



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

**SISTEMA DE APOYO A LA TOMA DE DECISIONES
PARA ELABORAR HORARIOS DE CLASES**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

ACTUARIO

P R E S E N T A:

PABLO GÓMEZ GARCÍA

**DIRECTOR DE TESIS:
M. EN I. MIGUEL EHÉCATL MORALES TRUJILLO**

MÉXICO, D.F.

2014.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno

Pablo
Gómez
García
3870 5097
Universidad Nacional Autónoma de México
Facultad de Ciencias
Actuaría
307328151

2. Datos del tutor

M. en I.
Miguel Ehécatl
Morales
Trujillo

3. Datos del sinodal 1

Dra.
Hanna Jadwiga
Oktaba

4. Datos del sinodal 2

M. en C.
María Guadalupe Elena
Ibargüengoitia
González

5. Datos del sinodal 3

M. en I.
Carlos Ricardo
Cruz
Mendoza

6. Datos del sinodal 4

M. en H.A.
Carlos Iván
Lingan
Pérez

7. Datos del trabajo escrito

Sistema de Apoyo a la Toma de Decisiones para Elaborar Horarios de Clases
100 p
2014

Agradecimientos

Agradezco a Dios por darme la oportunidad de vivir y poder haber logrado esta meta en mi vida. Agradezco también a mi familia por haberme dado todo su apoyo y comprensión, así como el sacrificio que hicieron para que yo pudiera terminar mi carrera. Quiero agradecer enormemente a mi tutor el M. en I. Miguel Ehécatl Morales Trujillo por el tiempo, la dedicación y el esfuerzo que puso en este proyecto, así como también el haberme enseñado la verdadera disciplina y el compromiso que se necesita para poder llevar a cabo una meta profesional, por su excelente calidad de trabajo y su agradable carácter personal.

Expreso mi agradecimiento a los Sinodales por aceptar ser parte del jurado calificador de este proyecto y por sus observaciones y comentarios que ayudaron a mejorar esta tesis.

Dedicatoria

Dedico este trabajo a mi madre María Antonieta García Colín, a mi abuela Isabel Colín Vega, y a mi Tía María Guadalupe García Colín.

ÍNDICE

Agradecimientos	3
Dedicatoria.....	3
ÍNDICE.....	4
CAPÍTULO 1 Introducción	6
1.1 Antecedentes.....	6
1.2 Planteamiento del Problema	7
1.3 Objetivo	7
1.4 Solución propuesta	7
CAPÍTULO 2 Business Intelligence (BI).....	9
2.1 ¿Qué son los BI?.....	9
2.2 ¿Para qué sirven los BI?	11
2.3 BI Existentes.....	11
2.4 Ventajas de los BI.....	13
2.5 Desventajas de los BI	13
CAPÍTULO 3 Sistema de Soporte de Decisiones (DSS)	14
3.1 Introducción a los DSS	14
3.2 DSS Existentes	15
3.3 Ventajas de los DSS	19
3.4 Desventajas de los DSS	20
3.5 Aplicaciones de los DSS.....	20
CAPÍTULO 4 Teoría de las Decisiones	21
4.1 Introducción a la Teoría de las Decisiones	21
4.2 Tipos de decisión	22
4.3 Toma de Decisiones.....	22
CAPÍTULO 5 Construcción de la base de datos	28
5.1 Supuestos en el diseño.....	28
5.2 Opinión de los alumnos como base de los supuestos	29
5.3 Diagrama Entidad\Relación del diseño de la base de datos	35
5.4 Diagrama de Clases UML del diseño de la base de datos	36
5.5 Entidades y atributos de la base de datos	37
5.6 Construcción de la base de datos y de los procedimientos almacenados (SP)	45
5.7 Procedimientos Almacenados.....	46
CAPÍTULO 6 Proceso de desarrollo del DSS.....	59
6.1 Justificación	59
6.2 Planificación	60

6.3 Análisis del Sistema	62
6.4 Diseño.....	64
6.6 Construcción.....	67
6.7 Implementación	71
CONCLUSIONES.....	73
Trabajo Futuro	73
Referencias	74
Bibliografía.....	76
Anexo A.....	77
Anexo B.....	83

