llave primaria de una entidad se propaga a la otra entidad como llave foránea con la restricción de unicidad, dependiendo de la cardinalidad mínima, si la cardinalidad mínima es cero, se pasa el atributo de la entidad de cardinalidad uno, hacia la de cardinalidad cero, si en ambas es la misma cardinalidad, dependerá del contexto.

m:1 ó 1:m La llave primaria de la entidad que tiene la cardinalidad uno se propaga a la relación de muchos como llave foránea.

m: m Se genera una nueva relación que contendrá la llave primaria de cada una de las entidades que une, pero como llaves foráneas y juntas como llave primaria, además de sus atributos propios (descriptivos) de la relación.

Las relaciones (1:1, 1:m, m:1) que contengan atributos descriptivos, se tienen dos opciones. Generar una nueva relación que contendrá la llave primaria de cada una de las entidades que une como llaves foráneas, además de sus atributos descriptivos, conservando la unicidad en la nueva relación. Propagar la llave primaria de la entidad uno junto con todos los atributos de la relación a la entidad m.

Clave primaria: Clave primaria: Si hay atributos propios de la relación:

Los atributos correspondientes al tipo de relación, a los que tal vez añadiremos algunos atributos propios dependiendo de la semántica del problema.

Claves externas: Una por cada una de las claves primarias de las de las entidades que intervienen en la relación.

Nota: Las relaciones entre entidades débiles y fuertes no hay que pasarlas a tablas porque la relación se recoge como parte de la clave primaria de la entidad débil (la parte correspondiente a la clave primaria de la entidad fuerte es una clave externa que apunta a la tabla derivada de la entidad fuerte).



## **BIBLIOGRAFÍA**

C.J. Date: "Introducción a los sistemas de bases de datos". Prentice Prentice Hall, 2001 [7ª edición]. ISBN 968 -444-419-22.

Ramez A. Elmasri & Shamkant B. Navathe: "Fundamentos de Sistemas de Bases de Datos". Addison - Wesley, 2007 [5ª edición]. ISBN 84-782 782-9085 -0.

Thomas M. Connolly & Carolyn E. Begg: "Sistemas de Bases de Datos " Addison -Wesley, 2005 [4ª edición]. ISBN 84-782-9075-33.

## PRÁCTICA 10

### GENERACIÓN DE ESPECIFICACIONES DE DESARROLLO DEL SISTEMA DE INFORMACIÓN

#### OBJETIVO

Generar las especificaciones para el desarrollo del sistema de información a partir del diseño detallado.

#### INTRODUCCIÓN

Estas especificaciones definen la construcción del sistema de información a partir de las unidades básicas de desarrollo (en adelante, componentes), entendiendo como tales unidades independientes y coherentes de construcción y ejecución, que se corresponden con un empaquetamiento físico de los elementos del diseño de detalle, como pueden ser módulos, clases o especificaciones de interfaz (Fernández, 2006).

La división del sistema de información en subsistemas de diseño proporciona, por continuidad, una primera división en subsistemas de construcción, definiendo para cada uno de ellos los componentes que lo integran. Si se considera necesario, un subsistema de diseño se podrá dividir a su vez en sucesivos niveles para mayor claridad de las especificaciones de construcción.

Las dependencias entre subsistemas de diseño proporcionan información para establecer las dependencias entre los subsistemas de construcción y, por lo tanto, definir el orden o secuencia que se debe seguir en la construcción y en la realización de las pruebas (Kendall, 2005).

También se generan las especificaciones necesarias para la creación de las estructuras de datos en los gestores de bases de datos o sistemas de ficheros.

Pressman (1997) menciona que el producto resultante de esta actividad es el conjunto de las especificaciones de construcción del sistema de información, que comprende:

- Especificación del entorno de construcción.
- Descripción de subsistemas de construcción y dependencias.
- Descripción de componentes.
- Plan de integración del sistema de información.
- Especificación detallada de componentes.
- Especificación de la estructura física de datos

A continuación, se presenta un ejemplo de las principales especificaciones con sus respectivas tareas, productos, técnicas y prácticas, así como los participantes:

Tarea	Productos	Técnicas y Prácticas	Participantes
Especificación del Entorno de Construcción	<ul style="list-style-type: none"> <li>- Especificaciones de Construcción del Sistema de Información:                             <ul style="list-style-type: none"> <li>o Especificación del Entorno de Construcción</li> </ul> </li> </ul>		<ul style="list-style-type: none"> <li>- Equipo de Arquitectura</li> <li>- Equipo del Proyecto</li> <li>- Equipo de Soporte Técnico</li> <li>- Equipo de Sistemas</li> <li>- Equipo de Seguridad</li> </ul>
Definición de Componentes y Subsistemas de Construcción	<ul style="list-style-type: none"> <li>- Especificaciones de Construcción del Sistema de Información:                             <ul style="list-style-type: none"> <li>o Descripción de Subsistemas de Construcción y Dependencias</li> <li>o Descripción de Componentes</li> <li>o Plan de Integración del Sistema de Información</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>- Diagrama de Estructura</li> <li>- Matricial</li> <li>- Diagrama de Componentes</li> <li>- Diagrama de Despliegue</li> </ul>	<ul style="list-style-type: none"> <li>- Equipo de Arquitectura</li> <li>- Equipo del Proyecto</li> </ul>
Elaboración de Especificaciones de Construcción	<ul style="list-style-type: none"> <li>- Especificaciones de Construcción del Sistema de Información:                             <ul style="list-style-type: none"> <li>o Especificación Detallada de Componentes</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>- Diagrama de Componentes</li> </ul>	<ul style="list-style-type: none"> <li>- Equipo del Proyecto</li> </ul>
Elaboración de Especificaciones del Modelo Físico de Datos	<ul style="list-style-type: none"> <li>- Especificaciones de Construcción del Sistema de Información:                             <ul style="list-style-type: none"> <li>o Especificación de la Estructura Física de Datos</li> </ul> </li> </ul>		<ul style="list-style-type: none"> <li>- Equipo del Proyecto</li> <li>- Administradores de la Base de Datos</li> </ul>

## DESARROLLO

En esta actividad se generan las especificaciones para la construcción del sistema de información, a partir del diseño detallado.

## CONCLUSIONES

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica:

---



---



---

## **BIBLIOGRAFÍA**

Fernández, V. (2006). Desarrollo de sistemas de información: una metodología basada en el modelado. México: UPC.

Kendall, K. E. (2005). Análisis y diseño de sistemas (6.<sup>a</sup> ed.). México: Pearson.

Pressman, R. (1997). Ingeniería del software: un enfoque práctico. México: McGraw-Hill.

