

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Contaduría y Administración

Maestría en Tecnologías de la Información y la Comunicación



Tema de Tesis:

BASE DE DATOS PARA SISTEMA DE GESTIÓN DE HORARIOS DE LA FCA DE LA
UABC (SIGAF)

PARA OBTENER EL GRADO DE

MAESTRO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Presenta:

DELIA YADIRA GAMBINO CANALES

Bajo la dirección de:

DRA. MARGARITA RAMÍREZ RAMÍREZ

Codirección

M.C. NORA DEL CARMEN OSUNA MILLAN

TIJUANA, BAJA CALIFORNIA, MÉXICO

OCTUBRE 2015

Dedicatoria

A mi esposo Julio, por su paciencia y apoyo incondicional en este proyecto, así como sus certeras y valiosas aportaciones que me hacen ver las cosas desde una perspectiva diferente.

A mis hijos Julio, David y Ana Gabriela por el tiempo que les pertenecía y que les ha hecho ver el estudio y trabajo como parte natural de la vida.

A mis padres, por haber sembrado en mí el deseo de superación a través del estudio en mi juventud y trabajo en mi madurez pero sobre todo por sus enseñanzas, consejos y ejemplo constante.

A mis hermanos, por apoyarme siempre con sus consejos.

Agradecimientos

Deseo agradecer a

La UABC por la oportunidad que me brindó de culminar una etapa más de mi vida profesional en esta institución.

A mi director de tesis, Dra. Margarita Ramírez Ramírez, por su apoyo y sus valiosas observaciones, así como por impulsar siempre la calidad a lo largo de esta maestría.

A mis profesores y que influyeron positivamente con su experiencia en mi formación académica.

Delia Yadira Gambino Canales

Resumen.

Gran parte de las instituciones educativas enfrentan dificultades al realizar la generación de horarios, situación que puede presentarse en cualquier país y nivel educativo. En la actualidad, existen estudios y desarrollos a nivel mundial enfocados a la solución de esta problemática en diferentes instituciones. La Facultad de Contaduría y Administración de la Universidad Autónoma de Baja California, no es ajena a esta situación ya que maneja un sistema con limitaciones en su Base de Datos, motivo por el cual se diseñó una Base de Datos que es el motivo del presente trabajo, diseñada dentro del marco de requerimientos actuales que formará parte de un sistema de horarios y que tomará ventaja de la información de una forma completa, eficiente y actualizada, que ofrecerá las ventajas con las que no se contaba, como disponibilidad de información presente y pasada, escalabilidad, seguridad y evolución técnica, para permitir completar el crecimiento del sistema en general si esto fuera necesario.

Abstract

Much of the educational institutions face difficulties in generating school schedules, a situation that can occur in any country and education level. Actually, there are studies and developments worldwide aimed at resolving this situation in different institutions. The Faculty of Accounting and Administration at the Autonomous University of Baja California, is no stranger to this situation and handles a system with limitations in their database, which this study was for. The new Database will be designed within the framework of existing requirements and will become part of an scheduling system taking advantage of the information in a complete, efficient and updated way and will offer advantages that were not available, such as availability of current and past information, scalability, security and technical evolution, thereby allowing complete growth of the overall system if necessary.

Índice

CAPÍTULO I INTRODUCCIÓN.....	1
1.1 Antecedentes	2
1.2 Planteamiento del problema de investigación	5
1.3 Objetivos	6
1.3.1 Objetivo General	6
1.3.2 Objetivos Específicos.....	6
1.4 Justificación.....	6
1.5 Alcance y limitaciones	7
1.5.1 Alcance:	7
1.5.2 Limitaciones:	7
CAPÍTULO II MARCO TEÓRICO	8
2.1 Bases de Datos.....	9
2.2 Diseño de bases de datos.	9
2.3 Diseño conceptual.	10
2.4 Modelo entidad-relación.	12
2.4.1 Entidades.....	12
2.4.2 Atributos.	12
2.4.3 Relaciones.	14
2.5 Diseño lógico.....	15
2.5.1 Claves	16
2.5.2 Integridad.....	17
2.5.3 Situaciones que podrían romper con la Integridad de la base de datos.....	20
2.5.4 Normalización.	21
2.6 Diccionario de Datos.	24
2.7 Metodologías para Desarrollo de software.	25
2.7.1 Metodología RUP.	26
2.7.2 Metodología Scrum.....	33

2.8	Gestores de bases de datos.	40
2.8.1	MySQL.	40
2.9	Diseño físico.	41
CAPÍTULO III METODOLOGÍA		42
3.1	Modelado de una base de datos.	43
3.2	Diseño conceptual.	44
3.2.1	Identificación de entidades.	46
3.2.2	Identificación de relaciones entre las entidades.	48
3.2.3	Identificar los atributos de las entidades y sus relaciones y asociarlos	49
3.3	Diseño Lógico.	50
3.3.1	Representación Gráfica de la Integridad Referencial.	50
3.4	Metodología Scrum.	51
CAPÍTULO IV DESARROLLO		54
4.1	Diseño Conceptual del Plan de Estudios.	55
4.2	Diseño Lógico del plan de estudios.	57
4.2.1	Aplicación de reglas de Integridad para el sistema SIGAF.	58
4.2.2	Entidad planestudio.	64
4.2.3	Entidad plan_programa.	65
4.2.4	Entidad uaprendizaje.	65
4.2.5	Entidad programaedu.	66
4.2.6	Entidad p_ua.	67
4.2.7	Entidad detalleseriacion.	68
4.2.8	Entidad Puesto.	70
4.2.9	Otras características de plan de estudios.	71
4.2.10	Otras características de las unidades de aprendizaje.	72
4.3	Diseño lógico carga académica.	73
4.3.1	Entidad carga.	73
4.3.2	Entidad grupos.	73
4.3.3	Entidad periodos.	74
4.3.4	Entidad turnos.	74
4.3.5	Entidad Status_procesos.	74
4.4	Diseño lógico disponibilidad docente.	75
4.4.1	Características que conforman el perfil profesional y laboral de docentes.	75
4.4.2	Entidad empleados.	76
4.4.3	Entidad Categorías.	76

4.4.4	Entidad dias.....	77
4.4.5	Entidad horas.	77
4.4.6	Disponibilidad docente.	78
4.5	Diseño lógico de Horarios.....	78
4.5.1	Entidad horarios.....	79
4.6	Diccionario de Datos.	79
4.7	Bitácora.....	79
4.7	Diseño físico.	81
CAPÍTULO V RESULTADOS.....		82
5.1	Diseñar el modelo conceptual y lógico de la Base de Datos.....	83
5.2	Elaboración diccionario de datos.	83
5.3	Diseño físico de la base de datos que permita el almacenamiento de los datos del proceso de horarios de una forma segura y eficiente que incluya una bitácora de movimientos, así como el almacenamiento de los datos básicos para su funcionamiento.	86
CAPÍTULO VI CONCLUSIONES Y TRABAJOS FUTUROS		91
6.1	Conclusiones.....	92
6.2	Trabajos futuros.....	92
BIBLIOGRAFIA		91
GLOSARIO		99

Índice de figuras

Figura 1. Desarrollo iterativo	27
Figura 2. Los casos de uso a través de todas las fases.....	28
Figura 3. Organización de RUP	29
Figura 4. Flujo de actividades de los core roles.	37
Figura 5. Forma de trabajo de Scrum, y una comparación respecto a la forma tradicional.....	39
Figura 6. Simbología que se utilizará en el modelo E-R. Fuente: Estructura y organización de una Base de Datos....	47
Figura 7. Ejemplo de representación de las entidades docente, curso y perfil.	47
Figura 8. Ejemplo de la representación relaciones entre entidades.	49
Figura 9. Ejemplo para la entidad docentes y algunos de sus atributos.....	50
Figura 10. Ejemplo de aplicación de metodología scrum para la BD del sistema SIGAF.	53
Figura 11. Ejemplo sobre seriación de uac.	68
Figura 12. Conexión entre uac.....	70
Figura 13. Esquema E-R para la BD del sistema SIGAF..	84
Figura 14. Esquema relacional para la BD del sistema SIGAF	85

Índice de tablas

Tabla 1. Ejemplo de entidades DOCENTES Y CONTRATOS.....	21
Tabla 2. Artefactos e Hitos de las fases de RUP.....	33
Tabla 3. Comparativo entre scrum y la gestión de proyectos tradicional	35
Tabla 4. Ejemplo Integridad Referencial.	58
Tabla 5. Ejemplo de registros en la bitácora de movimientos.....	64
Tabla 6. Ejemplo de ua seriadas para programas educativos diferentes	69
Tabla 7. Ejemplo sobre el almacenamiento de datos en la entidad Puestos.	70
Tabla 8. Valores que se almacenaran en la entidad días.....	77
Tabla 9. Valores que almacena la entidad horas	77
Tabla 10. Representación gráfica del diccionario de datos..	86
Tabla 11. Información de materias alimentada en la Base de Datos..	88
Tabla 12. Información de periodos alimentada en la Base de datos.....	88
Tabla 13. Ejemplo de datos de lo que la bitácora permite almacenar..	89

Capítulo I

Introducción

1.1 Antecedentes

Para las instituciones educativas es importante realizar con anticipación una planeación adecuada de los horarios escolares para que los alumnos, docentes y personal administrativo puedan disponer de la oferta educativa una vez iniciado el ciclo escolar. Ésta actividad se realiza en todos los niveles y es tema recurrente cada inicio de ciclo escolar en cualquier institución; algunas gestionan el proceso mediante un sistema, otras de forma manual; en este último caso se invierte un tiempo excesivo que puede extenderse a semanas o incluso, meses.

El tema de asignación de horarios escolares ha sido tratado de forma amplia en el área de matemáticas, de forma específica en el área de investigación de operaciones, materia que trata problemas matemáticos complejos y tiene una rama dedicada al estudio de los horarios llamada *Timetabling* (generación horaria), dentro de la cual existe una subdivisión llamada *Class Scheduling* y dentro de esta tres tipos de problemas: (Hernández, Miranda P., & Rey, 2008)

- Programación de horarios de evaluaciones y exámenes (*examination timetabling*).
- Programación de horarios de clases para colegios (*school course timetabling*).
- Programación de horarios de clases para instituciones educativas universitarias (*university course timetabling*).

La Facultad de Contaduría y Administración (FCA) de la Universidad Autónoma Baja California (UABC) cuenta con un sistema que maneja una Base de Datos (BD) de horarios realizado hace 15 años aproximadamente. El procedimiento actual para la gestión del proceso de horarios, es el siguiente:

- Se realiza una copia a la BD del periodo anterior.
- Se especifican los grupos considerados para el periodo.
- Coordinadores de carrera se encargan de recopilar información referente a las unidades de aprendizaje que se ofrecerán en cada programa educativo con base en el plan de estudios vigente, los grupos, las unidades de aprendizaje y turnos que se ofrecerán.

- Administradores del proceso gestionan la recopilación de información sobre disponibilidad de horarios y unidades de aprendizaje a impartir por parte de los docentes.
- La información recopilada se transfiere a la copia de la BD que contiene aún los datos del periodo anterior, los cuales serán modificados hasta que la información corresponda con la del periodo actual.

En la actualidad, la información del proceso se maneja en un sistema gestor de bases de datos de Interbase diseñado de tal forma que no puede escalarse, carece de documentación y no es posible realizar modificaciones directas a los datos excepto a través de una copia.

Como resultado de lo expuesto en los párrafos anteriores, surge la necesidad de una BD actualizada en un sistema gestor de base de datos (SGBD) para la administración de la información del proceso; la BD que formará parte de las diferentes etapas del desarrollo de software y del equipo de trabajo para el sistema SIGAF es el tema de estudio de este trabajo.

Las bases de datos conforman una herramienta poderosa para almacenar la información de cualquier proceso. Como definición, una base de datos es un conjunto de datos organizados y relacionados, los cuales corresponden al dominio de un problema. La organización de estos datos, supone una respuesta a las necesidades de información de dicho dominio que por lo general es manejado en formato electrónico y forma parte de un entorno computacional, llamado sistema de gestión de base de datos (SGBD) encargado del almacenamiento de los datos del dominio del problema, así como del software por medio del cual se gestiona la BD, los usuarios y los datos que describen la base de datos (metadatos). (Meneses E., Gálvez C., & Chavarro P., 2011). Por ese motivo, es importante realizar una selección adecuada y cuidadosa del SGBD y elegir el que más se ajuste a las necesidades propias de la institución en la que se implementará la BD.

Alrededor del mundo, se han realizado diversos trabajos sobre la asignación de horarios escolares y se ha tratado resolver cada caso de acuerdo a sus características particulares debido a la complejidad del proceso y a la cantidad de variables que intervienen, que son diferentes para cada organización. Existen sistemas comerciales genéricos pero no todos se ajustan al 100% a las necesidades y variables de cada institución educativa.

Dentro de los estudios que se han realizado sobre los horarios escolares, se encuentra el estudio publicado bajo el nombre “Asignación de horarios de clases universitarias mediante algoritmos evolutivos” que trata sobre la generación horaria en la universidad de La Guajira, Riohacha (Colombia), por medio del uso de la técnica meta heurística de algoritmos evolutivos. En dicha investigación se logró la realización de un software que cumple con un tiempo de 10 minutos en la asignación de los horarios. Esta Universidad, tienen integrados los alumnos dentro de su sistema, los cuales se consideran para la generación horaria. El sistema maneja una BD de Access y su enfoque principal es el algoritmo que resuelve las comparaciones para definir los horarios; desde el punto de vista de los autores se logran buenos resultados en cuanto a tiempo con este nuevo sistema, debido a que el proceso pasó de tomar días a tomar 10 minutos de cálculo del proceso con la información cargada. No se realiza una explicación amplia sobre la base de datos.

En ese trabajo se realizó un sistema de generación de horarios que asigna de forma satisfactoria las clases y salones para la carrera de ingeniería industrial de la facultad de ingeniería de la universidad de La Guajira y que resuelve las diferentes situaciones que se presentaban. Por ejemplo, existían docentes a los que se les asignaban horarios de clase a la misma hora para diferentes materias, otro caso era si un alumno no podía tomar todas las materias que deseara porque los horarios asignados a sus materias estaban a la misma hora y tenían que elegir una y dejar de tomar otra de las asignaturas. En esta universidad las personas encargadas de la asignación del horario escolar no contaban con información suficiente sobre el alumnado total que asistiría el siguiente periodo, lo que se daba como resultado una serie de problemas al terminar la asignación de horarios escolares: se generaba exceso de grupos y se tenían que asignar docentes de última hora, lo que repercutía en la calidad con que se realizaba la selección del personal docente. Una vez terminados los horarios de forma manual, existía desorganización en la Institución e inconformidades personales, pero lo que causaba mayor preocupación, era el tiempo que llevaba realizar la planeación de los Horarios.

Existe otro estudio realizado por una universidad de Túnez (Dammak, Elloumi, & Kamoun, An enterprise system component based on graph colouring for exam timetabling a case study in a Tunisian university, 2007) que se enfoca en resolver la calendarización de exámenes universitarios (exam timetabling) publicado bajo el nombre de “Un componente del sistema empresarial basado en gráficos de colores para la programación de horarios de

evaluaciones y exámenes de un caso de estudio de la universidad de Túnez”. El estudio comprende la asignación de horarios de recuperación de la Facultad de Economía y Gestión de Ciencias de Sfax y su objetivo primordial es generar el calendario de exámenes de los alumnos matriculados en la Universidad de Sfax, Túnez de forma automática, al aplicar dos procedimientos basados en gráficos de colores. El sistema resultante formara parte de un sistema integral que incluye además los horarios de cursos, estos últimos tratados en otro artículo (Dammak, Elloumi, & Kamoun, Course timetabling at a Tunisian University: A case study , 2008).

Los horarios de exámenes están conformados por restricciones duras y suaves. En las duras no se permite a ningún alumno tomar dos exámenes en el mismo intervalo de tiempo, aunque podría presentarse la situación en que las restricciones duras se satisfagan pero no necesariamente las suaves; por ejemplo que un estudiante no tome dos exámenes consecutivos y no más de dos exámenes por día; es decir existen restricciones que pueden violarse en caso necesario sin que el horario resulte perjudicado.

En esta universidad se tienen dos tipos de exámenes: El primero lo presentan todos los estudiantes y el segundo es solo para quienes reprobaron el primero.

El estudio menciona que el sistema fue probado con datos reales y como se menciono en párrafos anteriores, se usaron dos procedimientos basados en gráficos de colores, de ahí el énfasis en la importancia de entender las técnicas avanzadas de la teoría de grafos, particularmente las relacionadas a coloración grafica. Para resolver los horarios en este trabajo se aplicaron dos procedimientos heurísticos: heurística tamaño de examen y heurística examen grado max-min (Dammak, Elloumi, & Kamoun, An enterprise system component based on graph colouring for exam timetabling a case study in a Tunisian university, 2007).

1.2 Planteamiento del problema de investigación

El sistema de horarios actual que maneja la FCA de la UABC, cuenta con una Base de Datos de las que se desprenden circunstancias que es necesario corregir, mediante un diseño actualizado de base de datos, como por ejemplo:

- Se realiza una gran cantidad de capturas manuales, que repercute en el tiempo empleado en la generación de horarios.
- No se cuenta con todos los reportes que se requieren.
- No se cuenta con documentación sobre la Base de Datos.
- No es posible optimizar la Base de Datos.
- No se aprovechan los avances tecnológicos actuales para el manejo de BD.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar una base de datos para el sistema de gestión académica de la FCA de la UABC, basado en los requisitos especificados en el análisis realizado del sistema, los cuales fueron solicitados por el usuario en esta etapa que permita el almacenamiento de información del proceso de gestión de horarios.

1.3.2 Objetivos Específicos

1. Diseñar el modelo conceptual y lógico de la Base de Datos.
2. Elaboración diccionario de datos.
3. Diseño físico de la base de datos que permita el almacenamiento de los datos del proceso de horarios de una forma segura y eficiente que incluya una bitácora de movimientos, así como el almacenamiento de los datos básicos para su funcionamiento.

1.4 Justificación

Existe la necesidad de un nuevo sistema, y por lo tanto de un nuevo diseño de BD para gestionar el proceso de horarios, que sea capaz de cubrir y atender las necesidades actuales de la facultad que son las siguientes:

- Optimización del tiempo del proceso, que no es posible con la BD con la que se cuenta en la actualidad.
- Aprovechar las ventajas que ofrece la tecnología actual en cuanto al manejo de bases de datos, con un avance significativo respecto a con lo que se contaba hace 15 años.

- Manejo de una BD escalable que considere las necesidades actuales de la facultad, especificadas en el análisis del proyecto SIGAF.
- Se requiere el manejo de la información de una forma dinámica en cuanto a cubrir las necesidades actuales de reportes y el diseño de nuevos reportes de acuerdo a las circunstancias cambiantes del entorno.
- Base de Datos debidamente documentada que sirva como material de apoyo a los encargados del proceso y en mayor medida al nuevo personal contratado.

1.5 Alcance y limitaciones

El diseño e implementación de la base de datos para el Sistema SIGAF contempla el siguiente alcance y limitaciones:

1.5.1 Alcance:

- Diseño conceptual de la BD.
- Diseño lógico de la BD.
- Diseño físico.
- Definición de BD específica para SGBD MYSQL.
- Realización de Diccionario de Datos.

1.5.2 Limitaciones:

Se considera una fracción de la UABC que incluirá:

- Proceso de horarios
- FCA de la UABC.
- Campus Tijuana.
- Programas de licenciatura
- Basado en requerimientos de los usuarios y las necesidades de información especificadas en el documento de análisis.

Capítulo II

Marco Teórico

2.1 Bases de Datos

Existen diversos conceptos que se manejan respecto a la definición de una base de datos, pero en general todos se refieren de forma similar, como el mencionado en la introducción, donde se especifica que “Un conjunto de datos organizados y relacionados, los cuales corresponden al dominio de un problema. La organización de estos datos, supone una respuesta a las necesidades de información de dicho dominio que por lo general es manejado en formato electrónico y forma parte de un entorno computacional, llamado sistema de gestión de base de datos (SGBD) encargado del almacenamiento de los datos del dominio del problema, así como del software por medio del cual se gestiona la BD, los usuarios y los datos que describen la base de datos (metadatos)”. (Meneses E., Gálvez C., & Chavarro P., 2011).

La información almacenada que se menciona en el concepto, deberá cumplir con los siguientes propósitos:

- Disponibilidad al momento que el usuario la requiera, con una velocidad adecuada.
- Datos consistentes y precisos con relación entre ellos (integridad).
- Deberá proporcionar información, almacenamiento y recuperación eficientes (Trentin, 1992).
- La información por sí misma no dice nada, solo representa datos. Por tal motivo, para que la información sea útil, deberá presentarse de tal forma que pueda aportar conocimiento sobre la información almacenada, lo que la convierte en una herramienta que permite a las organizaciones administración, planeación y toma de decisiones estratégicas para el futuro. (Zepeda S., 2008)

2.2 Diseño de bases de datos.

El diseño de una base de datos se refiere a la forma en que ésta está organizada. Hoy en día en las organizaciones, el diseño de base de datos mantiene una importancia relevante; en el pasado se daba más importancia a los algoritmos que se utilizaban para el desarrollo de sistemas, pero en la actualidad se consideran con el mismo grado de importancia y las bases de datos han pasado a ser consideradas un tema que no debe pasarse por alto para la formación de personal de TI.

Como se mencionó en párrafos anteriores, existen herramientas poderosas que son los llamados Sistemas Gestores de Bases de Datos que ofrecen una cantidad considerable de herramientas con las cuales la información de una base de datos se puede organizar y manipular, al permitir entre otras cosas diseños de bases de datos relacionales, procesamiento, optimización y desarrollo de consultas poderosas por medio de lenguajes de consulta, control de concurrencia, gestión de transacciones y recuperación.

La dificultad que representa el diseño de una bases de datos resulta beneficiada al utilizar modelos, donde éstos representan abstracciones simplificadas de situaciones del mundo real de las cuales se obtienen relaciones y entidades que se pueden explorar de tal forma que si el modelo es suficientemente bueno, se tiene la ventaja de contar con los beneficios de la información que proporciona (Peter & Coronel, 2005). En ese contexto, es importante considerar los tres diferentes niveles de abstracción para bases de datos (Desiré Atangana & Sepúlveda Lima, 2007):

- Diseño Conceptual.
- Diseño lógico.
- Diseño físico.

Hay aspectos que no deben ser ignorados a la hora de diseñar una base de datos, como trabajar de cerca con los usuarios, y seguir una metodología que sirva de guía a la hora del diseño, lo que dará como resultado un trabajo con orden y estructura. No se debe asumir que el diseño se lleve a cabo estrictamente en el orden de cada una de las fases; en la mayoría de los casos, a lo largo del diseño hay que volver a examinar y realizar cambios; es un proceso en el que se tiene que volver a repetir el ciclo de revisión constantemente a lo largo de la vida del diseño, y a lo largo de todas las etapas. Se realiza una explicación detallada de las tres etapas mencionadas en los siguientes párrafos.

2.3 Diseño conceptual.

“Este modelo es una descripción de alto nivel, por medio del cual se abstraen del dominio de un problema de la realidad las características que se desean modelar mediante representaciones gráficas y lingüísticas. En este diseño se utiliza un alto nivel de abstracción porque su objetivo principal no es la definición de la estructura, sino el

contenido general de la base de datos donde todavía no importa el sistema gestor de base de datos que se vaya a utilizar” (Meneses E., Gálvez C., & Chavarro P., 2011).

Entre los modelos más utilizados para este diseño, se pueden mencionar, el modelo entidad-relación y el modelo orientado a objetos. Para efectos del presente trabajo, se hará uso del modelo entidad-relación.