CAPÍTULO 1 Introducción

El presente trabajo exhibe el desarrollo de un sistema de apoyo a la toma de decisiones para elaborar horarios de clases a partir de evaluaciones de profesores. La necesidad de crear un sistema de esta índole surge principalmente de la problemática que se presenta cada semestre dentro de la Facultad de Ciencias de la UNAM, cuando un alumno pretende elaborar su horario de clases. Dicha problemática consiste en que hasta la fecha no existe información sólida que permita a los alumnos tomar decisiones fundamentadas con respecto a los profesores que imparten cursos en la facultad, pues la mayoría de los alumnos deciden inscribirse con un profesor sin tener una idea clara de su curso, su manera de trabajar o su evaluación entre muchas otras cosas. De esta manera, semestre tras semestre, entre las causas de abandono de cursos está la que el profesor no llenó las expectativas del alumno, dándose cuenta demasiado tarde.

La idea de este proyecto es proveer a los alumnos de la Facultad de Ciencias, o de cualquier otra institución de educación superior, una herramienta que proporcione información sólida, que permita tomar buenas decisiones cuando se trate de elegir a un profesor para tomar clase con él. Cabe reconocer que el sistema no va a terminar con el problema de la deserción, sin embargo, pretende ser una alternativa para disminuirla a causa de una mala decisión por no contar con la información adecuada.

Esta tesis muestra paso a paso cómo se formó dicho sistema, así como los supuestos en los que está cimentado para así poder ofrecer una solución a la problemática antes planteada. En sí, la funcionalidad principal del sistema es la de realizar rankings de los profesores que mejor se adapten a una serie de criterios que el alumno proporciona al sistema, para que así, el alumno obtenga información que lo apoye a tomar una decisión. Adicionalmente permite ver los comentarios hechos acerca de un profesor sólo por los que han sido sus alumnos y no por otras personas ajenas a él.

1.1 Antecedentes

Actualmente existen sistemas de negocios inteligentes (BI)¹ que se encargan de la elaboración de rankings y reportes a partir de cierta información con el fin de facilitar la toma de decisión de los usuarios. Sin embargo, la mayoría son costosos y con un alcance demasiado extenso para ser utilizados en este proyecto. Es por ello que se pretende manipular la información con otros métodos.

Para este trabajo se han utilizado los conocimientos obtenidos de Bases de Datos y las herramientas que permiten administrarlas. Los Sistemas Manejadores de Bases de Datos (SMBD)² al ser software especializado para este fin, permiten definir funciones y operaciones particulares sobre los datos, haciendo uso de procedimientos almacenados (SP)³. Estos SP permiten operar con los datos de manera eficiente y a partir de ellos,

¹ BI por sus siglas en inglés Business Intelligence. Estos son un conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

² Un SMBD es un software que se usa especialmente para el manejo de bases de datos.

³ SP por sus siglas en inglés Stored Procedure. Es un programa el cual se almacena físicamente en una base de datos.

con ayuda de la Teoría de las Decisiones⁴, obtener información. Lo que se pretende con esto es asemejarse a un sistema de soporte de decisiones (DSS)⁵ que logre suministrar la información apropiada al usuario que la solicita para permitirle tomar decisiones mejor fundamentadas.

Por otro lado también existen páginas de internet que proporcionan información sobre los profesores que imparten clases en ciertas entidades educativas de nivel superior, incluyendo a la Facultad de Ciencias de la UNAM, pero mucha de esta información carece de respaldos de autoridad y no resulta ser lo suficientemente sólida como para ser tomada en cuenta.

1.2 Planteamiento del Problema

El problema a resolver en esta tesis consiste en dar solución a un caso particular de negocio, más adelante se verá detalladamente qué es y en qué consiste el caso de negocio, enfocado a la toma de decisiones de los alumnos de las entidades de educación superior para elaborar sus horarios de clases, mediante la recolección de información y la buena administración de la misma.

1.3 Objetivo

El objetivo de esta tesis es proveer al alumno de un sistema de apoyo a la toma de decisiones que le permita elaborar su horario de clases a partir de evaluaciones docentes, dicho sistema será llamado SATDEH. Las tomas de decisiones deben adecuarse a los requerimientos y necesidades del alumno, buscando así una administración y análisis adecuados de los datos registrados en dicho sistema.

