



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TESIS

**Automatización del Sistema de Medición de Desempeño
para la toma de decisiones estratégicas de negocio:
caso de estudio**

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTA

ANGEL ROBERTO SÁNCHEZ PÉREZ



DIRECTOR DE TESIS: DR. DANIEL TREJO MEDINA

CIUDAD UNIVERSITARIA ENERO 2015

Dedicatoria

A mi madre y padre quienes forjaron lo mejor de ellos en mi con inmenso amor, confianza absoluta y apoyo incondicional.

A mi hermana quien siempre ha sido cómplice en mis más grandes retos y motivadora de nuevos desafíos.

A mi hermano menor por obligarme a ser un ejemplo a seguir y a pesar de eso siempre darme una lección.

A mis padrinos por guiarme con sabiduría absoluta a través de sus invaluable consejos y siempre estar presentes en cada logro de mi vida.

A mi compañera de vida por creer siempre en mi y darme la oportunidad de ser mejor cada día.

A la Universidad Nacional Autónoma de México: productora de mentes creativas, punto de encuentro de ideas libres y generadora de cambios inteligentes.

1. INTRODUCCIÓN	10
2. ANTECEDENTES	12
2.1 Beneficios de un sistema de inteligencia de negocios	14
2.2 Herramientas comerciales de inteligencia de negocios	16
2.3 ¿Cuándo es necesaria la inteligencia de negocios?	23
2.4 Arquitectura de la inteligencia de negocios	25
2.5 Ciclo del análisis de inteligencia de negocios	28
2.6 Principales métricas de un sistema de inteligencia de negocios	29
2.6.1 Características de las métricas	31
2.6.2 Indicadores de desempeño	32
2.6.2.1 Principales características de los KPIs	34
2.6.3 Presentaciones visuales de las métricas	35
2.6.3.1 Tableros de control	36
2.6.3.2 Cuadros de mando	37
2.7 Caso de estudio	37
3. PROBLEMA A RESOLVER	40
4. OBJETIVO GENERAL	40
5. HIPÓTESIS	41
6. JUSTIFICACIÓN	42
6.1 Identificación de problemas, oportunidades y objetivos	43
7. MARCO DE REFERENCIA	46
7.1 Componentes básicos para el desarrollo de inteligencia de negocios	46
7.1.1 Sistemas operacionales	48
7.1.2 Depósito de datos	57
7.1.2.1 Esquema estrella	60
7.1.2.2 Esquema copo de nieve	61
7.1.3 Métodos de extracción de conocimiento	71
7.2 Aplicaciones de la minería de datos	79
7.3 Análisis y diseño de sistemas	84
7.3.1 Sistemas de procesamiento de transacciones	84
7.3.1.1 Sistemas de automatización de oficinas	85
7.3.1.2 Sistemas de información administrativa	85

7.3.1.3	Sistemas de soporte de decisiones	86
7.3.1.4	Sistemas de decisiones en grupo y sistemas de trabajo colaborativo asistido por computadora	86
7.3.1.5	Sistemas de soporte para ejecutivos	86
7.3.2	Ciclo de vida de desarrollo de sistemas	87
7.3.2.1	Identificación de los problemas, oportunidades y objetivos	88
7.3.2.2	Determinación de los requerimientos de información del factor humano	89
7.3.2.3	Análisis de las necesidades del sistema	90
7.3.2.4	Diseño del sistema recomendado	91
7.3.2.5	Desarrollo y documentación del software	92
7.3.2.6	Prueba y mantenimiento del sistema	93
7.3.2.7	Implementación y evaluación del sistema	93
7.3.3	Proceso de desarrollo para un proyecto ágil	94
8.	DESARROLLO	97
8.1	Análisis de la investigación preliminar	97
8.1.1	Factibilidad técnica	99
8.1.2	Factibilidad económica	103
8.1.3	Factibilidad operativa	104
8.2	Determinación de requerimientos y análisis de las necesidades del sistema	106
	Etapa I: entrevistas	106
	Etapa II: cuestionarios	107
	Etapa III: observación	111
8.2.1	Clasificación de los requerimientos	114
8.2.1.1	Requerimientos funcionales - módulo operativo	114
8.2.1.2	Requerimientos no funcionales – módulo operativo	115
8.2.1.3	Requerimientos funcionales – módulo de análisis	115
8.2.1.4	Requerimientos no funcionales – módulo de análisis	116
8.3	Diseño del sistema	116
8.3.1	Arquitectura de la solución	116
8.3.1.1	Diseño de la arquitectura de la solución	117
8.3.1.2	Diagrama de base de datos	119
8.3.1.3	Diseño de interfaz gráfica	121
9.	RESULTADOS	122
10.	CONCLUSIONES	142
12.	ANEXOS	149
12.1	Historia de usuario: módulo operativo	149
12.2	Historia de usuario: módulo de análisis	152
12.3	Especificación de requisitos del software (SRS, Software Requirements Specification)	154
12.4	Descripción del diseño del software (SDD, Software Design Document)	161

12.5	Declaración de trabajo (SOW, Software Statement of Work)	170
12.6	Estimación presupuestaria	177
12.7	Configuración del equipo donde funciona la solución	178
12.8	Código Fuente	179
	Clase Address	179
	Clase AgendaCurso	182
	Clase Alumno	185
	Clase AlumnoNivel	192
	Clase AlumnoRequisito	194
	Clase GastoCurso	196
	Clase Licencia	199
	Clase Nivel	202
	Clase ReciboPago	205
	Clase RequisitoAuditoria	209
	Clase Rol	211
	Clase Usuario	213

ÍNDICE DE CUADROS

CUADRO 1 PROVEEDORES EVALUADOS: INFORMACIÓN DE PRODUCTO. FUENTE: ADAPTADO DE FORRESTER RESEARCH, INC.	18
CUADRO 2 RESULTADOS DEL CRITERIO DE INVESTIGACIÓN: OFERTA ACTUAL. FUENTE: ADAPTADO DE FORRESTER RESEARCH, INC.	19
CUADRO 3 RESULTADOS DEL CRITERIO DE INVESTIGACIÓN: ESTRATEGIA. FUENTE: ADAPTADO DE FORRESTER RESEARCH, INC.	20
CUADRO 4 RESULTADOS DEL CRITERIO DE INVESTIGACIÓN: PRESENCIA EN EL MERCADO. FUENTE ADAPTADO DE FORRESTER RESEARCH, INC.	20
CUADRO 5 NÚMERO DE HORAS POR ACTIVIDAD DE DESARROLLO. FUENTE: ELABORACIÓN PROPIA.	104
CUADRO 6 SOLUCIONES DEL SISTEMA PARA LOS PRINCIPALES PROBLEMAS DE LA ORGANIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	139
CUADRO 7 BENEFICIOS DERIVADOS DEL CUMPLIMIENTO DE LOS OBJETIVOS DEL PROYECTO. FUENTE: ELABORACIÓN PROPIA	141

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1 ARQUITECTURA TÍPICA DE UN SISTEMA DE INTELIGENCIA DE NEGOCIOS. FUENTE: ADAPTADO DE (VERCELLIS, 2009).	26
ILUSTRACIÓN 2 COMPONENTES PRINCIPALES DE UN SISTEMA DE INTELIGENCIA DE NEGOCIOS. FUENTE: ADAPTADO DE (VERCELLIS, 2009).	27
ILUSTRACIÓN 3 CICLO DE UN ANÁLISIS DE INTELIGENCIA DE NEGOCIOS. FUENTE: ADAPTADO DE (VERCELLIS, 2009).	28
ILUSTRACIÓN 4 MAPA MENTAL DE LA INVESTIGACIÓN. FUENTE: ELABORACIÓN PROPIA.	41
ILUSTRACIÓN 5 PROBLEMAS, OPORTUNIDADES Y OBJETIVOS (CASO DE ESTUDIO). FUENTE: ELABORACIÓN PROPIA.	44
ILUSTRACIÓN 6 DIAGRAMA DE ESTADO DE UNA TRANSACCIÓN. FUENTE: ADAPTADO DE (ELMASRI, 2007).	51
ILUSTRACIÓN 7 ENTORNO DE UN SISTEMA DE BASE DE DATOS SIMPLIFICADO. FUENTE: ADAPTADO DE (ELMASRI, 2007).	54
ILUSTRACIÓN 8 REPRESENTACIÓN METAFÓRICA DE UN CUBO DE DATOS. FUENTE: ADAPTADO DE (WREMBEL, 2007).	59
ILUSTRACIÓN 9 EJEMPLO DE UN ESQUEMA ESTRELLA QUE CONTIENE UNA TABLA DE HECHOS Y DIVERSAS TABLAS DE DIMENSIONES. FUENTE: ADAPTADO DE (MOURYA, 2013).	61
ILUSTRACIÓN 10 ESQUEMA COPO DE NIEVE NORMALIZADO EN MÚLTIPLES TABLAS DE BÚSQUEDA. FUENTE: ADAPTADO DE (MOURYA, 2013).	62

ILUSTRACIÓN 11 ARQUITECTURA DE TRES NIVELES (INFERIOR, MEDIO Y SUPERIOR) DE UN ALMACÉN DE DATOS. FUENTE: ADAPTADO DE (MOURYA, 2013).	65
ILUSTRACIÓN 12 ARQUITECTURA MOLAP FUENTE: ADAPTADO DE (MOURYA, 2013).	69
ILUSTRACIÓN 13 ARQUITECTURA HOLAP. FUENTE: ADAPTADO DE (MOURYA, 2013).	70
ILUSTRACIÓN 14 PROCESO DE KDD. FUENTE: ADAPTADO DE (HERNÁNDEZ ORALLO, 2004).	73
ILUSTRACIÓN 15 EL PROCESO DE DESCUBRIMIENTO DE CONOCIMIENTO. FUENTE: ADAPTADO DE (MOURYA, 2013).	75
ILUSTRACIÓN 16 DISCIPLINAS QUE CONTRIBUYEN A LA MINERÍA DE DATOS. FUENTE: ADAPTADO DE (HERNÁNDEZ ORALLO, 2004).	77
ILUSTRACIÓN 17 LAS SIETE FASES DEL CICLO DE DESARROLLO DE SISTEMAS. FUENTE: ADAPTADO DE (KENDALL & KENDALL, 2011).	88
ILUSTRACIÓN 18 LAS CINCO ETAPAS DEL MODELO DE DESARROLLO ÁGIL. FUENTE: ADAPTADO DE (KENDALL & KENDALL, 2011).	95
ILUSTRACIÓN 19 INFRAESTRUCTURA DE RED DEL CASO DE ESTUDIOS. FUENTE: ELABORACIÓN PROPIA.	100
ILUSTRACIÓN 20 DIAGRAMA DE FLUJO DE PRIMER NIVEL PARA EL PROCESO DE REGISTRO DE NUEVOS ALUMNOS. FUENTE: ELABORACIÓN PROPIA.	108
ILUSTRACIÓN 21 DIAGRAMA DE FLUJO DE PRIMER NIVEL PARA EL PROCESO DE REINSCRIPCIÓN DE UN ALUMNO. FUENTE: ELABORACIÓN PROPIA.	108
ILUSTRACIÓN 22 DIAGRAMA DE FLUJO DE PRIMER NIVEL PARA EL PROCESO DE ACTUALIZACIÓN DE DATOS DE UN ALUMNO. FUENTE: ELABORACIÓN PROPIA.	108
ILUSTRACIÓN 23 DIAGRAMA DE FLUJO DE PRIMER NIVEL PARA EL PROCESO DE CONTROL DE REQUISITOS PARA SER INSTRUCTOR. FUENTE: ELABORACIÓN PROPIA.	109
ILUSTRACIÓN 24 DIAGRAMA DE FLUJO DE PRIMER NIVEL PARA EL PROCESO DE INVITACIÓN AL SIGUIENTE CURSO INMEDIATO DE LOS ALUMNOS.	109
ILUSTRACIÓN 25 DIAGRAMA DE FLUJO DE PRIMER NIVEL DEL PROCESO DE CONSULTA DE INSTRUCTORES ACTIVOS PARA DAR CURSO. FUENTE: ELABORACIÓN PROPIA.	110
ILUSTRACIÓN 26 DIAGRAMA DE FLUJO DE PRIMER NIVEL DEL PROCESO DE ELABORACIÓN DEL REPORTE CON LOS PRINCIPALES KPI'S DE LA ORGANIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	110
ILUSTRACIÓN 27 DIAGRAMA DE FLUJO DE SEGUNDO NIVEL PARA EL PROCESO DE REGISTRO DE NUEVOS ALUMNOS. FUENTE: ELABORACIÓN PROPIA.	111
ILUSTRACIÓN 28 DIAGRAMA DE FLUJO DE SEGUNDO NIVEL PARA EL PROCESO DE REINSCRIPCIÓN DE UN ALUMNO. FUENTE: ELABORACIÓN PROPIA.	112

ILUSTRACIÓN 29 DIAGRAMA DE FLUJO DE SEGUNDO NIVEL PARA EL PROCESO DE ACTUALIZACIÓN DE DATOS DE UN ALUMNO. FUENTE: ELABORACIÓN PROPIA.	112
ILUSTRACIÓN 30 DIAGRAMA DE FLUJO DE SEGUNDO NIVEL PARA EL PROCESO DE CONTROL DE REQUISITOS PARA SER INSTRUCTOR. FUENTE: ELABORACIÓN PROPIA.	112
ILUSTRACIÓN 31 DIAGRAMA DE FLUJO DE SEGUNDO NIVEL PARA EL PROCESO DE INVITACIÓN AL SIGUIENTE CURSO INMEDIATO DE LOS ALUMNOS.	113
ILUSTRACIÓN 32 DIAGRAMA DE FLUJO DE SEGUNDO NIVEL DEL PROCESO DE CONSULTA DE INSTRUCTORES DISPONIBLES PARA DAR CURSO. FUENTE: ELABORACIÓN PROPIA.	113
ILUSTRACIÓN 33 ARQUITECTURA DE J2EE APPLICATION SERVER. FUENTE: ADAPTADO DE (ORACLE, DOCS ORACLE).	117
ILUSTRACIÓN 34 PROCESAMIENTO DE UNA SOLICITUD DE UN CLIENTE POR UN SERVIDOR J2EE. FUENTE: ADAPTADO DE (ORACLE, DOCS ORACLE).	118
ILUSTRACIÓN 35 DIAGRAMA DE LA BASE DE DATOS. FUENTE: ELABORACIÓN PROPIA.	120
ILUSTRACIÓN 36 INTERFAZ GRÁFICA. FUENTE: ELABORACIÓN PROPIA.	121
ILUSTRACIÓN 37 INGRESO AL SISTEMA MEDIANTE USUARIO Y CONTRASEÑA. FUENTE: ELABORACIÓN PROPIA.	123
ILUSTRACIÓN 38 INFORMACIÓN PERSONAL Y DE CONTACTO DE LOS USUARIO DEL SISTEMA. FUENTE: ELABORACIÓN PROPIA.	124
ILUSTRACIÓN 39 VISTA DE LOS USUARIOS DEL SISTEMA. FUENTE: ELABORACIÓN PROPIA.	125
ILUSTRACIÓN 40 VISTA DEL CATÁLOGO DE NIVELES DE LA ORGANIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	126
ILUSTRACIÓN 41 VISTA DEL CATÁLOGO DE REQUISITOS PARA AUDITORIAS DE LA ORGANIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	127
ILUSTRACIÓN 42 VISTA GENERAL SOBRE LOS ALUMNOS DE LA ORGANIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	128
ILUSTRACIÓN 43 VISTA SOBRE LA FUNCIONALIDAD DE BÚSQUEDA DEL SISTEMA. FUENTE: ELABORACIÓN PROPIA.	129
ILUSTRACIÓN 44 VISTA SOBRE LA INFORMACIÓN PERSONAL DE UN ALUMNO. FUENTE: ELABORACIÓN PROPIA.	130
ILUSTRACIÓN 45 VISTA SOBRE LA INFORMACIÓN DE CONTACTO DE UN ALUMNO. FUENTE: ELABORACIÓN PROPIA.	130
ILUSTRACIÓN 46 VISTA DE LOS RECIBOS DE PAGO POR ALUMNO. FUENTE: ELABORACIÓN PROPIA.	131
ILUSTRACIÓN 47 VISTA DEL CONSECUTIVO DE RECIBOS DE PAGO. FUENTE: ELABORACIÓN PROPIA.	132
ILUSTRACIÓN 48 VISTA DE LA INFORMACIÓN DE UN RECIBO DE PAGO. FUENTE: ELABORACIÓN PROPIA.	132
ILUSTRACIÓN 49 VISTA DE LOS INSTRUCTORES DE LA ORGANIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	133

ILUSTRACIÓN 50 VISTA DE LAS LICENCIAS DEL PERSONAL DE LA ORGANIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	134
ILUSTRACIÓN 51 VISTA SOBRE LA INFORMACIÓN DE LAS LICENCIAS DE CADA INSTRUCTOR. FUENTE: ELABORACIÓN PROPIA.	135
ILUSTRACIÓN 52 VISTA DE CONTROL DE INTERVALO DE TIEMPO PARA LOS REPORTES DEL SISTEMA. FUENTE: ELABORACIÓN PROPIA.	135
ILUSTRACIÓN 53 VISTA SUPERIOR DE UN REPORTE DEL SISTEMA DESARROLLADO. FUENTE: ELABORACIÓN PROPIA.	136
ILUSTRACIÓN 54 VISTA MEDIA DE UN REPORTE DEL SISTEMA DESARROLLADO. FUENTE: ELABORACIÓN PROPIA.	137
ILUSTRACIÓN 55 VISTA SUPERIOR DE UN REPORTE DEL SISTEMA DESARROLLADO. FUENTE: ELABORACIÓN PROPIA.	138

1. Introducción

La formación que brinda la Facultad de Ingeniería permite analizar conceptos de negocio, que si bien no están directamente relacionados con el plan de estudios, si mantienen una relación muy estrecha, pues en su mayoría el éxito de dichos conceptos está fuertemente ligado a la evolución de las tecnologías de la información.

El perfil de egreso del estudiante de la carrera de Ingeniería en Computación de la Facultad de Ingeniería de la UNAM, lo comprometen a encontrar soluciones para las diferentes áreas de desarrollo de nuestro país por ello, al término de la formación profesional, a fin de obtener un título académico, es necesario elaborar un proyecto que atienda algún problema del entorno.

Con referencia en los fundamentos anteriores existe un tópico que a lo largo del desarrollo de mi carrera se mantuvo presente y que hoy en día el mundo laboral exige un número significativo de ingenieros capacitados para promover la innovación, es así como el enfoque técnico de esta tesis está orientado al concepto de la inteligencia de negocios, un término que ha desempeñado un papel fundamental dentro del área administrativa de múltiples empresas en nuestro país y el mundo.

La presente investigación pretende ahondar sobre la comprensión del concepto de la inteligencia de negocios, el trabajo revela como es que el termino surge a través de la historia, que necesidades pretende subsanar y como ha evolucionando para posicionarse como una herramienta indispensable para distintas organizaciones.

Otro punto clave del trabajo es clarificar el procedimiento para generar conocimiento a partir de grandes volúmenes de datos, analizando las técnicas, metodologías y herramientas de software que la ingeniería ha puesto a disposición en el mercado para obtener cada vez mejores resultados.

Un trabajo de esta índole en donde se pretende profundizar la concepción sobre la relación entre la ingeniería y la inteligencia de negocios da pie a encontrar áreas de oportunidad en donde la ingeniería pueda ofrecer distintas soluciones que además de optimizar el proceso de inteligencia de negocios permitan llevar el concepto a más sectores empresariales.

Durante el desarrollo de la tesis se logró identificar una área de oportunidad en donde verter las habilidades desarrolladas a partir del plan de estudios en un proyecto ligado al concepto de la inteligencia de negocios, a partir de este hecho el reto más importante se convirtió en encontrar un caso de estudio adecuado para la construcción de un proyecto que permitiera obtener resultados de valor para el tesista.

La adecuada selección del caso de estudio beneficio al aporte técnico de la tesis mediante la elaboración de un proyecto de desarrollo de software que logró optimizar los procesos que apoyan la toma de decisiones estratégicas particulares del caso de estudio. Dejando claro que el plan de estudios de la carrera brinda bases sólidas para encarar el campo de la inteligencia de negocios e iniciar un proceso de especialización en cualquiera de las herramientas que ofrece el mercado.

Finalmente es importante puntualizar que la tesis puede ser referente para futuros ingenieros en entender el contexto actual de la inteligencia de negocios en el campo de la ingeniería, conocer los principales agentes de cambio, proveedores, herramientas y características, lo que puede significar un apoyo importante dentro de la formación o especialización de su trayectoria profesional.

2. Antecedentes

En este apartado se analiza el contexto actual de la inteligencia de negocios como eje central de investigación, la evolución del concepto a través del tiempo y como es que las tecnologías de la información han aportado valor a dicho desarrollo.

Es imprescindible reconocer las contribuciones hechas por un sistema de inteligencia de negocios a las diversas organizaciones que han adoptado estas herramientas como parte de sus estrategias de crecimiento, analizar las soluciones propuestas por los proveedores líderes en el mercado e identificar los componentes básicos que integran dicho sistema.

En la actualidad el término inteligencia de negocios está presente en la mayoría de las empresas sin importar el giro en el que se desarrollen y como consecuencia de la evolución de las tecnologías de la información a tomado un importante posicionamiento dentro de la toma estratégica de decisiones de negocios.

El proceso de inteligencia de negocios, que actualmente se considera crítico en la mayoría de las empresas no debe interpretarse como un término nuevo, Curto Díaz (2010) en su obra *Introducción al Business Intelligence* hace un par de citas importantes a considerar sobre las primeras definiciones formales del concepto:

Hans Peter Luhn, investigador de IBM acuñó el término en Octubre de 1958 en el artículo *A Business Intelligence System* como:

“La habilidad de aprender las relaciones de hechos presentados de forma que guíen las acciones hacia una meta deseada.”

Howard Dresden en 1989, analista de Gartner, propone la siguiente definición:

“Conceptos y métodos para mejorar las decisiones de negocio mediante el uso de sistemas de soporte basados en hechos.”

En su obra el autor menciona la necesidad de tener mejores, más rápidos y más eficientes métodos para extraer y transformar los datos de una organización en información, como el agente acelerador de la evolución de la inteligencia de negocios.

Una organización puede ser rica en datos y pobre en información, sino sabe identificar, resumir y categorizar los datos Madnick (1993). El mercado de la inteligencia de negocios, BI por sus siglas en inglés (*Business Inteligencie*) se ha visto marcado por una clara evolución que lo destaca como un mercado maduro, en donde los principales agentes del mercado (SAP, IBM, Microsoft) han generado una consolidación mediante la compra de empresas pequeñas (Curto Díaz, 2010).

La introducción de soluciones a código abierto y la aparición de nuevas empresas con foco en la innovación cubren las múltiples necesidades del mercado como la visualización, el análisis predictivo, los dispositivos virtuales y la inteligencia de negocios en tiempo real, para Curto Díaz (2010) la integración de estos componentes ha sido fundamental para la maduración del mercado.

Curto Díaz (2010) define el concepto de inteligencia de negocios como:

“El conjunto de metodologías, aplicaciones, prácticas y capacidades enfocadas a la creación y administración de información que permite tomar mejores decisiones a los usuarios de una organización.”

El proceso de toma decisiones de negocio tiene como objetivo principal la generación de ventajas competitivas que aseguren un posicionamiento firme en el mercado frente a la competencia, la construcción de un sistema de inteligencia de negocios requiere la integración de recursos humanos, técnicos y de cómputo para

orientar y realizar las diversas labores de investigación, diseño y programación necesarias (Peña Ayala, 2006).

De tal manera hoy en día las empresas han otorgado un valor especial al uso de sistemas de información que contribuyen eficazmente en la toma de decisiones de negocio sin embargo, es importante puntualizar que la evolución de la inteligencia de negocios como parte de una competencia entre los líderes del mercado ha generado múltiples características entre las diversas herramientas del mercado por tanto, no es suficiente tener información para sustentar la toma de decisiones ya que es igualmente necesario evaluar los beneficios implícitos ofrecidos por los proveedores.

2.1 Beneficios de un sistema de inteligencia de negocios

Existen diversos artículos sobre los beneficios de la implementación de un sistema de inteligencia de negocios, en estos se puede identificar las mejoras al proceso de toma de decisiones derivadas del uso adecuado de dichos sistemas. Es indiscutible el aumento significativo del número de organizaciones interesadas en utilizar este tipo de soluciones como consecuencia de la efectividad que han demostrado a través del tiempo.

Curto Díaz (2010) en su libro *Introducción al Business Intelligence* enuncia los siguientes beneficios y aportaciones derivados de la integración de un sistema de inteligencia de negocios:

- Permitir una visión única, conformada, histórica, persistente y de calidad de toda la información.
- Crear, manejar y mantener métricas, indicadores clave de rendimiento (KPI, *Key Performance Indicators*) e indicadores clave de metas (KGI, *Key Goal Indicators*, KGI).

- Aportar información actualizada tanto en nivel agregado como en detalle.
- Mejorar la comprensión y documentación de los sistemas de información en el contexto de una organización.
- Mejorar la competitividad de la organización como resultado de ser capaces de:
 - Diferenciar lo relevante sobre lo superfluo.
 - Acceder más rápido a la información.
 - Tener mayor agilidad en la toma de decisiones.
- Crear un círculo virtuoso de información, en donde los datos se transforman en información que permite tomar mejores decisiones que se traducen en mejores resultados y generan nuevos datos.

Estos beneficios han sido alcanzados a través de la integración de los avances en materia de tecnologías de la información dentro de los sistemas de inteligencia de negocios tales como:

La evolución de las telecomunicaciones, la inclusión de los dispositivos móviles, los múltiples avances en materia de construcción de hardware que han generado nuevas técnicas de procesamiento y almacenamiento de la información como: la virtualización de servidores, almacenamiento de grandes volúmenes de datos en memoria y el procesamiento en la nube por citar algunos ejemplos.

La oferta de soluciones es cada vez mayor, los principales proveedores desarrollan nuevas herramientas y versiones cada vez más intuitivas y amigables para los usuarios pero aún más importante en la actualidad los líderes del mercado se

encuentran enfocados en desarrollar soluciones atractivas y de bajo costo que colaboren con el crecimiento de las pequeñas organizaciones.

Sin embargo, es importante analizar los conceptos vertidos en el párrafo anterior, los líderes del mercado son empresas internacionales como IBM, Oracle y SAP por citar sólo algunos de ellos, la situación en cuanto a la adaptabilidad de sus soluciones para el mercado de las pequeñas empresas mexicanas no ha sido cien por ciento satisfactoria.

Al parecer la concepción de estas empresas internacionales sobre las pequeñas empresas mexicanas no es adecuada, ya que no existe comparación directa entre las pequeñas empresas mexicanas y las pequeñas empresas originarias de los diferentes países de los líderes del mercado.

Lo anterior podría ser un factor de análisis para generar nuevas soluciones que entiendan las necesidades propias de la pequeña empresa mexicana y dar pie al desarrollo de nuevas soluciones hechas en México para resolver problemas específicos en favor de este sector que no pueden quedar excluido de los avances tecnológicos en materia de inteligencia de negocios.

2.2 Herramientas comerciales de inteligencia de negocios

Existen múltiples productos, servicios y precios que intentan atender las cambiantes necesidades y gustos de los clientes, la investigación sobre los principales proveedores del mercado del BI para esta tesis tomó como referencia el artículo *Agile Business Intelligence Platforms* que el autor Boris Evelson (2014), publicó para el *Forrester Wave*.

En el artículo del autor se evalúan a dieciséis proveedores como los más dominantes del negocio del BI: Actuate, Birst, GoodData, IBM, Information Builders,

Microsoft, MicroStrategy, Oracle, Panorama Software, Pentaho, Qlik, SAP, SAS, Tableau Software, Tibco JasperSoft, and Tibco Software.

El artículo evaluó las fortalezas y debilidades de los proveedores enunciados en el párrafo anterior desde la perspectiva de lo que el autor denomina como: *Agil BI Platforms*. El autor es claro en determinar que este término no debe confundirse con los términos: intuitivo o amigable al usuario; es así como define los siguientes criterios de evaluación para su investigación:

- **Oferta actual:** se evaluó la oferta actual de cada proveedor, considerando todas las características y capacidades. También se evaluó una breve demostración por cada vendedor delineando sus capacidades clave y se realizó una encuesta a un total de 112 clientes de los proveedores.
- **Estrategia:** se revisó la estrategia de cada proveedor y se consideró que tan buenos son los planes de cada uno de estos para la optimización de sus productos a fin de satisfacer las futuras demandas de sus clientes. También se evaluaron los recursos financieros y humanos que la empresa tiene disponibles para apoyar su estrategia.
- **Presencia en el mercado:** En esta investigación en particular, sólo se consideraron los ingresos de los proveedores de BI como una medida de la presencia en el mercado mundial.

En el siguiente cuadro se muestran los proveedores y sus correspondientes productos y versiones evaluadas (Evelson, 2014).

Proveedor	Producto evaluado	Versión
Actuate	BIRT iHub, BIRT onDemand, BIRT Analytics	iHub 3, BA 4.3
Birst	Birst Enterprise Edition	Release 5.10
GoodData	GoodData	Release 101

IBM	Cognos BI	10.2.1
Information Builders	8	8
Microsoft	▪ Excel ▪ SQL Server ▪ SharePoint ▪ Power BI for Office	2013 2014 2013 365
MicroStrategy	MicroStrategy Analytics Platform	9.4.1
Oracle	Oracle Business Intelligence Enterprise Edition	11.1.1.7.0
Panorama Software	Necto	14
Pentaho	Business Analytics	5.1
Qlik	QlinkView	11.2
SAP	▪ SAP Lumira ▪ SAP BusinessObjects Explorer	1.16 4.1
SAS	▪ SAS Visual Analytics ▪ SAS Enterprise BI Server ▪ SAS Office Analytics	6.4 4.4 6.1
Tableau Software	Tableau Software	8.2
Tibico Jaspersoft	Jaspersoft BI Enterprise Edition	5.6
Tibico Software	Spotfire	6.5

Cuadro 1 Proveedores evaluados: información de producto. Fuente: Adaptado de Forrester Research, Inc.

Los resultados documentados dentro del artículo del *Forrester Wave* sustentan una investigación sólida sobre las características actuales que brindan los líderes en el negocio del BI, sus estrategias y su presencia en el mercado; la investigación otorga una calificación en cada rubro permitiendo hacer una comparativa entre los distintos proveedores simplificando la comprensión del alcance que estos pueden otorgar a las diversas organizaciones.

La investigación sobre el criterio de **oferta actual** a la que fueron sometidos los dieciséis proveedores evaluó características tales como: la automatización del sistema, la funcionalidad avanzada de visualización de datos, la integración de los datos entre otros.

Los resultados de este criterio se muestran a continuación el siguiente cuadro:

	Ponderación de Forrester	Acutuate	Birst	GoodData	Information Builders	Tibico	Jaspersoft	Microsoft	MicroStrategy	Panorama Software	Pentaho	Qlik	SAP	SAS	Tableau Software	Tibico Software
OFERTA ACTUAL	50%	3.3	3.32	2.53	3.81	2.63	4.3	3.7	3.15	2.98	4	4.2	3.9	3.45	4.46	
TI - autoservicio de usuarios habilitado	5%	4	5	3	5	2	5	5	4	3	5	5	5	5	5	
Aprovisionamiento de autoservicio de usuarios	15%	5	4	2	4	2	5	4	3	4	5	4	5	3	5	
Autoservicio de usuarios - integración de los datos	10%	3	4	3	5	4	4	4	3	4	5	5	4	3	4	
Autoservicio de usuarios - interfaz de usuario	20%	2	2	2	4	2	5	3	3	2	4	4	3	3	5	
Automatización de Inteligencia de negocios	15%	2	2	2	2	2	3	2	2	2	2	4	3	2	4	
Visualización de contenido de datos avanzados	5%	3	4	3	3	3	3	3	4	3	3	4	3	4	3	
Funcionalidad de visualización de datos avanzados	15%	5	4	3	4	3	5	5	3	3	4	4	5	5	4	
Agil BI - Agilidad de respuesta	5%	1	2	3	3	2	3	3	3	2	3	5	3	2	5	
Agil BI - Retroalimentación del cliente	4%	4	5	3	4	4	4	5	5	4	5	4	4	5	4	
Autoservicio de usuarios - retroalimentación del cliente	3%	4	4	3	5	5	4	5	5	5	5	4	4	5	5	
ADV - retroalimentación del cliente	3%	4	5	4	5	4	4	5	5	4	5	4	4	5	5	

Cuadro 2 Resultados del criterio de investigación: oferta actual. Fuente: adaptado de Forrester Research, Inc.

En cuanto a la **estrategia** que siguen los líderes en el mercado del BI se consideraron puntos a evaluar como: la dirección del producto, retroalimentación de los clientes, el compromiso del departamento de investigación y desarrollo entre otros.

Estas evaluaciones nos dan un panorama general de la visión de la empresa para con sus productos pero sobre todo para con los consumidores, lo cual debe ser un punto importante a considerar antes de seleccionar cualquiera de estas herramientas. Los resultados de este criterio se muestran a continuación en el siguiente cuadro:

	Ponderación de Forrester	Acutuate	Birst	GoodData	Information Builders	Tibico	Jaspersoft	Microsoft	MicroStrategy	Panorama Software	Pentaho	Qlik	SAP	SAS	Tableau Software	Tibico	Software
ESTRATEGIA	50%	3.55	2.65	3.1	4.1	2.65	3.55	4.1	2.65	2.65	3.65	3.55	4.9	3.65	3.55		
Compromiso - R&D	45%	3	1	2	3	2	2	4	2	2	3	2	5	3	3		
Dirección del producto	45%	4	4	4	5	3	5	4	3	3	4	5	5	4	4		
Retroalimentación de los clientes sobre la visión del producto	10%	4	4	4	5	4	4	5	4	4	5	4	4	5	4		

Cuadro 3 Resultados del criterio de investigación: estrategia. Fuente: adaptado de Forrester Research, Inc.

Finalmente dentro de la evaluación que desarrolla el autor Evelson (2014) para el *Forrester Wave* interpreta los ingresos de los dieciséis proveedores como una medida de la **presencia en el mercado** mundial, en donde los resultados son presentados a través del siguiente cuadro:

	Ponderación de Forrester	Acutuate	Birst	GoodData	Information Builders	Tibico	Jaspersoft	Microsoft	MicroStrategy	Panorama Software	Pentaho	Qlik	SAP	SAS	Tableau Software	Tibico Software
PRESENCIA EN EL MERCADO	0%	2	1	1	3	1	5	3	3	1	1	3	5	5	2	2
Finanzas de la compañía	100%	2	1	1	3	1	5	3	3	1	1	3	5	5	2	2

Cuadro 4 Resultados del criterio de investigación: presencia en el mercado. Fuente adaptado de Forrester Research, Inc.

Una vez obtenidos los resultados el investigador Evelson generó una serie de perfiles de los principales proveedores los cuales clasificó en tres grupos que representan el nivel alcanzado por la evaluación de los criterios anteriores, para los fines de esta investigación de tesis a continuación se listan de forma ascendente los ocho primeros lugares obtenidos por los principales proveedores dentro de la investigación del *Forrester Wave*.

1. SAS
2. Tibico Spotfier
3. Information Builders
4. Microsoft
5. MicroStrategy
6. SAP
7. Qlik
8. Tableau

Esta investigación nos deja claro quiénes son los líderes en el mercado y nos da una visión objetiva de las ventajas competitivas que ofrecen los principales proveedores, lo cual agrega valor a las soluciones que ofrecen al consumidor; en un mercado creciente en donde las necesidades son cambiantes, esta información es útil para determinar el tipo de herramienta que necesita cada organización.

Es importante mencionar que existen otras opciones en el mercado del BI, en la actualidad se pueden encontrar diversas organizaciones que a través de un gran esfuerzo de investigación y desarrollo patrocinado por distintas entidades han logrado desarrollar software *open source* (código abierto) los cuales poco a poco han ido ganando terreno frente a las opciones comerciales posicionándose como opciones verdaderamente interesantes para considerar su implementación.

Algunas de las opciones más significativas del mercado de libre distribución que han desarrollado un nivel de madurez adecuado de sus productos y logrado el

reconocimiento y apoyo de grandes patrocinadores del medio de las tecnologías de la información son:

- **Pentaho:** Ofrece una suite de herramientas de inteligencia de negocios mediante dos ediciones una para la comunidad y otra para el sector empresarial lo que ha posicionado a *Pentaho* como una opción fuertemente posicionada en el sector de herramientas de tipo *open source* desde su fundación en el 2004.
- **BIRT:** Un proyecto de software de alto nivel de la fundación *Eclipse*, un consorcio independiente sin fines de lucro de proveedores de la industria de software y una comunidad de código abierto. Permite crear visualizaciones de datos e informes que pueden ser embebidos en aplicaciones web especialmente aquellas basadas en el lenguaje de programación *java*.
- **Spagobi:** Un proyecto de la iniciativa *SpagoWorld* el cual adopta el modelo de código abierto y permite la distribución de sus herramientas a través de una versión libre estable y en paralelo de una versión profesional.
- **Knime:** Es una herramienta de software de tipo modular que integra diversas herramientas para todo el proceso de análisis de datos: acceso a la información, transformación de los datos, análisis predictivo, visualización y presentación de informes.
- **Openreports:** Como su nombre lo indica esta es una herramienta de presentación de informes de código abierto que permite crear reportes dinámicos a través de un navegador web, existe una versión comercial la cual integra múltiples funcionalidades avanzadas.

Las primeras tres soluciones ofrecen herramientas avanzadas de inteligencia de negocios tales como: generador de reportes, análisis de modelos multidimensionales OLAP por sus siglas en inglés, integración y seguimiento de KPIs, tableros interactivos, procesamiento de información geográfica, minería de datos, tableros y consolas de control en tiempo real, herramientas de colaboración, herramientas de extracción, transformación y carga ETL por sus siglas en inglés entre otros.

En este punto de la investigación podemos determinar que, la implementación de un sistema de inteligencia de negocios implica el análisis de múltiples variables como por ejemplo: presupuesto del proyecto, infraestructura del cliente, niveles de servicio, necesidades primarias del cliente, proyecciones de crecimiento entre otros.

Derivado de lo anterior, es importante visualizar en qué momento es necesario considerar la implementación de estas herramientas dentro de una organización, quiénes son candidatos y por qué razón este tipo de herramientas aumentaría la asertividad de sus decisiones para administrar el negocio.

2.3 ¿Cuándo es necesaria la inteligencia de negocios?

Existen diversas situaciones en las que la implantación de un sistema de inteligencia de negocios resulta adecuada, por ejemplo (Curto Díaz, 2010):

- La toma de decisiones se efectúa de forma intuitiva en la organización.
- Identificación de problemas de calidad de información.
- Uso de hojas de cálculo como repositorios de información corporativos o de usuario.
- Necesidad de cruzar información de forma ágil entre departamentos.
- Evitar silos de información.

- Existe demasiada información en la organización para ser analizada de la forma habitual; se ha alcanzado la masa crítica de los datos.
- Es necesario automatizar los procesos de extracción y distribución de la información.

En definitiva los sistemas de *business intelligence* buscan responder las siguientes preguntas: ¿Qué pasó?, ¿Qué pasa ahora?, ¿Por qué pasó? y ¿Qué pasará? De tal forma sin importar el tamaño de la organización, si se experimentan las situaciones expuestas en la lista anterior es oportuno considerar un proyecto de implementación de un sistema de inteligencia de negocios.

El párrafo anterior deja expuesta una realidad del mercado mexicano pues el tipo empresarial predominante corresponde a las pequeñas y medianas empresas ,las cuales en su mayoría no cuentan con el personal adecuado para la implementación, administración y operación de estos sistemas, sin mencionar los elevados costos de las licencias comerciales y peor aún las opciones de libre distribución en su lucha por ser competitivas en el entorno comercial se vuelven soluciones cada vez más robustas y complejas lo que deja vulnerable a un sector importante de la actividad empresarial en México.

Sin embargo, este problema desde la perspectiva de la ingeniería debe y puede ser tratado como una área de oportunidad que hasta ahora se ha tratado de madurar mediante el desarrollo de soluciones a la medida pero el hecho es que este tipo de soluciones siguen siendo altamente costosas para las pequeñas empresas.

Si bien la solución probablemente no está en el desarrollo de software a la medida, no es del todo imposible pensar en un desarrollo de software como medida de introducción a este sector empresarial en el contexto de la inteligencia de negocios ya que, existen medios que pueden apoyar este tipo de desarrollos como las instituciones gubernamentales, universidades, entre otros.

El desarrollo de un proyecto académico que integre las bases del BI dentro de una organización que cubra características específicas del sector empresarial en cuestión puede traer diversos beneficios a dicha organización pero sobre todo cambiar la percepción que se tiene sobre el concepto del uso de herramientas de inteligencia de negocios que hoy día solo un cierto número de organizaciones utilizan dentro de sus actividades.

Para poder definir las características específicas del caso de estudio de esta tesis fue de vital importancia estudiar con más detalle los componentes básicos de un sistema de inteligencia de negocios a fin de identificar las necesidades técnicas y tecnológicas a las que debía enfrentarse el proyecto.

2.4 Arquitectura de la inteligencia de negocios

La arquitectura de cualquier sistema de inteligencia de negocios debe incluir los siguientes tres componentes (Vercellis, 2009):

- Fuentes de datos (*Data Sources*).
- Depósito de datos (*Data Warehouses and Datamarts*).
- Métodos de inteligencia de negocio (*Business Intelligence Methodologies*).

La justificación del autor sobre estos tres elementos básicos para cualquier sistema de inteligencia de negocios comienza en la necesidad absoluta de recolectar e integrar la información generada a través de las diferentes fuentes de datos (primarias y secundarias) dentro de una organización.

El hecho es que la información de las diversas fuentes de datos en la mayoría de las ocasiones es heterogénea en origen y tipo. Aun cuando la mayoría de la información proviene de sistemas operacionales, existe información no estructurada que puede jugar un rol importante dentro de las estrategias de negocio.

Ante esta situación, el uso de herramientas de extracción, transformación y carga permite homogeneizar la información dentro de almacenes de datos que soporten los procesos de análisis de inteligencia de negocios.

Finalmente, el uso de modelos matemáticos y metodologías de análisis de información son las encargadas de soportar las decisiones de negocio y es así como estos tres elementos definen la arquitectura básica para la inteligencia de negocios.

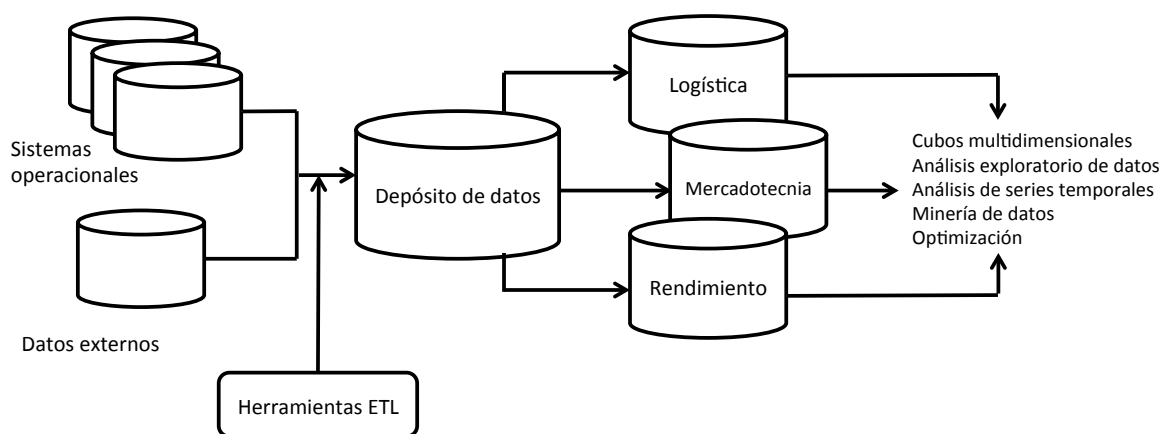


Ilustración 1 Arquitectura típica de un sistema de inteligencia de negocios.
Fuente: adaptado de (Vercellis, 2009).