El diagrama de ingeniería de software entidad-relación utilizado para modelar datos de alta calidad, se empieza a convertir en la herramienta más extendida de forma universal para el modelado de datos (Zapata J., González C., & Chaverra M., 2011).

Este modelo se basa en una percepción del mundo real, que consta de una colección de objetos básicos, llamados entidades, atributos y relaciones entre objetos. El diseño conceptual nace a partir de la especificación de los requisitos del usuario; es por ello que los requisitos deben estar claramente especificados y estudiados, debido a que habría problemas al momento de diseñar la BD si los requisitos no están claramente especificados o si hace falta información en su especificación. En caso de no tener disponible esta información, se podría recurrir a la realización de entrevistas con los usuarios, observación de los procedimientos y revisión de toda la documentación disponible del proceso. El esquema va a contener información sobre las entidades, relaciones, atributos e identificadores, así como su documentación correspondiente.

Un esquema conceptual puede estar constituido por varios esquemas llamados esquemas conceptuales locales; por lo general se realiza de esta manera para considerar los puntos de vista que los usuarios tienen de su área o especialidad de trabajo como por ejemplo: recursos humanos, producción, finanzas o las divisiones existentes dentro de la empresa donde dichas divisiones son llamadas vistas; en el caso de grandes sistemas, dichas vistas se combinan en un esquema global. Para el caso del presente trabajo se maneja un solo diagrama corresponde al área de escolares.

El diseño conceptual es totalmente independiente de las consideraciones físicas, del SGBD, de los detalles de implementación, el lenguaje de desarrollo que se usará y la infraestructura que se utilizará.

2.4 Modelo entidad-relación.

El modelo entidad-relación (E-R), es un modelo de datos conceptual, que como se mencionó en párrafos anteriores, representa una percepción del dominio de un problema. Para realizar esta representación, se realiza una abstracción de la realidad, de lo que es de interés en ese momento, a partir de la cual resultan una serie de objetos gráficos relacionados entre sí. Existen diferentes formatos de presentación para el modelo E-R, como el modelo chen, y el modelo pata de gallo (Peter & Coronel, 2005); para la realización del presente trabajo, se utilizará la notación lo más cercana posible al modelo chen. El modelo E-R está conformado por tres elementos principales:

2.4.1 Entidades.

Es un objeto de la realidad que se quiere representar, por lo general se definen como entidades los objetos como por ejemplo personas, cosas o eventos como compras o ventas; la forma de representarlos es mediante un rectángulo que contiene el nombre de la entidad en el centro en mayúsculas, donde cada entidad deberá representarse de forma única. Cabe señalar que para el modelo conceptual la representación del nombre de la entidad es por lo general en singular, mientras que para el diseño físico y lógico en plural.

2.4.2 Atributos.

Es una propiedad interesante que tiene la entidad que se representa; los atributos describen, cuantifican, califican o clasifican una entidad y se asocian a un tipo de dato como tipo fecha, tipo entero, entre otros. La forma de representarlos en el modelo E-R es especificar el nombre del atributo en minúsculas y singular dentro de un óvalo que irá conectado a la entidad a través de una línea (Bender, Deco, González S., Hallo, & Ponce G., 2014). Dentro de los atributos, los autores manejan diferentes clasificaciones, entre las que se manejan las siguientes:

- **Atributos de clave y clave primaria**

Las claves son un aspecto fundamental que debe quedar claramente comprendido si se trata de bases de datos; estas son usadas principalmente dentro del esquema relacional de base de datos, sin embargo, es posible identificar claves importantes desde el esquema E-R.

El concepto claves, está basado en el concepto determinación y dependencia funcional. La determinación dice que si en una BD se conoce el valor del atributo A, éste determina el valor de B. Por ejemplo, si en una entidad llamada docentes el atributo A representa el número de docente y B representa el apellido, al conocer el número de docente, automáticamente sabemos su apellido, e incluso otros datos, entonces se dice que atributo B está determinado por A, pero esto no puede ser a la inversa en este caso, ya que el apellido no podría determinar el número de docente, debido a que puede haber diferentes docentes con el mismo apellido. La dependencia funcional, se refiere a los atributos que dependen de la clave en las BD relacionales, y es precisamente en la dependencia funcional en la que se basa el concepto de claves. Existen diferentes tipos de claves como superclave, clave candidata, clave primaria, clave secundaria y clave foránea que se revisarán más adelante en el diseño lógico, que es donde son aplicadas en su totalidad, (Peter & Coronel, 2005). Por el momento basta entender una clave primaria como “el atributo o los atributos que determinan una entidad unívocamente” (Trentin, 1992). Una clave primaria no puede contener valores nulos.

En un diagrama E-R, las claves primarias se representan al subrayar el nombre del atributo. Lo ideal es que las claves primarias se conformen de un solo atributo, pero no siempre es así, en ocasiones se conforman de dos o más atributos, en cuyo caso son llamadas claves compuestas.

- **Atributos simples y compuestos.**

Se refiere a datos como la dirección o el nombre de las personas, los cuales deben revisarse con el usuario para verificar si será necesario acceder a una parte de esta información de forma independiente; por ejemplo en el caso del nombre de las personas, es necesario verificar si debe accederse al apellido paterno, al materno y al nombre de forma independiente; en este caso el atributo se dividirá en varios atributos simples, un atributo para el paterno, otro para el materno y otro para el nombre. Si no es necesario acceder de forma independiente a los detalles del atributo, este quedará conformado por nombre+paterno+materno en uno solo que conformarán el atributo nombre completo. Se debe tener cuidado de no confundir estos atributos con una clave compuesta.

- **Atributos univaluados y multivaluados.**

Los atributos univaluados son aquellos que tienen un solo valor por ejemplo, el registro federal de contribuyentes (RFC) de las personas representa un identificador único en México; no hay otro igual pero no necesariamente podría tratarse estrictamente de un atributo simple, sino compuesto y almacenado en un solo atributo, porque en este caso concreto, los dígitos representan cierta información: el primer dígito corresponden a la primer letra del apellido paterno, el segundo dígito corresponde a la segunda letra del apellido paterno (existen excepciones), el tercera dígito corresponderá a la primer letra del apellido materno, y de forma similar, el resto de los dígitos tienen un significado. Al momento que el campo es usado, no es necesario descomponerlo en partes, salvo con seguridad en la institución que lo emite de origen, por lo tanto este atributo es univaluado.

Los atributos multivaluados se refieren a los atributos que pueden tener más de un valor, por ejemplo, el teléfono; una persona podría tener más de un teléfono. Una alternativa de solución podría ser el agregar otra entidad que contenga los teléfonos de las personas; de esa manera se podrían agregar varios teléfonos de una sola persona u optar por la otra solución de tener dentro de la misma entidad los atributos teléfono, celular y teléfono de casa.

- **Atributos Derivados.**

Son aquellos que se derivan de otros, por ejemplo, la antigüedad de una persona puede ser el resultado de un cálculo: la fecha de ingreso y la fecha actual, da como resultado los días de antigüedad.

Una vez conformada la base de datos, una de las funciones relevantes de los atributos es el permitir la especificación de condiciones para filtrar los datos o definir ciertos criterios, para poder consultar únicamente la información de interés (Zepeda S., 2008). Por ejemplo, si se tienen almacenados diferentes planes de estudio de cierta universidad clasificados de acuerdo a la carrera, se podría llegar a extraer información referente únicamente a determinada carrera. Si se considera que en una base de datos se pueden llegar a almacenar una cantidad importante de información, la utilidad de los atributos en este sentido, resulta ser una herramienta útil.

2.4.3 Relaciones.

Es la forma en que las diferentes entidades están asociadas. Las relaciones, tienen dos propiedades: el grado se refiere al número de entidades sobre las cuales se realiza la relación; una relación que involucra una sola entidad, es una relación unaria, una relación de dos entidades, binaria y una relación de tres entidades se le llama relación terciaria. La otra propiedad, llamada cardinalidad se refiere al número de ocurrencias de una entidad asociadas a una ocurrencia de otra o la misma entidad, por medio de una relación. Existen tres tipos de cardinalidad: 1:1, 1: N, M: N. (Lara T., 2010).

En lo que se refiere al grado, por lo general, las relaciones resultantes son binarias, aunque pueden existir relaciones que involucren a más de dos entidades. En ese caso se descomponen en relaciones binarias equivalentes siempre que esto sea posible, para simplificar el modelo.

Respecto a la cardinalidad, si dos entidades tienen una interrelación de uno a uno (1 a 1), entonces el campo clave de una de las entidades debe aparecer en la otra. Si dos entidades tienen una interrelación de uno a muchos (1 a *), entonces el campo clave de la entidad del (1), debe aparecer en la entidad del muchos (*). Si dos entidades tienen una interrelación de muchos a muchos (* a *), entonces debe crearse una entidad que tenga los campos claves de las dos entidades.

2.5 Diseño lógico.

A partir del diseño conceptual, se empieza a trabajar el diseño lógico de la base de datos. El diseño lógico es una descripción de la estructura de la base de datos que ya puede ser procesado por el SGBD.

Los tipos de modelos lógicos que más se usan son el modelo relacional, el modelo de redes y el modelo jerárquico; a partir del diseño conceptual, que representa el esquema E-R se puede trabajar ya el diseño lógico hasta llegar a un esquema; en el caso de la presente tesis, el relacional que es el más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Una vez terminado el esquema relacional, el resultado es un modelo basado en las relaciones lógicas entre los datos que los organiza y representa en forma de tablas de dos dimensiones, consistente en filas conocidas como registros o tuplas y columnas de datos conocidas como campos (Dr. Codd en 1970) . El modelo

es fácil de entender, incluso para los usuarios que no son expertos debido a su sencillez: las relaciones representan una entidad que es un conjunto de filas o renglones.

Conceptualizar este esquema es sencillo: contiene la definición de las entidades de la base de datos y la relación que existe entre cada una, con su nombre. El tipo de información que se almacena en cada una de las entidades son los campos, así como los valores permitidos para los campos, llamados dominios. Los tipos de datos se refieren al tipo de dato que contiene cada campo: entero, carácter, fecha, texto, o cualquier otro tipo de dato. Es importante mencionar que el trabajo realizado en el diseño conceptual, servirá de base para el lógico por lo que el trabajo realizado en aquél podría representar un avance significativo, en base al nivel y detalle alcanzado. Una vez trasladado el esquema conceptual al esquema físico, se podrá almacenar información en forma de filas y columnas que ya podrá ser consultada a través de consultas, arma poderosa al momento de administrar información. Posterior a esto se puede trabajar con los conceptos de clave, integridad y normalización.

2.5.1 Claves

La clave está compuesta por atributos que por sí mismos, representan a otros. Para las bases de datos, existen diferentes tipos de claves: primaria, superclave, candidato, secundaria y foránea. Como se mencionó en párrafos anteriores, una clave primaria identificará unívocamente a cada tupla de la entidad, que no puede contener estos valores.

- **Clave Primaria.**

Es un atributo o conjunto de atributos que identifican de forma única a cada tupla de una entidad (Trentin, 1992). Si se toma como ejemplo la entidad docente el número de docente sería la clave primaria debido a que no hay dos números de docente repetidos en la entidad docente. Como se mencionó en párrafos anteriores, existen claves o primarias compuestas por dos columnas de una entidad y son llamadas claves compuestas o combinadas.

- **Superclave**

Un atributo (o combinación de atributos) que identifica de manera única a cada entidad en la tabla.

- **Clave candidato**

Una superclave mínima. Una superclave que no contiene un subconjunto de atributos que por sí mismo es una superclave.

- **Clave secundaria**

Es un atributo (o combinación de atributos) utilizado estrictamente para propósitos de recuperación de datos.

- **Clave foránea**

Un atributo (o combinación de atributos) en una tabla cuyos valores deben igualar a la clave primaria en otra tabla o ser nulos.

2.5.2 Integridad.

En las bases de datos relacionales, las reglas de integridad son fundamentales y deben considerarse a la hora trabajar el diseño; aunque algunos SGBD ofrecen esta funcionalidad de forma automática, es recomendable ratificar que dichas reglas se cumplan. La función de la integridad es asegurarse de que la base de datos no tenga inconsistencias en sus datos; es decir, garantiza que los datos sean válidos, seguros y precisos. La existencia de datos inconsistentes puede ser causa de errores que podría traer consecuencias a las empresas, una de estas económica.

¿Qué se debe entender por integridad? Su importancia puede ilustrarse con los siguientes ejemplos:

Si existieran datos inconsistentes en una base de datos en áreas clave de la organización como finanzas o contabilidad, de forma específica un error en una factura, es posible que tuviera que buscarse al cliente una o varias veces, con las consecuentes pérdidas económicas y de tiempo, además de las incomodidades de los stakeholders que pueden verse perjudicados por esta situación. Otro aspecto a considerar es que una BD representa datos del mundo real por ejemplo, para un empleado se deberá almacenar su nombre, debido a que tal como en el mundo real, todas las personas tienen nombres y un empleado sin nombre, no tendría razón de ser. En el ejemplo anterior, si en la vida real esto se cumple, también debe cumplirse en una base de datos a donde las reglas de la

organización se trasladan al diseño; si una persona necesitara un tratamiento médico consistente entre otras cosas en la administración diaria de una dosis de medicamento de 10 miligramos y llegara a presentarse la situación accidental o intencional de modificar o registrar el registro correspondiente de esa dosis a 100 mg, las consecuencias podrían ser mortales.

La integridad puede visualizarse desde diferentes ángulos, desde la perspectiva de los encargados de seguridad, administradores de bases de datos, arquitectos o modeladores de datos, propietarios de los datos o proveedores. Para un administrador de datos, “la integridad de los datos puede depender de que los datos alimentados en una BD sean precisos, válidos y coherentes, y existen posibilidades de que analicen también la integridad de entidades, dominios e integridad referencial”. Para un arquitecto o modelador de datos, la integridad de los datos “puede estar relacionada con el mantenimiento de entidades primarias únicas, no nulas. Las entidades contienen valores únicos definidos porque la no existencia de valores duplicados y por la existencia de una clave que los define, y permite el acceso a ellos de forma exclusiva” (Gelbstein, 2011).

De lo anterior, se deriva que los aspectos importantes que se deben considerar al momento de trabajar la integridad, son:

- Valor nulo.
- Datos requeridos
- Integridad de entidades
- Integridad referencial
- Restricciones de dominio
- Restricciones generales

- **Valor nulo.**

Valor nulo representa datos que faltan o que son desconocidos para un atributo de una entidad referencial. Se debe tener cuidado, y no confundir un valor nulo con un valor que contiene espacios en blanco o con un valor que contiene ceros, debido a que un valor nulo no tiene valor, al menos en ese momento.

- **Datos requeridos.**

Algunas columnas o campos, deben contener un valor (No Nulo). Algunos manejadores SGBD permiten especificar este punto al momento de crear la base de datos o incluso en cualquier momento aunque la recomendación es al inicio; de esta manera, no se permitirá que las nuevas tuplas que se den de alta contengan valores nulos en los campos que se hayan especificado como datos requeridos.

- **Integridad de Entidad.**

La regla de integridad de entidad dice que en una relación de entidades, ningún atributo de una clave primaria puede ser nulo, y todas las entradas de una clave primaria son únicas. Esto significa que en una clave primaria, no puede haber dos valores repetidos. “La integridad de la entidad asegura que las entidades (personas, cosas y eventos) se identifiquen de forma única en una base de datos” (Mannino, 2007).

- **Integridad Referencial.**

En una relación de entidades, la clave foránea puede tener un valor nulo o su valor puede tener un valor que corresponda al valor de una clave primaria de una entidad con la cual esté relacionada. “La integridad Referencial se asegura de que una base de datos tenga conexiones válidas” (Mannino, 2007).

Es importante considerar que la integridad referencial permite que las llaves foráneas puedan contener valores nulos, mas no siempre es recomendable, según sea el uso de los campos, tal como sucede en la vida real. Es responsabilidad del administrador de base de datos, evaluar cada situación y decidir si se permiten o no valores nulos en las llaves foráneas.

- **Restricciones de dominio. (Gelbstein, 2011)**

Todos los atributos de una entidad manejan un dominio, esto es, son los valores válidos que dicha columna puede almacenar. Las restricciones de dominio resultan de gran utilidad al momento de definir los atributos porque permite a los usuarios conocer los valores que un atributo puede manejar.

- **Restricciones Generales.**

Son reglas adicionales que especifican los usuarios o administradores de la base de datos que desean restringir algún aspecto. Estas reglas se refieren a las políticas o normas que maneja la institución a la que pertenece la base de datos. Por ejemplo, que no se acepta

personal egresado de cualquier universidad, por política de la empresa, sino únicamente se acepte a egresados de ciertas instituciones. Esto se puede especificar como una restricción general, para que no sea permitido ingresar valores inválidos en la base de datos.

2.5.3 Situaciones que podrían romper con la Integridad de la base de datos.

¿En qué errores se podría incurrir si no se revisa adecuadamente la integridad referencial?

Como ejemplo, se tomará la Tabla 1 que contiene las entidades DOCENTES Y CONTRATOS. Podría suceder por ejemplo que al momento de insertar un registro en la entidad DOCENTE, y en el campo DO_contrato, insertamos un valor de contrato que no existe en la entidad CONTRATOS, entonces se incurre en una inconsistencia en cuanto a la Integridad Referencial.

Otro caso es que podría suceder por ejemplo que el campo CON_contrato de la entidad CONTRATOS, que tiene un valor 3, se cambie por 5. ¿Qué va a suceder entonces con los Docentes de la entidad DOCENTES que tengan asignado el contrato 3? En este caso, se rompe con la integridad de la base de datos. De igual manera si ahora la modificación la realizamos en la entidad DOCENTES, ¿Qué sucedería ahora si a las tuplas o registros del campo DO_contrato que tienen valor de 3, le asignamos un valor de 6? Nuevamente se crea una inconsistencia, ya que el valor 6 que es clave foránea en otra entidad, no existe y crea así una inconsistencia en la Integridad de la base de datos.

También debe tenerse cuidado al momento de borrar registros. ¿Qué sucede por ejemplo si al igual que en el ejemplo anterior, en lugar de modificar las tuplas o registros de la entidad CONTRATOS se modifican aquellos registros que en la columna CON_contrato tienen valor de 3? esto, generaría una inconsistencia en la base de datos, ya que en la entidad DOCENTES hay tuplas o registros que tienen un valor de 3 en el campo DO_Contrato.

Tabla 1. Ejemplo de entidades DOCENTES Y CONTRATOS. Fuente propia.

DO_docente	DO_nombres	DO_ape_pat	DO_ape_mat	DO_fec_nac	DO_status	DO_fec_ing	DO_sexo	DO_contrato
500	Jose	Perez	Andrade	01/12/1973	A	01/12/2010	M	1
501	Maria	Lopez	Sánchez	04/06/1960	A	06/02/1996	F	2
502	Sonia	Martinez	Garcia	01/09/1980	A	02/04/1997	F	1
503	Juan Pedro	Robles	Gasca	28/07/1975	A	07/08/1997	M	
504	Jose	Gracia	Lopez	05/06/1975	A	08/09/2000	M	3
505	Alma	Martinez	Valenzuela	16/05/1973	A	03/05/2010	F	
506	Federico	Lopez	Rosales	27/01/1970	A	27/07/2000	M	2
507	Liliana	Carrillo	Gallegos	18/05/1965	A	04/05/2011	F	2
508	Jose Alfonso	Sanchez	Morales	09/03/1979	A	09/03/2013	M	2
509	Rosa Maria	Gomez	Sanchez	10/02/1981	A	05/01/2012	F	3

CON_contrato	CON_descripcion	CON_HrsTot	CON_HrsClaseLic	CON_HrsClasePost	CON_HrsActividad	CON_Hrs_Extra	CON_Hrs_Tope
1	Tiempo completo	40	40	0	0	0	40
2	Medio tiempo	20	20	0	0	0	20
3	Contrato 10 Hrs	5	0	0	5	0	5

2.5.4 Normalización.

Hasta ahora se ha tratado únicamente el sentido común del diseñador de BD al momento de la realización del diseño de una base de datos; sin embargo, esto puede no ser suficiente y puede surgir la necesidad de comprobar si existen anomalías en el diseño; esto puede lograrse a través de un método formal que indique cual es el esquema idóneo analizado desde una perspectiva de calidad; para esta situación, existe un concepto llamado normalización creado en 1972 por Edgar Frank Codd.

Una forma práctica de entender la esencia de la normalización es compararla a la tarea de analizar párrafos en la que tal como en la escuela, cuando los docentes explicaban a los alumnos que la forma correcta de separar los párrafos es hacerlo de tal forma que cada párrafo debe contener un solo tema; en caso contrario el párrafo deberá ser dividido en los párrafos necesarios cada uno de los cuales deberá contener un tema. De la misma forma se realiza en la normalización, donde cada relación normalizada debe manejar también un solo tema. En caso de que existan relaciones que manejen más de dos temas, deberán dividirse en dos o más relaciones cada una de las cuales deberá contener un solo tema. En caso que se presente una relación con anomalías de modificación, entonces se elimina esta anomalía al separar la relación en dos o más hasta que cada una contenga un tema. Se debe tener cuidado al momento de crear divisiones nuevas, debido a que esto generará restricciones de integridad referencial que se deberán verificar (Kroenke, 2003).

Codd se refiere a la normalización como el proceso de “pasar un esquema de relación por ciertas comprobaciones que certificarán que el esquema satisfaga determinada forma normal”; esto es, para el concepto de normalización se manejan reglas establecidas llamadas “forma normal” en las que las primeras tres formas normales son las más usadas, y si una base de datos cumple con la primera regla, se dice entonces que esa base de datos está en primera forma normal (1FN). Si se cumple con la segunda regla, está en segunda forma normal (2FN) y así sucesivamente hasta llegar a la tercera forma normal; estas tres formas normales son las únicas que fueron especificadas inicialmente por Edgar Frank Codd y se basan en la dependencia funcional entre los atributos de una relación. Existen más reglas, pero esas son las más usadas y por lo general, si una base de datos está en tercera forma normal, cumple con el máximo nivel necesario requerido en la mayor parte de las aplicaciones; la cuarta (4FN) y quinta forma normal (5FN) se basan en la dependencia multivalor y dependencia de concatenación.

Con base en lo anterior, se puede definir la normalización de datos “como el proceso por el cual se analiza una relación, basándose en sus dependencias funcionales y sus claves principales de tal forma que se obtengan las propiedades deseables y así minimizar las redundancias, anomalías de inserción, borrado y actualización. Los esquemas de relación que no cumplen con las pruebas de forma normal se descomponen en esquemas de relación más pequeños hasta que cumplan con las pruebas de forma normal”. (Elmazri & Navathe, 2007)

Una base de datos que no está normalizada, podrían generar problemas de espacio en disco, de mantenimiento así como anomalías de modificación. Las anomalías de modificación se refieren al hecho de que los datos que existen en más de una ubicación tendrían que cambiarse en todas sus ubicaciones; la normalización requiere por lo general agregar entidades adicionales, lo que desde el punto de vista práctico puede resultar engorroso, aunque debe considerarse el hecho de su importancia para la salud de la BD. Por el contrario, una base de datos normalizada ofrece los siguientes beneficios (Maza A., 2003):

- Se eliminan los datos redundantes
- Puesto que se reducen las anomalías de mantenimiento, esto trae como consecuencia una simplificación del mantenimiento de los datos al momento de actualizar, insertar y borrar.

- Se reduce la necesidad de reorganizar los datos en el momento que surjan nuevos cambios o aplicaciones.
- Debido a que se busca eliminar el hecho de tener una BD grande, por consecuencia se reduce el tamaño de la BD.
- Los datos se podrán recuperar de una forma más sencilla.