1.4 Solución propuesta

Para lograr el propósito de esta tesis es necesario contar con un módulo que recolecte las opiniones de los alumnos acerca de las características que buscan en un profesor. Los criterios deben ser obtenidos a partir de consultas hechas a alumnos de nivel licenciatura, principalmente de la Facultad de Ciencias y retomados de sistemas con objetivos similares.

Una vez que se cuente con la información que indique qué busca un alumno de un profesor, se procederá a la elaboración del diseño de la base de datos que dé solución a las necesidades antes expuestas. La base de datos será la que almacene los datos relativos a los profesores pertenecientes a una entidad educativa. Además de la información sobre los cursos que son impartidos por dichos profesores.

Un siguiente módulo será el encargado de generar un ranking específico sobre los profesores. Este ranking se construirá a partir de criterios proporcionados por el alumno que describan sus intereses. Estos criterios están asociados principalmente a la

⁴ Conciene a la forma y al estudio del comportamiento y fenómenos psíquicos de aquellos que toman las decisiones (reales o ficticios), así como las condiciones por las que deben ser tomadas las decisiones óptimas.

⁵ DSS por sus siglas en inglés Decision Support Systems. Son sistemas de información basados en computadora los cuales combinan modelos y datos para intentar resolver problemas no estructurados.

evaluación del desempeño docente del profesor. Todo lo anterior se alcanza utilizando características propias de los SMDB, permitiendo así concentrar la complejidad de la solución en el área de Bases de Datos.

Por último se creará una interfaz mediante un lenguaje de programación con la que el usuario del sistema interactúe. A través de ella pueda llevar a cabo la elaboración de su horario de clases, tomando en cuenta los rankings generados a partir de la información recolectada por el sistema.

En conjunto se habrá implementado un sistema de apoyo a la toma de decisiones, desde el diseño de la base de datos hasta la construcción de la interfaz para su explotación. Preponderando el papel de la base de datos y de las operaciones definidas sobre sus datos como el componente principal de la solución.

Finalmente el principal beneficiado será el alumno al proveerlo de una herramienta que le otorga los datos adecuados para tomar la decisión de qué materia seleccionar de acuerdo a sus preferencias.

Esta tesis está organizada de la siguiente manera, en el capítulo 2 el lector de esta tesis podrá encontrar la descripción de los sistemas de negocios inteligentes (BI), así como el proceso de desarrollo genérico que es llevado a cabo para construirlos. El capítulo 3 involucra parte de la Teoría de las Decisiones que justifica la toma de decisiones que realiza el SATDEH. En el capítulo 4 se encuentra una descripción acerca de algunos sistemas de soporte de decisiones (DSS) y la relación que éstos tienen con el SATDEH. Para el capítulo 5 el lector encontrará los supuestos en el diseño de la base de datos, la construcción de la misma y los procedimientos almacenados (SP) que representan el motor del SATDEH. Posteriormente, el capítulo 6 detalla el proceso de desarrollo seguido para la construcción del SATDEH. Finalmente se presentan las conclusiones sobre este proyecto y el trabajo futuro.

CAPÍTULO 2 Business Intelligence (BI)

En este capítulo se presentarán las generalidades de los BI, desde su definición, para qué sirven, sus ventajas y desventajas y algunos ejemplos de BI existentes.

2.1 ¿Qué son los BI?

Un sistema de Business Intelligence o BI, se puede definir como: *"un software que tiene como finalidad transformar los datos de una organización en información y conocimiento para obtener una ventaja competitiva"* (Preguntia, 2013). Desde una perspectiva más general: *"BI es un mercado que tiene como objetivo facilitar el acceso y análisis de la información corporativa, proporcionando las herramientas tecnológicas adecuadas para la toma de decisiones"* (BI fácil, 2013).

Aunque por otro lado otros autores tienen una concepción distinta de los BI: *"BI no es ni un producto ni un sistema. Se trata de una arquitectura y una colección de operaciones Integradas, así como aplicaciones de apoyo a la toma de decisiones y bases de datos que proporcionan el fácil acceso a la comunidad de los negocios y a los datos empresariales"* (Moss & Atre, 2003).