La generación de conocimiento para la toma de decisiones de negocio es un proceso que define Carlo Vercellis a través de una serie de componentes o bloques. El siguiente esquema representa los principales componentes de un sistema de inteligencia de negocios de acuerdo con el autor (Vercellis, 2009).

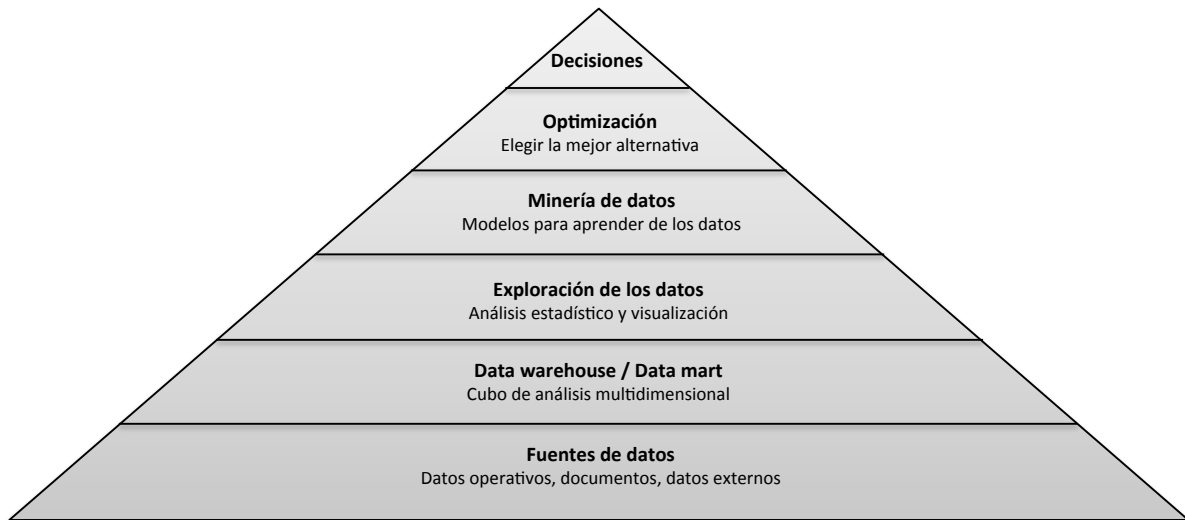


Ilustración 2 Componentes principales de un sistema de inteligencia de negocios. Fuente: adaptado de (Vercellis, 2009).

El fundamento de la pirámide está basado en la arquitectura de la inteligencia de negocios; de esta manera el autor detalla de una forma más explícita el proceso de soporte de decisiones basado en la generación de conocimiento.

Dentro dicha definición el autor hace énfasis sobre la utilidad de la exploración de datos, para después aplicar métodos estadísticos que verifiquen que las conclusiones del equipo de toma de decisiones sean adecuadas y respaldadas por la información.

El uso de modelos matemáticos para el reconocimiento de patrones, máquinas de aprendizaje y técnicas de minería de datos son un factor esencial para poder llegar al último nivel de la pirámide que corresponde a la toma y adopción de una decisión específica.

Una vez identificados los componentes principales de un sistema de inteligencia de negocios es de suma importancia entender cuáles son las interacciones que estos

tienen, con el propósito de comprender cómo se puede generar conocimiento que aporte valor en la toma de decisiones de una organización.

2.5 Ciclo del análisis de inteligencia de negocios

Es posible identificar un ciclo ideal sobre el análisis de la inteligencia de negocios. Este ciclo se conforma de cuatro fases: análisis, percepción, decisión y evaluación (Vercellis, 2009).

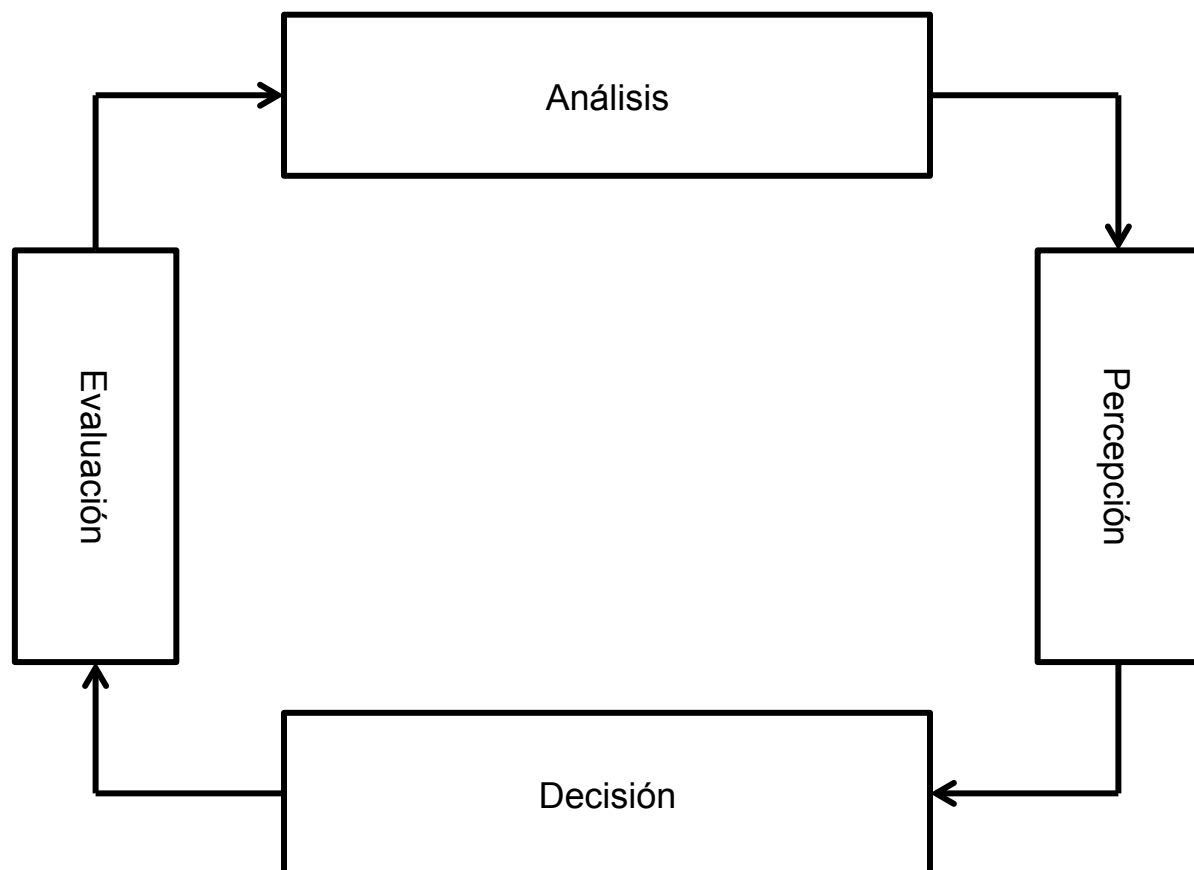


Ilustración 3 Ciclo de un análisis de inteligencia de negocios. Fuente: adaptado de (Vercellis, 2009).

A través de estas cuatro fases es claro que el autor determina un ciclo de inteligencia de negocios de forma genérica, partiendo del **análisis** donde la finalidad es que el personal de toma de decisiones defina claramente el problema y determine los datos a recopilar de la organización.

Una vez superada esta fase inicial el autor considera necesario un proceso que estudie minuciosamente los hechos y la situación previamente analizada, esto permite que puedan ser considerados todos los rumbos que puede tomar el caso de estudio y hacer un descubrimiento de patrones relevantes, a esta fase se le denominada como **percepción**. De esta manera el ciclo de análisis de la inteligencia de negocios llega a una etapa de ejecución, es decir, la toma de una **decisión**.

Al hablar de un ciclo es necesario contemplar un medio de medición, de tal forma el autor define la **evaluación** como el eje medular del ciclo, comparar los resultados obtenidos con los esperados facilita la comprensión del nivel de éxito del proceso así como los puntos de mejora para dar pie al inicio de un nuevo ciclo.

De esta manera la investigación del proyecto de tesis necesita profundizar sobre las métricas, conocer cuales son las más relevantes dentro de un sistema de inteligencia de negocios y que aportación dan al personal de toma de decisiones así como sus respectivas representaciones e interpretaciones.

2.6 Principales métricas de un sistema de inteligencia de negocios

Un elemento clave para la generación de inteligencia de negocios es el proceso de medición, y para los fines de esta investigación es importante entender qué es una métrica, cómo se define y cómo aporta valor en la toma de decisiones.

La definición más simple de una métrica es: algo que es medible; entonces, todo aquello que carece de una métrica no puede ser administrado, por ende no puede ser mejorado y difícilmente comprendido en su totalidad.

Las métricas pueden ser medidas y registradas como números, porcentajes, divisas, valoraciones (bueno, malo o neutral), características cualitativas vs cuantitativas, etc.

Las métricas son necesarias porque la aprobación de proyectos a menudo se basa en información insuficiente fundamentada en el rendimiento poco realista del retorno sobre la inversión o cálculos sobre el tiempo probable en que el proyecto alcanzará el punto de equilibrio (W. Hubbard, 2007).

Las métricas requieren una necesidad o un propósito, un objetivo a alcanzar o bien un punto de referencia pero sobre todo un significado para esa medición, una interpretación que siga una estructura de informes sin embargo, aun definiendo un conjunto de buenas métricas, la administración de las mismas puede fallar; algunas de las principales causas son (Kerzner, 2011):

- Un proceso ineficiente de toma de decisiones.
- Definir objetivos que necesiten más tiempo del estimado.
- Deficientes prácticas o métodos de administración de proyectos.
- Poco entendimiento sobre cómo utilizar las métricas.

Para lograr una adecuada definición de las métricas que debe contemplar una organización es importante conocer las características que las conforman con el fin de ser interpretadas de forma adecuada por el personal de toma de decisiones, quienes podrán tomar disposiciones oportunas para definir el rumbo de su organización.

2.6.1 Características de las métricas

A continuación se listan algunas características básicas que debe de cumplir una métrica (Kerzner, 2011):

- Tiene una necesidad o un propósito.
- Proporciona información útil.
- Se enfoca hacia un objetivo.
- Se puede medir con una precisión razonable.
- Refleja la verdadera situación del proyecto.
- Soporta gestión proactiva.
- Asistencias en la evaluación de la probabilidad de éxito o fracaso.
- Aceptado por las partes interesadas como herramienta para la toma de decisiones informada.

Existen diferentes tipos de métricas para la evaluación de un proyecto o proceso, dentro de las que destacan los indicadores de desempeño KPI por sus siglas en inglés, es por ello que el autor Harold Kerzner es muy claro al diferenciar una métrica de un KPI, ya que en la actualidad gran cantidad de las compañías utilizan una mezcla de ambos y los denominan KPIs.

De acuerdo con Kerzner (2011) la diferencia entre éstas es el período de tiempo de análisis. Las métricas se centran en un período de tiempo presente evaluando la situación actual del progreso, mientras que los KPIs se centran en resultados futuros es decir, la dirección final que puede tomar el proceso, de esta manera es importante hablar de las características y funcionalidades de los KPIs pues son un tipo de métrica de suma relevancia en cualquier organización.

2.6.2 Indicadores de desempeño

Como se vio en el párrafo anterior sobre las métricas, los *KPIs* o indicadores de desempeño son un tipo de métrica que proporciona información precisa sobre lo que podría ocurrir en un futuro si se mantiene la tendencia del proceso evaluado.

Erick T. Peterson (2006) en su libro *The Big Book of the Key Performance Indicators* determina que un indicador de desempeño no es más que un número (tasas, razones, promedios o porcentajes) diseñado para transmitir de manera sucinta la mayor información posible.

Un indicador bien definido pretende establecer expectativas para la generación oportuna de acciones que direccionen el proceso, proyecto u organización a una mejora continua (Peterson, 2006).

Es importante mencionar que si bien un indicador de desempeño es representado por un número, debe a bien diferenciarse o tener presente la diferencia entre un número sin contexto y un indicador de desempeño dentro de un reporte de análisis.

El hecho que un número no contenga un contexto, lo deja en desventaja con respecto a un indicador de desempeño sin embargo esto no quiere decir que los números crudos tengan que ser excluidos de un reporte de indicadores de desempeño.

Los números crudos son necesarios y a pesar de su carencia de contexto por si mismos juegan un papel fundamental en la proyección del contexto. Los indicadores de desempeño están diseñados para resumir datos dentro de un contexto de manera significativa (Peterson, 2006).

Si el objetivo de un sistema de medición de desempeño es mejorar la eficiencia y la eficacia, entonces podemos estar seguros que nuestra métrica es un indicador de

desempeño siempre y cuando refleje factores controlables, no tendría sentido la medición del proceso si los usuarios no pueden cambiar el resultado, esto es lo que reflejaría un número sin contexto.

La definición de indicador de desempeño bajo la percepción de Erick Peterson es la siguiente:

“Un KPI es una métrica que mide qué tan bien la organización o un individuo realiza una actividad operacional, táctico o estratégico que es crítico para el éxito actual y futuro de la organización” (Peterson, 2006).

Los KPIs se han utilizado en una variedad de industrias tales como:

- Construcción
- Gestión de Riesgo
- Seguridad
- Calidad
- Ventas
- *Marketing*
- Tecnologías de la Información
- Gestión de Cadenas de Suministro
- Organizaciones no lucrativas

David Parameter (2007) define tres categorías de indicadores de desempeño:

- **Los indicadores de resultados (RIS):** ¿Qué hemos logrado?
- **Indicadores de rendimiento (IP):** ¿Qué debemos hacer para aumentar o cumplir el rendimiento?
- **Indicadores clave de desempeño (KPI):** ¿Cuál es el rendimiento crítico?
Indicadores que pueden aumentar drásticamente el rendimiento o el logro de los objetivos.

2.6.2.1 Principales características de los KPIs

Wayne Eckerson (2006) en su libro *Performance Dashboards: Monitoring and Managing Your Business* establece doce características que deben de tener los indicadores de rendimiento efectivos:

- **Alineados:** los KPIs siempre están alineados con la estrategia y objetivos corporativos.
- **Propios:** cada KPI es "propiedad" de un individuo o grupo en la parte comercial quien es responsable de su resultado.
- **Predictivos:** los KPIs miden los elementos de valor para el negocio.
- **Procesables:** los KPIs, se alimentan con datos procesables oportunos para que los usuarios puedan intervenir para mejorar el rendimiento antes de que sea demasiado tarde.
- **Pocos en número:** los KPIs deben centrarse en tareas de alto valor para los usuarios, no dispersar su atención y energía en demasiadas cosas.
- **Fáciles de entender:** los KPIs deben de ser sencillos y de fácil comprensión, de nada sirve una base de índices complejos donde los usuarios no saben cómo influyen en el proceso de negocio.
- **Generadores de cambios:** el acto de la medición de un KPI debe dar lugar a una reacción en cadena de cambios positivos en la organización.
- **Estandarizados:** los KPIs se basan en las definiciones estándares, reglas y cálculos para que se puedan integrar a través de dashboards de toda la organización.

- **Contextualizados:** los KPIs miden el rendimiento con respecto a un contexto mediante la aplicación de los objetivos y umbrales de rendimiento para que los usuarios puedan medir su progreso en el tiempo.
- **Reforzados con incentivos:** las organizaciones pueden magnificar el impacto de los indicadores clave de rendimiento adjuntando una compensación o incentivos a ellos. Sin embargo, deben hacer esto con cautela, la aplicación de incentivos sólo para *KPIs* bien entendidos y estables.
- **Relevantes:** los KPIs pierden gradualmente su impacto en el tiempo, por lo que deben ser periódicamente revisados y renovados.

Hasta este punto de la investigación podemos reconocer cuales son las principales métricas de un sistema de inteligencia de negocios así como su utilidad para el personal de toma de decisiones, sin embargo es momento de profundizar entorno a los métodos de visualización de las métricas ya que este es un punto focal de todo el proceso de inteligencia de negocios pues es el destino final de todo el trabajo antes expuesto.

2.6.3 Presentaciones visuales de las métricas

La idea detrás de la representación visual de las métricas o indicadores de desempeño surge como consecuencia de los sistemas de soporte de decisiones en los años 70s, pero no es sino hasta finales de 1990 que con el avance tecnológico en materia de comunicaciones, las representaciones digitales fueron adoptadas por la mayoría de las organizaciones (Wayne W., 2006).

El hecho es que para el personal responsable de la toma de decisiones de negocio es fundamental generar elementos visuales que no solo faciliten la comprensión si no también permitan al usuario interactuar con las métricas a fin de establecer diversos escenarios que permitan establecer estrategias de negocio.

De esta manera es como los tableros de control (*dashboards*) y los cuadros de mando (*scorecards*) se han convertido en elementos indispensables dentro del ambiente empresarial. A continuación la investigación pretende entender las características de cada uno de ellos así como sus diferencias y principales usos.

2.6.3.1 Tableros de control

Wayne Eckerson (2006) define a un *dashboard* como una representación visual de métricas críticas o indicadores clave de desempeño, orientado a medir el desempeño en un enfoque operacional contra las metas y los umbrales a partir de datos de un determinado momento.

Es importante tener presente que un *dashboard* no es un reporte detallado, es por eso que en algunas ocasiones es necesario más de un *dashboard* para obtener la información adecuada.

El diseño eficiente de un *dashboard* no implica insertar en el diseño una gran cantidad de alertas sino todo lo contrario, a fin de no generar distracciones es indispensable identificar lo estrictamente necesario a través del entendimiento de lo que realmente necesita el usuario final, conocimiento sobre el proceso de obtención de las métricas y con qué frecuencia se actualizarán las mismas, así como tener claro cómo se actualizará el *dashboard* y, en la medida de lo posible, mantener una uniformidad en el diseño (Wayne W., 2006).

2.6.3.2 Cuadros de mando

En la actualidad existen algunas confusiones entre el uso de los *dashboards* y los *scorecards*; sin embargo, Eckerson Wayne (2006) define las diferencias entre estos de la siguiente manera.

Los *dashboard* son mecanismos de representación visual utilizados en un sistema de medición de desempeño, orientado operativamente a medir el desempeño contra las metas y los umbrales a partir de datos en el momento adecuado.

Los *scorecards* son representaciones visuales utilizados en un sistema de medición de desempeño encaminado estratégicamente para representar el progreso hacia el logro de las metas y objetivos estratégicos mediante la comparación de rendimiento frente a objetivos y umbrales.

Ambos *dashboards* y *scorecards* son mecanismos de visualización dentro de un sistema de medición de desempeño que transmiten información crítica. La principal diferencia entre estos es que los primeros monitorean procesos operacionales, como los utilizados en la gestión de proyectos, mientras que los *scorecards* trazan el progreso de los objetivos tácticos.

2.7 Caso de estudio

Una vez terminada la investigación de los antecedentes que marcan el desarrollo de esta tesis sobre los sistemas de inteligencia de negocios es momento de definir el alcance del proyecto. Dentro de la investigación se identificó una área de oportunidad entre los sistemas de inteligencia comerciales y los sistemas de inteligencia de distribución libre, de esta manera el proyecto pretende abordar un problema del sector empresarial, específicamente en la pequeña y mediana empresa mexicana que claramente tiene desventajas para la implementación, uso y administración de este tipo de herramientas.

Después de analizar las distintas herramientas de inteligencia de negocio y buscar los precios de dichas herramientas, se encontraron costos que van desde los 12 mil dólares hasta los 200 mil dólares, y en el caso de software libre había que aprender a utilizar las herramientas, todas las soluciones son muy interesantes, pero el tiempo de implementación y beneficio para el caso de estudio era un punto a analizar, con base en lo anterior se tenían varias peticiones de organizaciones civiles sin fines de lucro de tener un sistema de inteligencia que les permitiera dar un seguimiento a sus indicadores claves.

Para poder elegir un caso de estudio adecuado, y poder aportar valor a la misma, dentro de las organizaciones interesadas en la implementación de una herramienta de inteligencia de negocios que aportara beneficios a sus procesos de toma de decisiones fue necesario establecer los siguientes parámetros:

- Ser una persona moral registrada en el Registro Federal de Contribuyentes.
- Tener una antigüedad de operaciones de por lo menos 3 años.
- Tener una estructura organizacional bien definida.
- Tener claramente identificados los indicadores claves de la empresa.
- Contar con suficiente información digitalizada para la implementación del proyecto.
- Tener un problema claramente identificado en donde la implementación de un sistema de inteligencia de negocios pudiera ser parte de la solución.
- Tener la infraestructura de cómputo mínima necesaria para la implementación del proyecto.
- Ser una organización sin fines de lucro.

Los parámetros anteriores se definieron para facilitar la elección y ser más efectivos en la implementación de la solución, ya que si la organización no tenía información válida o de cierto tiempo no sería beneficioso para ella, ni para el autor de la tesis el lograr tener un efecto en el proyecto de manera positiva. Se siguió la metodología

vista en la materia de proyecto de investigación, para analizar proyectos y soluciones.

El elegir una organización sin fines de lucro fue determinante dado que, como nos enseñaron en la Facultad, debemos regresarle al país lo mas posible en beneficios a la sociedad, y esta era una buena manera de cerrar el ciclo de estudiante, ya que en este sentido la contribución de este proyecto académico aporta al desarrollo de las actividades de esta entidad que da valor a la sociedad civil mexicana.

La asociación civil seleccionada presta servicios desde el mes de Septiembre del año 2000 constituida como Amor en Ser A. C ante el Registro Federal de Contribuyentes y presta servicios de tipo asistencial y de beneficencia con base en los principios de la técnica de la energía universal. Uno de los objetivos de dicha institución es dar a conocer las técnicas de la energía universal a través de múltiples centros de capacitación alrededor de la República Mexicana.

El caso de estudio seleccionado para el desarrollo de esta tesis se encontraba en un proceso de centralización de la administración de todos sus centros de energía universal, lo que lo convirtió en un candidato adecuado para la implementación de un sistema de inteligencia de negocio ya que contaba con suficiente información histórica de forma digital y una serie de problemas para su administración pero sobre todo para la extracción de conocimiento que pudiera aportar valor al desarrollo de dicha empresa.

De esta manera se estableció contacto con la mesa directiva de la institución para la construcción del proyecto donde la finalidad era resolver un problema entorno al uso de alguna herramienta de inteligencia de negocio que aportara valor a las operaciones de la organización, de esta forma se construyó el proyecto con las siguientes características.

Durante las conversaciones con la organización se indicaron varios requerimientos, entre los que se encontraron que no contaban con un presupuesto definido para comprar licencias de software comercial, al analizar los requerimientos globales iniciales, se observó que no era necesario tampoco implementar un software de inteligencia de negocios de software libre, dado que era muy grande la implementación en tiempo y módulos para lo que necesitaban.

Considerando lo antes comentado y considerando que uno de los objetivos de una tesis es aplicar los conocimientos adquiridos en la carrera, se le sugirió a la organización hacer un sistema a la medida, considerando como referencia los sistemas de inteligencia de negocio comerciales y de software libre, pero, ajustado y alineado a los indicadores necesarios para la organización.

Después de conversarlo con el director de la asociación se llegó al acuerdo de ejecutarlo como un desarrollo de un sistema de mediación de desempeño, automatizando lo que ya tenían especificado en sus planes y manuales de operación.

3. Problema a resolver

La elaboración manual de reportes sobre los principales indicadores clave de desempeño del caso de estudio, dificulta la toma de decisiones del negocio, debido a que dentro de la organización no se tiene un sistema que facilite la misma toma de decisiones.

4. Objetivo General

Desarrollar una herramienta de software que automatice la elaboración de reportes sobre los principales indicadores de desempeño del caso de estudio.

5. Hipótesis

H_i: Si se desarrolla una herramienta de software que automatice la elaboración de reportes sobre los principales indicadores de desempeño del caso de estudio, facilitará la toma de decisiones del negocio.

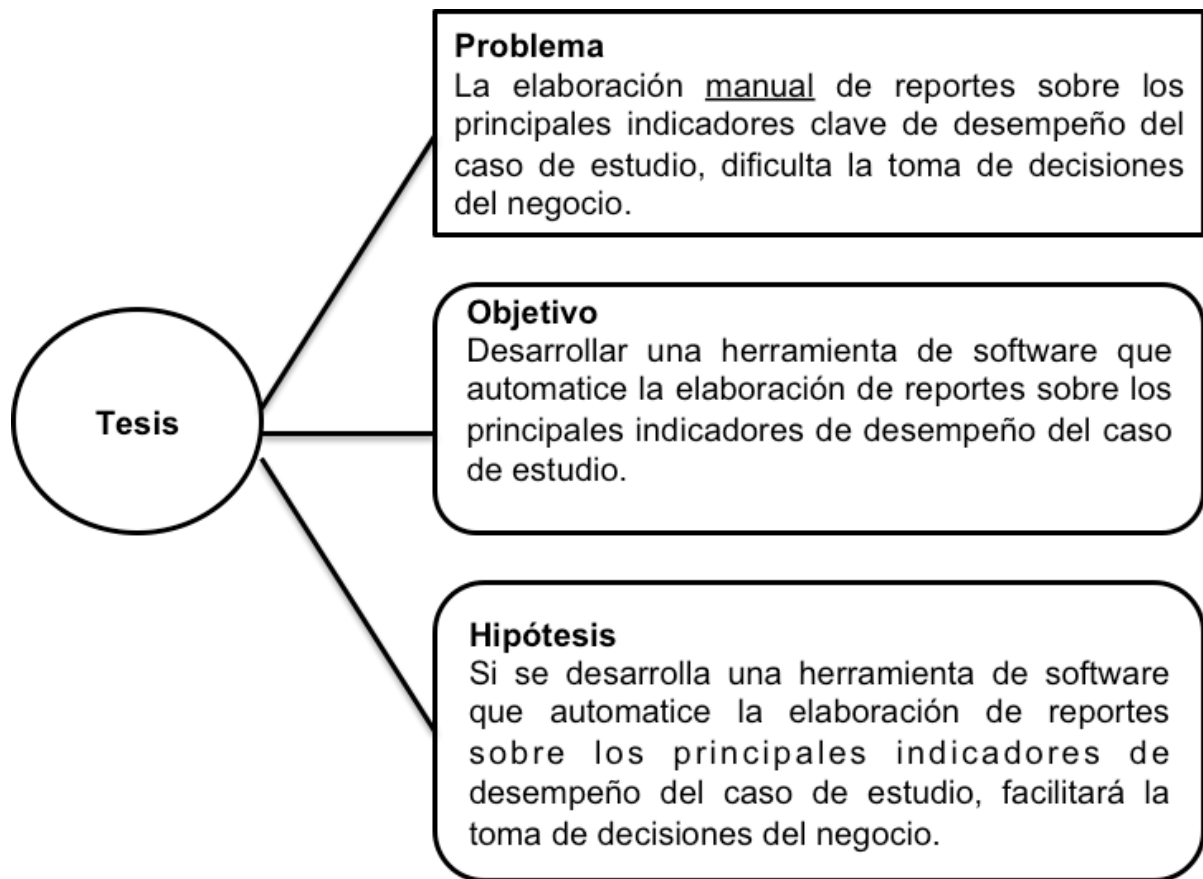


Ilustración 4 Mapa mental de la investigación. Fuente: elaboración propia.

6. Justificación

El caso de estudio presentado en esta tesis, necesitaba mejorar el control operativo de su negocio, optimizar sus métodos de almacenamiento y análisis de información histórica de su participación en el mercado, así como automatizar el proceso de generación de reportes que le permitan tomar decisiones oportunas y mejoren el desempeño del negocio.

Se planteó diseñar una herramienta de software que abordara los siguientes tres componentes de un sistema de inteligencia de negocios basado en las necesidades particulares del caso de estudio:

- Sistema operacional
- Depósito de datos
- Generador de reportes

En ese momento el caso de estudio contaba con un volumen significativo de datos almacenados en diversas fuentes donde predominaba el uso de sistemas de archivos (hojas de cálculo) lo que dificultaba la integración, manipulación y presentación de la información dando como resultado una deficiente administración de la misma.

Los principales problemas detectados por la organización en el uso de sus sistemas informáticos en ese momento eran los siguientes:

- Excesiva redundancia de la información que impacta negativamente en el desempeño del sistema actual.
- Deficiente integración de la información de la organización que dificulta la labor del personal operativo.
- Falta de control en la actualización de los datos que amenaza la integridad de la información.

- Dificultad para manipular la información del sistema lo que dificulta la ejecución de las diversas tareas del negocio.
- Realización manual de reportes que generan conocimiento para la toma de decisiones estratégicas del negocio.

De esta manera era imprescindible para la organización actualizar sus sistemas a fin de resolver los problemas identificados pues representaban un riesgo latente para la operación del negocio, por lo que la planeación, diseño e implementación de una herramienta de software basado en los tres elementos antes mencionados de un sistema de inteligencia de negocios era una posible solución para dicha organización.

6.1 Identificación de problemas, oportunidades y objetivos

La investigación preliminar dio como resultado que, para resolver el problema inicial y cumplir con el objetivo previamente definido, era necesario integrar un objetivo particular que facilitara el manejo de la información del caso de estudio, lo que beneficiaría en múltiples aspectos el desarrollo de la herramienta de reporte.

El resultado de la integración del nuevo objetivo particular y su respectiva hipótesis u oportunidad derivada del cumplimiento de dicho objetivo, se encuentra definido a través de la siguiente ilustración.

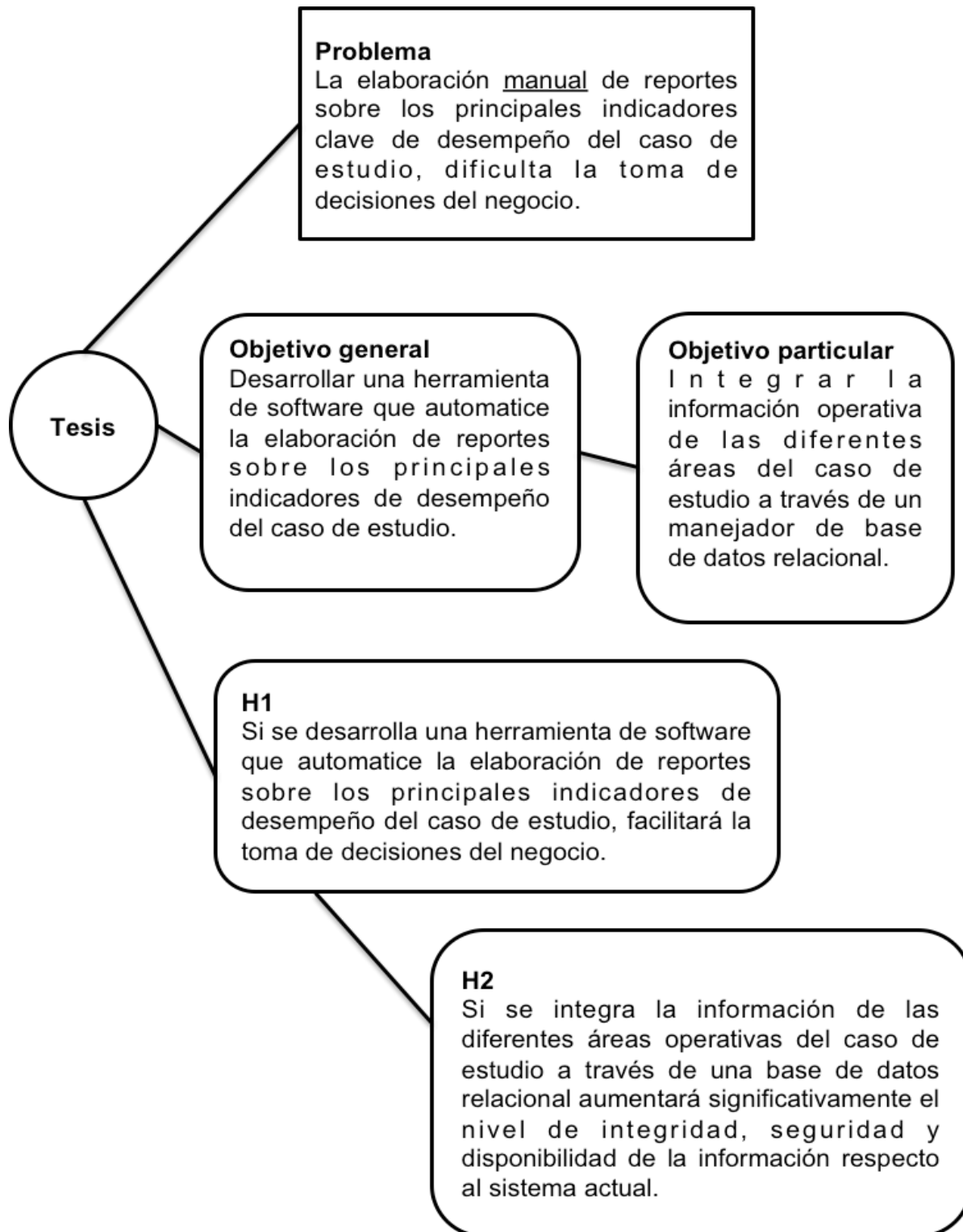


Ilustración 5 Problemas, oportunidades y objetivos (caso de estudio). Fuente: elaboración propia.

A partir de este punto el alcance del proyecto era claro, la solución propuesta para el problema era el desarrollo de un sistema con las características mínimas necesarias para que éste cubriera la función de un sistema de inteligencia de negocios para el caso de estudio en particular.

La selección de esta posible solución se dio a partir del análisis de la investigación de los antecedentes expuestos en el capítulo anterior, las características y necesidades del caso de estudio así como el presupuesto del proyecto fueron los principales factores que apoyaron el diseño de un software a la medida como la solución más adecuada.

La investigación de las herramientas comerciales dejó en claro que el costo del licenciamiento de las herramientas de cualquiera de los proveedores líderes en el mercado de la inteligencia de negocios era una opción prácticamente imposible de adoptar sin contemplar los costos por implementación, pólizas de mantenimiento, capacitación, entre otros.

Al estudiar las herramientas de libre distribución nos encontramos con soluciones muy robustas en donde la complejidad del sistema rebasa las necesidades del caso de estudio y finalmente la capacitación, administración y mantenimiento de estos sistemas generaría costos que aun cuando la licencia es prácticamente gratuita excederían el presupuesto del proyecto.

Desarrollar un sistema de esta naturaleza específicamente para el caso de estudio lo convierte en un proyecto totalmente viable gracias a la formación que el plan de estudios de la Facultad de Ingeniería ofrece a sus estudiantes pues a pesar de que el enfoque de la tesis está dirigido a la inteligencia de negocios y que este no es un tema formal de la Ingeniería en Computación, todos los componentes que integran el sistema finalmente son objeto de estudio de nuestra carrera.

7. Marco de referencia

El marco de estudio que fundamenta el desarrollo del proyecto se lista a continuación, es importante resaltar que las materias que lo conforman son las materias más representativas y que tienen una relación estrecha con el proyecto.

- Computación para ingenieros
- Programación avanzada y métodos numéricos
- Probabilidad y estadística
- Algoritmos y estructura de datos
- Ingeniería de software
- Bases de datos
- Administración de proyectos de software
- Redes de datos
- Administración de redes
- Bases de datos avanzadas
- Depósitos de datos
- Temas selectos de bases de datos
- Minería de datos

Una vez identificado el marco de referencia que el plan de estudios de la carrera de ingeniería en computación aporta al proyecto es de vital importancia para el desarrollo de la tesis profundizar la investigación sobre los componentes de un sistema de inteligencia de negocios, esta vez desde el punto de vista de la ingeniería para estructurar un desarrollo adecuado de la solución.

7.1 Componentes básicos para el desarrollo de inteligencia de negocios

La información es probablemente el elemento más importante para cualquier organización ya que refleja la trayectoria de la misma, es simple, sin datos no podría existir. A partir del hecho anterior surge la necesidad primordial de las

organizaciones en almacenar sus datos de una forma segura que garantice la integridad y consistencia de la información, mediante la menor redundancia posible a través de un acceso eficaz de acuerdo a las necesidades del negocio.

Para Mourya (20013) el desarrollo de la inteligencia de negocios surge a partir de los siguientes tres elementos:

- Datos
- Información
- Conocimiento

Su fundamento se basa en afirmar que los datos son la base de un todo, es decir el común denominador para la construcción de un sistema de información. Derivado de los datos está la información que finalmente converge en la generación de conocimiento. El autor define a los datos como: “cualquier hecho, número o texto que puede ser procesado por una computadora” y concluye que en la actualidad las organizaciones acumulan grandes cantidades de datos en diferentes formatos y bases de datos que incluyen (Mourya, 2013):

- Datos transaccionales u operacionales
- Datos no operacionales
- Metadatos

Para Mourya (2013) los patrones, asociaciones o relaciones entre los datos proveen información útil, dicha información puede ser transformada en conocimiento sobre patrones históricos y tendencias futuras que delimiten el desarrollo de las organizaciones.

Finalmente para el autor el flujo de desarrollo de inteligencia de negocios (datos, información y conocimiento) se lleva a cabo en la actualidad gracias a la

intervención de diversos componentes tecnológicos que ejecutan la extracción de conocimiento de los datos.

Basado en el diagrama del Vercellis (2009), los componentes mínimos necesarios para un sistema de inteligencia de negocios son:

- Sistema operacional
- Depósito de datos
- Método de extracción de conocimiento

A continuación se definen las características más importantes de cada uno de los elementos antes mencionados.

7.1.1 Sistemas operacionales

Los sistemas operacionales procesan y almacenan actividades diarias de una organización y tienen la capacidad de ejecutar reportes que representan la situación actual del negocio. Estos sistemas almacenan información a través de bases de datos regularmente de tipo relacional siendo estas el núcleo del funcionamiento de dichos sistemas.

Con base en el conocimiento vertido en el párrafo anterior es importante profundizar la investigación técnica sobre las bases de datos, conocer el rol que desempeñan dentro de un sistema operacional y sus principales métodos de operación.

El concepto de base de datos ha evolucionado a lo largo del tiempo, de una forma muy general puede ser definida como una colección de datos relacionados, sin embargo los datos por si solos no son capaces de transmitir información útil debido

a que la información de interés debe de obtenerse a través del procesamiento de los datos.

Otra definición del concepto de base de datos es: “Una colección de datos lógicamente coherente con algún tipo de significado inherente” (Elmasri, 2007). Es importante considerar la definición anterior ya que nos deja claro que no es correcto denominar base de datos a un surtido aleatorio de datos.

Para S.Sumathi coautora del libro “*Fundamentals of Relational Database Management Systems*” un surtido de datos es una representación de hechos, conceptos o instrucciones de manera formalizada, adecuada para la comunicación, interpretación o procesamiento por los seres humanos o medios automáticos.

Algunas de las características que definen a una base de datos, a diferencia de un conjunto aleatorio de datos, es el origen de los mismos, la interacción con eventos del mundo real y un segmento de individuos que están activamente interesados en su contenido (Elmasri, 2007).

Una base de datos puede ser de cualquier tamaño y complejidad, por lo que la implementación de un sistema de administración de base de datos (DBMS, *Data Base Management System*) facilita la creación y mantenimiento de la misma.

El DBMS es un sistema de software de propósito general que facilita los procesos de definición, construcción, manipulación y compartición de bases de datos entre varios usuarios y aplicaciones, Ramez Elmasri (2007) describe la funcionalidad de cada uno de estos procesos de la siguiente manera:

- **Definición:** especificar los tipos de datos, estructuras y restricciones de los datos que se almacenarán en la base de datos, así como la información descriptiva o metadatos.

- **Construcción:** proceso consistente en almacenar información en un medio de almacenamiento controlado por el DBMS.
- **Manipulación:** funciones de consulta y actualización de datos para reflejar los cambios y generar informes a partir de los datos.
- **Compartir:** permite a múltiples usuarios y programas acceder a la base de datos de forma simultánea.

Al tratarse de un sistema operacional que registra las actividades diarias de la organización el tipo de lenguaje más utilizado dentro del DBMS es el lenguaje de manipulación de los datos el cual, es ejecutado a través de diversas transacciones que garantizan la integridad de la información.

Según Ramez Elmasri (2007) una transacción es una unidad atómica de trabajo que se completa en su totalidad o no se lleva a cabo en lo absoluto. Los sistemas operacionales necesitan hacer un seguimiento de cuando se inicia, termina, confirma o aborta una transacción, de tal forma el administrador de base de datos hace un seguimiento de las siguientes operaciones:

- Inicio de la transacción (*BEGINING_TRANSACCION*)
- Manipulación de los datos (*READ OR WRITE*)
- Fin de la transacción (*END_TRANSACCION*)
- Confirmación de la transacción (*COMMIT_TRANSACCION*)
- Aborto de la transacción (*ROLLBACK*)

El autor propone el siguiente diagrama para ilustrar los estados de ejecución de una transacción.

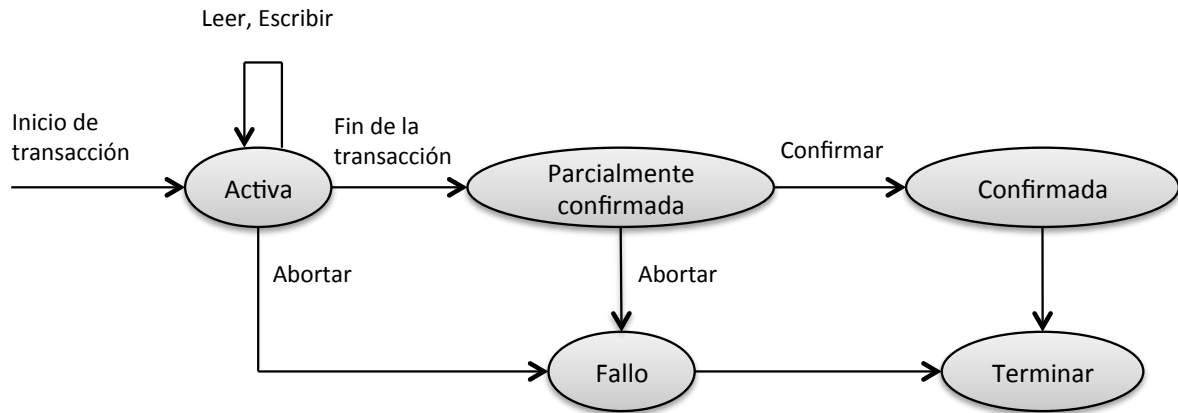


Ilustración 6 Diagrama de estado de una transacción. Fuente: adaptado de (Elmasri, 2007).

Una transacción alcanza su **punto de confirmación** cuando todas las operaciones que acceden a la base de datos se han ejecutado satisfactoriamente y el efecto de todas ellas se ha grabado en el registro. Gracias a este seguimiento de estados una base de datos puede garantizar de manera muy general la integridad de la información.

Las transacciones deben de poseer determinadas propiedades comúnmente denominadas **ACID** por sus siglas en inglés que deben de ser implementadas por el control de concurrencia y los métodos de recuperación del manejador de base de datos (Elmasri, 2007).

Según el autor las propiedades ACID son las siguientes:

- **Atomicidad:** una transacción es una unidad atómica de procesamiento, se ejecuta en su totalidad o no se ejecuta en lo absoluto.

- **Conservación de la consistencia:** una transacción está conservando la consistencia si su ejecución completa lleva a la base de datos de un estado consistente a otro.
- **Aislamiento:** una transacción debe aparecer como si estuviera ejecutándose de forma aislada a las demás. Es decir la ejecución de una transacción no debe de interferir en la ejecución de ninguna transacción simultánea.
- **Durabilidad:** los cambios aplicados a la base de datos por una transacción confirmada deben persistir en la base de datos. Estos cambios no deberán perderse por culpa de un fallo.

Un estado consistente de la base de datos satisface las restricciones específicas en el esquema, así como cualquier otra restricción que deba cumplirse dentro de la base de datos.

La abstracción de los datos y el soporte de varias vistas de los datos son algunos de los beneficios que conlleva utilizar un sistema DBMS en donde la **abstracción de los datos** se refiere a la capacidad de trabajar dichos datos sin necesidad de conocer más allá de las características fundamentales para un conocimiento mejorado de los mismos que permita a los diferentes usuarios percibirlos con un nivel de detalle adecuado.