- **Primera forma Normal.**

Para que una base de datos se encuentre en primera forma normal, se debe llegar a lo siguiente:

- Eliminarse los grupos repetidos de las tablas individuales lo que se conoce como atributos atómicos.
- Debe crearse una entidad independiente para cada conjunto de datos relacionados
- El conjunto de datos relacionados debe identificarse con una clave principal.

Se recomienda no usar campos en una sola entidad para identificar campos parecidos. Por ejemplo, si se requiere llevar el seguimiento de los teléfonos de los clientes, no se recomienda especificar estos como telefono1, teléfono 2, telefono3, etc. debido a que si se requiere en un futuro almacenar un campo adicional de este rubro, es decir un cuarto teléfono, se tendrían que modificar los registros de la entidad, sin mencionar los cambios que tendrían que realizarse en la aplicación; entonces lo más recomendable en caso de que la posibilidad de agregar más teléfonos sea elevada, agregar una entidad separada donde se almacenen los teléfonos y vincularlos estos a través el campo cliente que representara el campo clave entre las entidades teléfonos y clientes. (Microsoft, 07)

- **Segunda forma Normal.**

Para que estar en segunda forma normal (2FN), en primera instancia debe estar primero en primera forma normal. Si todos sus atributos que no son claves dependen por completo de la clave, no puede haber dependencias parciales.

- **Tercera forma normal.**

Se encuentra en un estado de tercera forma normal si está en segunda forma normal y no tiene dependencias transitivas entre los atributos; esto significa que no debe existir más de una forma de hacer referencia a un atributo de una entidad. Tercera forma normal implica deshacerse de cualquier elemento de las entidades que no dependa únicamente de la llave primaria, y de esta forma se elimina la dependencia transitiva que queda en una segunda forma normal. (Juárez S., 2009)

2.6 Diccionario de Datos.

El diccionario de datos (Codina, Metodología de análisis de sistemas de información y de diseño de bases de datos documentales: aspectos lógicos y funcionales, 1998) es un catálogo que contiene la descripción detallada de todos los elementos de un sistema donde se especifican los datos y la forma en que están estructurados; en este se encuentran todos los elementos que conforman el flujo de datos de un sistema completo como la descripción detallada de todas las entidades con cada uno de sus atributos y sus características como el nombre, alias, longitud del dato, valores de los datos, tipo de dato.

El diccionario de datos representa una herramienta efectiva para el diseñador de base de datos, y ayuda a garantizar la “calidad, fiabilidad, consistencia y coherencia de la información alimentada”; por cada elemento de los datos, el diccionario contempla al menos:

- Etiqueta de dominio.
- Tipo de dato.
- Indexación.
- Tratamiento documental.
- Lengua.
- Observaciones.
- Ejemplos válidos.
- Otros controles o especificaciones de acuerdo al tipo de campo.

2.7 Metodologías para Desarrollo de software.

En los proyectos que se desarrollan en la actualidad se tiene una inclinación a enfatizar los esfuerzos de la etapa de codificación y se resta importancia a las áreas de análisis, diseño, planeación, pruebas, y documentación, lo que puede resultar en proyectos con tiempos extralimitados, costos elevados y clientes insatisfechos; en este marco, pueden surgir problemas al momento corregir o modificar, originados por la falta de planeación, análisis y metodologías adecuadas para las etapas del desarrollo mencionadas.

Por esa razón debe realizarse un esfuerzo extra en el proceso de desarrollo completo, no solamente en una de las fases, para que de esta manera el producto final sea de calidad tanto en su proceso de realización, como en el producto resultante; se recomienda utilizar metodologías y herramientas de vanguardia que conducen de forma casi automática a entregar un producto de calidad. La calidad de un producto de software puede visualizarse desde diferentes ángulos: la satisfacción de los requerimientos del cliente final, la calidad desde el punto de vista del software (metodologías utilizadas eficientemente, tiempos de entrega) y la calidad desde el punto de vista del usuario final.

Usar una metodología puede evitar dentro de lo posible clientes insatisfechos con los resultados y retrasos en la realización del trabajo; esto no es garantía de éxito, pero si garantía de que la realización de las tareas durante el ciclo de vida del proyecto se llevará a cabo dentro de un marco ordenado y de calidad.

Dentro de la Ingeniería de software, existen marcos de trabajo enfocados a estructurar, planificar y controlar el proceso de un desarrollo de software, lo que conforma una metodología de la ingeniería de software. Entre las metodologías que existen y que pueden ser usadas para cualquier etapa del proyecto, se encuentra el modelo en cascada, modelo evolutivo, desarrollo en espiral, metodología RUP, y las metodologías ágiles extreme programming y scrum.

Las metodologías están divididas en tradicionales y ágiles, cada una con sus ventajas y desventajas. Se maneja también la combinación de ambas, las llamadas metodologías híbridas. Entre las metodologías tradicionales se pueden mencionar métrica 3, proceso unificado racional (RUP), OMT técnica de modelado de objetos (OMT) entre otras. Un aspecto relevante de las metodologías tradicionales es la documentación de sistemas enfocado a producir software de mantenimiento eficiente y de fácil comprensión; sin embargo, esta característica también puede llegar a representar una desventaja si la

cantidad de documentación generada es excesiva o innecesaria. Otra desventaja de esta metodología es el alto grado de disciplina requerido, la rigidez a los cambios, el tiempo invertido en el modelado, y dejar de lado el hecho de que en el desarrollo de software a diferencia de ser exacto, es impredecible.

Aunque las metodologías tradicionales son efectivas para proyectos grandes, existen algunos proyectos que manejan una dinámica cambiante en la que el mismo proyecto pide reducir los tiempos de visualización de resultados. En estos casos, surgen las metodologías ágiles, que tienen un enfoque orientado a la satisfacción del cliente y la entrega temprana de software, con equipos de trabajo por lo general pequeños, con una metodología de desarrollo simplificado y flexible. Entre las metodologías ágiles pueden mencionarse programación extrema o extreme programming (XP), Scrum, DSA & development software adaptative, o desarrollo de software adaptativo.

Tanto las metodologías tradicionales como las ágiles tienen sus ventajas y desventajas; sin embargo, hay situaciones en las que es necesario hacer uso de la combinación de ambas, lo que da como resultado el surgimiento de las metodologías híbridas que retoman las ventajas tanto de las metodologías tradicionales como las ágiles hasta conformar una combinación de ambas. Existen metodologías establecidas de manera formal, como la EssUP (essential unified process, proceso esencial unificado), considerada como la primera metodología considerada híbrida (Jiménez H., 2012).

2.7.1 Metodología RUP.

RUP se ha convertido en una de las metodologías más utilizadas en el análisis, implementación y documentación de sistemas orientados a objetos que puede aplicarse tanto en pequeños proyectos como en grandes empresas, aunque son estas últimas las que más las utilizan por el tamaño y la complejidad de los sistemas que manejan.

El proceso de desarrollo de software llamado Proceso Unificado de Rational o Rational Unified Process (RUP) por sus siglas en inglés está conformado por un conjunto de actividades orientadas a transformar los requisitos especificados por el usuario en un sistema de software. RUP utiliza el Lenguaje Unificado de Modelado o Unified modeling Lenguaje (UML) por sus siglas en inglés, por medio del cual se representan los esquemas del sistema de software (Arias, Carrero, Grimón, & Morales, 2006).

- **Mejores prácticas.**

Para realizar el modelado de sistemas, reducir la complejidad y el diseño de procesos, se han llevado a cabo estudios de los que han surgido un conjunto de mejores prácticas de ingeniería que representan RUP. Las seis mejores prácticas que RUP propone son: administración de requerimientos, desarrollo iterativo, basado en el modelado visual, realización de pruebas continuas, arquitectura basadas en componentes y control de cambios (Flórez F., 2009). En un proyecto de software, la identificación de requerimientos es sumamente importante debido a la cantidad de recursos económicos y técnicos que están en juego; por eso RUP destaca la identificación de requerimientos y enfatiza que esto debe realizarse de forma eficiente, sin dejar lugar a dudas en este punto y si se parte del hecho de que los requisitos pueden cambiar; dentro de este contexto RUP propone la administración de requerimientos como una de sus mejores prácticas. En cuanto al desarrollo de software iterativo, puesto que el significado de la palabra iterativo es repetición o reiteración (Real Academia Española, 2012); para RUP esto se traduce en que el proceso se repetirá cierto número de veces, y cada iteración arrojará una nueva versión. Las iteraciones también permiten identificar nuevos riesgos o posibles cambios que se pueden atacar a tiempo. La figura 1 explica de forma gráfica las iteraciones de RUP.

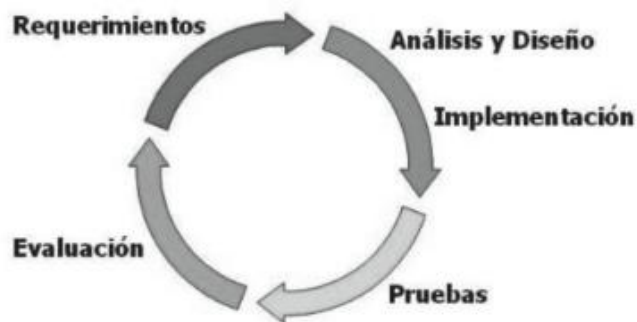


Figura 1. Desarrollo iterativo. (Flórez F., 2009)

La complejidad del software se puede administrar a través del modelado visual, práctica que ayudará al equipo a conceptualizar el proceso que se trata de trasladar a un sistema y facilitar su comprensión, visualización, documentación del comportamiento y la estructura

de la arquitectura de un sistema de software. Si a esto se agrega un lenguaje estándar de modelado, que sea usado no solamente en ese proyecto, sino de forma universal, esto facilitará la comprensión del modelo; es por ello que una de las mejores prácticas que propone RUP es el modelado de software de forma visual, dirigido por casos de uso a través de la utilización de UML, que servirá como guía durante todo el proceso, como se muestra en la figura 2. Otra de las mejores prácticas se refiere a la verificación continua de la calidad del software que se logra a través de revisiones y pruebas realizadas de acuerdo a las iteraciones del proyecto antes de tener el sistema en producción. Esto facilita la detección de errores a tiempo lo que dará margen para poder atacarlos de acuerdo a las iteraciones. Los aspectos que deben cuidarse en las verificaciones del software son la calidad en cuanto a funcionalidad, confiabilidad y performance. Estas verificaciones darán lugar a evaluaciones de software cuantitativas y continuas (Flórez F., 2009).



Figura 2. Los casos de uso a través de todas las fases. (Belloso, 2009)

- **Organización por componentes.**

RUP organiza las actividades de desarrollo de acuerdo a los criterios que muestra la figura 3, en la que se pueden visualizar dos dimensiones: en el eje vertical las actividades básicas o componentes, que especifican el avance en el desarrollo del sistema. En el eje horizontal los criterios utilizados para la planeación y control del tiempo que especifican cuándo se realizarán las actividades especificadas en el eje vertical. El nivel de intensidad de las actividades de los componentes varía en función de la organización en el tiempo (fases). En la figura 3 se representa a través de colores; de esta forma, se observa que la captura de requisitos es una actividad que inicia en la etapa de gestación, donde alcanza su nivel máximo, y aun se realiza en la fase de preparación, pero es una actividad que casi no se realiza en la fase de construcción y transición. Las pruebas empiezan actividad en la fase

de preparación, y tienen actividad similar en el resto de las fases, pero es en la fase de construcción donde alcanza su máxima actividad y disminuye a partir de esta.

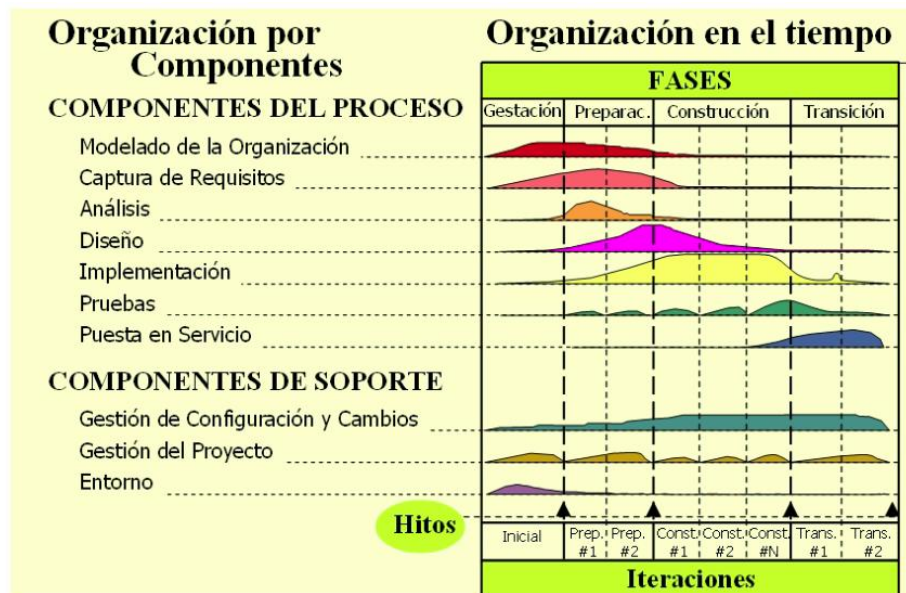


Figura 3. Organización de RUP (Rendón G., 2004).

Para entender las actividades de los componentes es importante conocer los términos en que estos se describen: artefacto se refiere a la información que se genera, modifica o utiliza en el desarrollo. Trabajadores son los roles que intervienen en los componentes. Flujos de trabajo y actividades son los que son llevados a cabo por las personas que desempeñan algún rol hasta obtener los artefactos propios del componente. Una vez entendidos estos términos es importante conocer los dos tipos de componentes que maneja RUP, como muestra la figura 3: componentes del proceso de ingeniería que son los relacionados con la obtención del producto y componentes de soporte que son las actividades administrativas del proceso.

Los componentes del proceso de ingeniería son los siguientes (Belloso, 2009), (Rendón G., 2004):

- **Modelado de la organización.** Es necesario identificar y conocer la organización o parte de ella sobre la cual se implementará el sistema. El personal involucrado debe entender de forma clara cuál es el problema de la organización y este deberá quedar claramente identificado y documentado.

- **Análisis.** Se realizará la definición de la estructura del sistema así como su comportamiento con el propósito de conocer su funcionamiento.
- **Captura de requisitos.** Durante esta etapa se trata de conseguir un listado de requisitos de usuario lo más completo posible y expresarlo en términos formales. Se consideran los requisitos dentro del siguiente contexto: descripciones, lista de requisitos y escenarios. Descripciones se refiere las descripciones de los problemas a resolver que los usuarios de forma frecuente hacen y en los que es muy común que se escondan requisitos que es necesario identificar. Una vez especificados, se prepara una lista de requisitos que se extraen de las descripciones mencionadas previamente o de lo que haya expresado el usuario de forma verbal o se hayan identificado de alguna otra forma. Los escenarios representan la descripción de situaciones que permiten describir el funcionamiento del sistema y facilitarán la comprensión del mismo; deberán identificarse si no todos, al menos los más importantes.
- **Diseño.** En esta etapa se define la arquitectura del sistema y se realiza el modelado que soportará todos los requisitos. Tiene como propósito adquirir una comprensión sobre los requisitos no funcionales. Los productos desarrollados en esta etapa son:
 - Modelo del diseño, incluido el diagrama de clases, informes y la realización de los casos de uso.
 - Modelo de interfaz de usuario, incluidos los bocetos del interfaz de usuario y el mapa de navegación storyboard.
 - Informe del modelo de interfaz de usuario. Modelo de datos e informe.
 - Arquitectura del software.
 - Informe de evaluación.
 - Casos de prueba.
- **Implementación.** Es la preparación de archivos necesarios para que el sistema funcione de forma adecuada.
- **Pruebas.** En esta etapa se realiza la verificación de los artefactos obtenidos, a fin de garantizar la calidad del software; debe realizarse durante todo el ciclo de vida, incluso desde los prototipos. Con esto se logra el principal beneficio de la prueba: la retroalimentación todavía con tiempo y recursos para realizar correcciones. En esta etapa se busca verificar la interacción e integración de componentes, verificar que los requisitos sean implementados de forma correcta, se identifica que los defectos encontrados hayan sido corregidos antes de la entrega y también se realizan pruebas para tratar de visualizar errores aun no detectados.

- **Puesta en servicio.** Se refiere a la realización de las actividades inherentes a la puesta en marcha del producto terminado en producción.

Los componentes de soporte son los siguientes:

- **Gestión de configuración y cambios.** Se refiere al control sobre los cambios y configuraciones de los prototipos generados, que son gestionados y documentados de manera formal.
- **Gestión de proyecto.** Se refiere a la administración del proyecto, que marca la pauta para la planificación, establecimiento de equipos de trabajo, ejecución y supervisión de proyectos y sus riesgos.
- **Entorno.** Para el proceso de desarrollo, se requiere un entorno adecuado, con las herramientas y los procesos necesarios, que deben ser proporcionados al personal involucrado en el proyecto.

- **Organización en el tiempo.**

El proceso de desarrollo se especifica en términos de ciclos, fases, iteraciones e hitos (milestones). Un ciclo se refiere al desarrollo de una nueva versión del producto. Las fases son etapas del desarrollo de una versión donde cada fase termina con un hito que puede estar dividida en varias iteraciones. Hito es un punto de control donde por lo general existe una revisión de resultados y se decide o no avanzar.

Las fases de RUP son (gestación, preparación, construcción y transición), las cuales pueden dividirse en iteraciones. Las iteraciones están divididas en disciplinas o flujos de trabajo (modelado de la organización, captura de requisitos, análisis, diseño, etc.)

Los flujos de trabajo producen modelos compuestos por artefactos (casos de uso, modelos de prueba, etc.). Un grupo de artefactos conforman un Hito.

La organización en el tiempo que realiza RUP, está especificado a nivel de fases; la tabla 2 muestra estas fases de forma gráfica en el eje horizontal: gestación, preparación, construcción y transición, a continuación se explica la función de cada fase de forma detallada (Rendón G., 2004):

Gestación. En esta fase se realiza el análisis, para lo cual se requiere determinar el alcance del proyecto y tener una propuesta de la arquitectura que se utilizará; incluye la identificación de actores, definición de riesgos y costos, mediante el estudio de factibilidad. Se trabaja en la identificación de casos de uso y se realiza la descripción de los más importantes. Se realiza la planeación de fases y la especificación de las fechas de los hitos principales. Componentes con los que se trabaja: modelado de la organización y captura de requisitos. A partir de lo anterior las personas a cargo del proyecto toman la decisión de la viabilidad del proyecto.

Preparación. Esta fase es la más crítica, donde el hito resultante es la definición de la arquitectura del sistema, los objetivos, el alcance y los riesgos. Durante esta etapa todavía se puede tomar la decisión de cancelar o replantear el proyecto; se construye un prototipo ejecutable de la arquitectura que puede realizarse en una o más iteraciones en base a la dificultad del proyecto donde el esfuerzo principal debe enfocarse a los casos de uso críticos que fueron identificados en la fase anterior, aunque no se excluye la posibilidad de enfocar el esfuerzo hacia cualquier otro componente para mitigar riesgos, realizar demostraciones a inversores o personal involucrado en el proyecto.

Construcción. Durante esta fase ya se cuenta con el producto de software en una versión inicial o versión que será puesto a disposición de los usuarios finales y que debe incluir la realización de pruebas. RUP se enfoca mucho en el esfuerzo que debe realizarse en la etapa de planeación, lo que conduce a que la dificultad esté mejor balanceada entre la etapa de planeación y construcción, debido primordialmente al esfuerzo previo invertido en la planeación.

Transición. En esta fase se realiza la entrega al usuario final y aún es válido el surgimiento de detalles que deberán corregirse. La entrega debe incluir una versión lo suficientemente madura, con la documentación correspondiente, realización de pruebas beta, implementación del sistema en paralelo con el sistema anterior si este llegara a existir; se realiza la capacitación a usuarios y administradores y una presentación formal a los stakeholders.

La tabla 2 muestra un resumen sobre las arquitecturas e hitos resultantes de cada fase.

Tabla 2. Artefactos e Hitos de las fases de RUP. Fuente: **(Rendón G., 2004)**

FASE	ARTEFACTOS	HITO
Gestación	Boceto inicial de la arquitectura, descripción de los objetivos del proyecto, requerimientos (planteados como casos de uso), modelo de negocio, versión muy preliminar del plan del proyecto.	Hito de objetivos del ciclo de vida.
Preparación	Casos de prueba, mayoría de los casos de uso, software como prototipo, plan detallado para las próximas iteraciones.	Hito de arquitectura de ciclo de vida.
Construcción	Casos de prueba, manuales de usuario, sistema de software	Hito de capacidad operativa inicial.
Transición	Casos de prueba, manuales de usuario, sistema de software	Hito de lanzamiento del producto

2.7.2 Metodología Scrum

A mediados de los ochenta, Hirotaka Takeuchi e Ikujiro Nonaka definieron un enfoque innovador para desarrollo de productos que tenía la característica de ser flexible e incluyente en la que el equipo de desarrollo estaba enfocado a un objetivo común. La propuesta, basada en el estudio realizado a diversas industrias de fabricación hace énfasis en que los equipos no deben manejarse de forma secuencial, sino trabajar en conjunto tal como se trabaja en el rugby, específicamente un concepto deportivo de rugby llamado “scrum”, donde el grupo de jugadores se junta y reinicia el juego.

Ken Schwaber y Jeff Sutherland toman esta propuesta y evalúan su funcionamiento en el desarrollo de software y es a partir de ese momento que scrum sigue un perfeccionamiento a través diferentes contribuciones de practicantes, expertos y autores, hasta llegar a convertirse en una de las metodologías ágiles más populares en la actualidad y preferida por diversas organizaciones a nivel mundial. Las organizaciones que aplican scrum en sus proyectos, obtienen altos rendimientos en su inversión.

La fortaleza de scrum se basa en usar equipos con poder que se auto organizan y son multifuncionales. Está orientado a proyectos de cualquier tamaño y grado de complejidad desde equipos (scrum teams) con seis personas, hasta proyectos conformados por cientos de miembros. Para trabajar de forma eficaz, los scrum teams ideales tienen de 6 a diez personas; entonces se cree de forma errónea que scrum no podrá trabajar con equipos

grandes conformados por cientos de personas, sin embargo, en casos de grandes proyectos, la organización se segmenta en pequeños equipos llamados scrum teams que son gestionados a través de un proceso llamado scrum of scrum. Scrum también se puede aplicar a los programas o proyectos de cualquier sector (portfolios); y a los productos o servicios (entregables).

Entre las ventajas más importantes al momento de seleccionar la metodología scrum, se pueden mencionar las siguientes:

- Los proyectos son adaptables y flexibles a los cambios que se presentan.
- La información del proyecto es manejada de forma compartida y transparente con una retroalimentación continua entre los miembros del equipo que tienen la característica de ser multifuncionales y estar calificados para alcanzar resultados en alta velocidad.
- A diferencia de otras metodologías, la responsabilidad recae en el equipo completo, no existe un solo individuo como responsable.
- Al formar parte de las metodologías ágiles una de sus características es agregar valor desde el inicio del proyecto lo que sucede de forma continua a lo largo del mismo, que conduce a una mejora continua por medio de entregables efectivos basados en los requerimientos del cliente (customer). En lo referente a la entrega anticipada de alto valor, los requisitos que tienen mayor valor para el customer son los primeros que se cubren.
- Resolución de problemas de forma rápida.
- Es en el customer en el que scrum pone especial énfasis, por eso se dice que scrum está centrado en el cliente.