Para esta tesis, tomando como base las definiciones anteriores, se considerará la siguiente definición de BI:

BI es un proyecto para cierto tipo de negocios, que apoyado en una base de datos, un repositorio de metadatos y un adecuado manejo de la información, proporciona, mediante herramientas tecnológicas, una gama de argumentos sólidos para la toma de decisiones, obteniendo con ello una solución al caso de negocio planteado.

Debido a que se asumirá que BI es un proyecto se debe precisar dicho concepto: *"Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único"* (PMBOK, 2004).

Cabe resaltar, que para llevar a cabo un proyecto es necesario que éste contenga un proceso que, a su vez, será ejecutado por los individuos involucrados en el proyecto. Un proceso está compuesto por fases delimitadas por hitos. Una fase es un conjunto de actividades o tareas realizadas por roles, algunas de esas tareas o actividades pueden requerir de productos de trabajo tanto de entrada como de salida. Para este proyecto se utilizarán dos procesos, uno para el desarrollo del sistema y otro para la explotación de los datos.

2.1.1 Proceso de desarrollo

De acuerdo con (Moss & Atre, 2003) los BI siguen la siguiente estructura en su operación:

- **Justificación.** Se refiere a la evaluación del caso de negocio a realizarse.
- **Planificación.** Consiste en la evaluación de la infraestructura organizacional y la planificación del proyecto.
- **Análisis del negocio.** Consiste en la especificación de los requerimientos del proyecto, el análisis de datos, análisis del repositorio de metadatos y construcción del prototipo de la solución.

- **Diseño.** Se concentra en el modelado de la base de datos, el diseño del repositorio de metadatos y el ETL.
- **Construcción.** Consiste en desarrollar tanto las aplicaciones como el repositorio de metadatos, así como también la minería de datos.
- **Implementación.** En esta parte se incluye la implementación y la liberación de la evaluación de la información.

Una primera visión esquemática del proceso de desarrollo antes descrito sería el mostrado por la Figura 2.1

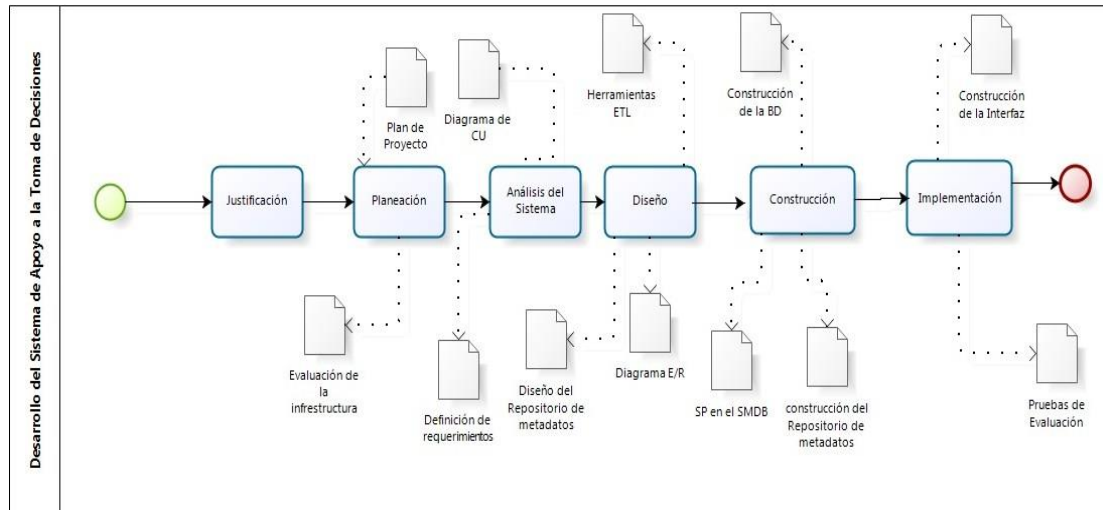


Figura 2.1. Esquema del desarrollo de un BI.

La descripción y adaptación del esquema anterior será presentada en el Capítulo 6 con mayor detalle y específicamente sobre los requerimientos que necesita el SATDEH.