El **soporte de varias vistas de los datos** es una característica que permite satisfacer las necesidades de cada uno de los múltiples usuarios que acceden a la base de datos ya que estos pueden necesitar una vista diferente de la base de datos; una vista es un subconjunto de la base de datos o una representación virtual de la misma.

La **compartición de datos y procesamiento de transacciones multiusuario** permite que varios usuarios puedan acceder a la base de datos, de tal forma el DBMS debe incluir un software de **control de concurrencia** para garantizar que la actualización de datos sea de manera controlada, garantizando que las transacciones operen de forma correcta y eficaz.

La siguiente ilustración propone un entorno de un sistema de base de datos simplificado que nos permite entender el funcionamiento de una transacción en donde una aplicación o un usuario accede a la base de datos enviando consultas o solicitudes de datos al DBMS lo que provoca la lectura o escritura de datos dentro de la base de datos.

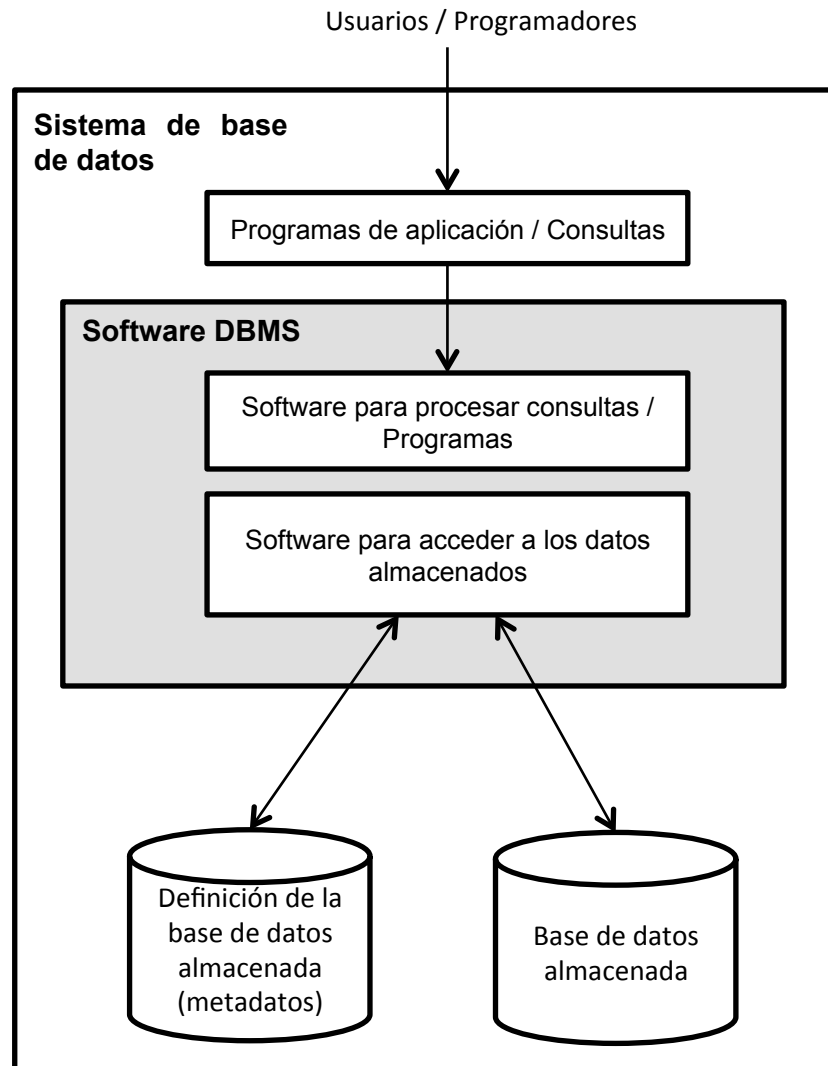


Ilustración 7 Entorno de un sistema de base de datos simplificado. Fuente: adaptado de (Elmasri, 2007).

Otras funciones importantes que integran los DBMS según Ramez Elmasri son la protección de la base de datos, protección contra eventualidades de funcionamiento defectuoso de hardware o software, accesos no autorizados y el mantenimiento de la misma durante un largo período de tiempo.

Para poder alcanzar los diferentes niveles de abstracción de la información es necesario implementar un **modelo de datos** que proporcione todos los medios necesarios para alcanzar dicha abstracción, así como un conjunto de operaciones

básicas para especificar las recuperaciones y actualizaciones de las bases de datos.

Es posible determinar una primera clasificación de los modelos de datos conforme los tipos de conceptos que utilizan para describir la estructura de la base de datos: tipos de datos, relaciones, restricciones (Elmasri, 2007).

- **Modelos de datos de alto nivel o conceptuales:** ofrecen conceptos muy cercanos a como los usuarios perciben los datos.
- **Modelos de datos de bajo nivel o físicos:** ofrecen conceptos que describen los detalles de cómo se almacenan los datos.

Entre estos dos extremos, el autor identifica una clase de **modelos representativos** o de implementación que ofrecen conceptos que los usuarios finales pueden entender y que no están muy alejados de la forma en que son almacenados, cabe mencionar que el término no es estándar y hace referencia a los modelos de datos disponibles en los manejadores de datos comerciales.

Los modelos de datos representativos o de implementación son los más utilizados en los sistemas de administración de base de datos (*Data Base Management System*, DMBS) comerciales tradicionales, entre los más comunes se encuentran los modelos de datos relacionales, modelos de datos heredados (los modelos jerárquicos y de red).

Los sistemas operacionales tienen un procesamiento denominado procesamiento de transacciones en línea (OLTP, *On-line Transaction Processing*), para Mourya (2013) la mayoría de las tareas de un sistema operacional están asociadas a la ejecución de diversas transacciones y múltiples consultas mediante un procesamiento en línea, estos sistemas son comúnmente llamados sistemas de procesamiento de transacciones en línea.

Este procesamiento está enfocando a hacer más eficientes los procesos de negocio mediante su automatización; una de sus prioridades es mantener una alta disponibilidad debido a que la información con la que trabaja es de alta criticidad y la modificación de los datos es continua, por ejemplo un sistema de punto de venta.

El nivel de detalle de la información es alto y generalmente se encuentra en una estructura de datos relacional; las operaciones de este tipo de sistemas son todos los elementos del lenguaje de manipulación de los datos (insertar, borrar, actualizar, leer).

Finalmente es importante mencionar que la información con la que trabaja un sistema operacional tiene un ciclo de vida el cual es determinado de acuerdo con las necesidades de cada organización sin embargo, no debe de perderse el objetivo de dichos sistemas en procesar y almacenar información actualizada del negocio.

De tal forma cuando la información cumple su ciclo de vida dentro del sistema operacional es momento de utilizar un depósito de datos el cual se encargará de concentrar la información histórica generada por los sistemas operacionales.

7.1.2 Depósito de datos

El siguiente componente de la infraestructura mínima necesaria para un sistema de inteligencia de negocios es conocido como depósito de datos, según W. H. Inmon quien es considerado como el padre del *data warehouse* (depósito de datos) define este concepto como: “Un conjunto de datos integrados orientados a una materia, que varían con el tiempo y que no son transitorios, los cuales soportan el proceso de toma de decisiones de una administración”.

Harjinder (1996) enuncia la gran divergencia que existe para establecer una definición precisa de un almacén de datos, sin embargo deja claro que existe un consenso entre los especialistas del tema en establecer que la tecnología de un depósito de datos es un componente esencial en el conjunto de soluciones para el soporte de decisiones en una empresa.

En términos generales, el concepto de almacén de datos surge de la necesidad de crear un acceso simple a grandes depósitos de datos estructurados con el objetivo de fundamentar la toma de decisiones. Para conocer la diferencia entre un depósito de datos y las bases de datos operacionales que soportan las aplicaciones de procesamiento de transacciones en línea (OLTP) es importante citar a Harjinder (1996) quien define las siguientes características que debe de tener un almacén de datos:

- **Está orientado a una materia:** organiza y orienta los datos desde la perspectiva del último usuario quien en el caso de un *data warehouse* es el personal de toma de decisiones.
- **Administra grandes cantidades de información:** la mayoría de los almacenes de datos contienen información histórica que se retira con frecuencia de los sistemas operacionales porque ya no es necesaria para las aplicaciones del ambiente de producción.

- **Guarda información en diversos medios de almacenamiento:** en consecuencia de los volúmenes de información que deben manejarse, un *data warehouse* frecuentemente se guarda información en diferentes medios de almacenamiento.
- **Comprende múltiples versiones de un esquema de base de datos:** debido a que un *data warehouse* tiene que controlar y administrar información originada en diferentes organizaciones de bases de datos es indispensable soportar los diferentes esquemas de base de datos en los que se encuentra almacenada la información histórica de la organización.
- **Condensa y agrega información:** debido a que la información transaccional tiene un alto nivel de detalle un depósito de datos condensa y agrega la información para presentarla en forma comprensible al personal de toma de decisiones.
- **Integra y asocia información de muchas fuentes de información:** a lo largo del tiempo las organizaciones han administrado históricamente sus operaciones en numerosas aplicaciones de software y múltiples bases de datos por lo que un *data warehouse* recopila y organiza en un solo lugar la información que estas aplicaciones han acumulado al paso de los años.

Un depósito de datos permite realizar un complejo análisis de la información histórica de una organización respaldando la toma de decisiones estratégicas de negocio, a este tipo de procesamiento se le denomina procesamiento analítico en línea (OLAP, *Online Analytical Processing*). Robert Wrembel (2007) encuentra la principal característica del procesamiento OLAP de un *data warehouse* en el modelo de datos multidimensional que intuitivamente representa la información bajo la metáfora de un cubo de datos.

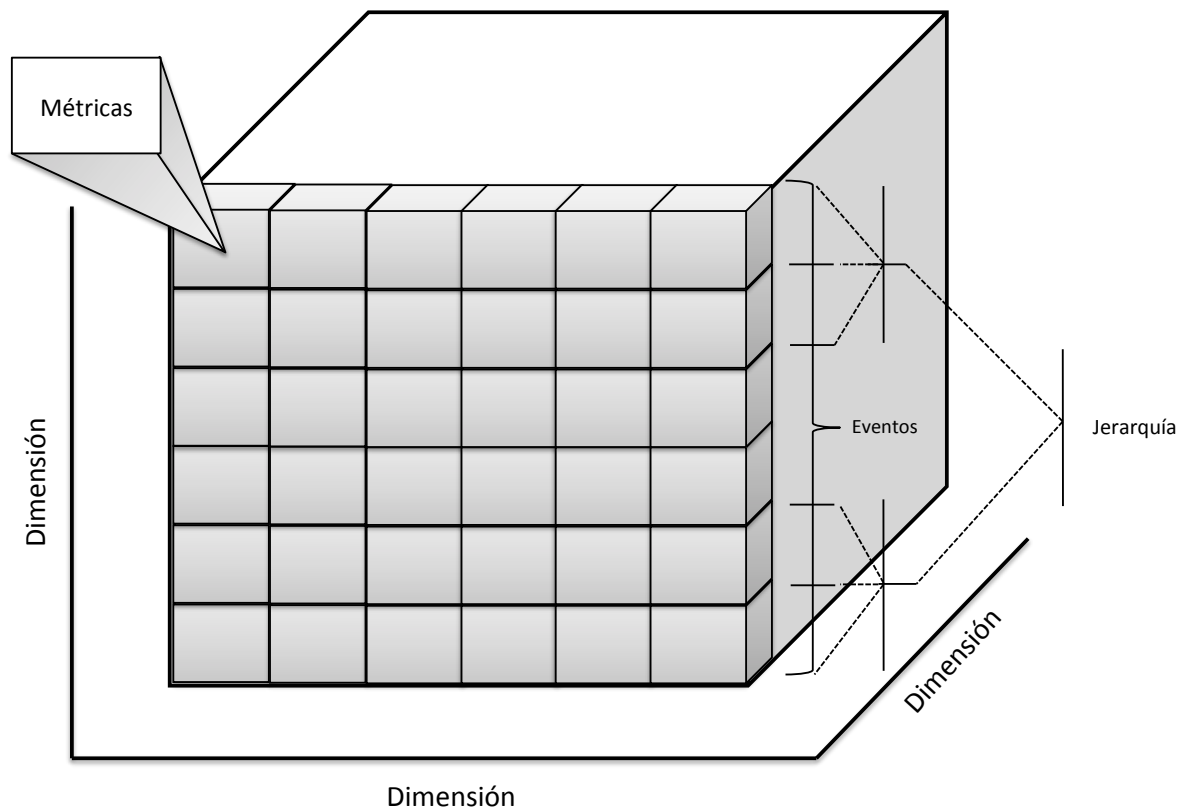


Ilustración 8 Representación metafórica de un cubo de datos. Fuente: adaptado de (Wrembel, 2007).

Cada cubo de datos está conformado por múltiples celdas que representan diferentes eventos acontecidos dentro del dominio del negocio. Cada evento es cuantificado por un conjunto de métricas; cada arista del cubo corresponde a una dimensión relevante para su análisis generalmente asociada a una jerarquía de atributos que la describe.

El modelo multidimensional trae consigo un doble beneficio, por un lado es una forma muy aproximada a la manera de pensar de los analistas de la información lo cual facilita la comprensión y por otra parte este modelo mejora significativamente el desempeño de los sistemas ya que es una estructura simple que permite a los diseñadores predecir las intenciones del usuario (Wrembel, 2007).

En un modelo de datos multidimensional los datos son organizados en múltiples dimensiones y cada una de ellas contienen diversos niveles de abstracción definidos por el concepto de jerarquías, a continuación se describen los principales tipos de esquemas de datos multidimensionales que Mourya (2013) define en su obra.

7.1.2.1 Esquema estrella

Es un esquema de base de datos relacional para representar datos multidimensionales, consiste en una o más tablas de hechos referenciadas a diferentes tablas de dimensiones, algunas de las ventajas de este esquema son proveer un rendimiento optimizado para las solicitudes típicas de un *data warehouse* y reducir el número de uniones entre las tablas de la base de datos, esta estructura genera un impacto positivo al rendimiento del sistema sin embargo, la desventaja de este esquema es que requiere más cantidad de espacio de almacenamiento.

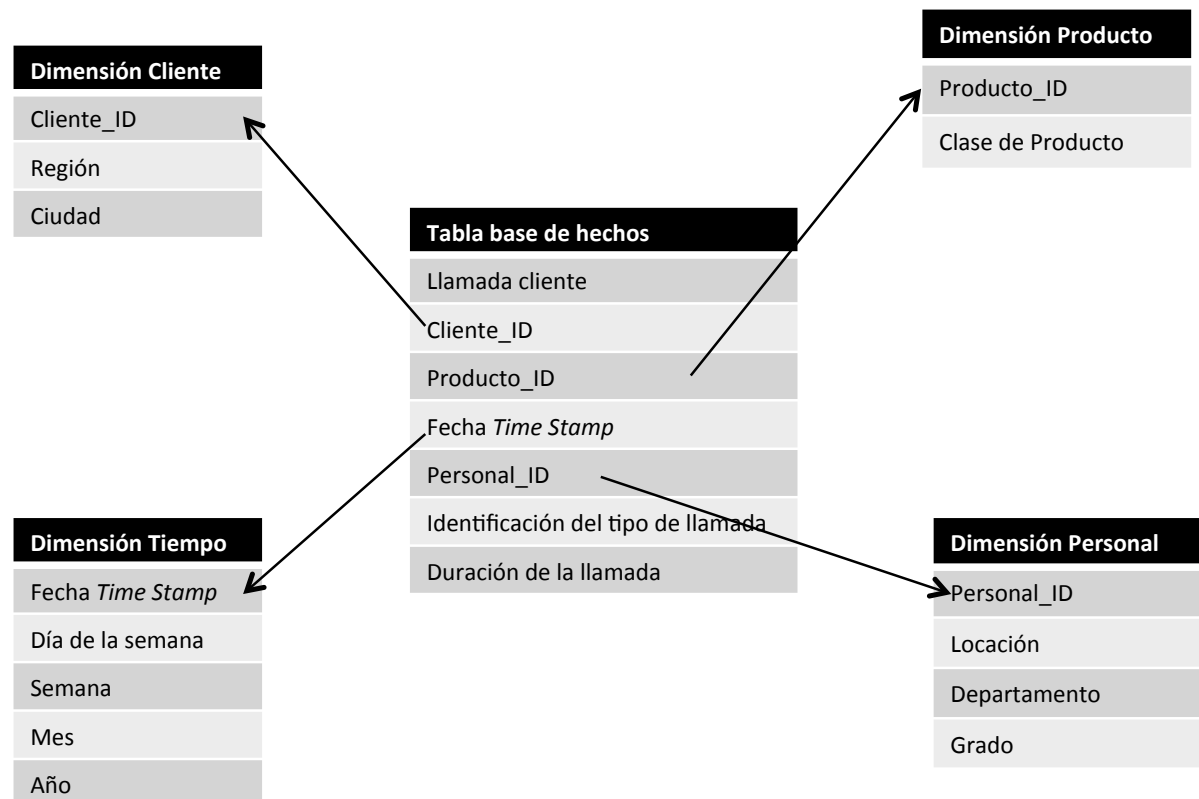


Ilustración 9 Ejemplo de un esquema estrella que contiene una tabla de hechos y diversas tablas de dimensiones. Fuente: adaptado de (Mourya, 2013).

7.1.2.2 Esquema copo de nieve

Este esquema es una extensión del esquema estrella, en donde cada punto de la estrella estalla en más puntos. En el esquema estrella cada dimensión es representada por una sola tabla dimensional, mientras que en copo de nieve esa tabla dimensional es normalizada en múltiples tablas de búsqueda.

Las ventajas de este esquema son una mejora en el rendimiento de las solicitudes debido a un requerimiento menor de espacio de almacenamiento, uniones más simples entre las tablas de búsqueda y no redundancia de los datos, lo que simplifica su mantenimiento, sin embargo al existir múltiples tablas de búsqueda el número de uniones se incrementan generando un impacto negativo en el desempeño del sistema.

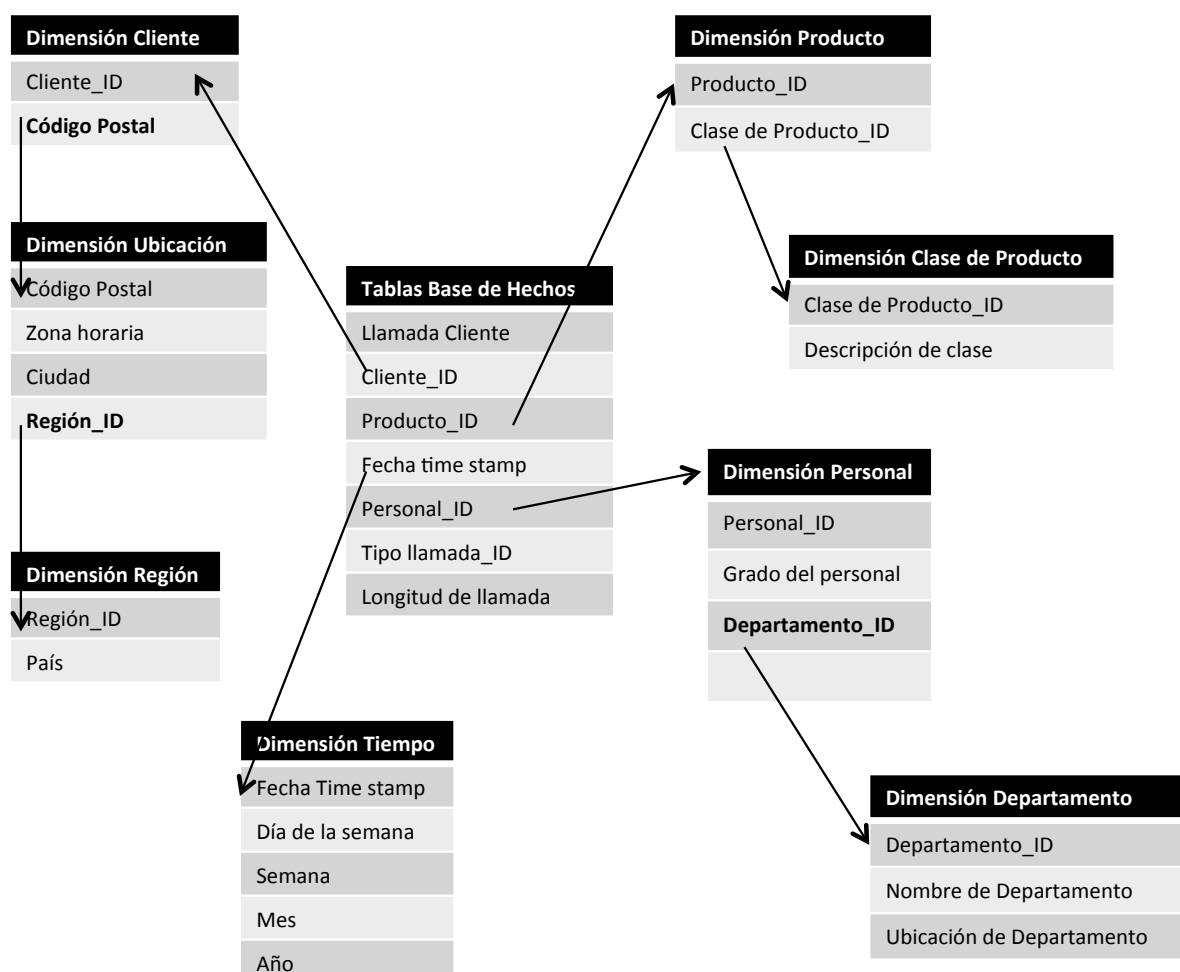


Ilustración 10 Esquema copo de nieve normalizado en múltiples tablas de búsqueda. Fuente: adaptado de (Mourya, 2013).

La interacción entre un sistema operacional y un depósito de datos está dada a partir de las herramientas de extracción, transformación y carga ETL por sus siglas

en inglés, estas herramientas son diversos tipos de software dedicados a realizar de forma automática tres funciones (Ponniah, 2010):

- **Extracción:** Durante esta primera fase la información es extraída de fuentes internas o externas de datos, principalmente se pueden diferenciar dos tipos de extracciones, la **extracción inicial** donde toda la información disponible de todos los períodos pasados alimenta un almacén de datos vacío y la **extracción incremental** que actualiza el almacén de datos utilizando información nueva que es generada a través del tiempo. La selección de los datos para ser importados es de suma importancia y está basada en el diseño del almacén de datos que a su vez depende de la información que necesitan los análisis de inteligencia de negocios y sistemas de apoyo de decisiones dentro un dominio de aplicación específico.
- **Transformación:** El objetivo de esta fase es la limpieza y transformación de los datos extraídos de las diversas fuentes de información a fin de mejorar la calidad de los mismos a través de la corrección de inconsistencias, inexactitudes y valores perdidos. Algunas de las principales deficiencias que se eliminan durante la etapa de limpieza de datos son:
 - Inconsistencias entre los valores registrados en los diferentes atributos que tienen el mismo significado.
 - Duplicación de datos.
 - Datos faltantes.
 - Existencia de valores inadmisibles.

Durante la fase de limpieza se aplican reglas automáticas predefinidas para corregir los errores más recurrentes; por otra parte, durante el proceso de transformación las conversiones de datos adicionales se producen con el fin de garantizar la homogeneidad y la integración con respecto a las fuentes de datos.

- **Carga:** Finalmente, después de ser extraída y transformada la información es cargada dentro de tablas del almacén de datos para ponerla a disposición de los analistas y las aplicaciones de apoyo de toma de decisiones.

Cuando la información atraviesa por etapas de extracción, transformación y carga tenemos que tener presente que también se está modificando el modelo de datos que alberga dicha información, de esta manera es claro el vínculo entre los sistemas transaccionales y los sistemas de análisis en línea.

Para poder extraer la información de los sistemas transaccionales y almacenarla en una estructura de almacenamiento dentro del *data warehouse* es necesario modelar dicha estructura a través de un modelo multidimensional MDM por sus siglas en inglés. Erik Thomsen (2002) afirma que la piedra angular de todos los negocios es el procesamiento de la información pues la importancia de tener información útil y en el momento adecuado es la diferencia entre tomar buenas o malas decisiones.

De esta manera es como el procesamiento analítico en línea es de suma importancia para las organizaciones y forma parte del ciclo de la inteligencia de negocios. Mourya (2013) mediante la siguiente ilustración representa la arquitectura de un almacén de datos, muestra la interacción de los dos principales tipos de procesamiento (OLTP y OLAP) en donde se puede visualizar la interacción que existe entre componentes básicos de un sistema de inteligencia de negocios.

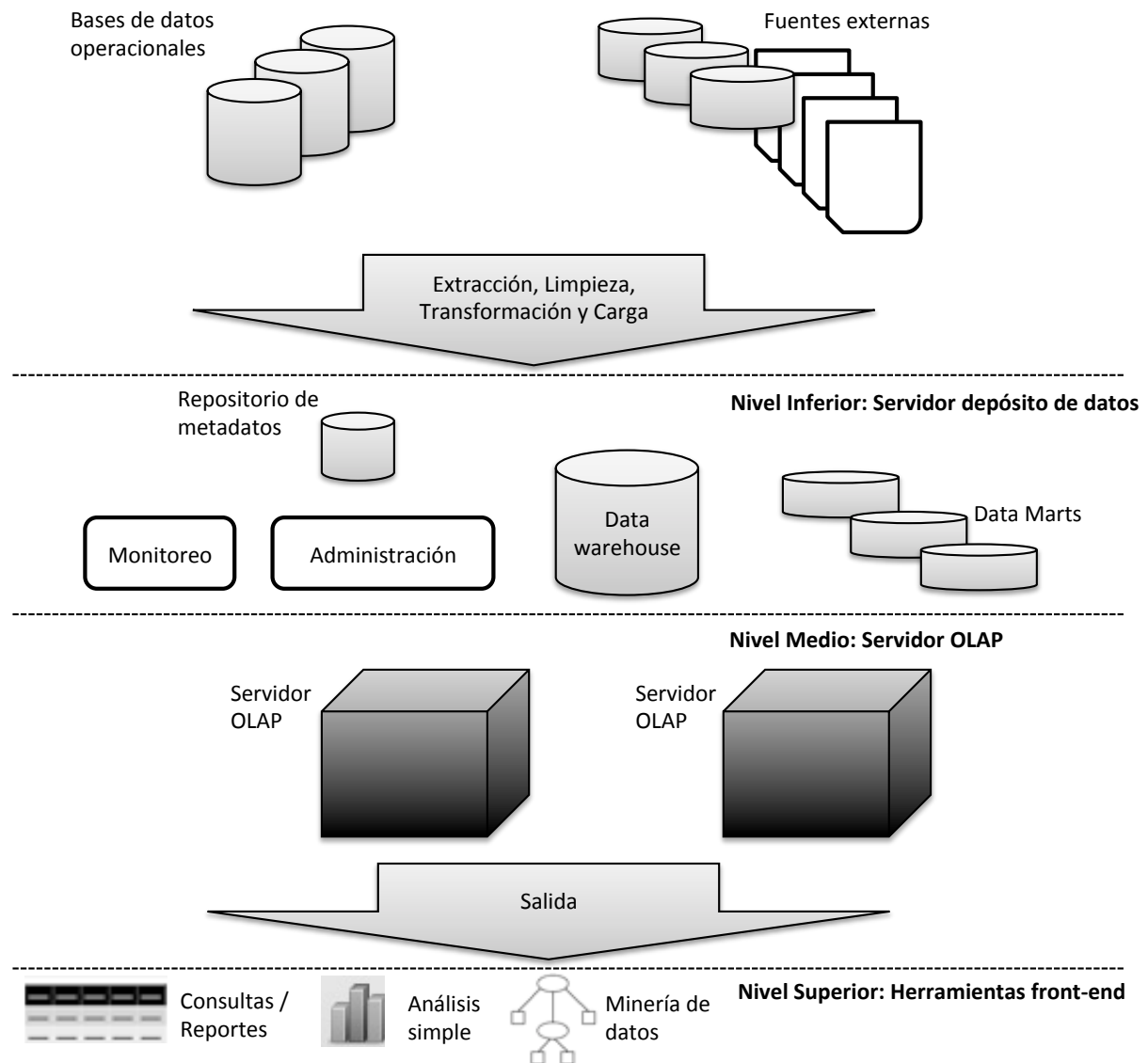


Ilustración 11 Arquitectura de tres niveles (inferior, medio y superior) de un almacén de datos. Fuente: adaptado de (Mourya, 2013).

La ilustración anterior representa claramente como los datos en una etapa inicial son procesados por bases de datos operacionales (OLTP) y algunas fuentes externas que en un principio son útiles para la operación del negocio. Con el paso del tiempo el crecimiento de estos datos es significativo y se convierten en

información histórica que necesita ser almacenada en una sola fuente para explotarla de forma apropiada.

En ese momento las herramientas de extracción, transformación y carga (ETL) transportan los datos al *data warehouse* donde son procesados mediante el procesamiento analítico en línea quien se encargan de presentar al personal de toma de decisiones reportes claros que aporten valor para estructurar las estrategias de negocio de cada organización.

El objetivo de este tipo de procesamiento es la generación de conocimiento que se traduce en ventajas competitivas, sus prioridades son crear sistemas de fácil uso y de flexible acceso a la información. La información almacenada tiene un nivel de detalle bajo pues la periodicidad de dicha información es considerada como histórica, generalmente almacenada en estructuras multidimensionales que facilitan la principal operación de estos sistemas: la lectura de grandes volúmenes de datos (Mourya, 2013).

Es importante mencionar que existen diferentes derivados del modelo OLAP, sin embargo el procesamiento de la información sigue siendo analítico en línea, básicamente la metodología de almacenamiento es lo que distingue uno de otro, para los fines de esta investigación técnica a continuación se describirán brevemente los principales derivados del modelo OLAP.

Procesamiento analítico en línea relacional (ROLAP, *Relational Online Analytical Processing*)

Se refiere al procesamiento analítico en línea relacional. En este caso, el sistema OLAP se construye encima de una base de datos relacional. Mourya (2013) enuncia que bajo esta metodología se utiliza un RDBMS para administrar el depósito de datos y agregaciones a menudo utilizando un esquema estrella. Una de las relaciones corresponde a una tabla de hechos la cual contiene datos crudos y

corresponden a lo que llamamos datos crudos del cubo. Otras relaciones dan información acerca de los valores de cada dimensión. El lenguaje de consulta y otras capacidades del sistema pueden ser adaptados bajo la suposición de que los datos se organizan de esta manera.

Para el autor las ventajas de esta metodología son las siguientes:

- **Puede almacenar grandes volúmenes de datos:** las limitaciones del tamaño de los datos en un sistema ROLAP corresponde directamente a las proporciones de la base de datos.
- **Puede aprovechar funciones inherentes de la base de datos relacional:** con regularidad la base de datos relacional trae consigo una serie de funcionalidades y debido a que la tecnología ROLAP se encuentra sobre la base de datos relacional es posible aprovechar todas esas funcionalidades.

Las desventajas que el autor Mourya determina son las siguientes:

- **El performance puede ser lento:** debido a que cada reporte ROLAP es esencialmente una o varias consultas SQL en una base de datos relacional el tiempo de consulta puede ser excesivamente largo a mayor cantidad de datos.
- **Limitado por las funcionalidades SQL:** debido a que esta tecnología principalmente depende de las consultas SQL no cubre todas las necesidades, por ejemplo es difícil ejecutar cálculos complejos usando SQL. Los proveedores ROLAP han tratado de reducir este problema construyendo herramientas que permiten la ejecución de complejas funciones así como la habilidad de permitir que los usuarios definan sus propias funciones.

Procesamiento analítico en línea multidimensional (MOLAP, *Multidimensional Analytical Processing*):

Este modelo es el más tradicional, se refiere al procesamiento analítico en línea multidimensional. En este caso, el sistema OLAP se implementa a través de una base de datos multidimensional especializada.

Las ventajas que Mourya (2013) enuncia son:

- **Excelente desempeño:** los cubos MOLAP son construidos especialmente para responder de forma óptima a la recuperación de grandes volúmenes de datos.
- **Pueden ejecutar cálculos complejos:** todos los cálculos son pre generados cuando el cubo es creado. Por lo tanto los cálculos complejos no son solo factibles sino sumamente rápidos.

Las desventajas que el autor enuncia son:

Limitados en la cantidad de datos que puede manejar: debido a que todos los cálculos son configurados cuando el cubo es construido, no es viable incluir una gran cantidad de datos en el cubo. Esto no quiere decir que los datos en el cubo no pueden derivarse de grandes volúmenes de datos, en efecto eso es posible pero en este caso solo un nivel resumido de información debe ser incluido en el cubo.

Requiere una inversión adicional: la infraestructura, la tecnología, el costo de licenciamiento para su implementación, los recursos humanos para la administración de este modelo son factores que las organizaciones tienen que considerar seriamente, pues la inversión capital puede ser significativa para las diversas organizaciones.

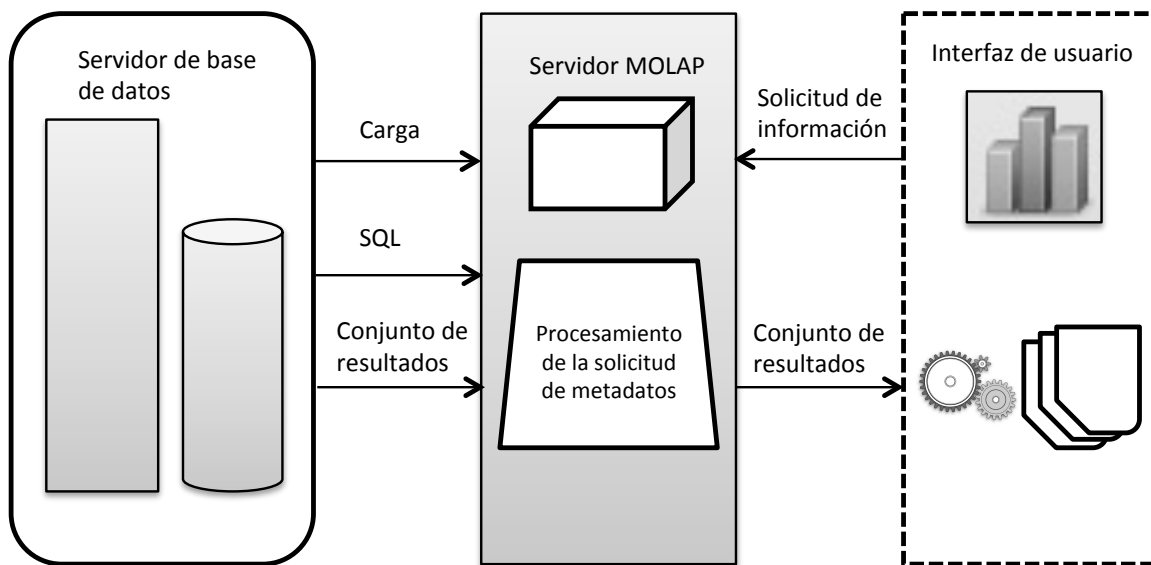


Ilustración 12 Arquitectura MOLAP Fuente: adaptado de (Mourya, 2013).

Procesamiento analítico en línea híbrido (HOLAP, *Hybrid Online Analytical Processing*)

Se refiere al procesamiento analítico en línea híbrido. Este modelo trata de combinar las fortalezas y características de ROLAP y MOLAP. Existe un desacuerdo sobre los detalles de los beneficios entre cada tipo de procesamiento, sin embargo Mourya (2013) enuncia las siguientes comparaciones siendo las más representativas para comprender las ventajas del procesamiento HOLAP.

Algunas implementaciones MOLAP son propensas a causar un fenómeno conocido como *Explosión de base de datos* lo que provoca un uso excesivo de espacio de almacenamiento cuando ciertas condiciones específicas se cumplen como: un número alto de dimensiones, resultados pre calculados y escasos datos multidimensionales.

MOLAP generalmente entrega un mejor desempeño debido a su indexación especializada y almacenamiento optimizado. Por lo general, el espacio de almacenamiento es menor en comparación al procesamiento ROLAP debido a que el almacenamiento especializado típicamente incluye técnicas de compresión.

ROLAP es generalmente más escalable. Sin embargo ante grandes volúmenes de datos es difícil implementar un sistema eficiente ya que las consultas de datos pueden tener un tremendo impacto negativo en su desempeño.

Debido a que ROLAP se apoya más en la base de datos para realizar cálculos, existen más limitaciones para ejecutar funciones especializadas.

HOLAP engloba una gama de soluciones que proporcionan una mezcla de lo mejor de ROLAP y MOLAP. Estas pueden ser generalmente un pre procesamiento rápido, escalable y ofrecer excelentes funciones de soporte.

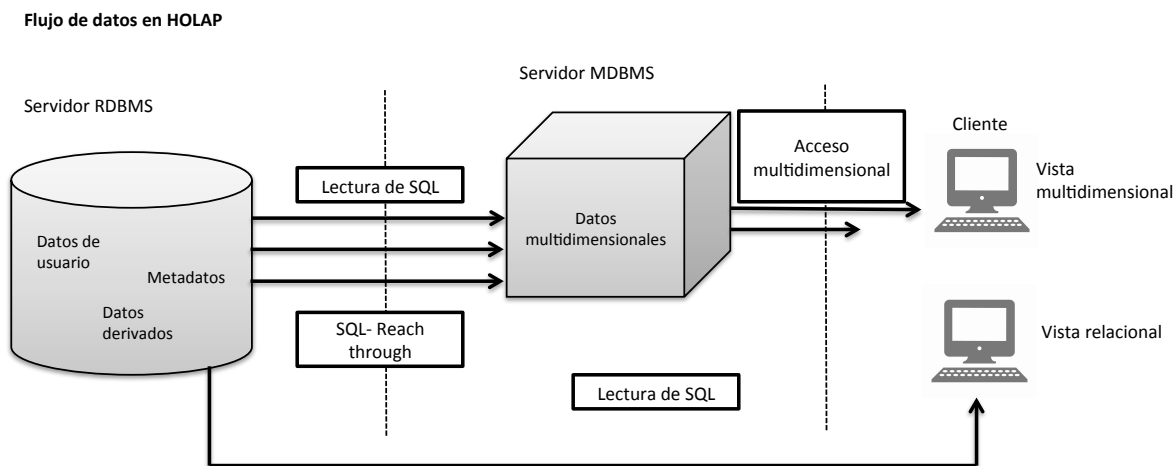


Ilustración 13 Arquitectura HOLAP. Fuente: adaptado de (Mourya, 2013).

7.1.3 Métodos de extracción de conocimiento

Hasta este punto se ha explicado la utilidad de los almacenes de datos como soporte para la toma de decisiones mediante el procesamiento de grandes volúmenes de información, sin embargo el proceso de toma de decisiones puede ser tan complejo que el uso de los almacenes de datos sea insuficiente.

De esta manera explica Ramez Elmasri (2007) la utilidad de técnicas como la minería de datos para el descubrimiento de nueva información en términos de patrones o reglas a partir de grandes cantidades de datos.

La perspectiva de este autor sobre la minería de datos como parte del proceso del descubrimiento de conocimiento se basa en un tema aún más grande que la propia minería de datos: **El Descubrimiento de conocimiento en bases de datos (KDD, Knowledge Discovery in Database).**

Se puede definir al KDD como *“el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, compresibles a partir de los datos”* (Hernández Orallo, 2004). En dicha definición, se resumen de igual manera cuáles deben ser las propiedades deseables del conocimiento extraído según el autor:

- **Válido:** hace referencia a que los patrones deben seguir siendo precisos para datos nuevos, y no sólo para aquellos que han sido usados en su obtención.
- **Novedoso:** que aporte algo desconocido tanto para el sistema y preferiblemente para el usuario.
- **Potencialmente útil:** la información debe conducir a acciones que reporten algún tipo de beneficio para el usuario.

- **Compresible:** la extracción de patrones no compresibles dificulta o imposibilita su interpretación, revisión, validación y uso en la toma de decisiones.

Entonces podemos deducir de lo anterior que, el KDD es un proceso realmente complejo que incluye no sólo la obtención de los modelos o patrones (el objetivo de la minería de datos), sino también la evaluación y posible interpretación de los mismos.

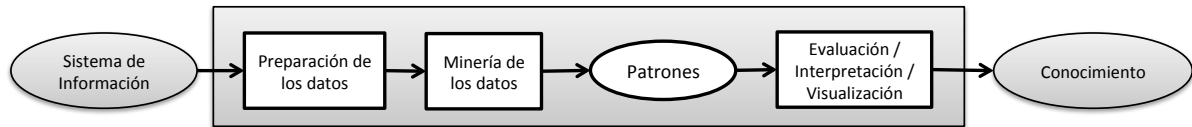


Ilustración 14 Proceso de KDD. Fuente: adaptado de (Hernández Orallo, 2004).

La minería de datos se refiere a la extracción de conocimiento a partir de grandes cantidades de datos mediante la aplicación de métodos inteligentes con la finalidad de extraer y evaluar patrones interesantes de la información.

El resultado de la minería puede derivar en la obtención de los siguientes tipos de conocimiento (Elmasri, 2007):

- **Reglas de asociación:** estas reglas relacionan la presencia de un conjunto de objetos con otro rango de valores en otro conjunto de variables. Ejemplo: Un objeto que contenga las características *a* y *b* probablemente contenga también las características.
- **Jerarquías de clasificación:** el objetivo es trabajar a partir de un conjunto existente de eventos o transacciones para crear una jerarquía de clases. Ejemplo: Es posible desarrollar un modelo para calcular los factores que determinan la idoneidad de la ubicación de una tienda dentro de una escala de 1 a 10.
- **Patrones secuenciales:** se busca una secuencia de acciones o eventos. La detección de patrones secuenciales es equivalente a la detección de asociaciones entre eventos con determinadas relaciones temporales.
- **Patrones dentro de series temporales:** es posible detectar parecidos dentro de las posiciones de una serie temporal de datos que es una

secuencia de datos tomados a intervalos regulares, como por ejemplo las ventas diarias.

- **Agrupamientos:** una población determinada de eventos o elementos puede ser dividida en conjuntos de elementos “similares”. Ejemplo: Una población entera de datos de tratamientos para una enfermedad se puede dividir en grupos basándose en el parecido de los efectos secundarios.

El descubrimiento de conocimiento dentro de una base de datos, como también es interpretada la minería de datos, combina herramientas de estadística e inteligencia artificial junto con la administración de bases de datos para analizar grandes colecciones digitales conocidas como conjunto de datos.

A lo largo del tiempo, la minería de datos se ha convertido en una herramienta sumamente importante dentro del campo de los negocios, entre los cuales destacan aseguradoras, bancos, etc. También ha tenido participación importante dentro de la astronomía y la medicina, por supuesto sin dejar de lado el campo militar para la detección de criminales y terroristas.

Según Ramez Elmasri (2007) los objetivos de la minería de datos se pueden clasificar de la siguiente manera:

- **Predicción:** la minería de datos puede mostrar cómo se comportarán en el futuro determinados atributos de los datos.
- **Identificación:** los patrones de datos se pueden utilizar para identificar la existencia de un objeto, un evento o una actividad.
- **Clasificación:** la minería de datos puede dividir los datos de forma que sea posible identificar distintas clases o categorías basándose en combinaciones de parámetros.

- **Optimización:** un posible uso de la minería de datos puede ser la optimización de recursos limitados como el tiempo, espacio, dinero, etc. Y la maximización de variables de salida.