La diferencia básica entre scrum y la gestión tradicional de proyectos reside en el hecho de que en la gestión tradicional el plan se realiza desde el principio e incluye tiempos, riesgos y costos; sin embargo, en la realidad se sabe que aunque un plan esté realizado, esto no garantiza un cliente satisfecho, como sucede de forma frecuente en los proyectos, donde el customer por lo general no revisa hasta el final de una fase o del proyecto. En cambio, scrum entrega valor al proyecto desde el principio, con lo que se asegura el cumplimiento de los requerimientos que marca el customer así como revisiones desde el principio, lo que se realizan de forma iterativa a lo largo del proyecto donde no existe un plan rígido, por el contrario los proyectos están abiertos al cambio. En lo referente al equipo de trabajo, scrum

considera que las personas son capaces de aportar al proyecto algo más que conocimientos técnicos y les brinda la oportunidad de realizar aportaciones mediante equipos multifuncionales. La tabla 3 muestra algunas diferencias importantes entre scrum y la gestión de proyectos tradicional.

Tabla 3. Comparativo entre scrum y la gestión de proyectos tradicional. Fuente (Navarro C., Fernández M., & Morales V., 2013)

Metodologías tradicionales	Metodologías ágiles
Predictivos.	Adaptativos.
Orientados a procesos.	Orientados a personas.
Proceso rígido.	Proceso flexible.
Se concibe como un proyecto.	Un proyecto subdividido en varios proyectos más pequeños.
Poca comunicación con el cliente.	Comunicación constante con el cliente.
Entrega de software al finalizar el desarrollo.	Entregas constantes de software.
Documentación extensa.	Poca documentación.

Scrum fomenta el trabajo en equipos que se auto organizan como mejor se adapten, donde existe una comunicación frente a frente, no hay roles preestablecidos ni se manejan jerarquías; todo el equipo participa y aporta su conocimiento individual, de forma general al proyecto; si un miembro del equipo tiene experiencia en diversas aéreas, puede aportar a las mismas. Está prohibido ocultar detalles en cuanto a la implementación del cliente, ya que este está perfectamente informado y está presente en las reuniones. En este marco de trabajo la constante es la lucha contra la jerarquía de personas, departamentos, negociaciones de precio, fechas o requisitos demasiado cerrados (Satpathy, 2013).

- **Roles de trabajo.**

Los roles manejados en esta metodología, se centra en dos principales: funciones básicas (core roles) y funciones no básicas (non-core roles).

Los core roles, son roles obligatorios dentro de un proyecto que maneja scrum, y se conforma de las personas responsables del proyecto. Dentro de los core roles se incluye al product owner, persona considerada el representante del customer, encargado que se

realicen los requisitos indicados por el customer; también es encargado del documento business justification. El scrum master se encarga de enseñar las prácticas de scrum a las personas que trabajan en el proyecto y de facilitar que las actividades del proyecto se realicen bajo un ambiente adecuado, de eliminar los obstáculos que se lleguen a presentar (impediments). El scrum team es el equipo de trabajo que se encarga de que los requisitos que especificó el product owner se cumplan a través de los entregables.

Los non-core roles se refieren a las personas que pueden participar directa o indirectamente en el proyecto pero el éxito del mismo no depende de ellos. Entre los non-core roles, están los stakeholders que comprende las personas que tienen influencia en el proyecto y aunque no son miembros formales, se producen beneficios para ellos, como los customers, usuarios y patrocinadores. Los scrum guidance body (SGB) también realizan una actividad del tipo non-core roles y su función es opcional; comprende las personas y documentos expertos involucrados con la calidad, regulaciones gubernamentales, seguridad y otros aspectos relacionados con la organización. Los vendedores (vendors) son las personas u organizaciones que ofrecen productos o servicios que no entran dentro de la organización del proyecto. Los chief product owner son los encargados de la organización de varios scrum teams en los proyectos grandes al apoyar a los product owners, así como el mantenimiento del documento llamado business justification. El chief scrum master, en proyectos grandes, facilita la organización de varios scrum teams que participan. La figura 4 representa un resumen del flujo de actividades de algunos de los roles mencionados en los párrafos anteriores (Satpathy, 2013).



- **Artefactos.**

Los artefactos de scrum son los subproductos resultantes de las actividades de scrum, y sirven de apoyo para el equipo de trabajo al facilitarle dirección y transparencia. Los artefactos que se manejan son product backlog, sprint backlog, Monitoreo de progreso e incremento (Barrios, et al., 2012).

Como se mencionó en párrafos anteriores, product backlog es una lista definida por el customer y ordenada por prioridades; este orden es de acuerdo al valor que el requisito representa para el negocio y también de acuerdo al esfuerzo de desarrollo que implica de acuerdo al product owner. Product backlog especifica todos los requerimientos y funcionalidades solicitadas y cambia constantemente a lo largo del proyecto, entonces se puede decir que nunca está terminada. Por lo general se usa el término user story para referirse a los requerimientos y funcionalidades del product backlog.

El sprint backlog es un subconjunto de ítems, derivados del product backlog y el plan que se llevará a cabo para el incremento. Se construye con los requerimientos más prioritarios y con aquellos que quedaron pendientes del sprint anterior. El sprint backlog deberá ser aceptado por el equipo y solo el equipo podrá modificarlo. Si se presentan nuevos requerimientos, estos serán incluidos en el product backlog y se desarrollarán en otro sprint de acuerdo a la prioridad que se le asigne. Sprint backlog describe en detalle cómo se implementarán los requerimientos durante el próximo sprint. Las tareas tienen una duración máxima de 16 horas y el equipo se pone de acuerdo en la forma de realizar éstas.

El monitoreo de progreso, también denominado sprint burndown chart, es una gráfica que muestra los requerimientos que faltan por terminar, y el progreso realizado por el equipo, expresado en horas. Este se actualiza al final del día, lo que da al product owner la oportunidad de evaluar el progreso en el momento que lo requiera. Si la gráfica muestra que el equipo no podrá terminar todas las áreas del sprint, el scrum master deberá identificar el motivo y los obstáculos para resolverlos y terminar el sprint en tiempo, (Satpathy, 2013).

El incremento se refiere a todos los ítems terminados en el sprint backlog. Los ítems incompletos se van al product backlog y se les asigna una prioridad alta para que se puedan concluir en el siguiente sprint. Para que un ítem se considere terminado, deberá ser funcional. La suma de todos los ítems terminados conforma el product que se le entregará al customer.

- **Metodología de trabajo.**

Scrum se caracteriza por el desarrollo iterativo e incremental, a través de ciclos de trabajo denominados sprint, los cuales siempre tendrán la misma duración, se haya concluido o no el trabajo. Se maneja una lista de productos por terminar (product backlog) ordenados por prioridad que son definidas por el cliente (producto owner).

El sprint, que en scrum equivale a las iteraciones, está organizado por periodos de tiempo que tienen una duración de una a cuatro semanas consecutivas. Un sprint está conformado por reunión de planeación del sprint, daily scrum, trabajo de desarrollo, revisión del sprint y retrospectiva del sprint.

El sprint inicia con una reunión del equipo de trabajo (scrum team) para realizar la planeación del sprint; esta reunión tiene una duración de ocho horas donde se define el plan de trabajo que se realizará para terminar el product backlog que se traduce en una incremento que producirá una versión utilizable del producto que el equipo se comprometió a terminar; en la reunión se especifica la forma en que se realizará y se entregará el trabajo. No se permiten cambios en los elementos elegidos una vez iniciado el sprint.

Dentro de cada sprint el equipo de trabajo realiza una reunión diaria denominada daily sprint con una duración de quince minutos donde se definen los obstáculos con las que el equipo de trabajo se ha encontrado, cuales son los avances que se han realizado conforme a la última reunión y las actividades que se realizarán. Los participantes se reúnen de forma rápida, en forma de círculo y de pie, para evitar que esta reunión se extienda más allá del tiempo establecido.

Al final de cada sprint este se revisa lo que se extiende cuatro horas para un proyecto de un mes, o proporcional si la duración no fuera de cuatro semanas. El product owner revisa que se realizó y que no, los problemas encontrados, la forma en que se resolvieron y se muestra la aplicación en funcionamiento.

La retrospectiva del sprint es una reunión en la que se analizan los diferentes elementos del equipo: las herramientas, la comunicación, los problemas encontrados, el proceso y se define el plan de mejoras para el próximo sprint. La duración es de tres horas para un sprint de cuatro semanas, o el proporcional si el sprint duró menos (Navarro C., Fernández M., & Morales V., 2013).

La figura 5 muestra la forma en que se trabaja en scrum, de acuerdo a lo descrito en los párrafos anteriores y una comparación con otras metodologías.

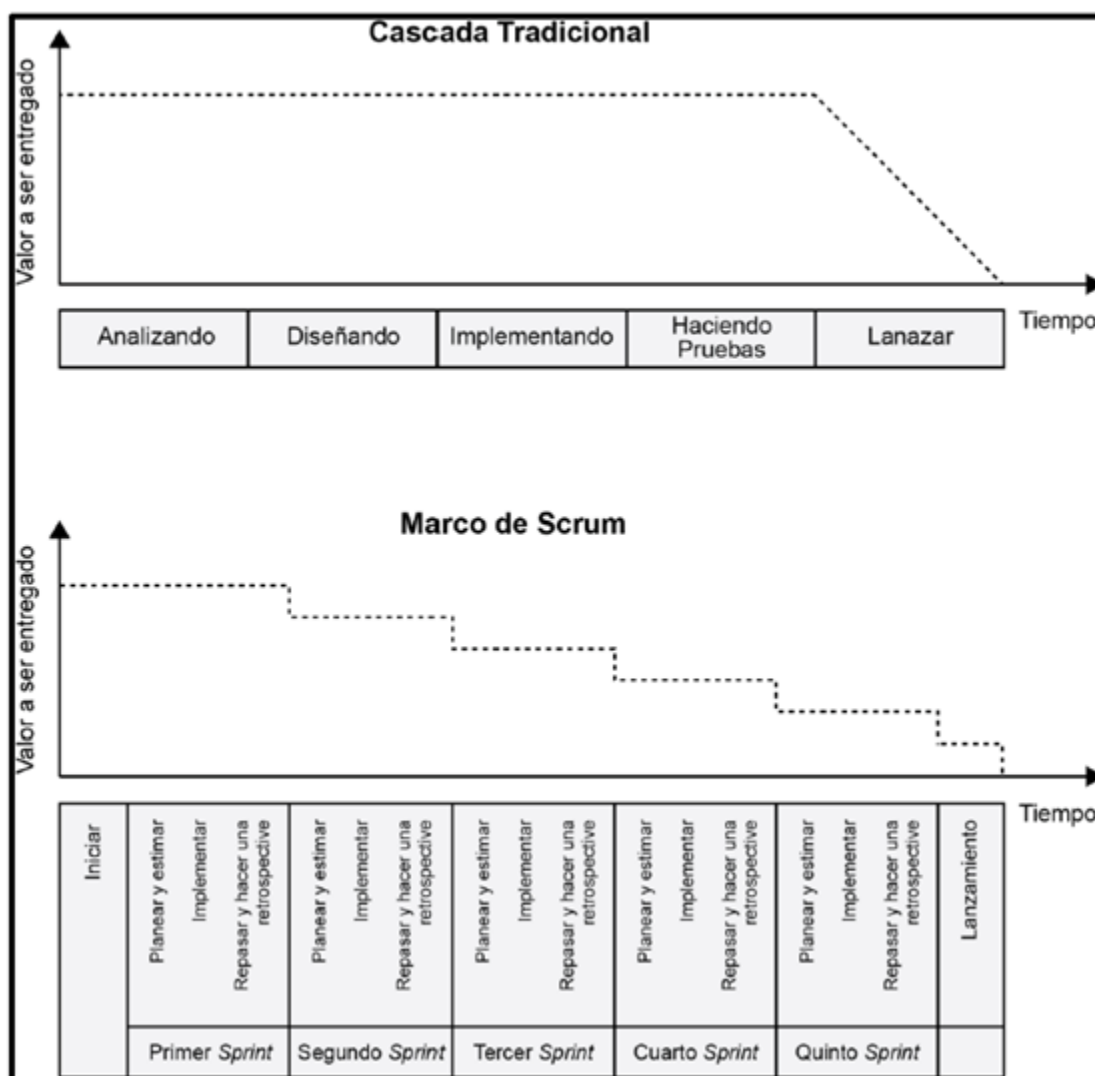


Figura 5. Forma de trabajo de Scrum, y una comparación respecto a la forma tradicional. Fuente (Satpathy, 2013)

- **Fases de scrum.**

El ciclo de vida de scrum está conformado por planeación, puesta en escena, desarrollo y entrega. En la planeación se especifica lo que se realizará, el equipo, las herramientas, se especifican los requerimientos priorizados y se estima el esfuerzo necesario para la realización de las tareas. En la puesta en escena se identifican más requerimientos y se realiza la priorización de los mismos. La implementación representa la parte ágil y aquí se realiza el desarrollo en sprints, donde cada sprint contiene los estados tradicionales de desarrollo de software: requerimientos, análisis, diseño, implementación y entrega. (Barrios, et al., 2012). Scrum es ágil, pero con eso no es suficiente; como cualquier actividad, implica dedicar tiempo y esfuerzo.

2.8 Gestores de bases de datos.

Un SGBD almacena datos interrelacionados y está compuesto por un conjunto de programas para gestionar los datos. Mediante un SGBD, se tiene un entorno que da la facilidad de gestionar la información de la base de datos: Almacenar y Extraer información, permite que los datos tengan un nivel de seguridad de la información que almacenan, esto significa que no cualquiera tendrá acceso a la base de datos, debido al control de seguridad que ejercen. Un SGBD también permite mantener la base de datos con integridad, disponer de mecanismos de recuperación, almacena los datos de forma muy diferente a como la ven los usuarios, pero igualmente eficiente para los administradores de la base de datos, revisa que las transacciones que se realizan cumplan con la integridad de la base de datos, debe permitir el procesamiento, inserción, recuperación y actualización de los datos. Para el presente proyecto, se usará MySQL.

2.8.1 MySQL.

Es uno de los gestores de base de datos más utilizados que tiene la característica de ser rápido, sólido y flexible. Tiene la mayoría de las funcionalidades principales para el almacenamiento de los datos, con la posibilidad de realizar múltiples y rápidas consultas.

Hay ciertas operaciones complejas que MySQL no permite realizar, pero las funcionalidades más importantes y necesarias, se pueden realizar con éste gestor de DB de software libre.

2.9 Diseño físico.

Para llegar al diseño físico se pasa de la estructura lógica de la base de datos a la estructura física con base en el SGBD elegido. El objetivo general del diseño físico es determinar estructuras de almacenamiento y estructuras de acceso para que las aplicaciones que accedan a la base de datos obtengan un buen rendimiento.

Cada SGBD ofrece varias opciones de organización de archivos y caminos de acceso: Diferentes tipos de índices, agrupamiento de registros (de distinto tipo) relacionados, en los mismos bloques de disco.

El diseño físico consiste en elegir las estructuras más apropiadas para los archivos de la base de datos de entre las opciones que ofrece el SGBD, y según el uso que se dará a los datos. Es dependiente del SGBD comercial seleccionado.

Un esquema lógico de base de datos puede tener diferentes esquemas físicos posibles en cierto SGBD. ¿Cuál es el más apropiado? Para ello se recomienda analizar:

1. Consultas y transacciones que se espera ejecutar
2. Frecuencia esperada de consultas y transacciones
3. Restricciones de tiempo sobre consultas y transacciones
4. Frecuencia esperada de operaciones de actualización
5. Restricciones de unicidad sobre los atributos

Capítulo III

Metodología

El diseño de la BD del presente trabajo, representa un subproyecto del proyecto principal “desarrollo del sistema SIGAF” y como se mencionó en el capítulo uno, para llevarlo a cabo se conformará un equipo en el que están definidas y delimitadas las áreas de trabajo del desarrollo de sistemas de información donde uno de los puntos es el diseño de la base de datos. A cada área le corresponde un actor responsable de las actividades y los artefactos correspondientes a ese rol.

Se manejarán reuniones periódicas, no estrictamente diarias, como lo especifica la metodología scrum; estas se realizan de acuerdo los requerimientos propios del desarrollo del trabajo, siempre que una de las áreas requiera comunicación por parte del resto para revisar algún aspecto del sistema.

Para una mejor organización del proyecto, éste se dividió en áreas, a las cuales se les llamará módulos, cada uno de los cuales engloba información con características similares. Esto podría entenderse como una vista de la base de datos, aunque en el presente caso, la información corresponde a la misma área de interés, en este caso, el proceso de horarios. Los módulos resultantes, son los siguientes:

1. Plan de estudio.
2. Carga académica.
3. Disponibilidad docente.
4. Creación de horarios.
5. Nivel de usuarios.

3.1 Modelado de una base de datos.

En ocasiones, se requiere un conocimiento más profundo respecto al dominio de un problema a fin de responder a las necesidades de información que se tienen del mismo, y que tienen una finalidad fijada con anterioridad; es entonces que se busca la adquisición de un conocimiento más profundo de dicho dominio. Esto se puede lograr mediante la realización de un modelo que se alcanza al enfatizar los puntos que más nos interesa conocer, para que más adelante, a partir de ese modelo se obtenga la información requerida, que puede conducir a la toma de decisiones.

Al momento de enfatizar los aspectos más importantes de nuestra realidad, realizamos una abstracción, al separar algunos aspectos de la realidad que conocemos, de otros, que también conocemos, pero que de momento nos interesa separar. Una vez que se determinaron los aspectos relevantes, estos se plasman en el modelo (Trentin, 1992). Como se mencionó en el marco teórico, para el presente trabajo se considerarán tres diferentes niveles de abstracción: (Desiré Atangana & Sepúlveda Lima, 2007):

- Diseño Conceptual.
- Diseño lógico.
- Diseño físico.

De esta forma, se aplicará una metodología que contempla estos tres niveles de abstracción. Como primer punto, se realizará un diseño conceptual, para el que se tomará en cuenta el documento de análisis del proyecto SIGAF. Este diseño contemplará el proyecto de forma global y considerará los conceptos que maneja el proceso de horarios de la FCA de la UABC. Este esquema será realizado una sola vez para la BD completa, y estará representado por un modelo entidad-relación. A partir del diseño conceptual, se realizará el diseño lógico, el cual constituye un refinamiento del conceptual, y será representado a través del modelo relacional. En este diseño lógico se agregará cada modulo, conforme se trabaja, hasta conformar el diagrama lógico completo.

Para la gestión del trabajo, se utilizará tanto la metodología rup, como la metodología scrum. No se sigue estrictamente ninguno de ellos, debido a que la gestión se adaptará a las naturalezas del proyecto mediante una mezcla de estas metodologías, por medio de las cuales se llevará control y seguimiento de las actividades que se realizarán hasta el término del diseño de la base de datos.

3.2 Diseño conceptual.

Codina, (1997) en su artículo, “Una Propuesta de Metodología para el Diseño de Bases de Datos Documentales”, menciona que “en el diseño de los sistemas de información, el cual incluye el diseño de una base de datos, es muy útil definir como primer punto un sistema de información como un sistema *S1*, al cual se le denominará sistema de información, que mantiene registros sobre otro sistema del mundo real que se desea representar, *S2*, al cual se denomina sistema objeto. Así, el proceso de análisis y diseño se puede entender como

un intento de obtener un modelo de aquella parte de la realidad, o sistema objeto (S2), que resulta de interés para el sistema de información (S1)".

Como se menciona en el marco teórico, es importante considerar el dominio de un problema, que para el caso de la presente tesis, se refiere al proceso de generación de horarios de la FCA de la UABC, esta sería la parte de la realidad que es de nuestro interés, y que representará el sistema objeto. Como en este modelo se realiza una abstracción de alto nivel, van a quedar plasmados únicamente los aspectos más relevantes del proceso de horarios, a fin de definir el contenido general de la base de datos.

Una herramienta en la que se apoyará el presente trabajo, es en el análisis realizado por la persona que tiene ese rol, a partir del cual será posible tener una idea clara respecto a los requerimientos de los usuarios. Este análisis debe ser previamente revisado con las personas involucradas (stakeholders) en el proyecto.

Dentro de este contexto, información como materias, planes de estudio, grupos o información curricular de los docentes, entre otros, quedarán representados de forma general en los requisitos así como también en los diagramas UML. A partir de esta información ya se pueden identificar las entidades y las relaciones que existen entre ellas, para conformar si no todas, las características más relevantes de dichas entidades, por ejemplo, para el caso de los docentes se requiere conocer información de los mismos, como nombre, número de empleado, apellidos, profesión, entre otros. Estas características conforman los atributos, los cuales quedarán representados en el esquema conceptual, que plasmará la información que se requiere controlar en el sistema y la relación existente entre dicha información en forma de verbos; así se podrá decir que las materias <conforman> planes de estudio, que las materias <conforman> grupos, o que los docentes <tienen> profesiones.

Para la realización del esquema conceptual, se llevan a cabo los siguientes pasos:

1. Identificar las entidades.
2. Identificar las relaciones entre las entidades.
3. Identificar los atributos de las entidades, las relaciones que tienen y asociarlos.
4. Especificar las claves (candidata, principal y alternativa).

5. Revisar las redundancias.
6. Validar el modelo.
7. Revisar el modelo conceptual con los usuarios.

Los pasos más importantes para la realización de las actividades anteriores se mencionan detalladamente en los siguientes puntos:

3.2.1 Identificación de entidades.

La identificación de entidades es el primer paso a la hora de realizar el diseño conceptual, identificar los objetos principales en los que los usuarios están interesados, o también los objetos de los cuales se habla o se tiene documentación en los requerimientos; por ejemplo, horarios, plan de estudios de la materia, dirección, profesión, universidad de egreso y especialidad del docente.

Se debe tener especial cuidado en la forma en que los usuarios se expresan, ya que es frecuente que utilicen sinónimos para expresar una misma cosa o pueden referirse también con los nombres de las personas o con los roles que desempeñan, aunque puedan pertenecer a una misma entidad; por ese motivo, se debe tener cuidado y ser observadores en ese aspecto.

Una vez que se identifican las entidades, se les asignan nombres que representen lo que el usuario quiere expresar, y se consideran los alias o nombres sinónimo; estas entidades deberán ser especificadas en el diccionario de datos. La simbología que se usará a partir de aquí para el diagrama entidad-relación, es el especificado en la figura 6, debido a la utilización del modelo chen para representarlo.

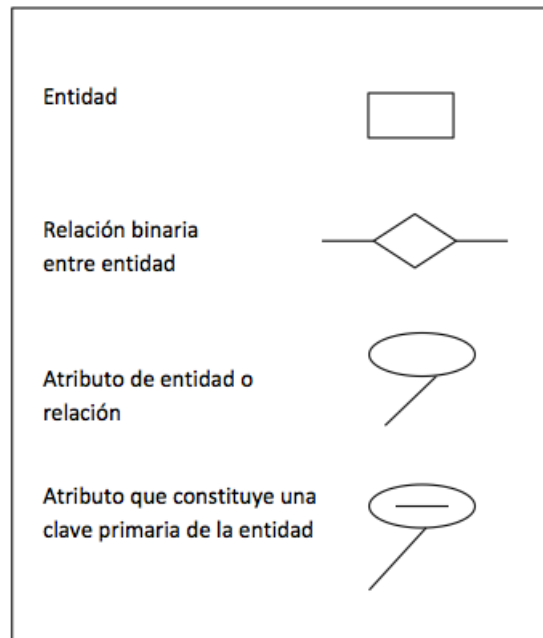


Figura 6. Simbología que se utilizará en el modelo E-R. Fuente: Estructura y organización de una Base de Datos. **(Trentin, 1992)**, Marco de descubrimiento de conocimientos para datos estructuralmente complejos con énfasis en el análisis de eventos en series temporales. **(Lara T., 2010)**

La realidad que se quiere representar para el presente trabajo, es el proceso de horarios; entre los objetos de esa realidad y basándose en el documento de análisis así como en la retroalimentación con usuarios, se podrán extraer por ejemplo docente, curso y perfil, que se convierten en entidades; de esta forma se identificarán el resto de las entidades hasta extraer aquella parte de la realidad en alto nivel, que es de interés para el proceso de horarios, hasta conformar el diagrama. La forma en que se representan esas entidades, son las que se muestran en la figura 7.