2.1.2 Proceso de explotación (ETL)

Por sus siglas en inglés Extract/Transform/Load, el proceso de ETL forma parte de los BI. La idea es que una aplicación ETL extraiga los datos almacenados a través de un SMDB, estos datos pueden haber sido obtenidos, por ejemplo, a través de una consulta. Posteriormente se realiza una transformación a los mismos, dicha transformación puede ser directa o bien, previamente, validar y filtrar los datos para así proceder a transformarlos, este cambio puede ser por medio de procesos cualitativos o cuantitativos. Por último, los datos arrojados después de la transformación, deben ser almacenados y así encontrarse listos para un análisis posterior que el usuario hará según le convenga.

A continuación se describen cada una de las fases del proceso ETL de acuerdo con (Wikipedia, 2013):

- **“Extraer:** En esta parte del proceso se extraen los datos del sistema de origen. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación. Una parte del proceso de extracción es la de analizar los datos extraídos, de lo que resulta una verificación si los datos cumplen con la estructura que se esperaba. De no ser así los datos son

rechazados. Un requerimiento importante para la tarea de extracción es que ésta cause un impacto mínimo en el sistema origen. Si los datos a extraer son muchos, el sistema de origen podría verse afectado en su desempeño e incluso colapsar, provocando que éste no pueda utilizarse con normalidad para su uso cotidiano.

- **Transformar:** Dentro de esta fase se aplica una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Algunas fuentes de datos pueden requerir alguna manipulación de los datos.
- **Cargar:** Aquí los datos de la fase anterior son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos”.

Existen dos maneras para ejecutar el proceso de carga: la **Acumulación simple** que consiste en concentrar en una única transacción la totalidad de las transacciones realizadas durante un período de tiempo; y el **Rolling**, que opta por agrupar la información de manera jerárquica o resumiéndola de acuerdo a diversas unidades de tiempo”.

2.2 ¿Para qué sirven los BI?

Las herramientas de BI permiten reunir, depurar y transformar los datos obtenidos de operaciones diarias en información estructurada lista para su explotación (Kimball & Caserta, 2004). Es justamente en la parte de explotación donde los datos, ya como información, son analizados y se convierten en conocimiento que sirve para el soporte a la toma de decisiones de un negocio. Los BI cuentan con funcionalidades que permiten la modificación de los reportes que se generan dependiendo del público objetivo de los mismos.

Dado que el sistema que se realizará en este proyecto estará al servicio de una entidad educativa, éste deberá adaptarse a las necesidades de los alumnos que son a quienes el BI proporcionará la información. Así mismo, será usado por los administrativos de la entidad encargados de administrarle la información al sistema, por lo que dicho sistema deberá proveer herramientas que faciliten la obtención, inyección y manejo de los datos.

2.3 BI Existentes

Algunos de los BI existentes se presentan a continuación.

Microsoft Business Intelligence: Recabando lo descrito en (White, 2010) Microsoft Business Intelligence conforma conjunto de aplicaciones de servidor, cliente y programador que proporciona la información requerida de acuerdo a las necesidades del tipo de negocio, además muestra información que se puede usar directamente en donde trabajan, colaboran y toman decisiones los usuarios.

SAS® Visual Analytics: SAS Visual Analytics ofrece una amplia variedad en BI, entre ellas se pueden mencionar a Enterprise Guide y Enterprise Miner, con los cuales el usuario puede realizar reportes, informes, análisis estadísticos de la información y aplicar minería de datos a diversos casos de negocio. La esencia de SAS se encuentra en la forma en que traduce de una manera simple el lenguaje natural al

lenguaje SQL para que cualquier usuario pueda utilizarlo sin necesidad de tener conocimientos, tanto en bases de datos como en dicho lenguaje. De igual manera SAS por medio de Enterprise Miner permite realizar diversas pruebas estadísticas, así como ajustar modelos, tales como arboles de decisión, redes neuronales y regresiones, a un conjunto determinado de datos y así proporcionar de manera concreta el mejor modelo que dé solución al caso de negocio planteado. SAS posee herramientas robustas y con grandes alcances en el ámbito empresarial.

SAS combina la analítica en la industria con una arquitectura *in-memory* (SAS, 2013). La solución de motor *in-memory* utiliza una memoria alternativa que permite una mayor velocidad en el procesamiento de la información, y está diseñada específicamente para sistemas que manejan una gran cantidad de datos, permitiendo el análisis de grandes volúmenes de datos y la obtención de información crítica en un tiempo razonable (SAS, 2013).