Para Mourya, (2013) la minería de datos es el núcleo del proceso de descubrimiento de conocimiento y determina dicho proceso mediante la siguiente ilustración:

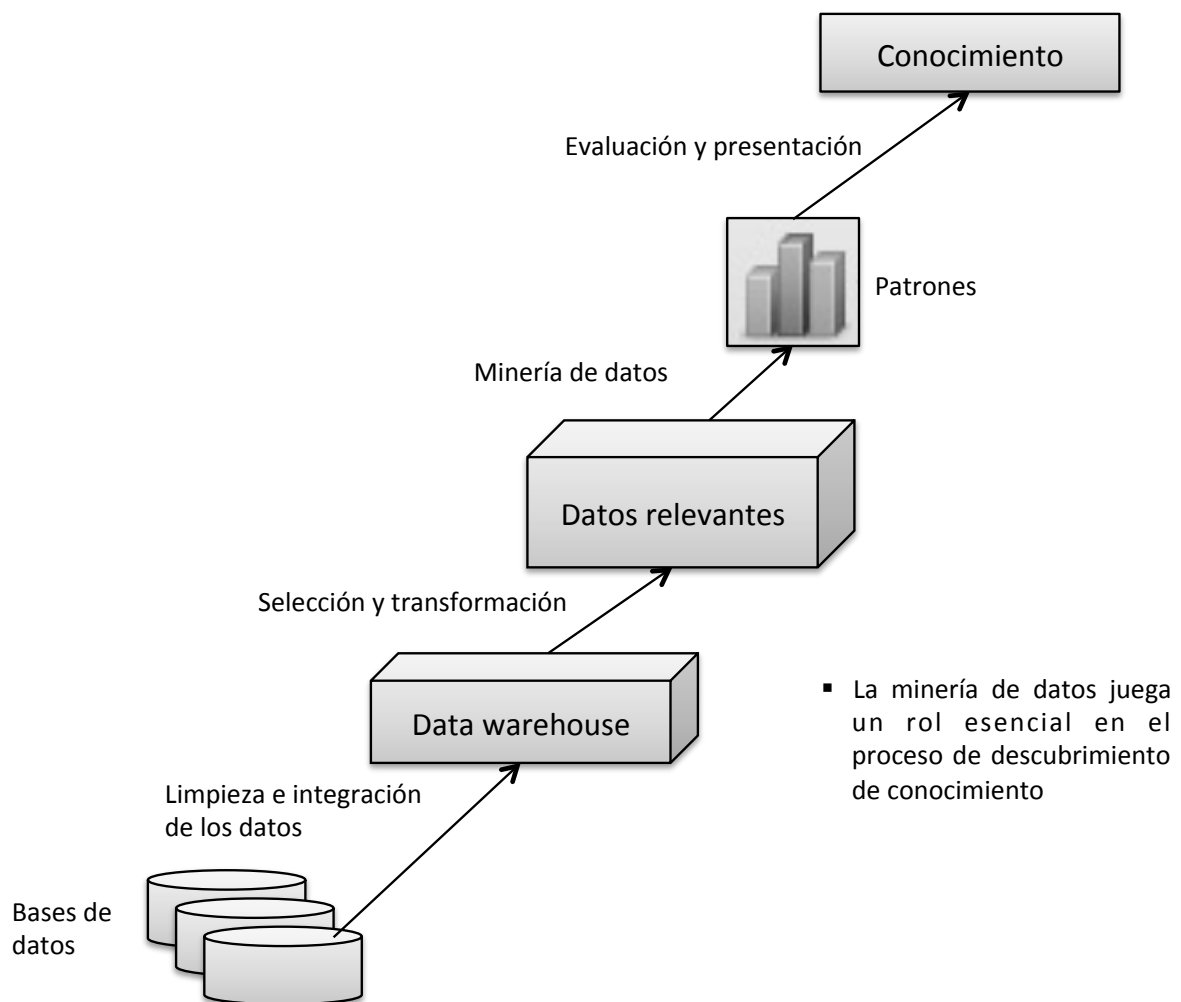


Ilustración 15 El proceso de descubrimiento de conocimiento. Fuente: adaptado de (Mourya, 2013).

El proceso comienza mediante la limpieza de los datos que es la fase en donde los datos irrelevantes son removidos del conjunto original para que posteriormente sean integrados y homogeneizados probablemente en una misma fuente.

Seguido del proceso anterior es necesario identificar los datos relevantes de análisis y sepáralos del resto de la colección para que puedan ser consolidados, es decir transformados en forma adecuada para el proceso de minería.

A través de técnicas inteligentes se extraen patrones potencialmente útiles que son evaluados y seleccionados únicamente los patrones que representen conocimiento basado en las métricas propuestas por el negocio.

Finalmente el conocimiento es visualizado en una representación gráfica para el usuario que ayuda a entender e interpretar los resultados de la minería.

Para José Hernández (2004) en su obra *Introducción a la Minería de Datos* determina que la tarea fundamental de la minería de datos es encontrar modelos inteligibles a partir de los datos y asegura que para que este proceso sea efectivo debe de ser automático o semi-automático (asistido) y el uso de los patrones descubiertos deberá ayudar a tomar decisiones más seguras y por lo tanto algún beneficio para la organización.

Existen diversos tipos de datos que pueden ser analizados bajo un proceso de minería de datos, en principio puede aplicarse la minería a cualquier tipo de información, siendo las técnicas de minería diferentes para cada una de ellas, sin embargo para el autor es importante enunciar los principales tipos de datos como: datos estructurados provenientes de bases de datos relacionales, otros tipos de datos estructurados en diversas bases de datos (espaciales, temporales, textuales y multimedia) y datos no estructurados provenientes de la web o de otros tipos de repositorios de documentos (Hernández Orallo, 2004).

La minería de datos guarda una estrecha relación con diversas disciplinas, es un campo multidisciplinario que se ha desarrollado en paralelo o como prolongación de otras tecnologías, es por eso que la investigación y los avances en la minería de datos se nutren de los que se producen en las diversas áreas relacionadas (Hernández Orallo, 2004).

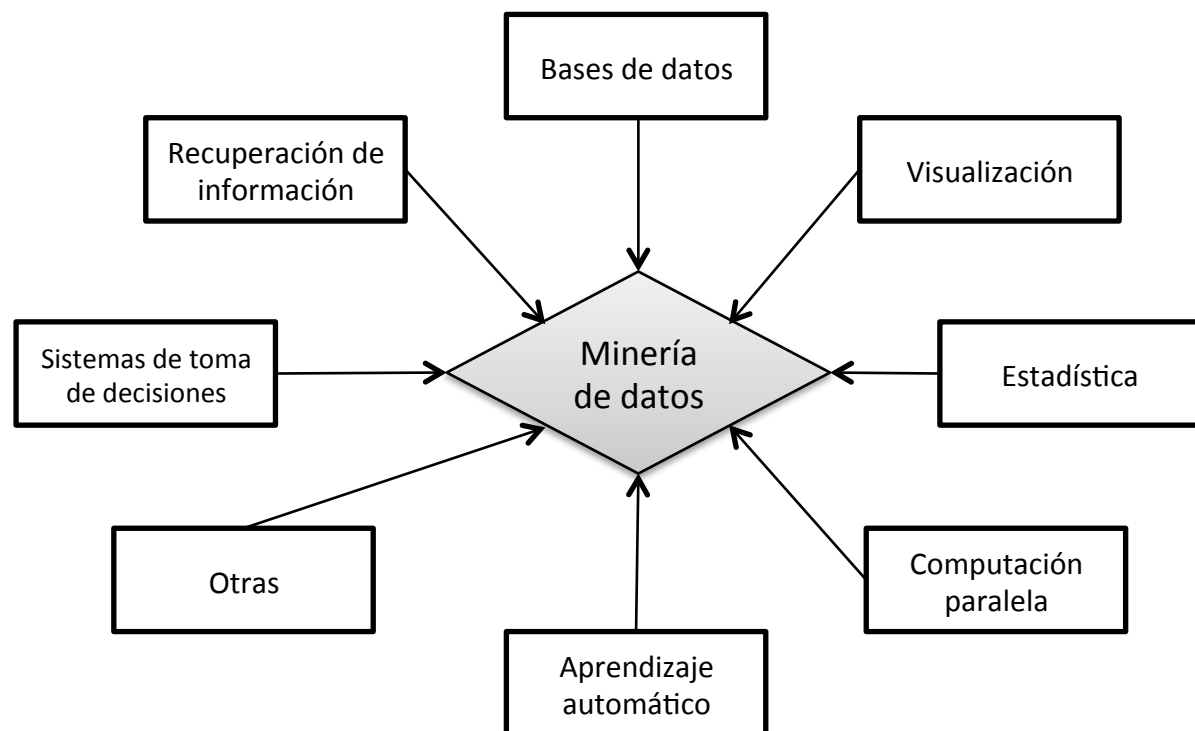


Ilustración 16 Disciplinas que contribuyen a la minería de datos. Fuente: adaptado de (Hernández Orallo, 2004).

Hernández (2004) enuncia las siguientes disciplinas como las más influyentes para la minería de datos:

- **Las bases de datos:** conceptos como los almacenes de datos y el procesamiento analítico en línea tienen una gran relación con la minería de datos desde la perspectiva de la extracción de conocimiento novedoso y compresible. Las técnicas de indexación y de acceso eficiente a los datos

son muy relevantes para el diseño de algoritmos eficientes de minería de datos.

- **La recuperación de la información (*Information Retrieval*, IR):** consiste en obtener información desde datos textuales. Una tarea típica es encontrar documentos a partir de palabras claves, lo cual puede verse como un proceso de clasificación de los documentos en función de estas palabras clave. Para ello se usan medidas de similitud entre los documentos y la consulta; muchas de estas medidas se han empleado en aplicaciones más generales de minería de datos.
- **La estadística:** esta disciplina ha proporcionado muchos de los conceptos, algoritmos, técnicas que se utilizan en la minería de datos, como por ejemplo, la media, la varianza, las distribuciones, el análisis univariante y multivariante, la regresión lineal y no lineal, etc.
- **El aprendizaje automático:** ésta es el área de la inteligencia artificial que se ocupa de desarrollar algoritmos y programas capaces de aprender y constituye, junto con la estadística, el corazón del análisis inteligente de los datos.
- **Los sistemas para la toma de decisión:** son herramientas, sistemas informatizados, que asisten a los directivos en la resolución de problemas y en la toma de decisiones. Herramientas como el análisis ROC o los mismos árboles de decisión provienen de esta área.
- **La visualización de los datos:** el uso de técnicas de visualización permite al usuario descubrir, intuir o entender patrones que serían más difíciles de “ver” a partir de descripciones matemáticas o textuales de los resultados. Existen técnicas de visualización, como por ejemplo las gráficas, las icónicas, las basadas en píxeles, las jerárquicas, entre otras.

- **La computación paralela y distribuida:** actualmente, muchos sistemas de bases de datos comerciales incluyen tecnologías de procesamiento paralelo, distribuido o de computación en *grid*. En estos sistemas el coste computacional de las tareas más complejas de minería de datos se reparte entre diferentes procesadores o computadores. Su éxito se debe en parte a la explosión de los almacenes de datos (su adaptación distribuida) y de la minería de datos, en lo que las presentaciones de los algoritmos de consulta son críticas.
- **Otras disciplinas:** dependiendo del tipo de datos a ser minados o del tipo de aplicación, la minería de datos usa también técnicas de otras disciplinas como el lenguaje natural, el análisis de imágenes, el procesamiento de señales, los gráficos por computadora, etc.

7.2 Aplicaciones de la minería de datos

Para los fines de esta investigación técnica es importante conocer las aplicaciones más significativas en cuanto a minería de datos se refiere, para Hernández (2004) la integración de las técnicas de minería de datos en las actividades del día a día se ha convertido en algo habitual.

Los negocios de la distribución y la publicidad dirigida han sido tradicionalmente las áreas en las que más se han empleado los métodos de minería, ya que han permitido reducir costos o aumentar la receptividad de ofertas; en su obra, Hernández lista algunas de las áreas más importantes para el desarrollo de la minería de datos así mismo algunas de las aplicaciones más significativas de cada rubro (Hernández Orallo, 2004).

- **Aplicaciones financieras y bancarias:**

- Obtención de patrones de uso fraudulento de tarjetas de crédito.
- Determinación del gasto en tarjeta de crédito por grupos.
- Cálculo de correlaciones entre indicadores financieros.
- Identificación de reglas de mercado de valores a partir de históricos.
- Análisis de riesgos en créditos.
- **Análisis de mercado, distribución y, en general, comercio:**
 - Análisis de la cesta de la compra (compras conjuntas, secuenciales, ventas cruzadas, señuelos, etc.).
 - Evaluación de campañas publicitarias.
 - Análisis de la fidelidad de los clientes. Reducción de fuga.
 - Segmentación de clientes.
 - Estimación de stocks, de costes, de ventas, etc.
- **Seguros y Salud privada:**
 - Determinación de los clientes que podrían ser potencialmente caros.
 - Análisis de procedimientos médicos solicitados conjuntamente.
 - Predicción de qué clientes contratan nuevas pólizas.
 - Identificación de patrones de comportamiento para clientes con riesgo.
 - Identificación de comportamiento fraudulento.
 - Predicción de los clientes que podrían ampliar su póliza para incluir procedimientos extras.
- **Educación:**
 - Selección o captación de estudiantes.
 - Detección de abandonos y de fracaso.
 - Estimación del tiempo de estancia en la institución.
- **Procesos industriales:**
 - Extracción de modelos sobre comportamiento de compuestos.
 - Detección de piezas con trabas. Modelos de calidad.
 - Predicción de fallos y accidentes.
 - Estimación de composiciones óptimas en mezclas.
 - Extracción de modelos de coste.

- Extracción de modelos de producción.
- **Medicina:**
 - Identificación de patologías. Diagnóstico de enfermedades.
 - Detección de pacientes con riesgo de sufrir una patología concreta.
 - Gestión hospitalaria y asistencial. Predicciones temporales de los centros asistenciales para el mejor uso de recursos, consultas, salas y habitaciones.
 - Recomendación priorizada de fármacos para una misma patología.
- **Biología, bioingeniería y otras ciencias:**
 - Análisis de secuencias de genes.
 - Análisis de secuencias de proteínas.
 - Predecir si un compuesto químico causa cáncer.
 - Clasificación de cuerpos celestes.
 - Predicción de recorrido y distribución de inundaciones.
 - Modelos de calidad de aguas, indicadores ecológicos.
- **Telecomunicaciones:**
 - Establecimiento de patrones de llamadas.
 - Modelos de carga en redes.
 - Detección de fraude.
- **Otras áreas**
 - Correo electrónico y agendas personales: clasificación y distribución automática de correo, detección de correo *spam*, gestión de avisos, análisis del empleo del tiempo.
 - Recursos Humanos: selección de empleados.
 - Web: análisis del comportamiento de los usuarios, detección de fraude en el comercio electrónico, análisis de los logs de un servidor web.
 - Turismo: determinar las características socioeconómicas de los turistas en un determinado destino o paquete turístico, identificar patrones de reservas, etc.

- Tráfico: modelos del tráfico a partir de fuentes diversas: cámaras, GPS, etc.
- Hacienda: detección de evasión fiscal.
- Policiales: identificación de posibles terroristas en un aeropuerto.
- Deportes: estudio de la influencia de jugadores y de cambios. Planificación de eventos.
- Política: diseño de campañas políticas, estudios de tendencias de grupos, etc.

Hernández concluye que existe una gran variedad de aplicaciones en donde el uso de la minería de datos puede ayudar a entender mejor el entorno donde se desenvuelve la organización y, en definitiva, mejorar la toma de decisiones en dicho entorno.

La efectividad de la minería de datos se basa a partir del uso de sistemas y herramientas para minar la información, Hernández, (2004) en su obra Introducción a la minería de datos clasifica los sistemas y herramientas de la minería de datos bajo los siguientes criterios:

- El modelo de datos que generan
- El tipo de datos que minan
- El tipo de Técnica
- El tipo de aplicación

Dando como resultado la siguiente clasificación:

- Tipo de base de datos minada: teniendo en cuenta los diferentes modelos de datos podemos hablar de sistemas de minería de datos relacionales, multidimensionales, orientados a objetos, etc. Asimismo, atendiendo al tipo de datos manejados, hablamos de sistemas textuales, multimedia, espaciales o web.

- Tipo de conocimiento minado: también pueden tenerse en cuenta los niveles de abstracción del conocimiento minado: conocimiento generalizado (alto nivel de abstracción), a nivel primitivo (a nivel de filas de datos), o conocimiento a múltiples niveles de abstracción. Por último, podemos igualmente distinguir entre los sistemas que buscan regularidades en los datos (patrones) frente a los que analizan las irregularidades (excepciones).
- Tipo de funcionalidad y de técnica: los sistemas de minería de datos se pueden clasificar basándose en su funcionalidad (clasificación, agrupamiento, etc.) o por los métodos de análisis de los datos empleados (técnicas estadísticas, redes neuronales, etc.).
- Tipo de aplicación: podemos distinguir dos clases de sistemas según la aplicación para la que se usan: los sistemas de propósito general y los sistemas específicos (como los usados en aplicaciones financieras, web, e-mail, etc.)

Esta perspectiva del autor es importante pues deja claro que minar datos no siempre tiene una relación directa con la infraestructura tecnológica de las grandes empresas, existen variaciones técnicas y tecnologías que permiten el uso de la minería de datos a diversos tipos y tamaños de organizaciones.

Hasta este punto se ha profundizado el conocimiento sobre los componentes básicos de un sistema de inteligencia de negocios sin embargo, al tratarse de un proyecto de desarrollo es indispensable estudiar los procedimientos de análisis y diseño de sistemas que el aporte bibliográfico propone.

7.3 Análisis y diseño de sistemas

Según Kendall (2011) durante mucho tiempo se ha reconocido la importancia de administrar recursos clave como las personas y la materia prima, el autor señala que actualmente la información ha encontrado su lugar apropiado como recurso clave y hace énfasis en la clasificación de los diversos tipos de sistemas que se deriva de las diferentes características individuales de los usuarios o empresas. El autor propone y define la siguiente clasificación de sistemas:

- Sistemas de procesamiento de transacciones (TPS)
- Sistemas de automatización de oficinas (OAS)
- Sistemas de trabajo de conocimiento (KWS)
- Sistemas de información administrativa (MIS)
- Sistemas de soporte de decisiones (DSS)
- Sistemas expertos
- Sistemas de soporte para ejecutivos (ESS)
- Sistemas de soporte de decisión en grupo (GDSS)
- Sistemas de trabajo colaborativo asistido por computadora (CSCWS)

Para los fines de esta investigación se definirán brevemente los sistemas que guardan una relación estrecha con el sistema de inteligencia de negocios a desarrollar para el caso de estudio de esta tesis.

7.3.1 Sistemas de procesamiento de transacciones

Son sistemas de información computarizados dirigidos al procesamiento de grandes cantidades de información para las transacciones de negocio rutinarias como nóminas e inventarios. Estos sistemas deben de tener una alta disponibilidad y confiabilidad en la información que generan debido a que dicha información

contiene la operación del negocio y constantemente es analizada para determinar la situación actual.

7.3.1.1 Sistemas de automatización de oficinas

El autor afirma que en el nivel de conocimiento de la organización existen dos clases de sistemas los cuales se caracterizan de la siguiente manera:

- Sistemas de automatización de oficinas: Brindan apoyo a las personas que trabajan con datos, no para crear conocimiento sino para analizar y transformar los datos o manipularlos de cierta forma antes de compartirlos o diseminarlos de manera formal a través de la organización.
- Sistemas de trabajo de conocimiento: Brindan apoyo a profesionales, ayudándoles a crear conocimiento y a integrarlo a su organización.

7.3.1.2 Sistemas de información administrativa

Los sistemas de información administrativa no sustituyen a los sistemas de procesamiento de transacciones, según el autor todos los sistemas de información administrativa incluyen el procesamiento de transacciones. Para el autor este tipo de sistemas pueden realizar un espectro mayor de tareas organizacionales en comparación a los sistemas TPS.

Estos sistemas de información pueden generar información que se utiliza en el proceso de toma de decisiones, para acceder a la información los usuarios comparten una base de datos que almacena tanto los datos como los modelos que permiten al usuario interactuar con ellos, interpretarlos y aplicarlos.

7.3.1.3 Sistemas de soporte de decisiones

Para el autor estos sistemas pertenecen a una clase superior de sistemas de información computarizados, son similares a los sistemas de información administrativa tradicional debido a que ambos dependen de una base de datos como fuente de datos, la diferencia radica en que los sistemas de soporte de decisiones están totalmente dirigidos a apoyar el proceso completo de toma de decisiones o inteligencia de negocio.

7.3.1.4 Sistemas de decisiones en grupo y sistemas de trabajo colaborativo asistido por computadora

Para las organizaciones, las decisiones en grupo son cada vez más importantes, entonces cuando una empresa toma decisiones semiestructuradas o no estructuradas, para el autor, es momento de que un sistema de decisiones en grupo apoye las actividades de la organización ya que el objetivo de estos sistemas, según el autor, es que un grupo resuelva un problema con la ayuda de varios apoyos como encuestas, cuestionarios, lluvias de ideas y creación de escenarios.

7.3.1.5 Sistemas de soporte para ejecutivos

Este tipo de sistemas en la actualidad son de suma importancia para la mesa directiva de cualquier empresa que busca tener un soporte para la toma de decisiones, ya que según el autor estos sistemas ayudan a los ejecutivos a organizar sus interacciones con el entorno externo ofreciendo tecnologías de gráficos y comunicaciones en sitios accesibles como salas de juntas u oficinas. Este tipo de sistemas se alimentan de la información generada por los sistemas TPS y MIS creando un entorno que facilite el entendimiento de los problemas de la organización de una manera informada.

7.3.2 Ciclo de vida de desarrollo de sistemas

Para garantizar un desarrollo adecuado del sistema para el caso de estudio es importante profundizar la investigación técnica sobre las metodologías más utilizadas en el desarrollo de sistemas a fin de garantizar que el proyecto alcance los objetivos propuestos en tiempo y forma, optimizando la explotación de los recursos asignados.

El ciclo de vida de desarrollo de sistemas es una metodología en fases para el análisis y diseño con la cual el desarrollo de los sistemas mejora sustancialmente al realizar un ciclo específico de actividades. Para el autor, las fases de desarrollo son las mostradas en la siguiente ilustración, sin embargo deja claro que entre los analistas de sistemas existen múltiples variantes.

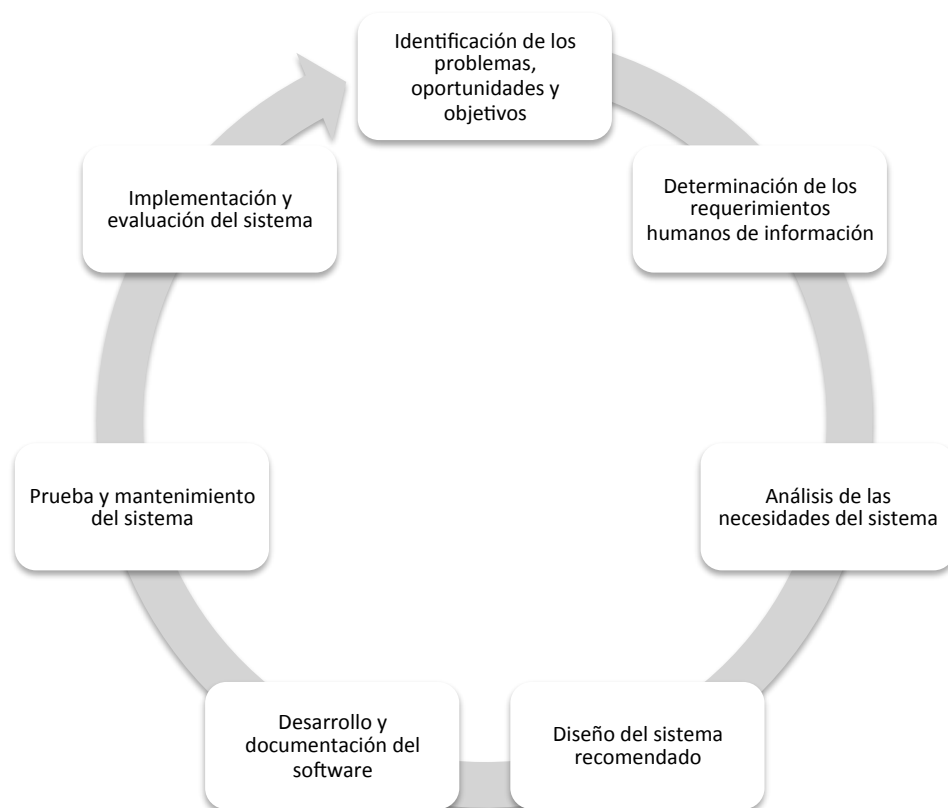


Ilustración 17 Las siete fases del ciclo de desarrollo de sistemas. Fuente: adaptado de (Kendall & Kendall, 2011).

A continuación se definen las siete fases desde la perspectiva del autor Kendall en su obra análisis y diseño de sistemas:

7.3.2.1 Identificación de los problemas, oportunidades y objetivos

Esta primera etapa del ciclo es crucial para el desarrollo del sistema pues indiscutiblemente una mala percepción del problema, una tardía identificación de oportunidades o la definición de un objetivo erróneo implicaría que se estuviera trabajando e invirtiendo recursos en un proyecto que muy probablemente no cumpla con las expectativas del solicitante e inclusive el tratar de corregir cualquiera de estos tres elementos durante el desarrollo del proyecto puede ser un factor determinante en el fracaso del mismo.

En el trabajo del analista esta fase inicia con un análisis de la situación actual de la empresa, el cual debe de reflejar con claridad y honestidad los problemas de la organización para que sean discutidos posteriormente con la mesa directiva. Es importante dejar claro que durante este análisis, el analista debe encontrar las oportunidades que permitan mejorar el sistema de información y así la empresa pueda generar ventajas competitivas o establecer un estándar en la industria.

La identificación de objetivos es también un elemento importante dentro de esta etapa inicial, el analista debe de entender en una primera instancia que es lo que está queriendo lograr la empresa u organización, para finalmente alinear los objetivos y enfrentar problemas u oportunidades específicas de la empresa de forma adecuada.

Para que esta fase tenga un alto grado de efectividad es necesario que exista una fuerte comunicación entre los usuarios, los analistas y los administradores del sistema que coordinan el proyecto.

Las actividades más relevantes para esta etapa son:

- Entrevistar a los encargados de la administración de los usuarios
- Sintetizar el conocimiento obtenido
- Estimar el alcance del proyecto
- Documentar los resultados

El resultado de esta fase es un informe de viabilidad el cual contiene la definición de un problema y sintetiza los objetivos, este reporte es analizado posteriormente por la mesa directiva quien se encarga de definir si el proyecto puede continuar.

7.3.2.2 Determinación de los requerimientos de información del factor humano

En esta fase el analista tiene la tarea de identificar las necesidades particulares de los usuarios involucrados, a través de diferentes técnicas las cuales le darán un conocimiento de la interacción que tienen los usuarios con el sistema de información para la realización de sus actividades laborales.

Las técnicas más comunes que un analista utiliza para desarrollar esta fase del proyecto son:

- Entrevistas
- Muestreos o investigación de datos duros
- Cuestionarios

- Métodos discretos como: observar el comportamiento de los encargados al tomar decisiones
- Métodos integrales como: la creación de prototipos

Algunos de los cuestionamientos que el resultado de esta fase debe de responder con claridad son los siguientes:

- ¿Cuáles son las fortalezas y debilidades físicas de los usuarios? En otras palabras,
- ¿Qué hay que hacer para que el sistema sea perceptible, legible y seguro?
- ¿Cómo puede diseñarse el nuevo sistema para que sea fácil de usar, aprender y recordar?
- ¿Cómo puede el sistema apoyar las tareas laborales individuales de un usuario y buscar nuevas formas de hacerlas más productivas?

Si existe un sistema actual el analista debe conocer los detalles de este y cuestionar por qué la organización lo utiliza, ya que pueden existir razones suficientemente buenas lo que dará al analista puntos clave para el nuevo sistema.

7.3.2.3 Análisis de las necesidades del sistema

Una vez entendidos los requerimientos de los usuarios involucrados en el sistema de información es momento de que el analista comience a identificar las necesidades propias del sistema para poder cubrir de manera efectiva las necesidades previamente identificadas.

Para esta labor el analista también cuenta con diferentes técnicas y herramientas como:

- Diagramas de flujo de datos: útiles para graficar las entradas, procesos y salidas de información.

- Diagramas de actividad o de secuencia: útiles para mostrar la secuencia de los eventos.

A partir de la realización de dichos diagramas es importante que el analista desarrolle un diccionario de datos para enlistar todos los elementos de datos utilizados en el sistema.

Durante esta fase el analista también tiene que identificar y analizar las decisiones estructuradas que se llevarán a cabo durante la ejecución del sistema, es decir determinar condiciones y acciones del sistema.

Existen tres métodos principales según el autor para el análisis de decisiones estructuradas:

- Inglés / Español estructurado
- Tablas de decisión
- Árboles de decisión

Para el resultado de esta fase el analista prepara una propuesta de sistemas en la que sintetiza todo lo que ha averiguado sobre los usuarios, la capacidad de uso y la utilidad de los sistemas actuales; incluye un análisis costo-beneficio de las alternativas y si se requiere hace algunas recomendaciones.

7.3.2.4 Diseño del sistema recomendado

Para la fase de diseño el analista utiliza toda la información antes recopilada para desarrollar el diseño lógico del sistema de información. Un buen diseño lógico constituido por diversos procedimientos que se adecue a los requerimientos particulares de los usuarios ayudará a simplificar la interacción de estos con el

sistema y así de esta manera garantizar que la información que se introduce en él sea confiable.

Una vez diseñados los procedimientos del sistema que conforman el diseño lógico del mismo, es imperativo que el analista contemple como parte de esta fase de desarrollo del sistema de información el diseño de la interfaz gráfica, la cual constituye un factor muy importante y debe realizarse con ayuda de los usuarios finales para garantizar que el sistema sea perceptible, legible y seguro.

Otro elemento imperativo a desarrollar en esta fase, es el diseño de la base de datos ya que parte del desempeño del sistema está ligado a una eficiente distribución de la información la cual debe estar organizada de una manera lógica y debe corresponder con la forma en que los usuarios ven su trabajo.

Por último el analista debe de crear procedimientos de respaldo para proteger el sistema y los datos para finalmente producir paquetes de especificaciones de programas para los programadores. Según el autor cada paquete debe de contener los diseños de las entradas y salidas, las especificaciones de los archivos y los detalles sobre el procesamiento también puede incluir árboles o tablas de decisiones, UML o diagramas de flujo de datos.

7.3.2.5 Desarrollo y documentación del software

En esta fase el analista trabaja con los programadores para desarrollar el sistema deseado, a su vez es necesario que este desarrolle la documentación que acompañara y dará soporte al software como:

- Manuales de usuario
- Manuales de procedimientos
- Ayuda en línea

- Sitios web con preguntas (FAQ)
- Archivos léame

Como los usuarios están involucrados desde el comienzo del proyecto, la fase de documentación debe lidiar con las preguntas que hicieron y resolvieron junto con el analista.

7.3.2.6 Prueba y mantenimiento del sistema

Esta fase es de suma importancia ya que es la antesala de la liberación del software, por lo que hacer pruebas antes de este punto disminuirá costos por las correcciones que deban realizarse. En este punto los programadores, el analista y en algunas ocasiones el personal de la organización hacen pruebas del funcionamiento con datos de muestra para después utilizar los datos actuales de la organización, analizan el desempeño del sistema para garantizar su efectividad y de ser necesario realizan las correcciones pertinentes.

Para finalizar esta fase el analista desarrolla un plan de mantenimiento el cual se llevará a cabo durante toda la vida del sistema de información para garantizar un funcionamiento adecuado del sistema.

7.3.2.7 Implementación y evaluación del sistema

En esta última fase, el analista es el responsable de coordinar la implementación del sistema de información a un estado de producción, en el caso de existir un sistema anterior el analista tiene la misión de asegurar una transición del sistema antiguo al sistema nuevo sin inconvenientes y con la interrupción mínima necesaria, lo que implica el convertir los archivos de los formatos anteriores a los nuevos, crear una nueva base de datos, instalar equipo, etc.

Finalmente y no menos necesario, en esta última fase también se ejecuta el plan de capacitación para que los usuarios puedan operar el sistema con los menores inconvenientes posibles.

Es importante hacer hincapié que si bien estas siete fases de desarrollo son un ciclo, no necesariamente deben de ejecutarse en serie, en algunas ocasiones es útil llevar algunas actividades en paralelo con las debidas precauciones para agilizar el desarrollo del sistema.

7.3.3 Proceso de desarrollo para un proyecto ágil

El autor Kendall (Kendall & Kendall, 2011) en su obra afirma que existen diversas metodologías de desarrollo sin embargo para fines de este trabajo es importante entender la metodología de desarrollo ágil, como una metodología alternativa al desarrollo estructurado.

Para el autor el desarrollo de un sistema a través de una metodología ágil contempla cinco etapas las cuales son representadas a través de la siguiente ilustración:

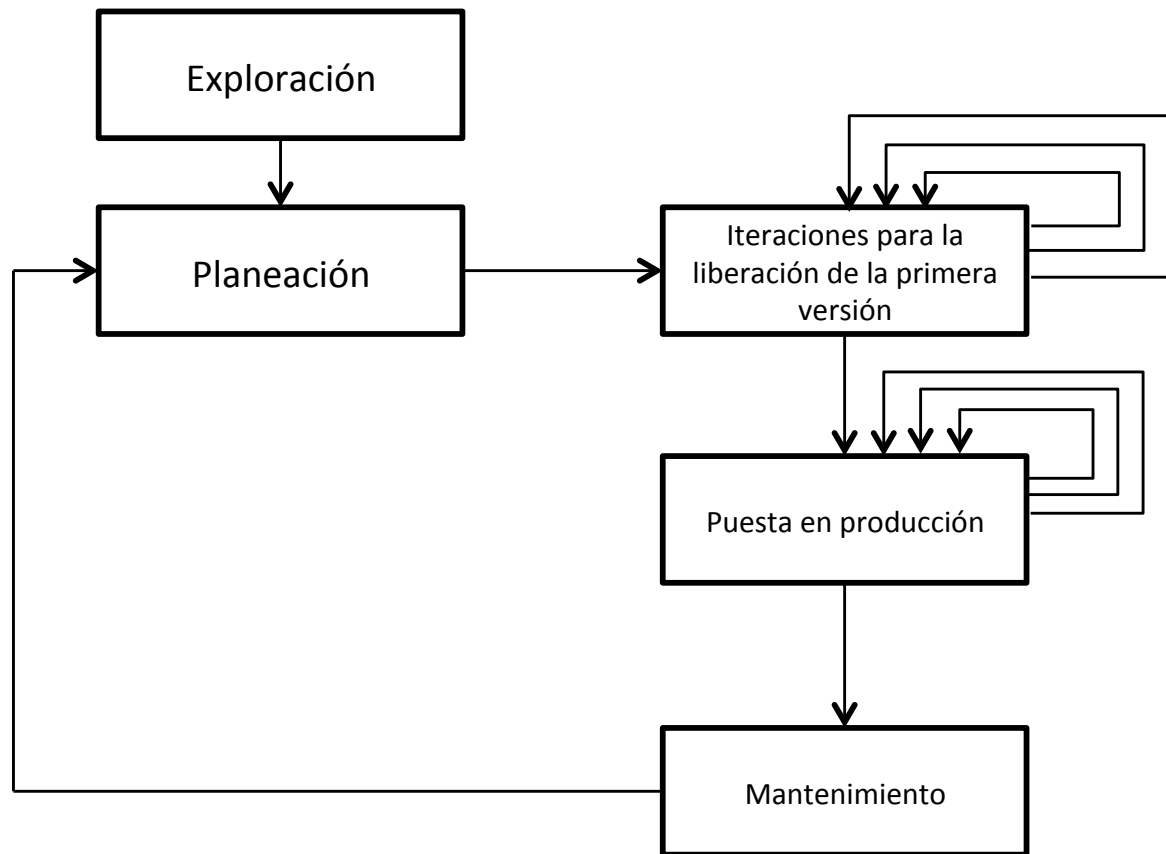


Ilustración 18 Las cinco etapas del modelo de desarrollo ágil. Fuente: adaptado de (Kendall & Kendall, 2011).

A continuación se explica de manera breve el desarrollo de la metodología de desarrollo ágil desde la perspectiva del autor Kendall en su obra *Análisis y diseño de sistemas*:

La primera etapa de esta metodología de desarrollo es la exploración, en esta el analista tiene que tener la habilidad de evaluar el entorno en el que se trabajara el problema a resolver de manera ágil, es imperativo estructurar eficientes equipos de trabajo, evaluar las habilidades de cada uno de sus miembros, así como las tecnologías potenciales para el proyecto.

En esta etapa es importante que el solicitante del desarrollo escriba de una manera suficientemente detallada historias de usuario que ayuden al analista a estimar de forma acertada el tiempo que llevará desarrollar el sistema deseado.

La segunda etapa de esta metodología es la planeación, en la cual el analista tiene que coordinar con el solicitante diversas sesiones para presentarle la solución propuesta para el problema, si las actividades de exploración fueron adecuadas esta etapa debe de ser muy corta.

Las interacciones para liberar la primera versión son la tercera etapa de esta metodología en donde el equipo de desarrollo mostrará el avance del proyecto con el solicitante, para obtener una retroalimentación y trabajar sobre esta.

Una vez realizadas las interacciones necesarias para obtener un producto funcional llega la cuarta etapa de esta metodología la puesta en producción, lo que quiere decir que el sistema estará trabajando con el personal de la organización, sin embargo aún continúan los ciclos de retroalimentación y corrección del sistema.

Finalmente la última etapa es el mantenimiento, el cual debe de encargarse de que el sistema liberado siga funcionando sin problemas.

8. Desarrollo

En este capítulo se exponen los argumentos que justifican las características del desarrollo del proyecto para el caso expuesto en esta tesis, tomando como base el ciclo de vida del desarrollo de sistemas que propone el autor James A. Senn (1992) en su obra *Análisis y diseño de sistemas de información* y el ciclo de vida de desarrollo del autor Kendall (2011) el desarrollo del sistema para esta tesis se llevó a cabo a través de las siguientes actividades:

- Investigación preliminar
- Identificación de problemas, oportunidades y objetivos
- Determinación de requerimientos
- Análisis de las necesidades del sistema
- Diseño del sistema
- Desarrollo del sistema
- Implementación del sistema

El seguimiento de dichas actividades de desarrollo que parten del fundamento teórico de los autores antes citados, garantizó que el desarrollo de la herramienta de software cubriera todos los requerimientos necesarios para atender de una forma adecuada la problemática identificada en el caso de estudio, mediante el uso adecuado de los recursos con los que contaba la organización.

8.1 Análisis de la investigación preliminar

La investigación preliminar surge a partir de la necesidad del caso presentado en esta tesis mediante la optimización en la toma de decisiones para generar nuevas ventajas competitivas en relación a su nicho de negocio. Dicha investigación se llevó a cabo a través de la revisión de documentos y entrevistas al personal directivo de la organización.

El desarrollo detallado de la investigación se encuentra documentado en los antecedentes de este trabajo de tesis y debido a que este capítulo está enfocado a la parte del desarrollo del sistema; se abordará el trabajo de investigación a partir de las conclusiones del análisis del mismo.

El problema que se planteó en la tesis supone que:

La elaboración manual de reportes sobre los principales indicadores clave de desempeño del caso de estudio, dificultan la toma de decisiones del negocio.

Sin embargo, atendiendo el problema desde la causa raíz, la investigación reveló que para mejorar o automatizar de una manera efectiva el análisis de la información del caso expuesto en este trabajo era necesario optimizar también el control operativo del negocio y mejorar sus métodos de almacenamiento.

Atendiendo estas necesidades de forma preliminar no solo se facilitaría el desarrollo de la herramienta de software que automatizara el proceso de reporte del caso de estudio, sino también garantizaría que la información con la que trabajaría esta herramienta tendría un nivel mayor de fiabilidad aumentando la probabilidad de tomar decisiones oportunas para el manejo adecuado del negocio.

Basado en lo anterior, el desarrollo de la herramienta de software que diera solución al problema planteado en esta tesis debía ser abordado más allá de la automatización de reportes, por lo que el proyecto justificó desarrollar la herramienta abordando los siguientes tres componentes de un sistema de inteligencia de negocios a partir de las necesidades particulares del caso de estudio:

- Sistema operacional
- Depósito de datos

- Generador de reportes

El análisis de la solución propuesta dentro de la investigación preliminar tuvo que ser sometido a un estudio de factibilidad técnica, económica y operativa para que el caso de estudio aprobara su desarrollo.

8.1.1 Factibilidad técnica

El estudio desde este aspecto de la factibilidad se resolvió a través de la siguiente pregunta:

¿Puede desarrollarse el proyecto con el equipo actual, tecnología de software y el personal disponible?

Contemplando la infraestructura física con la que contaba el caso de estudio dentro del período de tiempo de desarrollo de la solución, la implementación de un sistema bajo un entorno web era una opción totalmente viable. La infraestructura de red de la organización era la siguiente:

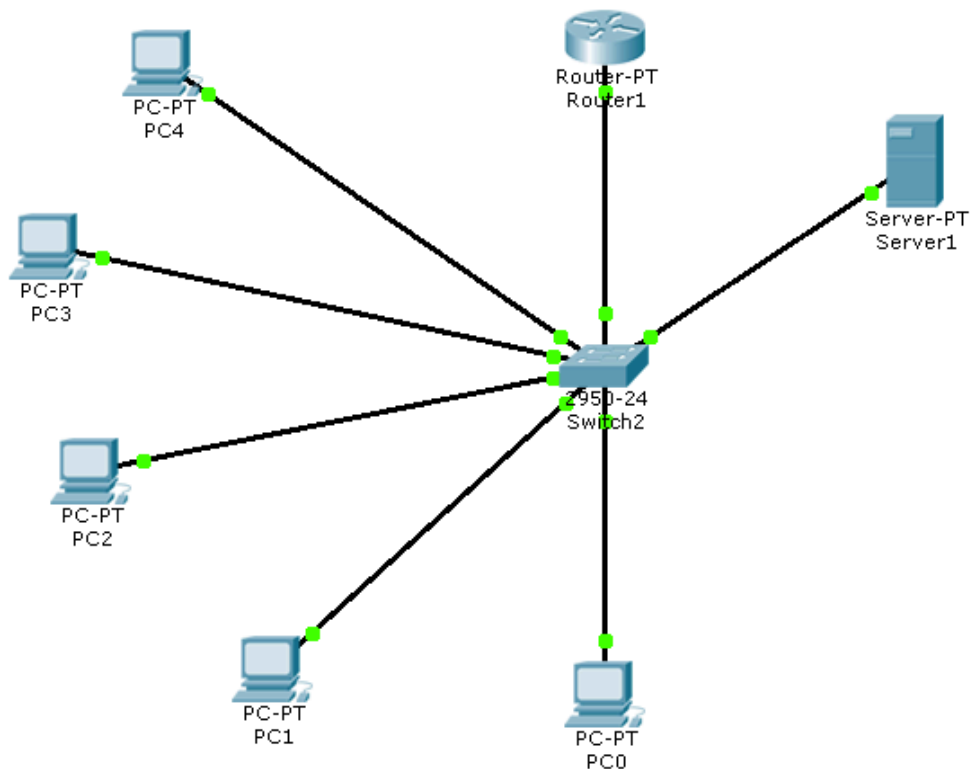


Ilustración 19 Infraestructura de red del caso de estudios. Fuente: Elaboración propia.

Como se puede observar en la ilustración superior como parte de la infraestructura de red del caso de estudio, se encuentra un servidor que se encarga de gestionar los servicios de acceso de usuarios y servicios de impresión el cual cuenta con las siguientes características técnicas:

- Procesador Intel Xeon 3.2 GHz
- Memoria de 4 GB en RAM
- Capacidad de almacenamiento de 500 GB en disco duro.
- Sistema operativo Windows Sever 2008 R2

Los equipos terminales (clientes) cuentan con las siguientes características técnicas:

- Procesador AMD Athlon Dual Core
- Memoria de 4 GB en RAM
- Capacidad de almacenamiento de 500 GB en disco duro
- Sistema operativo Windows 7

En cuanto a la viabilidad técnica se tuvieron que considerar las siguientes restricciones o requerimientos del hardware y software para que el sistema se ejecutara sin inconvenientes.

- Disponibilidad de un servidor para albergar la base de datos
- Disponibilidad de un servidor para levantar un servicio de aplicaciones web
- Disponibilidad de un equipo para el desarrollo del sistema (análisis, diseño, construcción y pruebas)
- Herramientas CASE de libre distribución para la construcción del diseño de la base de datos
- Herramienta IDE para el control y desarrollo del sistema
- Un manejador de base de datos de distribución libre para aplicaciones web
- Un *framework* de desarrollo que soporte el paradigma de la programación orientada a objetos

El caso de estudio contaba con la infraestructura física adecuada para la implementación de los requerimientos y uso de las herramientas necesarias para el desarrollo.