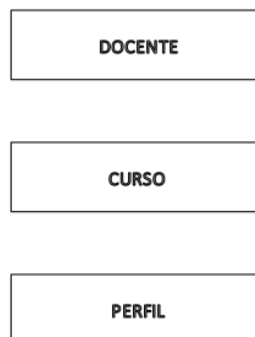


Figura 7. Ejemplo de representación de las entidades docente, curso y perfil. Fuente propia.

3.2.2 Identificación de relaciones entre las entidades.

La identificación de las relaciones, se realiza mediante la asociación entre entidades a través de verbos, o expresiones verbales que representan el nombre de la relación. Esto se expresa gráficamente mediante un diamante dentro del cual se indica el nombre de la relación para ser conectadas a la entidad con una línea recta. La forma en que se relacionarán las entidades y sus diferentes cardinalidades será como lo muestra la figura 8 en que se pueden observar algunos ejemplos. La figura 8 muestra una cardinalidad de uno a uno (1:1) lo que quiere decir que a cada carrera le corresponde una sola clasificación de tipo de carrera. En cuanto a la cardinalidad uno a muchos (1:M) la interpretación en la figura es que un docente puede tener muchos documentos almacenados como por ejemplo certificado, título universitario, cédula profesional entre otros. Para la cardinalidad muchos a muchos (M:N) en la imagen se muestra que las unidades académicas pueden tener varios directivos, y a cada directivo le puede corresponder diversas unidades académicas. En el último caso, al transferir esta información al esquema relacional se agregará una nueva entidad que vinculará las dos existentes a fin de simplificar el diagrama, que es lo que sucede por lo general con la cardinalidad de muchos a muchos. Esto se realiza al conservar las dos entidades existentes y la creación de una nueva entidad que incluirá todos los registros de la entidad uacademica y todas las combinaciones que tenga con la entidad directivo.

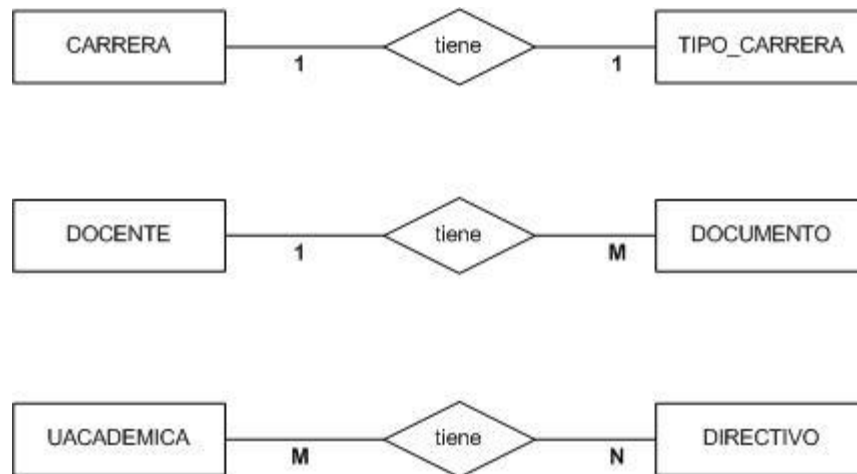


Figura 8. Ejemplo de la representación relaciones entre entidades. Fuente propia.

Una vez identificadas las entidades, se está en posibilidad de especificar los atributos o características de cada entidad.

3.2.3 Identificar los atributos de las entidades y sus relaciones y asociarlos

Para la identificación de los atributos, se determinan las características de la entidad y enseguida se puede plantear la pregunta ¿Qué se puede almacenar en este atributo? De aquí se determina el tipo de atributo que se almacenará. En el modelo chen, los atributos se representan por medio de un óvalo unido a la entidad a través de una línea. Si los atributos son multivaluados, la línea es doble.

En ese contexto, si se toma el ejemplo de entidades docente, curso y perfil que se ejemplifican en párrafos anteriores, se identificarán sus atributos de la siguiente forma:

- **Docente.** Se refiere a la información de identificación personal correspondiente a los docentes, como su nombre completo, número de empleado, fecha de nacimiento, teléfonos, correo electrónico, entre otros datos, los cuales representan sus características, y por lo tanto sus atributos.

- Curso. De acuerdo al análisis, las características que se quieren representar de los cursos, son: los cursos que un docente ha tomado, impartido o cursos en los cuales el docente está interesado en tomar, lo que los convierte en sus atributos.
- Perfil. Perfil académico del docente, como su grado académico, número de cédula, carrera de egreso, entre otros datos requeridos.

Si se toma en consideración lo expuesto en párrafos anteriores para la entidad docentes se obtiene la entidad y sus atributos correspondientes conforme a lo expuesto, como se muestra en la figura 9.



Figura 9. Ejemplo para la entidad docentes y algunos de sus atributos. Fuente propia.

3.3 Diseño Lógico.

La fase de diseño lógico de la base de datos, se enfoca en refinar el modelo conceptual de datos (Mannino, 2007), el cual no está totalmente desligado del diseño lógico; se debe tomar en cuenta que ya se tienen identificados algunos de los elementos que conformarán el diseño lógico, que al igual que el diseño conceptual, se va a representar de manera gráfica. En el diseño lógico quedaran plasmados las entidades y las relaciones existentes entre ellas, así como los atributos de cada una y la integridad definida para evitar la redundancia.

3.3.1 Representación Gráfica de la Integridad Referencial.

Actualmente existen herramientas que ayudan en la automatización de las tareas de un proyecto. En el caso de las bases de datos, existen aplicaciones que ayudan al desarrollador y administrador de bases de datos a visualizar de una manera sencilla la

integridad de entidad y de relación para darse cuenta de forma sencilla sobre la interconexión de entidades, y que entidades requerimos para extraer alguna información.

Para definir la clave primaria, se debe identificar que campo representa unívocamente a la entidad. Si se toma como ejemplo una entidad llamada docentes que almacena los datos personales de los docentes de cierta universidad, como numero de docente, nombre, apellido paterno, apellido materno, entre otros, cabe la posibilidad de que la clave primaria pueda ser el numero de docente, debido a que todos los docentes cuentan con un número asignado que es único; al momento del registro en la BD, no podría ir con un valor nulo.

3.4 Metodología Scrum.

Para la realización de la base de datos del sistema SIGAF, se usará la metodología scrum, en la que se separará cada sprint de acuerdo a los módulos del proyecto, para dividirse de la siguiente manera:

1. Plan de estudios
2. Carga académica
3. Disponibilidad docente
4. Creación de horarios
5. Nivel de usuarios

Cada uno de estos sprints, se va a dividir en historias de usuario, en cada una de las cuales se va a repetir el mismo procedimiento: registrar, consultar, modificar, eliminar y reportes, esto es, para la base de datos, se va a realizar cada una de las tareas que corresponden o están involucradas con la historia de usuario, por ejemplo:

1. Plan de estudios: registrar, consultar, modificar, eliminar y reportes Carga académica
2. Disponibilidad docente: registrar, consultar, modificar, eliminar y reportes

Como en los ejemplos anteriores, realizarán las mismas historias de usuario para cada sprint, es decir, sabemos que para que un plan de estudios se pueda registrar, es necesario que exista algo en que almacenarlo, por lo tanto, ya debe estar creada la base de datos, con las entidades correspondientes, las llaves primarias, y las relaciones que existen entre dichas entidades, la base de datos deberá ser capaz de almacenar de forma

transparente la información, de tal forma que esta pueda ser consultada, modificada, eliminada, y manejará las vistas pertinentes como para permitir las consultas y reportes que sean necesarios. Esto es dentro de cada historia, se realizarán tareas específicas encaminadas a completar una historia de usuario. Las tareas se deben cerrar con el status de completado, de tal forma que deben estar cerradas todas las tareas, y así cerrar una historia. Así se deberán cerrar todas las historias a fin de completar y cerrar cada uno de los sprints. Por ejemplo, para el sprint plan de estudios que se encuentra dividido en historias de usuario, que a la vez se dividen en tareas, las actividades a realizar serán de la siguiente forma:

(Sprint 1) Plan de estudios

US1 Registrar plan de estudios.

Tarea1

Tarea2

...

US2 Consultar plan de estudios.

Tarea1

Tarea2

...

US3 Modificar plan de estudios.

Tarea1

Tarea2

...

US4 Eliminar plan de estudios.

Tarea1

Tarea2

...

US5 Reporte carga académica

Tarea1

Tarea2...

Cada una de las historias llevará un registro de las horas estimadas para su realización, y las horas reales que se tomo esa actividad, para que al final sea posible totalizar las horas reales y estimadas por tarea, por historia de usuario y por sprint. En la figura 10 se muestra un ejemplo de la aplicación de la metodología para el desarrollo de base de datos.

Order	ID	Name	Iteration	State	Estimate	To Do	Actuals
			All	All	87.0	14.0	35.5
+	TA50	CA-Elaboración de procedimientos e implementación	2.0 Carga académica	D P C	10,0	0,0	
+	TA125	Investigacion sobre Integridad en MySQL	2.0 Carga académica	D P C	10,0	0,0	
+	TA134	PE-Cambios en Diagramas/Rev 25 marzo 2014	2.0 Carga académica	D P C	10,0	0,0	10,0
+	TA135	PE-Cambios en Diccionario s/Rev 25 marzo 14	2.0 Carga académica	D P C	5,0	0,0	3,0
+	TA136	PE-Cambios en SP	2.0 Carga académica	D P C	5,0	0,0	5,0
+	TA252	PE-Revision 25 mar 14	2.0 Carga académica	D P C	1,5	0,0	1,5
+	TA253	Investigacion WorkBench	2.0 Carga académica	D P C	15,0	0,0	6,5
+	TA254	PE-Revision 01 abril 14	2.0 Carga académica	D P C	1,5	0,0	1,5
+	TA255	PE-Cambios Logico s/Rev 1 abril 14	2.0 Carga académica	D P C	10,0	0,0	2,0
+	TA256	PE-Cambios Diccionario s/Rev 01 abril 14	2.0 Carga académica	D P C	5,0	0,0	2,0
+	TA257	PE-Documentacion s/Rev 25 mar 14	2.0 Carga académica	D P C	0,0	0,0	2,0
+	TA263	PE-Elaboracion DB	2.0 Carga académica	D P C	1,0	1,0	1,0
+	TA264	PE-Revision Integridad en DB	2.0 Carga académica	D P C	10,0	10,0	1,0
+	TA265	PE-Elaboracion de Vista	2.0 Carga académica	D P C	3,0	3,0	

Figura 10. Ejemplo de aplicación de metodología scrum para la BD del sistema SIGAF. Fuente propia.

Capítulo IV

Desarrollo

En este capítulo se describen las actividades relacionadas con el diseño de la BD para el sistema SIGAF. Se describe la transformación que sufre el diseño al agregar cada una de las entidades que participan, y se hace énfasis sobre el papel que juegan para satisfacer las necesidades de información del proceso especificadas detalladamente en el análisis. También se da a conocer como cada una de estas entidades se entrelazan hasta culminar en el diagrama lógico completo que contiene las especificaciones técnicas de una BD, como los enlaces entre las entidades, la información que se requiere sea única, los datos que por ningún motivo deberán ir vacíos, los procedimientos necesarios, los índices establecidos para eficientizar las consultas, los nombres de las entidades y campos para completar la BD final mediante las sentencias y opciones que ofrece el manejador de BD MySQL. Se explica la realización del diseño conceptual, lógico y físico. El diseño conceptual se realiza al inicio, de forma única para la BD completa y se continúa con el diseño lógico y físico de cada módulo, y a su vez, por cada entidad.

4.1 Diseño Conceptual del Plan de Estudios.

Como se explicó en el capítulo II, el diseño conceptual es una representación del mundo real, donde no es relevante que tipo de infraestructura, lenguaje de desarrollo o manejador de base de datos se va a utilizar. Para la presente tesis, se tomaron como base los requerimientos especificados en el análisis, y se obtuvo una serie de información que se consideró para conformar el diagrama lógico. De forma general, los principales rubros de información obtenidos son los siguientes:

1. Docentes. Se refiere a la información de identificación personal correspondiente a los docentes, como su nombre completo, número de empleado, fecha de nacimiento, teléfonos, correo electrónico, entre otros datos
2. Cursos. Cursos que un docente ha tomado, impartido o en los cuales está interesado.
3. Perfil. Perfil académico del docente, como el grado académico que tiene actualmente, número de cédula, la carrera de la que egresó, entre otros datos.
4. Documentos. Para la FCA es muy importante almacenar en formato digital la documentación que ampara su formación profesional, como fotografía, cédula profesional, entre otros.

5. Categoría. Se refiere al tipo de categoría del docente. La categoría especifica si un docente es de tiempo completo, de asignatura, o alguna otra, entre las diversas categorías que maneja la UABC.
6. Disponibilidad. Se refiere a la disponibilidad de un docente para impartir determinada materia. En este punto se debe considerar la fecha y hora en la que el docente está disponible.
7. Días. Catálogo de días disponibles.
8. Horas. Catálogo de horas disponibles.
9. Unidades de aprendizaje. Catálogo de las materias que conforman los planes de estudio de la UABC.
10. Planes de Estudio. Se refiere a los planes de estudio que maneja la universidad.
11. Periodos. Son los periodos a los que pertenece la información. Por ejemplo una carga, un horario, un grupo pueden pertenecer a determinado periodo.
12. Carreras. Es el catálogo que almacena la información de los programas educativos.
13. Tipo de Carrera. Una carrera puede ser semestral, cuatrimestral, o algún otro tipo que se especifique.
14. Unidad académica. La UABC maneja de forma oficial una serie de unidades académicas (uac), cada una de las cuales ofrece carreras diferentes que pueden estar ubicadas dentro diferente campus o ciudad. Ejemplo: Facultad de Contaduría y Administración (FCA), Facultad de Ciencias Químicas e Ingeniería (FCQI).
15. Nivel de la Carrera. Los diferentes niveles a los que puede pertenecer un programa educativo, como licenciatura o posgrado.
16. Etapas. Cada unidad de aprendizaje pertenece a determinada etapa. Las tres etapas que se manejan en la UABC, son: básica, disciplinaria o terminal.
17. Grupos. Un grupo está conformado por tres dígitos que tienen un significado establecido con la siguiente interpretación: El primer dígito corresponde al programa educativo, el segundo representa el semestre y el último dígito corresponde al número consecutivo de grupo. Esta nomenclatura es de gran apoyo al momento interpretar el código de grupo, debido a que se identifican de forma sencilla los datos que lo conforman.
18. Turnos. Información referente a los turnos manejados por la FCA, que generalmente son: matutino, vespertino e interturno.
19. Aulas. Catálogo de las aulas disponibles. Para este proyecto, se consideran únicamente las aulas pertenecientes a la FCA.
20. Usuarios. Identifica los usuarios que pueden firmarse en el sistema de horarios.

21. Horario Escolar. Una vez recopilada la información y capturados los datos necesarios, estos en conjunto nos llevan a tener disponibles las herramientas de información necesarias para establecer un horario escolar para el periodo que se trabaja. Un horario contiene información relevante: grupo, periodo activo, unidad de aprendizaje, día y hora en que se impartirá esa unidad de aprendizaje, así como el docente que impartirá; también manejará los siguientes status: completado (1), capturado (2), en proceso (3). Sin esta información el horario, no puede establecerse.

Después que de identificar la información necesaria, se realizó la unión de la misma, mediante relaciones en forma de verbos. Algunas de las uniones realizadas, fueron las que se especifican a continuación:

```
Docentes <tienen> cursos
Docentes <tienen> documentos
Materias <forman> planes de estudio
Planes de Estudio <asignados a> Periodos
Materias <distribuidas> en Etapas
```

Después de lo anterior se trasladó esa información a un formato gráfico.

4.2 Diseño Lógico del plan de estudios.

El primer módulo que se empieza a diseñar, es el plan de estudios. Como se especifica en el análisis, este módulo tiene como objetivo principal gestionar las características necesarias de las unidades de aprendizaje que lo conforman y que deben cursarse en una carrera, para obtener el grado. El plan de estudios sentará las bases para la generación de horarios y podrá ser utilizado por el director, subdirector, coordinadores de materia y de carrera, tutores y personal administrativo (analistas) de la facultad. Para efectos del sistema SIGAF, se llama empleado a las personas que tienen una relación con la UABC y con el sistema SIGAF.

Cualquier docente agregado al sistema SIGAF será conocido como <empleado>; se llama así para tener una forma estándar de referirse a las personas que tienen diferentes roles dentro del

sistema que pueden ser usuarios, docentes, directivos, administradores debido a que la forma en que se encuentran estructuradas las entidades está estrechamente relacionada.

Este módulo, está conformado por las entidades plan de estudios, unidad de aprendizaje, campus, facultades, carreras, ciudad, estado y país donde se encuentra el campus; y otras características del plan: especialidad, nivel, periodo, coordinador de carrera, administrador, director y subdirector de facultad.

Un aspecto importante en el diseño lógico, es la aplicación de las reglas de integridad, para establecer las relaciones entre las diferentes entidades que conforman el sistema.

4.2.1 Aplicación de reglas de Integridad para el sistema SIGAF.

Para ejemplificar la integridad, observemos las entidades que se muestran en la tabla 4, que muestra las entidades docentes y contratos.

Tabla 4. Ejemplo Integridad Referencial. Fuente propia.

DO_docente	DO_nombres	DO_ape_pat	DO_ape_mat	DO_fec_nac	DO_status	DO_fec_ing	DO_sexo	DO_contrato
500	Jose	Perez	Andrade	01/12/1973	A	01/12/2010	M	1
501	Maria	Lopez	Sánchez	04/06/1960	A	06/02/1996	F	2
502	Sonia	Martinez	Garcia	01/09/1980	A	02/04/1997	F	1
503	Juan Pedro	Robles	Gasca	28/07/1975	A	07/08/1997	M	
504	Jose	Gracia	Lopez	05/06/1975	A	08/09/2000	M	3
505	Alma	Martinez	Valenzuela	16/05/1973	A	03/05/2010	F	
506	Federico	Lopez	Rosales	27/01/1970	A	27/07/2000	M	2
507	Liliana	Carrillo	Gallegos	18/05/1965	A	04/05/2011	F	2
508	Jose Alfonso	Sanchez	Morales	09/03/1979	A	09/03/2013	M	2
509	Rosa Maria	Gomez	Sanchez	10/02/1981	A	05/01/2012	F	3

CON_contrato	CON_descripcion	CON_HrsTot	CON_HrsClaseLic	CON_HrsClasePost	CON_HrsActividad	CON_Hrs_Extra	CON_Hrs_Tope
1	Tiempo completo	40	40	0	0	0	40
2	Medio tiempo	20	20	0	0	0	20
3	Contrato 10 Hrs	5	0	0	5	0	5

- *Entidad docente.* Contiene una clave primaria llamada DO_docente, y una clave foránea llamada DO_contrato.
- *Entidad contratos.* Entidad que maneja como clave primaria DO_contrato.

Las llaves de una entidad, se pueden designar como parte de una sentencia CREATE TABLE; se usa la sentencia CONSTRAINT para designar la llave primaria o foránea al final de la sentencia CREATE, como se especifica en la siguiente sentencia:

```

CREATE TABLE CONTRATOS (
CON_contrato      INTEGER,
CON_descripcion   VARCHAR(20),
CON_HsTot         INTEGER,
CON_HrsClaseLic   INTEGER,
CON_HrsClasePost  INTEGER,
CON_HrsActividad  INTEGER,
CON_Hrs_Extra     INTEGER,
CON_Hrs_Tope      INTEGER
CONSTRAINT PKCONTRATOS PRIMARY KEY (CON_contrato))

```

En el párrafo anterior, se ejemplifica una integridad de entidad, debido a que la columna CON_contrato contribuye a identificar de forma única los valores almacenados.

```

CREATE TABLE DOCENTES (
DO_docentes      INTEGER,
DO_nombres       VARCHAR(20),
DO_ape_pat       VARCHAR (20),
DO_ape_mat       VARCHAR(20),
Do_fec_nac       DATE,
DO_status        INT,
DO_fec_ing       DATE,
DO_sexo          CHAR(1),
DO_contrato      INTEGER,
CONSTRAINT PKDOCENTES PRIMARY KEY (DO_docente),
CONSTRAINT FKDO_contrato FOREIGN KEY (DO_contrato) REFERENCES
CONTRATOS (CON_contrato))

```

En el párrafo anterior, se aplica una integridad de entidad, y una integridad referencial, debido a que la columna DO_docente identifica de forma única los valores almacenados en la entidad DOCENTES. En cuanto a la integridad referencial, a través de la referencia FKDO_contrato se especifica que la columna DO_contrato es llave foránea de la clave CON_contrato de la entidad CONTRATOS, lo que relacionará la entidad DOCENTES y la entidad CONTRATOS.

En un sistema de Horarios, podría darse el caso de que una entidad llamada GRUPOS manejara una llave primaria compuesta. Esto se especificaría en la sentencia create, de la siguiente sentencia:

```
CREATE TABLE GRUPOS (  
GR_grupo          CHAR(19),  
GR_facultad       CHAR (2),  
GR_plan           CHAR (5),  
GR_periodo  CHAR (5),  
GR_turno          INT,  
GR_carrera  CHAR(2),  
GR_grado          INT,  
GR_gpo           INT,  
GR_creditos INT,  
CONSTRAINT PKGRUPOS PRIMARY KEY  
(GR_grupo,GR_facultad,GR_plan,GR_periodo,GR_Turno),  
CONSTRAINT FKGR_facultad  FOREIGN KEY (GR_facultad)  
REFERENCES PLANenc(PE_facultad),  
CONSTRAINT FKGR_plan  FOREIGN KEY (GR_plan)  
REFERENCES PLANenc(PE_plan) ,  
CONSTRAINT FKGR_carrera  FOREIGN KEY (GR_carrera)  
REFERENCES PLANenc(PE_Carrera),  
CONSTRAINT FKGR_periodo  FOREIGN KEY (GR_periodo)  
REFERENCES PERIODOS(PR_periodo),  
CONSTRAINT FKGR_turno  FOREIGN KEY (GR_Turno)  
REFERENCES TURNOS(Turno),  
CONSTRAINT FKGR_grado  FOREIGN KEY (GR_Grado)  
REFERENCES GRADOS(Grado),  
CONSTRAINT FKGR_gpo  FOREIGN KEY (GR_Gpo)  
REFERENCES catGRUPOS(Gpo)  
)
```

Debido que las columnas GR_grupo,GR_facultad,GR_plan,GR_periodo,GR_Turno son útiles para identificar de forma única los valores almacenados en la entidad GRUPOS, juntos conforman un valor que no se encuentra repetido en el resto de la entidad mediante una llave primaria compuesta, conformada por varios campos. Además de la llave compuesta, se aplica una integridad referencial, al especificar que las columnas GR_facultad, GR_plan, GR_carrera, GR_periodo, GR_turno, GR_grado, GR_Gpo de la misma entidad, son llave foránea en otras entidades, y de esta forma se aplica una integridad relacional entre la entidad GRUPOS y las entidades PLANenc, PERIODOS, TURNOS GRADOS Y catGRUPOS para establecer una relación entre diferentes entidades así como la especificación de los campos llave.