Cabe mencionar que la arquitectura *in-memory* fue un elemento importante para concebir la idea de centrar la solución de SATDEH en la base de datos, en particular con el uso de SPs.

Business Objects: enfocada a preparar informes y desarrollar tableros de control. Sus interfaces buscan ser ricas en atributos y facilitar a los usuarios la comprensión de negocio a través de sus datos, obteniendo tácticas para alinear personas, procesos e información con las estrategias corporativas (Crystalis, 2013). Sus principales capacidades son:

- Permite realizar informes dinámicos.
- Proporciona una visión a futuro del caso de negocio bajo ciertas condiciones iniciales.
- Permite tener un panorama general del caso de negocio tratado.
- Permite la manipulación de los reportes para simular o representar comportamientos o situaciones definidas por el usuario.

MicroStrategy: de acuerdo con Gartner (Gartner, 2013) MicroStrategy es líder global en BI. MicroStrategy ofrece distintas soluciones de negocio proveyendo información mediante la generación de reportes, herramientas de análisis para detección de tendencias, permite monitorear en tiempo real el desempeño de la organización y ofrece la opción de manipular reportes con el fin de visualizar posibles escenarios futuros (MicroStrategy, 2014).

IBM Cognos 8 Business Intelligence: Este BI permite al usuario la integración de datos, la elaboración de reportes, permite inferir en los datos para obtener indicadores representativos en el caso de negocio planteado. En cuestiones de usabilidad es flexible para que los usuarios se familiaricen de una manera simple con su interfaz y sus funciones. Logra un manejo multivariado de los datos para visualizar factores importantes que beneficien o afecten a cierto caso de negocio en tiempo determinado (IBM, 2014).

2.4 Ventajas de los BI

Los BI presentados en la sección anterior cuentan con varias ventajas, después de analizarlos, a continuación se presentan algunas de ellas:

- Permiten recabar información de distintas fuentes de datos y consolidarla en una sola
- Proveen de lineamientos de seguridad a los datos, así como de los usuarios por medio de un manejo adecuado de los roles
- La solución de problemas técnicos puede realizarse, en la gran mayoría de los casos, por el usuario mismo
- Optimizan el tiempo de ejecución para la elaboración de reportes e informes y permiten al usuario dedicar más tiempo a la parte del análisis de información
- Permiten la transformación de la información y su almacenamiento para que ésta se encuentre disponible cuando el usuario la requiera
- Mejora la eficiencia organizacional de la empresa
- Permite detectar nuevas oportunidades de negocio
- Reduce los gastos de la empresa

2.5 Desventajas de los BI

En vista de que un BI tiene un alto grado de complejidad, su estructura es sumamente sofisticada y el alcance que tienen en la solución de un caso de negocio es grande, esto puede traer consecuencias desfavorables al mismo, las cuales se ven reflejadas en las desventajas que presentan y que a continuación se listan:

- Entre más sofisticado sea el sistema se dificulta también el análisis de la información.
- Poca facilidad de personalización y adecuación de funcionalidades estándar
- Debido a los cambios trascendentes en las tecnologías de la información, puede que el BI deje de ser útil después de dicho cambio o bien deje de ser compatible
- Conforme pasa el tiempo el BI puede ser obsoleto si deja de cumplir con los requerimientos de los usuarios, los cuales resultan ser complicados de actualizar
- Falta de personal con los conocimientos y habilidades requeridos para la implementación, uso y mantenimiento
- Requiere una amplia inversión, tanto de tiempo como de logística para su desarrollo
- Su construcción e implementación requieren de un alto consumo de recursos monetarios y humanos así como contar con una infraestructura adecuada

CAPÍTULO 3 Sistema de Soporte de Decisiones (DSS)

En este capítulo se presentarán las generalidades de los DSS, desde su definición, para qué sirven, sus ventajas y desventajas y algunos ejemplos existentes.