Al contar con una red local ya instalada, levantar un servicio de aplicaciones web que atendiera las solicitudes de los diversos usuarios a través de una arquitectura

cliente-servidor para el sistema no tendría un impacto negativo en el desempeño de los equipos terminales.

El software que se utilizaría para el desarrollo e implementación debía ser totalmente compatible con las características físicas y sistemas operativos del caso de estudio lo cual era totalmente viable ya que en el mercado existen herramientas comerciales y de libre distribución totalmente compatibles para dicha infraestructura.

En cuanto al personal para la realización de este sistema, se tenía la total disposición de colaboración de la mesa directiva y el personal operativo quienes proporcionarían toda la información necesaria para que el proyecto se llevara a cabo; en cuanto a la parte técnica del desarrollo, los conocimientos sólidos proporcionados por la Facultad de Ingeniería permitirían que el desarrollo se llevara a cabo bajo cualquiera de los lenguajes de programación que fueron estudiados dentro del plan de estudios.

De esta forma se concluyó que la factibilidad técnica estaba cubierta totalmente desde los diferentes puntos de análisis antes mencionados.

8.1.2 Factibilidad económica

A partir de los requerimientos técnicos que se determinaron en el análisis de la factibilidad técnica se establecieron las siguientes consideraciones:

- No se destinaría un porcentaje del presupuesto del proyecto para la adquisición de infraestructura ya que el análisis previo reveló que era adecuada para el desarrollo del proyecto.
- El software que se utilizaría para el proyecto se consideraría de libre distribución por lo que el uso de estas herramientas no generarían un impacto económico que impidiera la realización del proyecto.
- Las jornadas para el desarrollo del proyecto no debían de exceder de cuatro horas diarias.

A pesar de las consideraciones anteriores existen costos indirectos como el uso de la infraestructura y la colaboración de los recursos humanos involucrados en el desarrollo del sistema que deben de considerarse ya que generan un impacto en el presupuesto del proyecto.

El siguiente cuadro muestra el número de horas destinadas a cada actividad del plan de desarrollo del proyecto establecido previamente.

Actividades	Número de horas
Investigación preliminar	12
Identificación de problemas, oportunidades y objetivos	12
Determinación de requerimientos	20
Análisis de las necesidades del sistema	20
Diseño del sistema	12

Desarrollo del sistema	70
Implementación del sistema	8
Total 154	

Cuadro 5 Número de horas por actividad de desarrollo. Fuente: elaboración propia.

Como se puede observar en el cuadro anterior, la duración del proyecto basado en jornadas de trabajo de cuatro horas diarias establecen una duración de treinta y un días para la conclusión del proyecto; es claro que la intervención del personal del caso de estudio esta estrechamente ligada a la realización de las tres primeras actividades y disminuye su participación conforme el avance del proyecto.

El personal involucrado en la realización del proyecto del caso de estudio es el siguiente:

- Director General
- Administrador
- Personal operativo

De tal manera y basado en la planeación por horas para cada actividad del proyecto, la mesa directiva evaluó el costo que generaría la intervención de su personal con respecto a la planeación del proyecto determinando así que la factibilidad económica del proyecto era adecuada.

8.1.3 Factibilidad operativa

La factibilidad operativa se analizó bajo el siguiente cuestionamiento:

¿Si se desarrolla y pone en marcha el sistema, será utilizado?

La resistencia al cambio es un factor importante a considerar para el funcionamiento adecuado de cualquier sistema y el desarrollo de este proyecto no es la excepción, para atender este punto se tiene que contemplar un plan de capacitación, el cual debe garantizar que el personal tenga los elementos necesarios para utilizar el software de manera adecuada.

Un punto importante dentro de la factibilidad operativa es que al tratarse de un sistema bajo un ambiente web, todo el personal esta familiarizado con el uso de diversos navegadores, aunado a la colaboración del mismo en cuanto al desarrollo del sistema se puede garantizar que las interfaces sean de fácil uso para el personal dando como resultado una adaptación al cambio de forma adecuada.

La documentación del proyecto, como los manuales de usuario y de administración que se generen durante el desarrollo, son materiales que aportan una base de conocimiento que simplifica de forma significativa la interacción con el sistema, de esta manera el personal tendrá acceso en cualquier momento a esta documentación que resuelva algún conflicto conocido para operar el sistema.

Finalmente, para garantizar que el sistema funcione correctamente a través de su ciclo de vida en la organización es indispensable contemplar y aplicar un plan de mantenimiento el cual debe garantizar el desempeño adecuado de los componentes del sistema tales como la base de datos y el servidor de aplicaciones web por mencionar algunos.

Una vez planteadas todas las consideraciones anteriores la mesa directiva sometió a un proceso interno de evaluación el proyecto, dando como resultado la aprobación del mismo y de esta manera se procedió a continuar con la ejecución de cada una de las actividades que conformaban el desarrollo del proyecto.

8.2 Determinación de requerimientos y análisis de las necesidades del sistema

La determinación de los requerimientos funcionales y no funcionales del sistema se llevó a cabo bajo la ejecución de un **plan de análisis de requerimientos**, el cual se basó en tres etapas en donde se aplicaron diversas técnicas para recabar los datos:

- Etapa I: entrevistas.
- Etapa II: cuestionarios.
- Etapa III: observación.

Etapas I: entrevistas

En esta primera etapa se tuvo un acercamiento inicial con el personal de la organización quienes expresaron características cualitativas de su sistema actual así como comentarios, ideas y sugerencias en relación a cómo podrían optimizarse los procesos en donde ellos interactúan con el sistema.

Después de realizadas las entrevistas se pudo definir en una primera instancia muy general las necesidades particulares de cada uno de los involucrados en el uso del sistema, en donde los hallazgos principales fueron los siguientes:

- El sistema deberá estar constituido por dos módulos: un **módulo operativo** que registre las actividades diarias de la organización y un **módulo de análisis** que permita determinar el estado actual del negocio, así como analizar la información histórica a fin de generar conocimiento que apoye la toma de decisiones.

- Cada módulo necesita un usuario con atributos y autorizaciones propias para desarrollar de forma efectiva sus actividades.
- Se determinaron dos tipos de roles en el sistema: **alumno** e **instructor**

Una vez identificados los roles y usuarios involucrados en el sistema se determinó necesario establecer un nivel de detalle mayor en las interacciones que existen entre estos, de esta manera se aplicaron cuestionarios al personal lo que ayudó a profundizar el análisis.

Etapas II: cuestionarios

Para la ejecución de esta segunda etapa se realizaron cuestionarios al personal operativo y directivo de la institución, después de aplicar dichos cuestionarios el análisis de la información pudo determinar los siguientes **flujos de datos para el personal operativo**:

- Registrar nuevos alumnos
- Reinscribir un alumno a un siguiente nivel
- Actualizar la información del alumno
- Controlar los requisitos para ser instructor
- Invitar a los alumnos a tomar el siguiente nivel consecutivo
- Controlar instructores disponibles por nivel

Como parte del análisis de las necesidades del sistema y basado en la nomenclatura del autor James A. Senn, (Senn, 1992) se realizaron los siguientes diagramas de flujo de datos en un primer nivel:

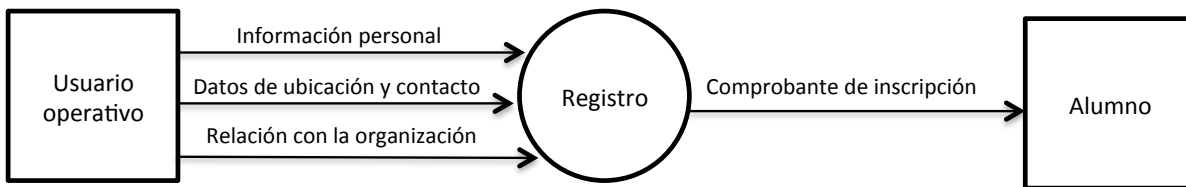


Ilustración 20 Diagrama de flujo de primer nivel para el proceso de registro de nuevos alumnos. Fuente: elaboración propia.

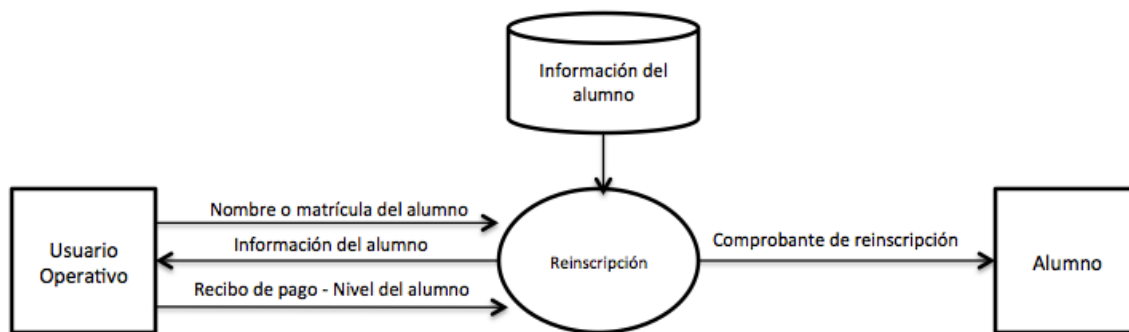


Ilustración 21 Diagrama de flujo de primer nivel para el proceso de reinscripción de un alumno. Fuente: elaboración propia.

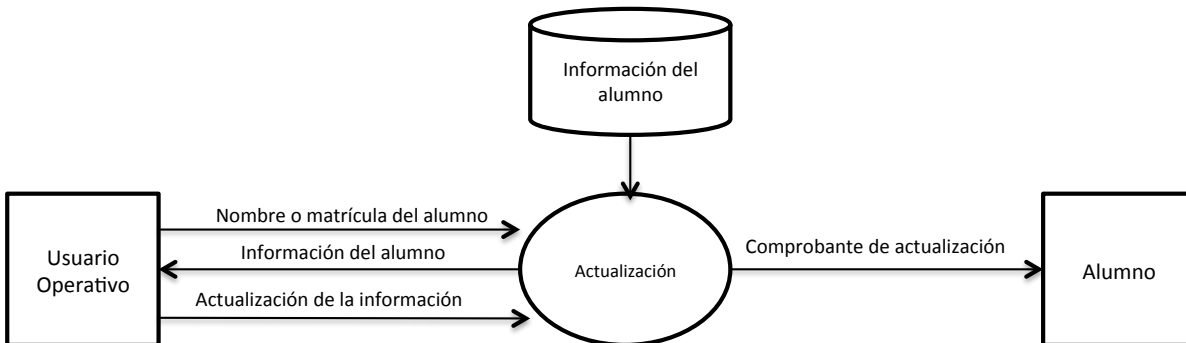


Ilustración 22 Diagrama de flujo de primer nivel para el proceso de actualización de datos de un alumno. Fuente: elaboración propia.

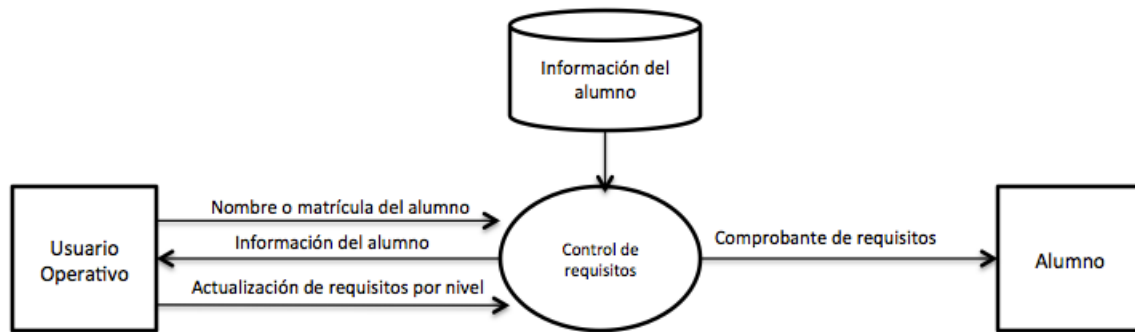


Ilustración 23 Diagrama de flujo de primer nivel para el proceso de control de requisitos para ser instructor. Fuente: elaboración propia.

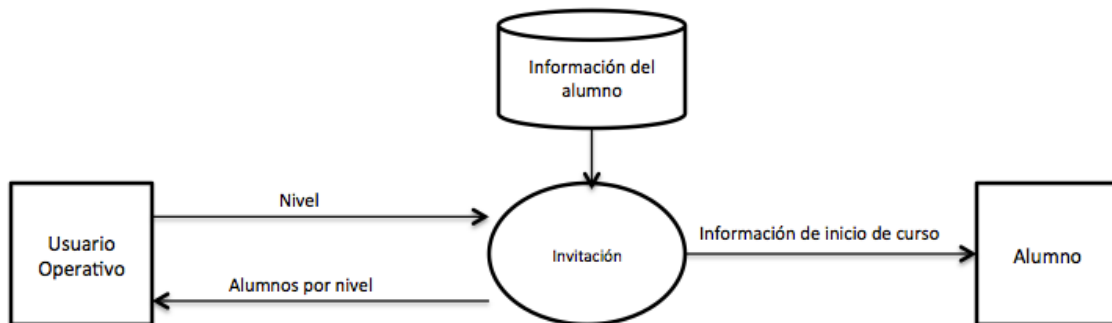


Ilustración 24 Diagrama de flujo de primer nivel para el proceso de invitación al siguiente curso inmediato de los alumnos.

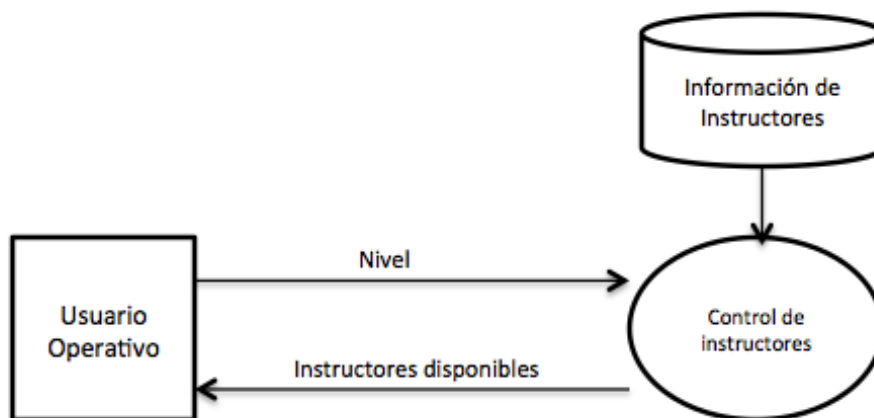


Ilustración 25 Diagrama de flujo de primer nivel del proceso de consulta de instructores activos para dar curso. Fuente: elaboración propia.

La intervención del **personal directivo** con el sistema está basada en la consulta y elaboración de reportes para sustentar la toma de decisiones de la organización, de esta manera se determinaron los principales indicadores que deben ser analizados y comparados dentro de diferentes períodos de tiempo:

- Número de cursos
- Utilidad por curso
- Número de alumnos por curso
- Número de cursos por instructor
- Ingresos / Egresos
- Desglose de Ingresos

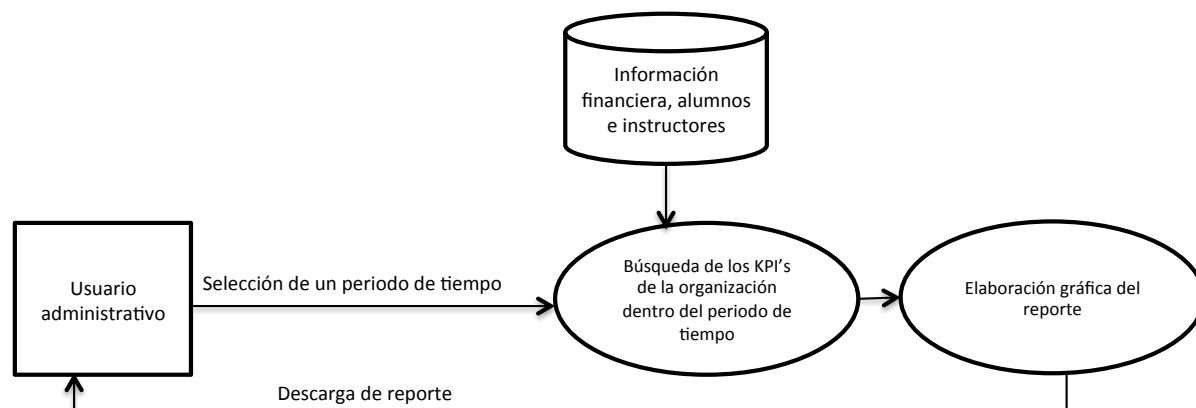


Ilustración 26 Diagrama de flujo de primer nivel del proceso de elaboración del reporte con los principales KPI's de la organización. Fuente: elaboración propia.

Una vez terminados los flujos de datos principales del sistema del personal operativo en un primer nivel y los principales indicadores a medir para el personal directivo fue necesario ejecutar la última etapa del plan de análisis de

requerimientos para definir con un nivel de detalle más alto cada flujo y así generar una extensión de cada uno de estos y conocer con más detalle el proceso de toma de decisiones del personal directivo para automatizar el análisis de la información.

Etapa III: observación

La observación permitió entender de una manera más detallada la forma en que se efectúan las actividades de la organización, se validaron y/o modificaron los flujos que se habían determinado en la entrevista con el personal operativo, dando como resultado las siguientes extensiones de los diagramas de primer nivel:

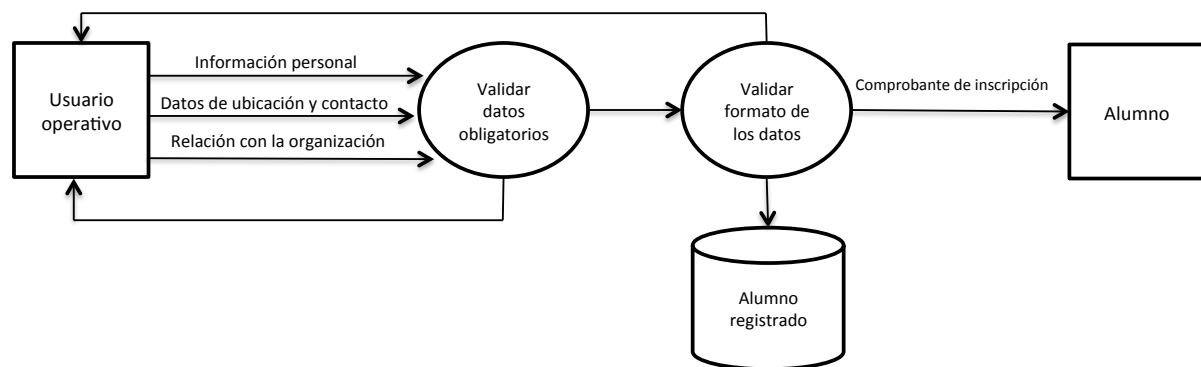


Ilustración 27 Diagrama de flujo de segundo nivel para el proceso de registro de nuevos alumnos. Fuente: elaboración propia.

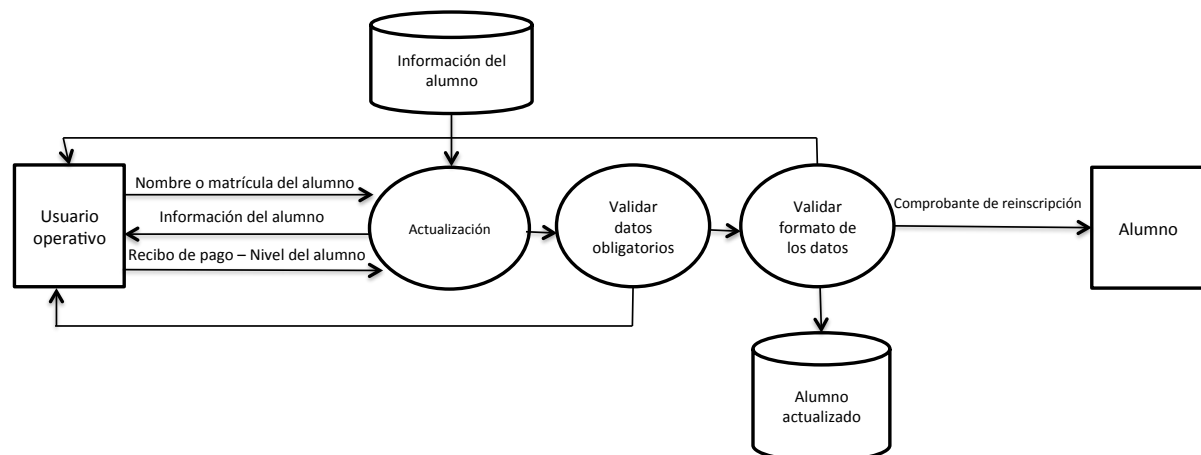


Ilustración 28 Diagrama de flujo de segundo nivel para el proceso de reinscripción de un alumno. Fuente: elaboración propia.

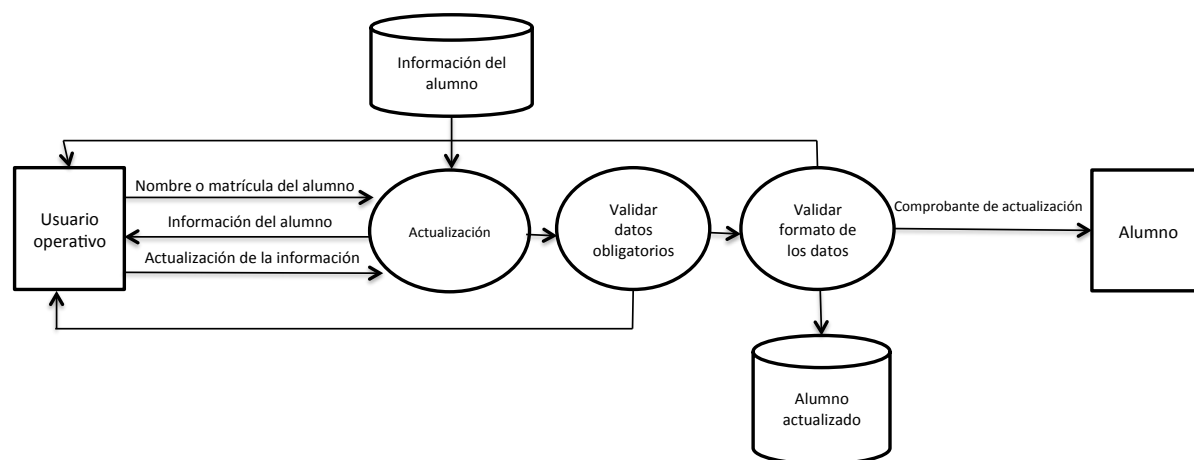


Ilustración 29 Diagrama de flujo de segundo nivel para el proceso de actualización de datos de un alumno. Fuente: elaboración propia.

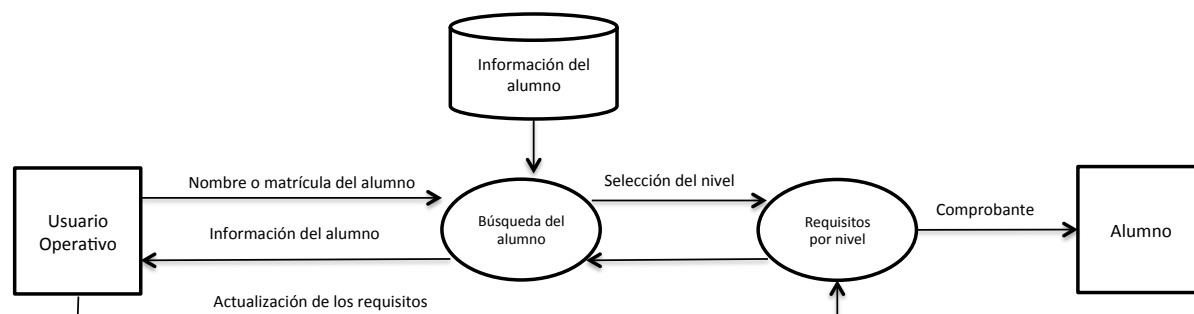


Ilustración 30 Diagrama de flujo de segundo nivel para el proceso de control de requisitos para ser instructor. Fuente: elaboración propia.

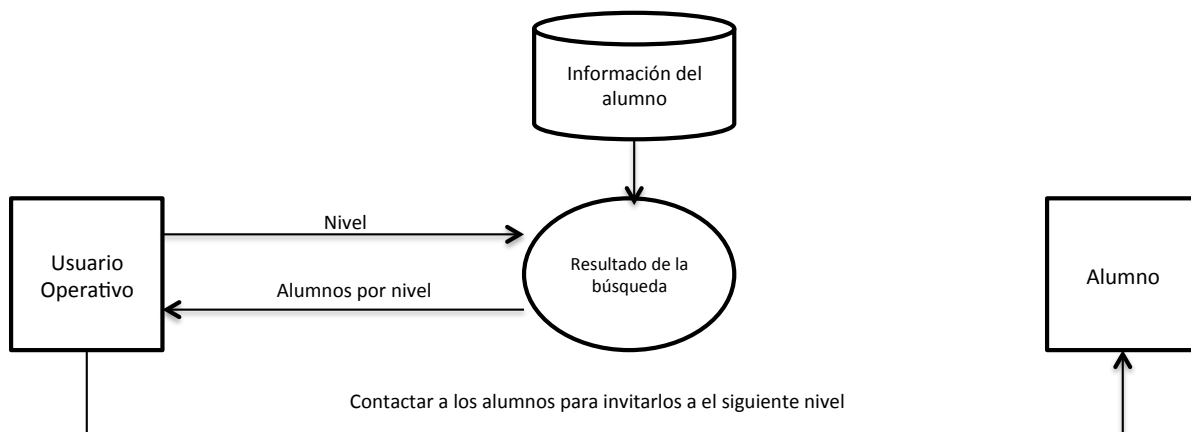


Ilustración 31 Diagrama de flujo de segundo nivel para el proceso de invitación al siguiente curso inmediato de los alumnos.

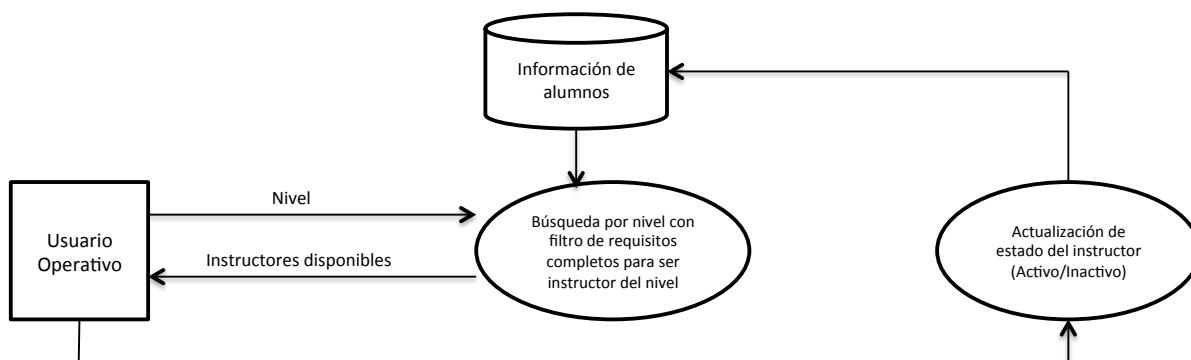


Ilustración 32 Diagrama de flujo de segundo nivel del proceso de consulta de instructores disponibles para dar curso. Fuente: elaboración propia.

Una vez ejecutado completamente el plan de análisis de requerimientos se procedió a realizar dos historias de usuario correspondientes a cada módulo (operativo y análisis) con la finalidad de validar con el personal de la organización que se cubrieran todas las necesidades informáticas de la misma.

8.2.1 Clasificación de los requerimientos

Una vez obtenida toda la información e identificadas las necesidades de la organización se procedió a identificar los requerimientos de cada módulos como funcionales y no funcionales.

8.2.1.1 Requerimientos funcionales - módulo operativo

- El sistema debe controlar el acceso para los usuarios operativos.
- El sistema debe de considerar dos roles: instructor y alumno.
- El sistema debe de permitir al recepcionista el registro de nuevos alumnos, ingresando la siguiente información: información personal, ubicación, contacto, datos de la organización.
- El sistema debe permitir al recepcionista inscribir a los alumnos al siguiente nivel inmediato del plan de formación de la organización, registrando su recibo de pago correspondiente.
- El sistema debe de permitir al recepcionista la actualización de la información del alumno.
- El sistema debe permitir al recepcionista llevar un control de los requisitos para que un alumno pueda ser instructor de un nivel.
- El sistema debe de generar automáticamente una matrícula para cada nuevo alumno.
- El sistema debe de permitir al administrador buscar a los alumnos por nombre o matrícula.
- El sistema debe permitir al recepcionista almacenar el número de licencia y fecha de expedición para cada nuevo instructor.

8.2.1.2 Requerimientos no funcionales – módulo operativo

- El sistema será desarrollado con una interfaz gráfica basada en un entorno web.
- El sistema estará disponible las 24 horas a través de la red local.
- El sistema será accesible desde cualquier equipo de trabajo conectado a la red local.
- El sistema se ejecutará sobre un servidor de aplicaciones web con sistema operativo Windows server 2008 en adelante.
- El sistema trabajara con un manejador de base de datos relacional para entornos web MySQL.

8.2.1.3 Requerimientos funcionales – módulo de análisis

- El sistema debe de generar reportes de manera automatizada con los siguientes indicadores de desempeño de la organización:
 - Número de cursos
 - Utilidad por curso
 - Número de alumnos por curso
 - Ingresos / Egresos
 - Desglose de Ingresos
- El sistema debe de permitir seleccionar diferentes períodos de tiempo para facilitar el análisis del desempeño.

8.2.1.4 Requerimientos no funcionales – módulo de análisis

- Los reportes deberán ser visualizados a través de cualquier software de hoja de cálculo.

8.3 Diseño del sistema

En esta sección se detalla brevemente el proceso de diseño de la solución para el caso de estudio.

8.3.1 Arquitectura de la solución

Basado en los requerimientos anteriores para la solución del caso de estudio la opción más adecuada para el desarrollo del proyecto fue a través de la arquitectura que propone el desarrollo de aplicaciones web mediante la plataforma de java en su edición empresarial (J2EE, Java 2 Enterprise Edition), esta arquitectura no solo integra las bases de la arquitectura cliente servidor sino que también implementa de forma nativa una arquitectura multicapa que permite aislar de forma independiente los componentes de la aplicación web dando como resultado un sistema más seguro, con un nivel mayor de disponibilidad y escalabilidad.

A continuación se ejemplifica la arquitectura de un servidor aplicativo de java a través de la siguiente ilustración:

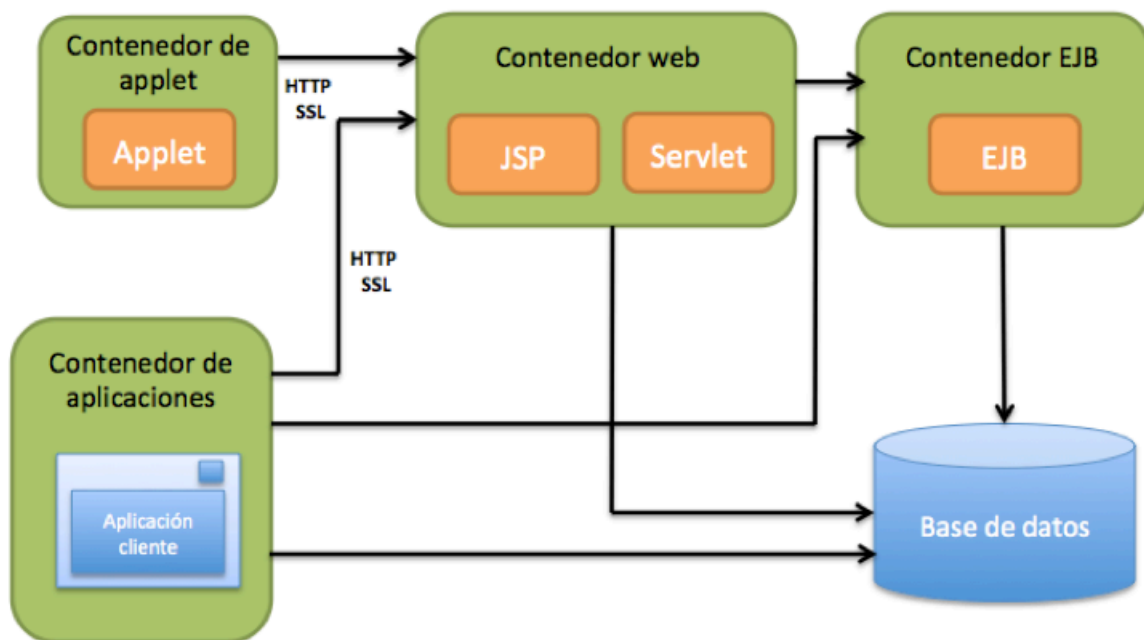


Ilustración 33 Arquitectura de J2EE Application Server. Fuente: adaptado de (Oracle, Docs Oracle).

Mediante el uso de esta arquitectura el desarrollador puede separar la lógica del negocio de la presentación o interfaces gráficas del sistema así mismo puede utilizar los servicios nativos del servidor de aplicación para establecer comunicación con diferentes fuentes de datos, de esta manera es posible atender cada componente del sistema de manera aislada lo que garantiza un desarrollo más ordenado que facilitará la futura administración del sistema.

8.3.1.1 Diseño de la arquitectura de la solución

Finalmente la definición de la arquitectura para este proyecto basado en la funcionalidad que otorga la implementación del servidor de aplicaciones de java queda definida como una arquitectura cliente-servidor de tres capas: presentación, aplicación y datos. La siguiente ilustración define con un mayor nivel de detalle en cuanto al funcionamiento de dicha arquitectura.

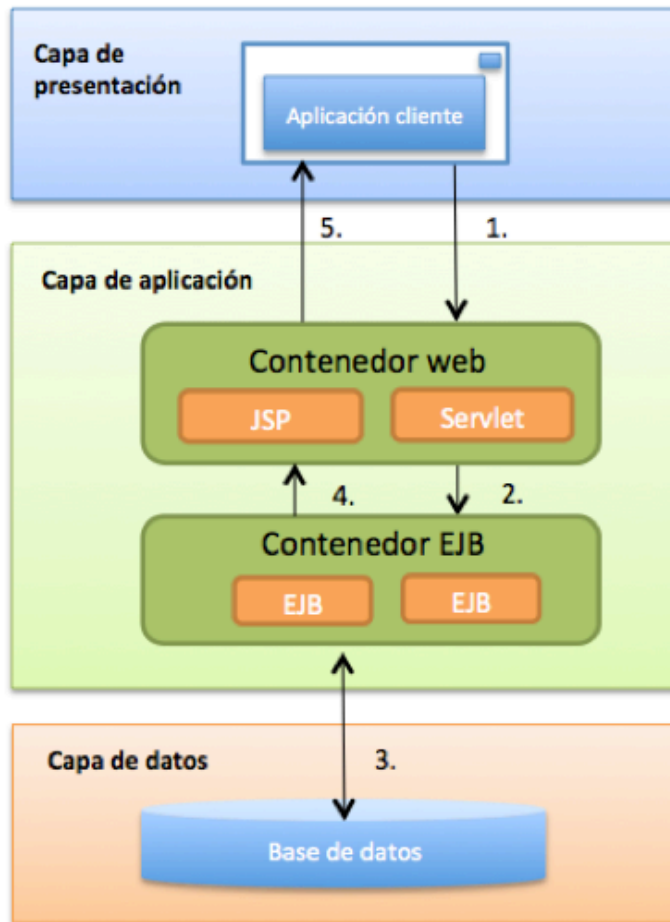


Ilustración 34 Procesamiento de una solicitud de un cliente por un servidor J2EE. Fuente: adaptado de (Oracle, Docs Oracle).

Como se puede observar en la ilustración superior la capa de presentación es aquella con la que el cliente interactúa a través de un navegador web, es así como el cliente envía una serie de peticiones a la capa de aplicación las cuales son atendidas de primera instancia por un *servlet* quien se encarga de definir si dicha solicitud debe de ser atendida mediante la lógica del negocios (EJB) o puede darle una respuesta inmediata mediante un JSP.

Si la solicitud necesita ser atendida por la lógica del negocio (EJB container) esta determinará si es necesario llegar a la capa de datos a través de algún servicio para extraer y procesar la información solicitada o simplemente procesar su

solicitud y responder al cliente nuevamente mediante un JSP el cual se encarga de ofrecerle una vista dinámica de respuesta al solicitante.

Es claro que con el uso de esta arquitectura se cubren plenamente todos los requerimientos previos del sistema pues dicha arquitectura está diseñada para el desarrollo de aplicaciones web, esto significa que se puede tener acceso al sistema desde cualquier equipo que esté conectado a la red local de la empresa.

Al tratarse de una arquitectura java el servidor de aplicaciones puede ejecutarse sin problemas desde el servidor de la empresa, aunado al soporte nativo para interactuar con el manejador de base de datos seleccionado convierten a esta arquitectura la opción más adecuada para el desarrollo del proyecto.

8.3.1.2 Diagrama de base de datos

Una vez definida la arquitectura del sistema es indispensable diseñar la base de datos en donde se concentrará la información de la organización, después de analizar todas las diversas fuentes de datos del caso de estudio la siguiente ilustración muestra el resultado final del diagrama utilizado para el desarrollo de la capa de datos.

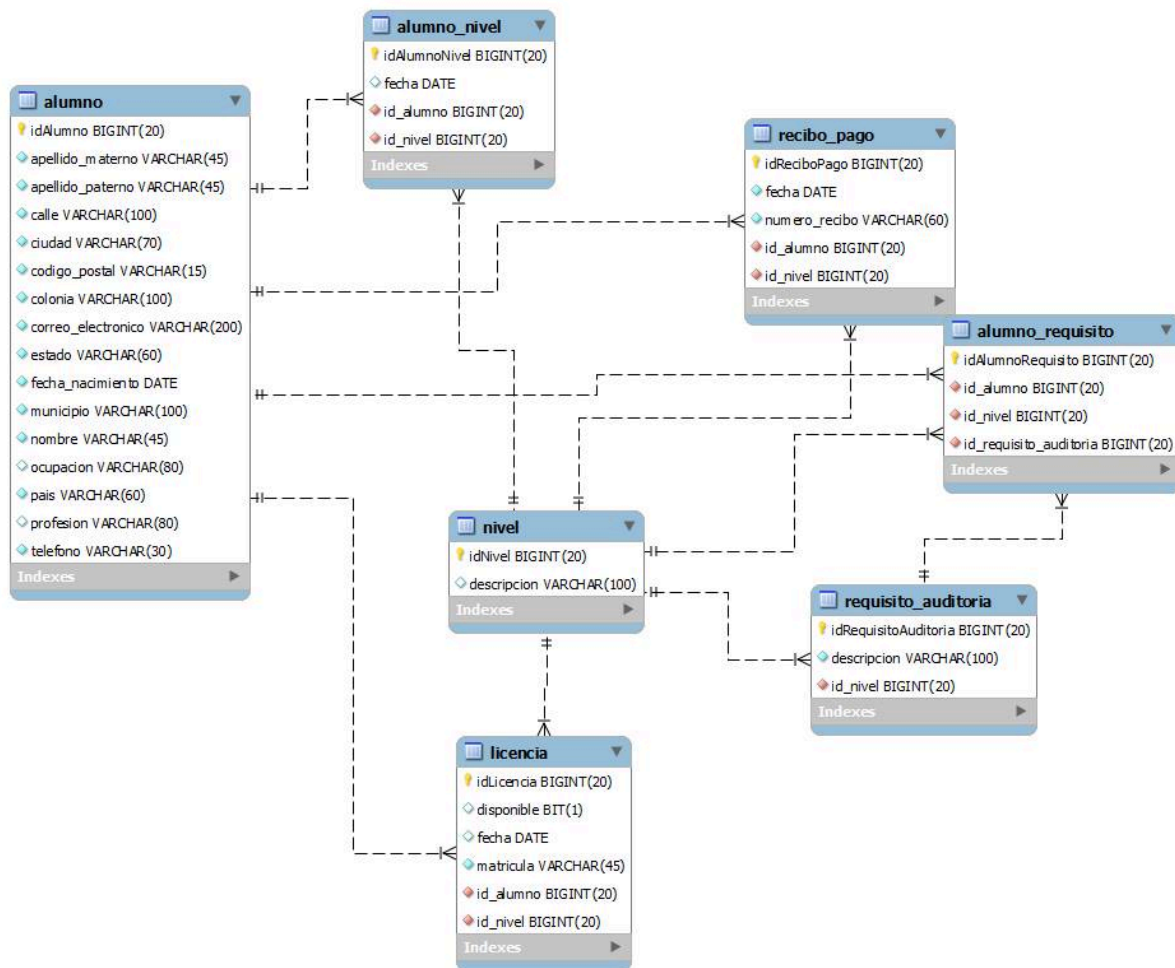


Ilustración 35 Diagrama de la base de datos. Fuente: elaboración propia.

El diagrama de la base de datos propuesto facilitó la exportación de la información histórica del caso de estudio. El proceso de migración se llevó a cabo mediante diversos programas de desarrollo propio dentro del lenguaje de programación java, los cuales estandarizaron y ordenaron la información en un formato CSV para su posterior carga dentro de la base de datos.

8.3.1.3 Diseño de interfaz gráfica

Una vez terminado el diseño de la capa de datos, se decidió trabajar con la capa de presentación a través del diseño de la interfaz gráfica, la ilustración siguiente muestra el resultado final de dicha interfaz la cual fue aceptada por el personal gracias a su simplicidad y fácil manejo.



Ilustración 36 Interfaz gráfica. Fuente: elaboración propia.

Finalmente una vez que se terminó la construcción de la capa de presentación y de aplicación se continuó con el desarrollo de la capa aplicativa la cual contiene la lógica del negocio.

9. Resultados

En este capítulo se encuentran documentados los resultados sobre el desarrollo del sistema y se realiza un breve análisis del impacto que tuvo el sistema al resolver las necesidades previamente planteadas sobre el caso de estudio.

La integración de la información a través de una base de datos relacional permitió que el sistema desarrollado a partir de un entorno web pudiera atender las diversas solicitudes de información para los diferentes usuarios, quienes además reportaron los siguientes beneficios respecto del sistema anterior.

- Mejor disponibilidad de la información
- Mejor control sobre la información
- Mejor organización de la información
- Mejores métodos de acceso

Se lograron establecer controles de seguridad a través de diferentes niveles de acceso de la información mediante la generación de perfiles de usuario los cuales deben seguir cualquiera de los roles que se establecieron en el desarrollo del sistema.

La siguiente ilustración muestra el método de acceso para el sistema a través de un usuario y contraseña asignado por el administrador.