Hasta ahora se ha especificado las llaves primarias y llaves foráneas junto con la sentencia CREATE; Pero ¿qué sucede para especificar los campos que pueden o no ser nulos? La integridad referencial que permite o no valores nulos para un campo, se puede especificar también por medio de la sentencia CREATE con las palabras reservadas NOT NULL, como se muestra en el ejemplo del siguiente párrafo.

```
CREATE TABLE GRUPOS
GR_grupo          CHAR(19)          NOT NULL,
GR_facultad       CHAR (2)          NOT NULL,
GR_plan           CHAR (5)          NOT NULL,
GR_periodo CHAR (5)          NOT NULL,
GR_turno          INT              NOT NULL,
GR_carrera CHAR(2)          NOT NULL,
GR_grado          INT              NOT NULL,
GR_gpo            INT              NOT NULL,
GR_creditos INT,
CONSTRAINT PKGRUPOS PRIMARY KEY
(GR_grupo,GR_facultad,GR_plan,GR_periodo,GR_Turno),
CONSTRAINT FKGR_facultad FOREIGN KEY (PE_facultad)
REFERENCES PLANenc(PE_facultad),
CONSTRAINT FKGR_plan FOREIGN KEY (PE_plan)
REFERENCES PLANenc(PE_plan) ,
CONSTRAINT FKGR_carrera FOREIGN KEY (PE_Carrera)
REFERENCES PLANenc(PE_Carrera),
```

```

CONSTRAINT FKGR_periodo FOREIGN KEY (PR_periodo)
REFERENCES PERIODOS (PR_periodo),
CONSTRAINT FKGR_turno FOREIGN KEY (Turno)
REFERENCES TURNOS (Turno),
CONSTRAINT FKGR_grado FOREIGN KEY (Grado)
REFERENCES GRADOS (Grado),
CONSTRAINT FKGR_Gpo FOREIGN KEY (Gpo)
REFERENCES catGRUPOS (Gpo)
)

```

En caso de que uno de los campos pueda contener valores nulos, no se especifica nada, solamente se especifica para los campos que NO pueden almacenar valores nulos. En el caso de la entidad GRUPOS, se requiere saber a detalle el código de grupo, unidad académica, plan de estudios, período escolar actual (Enero-Junio 2014, Julio-Agosto 2014, o cualquier otro periodo), Turno (matutino, vespertino), programa educativo (licenciatura en Informática, administración de empresas, o alguna otra), grado (1ero, 2do, 3ero) y grupo (puede haberse abierto 3 o 4 grupos para ese periodo). Esta información debe estar especificada, no pueden existir valores nulos; esto no quiere decir que una llave foránea no pueda almacenar valores nulos; como se comentó anteriormente, las llaves foráneas pueden almacenar estos valores, depende del caso particular de la organización para la cual se diseña la base de datos.

En el caso de la integridad de datos requeridos, por ejemplo, si se observa en la tabla 4, la primera entidad es DOCENTES. En este caso, la columna nombre tendría que contener un valor y no ser nulo; si se toma en consideración que las bases de datos son una representación del mundo real, podemos decir entonces que todas las personas deben tener un nombre, por eso esta columna debe especificarse como columna con datos requeridos. Otro caso sería la columna DO_docente. Si esta columna representa el número de empleado que tiene el docente en la organización, podría especificarse esta columna como una columna obligatoria, que no almacenará valores nulos, porque todos los empleados tienen un número de empleado asignado, y nuevamente, la BD es una representación del mundo real.

Estas entidades cumplen con la regla de integridad de entidad, porque tal como lo dice la regla, en la entidad DOCENTES, la clave primaria (DO_docente) no tiene valores nulos, y no se repite ninguno de los valores de la clave primaria. En la entidad CONTRATOS, la clave primaria

(DO_contrato) tampoco maneja ningún valor nulo y sus valores también son únicos porque no pueden existir dos contratos iguales, o no pueden existir dos docentes con el mismo número de empleado y tampoco un contrato o número de empleado desconocido, esto como ejemplo en caso de que fuera especificado en el análisis. La sentencia completa para estas reglas, que incluye la integridad, y los valores nulos, para la entidad CONTRATOS, se especifica de la siguiente manera:

```
CREATE TABLE CONTRATOS (  
  CON_contrato          INTEGER      NOT NULL,  
  CON_descripcion       VARCHAR(20),  
  CON_HrsTot            INTEGER,  
  CON_HrsClaseLic        INTEGER,  
  CON_HrsClasePost       INTEGER,  
  CON_HrsActividad       INTEGER,  
  CON_Hrs_Extra          INTEGER,  
  CON_Hrs_Tope           INTEGER,  
  CONSTRAINT PKCONTRATOS PRIMARY KEY (CON_contrato))
```

Para la regla de integridad referencial, como se muestra en la tabla 4 en la entidad DOCENTES, campo DO_contrato, se muestran dos docentes que tienen este campo con un valor nulo. Esto es posible, primero, porque DO_contrato es llave foránea en otra entidad, y la regla nos dice que en esos casos es posible que tenga un valor nulo, y es posible interpretar esto de forma que un docente puede no tener aun asignado un contrato, lo cual puede ser correcto en el mundo real, siempre que se haya especificado en el análisis.

Para el caso de restricciones de dominio, por ejemplo, si se quiere almacenar el sexo del individuo, el valor válido en este caso es una cadena de caracteres cuyas únicas opciones permitidas son 'F' y 'M'. En el ejemplo de la entidad CONTRATOS, en la columna CON_HrsTot, es un campo numérico que maneja horas, y cuyo número máximo no puede ser mayor de cuarenta para un docente de tiempo completo.

Una vez establecidas las interrelaciones entre las entidades necesarias para el sistema SIGAF, estas quedarán conectadas, a fin de tener un acceso y almacenamiento eficientes. Las entidades más importantes que conforman el sistema SIGAF, se especifican a continuación.

4.2.2 Entidad planestudio.

Entidad que describe las características necesarias que identifican el plan de estudios de cada programa educativo. Los datos que almacena son datos oficiales que expide la coordinación de formación profesional de la UABC. Los campos y características más importantes que conforman esta entidad son los siguientes:

El código de plan, almacenado en el campo plan, corresponde a la clave con la que se identifica un plan de estudios y está conformada por el año en que fue expedido oficialmente y un número entero. El número entero para los periodos semestrales, por lo general le corresponde el valor de 1 ó 2 debido a que los periodos de Enero a Diciembre tendrán asignado el número 1, y el periodo comprendido de Septiembre a Diciembre el número 2. De esta forma, un plan de estudios de Enero a Diciembre del año 2009, podría tener como clave 20091. En esta entidad, el plan de estudios es único.

Esta entidad también contiene información acerca de los campos para las fechas de inicio y fin de vigencia del plan, que son los campos denominados feciniciovig y fecfinvig, así como un campo que almacena el número de créditos correspondientes a las prácticas, llamado credpracticass. Por medio de un número entero se identificara el nivel del plan que puede pertenecer a una licenciatura o postgrado.

Para identificar el usuario que ingresó, modificó o borró un registro, se utiliza el campo users_id que va a estar presente en la estructura de la mayor parte de las entidades que conforman la BD. Por medio de la aplicación se graba e identifica en este campo el usuario que realizó una alta, actualización o borrado del registro para dar paso a la activación de un disparador encargado de grabar en la bitácora (entidad bitacora) este registro. Los datos del registro son grabados para efectos de la bitácora en la forma que lo muestra la tabla 5.

Tabla 5. Ejemplo de registros en la bitácora de movimientos. Fuente propia.

Bitácora almacena:		
Movimiento	Registro Anterior	Registro Posterior
Alta	✓	
Modificación	✓	✓
Borrado		✓

4.2.3 Entidad plan_programa.

Entidad que se diseñó para almacenar las relaciones existentes entre el programa educativo (carrera) y el plan de estudios, de tal forma que mediante esta entidad se conocen todos los planes de estudio que un programa educativo maneja o ha manejado. Los campos plan y programaeducativo, conforman una llave compuesta para identificar el registro de forma única.

4.2.4 Entidad uaprendizaje.

De acuerdo al análisis, para almacenar las unidades de aprendizaje (ua), es necesario contar con cierta información básica de las mismas, como número de horas que le corresponden de clases (HC), laboratorio (HL), taller (HT), prácticas de campo (HPC), clínicas (HCL), extra (HE); número de créditos (creditos) y fecha en que se autorizó (fec_aut). Es necesario un campo para especificar la clave de la unidad de aprendizaje (uaprendizaje). Toda unidad de aprendizaje pasa por la coordinación de formación profesional y se les asigna una clave. En caso de que no exista una clave definitiva se habilita un clave temporal para proceder a su registro, esto significa que no hay unidad de aprendizaje sin clave, y también es importante resaltar que una clave de unidad de aprendizaje puede ser compartida por dos o más carreras; esto sucede principalmente en tronco común, debido a que una unidad de aprendizaje puede ser cursada en dos o más carreras, siempre que se trate realmente de la misma unidad de aprendizaje, con todas sus características iguales: horas, créditos, nombre y el resto de características oficiales. Estos datos ya vienen definidos y autorizados por parte de la coordinación de formación profesional de la UABC de forma oficial, y deberán respetarse al registrarse en el sistema SIGAF. Al momento que un nuevo plan de estudios se registra en el sistema, como ya se menciono en párrafos anteriores, éste está conformado por una cantidad determinada de unidades académicas que ya pasaron por un proceso de autorización previo de la UABC, de esta forma los datos que se almacenan en el sistema son oficiales. La información que podría ser temporal es el de código de unidad de aprendizaje, como ya se mencionó. En cuanto se tenga asignado el número oficial de unidad de aprendizaje, la base de datos debe ser capaz de permitir el nuevo cambio. Para los números de unidad de aprendizaje temporales, se asignaran números 00001 en delante de forma consecutiva, de tal forma que se pueden identificar si se trata de un numero de unidad de aprendizaje oficial o un numero temporal, debido a que todos los números temporales empezaran con cero. En cuanto al nombre de la unidad de

aprendizaje, se especifica el nombre completo de la unidad de aprendizaje (descripcionmat), el cual también es el nombre oficial por parte de la UABC. Para esta entidad se maneja un campo para cualquier observación (observa), un semestre sugerido (semestre_sug) que corresponde al semestre en el que por lo general esa materia se cursa. Se manejan campos que son llave foránea y tienen su llave primaria en la entidad correspondiente; estos campos son carácter (caracter), requisito seriación (reqseriacion), etapa (etapa), coordinación académica (coordinaciona), clave dependiente (claveD) y plan (plan), que ya se explicaron en forma detallada en párrafos anteriores.

Existen casos en que una materia depende de otra para poder cursarse. El campo claveD almacena esta materia previa que es requisito cursarse antes; son materias que están encadenadas una a la otra. El campo claveD es llave foránea de la misma entidad unidad de aprendizaje, debido a que la materia previa podría ser cualquiera de las materias existentes en el catálogo de materias.

4.2.5 Entidad programaedu.

Esta entidad se refiere a las carreras que se ofertan dentro de cada unidad académica, y se le asigna el nombre programa educativo conforme al análisis y como se le conoce oficialmente dentro de la UABC. Al inicio, el sistema estará enfocado únicamente a la unidad académica de contaduría y administración y las carreras que se tienen almacenadas e identificadas para esta facultad, actualmente, son:

- Tronco común.
- Licenciatura en contaduría (LC).
- Licenciatura en administración de empresas (LAE).
- Licenciatura en informática (LI).
- Licenciatura en negocios internacionales (LNI).

Para esta entidad, es importante conocer, si la carrera esta activa o baja, situación que puede llegar a suceder, también es necesario saber cuál es la especialidad del programa educativo; este último dato aplicaría únicamente para postgrados, pero aunque actualmente solamente se manejen licenciaturas, debe quedar abierto para que la escalabilidad de la base de datos permita en un futuro especificar este dato en postgrados. Para efectos de los horarios, es importante conocer el tipo de periodo que puede ser semestral, cuatrimestral o sabatino. De la misma forma, se debe especificar el nivel del programa educativo que identificará si el

programa educativo es una licenciatura o postgrado. Es necesario los datos de carreras, planes de estudio, unidades de aprendizaje. La UABC está conformada por campus (Tecate, Tijuana, Mexicali); cada campus tiene facultades (FCA, FCQI y otras) y dentro de las facultades se ofertan carreras (Licenciaturas en Informática, Contaduría, Mercadotecnia entre otras). Para que un alumno se considere graduado de estas carreras, el alumno tendría que cursar una serie de materias que conforman un plan de estudios diferente para cada carrera. Adicional a esto, hay cierta información a considerar para un plan de estudios: ciudad, estado y país donde se encuentra el campus; características del programa educativo: especialidad (especialidad de postgrado), nivel (licenciatura, postgrado, doctorado), periodo (semestral, cuatrimestral), coordinador de carrera, administrador, director y subdirector de facultad, para lo cual se almacenarán los empleados, donde se especificará el número de empleado de la persona que ejerce los puestos mencionados.

4.2.6 Entidad p_u.

Existe un catálogo general (uaprendizaje) para registrar las materias que se manejan en la FCA con sus características principales las cuales son compartidas entre los diferentes programas; sin embargo se manejan características independientes por programa como es el caso de la etapa, en la que una materia puede pertenecer a la etapa básica en un programa y pertenecer a la etapa disciplinaria en otro programa. La entidad p_u es la encargada de gestionar estos datos mediante la siguiente información:

Esta entidad contiene un campo de tipo entero llamado programaedu, el cual va referenciado a la entidad programaedu, por lo tanto, los datos que contiene deben existir también en la entidad programaedu. No permite valores nulos y es usada para especificar a qué programa pertenece la materia del registro. Conformar la llave compuesta junto con el campo uaprendizaje.

El código de materia debe existir en esta entidad así como en uaprendizaje para que exista consistencia en los datos, en caso contrario, tampoco podría existir en p_u. El dato entero que no permite valores nulos, conforma llave compuesta junto con el campo programaedu.

El dato numérico de la etapa a la que pertenece la materia es almacenado en el campo etapa. Para que exista una correcta integridad de datos, el código de la etapa también debe existir en la entidad del mismo nombre antes de su registro en p_u. Los valores que puede almacenar hacen referencia a que una materia puede pertenecer a la etapa básica, disciplinaria o terminal.

El caracter de una materia varía de acuerdo al programa al que pertenecen y puede ser obligatoria u optativa. Las materias obligatorias se deberán cursar forzosamente. Las materias optativas, son una opción, debido a que hay una serie de materias entre las que se puede elegir, hasta cumplir con el número de créditos oficial para optativas, por lo tanto, existe la posibilidad de que materias optativas queden sin cursarse por algunos alumnos. Esto dependerá de la elección del alumno o la elección por grupo al seleccionar sus materias a cursar. Este dato llamado caracter deberá existir en la entidad caracter, a fin de que la integridad de la BD no se vea comprometida.

La entidad también debe sugerir un semestre en el que se recomienda cursar la materia, pero no necesariamente tiene que ser el semestre definitivo, se puede cambiar de acuerdo a las necesidades de carga académica de los grupos o de los alumnos. El dato puede almacenar cualquiera de los semestres de los programas y es denominado semestre_sug.

4.2.7 Entidad detalleseriacion.

Las materias que se llevan en los planes de estudio de la UABC, pueden cursarse solas, o pueden formar parte de una cadena de materias. Este último caso sucede si dos o más materias están seriadas en las que no es posible cursar la última materia sin haber cursado las anteriores, debido a la seriación obligatoria existente.

En el caso de que una materia vaya seriada, es decir vaya tras de sí una o varias materias que deben cursarse como requisito, se origina una dinámica como la que muestra la figura 11, en la que las materias deberán cursarse con un orden que no se puede romper. En el ejemplo, primero se deberá cursar sistemas de información; una vez cursada esta, se puede cursar análisis de sistemas, y entonces se está en posibilidades de cursar diseño de sistemas. Bajo ninguna circunstancia podrá cursarse diseño de sistemas si no se ha cursado las materias requisito (análisis de sistemas y sistemas de información). De la misma forma, no podrá cursarse análisis de sistemas si no se ha cursado sistemas de información.

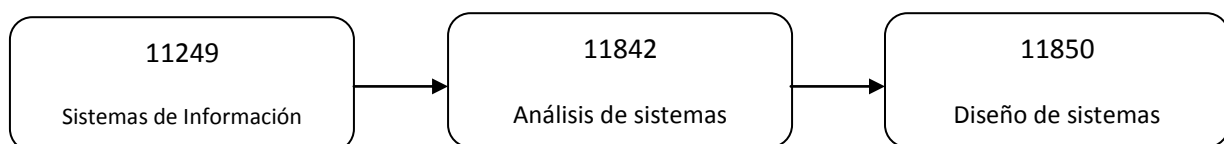


Figura 11. Ejemplo sobre seriación de uac. Fuente: plan de estudios FCA UABC, programa Lic. Informática.

Un aspecto relevante es que no todos los programas educativos manejan la misma seriación, aunque compartan materias iguales. Algunas materias pueden ser seriadas para un programa educativo, pero para otro no. Por este motivo, para materias seriadas, los registros se dan de alta de forma independiente para cada uno de los programas educativos sobre los que se especifica la seriación. Si se toma como ejemplo la figura 11, una seriación que aplique tanto para el programa de licenciatura en informática, como para el de contaduría, se registraría como lo muestra la tabla 6.

Tabla 6. Ejemplo de ua seriadas para programas educativos diferentes. Fuente propia.

programaedu	uaprendizaje	uaprequisito	reqseriacion	users_id
11850	01	11842	1	1
11842	01	11249	1	1
11850	02	11842	1	1
11842	02	11249	1	1

En el ejemplo de la tabla, se especificaron los registros para representar las seriaciones correspondientes a la Figura 11, donde la materia 11850, tiene como materia requisito a cursarse la materia 11842. De la misma forma la materia 11842, tiene como materia requisito a cursarse la materia 11249. Los registros serían iguales para ambos programas, pero cada programa tiene sus propios registros; de esta forma, existen registros propios del programa contabilidad y del programa administración.

Requisito seriación indica si una unidad de aprendizaje es obligada, sugerida o no tiene seriación. Las conexiones de uac seriadas obligadas se representan con una línea continua en los diagramas oficiales de los planes de estudio de la FCA. Las conexiones entre uac sugeridas se representan mediante una línea punteada, como lo muestra el ejemplo de la figura 12.



Figura 12. Conexión entre uac. Fuente: plan de estudios de FCA UABC, programa LNI.

Como parte de la información que se maneja para el plan de estudios, surge la necesidad de almacenar los datos que corresponden al director, subdirector y administrador de las uac, de la misma forma que fue necesario el almacenamiento de los datos de la persona que funge como coordinador de carrera. La forma en que se realiza el almacenamiento de estos datos para uac, está distribuido en dos entidades: puesto y puestos.

4.2.8 Entidad Puesto.

Esta entidad contiene los códigos y descripciones de los diferentes puestos que se requieren manejar para efecto de uac mencionados en el párrafo anterior, de tal forma que se almacenarán códigos y descripciones correspondientes a los puestos de director, subdirector y administrador con sus códigos para conformar el catálogo de puestos.

La entidad puesto, está ligada puesto y puesto por medio del código del puesto, de tal forma que se puede conocer la descripción del puesto para puesto y puesto a través de la relación que existe entre las dos entidades. Un ejemplo sobre los valores que puede almacenar esta entidad, se muestran en la tabla 7.

Tabla 7. Ejemplo sobre el almacenamiento de datos en la entidad Puestos. Fuente propia.

puesto	descripción	Users_id
1	Director	1
2	Subdirector	1
3	Administrador	1

4.2.9 Otras características de plan de estudios.

Los programas educativos tienen características que los identifican. Una de esas características es el campus; cada programa está ubicado dentro de un campus, los cuales se ubican en una ciudad, estado, y país. Los campus actuales de la UABC son los siguientes:

- Tijuana Unidad Otay.
- Tijuana Unidad Tecate.
- Tijuana Unidad Rosarito.
- Tijuana Unidad Valle de las Palmas.

La UABC está dividida uac. Para cada unidad académica, es necesario tener identificado el administrador, director y subdirector de facultad, para lo cual se almacenarán el número de empleado de la persona que ejerce cualquiera de los puestos mencionados. Las uac con las que cuenta la UABC, son las siguientes:

- Facultad de contaduría y administración (FCA).
- Facultad de ciencias químicas e ingeniería (FCQI).
- Facultad de humanidades y ciencias sociales (FHCS).
- Facultad de Derecho (FD).
- Facultad de economía y relaciones internacionales (FERI).
- Facultad de ingeniería y negocios (FIN).
- Facultad de medicina y psicología (FMP).
- Facultad de odontología (FO).
- Facultad de turismo y mercadotecnia (FTM).
- Facultad de artes (FA).

Existe una entidad para establecer la relación que existe entre la unidad académica, los puestos y los empleados, de tal forma que para las uac agregadas, se especifica quien es el director, subdirector y administrador. A esta entidad se le llama *puestosemp*.

La especialidad aplica únicamente para los postgrados dentro de los que se manejan especialidades y maestrías. Las especialidades con las que se cuenta son fiscal y dirección financiera y las maestrías son Maestría en Administración y Maestría en Tecnologías de la información y la comunicación (MTIC). Se conocen como especialidad del programa educativo.

Otra característica de los programas educativos, es el nivel al que pertenece. Puede ser una licenciatura o un postgrado, lo cual deberá ser especificado en el diseño y en la utilización de los datos.

También se especifica para los programas la periodicidad que manejan, porque existen programas que se imparten semestralmente y algunos otros cuatrimestralmente.

4.2.10 Otras características de las unidades de aprendizaje.

Como se explico en párrafos anteriores, las unidades de aprendizaje son materias que conforman los programas educativos de UABC y tienen características que las identifican y que forman parte de las entidades del módulo de plan de estudios. Estas características, se explican a continuación:

Como se mencionó anteriormente, una ua, puede tomar el valor de obligatoria u optativa, lo que significa que un alumno deberá cursar todas las materias catalogadas como obligatorias, y cubrir un número determinado de créditos con materias de carácter optativo. Esta característica es conocida como carácter. Otra característica es requisito seriación, que clasifica el tipo de seriación que pueden tener las ua. Algunas tienen una seriación obligada, que no puede cambiar, mientras que otras no tienen seriación o la seriación es solamente una sugerencia. El requisito seriación puede ser diferente para cada plan de estudio, aunque se trate de la misma ua. Cada ua puede pertenecer a una coordinación académica, y cada coordinación académica pertenece a un programa educativo. Algunas de las coordinaciones a las que puede pertenecer una ua, se pueden mencionar:

- Contabilidad básica (LI)
- Contabilidad avanzada (LC)
- Finanzas (LC)
- Auditoria (LC)

Las unidades de aprendizaje también se dividen en tres etapas: básica, disciplinaria y terminal. Una unidad de aprendizaje puede pertenecer solamente a una de estas etapas.

4.3 Diseño lógico carga académica.

El módulo de carga académica, se encarga de almacenar la información correspondiente los grupos que se van a abrir y las materias asignadas a cada grupo, así como su turno y periodo. Corresponde a los coordinadores de carrera alimentar la carga académica, debido a que son ellos quienes tienen conocimiento de esta información que servirá de base para que personal docente pueda llenar el formato de disponibilidad, donde ellos especifican las materias que están en posibilidades de impartir.

Este módulo está conformado por las entidades turno, periodo, status, grupos y carga.

4.3.1 Entidad carga.

Entidad diseñada para el almacenamiento de las unidades de aprendizaje que se van a ofertar en determinado periodo. La carga académica se identifica por un grupo, de la misma forma que la entidad grupos. La nomenclatura del grupo para la carga académica es de tres números, que tienen el mismo significado que en la entidad grupos. Además del grupo, se agregaron los siguientes campos:

- Periodo, que funciona de la misma forma que en la entidad grupos
- uaprendizaje, se refiere a las unidades de aprendizaje que se ofertarán en el periodo.
- status se refiere al estado de captura de la carga, que se explicó en la entidad planestudio.