3.1 Introducción a los DSS

Intuitivamente el nombre de Sistema de Soporte de Decisiones (DSS por sus siglas en inglés) da una idea simple de lo que es: un sistema que proporciona ayuda para tomar decisiones, ya sea en el ámbito empresarial o cualquier otro al que dicho sistema esté orientado. Un DSS, a grandes rasgos, es una herramienta de apoyo para tomar decisiones, sin embargo, un DSS está conformado por diversos programas y herramientas que en conjunto permiten su operación. Cada DSS realiza tareas diferentes y su desempeño depende del objetivo para el que fue creado.

Los DSS tienen como objetivo primordial proporcionar información a partir de datos. Dicha información debe ser específica y haber pasado por un proceso de filtrado por el mismo sistema, para así ser lo suficientemente útil y dar argumentos sólidos al usuario que le permitan tomar decisiones. Una definición simple de un DSS es la siguiente: *"Un DSS es un sistema informático utilizado para servir de apoyo, más que automatizar, el proceso de toma de decisiones"* (Alter, 1980).

Abordando una definición más completa la cual asumiremos como definición de DSS para esta tesis es la de (Ruíz, Hernández & Giraldo, 2009):

Un DSS es definido como un sistema interactivo basado en computador que da soporte a los tomadores de decisiones en vez de reemplazarlos, utiliza datos y modelos para resolver problemas con diferentes grados de estructura: no estructurados, semiestructurados, tareas estructuradas y no estructuradas, semiestructuradas y no estructuradas y se enfoca en la efectividad más que en la eficiencia de los procesos de decisión (facilita el proceso de decisión).

Los DSS se distinguen de cualquier otro sistema que pueda existir dentro de una organización porque, de acuerdo con (Ruíz, Hernández & Giraldo, 2009), está integrado por los siguientes subsistemas:

- *“Administrador de datos: incluye una base de datos que contiene información relevante para una situación y es administrada por un sistema manejador de la base de datos (SMBD).*
- *Administrador del modelo: es un paquete software que contiene modelos cuantitativos, estadísticos, financieros y científicos que proveen capacidades analíticas al sistema.*
- *Interfaz de usuario: el sistema debe proveer una interfaz gráfica de usuario desde la cual éste pueda utilizar las funcionalidades del sistema.*
- *Administrador del conocimiento: soporta cada uno de los demás subsistemas o actúa como componente independiente. Provee inteligencia para argumentar la toma de decisiones”.*

3.2 DSS Existentes

En la actualidad existen diversos DSS, sin embargo, una de las principales desventajas de estos sistemas es su alto costo y su alta especificidad. Para propósitos de esta tesis, se realizó una búsqueda de DSS existentes y se analizaron los que más se acercaban a los objetivos del sistema a construir. Entre los DSS analizados se destacan Decision Oven, Decision Explorer y LPA VisiRule, todos software libre. A continuación se presentan con mayor detalle.

Decision Oven: Es un DSS que apoya al usuario en tomar decisiones a partir de que éste le proporcione opciones y criterios, devolviéndole así un ranking de las opciones que el usuario proporcionó para que, en base a dicho ranking, el usuario pueda tomar una decisión.

Esta es una herramienta muy sencilla desde la perspectiva de usabilidad, ya que cuenta con una interfaz de usuario accesible y resuelve problemas de toma de decisiones cotidianos. Sin embargo, este DSS no es lo suficientemente robusto para dar solución a otro tipo de problemas de toma de decisiones, como por ejemplo el problema planteado en esta tesis, puesto que no cuenta con una base de datos adaptable que permita almacenar la información requerida para cubrir los requerimientos que se pretenden implementar.

Decision Oven cuenta con una versión libre que puede descargarse desde la siguiente liga: <http://www.decisionoven.com/>

En la Figura 3.1, se muestra una captura de pantalla de esta herramienta, en la que se modela el problema de la elección de un automóvil a partir de su velocidad, precio y capacidad. Dicha figura muestra los criterios, los pesos que estos poseen en la decisión, si "más" de un criterio es mejor o "menos" de un criterio es mejor. En este caso el criterio con mayor peso asignado es el Precio con un 59% mientras que la capacidad tiene un 20% y la velocidad un 30%.