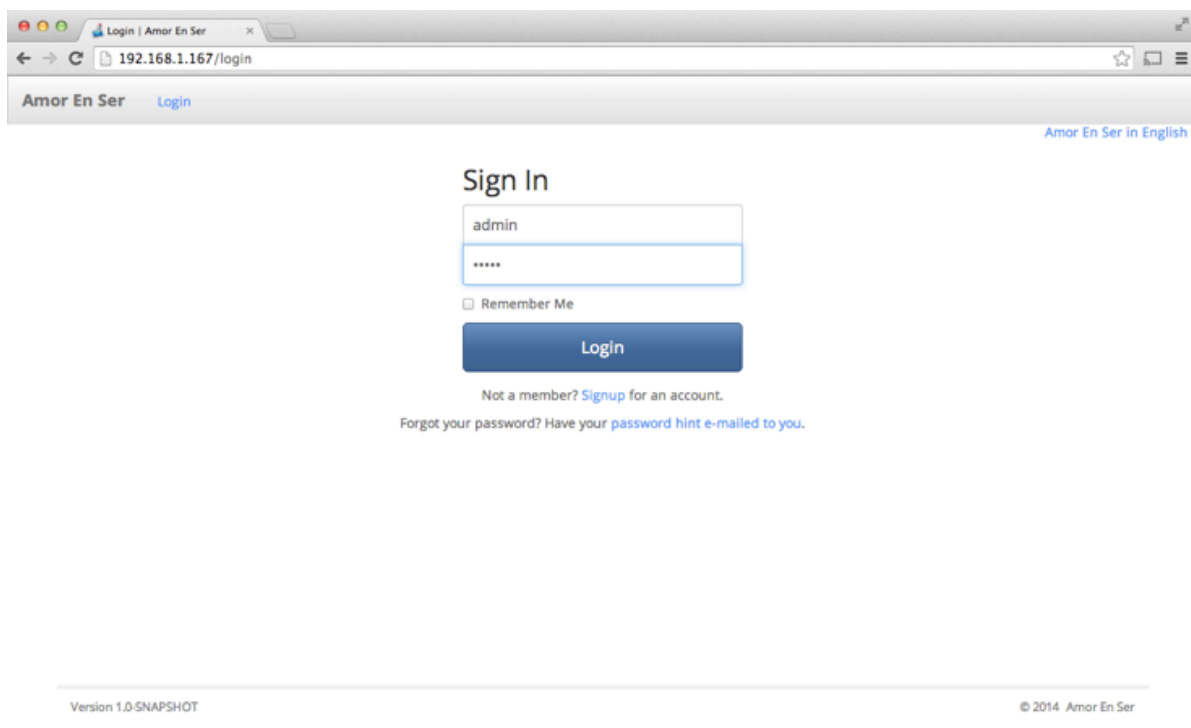


Ilustración 37 Ingreso al sistema mediante usuario y contraseña. Fuente: elaboración propia.

Al controlar el acceso al sistema a través de una autenticación de usuario bajo las credenciales de nombre de usuario y contraseña que fueron asignados a cada miembro de la organización que interactúa con el sistema, el resultado fue un control más adecuado sobre la manipulación de la información. Se crearon perfiles para cada uno de los usuarios dados de alta en el sistema, los cuales reportaron los siguientes beneficios:

- Mayor comunicación entre los involucrados en el sistema
- Mayor control sobre el proceso de actualización de datos

Las siguientes ilustraciones muestran información relacionada con el perfil de usuario, así mismo en la *ilustración 20* se puede observar el panel de administración de usuario el cual permite agregar u modificar tanto usuarios como perfiles.

User Profile
Please update your information.

Username
admin

Password

Confirm Password

Password Hint
Not a female kitty.

First Name
Angel

Last Name
Sánchez

E-Mail
a.sanchez@gccconsultores.com.mx

Phone Number
5542342691

Website
www.gccconsultores.com.mx

Address

Current Roles: ROLE_ADMIN

Version 1.0-SNAPSHOT © 2014 Amor En Ser

Ilustración 38 Información personal y de contacto de los usuario del sistema.
Fuente: elaboración propia.

Current Users

Add Done

(1 of 1) 1 5

Username	Full Name	E-Mail	Enabled
admin	Angel Sánchez	a.sanchez@gccconsultores.com.mx	✓
two_roles_user	Two Roles User	two_roles_user@appfuse.org	✓
user	Tomcat User	matt_raible@yahoo.com	✓

(1 of 1) 1 5

Administration

View Users

Current Users

Niveles

Requisitos para auditoría

Recibos de pago

Administrar cursos

Licencias

Generar reporte

Version 1.0-SNAPSHOT © 2014 Amor En Ser

Ilustración 39 Vista de los usuarios del sistema. Fuente: elaboración propia.

El sistema contiene diversos catálogos de información almacenados dentro de la base de datos los cuales pueden ser actualizados a través del mismo, utilizando un usuario con privilegios de administrador. El giro del caso de estudio presentado en esta tesis está dirigido a la capacitación de personal, lo que lo conlleva a la renovación y actualización de métodos y planes de estudio como un proceso continuo de cambio dentro de la organización.

Los beneficios que los usuarios reportaron en torno a la flexibilidad del sistema para modificar dichos catálogos fueron los siguientes:

- Disminuye significativamente los tiempos de actualización del plan de capacitación para los alumnos activos dentro de la institución.
- Permite mantener el registro histórico de los alumnos a pesar de las modificaciones al plan de capacitación.
- Facilita la reactivación de un alumno independientemente del tiempo que dure su inactividad para completar el plan de capacitación.

A continuación se muestran algunos de los catálogos del sistema:

Niveles | Amor En Ser

192.168.1.167/niveles

Amor En Ser Home Edit Profile Administration Gestión de alumnos Búsqueda rápida Logout

Amor En Ser in English

Niveles

Esta es la lista de los niveles.

Add Done

(1 of 1) 1 5

Id	Descripción	Costo
1	01 HUE NIVEL BÁSICO	
2	02 HUE NIVEL INTERMEDIO 1	
3	03 HUE NIVEL INTERMEDIO 2	
4	04 HUE NIVEL AVANZADO 1	
5	05 HUE NIVEL AVANZADO 2	
6	06 HUE NIVEL ESPECIALIZADO 1 - FILOSOFÍA ORIENTAL	0.0
7	07 HUE NIVEL ESPECIALIZADO 2 - CIENCIA ESPIRITUAL I	
8	08 HUE NIVEL ESPECIALIZADO 3 - CIENCIA ESPIRITUAL II	

(1 of 1) 1 5

Version 1.0-SNAPSHOT © 2014 Amor En Ser

Ilustración 40 Vista del catálogo de niveles de la organización. Fuente: elaboración propia.

Requisitos para auditoría | Amor En Ser

192.168.1.167/requisitoAuditorias

Amor En Ser Home Edit Profile Administration Gestión de alumnos Búsqueda rápida Logout

Amor En Ser in English

RequisitoAuditorias

Esta es la lista de los requisitos para auditoría.

Add Done

(1 of 3) 1 2 3 5

Nivel del requisito	Descripción
01 HUE NIVEL BÁSICO	1. Asistir y obtener un certificado del 08 NIVEL ESPECIALIZADO 3. Ciencia espiritual II con éxito.
01 HUE NIVEL BÁSICO	2. Probar experiencia en el campo de los negocios.
01 HUE NIVEL BÁSICO	3. Asistir al programa de capacitación de formadores para éste nivel.
01 HUE NIVEL BÁSICO	4. Auditar al menos tres niveles básicos
01 HUE NIVEL BÁSICO	5. Firmar un contrato con "EL LICENCIANTE".
01 HUE NIVEL BÁSICO	6. Recibir el número de registro de "EL LICENCIANTE".
01 HUE NIVEL BÁSICO	7. Copia de número de registro a HUE Faculty Inc.
02 HUE NIVEL INTERMEDIO 1	1. Asistir y obtener un certificado a 08 HUE NIVEL ESPECIALIZADO 3. Ciencia espiritual II con éxito.
02 HUE NIVEL INTERMEDIO 1	2. Probar experiencia en el campo de los negocios.
02 HUE NIVEL INTERMEDIO 1	3. Asistir al programa de capacitación de instructores para éste nivel.
02 HUE NIVEL INTERMEDIO 1	4. Auditar al menos tres niveles intermedios 1
02 HUE NIVEL INTERMEDIO 1	5. Haber impartido el nivel 01 HUE al menos cinco veces o equivalente a 50 estudiantes.
02 HUE NIVEL INTERMEDIO 1	6. Firmar un contrato con "EL LICENCIANTE".

Version 1.0-SNAPSHOT © 2014 Amor En Ser

Ilustración 41 Vista del catálogo de requisitos para auditorías de la organización. Fuente: elaboración propia.

La administración de alumnos en el sistema incluye las funcionalidades de filtrado por nivel y búsqueda por nombre y/o apellido, para los usuarios del sistema esta funcionalidad represento los siguientes beneficios:

- Disminución del tiempo de búsqueda de información referente a un alumno.
- Disminución de tiempo en el proceso de planeación de un nuevo ciclo del plan de capacitación.
- Simplificó el proceso de actualización de datos personales y de contacto.

A continuación se muestra una serie de imágenes que muestran el flujo de la manipulación de información sobre la administración de alumnos.

Amor En Ser | Home | Edit Profile | Administration | Gestión de alumnos | Búsqueda rápida | Logout

Amor En Ser in English

Alumnos

Esta es la lista de los alumnos, da click en la matricula del alumno para ver el detalle.

[Add](#) [Done](#)

(1 of 14) 1 2 3 4 5 6 7 8 9 10 5

Matricula	Nombre	Apellido Paterno	Apellido Materno
2001	MAYRA JUANA	CARRERA	GÓMEZ
2002	CARMEN YOLOXOCHITL	NOVELO	TORRES
2003	J. RAYMUNDO	HERNÁNDEZ	LÓPEZ
2004	ESTEBAN DE JESÚS	JIMENEZ	SANTILLAN
2005	DIANA PAMELA	RODRIGUEZ	BEJAR
2006	BLANCA	LOURDES	BAUTISTA
2007	JUAN CARLOS	HI	AGUILA
2008	ANABEL	JUÁREZ	LÓPEZ
2009	GUADALUPE ANGELICA	GARRIDO	VALLEJO
2010	SILVIA	SUAREZ	MORA
2011	RAUL	FRAGOZO	ALCOCER
2012	MARIA EUGENIA	ZAPATA	GUTIERREZ
2013	DIANA LUCERO	SALINAS	PÉREZ

Version 1.0-SNAPSHOT © 2014 Amor En Ser

Ilustración 42 Vista general sobre los alumnos de la organización. Fuente: elaboración propia.

Amor En Ser | Home | Edit Profile | Administration | Gestión de alumnos | Búsqueda rápida | Logout

Amor En Ser in English

Búsqueda de alumnos por nivel

Seleccione un nivel

04 HUE NIVEL AVANZADO 1

(1 of 1) 1 5

Matricula	Nombre	Fecha	Contacto
2207	ABEL ALCALA MADRID	23/06/2014	Correo ALCALAMA@HOTMAIL.COM Teléfono 57602236 / 5527026178
2119	LUIS ALBERTO MELGAREJO GONZALEZ	19/06/2014	Correo ALBERTLUIS@HOTMAIL.COM Teléfono 55111271/5519317848
2206	ADRIANA CORTEZ MORALES	18/06/2014	Correo CORTEZA_WX1966@YAHOO.COM.MX Teléfono 58495562
2336	EVA CATALINA ALBORES RODRIGUEZ	05/06/2014	Correo EVACALBORES@HOTMAIL.COM Teléfono 8747470709

Version 1.0-SNAPSHOT © 2014 Amor En Ser

Ilustración 43 Vista sobre la funcionalidad de búsqueda del sistema. Fuente: elaboración propia.

La información de un alumno dentro del sistema es visualizada como un solo expediente independientemente del número de relaciones que existen dentro de la base de datos para concentrar dicha información, esto derivó para los usuarios del sistema en los siguientes beneficios con respecto al sistema anterior.

- Mayor integración de la información
- Fácil acceso a la información
- Mejor manipulación de la información debido a una mejor organización.

A continuación se muestra las vistas que conforma el expediente para cada alumno de la organización.

The screenshot displays a web browser window with the title 'Detalle del alumno | Amor En Ser'. The address bar shows '192.168.1.167/alumnos'. The navigation menu includes 'Amor En Ser', 'Home', 'Edit Profile', 'Administration', 'Gestión de alumnos', 'Búsqueda rápida', and 'Logout'. The main content area is titled 'Información personal' and contains the following fields:

Nombre	ANGEL
Apellido Paterno	SÁNCHEZ
Apellido Materno	PÉREZ
Calle	CUATRO NO 20
Ciudad	NEZAHUALCOYOTL
Código Postal	57120
Colonia	CAMPESTRE GUADALUPANA

At the bottom of the page, the footer indicates 'Version 1.0-SNAPSHOT' on the left and '© 2014 Amor En Ser' on the right. A link 'Amor En Ser in English' is visible in the top right corner.

Ilustración 44 Vista sobre la información personal de un alumno. Fuente: elaboración propia.

The screenshot shows a web browser window with the address bar displaying '192.168.1.167/alumnos'. The page title is 'Detalle del alumno | Amor En Ser'. The navigation menu includes 'Home', 'Edit Profile', 'Administration', 'Gestión de alumnos', 'Búsqueda rápida', and 'Logout'. The main content area displays a form for a student's personal information. The form fields are as follows:

municipio	
NEZAHUALCOYOTL	

Ocupación	
ADMINISTRADOR DE PROYECTOS	

Pais	
MÉXICO	

Profesión	
INGENIERO EN COMPUTACIÓN	

Información de contacto	
Telefono	
5542342691	
Correo Electronico	
A.SANCHEZ@GCCCONSULTORES.COM.MX	

At the bottom of the form are three buttons: 'Save', 'Delete', and 'Cancel'. The footer of the page shows 'Version 1.0-SNAPSHOT' on the left and '© 2014 Amor En Ser' on the right.

Ilustración 45 Vista sobre la información de contacto de un alumno. Fuente: elaboración propia.

El sistema permite tener un control de los ingresos del caso de estudio presentado en este trabajo de tesis a través del registro de los recibos de pago los cuales crean un histórico del monto, fecha y número de recibo por alumno.

Recibos de pago | Amor En Ser

192.168.1.167/reciboPagoes

Amor En Ser Home Edit Profile Administration Gestión de alumnos Búsqueda rápida Logout

Amor En Ser in English

Recibos de pago de los alumnos

Seleccione un alumno para poder borrar o modificar sus recibos de pago

(1 of 14) 1 2 3 4 5 6 7 8 9 10 5

Matricula	Nombre	Apellido Materno	Apellido Paterno
2001	MAYRA JUANA	GÓMEZ	CARRERA
2002	CARMEN YOLOXOCHITL	TORRES	NOVELO
2003	J. RAYMUNDO	LÓPEZ	HERNÁNDEZ
2004	ESTEBAN DE JESÚS	SANTILLAN	JIMENEZ
2005	DIANA PAMELA	BEJAR	RODRIGUEZ
2006	BLANCA	BAUTISTA	LOURDES
2007	JUAN CARLOS	AGUILA	HI
2008	ANABEL	LÓPEZ	JUÁREZ
2009	GUADALUPE ANGELICA	VALLEJO	GARRIDO
2010	SILVIA	MORA	SUAREZ
2011	RAUL	ALCOCER	FRAGOZO
2012	MARIA EUGENIA	GUTIERREZ	ZAPATA
2013	DIANA LUCERO	PÉREZ	SALINAS
2014	IVONNE	JUÁREZ	PÉREZ

Version 1.0-SNAPSHOT © 2014 Amor En Ser

Ilustración 46 Vista de los recibos de pago por alumno. Fuente: elaboración propia.

ReciboPago List | Amor En Ser

192.168.1.167/reciboPagoes

Amor En Ser Home Edit Profile Administration Gestión de alumnos Búsqueda rápida Logout

Amor En Ser in English

Recibos de pago

Seleccione un recibo de pago para modificarlo o borrarlo

(1 of 1) 1 5

Numero de recibo de pago	Fecha	Numero Recibo
6	05/03/2014	001
292	29/05/2014	002
305	02/06/2014	003

(1 of 1) 1 5

Version 1.0-SNAPSHOT © 2014 Amor En Ser

Ilustración 47 Vista del consecutivo de recibos de pago. Fuente: elaboración propia.

Esta información no solo permite llevar un control de los ingresos si no también permite llevar un control del avance de los alumnos dentro del programa de capacitación.

The screenshot displays a web browser window with the title 'ReciboPago Detail | Amor En Ser'. The address bar shows '192.168.1.167/reciboPagoFiltrado'. The navigation menu includes 'Amor En Ser', 'Home', 'Edit Profile', 'Administration', 'Gestión de alumnos', 'Búsqueda rápida', and 'Logout'. A link for 'Amor En Ser in English' is visible in the top right. The main content area features a form titled 'Informacion de recibo de pago'. This form contains a 'Nivel' dropdown menu set to '04 HUE NIVEL AVANZADO 1' and a 'Numero Recibo' text input field containing '003'. Below these fields are three buttons: 'Save', 'Delete', and 'Cancel'. The footer of the page indicates 'Version 1.0-SNAPSHOT' on the left and '© 2014 Amor En Ser' on the right.

Ilustración 48 Vista de la información de un recibo de pago. Fuente: elaboración propia.

El sistema permite tener un control de los instructores, lo que facilita la labor de calendarización de fechas para nuevos cursos. El sistema informa sobre la disponibilidad de los instructores para participar dentro de un curso así como la

información sobre la licencia que da autorización para capacitar dentro del nivel solicitado.

Recibos de pago | Amor En Ser

192.168.1.167/instructores

Amor En Ser Home Edit Profile Administration Gestión de alumnos Búsqueda rápida Logout

Amor En Ser in English

Esta es la lista de los instructores

Es posible filtrar los instructores por nombre y apellidos.

(1 of 14)

Nombre	Apellido Materno	Apellido Paterno
MAYRA JUANA	GÓMEZ	CARRERA
CARMEN YOLOXOCHITL	TORRES	NOVELO
J. RAYMUNDO	LÓPEZ	HERNÁNDEZ
ESTEBAN DE JESÚS	SANTILLAN	JIMENEZ
DIANA PAMELA	BEJAR	RODRIGUEZ
BLANCA	BAUTISTA	LOURDES
JUAN CARLOS	AGUILA	HI
ANABEL	LÓPEZ	JUÁREZ
GUADALUPE ANGELICA	VALLEJO	GARRIDO
SILVIA	MORA	SUAREZ
RAUL	ALCOCER	FRAGOZO
MARIA EUGENIA	GUTIERREZ	ZAPATA
DIANA LUCERO	PÉREZ	SALINAS
IVONNE	JUÁREZ	PÉREZ
CARLOS	MARIA VALENTE	GÓMEZ

Version 1.0-SNAPSHOT

© 2014 Amor En Ser

Ilustración 49 Vista de los instructores de la organización. Fuente: elaboración propia.

Licencias | Amor En Ser

192.168.1.167/licencias

Amor En Ser Home Edit Profile Administration Gestión de alumnos Búsqueda rápida Logout

Amor En Ser in English

Licencias

Esta es la lista de las licencias.

04 HUE NIVEL AVANZADO 1

Numero de licencia	Alumno	Contacto	Disponible	Fecha	Matricula	Nivel
6	KARLA CHAVEZ CASTILLO	Correo = Teléfono =	NO	2014-06-17	sin matricula	04 HUE NIVEL AVANZADO 1
3	ALICIA SORIA RAZO	Correo = Teléfono 53011594	NO	2014-04-30	sin matricula	04 HUE NIVEL AVANZADO 1

Version 1.0-SNAPSHOT

© 2014 Amor En Ser

Ilustración 50 Vista de las licencias del personal de la organización. Fuente: elaboración propia.

Licencia Detail | Amor En Ser

192.168.1.167/licencias

Amor En Ser Home Edit Profile Administration Gestión de alumnos Búsqueda rápida Logout

Amor En Ser in English

Licencia Information

Disponible ☐

Fecha
08/05/2014

Matricula
sin matricula

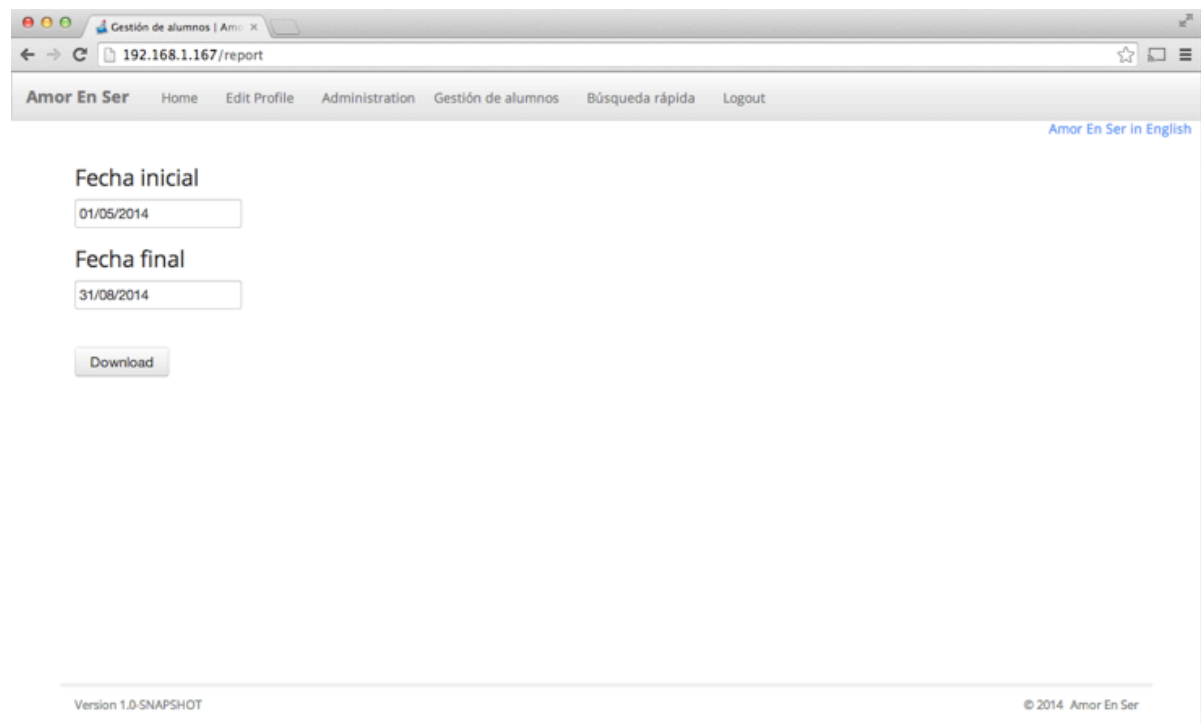
Save Delete Cancel

Version 1.0-SNAPSHOT

© 2014 Amor En Ser

Ilustración 51 Vista sobre la información de las licencias de cada instructor.
Fuente: elaboración propia.

El sistema contiene un módulo que permite al personal directivo establecer un período de tiempo determinado para generar un reporte sobre la situación del negocio de acuerdo a los parámetros previamente establecidos.



The screenshot shows a web browser window with the address bar displaying '192.168.1.167/report'. The page title is 'Gestión de alumnos | Amor En Ser'. The navigation menu includes 'Home', 'Edit Profile', 'Administration', 'Gestión de alumnos', 'Búsqueda rápida', and 'Logout'. The main content area contains two date input fields: 'Fecha inicial' with the value '01/05/2014' and 'Fecha final' with the value '31/08/2014'. Below these fields is a 'Download' button. The footer of the page shows 'Version 1.0-SNAPSHOT' on the left and '© 2014 Amor En Ser' on the right.

Ilustración 52 Vista de control de intervalo de tiempo para los reportes del sistema. Fuente: elaboración propia.

El reporte se descarga automáticamente desde cualquier ordenador que ejecuta dicha instrucción permitiendo analizar la situación del negocio a través de una serie de gráficos que puede ser visualizados desde cualquier manejador de hojas de cálculo. El desarrollo de este módulo de análisis representó los siguientes beneficios para los usuarios:

- Disminución del tiempo de búsqueda y recolección de información para la toma de decisiones.
- Una visión más certera de la situación actual del negocio.
- Un mayor control de la efectividad del negocio.
- Descubrimiento de conocimiento de datos históricos.

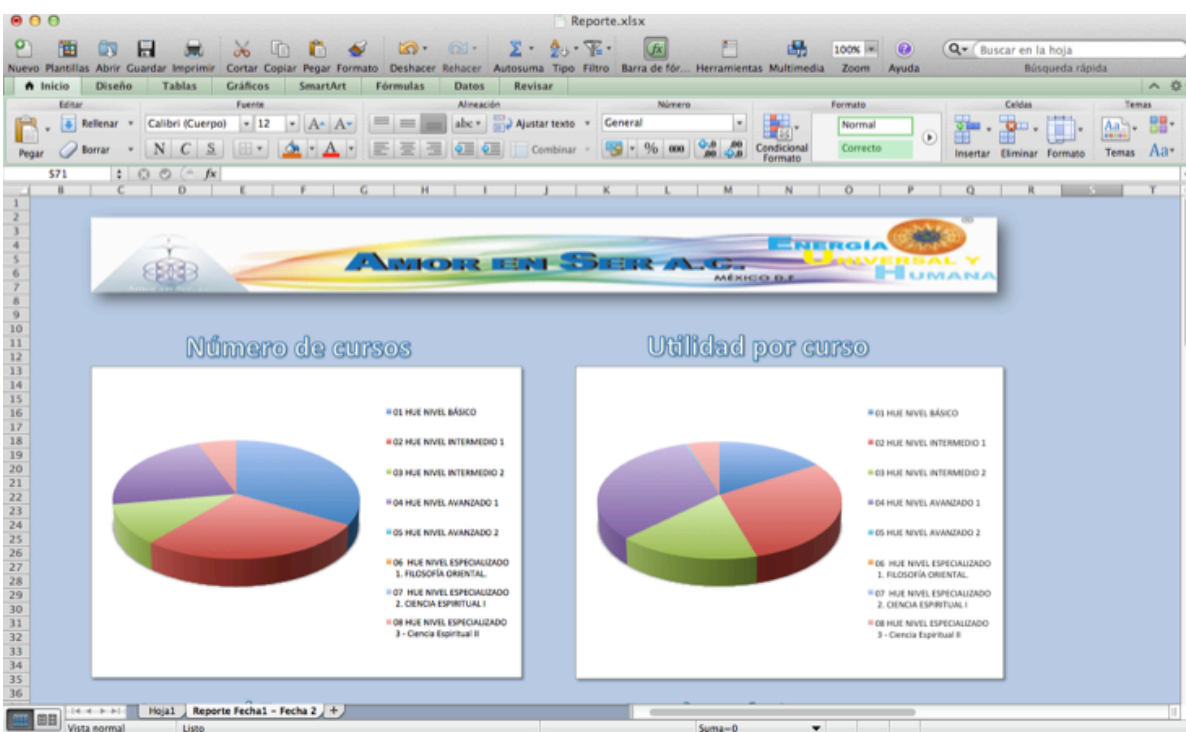


Ilustración 53 Vista superior de un reporte del sistema desarrollado. Fuente: elaboración propia.

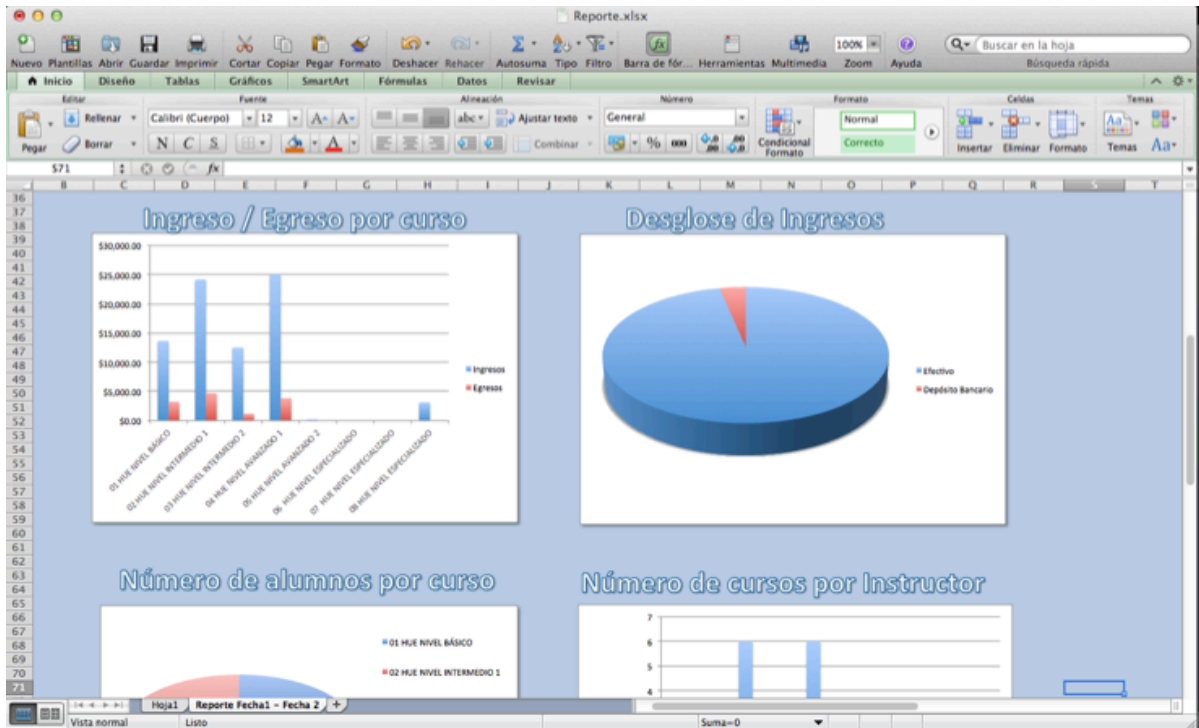


Ilustración 54 Vista media de un reporte del sistema desarrollado. Fuente: elaboración propia.

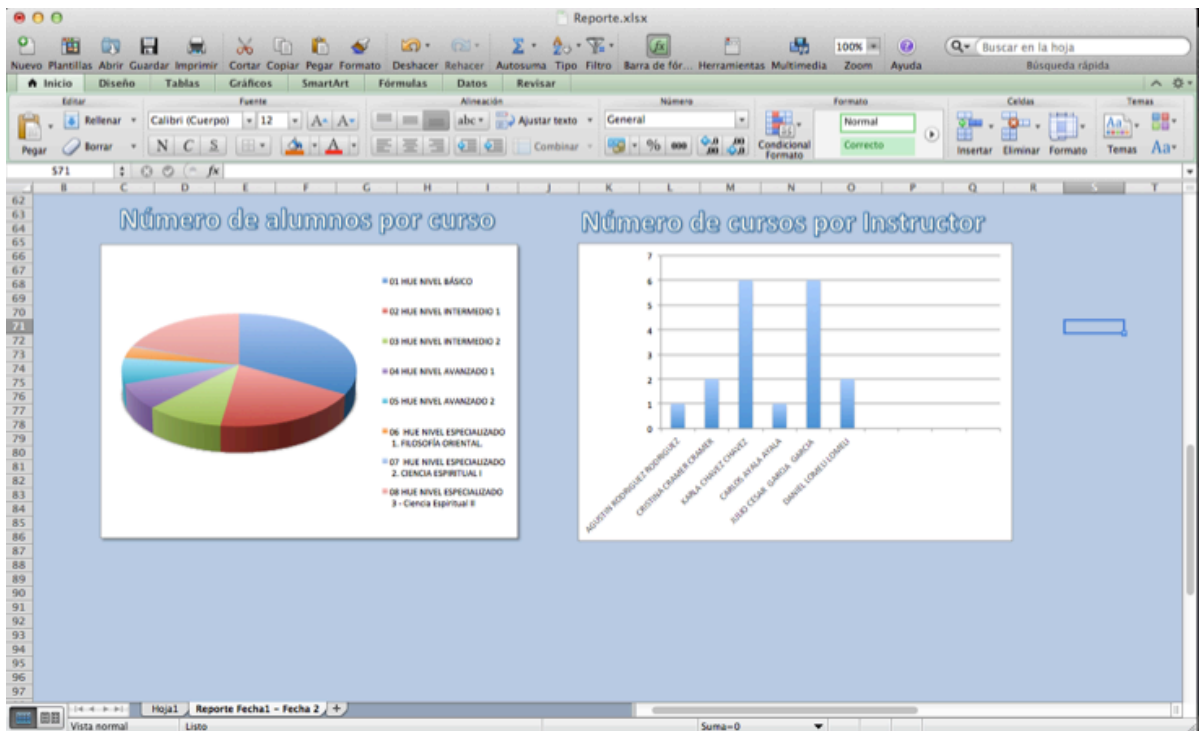


Ilustración 55 Vista superior de un reporte del sistema desarrollado. Fuente: elaboración propia.

Como parte del análisis de los resultados, a continuación se presenta un cuadro con los principales problemas de la organización los cuales fundamentaron la justificación del proyecto y de la misma manera se establecen las soluciones que el proyecto proporciona para la resolución a dichos problemas.

Problemas identificados	Soluciones del proyecto
Excesiva redundancia de la información que impacta negativamente en el desempeño del sistema actual.	Se realizó una migración de la información del sistema antiguo a una base de datos relacional
Deficiente integración de la información de la organización que dificulta la labor del personal operativo.	El sistema desarrollado permite el acceso total a la información necesaria para el cumplimiento de las actividades de cada usuario
Falta de control en la actualización de los datos que amenaza la integridad de la información.	El control de la integridad de la información recae sobre el control nativo de bloqueos del manejador de base de datos.
Dificultad para manipular la información del sistema, lo que dificulta la ejecución de las diversas tareas del negocio.	El sistema desarrollado permite el acceso múltiple de usuarios al sistema, aunado a la simplicidad de navegación y la integración de toda la información dentro de una base de datos única simplifica las tareas de manipulación de

la información.

Realización manual de reportes que generan conocimiento para la toma de decisiones estratégicas del negocio.	Se realizó un módulo de generación de reportes de manera automática para diferentes períodos de tiempo.
--	---

Cuadro 6 Soluciones del sistema para los principales problemas de la organización. Fuente: elaboración propia.

Como se puede observar en el cuadro comparativo superior, la culminación del proyecto resolvió los principales problemas informáticos del caso de estudio presentado en esta tesis, es importante mencionar que para que esto sucediera se debió cumplir con los objetivos definidos durante la planeación y desarrollo del proyecto.

A continuación se presenta un cuadro que relaciona los objetivos previamente establecidos y los beneficios derivados del desarrollo de los mismos. Dichos beneficios fueron reportados por los usuarios después de utilizar el sistema, de esta manera se pretende iniciar un análisis para saber si las hipótesis se cumplieron tras concluir el desarrollo e implementación del proyecto.

Objetivos	Beneficios obtenidos
O1:	1. Disminución del tiempo de búsqueda y recolección de información para la toma de decisiones.
Desarrollar una herramienta de software que automatice la elaboración de reportes sobre los	2. Una visión más certera de la situación actual del negocio.
	3. Un mayor control de la

principales indicadores de desempeño del caso de estudio.	efectividad del negocio.
O2:	4. Descubrimiento de conocimiento de datos históricos.
	1. Mayor disponibilidad de la información.
	2. Mayor control sobre la información.
	3. Mayor organización de la información.
	4. Mejores métodos de acceso.
	5. Mayor comunicación entre los involucrados en el uso del sistema.
	6. Mayor control sobre el proceso de actualización de datos.
	7. Disminuyó significativamente el tiempo de actualización del plan de capacitación para los alumnos activos dentro de la institución.
	8. Permite mantener el registro histórico de los alumnos a pesar de las modificaciones al plan de capacitación.
	9. Facilita la reactivación de un alumno independientemente del tiempo que dure su inactividad para completar el plan de capacitación.
	10. Disminución del tiempo de búsqueda de información referente a un alumno.
	11. Disminución de tiempo en el proceso de planeación de un nuevo

	ciclo del plan de capacitación.
	12. Simplificó el proceso de actualización de datos personales y de contacto.
	13. Mayor integración de la información.
	14. Fácil acceso a la información.
	15. Mejor manipulación de la información debido a una mejor organización.

Cuadro 7 Beneficios derivados del cumplimiento de los objetivos del proyecto. Fuente: elaboración propia.

10. Conclusiones

Una vez analizados los resultados se puede concluir que el proyecto logró mejorar significativamente la administración de la información del caso de estudio, optimizando procesos de almacenamiento, análisis y presentación de los datos. Los procesos anteriores están implícitos dentro de la operación diaria de la organización y al ser integrados en el desarrollo del sistema simplificaron las actividades diarias del personal.

La tesis abordó el concepto de inteligencia de negocios a través de sus componentes mínimos necesarios definidos con anterioridad y es un claro ejemplo de cómo se relacionan dichos componentes para sustentar la toma de decisiones de una organización. El sistema desarrollado para el caso de estudio automatizó el proceso de análisis de la información dándole a la mesa directiva reportes oportunos y confiables para elaborar nuevas estrategias de negocio.

El proyecto demuestra que el concepto de inteligencia de negocios puede ser adoptado por pequeñas organizaciones mexicanas, sin embargo no es simple su adquisición, factores como la infraestructura física, costos de licenciamiento, costos de desarrollo, capacitación entre otros limitan el acceso a este tipo de herramientas que han demostrado ser un apoyo excepcional para el crecimiento de las empresas.

Por lo general los sistemas de inteligencia de negocios son explotados por grandes organizaciones, sin embargo el desarrollo del proyecto para nuestro caso de estudio deja al descubierto que existen necesidades comunes entre las empresas sin importar el tamaño de estas, es decir, una empresa pequeña al igual que una grande necesita información confiable y en tiempo adecuado para tomar decisiones, requiere saber que está haciendo bien y que puede mejorar, precisa administrar sus recursos financieros y optimizar sus operaciones diarias.

Desde la perspectiva anterior pudiera afirmarse que una empresa pequeña necesita aún más un sistema adecuado de inteligencia de negocios pues tiene mayor riesgo de responder tarde a las necesidades cambiantes del mercado y menor capital para sobrevivir ante dichos fracasos, este hecho hace imprescindible pasar por alto la necesidad de atender este sector con soluciones innovadoras pero sobre todo que entiendan el contexto en el que se encuentran dichas empresas.

La tesis a pesar de tratar el problema de forma aislada mediante un caso de estudio debe ser referente de los conflictos que enfrenta una empresa pequeña mexicana para utilizar este tipo de tecnologías pero sobre todo destacar que se logró implementar un sistema de inteligencia de negocios adecuado a la infraestructura y necesidades de la organización.

Este proyecto corresponde a los esfuerzos de la Universidad Nacional Autónoma de México en responder a las necesidades sociales de nuestro entorno y forma parte de la solución, sin embargo es importante exhortar a las entidades gubernamentales e incluso la inversión privada a trabajar en conjunto para generar oportunidades que permitan integrar nuevas y mejores soluciones de inteligencia a todos los sectores empresariales posibles.

Por su parte la Facultad de Ingeniería de la UNAM continúa formando profesionales capaces de atender y entender el problema; la formación que me brindó como egresado de la misma me permitió tener un conocimiento amplio y muy general para hacer frente a un proyecto relacionado con la inteligencia de negocios en donde conseguí poner en práctica todas las habilidades desarrolladas a través de mi estancia como estudiante.

Del párrafo anterior es importante mencionar que el plan de estudios de la facultad me dotó de conocimientos técnicos necesarios para la construcción del proyecto, de acuerdo con la arquitectura que se planteó en la solución del problema, tópicos relacionados con bases de datos, administración de redes y programación entre

otros quedan totalmente cubiertos con las asignaturas impartidas en la carrera de ingeniería en computación.

Es necesario hacer énfasis en un par de asignaturas que desde mi punto de vista logran combinar todos los aspectos técnicos del ingeniero en computación para resolver un problema a partir del desarrollo de software, cuyo caso fue aplicado en la tesis. Las materias de ingeniería de software y administración de proyectos de software fungieron como cinturón de seguridad en el desarrollo del proyecto.

Explicando la idea anterior el dominio de las asignaturas antes mencionadas permite minimizar el riesgo al fracaso de un proyecto de desarrollo, es decir se puede tener un conocimiento técnico extraordinario pero si no se tiene la capacidad de comprender los requerimientos, delimitar alcances, definir metodologías de desarrollo por mencionar algunos puntos, la probabilidad de éxito es muy baja.

Como experiencia profesional la culminación de la tesis me permitió concebir con claridad la relación entre la ingeniería y la inteligencia de negocios que cada día es más grande, continuamente se diseñan nuevas herramientas y tecnologías para tratar de cubrir todos los nichos de negocio; la integración de herramientas que doten de facultades a la pequeña empresa mexicana sigue siendo un gran reto, los resultados del proyecto demuestran que independientemente del tamaño de la organización si se encuentra o desarrolla la herramienta de software adecuada las posibilidades de ejecutar decisiones acertadas y oportunas para su crecimiento aumentan significativamente.

Es importante hacer énfasis sobre el proceso de generación de conocimiento dentro de una organización ya que debe ser tratado como un todo a partir de sus componentes básicos, es imposible pensar en inteligencia de negocios si no se tiene estructurado un sistema que los englobe, al final el éxito de este proceso se basa en el óptimo desempeño de cada uno de sus elementos.

En cuanto a las dos hipótesis planteadas en la tesis se puede concluir lo siguiente:

H1:

Si se desarrolla una herramienta de software que automatice la elaboración de reportes sobre los principales indicadores de desempeño del caso de estudio, facilitará la toma de decisiones del negocio.

R1:

El proyecto demuestra claramente que la comprobación de esta hipótesis es verdadera ya que efectivamente la herramienta de software que automatizó este proceso otorga información oportuna al personal directivo y esto facilita significativamente la toma de decisiones del caso de estudio.

H2:

Si se integra la información de las diferentes áreas operativas del caso de estudio a través de una base de datos relacional aumentará significativamente el nivel de integridad, seguridad y disponibilidad de la información respecto al sistema actual.

R2:

La comprobación de esta segunda hipótesis es verdadera pues en definitiva el uso de un manejador de base de datos relacional aporta mayor control de la información comparado con un sistema de archivos que para el caso de estudio era el tipo de sistema con el que se contaba.

A partir de la validación de ambas hipótesis es trascendental indicar que si se analiza cada una de ellas de forma aislada difícilmente se lograría tener un sistema que aportara un auténtico valor a la toma de decisiones del caso de estudio, de

nada serviría tener información oportuna si la información que es procesada por el sistema presentara múltiples inconsistencias y por otro lado no tendría sentido almacenar información histórica si no es posible analizarla de forma adecuada para desarrollar procesos de mejora continua dentro de la organización.

Recomendaciones

Se recomienda ampliamente a la organización respaldar toda la documentación del proyecto como: manuales de usuario, archivos de preguntas frecuentes (FAQ), especificación de requisitos del software (SRS), descripción de diseño del software (SDD) y la declaración de trabajo (SOW) así como el código fuente de la aplicación para controlar futuras versiones que pudieran desarrollarse. Es de suma importancia seguir el plan de mantenimiento del sistema para garantizar que el desempeño del mismo sea adecuado y ejecutar con regularidad el *script* de respaldo de información de la base de datos.

Es vital considerar dentro del presupuesto del próximo ciclo la adquisición de una póliza de mantenimiento preventivo para su infraestructura lo que disminuirá sustancialmente el riesgo de fallo del sistema o pérdidas de información a consecuencia de problemas físicos (hardware) o lógicos (software) ajenos al desarrollo del sistema.

Es altamente recomendable que la póliza de mantenimiento incluya un mantenimiento de alto nivel para los componentes del sistema a fin de garantizar una disponibilidad adecuada de elementos como: base de datos y servidor aplicativo.