4.3.2 Entidad grupos.

Esta entidad se diseñó para almacenar la información correspondiente a todos los grupos que se van a ofertar en el periodo. El código está conformado por tres números. El primero numero corresponde a la carrera, el segundo al semestre, y el tercero al consecutivo de grupo, conforme estos se abren, de tal forma que con ver el código de grupo, se saben, los tres datos básicos: a que carrera pertenece, de que semestre y en que consecutivo de grupo es.

En lo que se refiere al plan, no todos los grupos pueden cursar el mismo plan de estudios, podría darse la situación donde dos grupos cursen planes de estudio diferentes en el mismo periodo, situación que se presenta por lo general al momento de cambiar el plan de estudios a uno nuevo.

4.3.3 Entidad periodos.

Para el manejo de los periodos se diseño una entidad llamada periodos que almacena toda la información correspondiente a los periodos escolares.

- Los campos año y mes fueron agregados para manejar el año de trabajo del periodo. También se lleva un control de fecha de inicio y fin del periodo para la validez del periodo. El status del periodo especifica si un periodo es capturado, completado o en proceso, lo que permite a quien revise la información, de tener conocimiento del status.

4.3.4 Entidad turnos

Este catálogo almacena los posibles turnos que la FCA puede llegar a manejar, y su descripción. Se realizó un diseño que incluye el campo turno para almacenar los códigos de los turnos, es suficiente con el código y descripción del turno, debido que hasta aquí, los horarios del turno son irrelevantes. Los turnos que podrían manejarse son matutino, vespertino e interturno.

4.3.5 Entidad Status_procesos.

Dentro del sistema se realizan las capturas de forma repetitiva cada inicio de periodo, porque la información puede tener desde poca hasta una variación significativa. La diferencia de información se puede presentar principalmente en docentes, cambio de planes de estudio, carga de materias de los grupos y horarios en que se toman las materias. Por ese motivo dentro del sistema se llevan controles sobre el status que guarda cada una de las capturas; para la carga académica y el plan de estudios, se maneja un status por periodo. La entidad status_procesos es la encargada de gestionar el control de estas capturas, en la que se especifica si el proceso esta capturado, completado o en proceso. Las personas que tienen permiso de especificar estos status varían por proceso. Para el plan de estudios, los usuarios de administración son las responsables de la captura de información, y les corresponde a ellas especificar el status de la captura. En cuanto a la carga académica, serán los coordinadores los encargados de especificar el status de captura, debido su responsabilidad de recuperación y entrega de dicha información.

4.4 Diseño lógico disponibilidad docente.

Para efectos de disponibilidad docente, la información relevante son las materias que se ofrecerán en el periodo; basado en esta información, los docentes proponen su propio horario de disponibilidad y materias que son capaces de impartir. Es importante mencionar que aunque este modulo se llama disponibilidad docente y la información sobre disponibilidad corresponde a los docentes, en algunos casos el módulo comparte información sobre directores, subdirectores, administradores, docentes y los datos de todos ellos se almacenan en la misma entidad. Dentro de este contexto, se hace referencia a empleados en general para datos como perfil académico, datos personales, documentación digitalizada recopilada, capacitaciones, perfil profesional en los casos que el empleado tiene una actividad laboral complementaria a su actividad con la UABC.

4.4.1 Características que conforman el perfil profesional y laboral de docentes.

La información correspondiente al perfil académico del docente, tiene como finalidad conocer su preparación académica, las instituciones de egreso, y carrera terminada.

El catálogo de carreras almacena información correspondiente a las carreras que cualquiera de los empleados puede haber cursado como parte de su preparación profesional. De la misma forma, se tiene el catalogo de universidades, que almacenará las universidades en las que cualquiera de los empleados cursó su carrera; este catalogo contiene el nombre de la universidad, una descripción abreviada y la ciudad en la que se encuentra ubicada. Una vez que se tienen los datos de la carrera y la universidad donde se curso, se puede entonces alimentar el catalogo de actualización profesional, donde se almacenan los datos del docente que se desea registrar y su perfil académico: carrera terminada, universidad de donde egresó, si se trata de licenciatura o postgrado, cedula, fecha de titulación y si se obtiene o no el grado. En esta entidad no necesariamente tendría que ser un docente el que se registra; puede ser cualquier empleado que es usuario del sistema. Esta información se encuentra almacenada en las entidades `act_profesional_emp`, `carreras_emp` y `universidades_emp`.

Una vez registrados los datos del perfil profesional, se atiende la necesidad especificada en el análisis, de recopilar la documentación digitalizada necesaria que respalde dicho perfil. Los formatos que de acuerdo al análisis serán validos para la recopilación de documentos serán .jpeg, .jpg, .bmp, .tiff, .gif donde la recomendación es .jpg. El documento principal obligatorio al inicio del registro es la fotografía, pero se tiene la posibilidad de registrar cualquier tipo de

documento: cédulas, certificados, constancias de cursos, títulos, entre otros. Para ello se cuenta con un catálogo de documentos y otro catálogo que relaciona el documento y el empleado correspondiente, la ruta donde se guardara en disco y cualquier observación que se requiera. Las entidades para el almacenamiento de documentos son documentos y documentos_emp.

Otro de los requerimientos de información es la recopilación del perfil laboral del empleado en caso tener una actividad laboral que complemente la relación que se tiene con UABC. Esta información se almacena en empresa_adic, representa un catálogo que almacenaran las empresa donde los empleados han laborado. Una vez que se tiene registrada la empresa, se realiza una vinculación del empleado y la empresa en la entidad trabadicional_emp, que contiene los datos del empleado, la empresa donde labora, la fecha de ingreso y el puesto que tiene actualmente.

Las unidades académicas de la UABC pueden ofrecer capacitaciones que son catalogados por la facultad responsable como cursos, congresos o talleres, y se ofrecen al personal de la facultad en general, desde directivos, docentes, o personal administrativo. El interés de la persona por estos cursos puede variar desde impartirlos, tomarlos o estar interesados en ellos. Para este sistema, se hace referencia a estas capacitaciones de forma general como cursos y la información de este concepto está conformado por cuatro entidades: tipo_curso_emp, act_profesional_adic, act_adicional_emp y característica_cur.

4.4.2 Entidad empleados.

Se desarrolló una entidad llamada empleados que guarda información que sirve de base para las diferentes etapas donde los empleados son requeridos como información personal: nombre completo, apellidos, fecha de nacimiento, celular, teléfonos personales y de oficina, sexo. Es necesario almacenar algunos datos laborales. El campo por medio del cual se vincula al empleado en otras entidades, será el número de empleado representado por el campo llave primaria para esta entidad que tiene la característica de ser único e irrepetible y que corresponde al número de empleado real en la UABC.

4.4.3 Entidad Categorías.

La forma de laborar de los docentes, es diversa. Pueden existir docentes que laboran en la UABC tiempo completo, tiempo parcial entre otros. Para estas categorías existe una definición

formal que en el sistema SIGAF se definieron en la entidad categorías, y que es utilizada en el sistema como mera información.

4.4.4 Entidad días.

Un horario se conforma entre otros datos, de días y horas. Esta entidad representa el catalogo de los días que usados para conformar los horarios. Los datos que almacenara la entidad son los que se especifican en la tabla 8, donde se muestran los valores y los nombres de los campos en la parte superior.

Tabla 8. Valores que se almacenaran en la entidad días. Fuente propia.

Día	Descripción
1	Lunes
2	Martes
3	Miércoles
4	Jueves
5	Viernes
6	Sábado
7	Domingo

4.4.5 Entidad horas.

Esta entidad representa el catálogo de horas, así como la hora de inicio y fin de cada hora. Los atributos que conforman esta entidad, donde ahora si los horarios empiezan a tomar relevancia, son los siguientes:

La tabla 9 muestra algunos valores que podría almacenar esta entidad.

Tabla 9. Valores que almacena la entidad horas. Fuente propia.

Hora	Horaini	Horafin
1	07:00	07:59
2	08:00	08:59
3	09:00	09:59
4	10:00	10:59
5	11:00	11:59

6	12:00	12: 59
7	13:00	13: 59
8	14:00	14: 59
9	15:00	15: 59
10	16:00	16: 59

4.4.6 Disponibilidad docente.

Para especificar la disponibilidad de días y horas en que un docente puede impartir clases, se creó la entidad `disp_hrs_dias`, que contiene quien, en que periodo, que día y a qué hora un docente tiene disponible; sin importar cual materia, se especifica solo el horario disponible.

Para especificar cuáles materias el docente está capacitado para impartir, se diseñó la entidad `disp_ua`. Esta entidad almacena las materias que el docente está en condiciones de impartir, por periodo. El horario escolar se almacena en una tercera entidad llamada `horarios`. Basado en esta información, a continuación se explica la forma en que se conforma el horario escolar.

4.5 Diseño lógico de Horarios.

Dentro del contexto explicado a lo largo de este capítulo, el horario de disponibilidad docente se almacena en una entidad, y su disponibilidad de materias en otra para que al momento de realizar una combinación de las materias con las horas disponibles se conforme el horario escolar, información almacenada en una tercera entidad llamada `horarios`. Es importante resaltar que en el nivel de conformación del horario, ya se tiene un trabajo previo realizado de recopilación de información explicada a lo largo de este capítulo, que sirve de base para la clasificación de la información de los horarios que incluye los programas educativos, planes de estudios, unidades académicas, docentes, disponibilidades, perfiles de los docentes, campus, etapas, categorías, puestos, entre otros. Los horarios son el punto final del trabajo, donde la información mencionada será utilizada para su clasificación, organización o extracción de información que resulte relevante al momento de consultarla; basado en ello, la información del horario se conforma de la siguiente forma:

4.5.1 Entidad horarios.

Entidad que contiene el almacenamiento del horario que será el establecido de manera formal. Conformada por una llave primaria compuesta por grupo, período, unidad de aprendizaje, docente, día y hora. Esta información resulta relevante para la conformación de los horarios para el próximo periodo, debido a que contiene:

- Grupo al que se impartirá la clase.
- Período en que se impartirá.
- Unidad de aprendizaje que se impartirá en ese periodo y a ese grupo.
- Docente que impartirá.
- Día y hora.

Esto significa que estos campos juntos no se repetirán, son únicos. Los campos día y hora, están ligadas a la entidad días y horas respectivamente y el resto de los campos maneja su propia relación con las entidades pertinentes. Se podría tener el registro de un horario con un grupo, periodo, unidad de aprendizaje, docente día y hora, pero con el enlace que se maneja con el resto de los módulos, se puede tener la extracción de información necesaria como la unidad académica de ese grupo, o el nombre y perfil del docente o el detalle de todas las características de la materia que conforma ese horario, entre otros datos.

4.6 Diccionario de Datos.

Como resultado de la información especificada en el capítulo cuatro, se obtiene el diccionario de datos. Este documento contiene la información detallada de cada una de las entidades y sus atributos, a fin de formar un panorama general de la información técnica de los atributos, los nombres y las descripciones de los mismos.

4.7 Bitácora.

Dentro del análisis para el sistema SIGAF se especificó el requerimiento de control de las transacciones de inserción, borrado y actualización que se realizan a cada entidad. Esto con el fin de llevar un seguimiento a nivel base de datos sobre las personas que realizan los movimientos mencionados. La finalidad de esta bitácora es que estará disponible para que pueda ser consultada por los usuarios que tienen el nivel de administrador del sistema. Para esta finalidad, fue creada una entidad llamada bitácora, la cual será capaz de almacenar las

transacciones conforme estas se ejecuten. Para la funcionalidad de la misma, la entidad contiene los siguientes campos, conformados de acuerdo a lo especificado en el documento de análisis:

- *Id.* Representa el código de transacción realizado.
- *Tabla.* Contiene el nombre de la tabla sobre la cual se realizó la modificación, actualización o borrado.
- *Usuario.* Sirve para almacenar el nombre del usuario que realizó la transacción.
- *Operación.* Contiene la operación realizada. Para inserción, se especificara I, para Borrado D, para modificación U.
- *Fecha.* almacena la fecha de la transacción.
- *Registro.* Contiene el registro sobre el que se realiza la transacción, con todos los campos de la tabla, separados por el carácter “|”.
- *Registro_old.* Contiene el registro anterior, con todos los campos de la entidad, separados por el carácter “|”.

Para el funcionamiento de la bitácora, se implementaron disparadores que se ejecutarán después que el usuario realiza alguna de las transacciones mencionadas; por sus siglas en inglés las especificaciones que se realizan son las siguientes:

- <After update >.
- <After insert >.
- <After delete >.

Una vez realizada la transacción, el disparador mandara los datos de la tabla para que sean almacenados los valores, y de esta forma los usuarios que la consulten sabrán entre otros datos, quien realiza la transacción, que fecha, que movimiento es (inserción, modificación o borrado) en que tabla y cuáles son los valores de los campos. Para inserción almacenara únicamente en el campo registro, los valores insertados. Para los Borrados almacenara los valores de los campos resgistro_old. En caso de las modificaciones, se almacenara el registro anterior en el campo registro y se almacenara el registro modificado en el campo registro_old.

4.7 Diseño físico.

Las entidades se codificaron en el manejador de base de datos Mysql, a fin de poder almacenar la información requerida para el funcionamiento de las bases de datos. El codificado de las entidades se realizo en lenguaje mysql. La codificación consistió en trasladar el diseño lógico completo a la Base de Datos, y se trató de mantener sincronizados el diseño lógico y físico.

Capítulo V

Resultados

En este capítulo se presentan los artefactos que resultan de la base de datos del sistema SIGAF, donde el punto de partida es el análisis proporcionado por el analista del proyecto SIGAF. Los artefactos incluyen desde la base de datos que permitirá el almacenamiento de la información para el proceso de gestión de horarios y los datos que la conforman, como resultado de modelar el proceso a través de un diseño conceptual, diseño lógico y diccionario de datos. Estos artefactos, contienen información de interés para el proceso de horarios que fue segmentado en los módulos plan de estudios, carga académica, disponibilidad y perfil de docentes así como horarios que se obtienen de forma detallada y conforman la Base de Datos.

La base de datos que fue diseñada en el gestor de bases de datos mysql y contiene la infraestructura necesaria para ser funcional en cuanto a información, relacionada en los segmentos que la conforman de una forma segura, a través de las sentencias necesarias que hacen cumplir la integridad y contienen las sentencias necesarias para el funcionamiento de una bitácora.

5.1 Diseñar el modelo conceptual y lógico de la Base de Datos.

El diseño de la base de datos incluye el diseño lógico, que fue realizado en un alto nivel de abstracción que resulta en un modelo entidad relación bajo el modelo Chen, que muestra ya un panorama completo acerca de la información que maneja el proceso. Se incluyeron algunos campos en las entidades y los datos están vinculados para mostrar el flujo del proceso de una forma general. Un fragmento del modelo E-R que resultó desde la perspectiva de la Base de Datos, es el que se muestra en la figura 13.

A partir del diseño lógico, se realiza el refinamiento de la información que da como resultado el diagrama lógico del proceso de Horarios, representado a través del esquema relacional, en el que ya se muestran los datos que integran la Base de Datos, ahora en un bajo nivel de abstracción, con cada una de sus características y detalles, que muestra el flujo real en el cual la información está vinculada. El esquema resultante, es el que se muestra en la figura 14

5.2 Elaboración diccionario de datos.

El diccionario de datos representa un listado detallado de las entidades que forman parte del diagrama lógico de la base de datos SIGAF. El nivel de abstracción incluye las entidades y los detalles que la conforman como nombre del campo, descripción y tipo de dato que almacena.

Además de los datos mencionados, se especifica para cada entidad, la llave primaria, si el campo permite almacenar valores nulos, que campos pueden contener un valor único, el valor por default al dar de alta el registro si este valor existe, campos autoincrementables, columnas binarias, campos que tendrán llenado de ceros así como los valores positivos.

Una fracción del diccionario de datos final, se muestra en la tabla 10, en la cual se observa la representación para las entidades horarios, horas, levels y nvprograma. El resto de las entidades con sus detalles específicos se representan de esta misma forma.

Tabla 10. Representación gráfica del diccionario de datos. Fuente: propia basada en el diseño de la BD.

horarios										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
grupo	INT	✓	✓							Gpo
periodo	INT	✓	✓							Periodo
usprendizaje	INT	✓	✓							codigo usp
id	INT	✓	✓							Docente
dia	INT	✓	✓							Dia asignado
hora	INT	✓	✓							Horario asignado
users_id	INT		✓							Usr inserta/modif/borra

horas										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
hora	INT	✓	✓					✓		Codigo hora (1,2,3,4)
horaini	CHAR(5)		✓							Hr exacta inicio (0800)
horafin	CHAR(5)		✓							Hr exacta fin (0859)
users_id	INT		✓							Usr inserta/modif/borra

levels										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
level	INT	✓	✓							Nivel
descrip_level	VARCHAR(45)		✓							Descripcion Nivel
users_id	INT		✓							Usr inserta/modif/borra

nvprograma										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
nivel	INT	✓	✓					✓		Nivel (1,2,3)
descripcion	VARCHAR(15)		✓							Licenciatura Postgrado
users_id	INT		✓							Usr inserta/modif/borra

5.3 Diseño físico de la base de datos que permita el almacenamiento de los datos del proceso de horarios de una forma segura y eficiente que incluya una bitácora de movimientos, así como el almacenamiento de los datos básicos para su funcionamiento.

La base de datos fue elaborada en el manejador de base de datos MySQL e incluye la información suficiente para que el proceso de horarios se pueda gestionar en esta y se incluyen los módulos en lo que fue segmentado el proceso.

Para el módulo Plan de Estudios se recopila información correspondiente a la estructura que da soporte a las materias que se cursaran en la FCA de la UABC y así como la información alrededor de estas, tales como programas educativos, campus, nivel correspondiente al programa entre otros.

La base de datos también clasifica la información referente a la estructura de la carga académica, encargada de la información sobre las materias que se ofrecerán para el período activo. Esta información está directamente ligada a la anterior, porque la clasificación que se realiza es por programa educativo con cada una de sus con características propias, y están directamente involucradas las materias. Existe información adicional al módulo de plan de estudios que es necesaria como los turnos, grupos, periodos y status. A partir de esta información, se tiene ya un panorama sobre la carga académica que se ofrecerá en el periodo, complementada con las características de la información de materias, localizadas en el plan de estudios.

La parte complementaria a la información mencionada, corresponde a los datos de los docentes y su disponibilidad, catalogada como la información correspondiente al modulo de disponibilidad docente. Parte de la información sobre los docentes se refiere a los datos personales, perfil profesional, perfil laboral, y un análisis sobre el perfil respecto a las capacitaciones que ofrece en la FCA. También contiene información que servirá de base para la gestión de los usuarios en la que se incluye la clasificación del nivel de seguridad por docente.

La información de carga académica se complementa con información sobre la disponibilidad de horario y materias que el docente esta en posibilidad de impartir. Hasta esta parte, la Base de Datos es capaz de ofrecer un panorama sobre la disponibilidad y perfil de docentes, usuarios y los niveles de seguridad que le corresponden, materias que se ofrecerán en el periodo, así como las características del plan de estudios y de los programas educativos desde lo básico a lo específico, de acuerdo a la información recopilada en el análisis y revisada por el equipo del proyecto de desarrollo del sistema SIGAF.

Para el funcionamiento adecuado de la información, existen datos básicos que deberán alimentarse como requisito para que el resto de la información pueda ser alimentada. Tal es el caso de las ciudades, estados países, campus, especialidades, unidades académicas, al menos la unidad académica que se alimentará en ese momento, programas educativos de la unidad académica con la que se trabaja, etapas existentes, tipos de carácter existentes, entre los más importantes. Esto obedece al hecho de que al dar de alta un programa educativo en ese momento se deberán especificar de forma previa los datos mencionados que le corresponden a esa materia.

De la misma forma, no podrá capturarse una carga académica si no se han alimentados de forma previa las materias y sus características, o si no se alimentan previamente los grupos.

Las tablas 11, 12 y 13 muestran ejemplos de información base alimentada en la Base de Datos.

Tabla 11. Información de materias alimentada en la Base de Datos. Fuente: propia basada en información real del proyecto.

uaprendizaje Código uaprendizaje (SOLO NUMERICOS)	descripcionmat Descripción uaprendizaje	HC Horas Clase	HL Horas Laboratorio	HT Horas Taller	HPC Horas	HCL Horas Clínica	HE Horas Extra	creditos Número de Créditos 1hr=Clase 2c	fec_aut Fecha autorización	observa Observaciones	semestre_sug Semestre sugerido	caracter Código caracter para especificar 1=Obligatoria/2=Optativa	reqseriacion Código req seriación para especificar Obligada, Sugerida, Asesoración	etapa Código etapa para especificar Básica, Disciplinario, Terminal	coordinaciona Código coordinación area	claveD Clave materia dependiente	plan Código plan estudio	users_id
11236	COMUNICACION ORAL Y ESCRITA	1	0	2	0	0	0	4	NULL	NULL	0	1	3	1	21	NULL	20092	1
11237	CONTABILIDAD	2	0	3	0	0	0	7	NULL	NULL	0	1	1	1	1	NULL	20092	1
11238	DESARROLLO HUMANO	1	0	2	0	0	0	4	NULL	NULL	0	1	3	1	8	NULL	20092	1
11239	ENTORNO GLOBAL DE LOS NEGOCIOS	2	0	2	0	0	0	6	NULL	NULL	0	1	3	1	10	NULL	20092	1
11240	MATEMATICAS	2	0	2	0	0	0	6	NULL	NULL	0	1	3	1	22	NULL	20092	1
11241	RESPONSABILIDAD SOCIAL	2	0	1	0	0	0	5	NULL	NULL	0	1	3	1	21	NULL	20092	1
11242	TECNOLOGIAS DE LA INFORMACION	1	3	0	0	0	0	5	NULL	NULL	0	1	3	1	12	NULL	20092	1
11243	CONTABILIDAD APLICADA	2	0	3	0	0	0	7	NULL	NULL	0	1	1	1	1	11237	20092	1
11244	DERECHO	3	0	0	0	0	0	6	NULL	NULL	0	1	3	1	19	NULL	20092	1
11245	ECONOMIA	2	0	2	0	0	0	6	NULL	NULL	0	1	3	1	17	NULL	20092	1

Tabla 12. Información de periodos alimentada en la Base de datos. Fuente: propia basada en información real del proyecto.

periodo Periodo de trabajo(Año 4 dígitos+ semestre o cuatrím) ej. 20091,20092	periodo_pedu Tipo: 1=semestral/2=cuatrimestral	year Año periodo	mes Mes periodo	descripcion Descripción	inicio Fecha inicio periodo	fin Fecha fin periodo	status Status 1,2 (1=Abierto, 2=Cerrado)
20141	1	2014	1	PERIODO ENERO-JUNIO 2014	2014-01-01	2014-06-30	3
20142	1	2014	2	PERIODO-SEPTIEMBRE DICIEMBRE 2014	2014-09-01	2014-12-31	2

La información que contiene la Base de Datos es confiable, debido a que respeta las reglas de integridad que se trabajaron desde el diagrama lógico y se codificaron al momento del diseño físico de la Base de datos. De esta forma, no aceptará información inválida de acuerdo a dichas restricciones; no aceptará por ejemplo un campus ubicado en una ciudad inexistente, o una materia ubicada en un programa educativo que no existe o que pertenezca a una etapa que no se haya dado de alta con anterioridad. De la misma forma, la base de datos no permitirá grabar valores nulos donde se haya especificado y no permitirá avanzar con la captura si se pretende agregar nulo a un campo donde se exige un valor. La Base de Datos está diseñada para que se respeten las restricciones referentes de valores nulos y las llaves primarias que representen un valor único en la Base de Datos. La Base de Datos está vinculada para respetar las reglas de integridad, que representa un candado para la seguridad de la misma.

Tabla 13. Ejemplo de datos de lo que la bitácora permite almacenar. Fuente: Propia basada en BD SIGAF real.

id Consecutivo	tabla Tabla	usuario Usuario	operacion Operación realizada	fecha Fecha realización cambio	registro Registro insertado/Borrado	registro_old Registro Borrado / Anterior
1	periodo_prog_edu	tempo	I	2014-07-24 00:00:00	1 SEMESTRAL	NULL
2	periodo_prog_edu	tempo	I	2014-07-24 00:00:00	2 CUATRIMESTRAL	NULL
3	periodo_prog_edu	tempo	I	2014-07-24 00:00:00	3 TRIMESTRAL	NULL
4	periodo_prog_edu	tempo	I	2014-07-24 00:00:00	4 SABATINO	NULL
5	directivopuesto	tempo	I	2014-07-24 00:00:00	1 DIRECTOR	NULL
6	directivopuesto	tempo	I	2014-07-24 00:00:00	2 SUBDIRECTOR	NULL
7	directivopuesto	tempo	I	2014-07-24 00:00:00	3 ADMINISTRADOR	NULL
8	directivosua	tempo	I	2014-07-24 00:00:00	1 1	NULL
9	programaedu	tempo	I	2014-07-24 00:00:00		NULL
10	programaedu	tempo	I	2014-07-24 00:00:00		NULL
11	programaedu	tempo	I	2014-07-24 00:00:00		NULL
12	programaedu	tempo	I	2014-07-24 00:00:00		NULL
13	programaedu	tempo	I	2014-07-24 00:00:00		NULL
14	planestudiodesc	tempo	I	2014-07-24 00:00:00	20092 PLAN DE ESTUDIOS 2009-2 20 2009-01-01 2015-1...	NULL
15	planestudio	tempo	I	2014-07-24 00:00:00	20092 5	NULL
16	planestudio	tempo	I	2014-07-24 00:00:00	20092 6	NULL
17	uaprendizaje	tempo	I	2014-07-24 00:00:00	11236 COMUNICACION ORAL Y ESCRITA 1 0 2 0 0 0 1 4 ...	NULL
18	uaprendizaje	tempo	I	2014-07-24 00:00:00	11237 CONTABILIDAD 2 0 3 0 0 0 2 7 0 1 1 1 20092	NULL
19	uaprendizaje	tempo	I	2014-07-24 00:00:00	11238 DESARROLLO HUMANO 1 0 2 0 0 0 1 4 0 1 3 1 8 ...	NULL
20	uaprendizaje	tempo	I	2014-07-24 00:00:00	11239 ENTORNO GLOBAL DE LOS NEGOCIOS 2 0 2 0 0 0 2 ...	NULL
21	uaprendizaje	tempo	I	2014-07-24 00:00:00	11240 MATEMATICAS 2 0 2 0 0 0 2 6 0 1 3 1 22 20092	NULL
22	uaprendizaje	tempo	I	2014-07-24 00:00:00	11241 RESPONSABILIDAD SOCIAL 2 0 1 0 0 0 2 5 0 1 3 ...	NULL
23	uaprendizaje	tempo	I	2014-07-24 00:00:00	11242 TECNOLOGIAS DE LA INFORMACION 1 3 0 0 0 0 1 ...	NULL
24	uaprendizaje	tempo	I	2014-07-24 00:00:00	11243 CONTABILIDAD APLICADA 2 0 3 0 0 0 2 7 0 1 1 ...	NULL
25	uaprendizaje	tempo	I	2014-07-24 00:00:00	11244 DERECHO 3 0 0 0 0 0 3 6 0 1 3 1 19 20092	NULL

La base de datos almacena información histórica, que permite analizar información anterior que puede consultarse para permitir entre otras cosas realizar comparaciones o visualizar el comportamiento de docentes en el pasado o de algunas materias.

Los administradores del sistema se verán beneficiados por el hecho de que el sistema maneje una bitácora de movimientos, lo que le da un agregado en cuanto a seguridad, porque como cualquier sistema, está sujeto a errores humanos. La bitácora brindará la oportunidad de identificar las personas o procesos que realizaron movimientos no válidos, lo que permitiría detectar posibles errores que podrían existir en cualquier nivel.

Para llegar a lo anterior, fue necesario realizar las sentencias necesarias para implementarlas en el Manejador de Base de Datos MySQL. Como ejemplo, se muestra la codificación en SQL para la entidad uaprendizaje:

```
--
-- Estructura de tabla para la tabla `uaprendizaje`
--

DROP TABLE IF EXISTS `uaprendizaje`;
CREATE TABLE IF NOT EXISTS `uaprendizaje` (
  `uaprendizaje` int(11) NOT NULL COMMENT 'Código uaprendizaje (SOLO NUMERICOS)',
```

```

`descripcionmat` varchar(60) CHARACTER SET latin1 COLLATE latin1_spanish_ci NOT NULL
COMMENT 'Descripción uaprendizaje',
`HC` int(11) NOT NULL COMMENT 'Horas Clase',
`HL` int(11) NOT NULL COMMENT 'Horas Laboratorio',
`HT` int(11) NOT NULL COMMENT 'Horas Taller',
`HPC` int(11) NOT NULL COMMENT 'Horas',
`HCL` int(11) NOT NULL COMMENT 'Horas Clinica',
`HE` int(11) NOT NULL COMMENT 'Horas Extra',
`creditos` int(11) NOT NULL COMMENT 'Numero de Créditos\n1Hr Clase= 2c',
`fec_aut` date DEFAULT NULL COMMENT 'Fecha autorización',
`observa` varchar(60) CHARACTER SET latin1 COLLATE latin1_spanish_ci DEFAULT NULL
COMMENT 'Observaciones',
`semestre_sug` int(11) NOT NULL COMMENT 'Semestre sugerido',
`caracter` int(11) NOT NULL COMMENT 'Codigo caracter para especificar
1=Obligatoria/2=Optativa',
`reqseriacion` int(11) NOT NULL COMMENT 'Codigo req seriacion para especificar
Obligada, Sugerida, s/seriación',
`etapa` int(11) NOT NULL COMMENT 'Código etapa para especificar Basica, Disciplinario,
Terminal',
`coordinaciona` int(11) NOT NULL COMMENT 'Codigo coordinacion area',
`claveD` int(11) DEFAULT NULL COMMENT 'Clave materia dependiente.',
`plan` int(11) NOT NULL COMMENT 'Codigo plan estudio',
`users_id` int(11) NOT NULL,
PRIMARY KEY (`uaprendizaje`),
UNIQUE KEY `UA_clave_UNIQUE` (`uaprendizaje`),
KEY `fk_uaprendizaje_caracter1` (`caracter`),
KEY `fk_uaprendizaje_reqseriacion1` (`reqseriacion`),
KEY `fk_uaprendizaje_etapas1` (`etapa`),
KEY `fk_uaprendizaje_coordinacional` (`coordinaciona`),
KEY `fk_uaprendizaje_uaprendizaje1` (`claveD`),
KEY `fk_uaprendizaje_planestudio1` (`plan`),
KEY `fk_uaprendizaje_users1` (`users_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='<double-click to overwrite multiple
objects>';

```

Capítulo VI

Conclusiones y Trabajos Futuros

En este capítulo se analizarán las conclusiones derivadas de la realización del presente trabajo, las cuales se presentan en el apartado 6.1, así como en los trabajos futuros en el apartado 6.2

6.1 Conclusiones

La Base de datos realizada en el manejador de base de datos MySQL, se adecuó de forma transparente al lenguaje de desarrollo y al diseño de interface, para lograr una base de datos documentada y funcional que incluye el proceso de Horarios dentro de la FCA de la UABC, campus Tijuana, programas de licenciatura, acorde a los requerimientos especificados en el análisis.

El desarrollo del presente proyecto, permitió trabajar en conjunto con otras etapas como análisis, diseño, desarrollo, implementación, pruebas y documentación, para lograr su integración y llegar al mismo objetivo que fue marcado al inicio del proyecto y en el cual es requisito que todas las etapas coincidan así como sus artefactos que garantizan el funcionamiento del sistema. Para lograr lo anterior, fue necesario un trabajar en conjunto, que requirió revisiones, correcciones y cambios constantes con el personal involucrado.

El seguir una metodología, significó una base sólida para llevar a cabo el proyecto con orden, y aunque los cambios son impredecibles en este tipo de proyectos, existió flexibilidad en ese aspecto.

Parte de la implementación de la Base de Datos, incluyó las pruebas que se realizaron para garantizar que la información alimentada así como la estructura, sirviera de base para las otras etapas del proyecto, mediante la revisión constante de la información con el personal involucrado así como la generación de versiones completas que permitió verificar el funcionamiento adecuado de la Base de Datos.

6.2 Trabajos futuros

El trabajo de la presente tesis permitirá que en un futuro la información que el sistema almacene pueda servir de base para la gestión de otros procesos, dentro de los que se pueden mencionar por ejemplo el proceso de calendarización de exámenes, o el de gestión de inasistencia docentes, procesos para los que el presente trabajo deja la posibilidad de beneficiarse con información que comparten y que estaría actualizada. Así como se pueden integrar procesos, la información también

puede crecer, y está diseñada de forma que pueda ser escalable. En los párrafos siguientes se explican las posibilidades de crecimiento más importantes que la Base de Datos esta en posibilidades de soportar.

- La BD del sistema, soporta información referente a la Facultad de Contaduría y Administración campus Tijuana, pero esta información puede ser ampliada para agregar nuevas unidades académicas pertenecientes a la UABC así como nuevos campus de acuerdo a nuevos requerimientos. Para las nuevas unidades académicas, existe una alta probabilidad de que la información referente a los planes de estudio que manejan sea diferente a los planes de estudio de las unidades académicas existentes. Por ese motivo, dentro de las nuevas unidades académicas se agregarán programas educativos actualizados asociados a sus respectivos planes de estudio, cada uno conformado por sus propias materias con sus características propias, que sería necesario dar de alta, para agregar a las ya existentes de otros programas.
- La forma en que la disponibilidad se conforma, se lleva a cabo por facultad. Por ese motivo cada facultad sería la encargada de conformar sus propios grupos, posiblemente con una numeración diferente a la manejada por la FCA de la UABC.
- En cuanto a la información que se maneja para la disponibilidad docente, la posibilidad de que la información de interés para docentes pueda ser aprovechadas en nuevas unidades académicas es amplia, porque la estructura maneja información general que puede ser de interés para cualquier rubro, como datos personales, perfil académico, documentación que avala dicho perfil, perfil de capacitaciones, así como disponibilidad docente en cuanto a días y horas, que es el aspecto más relevante del módulo disponibilidad docentes.

La Base de Datos ofrece ventajas desde diferentes perspectivas, por ejemplo, desde el punto de vista de un administrador, la información que resulte de mayor interés puede ser la que se refiere al personal involucrado en el proceso de horarios, principalmente los docentes, perfil académico, antigüedad o edad, lo que propiciará toma de decisiones por ejemplo en cuanto al docente idóneo para cubrir determinada vacante de alguna materia en particular. Existe otra información para realizar análisis estadísticos del personal docente o análisis de los perfiles profesionales de los docentes.

La base de datos puede permitir la toma de decisiones sobre el status del proceso de horarios a través de información de la carga académica, perfil y disponibilidad docente o los horarios

conformados, para identificar por ejemplo, los grupos cargados, los docentes que alimentaron su disponibilidad o los docentes que ya están dados de alta dentro del sistema, entre otra información.

Bibliografía

- Aderemi O, A., Babatunde A, S., & Montaz, A. (2008). A heuristic solution to the university timetabling problem. *Engineering Computations. Engineering Computations: International Journal for Computer-Aided Engineering and Software*, 26, 972-984.
- Alarcón, A., & Sandoval, E. (2008). Herramientas CASE para ingeniería de Requisitos. *Cultura científica JDC*, 74-75.
- Arias, M., Carrero, D., Grimón, F., & Morales, O. (2006). Software para la evaluación online bajo el enfoque de procesos. *Anales de la Universidad Metropolitana*, 6(2), 53-74.
- Barra, C. P. (1998). Software e Ingeniería de Software. *Revistamarina*, 1-14.
- Barrios, W. G., Godoy, M. V., Fernández, M. G., Mariño, S. I., Ferreira, F. M., & Zarrabeitia, C. T. (2012). SCRUM: Application experience in a software development PyME in the NEA. *JCS&T*, 12, 110-115.
- Belloso, C. I. (2009). Monografía sobre la metodología de desarrollo de software, rational unified process (RUP). El Salvador, El Salvador.
- Bender, C. M., Deco, C., González S., J. S., Hallo, M., & Ponce G., J. C. (2014). *Tópicos avanzados de Bases de Datos* (1a ed.). Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn).
- Codina, L. (1997 йил Noviembre). Una propuesta de metodologia para el diseño de bases de datos documentales (Parte I). *El Profesional de la Información*, 23-29.
- Codina, L. (Febrero de 1998). Metodología de análisis de sistemas de información y de diseño de bases de datos documentales: aspectos lógicos y funcionales. *ResearchGate*, 195-207.
- Connolly, T. M., & Begg, C. E. (2005). *Sistemas de bases de datos, un enfoque práctico para el diseño*,. Madrid: Pearson Educational Limited.
- Cuadra, D., Castro, E., Iglesias, A. M., Martínez, P., Calle, F. J., De Pablo, C., . . . Moreno, L. (2009). *Desarrollo de Bases de Datos: casos prácticos desde el análisis a la implementación*. México, DF: RA-MA Editorial.
- Dammak, A., Elloumi, A., & Kamoun, H. (2007). An enterprise system component based on graph colouring for exam timetabling a case study in a Tunisian university. *Engineering Computations: International Journal for Computer-Aided Engineering and Software, University of Sfax*, 1(3), 255-270. Retrieved 2013 йил Nov
- Dammak, A., Elloumi, A., & Kamoun, H. (Sept de 2008). Course timetabling at a Tunisian University: A case study . (J. o. Engineering, Ed.) *Journal of Systems Science and Systems Engineering*, 17(3), 334-352.

Recuperado el Nov de 2013, de http://www-labs.iro.umontreal.ca/~ferland/Publications/J_Syst_Sci_Sys_Eng.pdf

- Desiré Atangana, M., & Sepúlveda Lima, R. (2007). Roles y responsabilidades en entornos de ingeniería inversa de bases de datos. *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal Scientific Information System*, 58.
- Elmazri, R., & Navathe, S. B. (2007). *Fundamentos de sistemas de bases de datos*. Madrid, España: Pearson Educacion, S.A. Obtenido de <http://www.scribd.com/doc/105839623/Fundamentos-de-Sistemas-de-Bases-de-Datos-Elmasri-amp-Navathe#scribd>
- Flórez F., H. A. (Enero-Junio de 2009). Procesos de ingeniería de software, Software engineering processes. *Vínculos*, 6(1), 26-39.
- Gelbstein, E. (2011). La integridad de los datos: el aspecto más relegado de la seguridad de la información. *ISACA journal*, 6, 1-6.
- Greenough, C., & Worth, D. (n.d.). Software Development process n CSED: The future? *Computational Science & Engineering*.
- Hendrik, D. (2004). Integridad de datos relacionales. *Actualidad TIC, Revista del Instituto Tecnológico de Informática*, 1, 10-13.
- Hernández G., A. (2004). Identificación de procesos de negocio. *Industrial*, XXV(3), 84-88.
- Hernández, R., Miranda P., J., & Rey, P. A. (2008). Programación de horarios de clases y asignación de salas para la Facultad de Ingeniería de la Universidad Diego Portales mediante un enfoque de programación entera. *Revista Ingeniería de sistemas*, XXII, 122.
- Jiménez H., E. M. (22 de Octubre de 2012). Meta (Metodología tradicional y ágil), un metodología híbrida para desarrollo de software web propuesta para las empresas de software en México. D.F., D.F., México.
- Juárez S., J. F. (Julio de 2009). Sistema para gestión de una base de datos modular, segura e inteligente, utilizando como medio de comunicación la red eléctrica. 30-33. México, DF, México: Instituto politécnico nacional. Recuperado el 21 de 05 de 2015, de <http://www.cic.ipn.mx/sitioCIC/images/sources/cic/tesis/A070240.pdf>
- Kroenke, D. M. (2003). *Procesamiento de bases de datos. Fundamentos, diseño e implementación* (8 ed.). Naucalpan de Juárez, Edo. de México, México: Pearson educación de México, S.A. de C.V. Recuperado el 20 de 11 de 2015, de https://books.google.com.mx/books?id=7ORUWItwcNEC&pg=PA121&lpg=PA121&dq=normalizacion+de+base+de+datos&source=bl&ots=KvONOReMsQ&sig=vO0C6oPJ9IEiSfjgg_4P5VgEI4&hl=en&sa=X&ved=0CCoQ6AEwAjgUahUKEwis2qLZhe7GAhUJNYgKHWg3C7I#v=onepage&q=normalizacion%20de%20ba

- Lara T., J. A. (2010). *Marco de descubrimiento de conocimientos para datos estructuralmente complejos con énfasis en el análisis de eventos en series temporales*. Madrid.
- López Q., J. (2007). *Domine PHP y MySQL (Programación dinámica en el lado del servidor)*. México: AlfaOmega Grupo Editor.
- Mannino, M. V. (2007). *Administración de Bases de Datos, Diseño y desarrollo de aplicaciones*. México, DF: McGraw-Hill Interamericana.
- May, M., Morales, Y., Marrufo, J., & Martín, M. (2013). Implementación de un sistema para el control de activos ISOPTEC, bajo el estándar ITIL y metodología ágil SCRUM. *Congreso Interdisciplinario de Cuerpos Académicos. II*, págs. 176-181. Guanajuato: ECORFAN®.
- Maza A., G. L. (2003). Análisis, diseño e implementación de un sistema de información como soporte a la gestión académica para la escuela tecnológica de la universidad nacional de Piura. Piura, Perú, Perú: Universidad nacional de Piura. Recuperado el 20 de Noviembre de 2014, de <http://www.eumed.net/libros-gratis/2009c/607/>
- Mejía C., J. M. (Junio 2010). *Asignación de horarios de clases universitarias mediante algoritmos evolutivos*. (Vol. 9). Asociación Colombiana de Facultades de Ingeniería -ACOFI-.
- Menéndez, R., & Barzanallana, A. (2011 jul 13-oct). *www.um.es*. From Universidad de Murcia: <http://www.um.es/docencia/barzana/IAGP/IAGP2-Ingenieria-software-introduccion.html>
- Meneses E., C. A., Gálvez C., J. A., & Chavarro P., J. C. (2011). Especificación de temporalidad en modelos conceptuales para bases relacionales y orientadas a objetos. *Scientia et Technica*, 131.
- Microsoft. (2013 de Diciembre de 07). *Microsoft*. Recuperado el 2015 de Julio de 21, de <https://support.microsoft.com/es-es/kb/283878>
- Navarro C., A., Fernández M., J. D., & Morales V., J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *11*, 30-39.
- Pérez T., A., & Soler S., J. (s.f.). Metodología para el aprendizaje del diseño de bases de datos relacionales. 2. Obtenido de <http://www.revistaluz.rimed.cu/articulospdf/edicion28/amajo.pdf>
- Peter, R., & Coronel, C. (2005). *Sistemas de Bases de Datos*. México, DF: International Thompson Learning, Inc.
- Real Academia Española. (2012). *Real academia española*. Recuperado el 8 de Julio de 2015, de Real academia española: <http://lema.rae.es/drae/?val=iterativo>
- Rendón G., Á. (Agosto de 2004). *Desarrollo de sistemas informáticos usando uml y rup una visión general*. Popayán, Cauca, Colombia: Universidad del Cauca - Departamento de telemática.
- Rodríguez de León, J. I. (s.f.). Tema 6. Restricciones a la Base de Datos: Integridad y Seguridad. 3. Obtenido de <http://s3.amazonaws.com/UNED/apuntes/Tema6.pdf>

- Rodríguez, I., Ceballos, A., Hernández, J., & Prieto, F. (2007). Diseño e Implementación de una base de datos de Imágenes de rango con decimación para visualización en WEB. *Revista Ingenierías, Universidad de Medellín*, 181-198.
- Romero, P. (2005 йил 16-Mayo). Un barco-factoría de 'software' en alta mar, el colmo de la 'deslocalización'. *elmundo.es*.
- Ross, N. C., & Haddad, H. M. (2010). Software process improvement and metrics adoption in small. *JISTP Kennesaw State University, Volume 3(6)*, 6-29.
- Satpathy, T. (2013). *Una guía para el conocimiento de scrum (guía sbook tm)* (edicion 2013 ed.). Phoenix, Arizona: SCRUMstudy tm.
- Trentin, G. (1992). Estructura y Organización de una Base de Datos. *CL&E*, 81-83.
- Zapata J., C. M., González C., G., & Chaverra M., J. J. (2011). Generación automática del diagrama entidad-relación y su representación en sql desde un lenguaje controlado (un-lecep). *Revista Ingenierías Universidad de Medellín*, 10, 129.
- Zepeda S., L. Z. (2008 йил 01-06). Tesis Doctoral: Metodología para el diseño conceptual de almacenes de datos. Valencia, Valencia, España.

Glosario

Portable

Es un programa que puede ser ejecutado sin necesidad de una instalación. Puedes llevarlos en cualquier dispositivo extraíble (USB, memorias, etc.)

Multiplataforma

Software que funciona en diversas plataformas, por ejemplo: Algún software que funcione en diversos sistemas operativos, como Linux, Windows, en Mac OS X.

Datawarehousing

Es una copia de los datos transaccionales específicamente estructurada para la consulta y el análisis que proviene de distintas plataformas de la compañía. Esta información se utiliza como soporte para la toma de decisiones gerenciales.

Data Mining

Es la técnica de explotar la información y poder entender y solucionar los problemas de un negocio. Es la manera de ver datos que pareciera que están escondidos y podemos ver gráficas y tendencias. Existen diversas aplicaciones para data mining, y entre los ejemplos, podemos citar tendencias de ventas, relación existente entre clientes, perfiles de clientes, segmentación de mercado, etc. La información generada sirve para la toma de decisiones.

Automatización de tareas

Si es necesario realizar tareas periódicas (Como ejemplo: ejecutar antivirus, realización de respaldos, realizar algún mantenimiento, etc.), y éstas van a realizarse en el período que se requiera sin necesidad de la intervención de una persona para su realización

Escalabilidad

Es la capacidad de un sistema, o una red de poder aumentar su tamaño, manejar el crecimiento continuo, adaptarse sin comprometer su funcionamiento calidad.

Multithreading

Es la capacidad de una aplicación de realizar más de una tarea a la vez (múltiples hilos de ejecución), Así podemos decir a nivel programación, por ejemplo que un hilo se puede encargar de interactuar con el usuario, otro podría realizar cálculos, y otro se podría encargar de la comunicación.

Modelo

Descripción analógica que se utiliza para visualizar algo, abstracción de un objeto del mundo real.

Modelo de datos.

Representación gráfica relativamente simple de estructuras de datos del mundo real.

Programa Educativo.

Se refiere a las carreras que tienen presencia en la UABC. Por ejemplo: Licenciatura en informática.

Unidad Académica.

La UABC esta segmentada en unidades académicas, a cada una de las cuales se le asigna un director, subdirector y administrador y es responsable en gran medida de sus propias carreras, postgrados y especialidades. Por ejemplo, la unidad académica de la Facultad de Contaduría y

Administración maneja las carreras de licenciatura en informática, licenciatura en contaduría, licenciatura en administración de empresas y negocios internacionales, así como también algunos postgrados y especialidades.