11. Bibliografía

- Curto Díaz, J. (2010). *Introducción al Business Intelligence*. España (Barcelona): Editorial UOC.
- Senn, J. A. (1992). *Análisis y diseño de sistemas de información*. Macgraw-hill.
- Singh Gill, H. (1996). *The Official Client/server Computing Guide to Data Warehousing*. Macmillan Publishers Limited.
- W. Hubbard, D. (2007). *How to Measure Anything; Finding the Value of Intagibles in Business*. Hoboken , NJ: John Wiley & Sons Publishers.
- Wayne W., E. (2006). *Performance Dashboards: Measuring, Monitoring and Managing your Busines*. Hoboken, NJ: John Wiley and Sons Publishers.
- Wrembel, R. (2007). *Data warehouses and OLAP : concepts, architectures and solutions*. Hershey Pennsylvania.
- Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*. Milano, Italia: John Wiley & Sons Ltd.
- Thomsen, E. (2002). *OLAP solutions: Building multidimensional information system*. New York: Wiley Computer.
- Bonilla, I. M. (2013). *Modelado de Procesos y Construcción de Indicadores de Desempeño*. México, D.F.
- Elmasri, R. (2007). *Fundamentos de Sistemas de Bases de Datos*. México, D.F, México: Pearson Educación S.A.
- Evelson, B. (2014). *Agile Business Intelligence Platforms* . The Forrester Wave.
- Hernández Orallo, J. (2004). *Introducción a la Minería de Datos*. Madrid: Pearson .
- Hernández, M. Á., & Saldívar Mendoza, J. G. (2009). *Implantación de un área de operación para un Datawarehouse*. México, D.F.
- García, R. R. (2012). *Minería de Datos para la toma de decisiones e Inteligencia de Negocios: Aplicaciones en la Mercadotecnia*. México, D.F.
- Kendall, K., & Kendall, J. (2011). *Análisis y diseño de sistemas*. Prentice Hall.
- Kerzner, H. (2011). *Project Manajemet Metrics, KPIs and Dashboards*. Hoboken, NJ: John Wiley & Sons, Inc. Pubishers.
- Mourya, S. K. (2013). *Data Mining and Data Warehousing*. Alpha Science International.
- Parameter, D. (2007). *Key Performance Indicators*. Hoboken, NJ: John Wiley & Sons Publishers.
- Peterson, E. (2006). *The Big Book of Key Performance Indicators*.
- Peña Ayala, A. (2006). *Inteligencia de Negocios: Una Propuesta para su Desarrollo en las organizaciones*. México, D.F: Dirección de Publicacioens.
- Phillips, J. (2013). *Building a Dgital Analytics Organization*. Upper Saddle River, New Jersey: Pearson.
- Ponniah, P. (2010). *Data warehousing fundamentals for it professionals*. John wiley & sons, INC.
- Oracle. (s.f.). *Docs Oracle*. Obtenido de <https://docs.oracle.com/cd/E19830-01/819-4712/ablat/index.html>
- Oracle. (s.f.). *Docs Oracle*. Obtenido de <https://docs.oracle.com/javaee/5/tutorial/doc/bnaay.html>

12. Anexos

12.1 Historia de usuario: módulo operativo

Usuario:

- Recepcionista: Entidad de interacción directa con el Sistema

Roles:

- Alumno: Entidad de interacción indirecta en el sistema
- Instructor: Entidad de interacción indirecta en el sistema

Recepcionista:

Actividades:

Registrar nuevos alumnos

Yo como recepcionista tengo que registrar a los nuevos alumnos en el sistema bajo el siguiente procedimiento:

1. Registrar información de los nuevos alumnos:
 - a. Información Personal
 - i. Nombre
 - ii. Apellido Paterno
 - iii. Apellido Materno
 - iv. Fecha de Nacimiento
 - v. Profesión
 - vi. Ocupación
 - b. Ubicación
 - i. País
 - ii. Estado

- iii. Ciudad
- iv. Colonia
- v. Municipio
- vi. Calle
- vii. Código Postal
- c. Contacto
 - i. Teléfono (Local)
 - ii. Teléfono (Móvil)
 - iii. Correo Electrónico
- d. Datos de la Organización
 - i. Matrícula *
 - ii. Nivel (8 Niveles)
 - iii. Número de Recibo de Pago **

- La matrícula la debe de generar el sistema
- Es un registro numérico generado por la administración (no es necesario automatizarlo en el sistema).

Reinscribir al alumno a un siguiente nivel

Yo como recepcionista tengo que reinscribir a un alumno a su siguiente nivel inmediato cuando este me muestre su certificado de pago, de esta manera procedo a ejecutar el siguiente procedimiento:

1. Buscar al alumno por Nombre o Matrícula
2. Actualizar el número de recibo de pago
3. Actualizar el Nivel Actual del alumno

Actualizar la información del alumno

Yo como recepcionista tengo la necesidad de actualizar la información de los alumnos, por lo que cuando alguno me lo solicita procedo a realizar el siguiente procedimiento:

1. Buscar al alumno por Nombre o Matrícula
2. Actualizar la información solicitada por el alumno

Llevar un control de los requisitos para que un alumno pueda ser instructor de un nivel

Cuando un alumno termina un nivel tiene la oportunidad de realizar una serie de actividades que son requisitos (8 *checks*) para poder ejecutar ese nivel como instructor. Por lo que yo como recepcionista tengo que llevar un control de esas actividades por nivel acreditado por el alumno.

1. Revisar de manera visual los comprobantes que certifiquen los requisitos del nivel por el que se está preparando para ser instructor.
2. Buscar al alumno por Nombre o Matrícula.
3. Seleccionar el nivel por el que se está preparando para ser instructor.
4. Actualizar el *checklist* de los requisitos cumplidos para el nivel.
5. Si el alumno cumple con todos los requisitos del nivel para ejecutarlo como instructor solicitar a la administración una licencia.
6. Almacenar el número de licencia y la fecha de expedición.

Invitar a los alumnos a tomar el siguiente nivel consecutivo

Al finalizar los cursos yo como recepcionista tengo la encomienda de invitar a los alumnos vía telefónica o vía e-mail al siguiente nivel consecutivo con respecto a su avance en la institución, por lo que tengo que seguir el siguiente procedimiento:

1. Filtrar alumnos por nivel
2. Visualizar la información de los alumnos (vista de sólo información de contacto).
3. Llamar y/o enviar correo electrónico para invitar a los alumnos al próximo nivel consecutivo

Consultar Instructores disponibles por nivel

Para la planeación sobre la apertura de nuevos cursos necesito consultar información sobre los instructores disponibles por curso

1. Filtrar Instructores (alumnos que cumplen con todos los requisitos para dar un nivel) por nivel.
2. Visualizar la información de los instructores (vista de sólo información de contacto)

12.2 Historia de usuario: módulo de análisis

Usuario:

- Director General

Roles:

- Alumno
- Instructor

Actividades:

Yo como director general necesito medir los siguientes indicadores de desempeño de la organización:

- Número de cursos que se impartieron desglosados por nivel
- Número de alumnos por curso
- Número de cursos por instructor

Yo como director general necesito controlar la siguiente información financiera de la organización:

- Ingresos y egresos de cada curso
- Utilidad por curso
- Desglose de los ingresos (efectivo y depósito bancario)

Yo como director general necesito generar un reporte que contenga esta información de forma automática y visualmente clara para diversos períodos de tiempo dando como resultado una comparativa que sustente la toma de decisiones del negocio.

12.3 Especificación de requisitos del software (SRS, Software Requirements Specification)

I. Introducción

Dentro de este documento se encuentran los requerimientos generales sobre el funcionamiento del sistema desarrollado para la Asociación Civil registrada ante el registro federal de contribuyentes como: **Amor en Ser A. C.**

I.1. Propósito

Especificar de forma clara la funcionalidad del sistema en cuestión al personal responsable del proyecto dentro de la organización, la cual deberá cubrir satisfactoriamente los requerimientos especificados dentro de este documento.

I.2. Personal involucrado

Nombre	Angel Roberto Sánchez Pérez
Rol	Analista y desarrollador
Categoría profesional	Pasante de Ingeniería en Computación
Contacto	angelsanchez.fi.unam@gmail.com

Nombre	-----
Rol	Director general
Categoría profesional	Licenciado en Derecho
Contacto	-----

Nombre	-----
---------------	-------

Rol	Administrador
Categoría profesional	Licenciado en Administración de Empresas
Contacto	-----

I.3. Referencias

Pressman S. Roger (2005). *Ingeniera del Software: Un enfoque practico*. Mc Graw-Hill. México

II. Descripción general

II.1. Funcionalidad del producto

El sistema debe de interactuar con la operación diaria del centro de capacitación, manteniendo un control sobre la trayectoria de los alumnos dentro de la organización, simplificando la administración de nuevos cursos mediante un registro de instructores que permita conocer de forma simple el nivel de conocimiento de los instructores y la disponibilidad de estos y así calendarizar las sesiones de trabajo de una forma más asertiva, así mismo el sistema debe de registrar las operaciones financieras que la organización realice dentro de sus actividades diarias.

La información anterior debe estar organizada de una forma simple, eliminando redundancias de la información que puedan impactar negativamente en el desempeño del sistema, sin embargo este debe de garantizar que la integración de dicha información permita al personal efectuar de manera adecuada sus actividades laborales diarias.

El sistema debe de contemplar que existen registros históricos de las operaciones de la organización, los cuales constantemente son analizados por el personal de toma de decisiones quienes se encargan de elaborar estadísticas sobre el comportamiento del negocio respecto de ciclos anteriores, de tal forma el sistema debe de considerar la integración de dicha información y lograr automatizar el proceso de elaboración de reportes estadísticos sobre los principales indicadores de desempeño establecidos por la organización.

II.2. Interfaces externas

La infraestructura de red de la organización esta conformada por un servidor y cinco equipos clientes con las siguientes características:

- Procesador Intel Xeon 3.2 GHz
- Memoria de 4 GB en RAM
- Capacidad de almacenamiento de 500 GB en disco duro
- Sistema operativo Windows Sever 2008 R2

Las características técnicas de los equipos terminales (clientes) son las siguientes:

- Procesador AMD Athlon Dual Core
- Memoria de 4 GB en RAM
- Capacidad de almacenamiento de 500 GB en disco duro
- Sistema operativo Windows 7

El sistema debe de interactuar con dicha infraestructura y soportar múltiples accesos de forma simultanea desde cualquier equipo terminal conectado a la red local de la organización.

II.3. Rendimiento

La disponibilidad del sistema debe estar garantizada durante las jornadas laborales de la organización definidas a continuación:

Lunes a Viernes	9:00 am – 6:00 pm
Sábado	9:00 am – 2:00 pm

II.4. Atributos

- Las actividades programadas de mantenimiento del sistema deberán efectuarse a través de una ventana de tiempo autorizada por el personal administrativo a fin de mantener tiempos de respuesta adecuados para la operación del sistema y garantizar la seguridad de la información.
- El sistema debe de controlar los accesos a fin de asegurar que solo el personal adecuado manipule la información de la organización.
- El sistema debe de tener la capacidad de atender múltiples solicitudes de forma paralela.

II.5. Restricciones del diseño en la implementación

La infraestructura actual de la organización deberá de soportar la operación del sistema, las herramientas de software para el desarrollo del sistema deberán ser de libre distribución y código abierto.

III. Requisitos específicos

- **Registrar nuevos alumnos**
 - a. Información Personal
 - i. Nombre
 - ii. Apellido Paterno
 - iii. Apellido Materno
 - iv. Fecha de Nacimiento
 - v. Profesión
 - vi. Ocupación

b. Ubicación

- vii. País
- viii. Estado
- ix. Ciudad
- x. Colonia
- xi. Municipio
- xii. Calle
- xiii. Código Postal

c. Contacto

- xiv. Teléfono (Local)
- xv. Teléfono (Móvil)
- xvi. Correo Electrónico

d. Datos de la Organización

- xvii. Matrícula *
- xviii. Nivel (8 Niveles)
- xix. Número de Recibo de Pago **

* La matrícula la debe de generar el sistema

** Es un registro numérico generado por la administración (no es necesario automatizarlo en el sistema).

- **Reinscribir alumnos**

- a. Buscar al alumno por nombre o matrícula
- b. Actualizar el número de recibo de pago
- c. Actualizar el nivel actual del alumno

- **Actualizar alumnos**

- a. Buscar al alumno por nombre o matrícula
- b. Actualizar la información solicitada por el alumno

- **Control de instructores**

- a. Revisar de manera visual los comprobantes que certifiquen los requisitos del nivel por el que se está preparando para ser instructor.
- b. Buscar al alumno por nombre o matrícula.
- c. Seleccionar el nivel por el que se está preparando para ser instructor.
- d. Actualizar el *checklist* de los requisitos cumplidos para el nivel.
- e. Si el alumno cumple con todos los requisitos del nivel para ejecutarlo como instructor, solicitar a la administración una licencia.
- f. Almacenar el número de licencia y la fecha de expedición.

- **Generar un reporte sobre los KPIs**

- a. Número de cursos que se impartieron desglosados por nivel
- b. Número de alumnos por curso
- c. Número de cursos por instructor
- d. Ingresos y egresos de cada curso
- e. Utilidad por curso
- f. Desglose de los ingresos (efectivo y depósito bancario)

12.4 Descripción del diseño del software (SDD, Software Design Document)

I. Introducción

En este documento se encuentra la descripción sobre el diseño del software del sistema para la Asociación Civil registrada ante el registro federal de contribuyentes como: **Amor en Ser A. C.**, de manera que en este documento se describe la estructura del software mediante la descomposición de sus módulos, procesos, datos, así como de sus interfaces. El contenido está desarrollado tomando como base el documento que establece la especificación de requisitos del software (SRS).

I.1. Propósito

Proporcionar una visión clara al desarrollador sobre los módulos del sistema, las interacciones con los usuarios, los flujos de la información así como la estructura de almacenamiento a fin de generar un proceso de desarrollo ágil.

I.2. Definiciones y acrónimos

- Diagrama caso de uso: es una representación gráfica del contexto del sistema, muestra los límites del sistema, lo que permanece fuera de él y cómo se utiliza.
- Diagrama entidad relación: es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.
- Diagrama de flujo de datos: es una representación gráfica del flujo de datos a través de un sistema de información.

II. Descomposición de la información

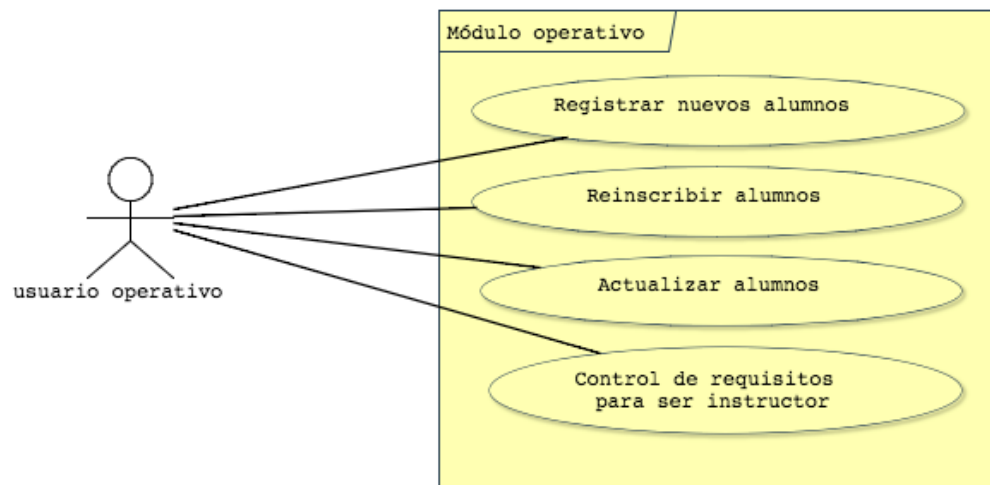
II.1. Descomposición modular

El sistema esta constituido por dos módulos: operativo y de análisis. Estos módulos en conjunto cubren la operación diaria de la organización y automatizan el proceso de toma de decisiones. Cada módulo tiene un tipo de usuario diferente:

- Operativo
- Administrador

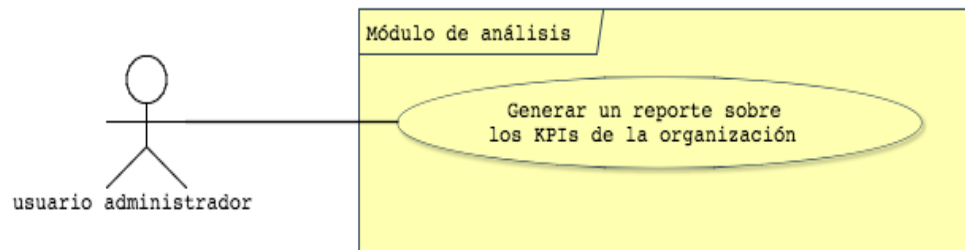
II.1.1. Módulo Operativo

El módulo operativo esta representado a través del siguiente diagrama de caso de uso.



II.1.2. Módulo de Análisis

El módulo de análisis esta representado a través del siguiente diagrama de caso de uso.

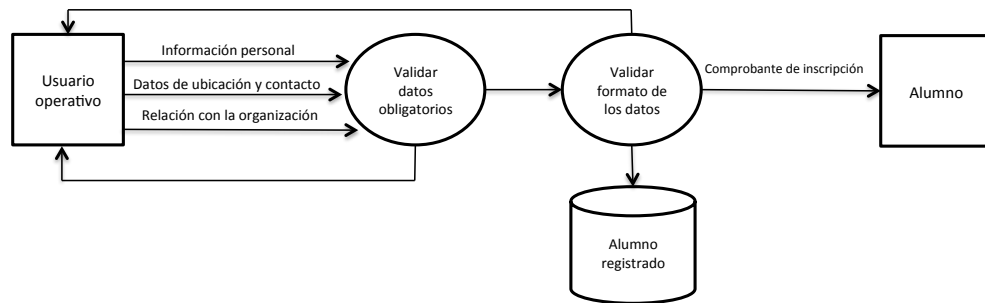


La interacción de cada uno de estos usuarios con el sistema está especificada a través de diagramas de flujo de datos que se encuentran en la descomposición de los procesos de negocio.

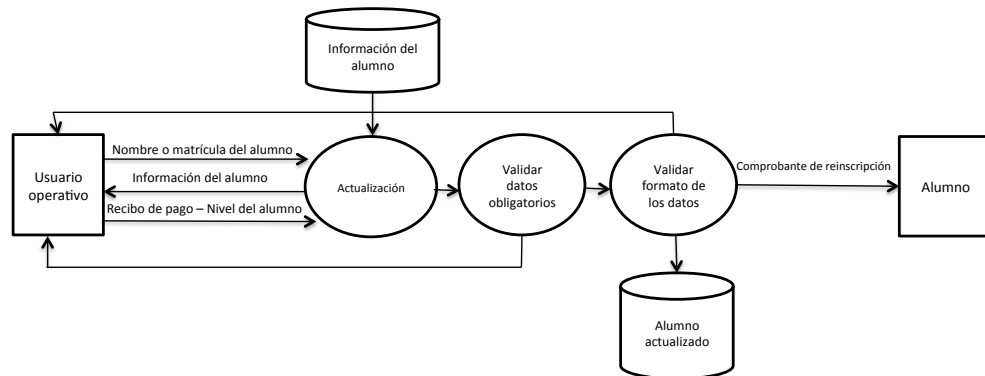
III. Descomposición de los procesos de negocio

Módulo operativo:

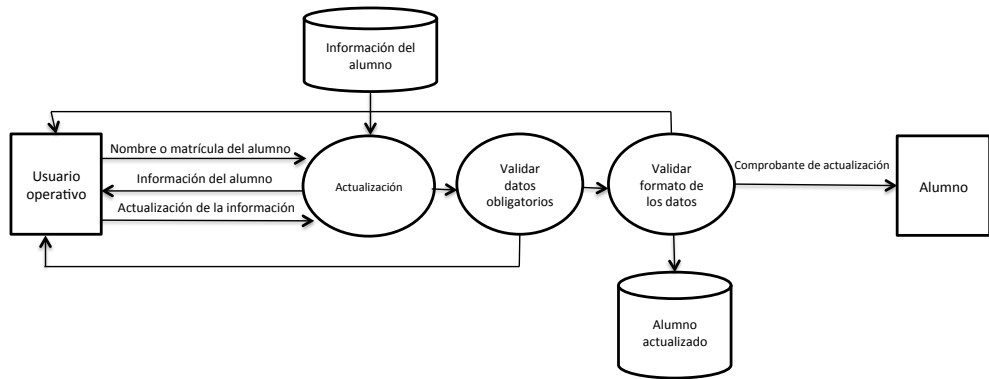
a. Registrar nuevos alumnos



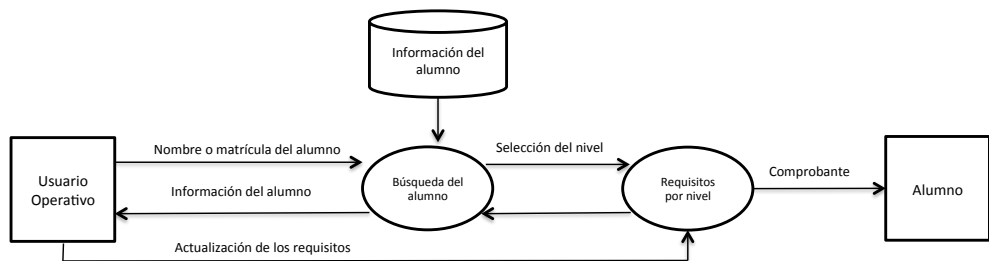
b. Reinscribir alumnos



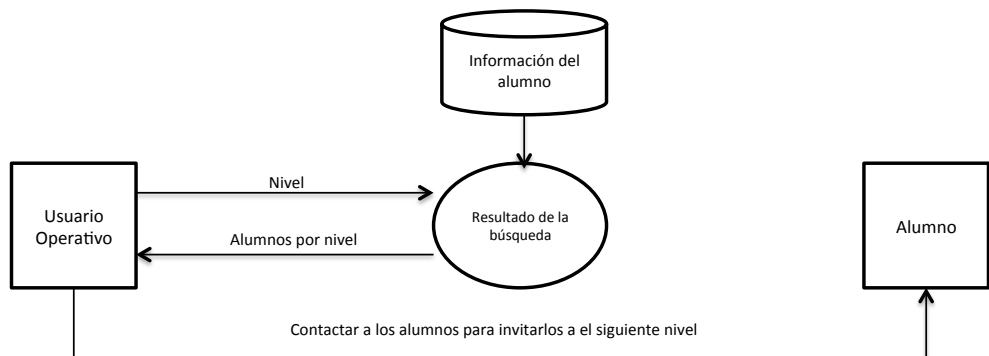
c. Actualización de datos



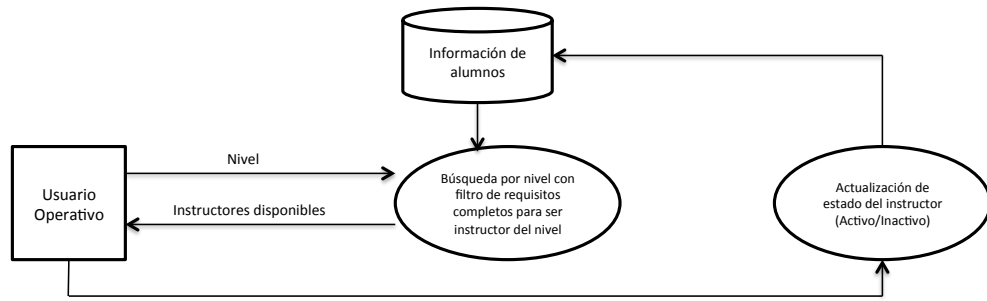
d. Control de instructores



e. Invitación a un curso

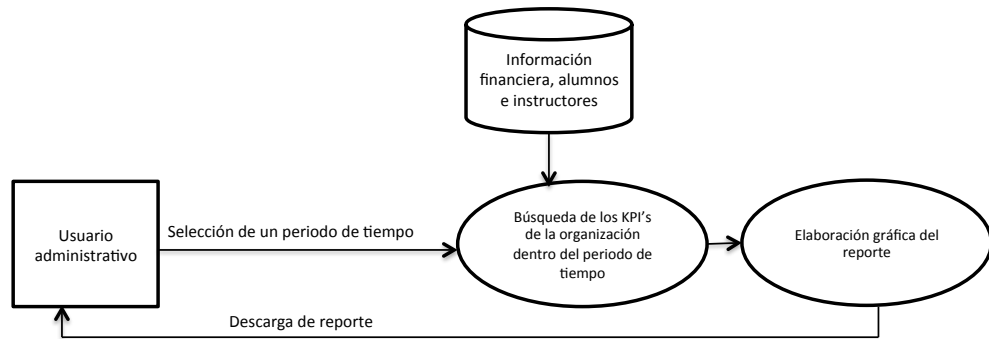


f. Disponibilidad de instructores



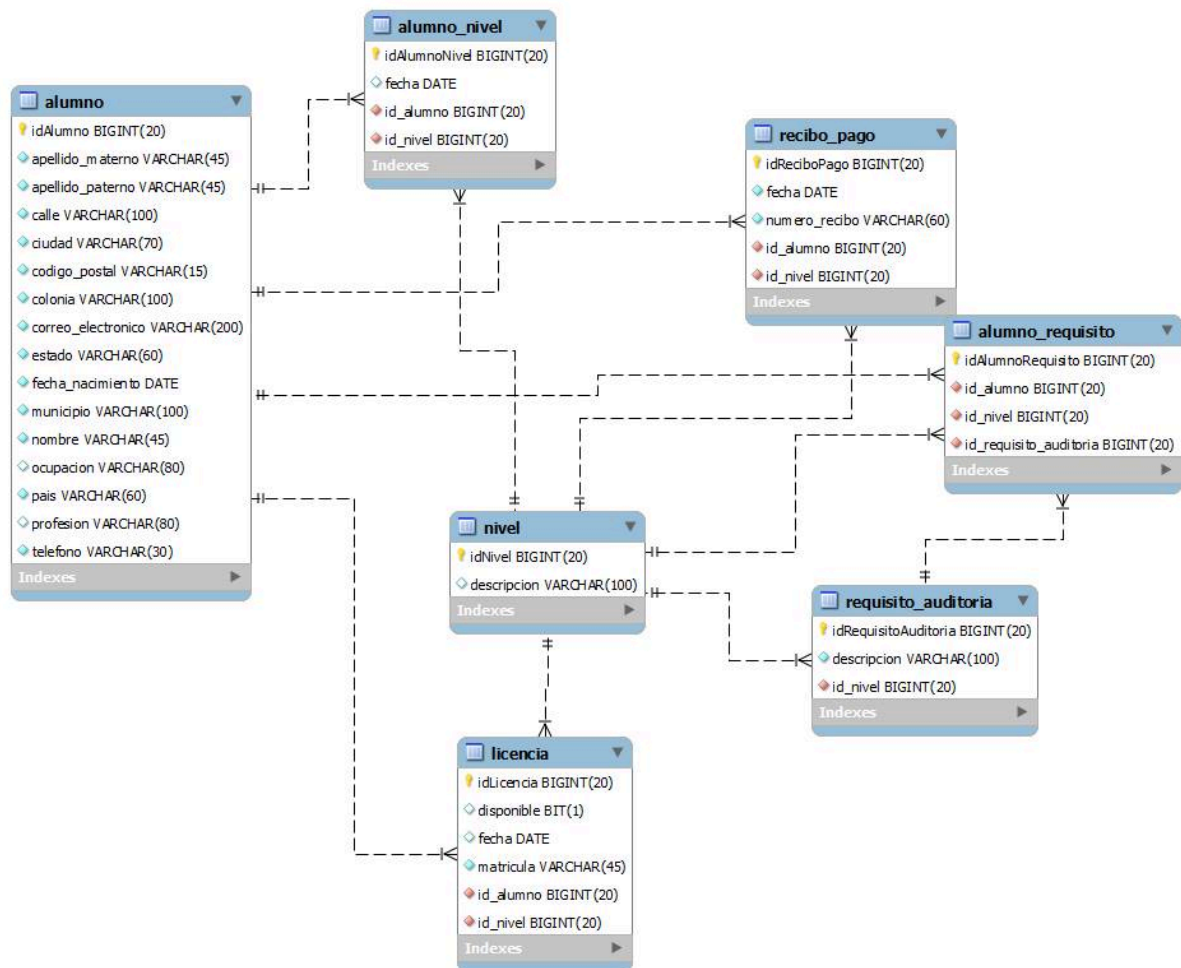
Módulo de análisis

a. Elaboración de reporte sobre los principales KPIs de la organización



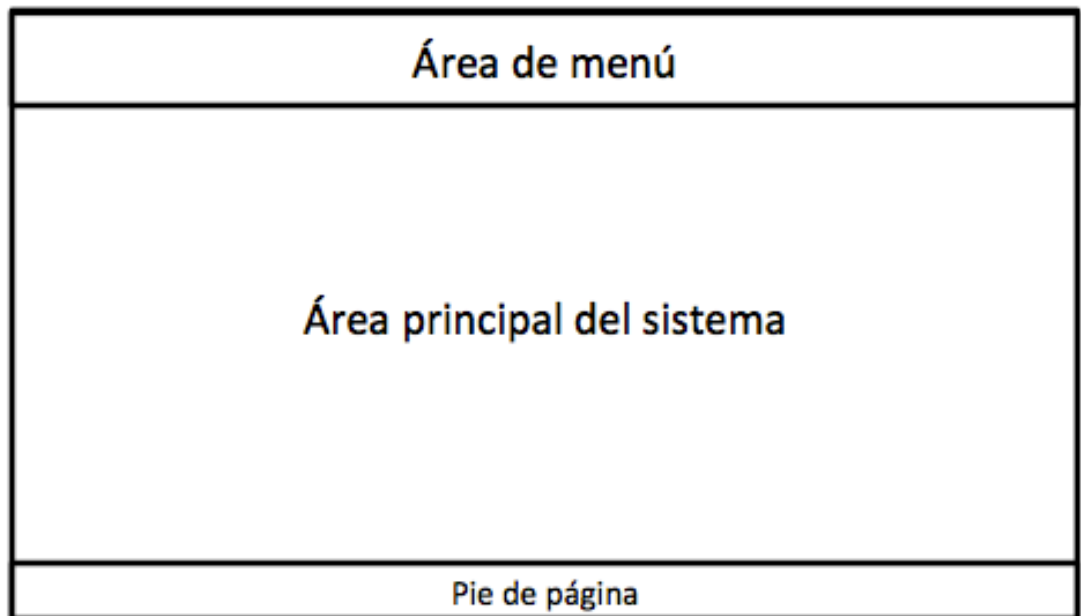
IV. Descomposición de los datos

La siguiente estructura de almacenamiento de la información (relacional) aporta valor para la consulta y procesamiento de los datos determinados en los diagrama de flujo anteriores.



V. Descripción de las interfaces

Todas las páginas del sistema seguirán el patrón gráfico mostrado a continuación:



12.5 Declaración de trabajo (SOW, Software Statement of Work)

I. Antecedentes e Introducción

La Asociación Civil registrada ante el registro federal de contribuyentes como **Amor en Ser A. C.** es una organización que presta servicios de tipo asistencial y de beneficencia a la sociedad mexicana a través de diversos centros de capacitación dentro de la República Mexicana, dicha institución se encuentra en un proceso de centralización de su administración lo que le ha generado una serie de conflictos de administración de la información tales como:

- Excesiva redundancia de la información que impacta negativamente en el desempeño del sistema actual.
- Deficiente integración de la información de la organización que dificulta la labor del personal operativo.
- Falta de control en la actualización de los datos que amenaza la integridad de la información.
- Dificultad para manipular la información del sistema lo que dificulta la ejecución de las diversas tareas del negocio.
- Realización manual de reportes que generan conocimiento para la toma de decisiones estratégicas del negocio.

Se propone diseñar una herramienta de software que integre los siguientes tres componentes de un sistema de inteligencia de negocios:

- Sistema operacional
- Depósito de datos
- Generador de reportes

El desarrollo deberá adaptarse a los recursos físicos (infraestructura de la organización) de la organización y utilizar herramientas de tipo open source para la construcción y administración del sistema.

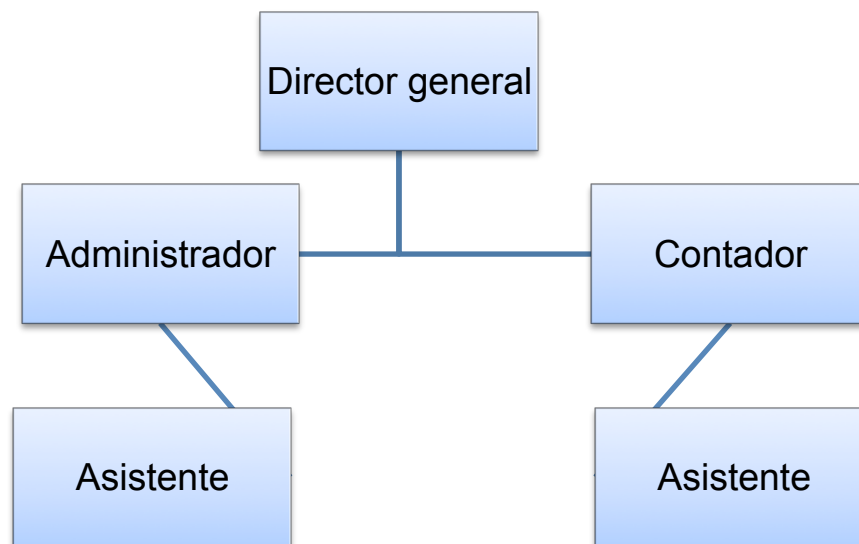
a. Objetivos del negocio y del proyecto

La empresa tiene una gran cantidad de información histórica la cual esta estructurada a través de diversas fuentes de datos, principalmente sistemas de archivos los cuales dificultan las tareas de análisis de la información para la toma de decisiones del negocio. La información operativa de la empresa continúa creciendo y genera diariamente nuevos registros, los cuales impactan negativamente en el desempeño del sistema actual ya que la integración de la información es deficiente lo que dificulta la realización de las actividades del personal de la organización.

El objetivo del proyecto es simplificar la administración de la información operativa y automatizar el análisis de la información histórica de la organización, a través del desarrollo de una herramienta de software que contemple los componentes antes mencionados de un sistema de inteligencia de negocios con la finalidad de garantizar un mejor desempeño del sistema ya que el almacenamiento y procesamiento de la información utilizará técnicas y herramientas especializadas para cada componente.

b. Organigrama del cliente

El siguiente organigrama representa la estructura organizacional del personal de la organización involucrado en el proyecto.



II. Alcance del proyecto

El sistema utiliza una arquitectura cliente servidor de tres capas: presentación, aplicación y datos. La arquitectura del sistema está basada en el desarrollo de aplicaciones web con el servidor aplicativo de java (J2EE) y el framework de desarrollo AppFuse.

A continuación se listan las actividades por arquitectura de capas para la construcción del sistema:

- I. Construcción de la capa de datos (TOTAL 34 horas)**
 - a. Diseño del esquema de la base de datos (20 horas)
 - b. Instalación de MySQL en Windows Server(2 horas)
 - c. Creación del script de base de datos (12 horas)
- II. Construcción de la capa applicativa (TOTAL 34 horas)**

- a. Instalación del IDE de desarrollo (2 horas)
- b. Instalación del framework de desarrollo AppFuse (2 horas)
- c. Programación de las reglas de negocio (30 horas)

III. Construcción de la capa de presentación (TOTAL 12 horas)

- a. Construcción de la interfaz gráfica (12 horas)

a. Requerimientos funcionales

Módulo operativo:

- El sistema debe controlar el acceso para los usuarios operativos.
- El sistema debe de considerar dos roles: instructor y alumno.
- El sistema debe de permitir al recepcionista el registro de nuevos alumnos, ingresando la siguiente información: información personal, ubicación, contacto, datos de la organización.
- El sistema debe permitir al recepcionista inscribir a los alumnos al siguiente nivel inmediato del plan de formación de la organización, registrando su recibo de pago correspondiente.
- El sistema debe de permitir al recepcionista la actualización de la información del alumno.
- El sistema debe permitir al recepcionista llevar un control de los requisitos para que un alumno pueda ser instructor de un nivel.
- El sistema debe de generar automáticamente una matrícula para cada nuevo alumno.
- El sistema debe de permitir al administrador buscar a los alumnos por nombre o matrícula.

- El sistema debe permitir al recepcionista almacenar el número de licencia y fecha de expedición para cada nuevo instructor.

Módulo de análisis:

- El sistema debe de generar reportes de manera automatizada con los siguientes indicadores de desempeño de la organización:
 - Número de cursos
 - Utilidad por curso
 - Número de alumnos por curso
 - Ingresos / Egresos
 - Desglose de Ingresos
- El sistema debe de permitir seleccionar diferentes períodos de tiempo para facilitar el análisis del desempeño.

b. Requisitos adicionales del proyecto

Requerimientos no funcionales – módulo operativo

- El sistema será desarrollado con una interfaz gráfica basada en un entorno web.
- El sistema estará disponible las 24 horas a través de la red local.
- El sistema será accesible desde cualquier equipo de trabajo conectado a la red local.
- El sistema se ejecutará sobre un servidor de aplicaciones web con sistema operativo Windows server 2008 en adelante.

- El sistema trabajará con un manejador de base de datos relacional para entornos web MySQL.

Requerimientos no funcionales – módulo de análisis

- Los reportes deberán ser visualizados a través de cualquier software de hoja de cálculo.

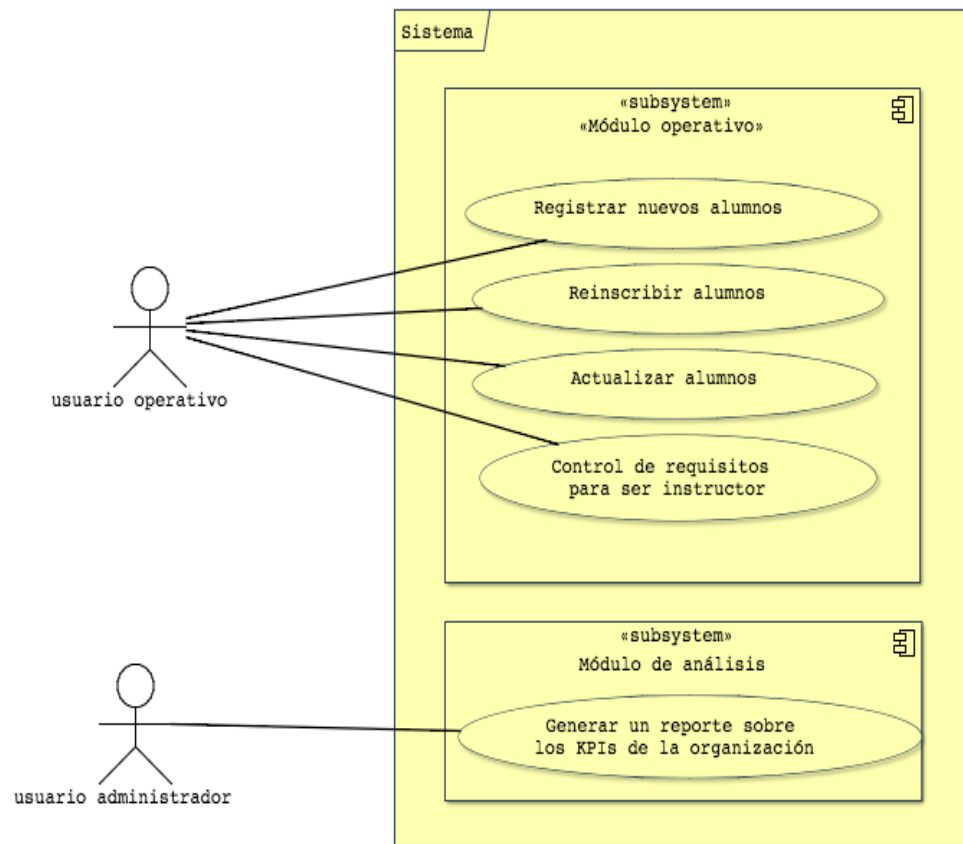
c. Requisitos de rendimiento

El sistema debe de proporcionar servicio a cada una de los equipos terminales de la organización conectados a la red local y ser capaz de atender un promedio de diez usuarios simultáneos sin impactar negativamente el desempeño del mismo.

d. Seguridad

La seguridad del sistema será administrada por los arquetipos básicos de gestión de usuarios y seguridad del framework AppFuse, los cuales pretenden simplificar y optimizar la experiencia del usuario final. El nivel de seguridad es adecuado ya que solo se podrá acceder a la aplicación a través del acceso a la red local de la organización.

e. Diagrama de caso de uso



III. Aceptación y autorización de arranque del proyecto

Nombre y firma de quien autoriza

12.6 Estimación presupuestaria

El costo aproximado del proyecto de haberse vendido o desarrollarlo fuera del campo académico esta estimado en relación a los siguientes costos que se manejan en el mercado:

- 20 USD / HRS Consultoría
- 5 USD / HRS Toma requerimientos

Actividades	Número de horas
1. Investigación preliminar	12
2. Identificación de problemas, oportunidades y objetivos	12
3. Determinación de requerimientos	20
4. Análisis de las necesidades del sistema	20
5. Diseño del sistema	12
6. Desarrollo del sistema	70
7. Implementación del sistema	8
Total 154	

Tomando en cuenta la tabla superior de actividades el cálculo se desglosa de la siguiente manera:

Actividades (1) (2) (3) y (4) → Toma de requerimientos (64 horas)
Actividades (5) (6) (7) → Consultoría (90 horas)

Costo = (64 HRS) (5 USD/HRS) + (90 HRS) (20 USD/HRS) = 2,120 USD

12.7 Configuración del equipo donde funciona la solución

El equipo final donde funciona la aplicación tiene un sistema operativo Windows Server 2008 R2 con un procesador Intel Xeon a 3.2 GHz, memoria RAM de 4 GB y una capacidad de almacenamiento de 500 GB en disco duro.

La configuración para ejecutar la aplicación es mediante un servidor web Apache Tomcat SE versión 7.0.57 con acceso a través del puerto predeterminado HTTP (8080) y con una versión de maquina virtual de java JRE 7. La dirección IP del servidor es: 192.168.1.167.

El desarrollo del aplicativo se hizo en lenguaje Java 7 actualización 51 utilizando el framework de desarrollo AppFuse versión 3.0 mediante una interfaz de desarrollo (IDE) Eclipse Kepler y usando un base de datos MySQL Community Server versión 5.6.22.

12.8 Código Fuente

Clase Address

```
@Embeddable
@Indexed
public class Address extends BaseObject
    implements Serializable
{
    private static final long serialVersionUID = 3617859655330969141L;
    private String address;
    private String city;
    private String province;
    private String country;
    private String postalCode;

    @Column(length=150)
    @Field
    public String getAddress()
    {
        return this.address;
    }
    @Column(length=50)
    @Field
    public String getCity() {
        return this.city;
    }
    @Column(length=100)
    @Field
    public String getProvince() {
        return this.province;
    }
    @Column(length=100)
    @Field
    public String getCountry() {
        return this.country;
    }
    @Column(name="postal_code", length=15)
    @Field(analyze=Analyze.NO)
    public String getPostalCode() {
        return this.postalCode;
    }

    public void setAddress(String address) {
```

```

    this.address = address;
}

public void setCity(String city) {
    this.city = city;
}

public void setCountry(String country) {
    this.country = country;
}

public void setPostalCode(String postalCode) {
    this.postalCode = postalCode;
}

public void setProvince(String province) {
    this.province = province;
}

public boolean equals(Object o)
{
    if (this == o) {
        return true;
    }
    if (!(o instanceof Address)) {
        return false;
    }

    Address address1 = (Address)o;

    return hashCode() == address1.hashCode();
}

public int hashCode()
{
    int result = this.address != null ? this.address.hashCode() : 0;
    result = 29 * result + (this.city != null ? this.city.hashCode() : 0);
    result = 29 * result + (this.province != null ? this.province.hashCode() : 0);
    result = 29 * result + (this.country != null ? this.country.hashCode() : 0);
    result = 29 * result + (this.postalCode != null ? this.postalCode.hashCode() : 0);
    return result;
}

public String toString()
{
    return new ToStringBuilder(this, ToStringStyle.MULTI_LINE_STYLE)

```



```
.append("country", this.country)
.append("address", this.address)
.append("province", this.province)
.append("postalCode", this.postalCode)
.append("city", this.city).toString();
}
}
```

Clase AgendaCurso

```
@Entity
@Table(name="agenda_curso", catalog="gest")
@Indexed
@XmlRootElement
public class AgendaCurso extends BaseObject
    implements Serializable
{
    private Long idAgendaCurso;
    private Nivel nivel;
    private Licencia licencia;
    private Date fecha;
    private Set<GastoCurso> gastoCursos = new HashSet(0);

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdAgendaCurso() { return this.idAgendaCurso; }

    public void setIdAgendaCurso(Long idAgendaCurso)
    {
        this.idAgendaCurso = idAgendaCurso;
    }
    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_nivel", nullable=false)
    public Nivel getNivel() { return this.nivel; }

    public void setNivel(Nivel nivel)
    {
        this.nivel = nivel;
    }
    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_licencia", nullable=false)
    public Licencia getLicencia() { return this.licencia; }

    public void setLicencia(Licencia licencia)
    {
        this.licencia = licencia;
    }
    @Temporal(TemporalType.DATE)
    @Column(name="fecha", nullable=false, length=10)
    @Field
    public Date getFecha() { return this.fecha; }

    public void setFecha(Date fecha)
```

```

{
    this.fecha = fecha;
}
@OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="agendaCurso")
public Set<GastoCurso> getGastoCursos() {
    return this.gastoCursos;
}

public void setGastoCursos(Set<GastoCurso> gastoCursos) {
    this.gastoCursos = gastoCursos;
}

public boolean equals(Object o) {
    if (this == o) return true;
    if ((o == null) || (getClass() != o.getClass())) return false;

    AgendaCurso pojo = (AgendaCurso)o;

    if (this.nivel != null ? !this.nivel.equals(pojo.nivel) : pojo.nivel != null) return false;
    if (this.licencia != null ? !this.licencia.equals(pojo.licencia) : pojo.licencia != null)
return false;
    if (this.fecha != null ? !this.fecha.equals(pojo.fecha) : pojo.fecha != null) return
false;

    return true;
}

public int hashCode() {
    int result = 0;
    result = this.nivel != null ? this.nivel.hashCode() : 0;
    result = 31 * result + (this.licencia != null ? this.licencia.hashCode() : 0);
    result = 31 * result + (this.fecha != null ? this.fecha.hashCode() : 0);

    return result;
}

public String toString() {
    StringBuffer sb = new StringBuffer(getClass().getSimpleName());

    sb.append(" [");

sb.append("idAgendaCurso").append("=").append(getIdAgendaCurso()).append(",
");
    sb.append("nivel").append("=").append(getNivel()).append(", ");
    sb.append("licencia").append("=").append(getLicencia()).append(", ");

```

```
sb.append("fecha").append("=").append(getFecha()).append(", ");  
sb.append("]");  
return sb.toString();  
}  
}
```

Clase Alumno

```
@Entity
@Table(name="alumno", catalog="gest")
@Indexed
@XmlRootElement
public class Alumno extends BaseObject
    implements Serializable
{
    private Long idAlumno;
    private String apellidoMaterno;
    private String apellidoPaterno;
    private String calle;
    private String ciudad;
    private String codigoPostal;
    private String colonia;
    private String correoElectronico;
    private String estado;
    private Date fechaNacimiento;
    private String municipio;
    private String nombre;
    private String ocupacion;
    private String pais;
    private String profesion;
    private String telefono;
    private Set<AlumnoRequisito> alumnoRequisitos = new HashSet(0);
    private Set<AlumnoNivel> alumnoNiveles = new HashSet(0);
    private Set<ReciboPago> reciboPagos = new HashSet(0);
    private Set<Licencia> licencias = new HashSet(0);

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdAlumno() { return this.idAlumno; }

    public void setIdAlumno(Long idAlumno)
    {
        this.idAlumno = idAlumno;
    }
    @Column(name="apellido_materno", nullable=false, length=45)
    @Field
    public String getApellidoMaterno() {
        return this.apellidoMaterno;
    }
}
```

```

public void setApellidoMaterno(String apellidoMaterno) {
    this.apellidoMaterno = apellidoMaterno;
}
@Column(name="apellido_paterno", nullable=false, length=45)
@Field
public String getApellidoPaterno() {
    return this.apellidoPaterno;
}

```

```

public void setApellidoPaterno(String apellidoPaterno) {
    this.apellidoPaterno = apellidoPaterno;
}
@Column(name="calle", nullable=false, length=100)
@Field
public String getCalle() {
    return this.calle;
}

```

```

public void setCalle(String calle) {
    this.calle = calle;
}
@Column(name="ciudad", nullable=false, length=70)
@Field
public String getCiudad() {
    return this.ciudad;
}

```

```

public void setCiudad(String ciudad) {
    this.ciudad = ciudad;
}
@Column(name="codigo_postal", nullable=false, length=15)
@Field
public String getCodigoPostal() {
    return this.codigoPostal;
}

```

```

public void setCodigoPostal(String codigoPostal) {
    this.codigoPostal = codigoPostal;
}
@Column(name="colonia", nullable=false, length=100)
@Field
public String getColonia() {
    return this.colonia;
}

```

```

public void setColonia(String colonia) {

```

```

        this.colonia = colonia;
    }
    @Column(name="correo_electronico", nullable=false, length=200)
    @Field
    public String getCorreoElectronico() {
        return this.correoElectronico;
    }

    public void setCorreoElectronico(String correoElectronico) {
        this.correoElectronico = correoElectronico;
    }
    @Column(name="estado", nullable=false, length=60)
    @Field
    public String getEstado() {
        return this.estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }
    @Temporal(TemporalType.DATE)
    @Column(name="fecha_nacimiento", nullable=false, length=10)
    @Field
    public Date getFechaNacimiento() { return this.fechaNacimiento; }

    public void setFechaNacimiento(Date fechaNacimiento)
    {
        this.fechaNacimiento = fechaNacimiento;
    }
    @Column(name="municipio", nullable=false, length=100)
    @Field
    public String getMunicipio() {
        return this.municipio;
    }

    public void setMunicipio(String municipio) {
        this.municipio = municipio;
    }
    @Column(name="nombre", nullable=false, length=45)
    @Field
    public String getNombre() {
        return this.nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

```

```

@Column(name="ocupacion", length=80)
@Field
public String getOcupacion() {
    return this.ocupacion;
}

public void setOcupacion(String ocupacion) {
    this.ocupacion = ocupacion;
}
@Column(name="pais", nullable=false, length=60)
@Field
public String getPais() {
    return this.pais;
}

public void setPais(String pais) {
    this.pais = pais;
}
@Column(name="profesion", length=80)
@Field
public String getProfesion() {
    return this.profesion;
}

public void setProfesion(String profesion) {
    this.profesion = profesion;
}
@Column(name="telefono", nullable=false, length=30)
@Field
public String getTelefono() {
    return this.telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}
@OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="alumno")
public Set<AlumnoRequisito> getAlumnoRequisitos() {
    return this.alumnoRequisitos;
}

public void setAlumnoRequisitos(Set<AlumnoRequisito> alumnoRequisitos) {
    this.alumnoRequisitos = alumnoRequisitos;
}

```



```

    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="alumno")
    public Set<AlumnoNivel> getAlumnoNivels() {
        return this.alumnoNivels;
    }

    public void setAlumnoNivels(Set<AlumnoNivel> alumnoNivels) {
        this.alumnoNivels = alumnoNivels;
    }
    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="alumno")
    public Set<ReciboPago> getReciboPagos() {
        return this.reciboPagos;
    }

    public void setReciboPagos(Set<ReciboPago> reciboPagos) {
        this.reciboPagos = reciboPagos;
    }
    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="alumno")
    public Set<Licencia> getLicencias() {
        return this.licencias;
    }

    public void setLicencias(Set<Licencia> licencias) {
        this.licencias = licencias;
    }

    public boolean equals(Object o) {
        if (this == o) return true;
        if ((o == null) || (getClass() != o.getClass())) return false;

        Alumno pojo = (Alumno)o;

        if (this.apellidoMaterno != null ?
!this.apellidoMaterno.equals(pojo.apellidoMaterno) : pojo.apellidoMaterno != null)
return false;
        if (this.apellidoPaterno != null ? !this.apellidoPaterno.equals(pojo.apellidoPaterno)
: pojo.apellidoPaterno != null) return false;
        if (this.calle != null ? !this.calle.equals(pojo.calle) : pojo.calle != null) return false;
        if (this.ciudad != null ? !this.ciudad.equals(pojo.ciudad) : pojo.ciudad != null) return
false;
        if (this.codigoPostal != null ? !this.codigoPostal.equals(pojo.codigoPostal) :
pojo.codigoPostal != null) return false;
        if (this.colonia != null ? !this.colonia.equals(pojo.colonia) : pojo.colonia != null)
return false;

```

```

        if (this.correoElectronico != null ?
!this.correoElectronico.equals(pojo.correoElectronico) : pojo.correoElectronico !=
null) return false;
        if (this.estado != null ? !this.estado.equals(pojo.estado) : pojo.estado != null)
return false;
        if (this.fechaNacimiento != null ?
!this.fechaNacimiento.equals(pojo.fechaNacimiento) : pojo.fechaNacimiento != null)
return false;
        if (this.municipio != null ? !this.municipio.equals(pojo.municipio) : pojo.municipio
!= null) return false;
        if (this.nombre != null ? !this.nombre.equals(pojo.nombre) : pojo.nombre != null)
return false;
        if (this.ocupacion != null ? !this.ocupacion.equals(pojo.ocupacion) :
pojo.ocupacion != null) return false;
        if (this.pais != null ? !this.pais.equals(pojo.pais) : pojo.pais != null) return false;
        if (this.profesion != null ? !this.profesion.equals(pojo.profesion) : pojo.profesion !=
null) return false;
        if (this.telefono != null ? !this.telefono.equals(pojo.telefono) : pojo.telefono != null)
return false;

        return true;
    }

```

```

public int hashCode() {
    int result = 0;
    result = this.apellidoMaterno != null ? this.apellidoMaterno.hashCode() : 0;
    result = 31 * result + (this.apellidoPaterno != null ?
this.apellidoPaterno.hashCode() : 0);
    result = 31 * result + (this.calle != null ? this.calle.hashCode() : 0);
    result = 31 * result + (this.ciudad != null ? this.ciudad.hashCode() : 0);
    result = 31 * result + (this.codigoPostal != null ? this.codigoPostal.hashCode() :
0);
    result = 31 * result + (this.colonia != null ? this.colonia.hashCode() : 0);
    result = 31 * result + (this.correoElectronico != null ?
this.correoElectronico.hashCode() : 0);
    result = 31 * result + (this.estado != null ? this.estado.hashCode() : 0);
    result = 31 * result + (this.fechaNacimiento != null ?
this.fechaNacimiento.hashCode() : 0);
    result = 31 * result + (this.municipio != null ? this.municipio.hashCode() : 0);
    result = 31 * result + (this.nombre != null ? this.nombre.hashCode() : 0);
    result = 31 * result + (this.ocupacion != null ? this.ocupacion.hashCode() : 0);
    result = 31 * result + (this.pais != null ? this.pais.hashCode() : 0);
    result = 31 * result + (this.profesion != null ? this.profesion.hashCode() : 0);
    result = 31 * result + (this.telefono != null ? this.telefono.hashCode() : 0);

    return result;
}

```

```

    }

    public String toString() {
        StringBuffer sb = new StringBuffer(getClass().getSimpleName());

        sb.append(" [");
        sb.append("idAlumno").append("=").append(getIdAlumno()).append(", ");

        sb.append("apellidoMaterno").append("=").append(getApellidoMaterno()).append(", ");

        sb.append("apellidoPaterno").append("=").append(getApellidoPaterno()).append(", ");

        sb.append("calle").append("=").append(getCalle()).append(", ");
        sb.append("ciudad").append("=").append(getCiudad()).append(", ");
        sb.append("codigoPostal").append("=").append(getCodigoPostal()).append(", ");
        sb.append("colonia").append("=").append(getColonia()).append(", ");

        sb.append("correoElectronico").append("=").append(getCorreoElectronico()).append(", ");
        sb.append("estado").append("=").append(getEstado()).append(", ");

        sb.append("fechaNacimiento").append("=").append(getFechaNacimiento()).append(", ");
        sb.append("municipio").append("=").append(getMunicipio()).append(", ");
        sb.append("nombre").append("=").append(getNombre()).append(", ");
        sb.append("ocupacion").append("=").append(getOcupacion()).append(", ");
        sb.append("pais").append("=").append(getPais()).append(", ");
        sb.append("profesion").append("=").append(getProfesion()).append(", ");
        sb.append("telefono").append("=").append(getTelefono()).append(", ");

        sb.append("]");

        return sb.toString();
    }
}

```

Class AlumnoNivel

```
@Entity
@Table(name="alumno_nivel", catalog="gest")
@Indexed
@XmlRootElement
public class AlumnoNivel extends BaseObject
    implements Serializable
{
    private Long idAlumnoNivel;
    private Alumno alumno;
    private Nivel nivel;
    private Date fecha;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdAlumnoNivel()
    {
        return this.idAlumnoNivel;
    }

    public void setIdAlumnoNivel(Long idAlumnoNivel) {
        this.idAlumnoNivel = idAlumnoNivel;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_alumno", nullable=false)
    public Alumno getAlumno() { return this.alumno; }

    public void setAlumno(Alumno alumno)
    {
        this.alumno = alumno;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_nivel", nullable=false)
    public Nivel getNivel() { return this.nivel; }

    public void setNivel(Nivel nivel)
    {
        this.nivel = nivel; }
    @Temporal(TemporalType.DATE)
    @Column(name="fecha", length=10)
    @Field
    public Date getFecha() { return this.fecha; }
```

```

public void setFecha(Date fecha)
{
    this.fecha = fecha;
}

public boolean equals(Object o) {
    if (this == o) return true;
    if ((o == null) || (getClass() != o.getClass())) return false;

    AlumnoNivel pojo = (AlumnoNivel)o;

    if (this.alumno != null ? !this.alumno.equals(pojo.alumno) : pojo.alumno != null)
return false;
    if (this.nivel != null ? !this.nivel.equals(pojo.nivel) : pojo.nivel != null) return false;
    if (this.fecha != null ? !this.fecha.equals(pojo.fecha) : pojo.fecha != null) return
false;

    return true;
}

public int hashCode() {
    int result = 0;
    result = this.alumno != null ? this.alumno.hashCode() : 0;
    result = 31 * result + (this.nivel != null ? this.nivel.hashCode() : 0);
    result = 31 * result + (this.fecha != null ? this.fecha.hashCode() : 0);

    return result;
}

public String toString() {
    StringBuffer sb = new StringBuffer(getClass().getSimpleName());

    sb.append(" [");
    sb.append("idAlumnoNivel").append("=").append(getIdAlumnoNivel()).append(",
");
    sb.append("alumno").append("=").append(getAlumno()).append(", ");
    sb.append("nivel").append("=").append(getNivel()).append(", ");
    sb.append("fecha").append("=").append(getFecha()).append("");
    sb.append("]");

    return sb.toString();
}
}

```

Clase AlumnoRequisito

```
@Entity
@Table(name="alumno_requisito", catalog="gest")
@Indexed
@XmlRootElement
public class AlumnoRequisito extends BaseObject
    implements Serializable
{
    private Long idAlumnoRequisito;
    private RequisitoAuditoria requisitoAuditoria;
    private Alumno alumno;
    private Nivel nivel;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdAlumnoRequisito()
    {
        return this.idAlumnoRequisito;
    }

    public void setIdAlumnoRequisito(Long idAlumnoRequisito) {
        this.idAlumnoRequisito = idAlumnoRequisito;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_requisito_auditoria", nullable=false)
    public RequisitoAuditoria getRequisitoAuditoria() { return this.requisitoAuditoria; }

    public void setRequisitoAuditoria(RequisitoAuditoria requisitoAuditoria)
    {
        this.requisitoAuditoria = requisitoAuditoria;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_alumno", nullable=false)
    public Alumno getAlumno() { return this.alumno; }

    public void setAlumno(Alumno alumno)
    {
        this.alumno = alumno;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_nivel", nullable=false)
    public Nivel getNivel() { return this.nivel; }
```

```

public void setNivel(Nivel nivel)
{
    this.nivel = nivel;
}

public boolean equals(Object o) {
    if (this == o) return true;
    if ((o == null) || (getClass() != o.getClass())) return false;

    AlumnoRequisito pojo = (AlumnoRequisito)o;

    if (this.requisitoAuditoria != null ?
    !this.requisitoAuditoria.equals(pojo.requisitoAuditoria) : pojo.requisitoAuditoria !=
    null) return false;
    if (this.alumno != null ? !this.alumno.equals(pojo.alumno) : pojo.alumno != null)
    return false;
    if (this.nivel != null ? !this.nivel.equals(pojo.nivel) : pojo.nivel != null) return false;

    return true;
}

public int hashCode() {
    int result = 0;
    result = this.requisitoAuditoria != null ? this.requisitoAuditoria.hashCode() : 0;
    result = 31 * result + (this.alumno != null ? this.alumno.hashCode() : 0);
    result = 31 * result + (this.nivel != null ? this.nivel.hashCode() : 0);

    return result;
}

public String toString() {
    StringBuffer sb = new StringBuffer(getClass().getSimpleName());

    sb.append(" [");

    sb.append("idAlumnoRequisito").append("=").append(getIdAlumnoRequisito()).app
    end("", "");

    sb.append("requisitoAuditoria").append("=").append(getRequisitoAuditoria()).appen
    d("", "");
    sb.append("alumno").append("=").append(getAlumno()).append("", "");
    sb.append("nivel").append("=").append(getNivel()).append("");
    sb.append("]");

    return sb.toString();
}

```

Clase GastoCurso

```
@Entity
@Table(name="gasto_curso", catalog="gest")
@Indexed
@XmlRootElement
public class GastoCurso extends BaseObject
    implements Serializable
{
    private Long idGastoCurso;
    private Licencia licencia;
    private AgendaCurso agendaCurso;
    private Float costo;
    private String descripcion;
    private Date fecha;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdGastoCurso()
    {
        return this.idGastoCurso;
    }

    public void setIdGastoCurso(Long idGastoCurso) {
        this.idGastoCurso = idGastoCurso;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_licencia", nullable=false)
    public Licencia getLicencia() { return this.licencia; }

    public void setLicencia(Licencia licencia)
    {
        this.licencia = licencia;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_agenda_curso", nullable=false)
    public AgendaCurso getAgendaCurso() { return this.agendaCurso; }

    public void setAgendaCurso(AgendaCurso agendaCurso)
    {
        this.agendaCurso = agendaCurso;
    }

    @Column(name="costo", nullable=false, precision=12, scale=0)
    @Field
```



```

public Float getCosto() {
    return this.costo;
}

public void setCosto(Float costo) {
    this.costo = costo;
}
@Column(name="descripcion", nullable=false, length=300)
@Field
public String getDescripcion() {
    return this.descripcion;
}

public void setDescripcion(String descripcion) {
    this.descripcion = descripcion; }
@Temporal(TemporalType.DATE)
@Column(name="fecha", length=10)
@Field
public Date getFecha() { return this.fecha; }

public void setFecha(Date fecha)
{
    this.fecha = fecha;
}

public boolean equals(Object o) {
    if (this == o) return true;
    if ((o == null) || (getClass() != o.getClass())) return false;

    GastoCurso pojo = (GastoCurso)o;

    if (this.licencia != null ? !this.licencia.equals(pojo.licencia) : pojo.licencia != null)
return false;
    if (this.agendaCurso != null ? !this.agendaCurso.equals(pojo.agendaCurso) :
pojo.agendaCurso != null) return false;
    if (this.costo != null ? !this.costo.equals(pojo.costo) : pojo.costo != null) return
false;
    if (this.descripcion != null ? !this.descripcion.equals(pojo.descripcion) :
pojo.descripcion != null) return false;
    if (this.fecha != null ? !this.fecha.equals(pojo.fecha) : pojo.fecha != null) return
false;

    return true;
}

public int hashCode() {

```

```

    int result = 0;
    result = this.licencia != null ? this.licencia.hashCode() : 0;
    result = 31 * result + (this.agendaCurso != null ? this.agendaCurso.hashCode() :
0);
    result = 31 * result + (this.costo != null ? this.costo.hashCode() : 0);
    result = 31 * result + (this.descripcion != null ? this.descripcion.hashCode() : 0);
    result = 31 * result + (this.fecha != null ? this.fecha.hashCode() : 0);

    return result;
}

public String toString() {
    StringBuffer sb = new StringBuffer(getClass().getSimpleName());

    sb.append(" [");
    sb.append("idGastoCurso").append("=").append(getIdGastoCurso()).append(",
");
    sb.append("licencia").append("=").append(getLicencia()).append(", ");
    sb.append("agendaCurso").append("=").append(getAgendaCurso()).append(",
");
    sb.append("costo").append("=").append(getCosto()).append(", ");
    sb.append("descripcion").append("=").append(getDescripcion()).append(", ");
    sb.append("fecha").append("=").append(getFecha()).append("");
    sb.append("]");

    return sb.toString();
}
}

```

Clase Licencia

```
@Entity
@Table(name="licencia", catalog="gest")
@Indexed
@XmlRootElement
public class Licencia extends BaseObject
    implements Serializable
{
    private Long idLicencia;
    private Alumno alumno;
    private Nivel nivel;
    private Boolean disponible;
    private Date fecha;
    private String matricula;
    private Set<GastoCurso> gastoCursos = new HashSet(0);
    private Set<AgendaCurso> agendaCursos = new HashSet(0);

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdLicencia() { return this.idLicencia; }

    public void setIdLicencia(Long idLicencia)
    {
        this.idLicencia = idLicencia;
    }
    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_alumno", nullable=false)
    public Alumno getAlumno() { return this.alumno; }

    public void setAlumno(Alumno alumno)
    {
        this.alumno = alumno;
    }
    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_nivel", nullable=false)
    public Nivel getNivel() { return this.nivel; }

    public void setNivel(Nivel nivel)
    {
        this.nivel = nivel;
    }
    @Column(name="disponible")
    @Field
```

```

public Boolean getDisponible() {
    return this.disponible;
}

public void setDisponible(Boolean disponible) {
    this.disponible = disponible; }
@Temporal(TemporalType.DATE)
@Column(name="fecha", length=10)
@Field
public Date getFecha() { return this.fecha; }

public void setFecha(Date fecha)
{
    this.fecha = fecha;
}
@Column(name="matricula", nullable=false, length=45)
@Field
public String getMatricula() {
    return this.matricula;
}

public void setMatricula(String matricula) {
    this.matricula = matricula;
}
@OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="licencia")
public Set<GastoCurso> getGastoCursos() {
    return this.gastoCursos;
}

public void setGastoCursos(Set<GastoCurso> gastoCursos) {
    this.gastoCursos = gastoCursos;
}
@OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="licencia")
public Set<AgendaCurso> getAgendaCursos() {
    return this.agendaCursos;
}

public void setAgendaCursos(Set<AgendaCurso> agendaCursos) {
    this.agendaCursos = agendaCursos;
}

public boolean equals(Object o) {
    if (this == o) return true;
    if ((o == null) || (getClass() != o.getClass())) return false;

```

```

Licencia pojo = (Licencia)o;

if (this.alumno != null ? !this.alumno.equals(pojo.alumno) : pojo.alumno != null)
return false;
if (this.nivel != null ? !this.nivel.equals(pojo.nivel) : pojo.nivel != null) return false;
if (this.disponible != null ? !this.disponible.equals(pojo.disponible) :
pojo.disponible != null) return false;
if (this.fecha != null ? !this.fecha.equals(pojo.fecha) : pojo.fecha != null) return
false;
if (this.matricula != null ? !this.matricula.equals(pojo.matricula) : pojo.matricula !=
null) return false;

return true;
}

public int hashCode() {
int result = 0;
result = this.alumno != null ? this.alumno.hashCode() : 0;
result = 31 * result + (this.nivel != null ? this.nivel.hashCode() : 0);
result = 31 * result + (this.disponible != null ? this.disponible.hashCode() : 0);
result = 31 * result + (this.fecha != null ? this.fecha.hashCode() : 0);
result = 31 * result + (this.matricula != null ? this.matricula.hashCode() : 0);

return result;
}

public String toString() {
StringBuffer sb = new StringBuffer(getClass().getSimpleName());

sb.append("[");
sb.append("idLicencia").append("=").append(getIdLicencia()).append(", ");
sb.append("alumno").append("=").append(getAlumno()).append(", ");
sb.append("nivel").append("=").append(getNivel()).append(", ");
sb.append("disponible").append("=").append(getDisponible()).append(", ");
sb.append("fecha").append("=").append(getFecha()).append(", ");
sb.append("matricula").append("=").append(getMatricula()).append(", ");

sb.append("]");

return sb.toString();
}
}

```

Clase Nivel

```
@Entity
@Table(name="nivel", catalog="gest")
@Indexed
@XmlRootElement
public class Nivel extends BaseObject
    implements Serializable
{
    private Long idNivel;
    private Float costo;
    private String descripcion;
    private Set<AgendaCurso> agendaCursos = new HashSet(0);
    private Set<ReciboPago> reciboPagos = new HashSet(0);
    private Set<RequisitoAuditoria> requisitoAuditorias = new HashSet(0);
    private Set<Licencia> licencias = new HashSet(0);
    private Set<AlumnoRequisito> alumnoRequisitos = new HashSet(0);
    private Set<AlumnoNivel> alumnoNiveles = new HashSet(0);

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdNivel() { return this.idNivel; }

    public void setIdNivel(Long idNivel)
    {
        this.idNivel = idNivel;
    }
    @Column(name="costo", precision=12, scale=0)
    @Field
    public Float getCosto() {
        return this.costo;
    }

    public void setCosto(Float costo) {
        this.costo = costo;
    }
    @Column(name="descripcion", length=100)
    @Field
    public String getDescripcion() {
        return this.descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
}
```

```

    }
    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="nivel")
    public Set<AgendaCurso> getAgendaCursos() {
        return this.agendaCursos;
    }

    public void setAgendaCursos(Set<AgendaCurso> agendaCursos) {
        this.agendaCursos = agendaCursos;
    }
    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="nivel")
    public Set<ReciboPago> getReciboPagos() {
        return this.reciboPagos;
    }

    public void setReciboPagos(Set<ReciboPago> reciboPagos) {
        this.reciboPagos = reciboPagos;
    }
    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="nivel")
    public Set<RequisitoAuditoria> getRequisitoAuditorias() {
        return this.requisitoAuditorias;
    }

    public void setRequisitoAuditorias(Set<RequisitoAuditoria> requisitoAuditorias) {
        this.requisitoAuditorias = requisitoAuditorias;
    }
    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="nivel")
    public Set<Licencia> getLicencias() {
        return this.licencias;
    }

    public void setLicencias(Set<Licencia> licencias) {
        this.licencias = licencias;
    }
    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
fetch=FetchType.LAZY, mappedBy="nivel")
    public Set<AlumnoRequisito> getAlumnoRequisitos() {
        return this.alumnoRequisitos;
    }

    public void setAlumnoRequisitos(Set<AlumnoRequisito> alumnoRequisitos) {
        this.alumnoRequisitos = alumnoRequisitos;
    }

```

```

    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
    fetch=FetchType.LAZY, mappedBy="nivel")
    public Set<AlumnoNivel> getAlumnoNivels() {
        return this.alumnoNivels;
    }

    public void setAlumnoNivels(Set<AlumnoNivel> alumnoNivels) {
        this.alumnoNivels = alumnoNivels;
    }

    public boolean equals(Object o) {
        if (this == o) return true;
        if ((o == null) || (getClass() != o.getClass())) return false;

        Nivel pojo = (Nivel)o;

        if (this.costo != null ? !this.costo.equals(pojo.costo) : pojo.costo != null) return
false;
        if (this.descripcion != null ? !this.descripcion.equals(pojo.descripcion) :
pojo.descripcion != null) return false;

        return true;
    }

    public int hashCode() {
        int result = 0;
        result = this.costo != null ? this.costo.hashCode() : 0;
        result = 31 * result + (this.descripcion != null ? this.descripcion.hashCode() : 0);

        return result;
    }

    public String toString() {
        StringBuffer sb = new StringBuffer(getClass().getSimpleName());

        sb.append(" [");
        sb.append("idNivel").append("=").append(getIdNivel()).append(", ");
        sb.append("costo").append("=").append(getCosto()).append(", ");
        sb.append("descripcion").append("=").append(getDescripcion()).append(", ");

        sb.append("]");

        return sb.toString();
    }
}

```


Clase ReciboPago

```
@Entity
@Table(name="recibo_pago", catalog="gest")
@Indexed
@XmlRootElement
public class ReciboPago extends BaseObject
    implements Serializable
{
    private Long idReciboPago;
    private Alumno alumno;
    private Nivel nivel;
    private boolean concepto;
    private Date fecha;
    private Float importe;
    private String numeroRecibo;
    private String observaciones;
    private boolean tipoPago;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdReciboPago()
    {
        return this.idReciboPago;
    }

    public void setIdReciboPago(Long idReciboPago) {
        this.idReciboPago = idReciboPago;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_alumno", nullable=false)
    public Alumno getAlumno() { return this.alumno; }

    public void setAlumno(Alumno alumno)
    {
        this.alumno = alumno;
    }

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_nivel", nullable=false)
    public Nivel getNivel() { return this.nivel; }

    public void setNivel(Nivel nivel)
    {
        this.nivel = nivel;
    }
}
```

```

@Column(name="concepto", nullable=false)
@Field
public boolean isConcepto() {
    return this.concepto;
}

public void setConcepto(boolean concepto) {
    this.concepto = concepto; }
@Column(TemporalType.DATE)
@Column(name="fecha", nullable=false, length=10)
@Field
public Date getFecha() { return this.fecha; }

public void setFecha(Date fecha)
{
    this.fecha = fecha;
}
@Column(name="importe", nullable=false, precision=12, scale=0)
@Field
public Float getImporte() {
    return this.importe;
}

public void setImporte(Float importe) {
    this.importe = importe;
}
@Column(name="numero_recibo", nullable=false, length=60)
@Field
public String getNumeroRecibo() {
    return this.numeroRecibo;
}

public void setNumeroRecibo(String numeroRecibo) {
    this.numeroRecibo = numeroRecibo;
}
@Column(name="observaciones", length=300)
@Field
public String getObservaciones() {
    return this.observaciones;
}

public void setObservaciones(String observaciones) {
    this.observaciones = observaciones;
}
@Column(name="tipo_pago", nullable=false)
@Field

```

```

public boolean isTipoPago() {
    return this.tipoPago;
}

public void setTipoPago(boolean tipoPago) {
    this.tipoPago = tipoPago;
}

public boolean equals(Object o) {
    if (this == o) return true;
    if ((o == null) || (getClass() != o.getClass())) return false;

    ReciboPago pojo = (ReciboPago)o;

    if (this.alumno != null ? !this.alumno.equals(pojo.alumno) : pojo.alumno != null)
return false;
    if (this.nivel != null ? !this.nivel.equals(pojo.nivel) : pojo.nivel != null) return false;
    if (this.concepto != pojo.concepto) return false;
    if (this.fecha != null ? !this.fecha.equals(pojo.fecha) : pojo.fecha != null) return
false;
    if (this.importe != null ? !this.importe.equals(pojo.importe) : pojo.importe != null)
return false;
    if (this.numeroRecibo != null ? !this.numeroRecibo.equals(pojo.numeroRecibo) :
pojo.numeroRecibo != null) return false;
    if (this.observaciones != null ? !this.observaciones.equals(pojo.observaciones) :
pojo.observaciones != null) return false;
    if (this.tipoPago != pojo.tipoPago) return false;

    return true;
}

public int hashCode() {
    int result = 0;
    result = this.alumno != null ? this.alumno.hashCode() : 0;
    result = 31 * result + (this.nivel != null ? this.nivel.hashCode() : 0);
    result = 31 * result + (this.concepto ? 1 : 0);
    result = 31 * result + (this.fecha != null ? this.fecha.hashCode() : 0);
    result = 31 * result + (this.importe != null ? this.importe.hashCode() : 0);
    result = 31 * result + (this.numeroRecibo != null ? this.numeroRecibo.hashCode()
: 0);
    result = 31 * result + (this.observaciones != null ? this.observaciones.hashCode()
: 0);
    result = 31 * result + (this.tipoPago ? 1 : 0);

    return result;
}

```

```

public String toString() {
    StringBuffer sb = new StringBuffer(getClass().getSimpleName());

    sb.append(" [");
    sb.append("idReciboPago").append("=").append(getIdReciboPago()).append(",
");
    sb.append("alumno").append("=").append(getAlumno()).append(", ");
    sb.append("nivel").append("=").append(getNivel()).append(", ");
    sb.append("concepto").append("=").append(isConcepto()).append(", ");
    sb.append("fecha").append("=").append(getFecha()).append(", ");
    sb.append("importe").append("=").append(getImporte()).append(", ");
    sb.append("numeroRecibo").append("=").append(getNumeroRecibo()).append(",
");
    sb.append("observaciones").append("=").append(getObservaciones()).append(",
");
    sb.append("tipoPago").append("=").append(isTipoPago()).append("");
    sb.append("]");

    return sb.toString();
}
}

```

Clase RequisitoAuditoria

```
@Entity
@Table(name="requisito_auditoria", catalog="gest")
@Indexed
@XmlRootElement
public class RequisitoAuditoria extends BaseObject
    implements Serializable
{
    private Long idRequisitoAuditoria;
    private Nivel nivel;
    private String descripcion;
    private Set<AlumnoRequisito> alumnoRequisitos = new HashSet(0);

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @DocumentId
    public Long getIdRequisitoAuditoria() { return this.idRequisitoAuditoria; }

    public void setIdRequisitoAuditoria(Long idRequisitoAuditoria)
    {
        this.idRequisitoAuditoria = idRequisitoAuditoria;
    }
    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="id_nivel", nullable=false)
    public Nivel getNivel() { return this.nivel; }

    public void setNivel(Nivel nivel)
    {
        this.nivel = nivel;
    }
    @Column(name="descripcion", nullable=false, length=100)
    @Field
    public String getDescripcion() {
        return this.descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    @OneToMany(cascade={javax.persistence.CascadeType.ALL},
    fetch=FetchType.LAZY, mappedBy="requisitoAuditoria")
    public Set<AlumnoRequisito> getAlumnoRequisitos() {
        return this.alumnoRequisitos;
    }
}
```

```

public void setAlumnoRequisitos(Set<AlumnoRequisito> alumnoRequisitos) {
    this.alumnoRequisitos = alumnoRequisitos;
}

public boolean equals(Object o) {
    if (this == o) return true;
    if ((o == null) || (getClass() != o.getClass())) return false;

    RequisitoAuditoria pojo = (RequisitoAuditoria)o;

    if (this.nivel != null ? !this.nivel.equals(pojo.nivel) : pojo.nivel != null) return false;
    if (this.descripcion != null ? !this.descripcion.equals(pojo.descripcion) :
    pojo.descripcion != null) return false;

    return true;
}

public int hashCode() {
    int result = 0;
    result = this.nivel != null ? this.nivel.hashCode() : 0;
    result = 31 * result + (this.descripcion != null ? this.descripcion.hashCode() : 0);

    return result;
}

public String toString() {
    StringBuffer sb = new StringBuffer(getClass().getSimpleName());

    sb.append(" [");

    sb.append("idRequisitoAuditoria").append("=").append(getIdRequisitoAuditoria()).a
    ppend(", ");
    sb.append("nivel").append("=").append(getNivel()).append(", ");
    sb.append("descripcion").append("=").append(getDescripcion()).append(", ");

    sb.append("]");

    return sb.toString();
}
}

```

Class Rol

```
@Entity
@Table(name="role")
@NamedQueries({@javax.persistence.NamedQuery(name="findRoleByName",
query="select r from Role r where r.name = :name ")})
public class Role extends BaseObject
    implements Serializable, GrantedAuthority
{
    private static final long serialVersionUID = 3690197650654049848L;
    private Long id;
    private String name;
    private String description;

    public Role()
    {
    }

    public Role(String name)
    {
        this.name = name;
    }

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    public Long getId() {
        return this.id;
    }

    @Transient
    public String getAuthority()
    {
        return getName();
    }

    @Column(length=20)
    public String getName() {
        return this.name;
    }

    @Column(length=64)
    public String getDescription() {
        return this.description;
    }

    public void setId(Long id) {
        this.id = id;
    }
}
```

```

    }

    public void setName(String name) {
        this.name = name;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public boolean equals(Object o)
    {
        if (this == o) {
            return true;
        }
        if (!(o instanceof Role)) {
            return false;
        }

        Role role = (Role)o;

        return this.name != null ? this.name.equals(role.name) : role.name == null;
    }

    public int hashCode()
    {
        return this.name != null ? this.name.hashCode() : 0;
    }

    public String toString()
    {
        return new ToStringBuilder(this, ToStringStyle.SIMPLE_STYLE)
            .append(this.name)
            .toString();
    }
}

```


Clase Usuario

```
@Entity
@Table(name="app_user")
@Indexed
@XmlRootElement
public class User extends BaseObject
    implements Serializable, UserDetails
{
    private static final long serialVersionUID = 3832626162173359411L;
    private Long id;
    private String username;
    private String password;
    private String confirmPassword;
    private String passwordHint;
    private String firstName;
    private String lastName;
    private String email;
    private String phoneNumber;
    private String website;
    private Address address = new Address();
    private Integer version;
    private Set<Role> roles = new HashSet();
    private boolean enabled;
    private boolean accountExpired;
    private boolean accountLocked;
    private boolean credentialsExpired;

    public User()
    {
    }

    public User(String username)
    {
        this.username = username;
    }

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @DocumentId
    public Long getId() { return this.id; }

    @Column(nullable=false, length=50, unique=true)
    @Field
    public String getUsername() {
        return this.username;
    }
}
```

```

}
@Column(nullable=false)
@XmlTransient
@JsonIgnore
public String getPassword() { return this.password; }
@Transient
@XmlTransient
@JsonIgnore
public String getConfirmPassword() {
    return this.confirmPassword;
}
@Column(name="password_hint")
@XmlTransient
public String getPasswordHint() {
    return this.passwordHint;
}
@Column(name="first_name", nullable=false, length=50)
@Field
public String getFirstName() {
    return this.firstName;
}
@Column(name="last_name", nullable=false, length=50)
@Field
public String getLastName() {
    return this.lastName;
}
@Column(nullable=false, unique=true)
@Field
public String getEmail() {
    return this.email;
}
@Column(name="phone_number")
@Field(analyze=Analyze.NO)
public String getPhoneNumber() {
    return this.phoneNumber;
}

@Field
public String getWebsite() {
    return this.website;
}

@Transient
public String getFullName()
{
    return this.firstName + ' ' + this.lastName;
}

```

```

    }
    @Embedded
    @IndexedEmbedded
    public Address getAddress() {
        return this.address;
    }

    @ManyToMany(fetch=FetchType.EAGER)
    @Fetch(FetchMode.SELECT)
    @JoinTable(name="user_role",
joinColumns={@javax.persistence.JoinColumn(name="user_id")},
inverseJoinColumns={@javax.persistence.JoinColumn(name="role_id")})
    public Set<Role> getRoles()
    {
        return this.roles;
    }

    @Transient
    public List<LabelValue> getRoleList()
    {
        List userRoles = new ArrayList();

        if (this.roles != null) {
            for (Role role : this.roles)
            {
                userRoles.add(new LabelValue(role.getName(), role.getName()));
            }
        }

        return userRoles;
    }

    public void addRole(Role role)
    {
        getRoles().add(role);
    }

    @Transient
    public Set<GrantedAuthority> getAuthorities()
    {
        Set authorities = new LinkedHashSet();
        authorities.addAll(this.roles);
        return authorities;
    }

    @Version

```

```

public Integer getVersion() {
    return this.version;
}

@Column(name="account_enabled")
public boolean isEnabled() {
    return this.enabled;
}

@Column(name="account_expired", nullable=false)
public boolean isAccountExpired() {
    return this.accountExpired;
}

@Transient
public boolean isAccountNonExpired()
{
    return !isAccountExpired();
}

@Column(name="account_locked", nullable=false)
public boolean isAccountLocked() {
    return this.accountLocked;
}

@Transient
public boolean isAccountNonLocked()
{
    return !isAccountLocked();
}

@Column(name="credentials_expired", nullable=false)
public boolean isCredentialsExpired() {
    return this.credentialsExpired;
}

@Transient
public boolean isCredentialsNonExpired()
{
    return !this.credentialsExpired;
}

public void setId(Long id) {
    this.id = id;
}

```

```

public void setUsername(String username) {
    this.username = username;
}

public void setPassword(String password) {
    this.password = password;
}

public void setConfirmPassword(String confirmPassword) {
    this.confirmPassword = confirmPassword;
}

public void setPasswordHint(String passwordHint) {
    this.passwordHint = passwordHint;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

public void setEmail(String email) {
    this.email = email;
}

public void setPhoneNumber(String phoneNumber) {
    this.phoneNumber = phoneNumber;
}

public void setWebsite(String website) {
    this.website = website;
}

public void setAddress(Address address) {
    this.address = address;
}

public void setRoles(Set<Role> roles) {
    this.roles = roles;
}

public void setVersion(Integer version) {
    this.version = version;
}

```

```

    }

    public void setEnabled(boolean enabled) {
        this.enabled = enabled;
    }

    public void setAccountExpired(boolean accountExpired) {
        this.accountExpired = accountExpired;
    }

    public void setAccountLocked(boolean accountLocked) {
        this.accountLocked = accountLocked;
    }

    public void setCredentialsExpired(boolean credentialsExpired) {
        this.credentialsExpired = credentialsExpired;
    }

    public boolean equals(Object o)
    {
        if (this == o) {
            return true;
        }
        if (!(o instanceof User)) {
            return false;
        }

        User user = (User)o;

        return this.username != null ? this.username.equals(user.getUsername()) :
        user.getUsername() == null;
    }

    public int hashCode()
    {
        return this.username != null ? this.username.hashCode() : 0;
    }

    public String toString()
    {
        ToStringBuilder sb = new ToStringBuilder(this, ToStringStyle.DEFAULT_STYLE)
            .append("username", this.username)
            .append("enabled", this.enabled)
            .append("accountExpired", this.accountExpired)
            .append("credentialsExpired", this.credentialsExpired)
            .append("accountLocked", this.accountLocked);
    }

```

```

if (this.roles != null) {
    sb.append("Granted Authorities: ");

    int i = 0;
    for (Role role : this.roles) {
        if (i > 0) {
            sb.append(", ");
        }
        sb.append(role.toString());
        i++;
    }
} else {
    sb.append("No Granted Authorities");
}
return sb.toString();
}
}

```



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

MTRO. JOSÉ GONZALO GUERRERO ZEPEDA
DIRECTOR DE LA FACULTAD DE INGENIERÍA
DE LA U.N.A.M.
Presente.

FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
COMITÉ DE TITULACIÓN

OFICIO: FING/DIE/CSSES/094/2015

ASUNTO: Solicitud de Jurado Para Examen Profesional

Por medio del presente, el alumno **SANCHEZ PEREZ ANGEL ROBERTO** registrado en esta facultad con el número de cuenta **306225545** en la carrera de **INGENIERÍA EN COMPUTACIÓN** e inscrito en la modalidad de titulación denominada:

TITULACIÓN MEDIANTE TESIS O TESINA Y EXAMEN PROFESIONAL

quien cumpliendo los requisitos de egreso de su plan de estudios necesarios para realizar sus trámites de examen profesional, presentó como trabajo escrito la **TESIS** con el siguiente título:

AUTOMATIZACIÓN DEL SISTEMA DE MEDICIÓN DE DESEMPEÑO PARA LA TOMA DE DECISIONES ESTRATÉGICAS DE NEGOCIO: CASO DE ESTUDIO

solicita atentamente se sirva autorizar la programación de su examen profesional con el siguiente jurado, **que ha sido previamente validado como personal académico activo de la UNAM y sin goce de periodo sabático.**

ASIGNACIÓN	NOMBRE
------------	--------

PRESIDENTE:	ING. ALBERTO TEMPLOS CARBAJAL
-------------	-------------------------------

VOCAL:	DR. DANIEL TREJO MEDINA
--------	-------------------------

SECRETARIO:	ING. ORLANDO ZALDIVAR ZAMORATEGUI
-------------	-----------------------------------

1ER. SUPLENTE:	M.C. ALEJANDRO VELAZQUEZ MENA
----------------	-------------------------------

2DO. SUPLENTE:	LIC. CARLOS AYALA GOMEZ
----------------	-------------------------

Atentamente,

"POR MI RAZA HABLARÁ EL ESPÍRITU"

Cd. Universitaria, D.F. a 27 de Enero de 2015.

EL PRESIDENTE DEL COMITÉ

DR. FRANCISCO JAVIER GARCÍA UGALDE

El interesado deberá acudir dentro de los 20 días hábiles siguientes a la fecha del presente, a entregar la documentación necesaria para el examen, título y cédula, de acuerdo a los términos e indicaciones del instructivo correspondiente. De lo contrario, deberá obtener un refrendo del presente oficio por parte de este Comité de Titulación.

CCP: SANCHEZ PEREZ ANGEL ROBERTO. Interesado

FEX-3
MLDV