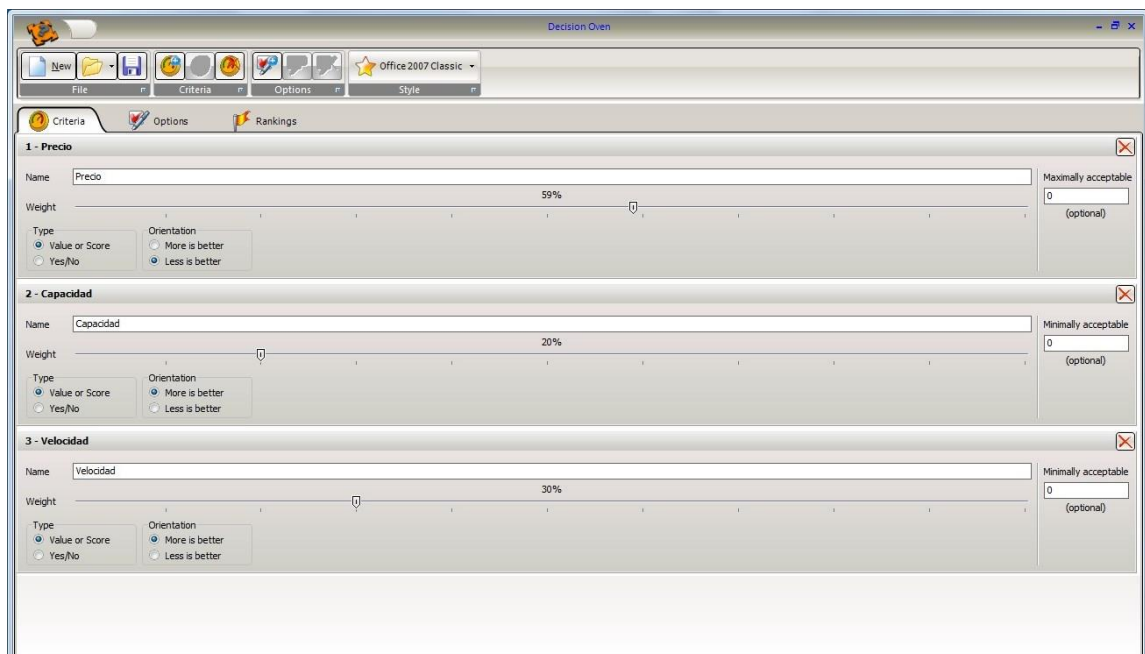


Figura 3.1. Captura de pantalla de Decision Oven que contiene los criterios de decisión.

La Figura 3.2 muestra las opciones proporcionadas a Decision Oven sobre las cuales genera el ranking de decisión, en este ejemplo tres marcas de vehículos.

Option name	Precio	Capacidad	Velocidad
Suburban	1200000,00	10,00	160,00
BMW	1100000,00	5,00	180,00
Ferrari	1150000,00	2,00	220,00

Figura 3.2. Captura de pantalla de Decision Oven que contiene las opciones para la decisión.

La Figura 3.3 muestra el ranking de los tres automóviles que se ingresaron en la parte de opciones, la mejor recomendación de Decision Oven es el BMW según los criterios de la Figura 3.1, en segundo lugar está el Ferrari y por último la Suburban.

	Option name	Total score	Precio	Capacidad	Velocidad
1	BMW	70,18	54,13	6,88	9,17
2	Ferrari	54,59	27,06	0,00	27,52
3	Suburban	18,35	0,00	18,35	0,00

Figura 3.3. Captura de pantalla de Decision Oven que contiene el ranking para la toma de decisión.

Decision Explorer: Este BI permite manejar de una forma cualitativa la información que se presenta en casos de negocio sumamente complejos y que poseen un alto grado de incertidumbre. Logra que el usuario plasme su manera de pensar respecto al caso de negocio, así como sus ideas y permite analizar cuidadosamente los que el usuario ha expuesto. Su enfoque está centrado en un panorama conceptual pues se basa en mapas mentales y la manipulación de los esquemas respecto de información específica. De acuerdo con (Banxia Decision Explorer, 2013) este sistema ha sido

desarrollado por académicos de las universidades de Bath y Strathclyde y ahora por Banxia Software.

Con esta herramienta se pueden modelar las relaciones entre los componentes de un tema, mismos que van siendo descubiertos y analizados, el resultado es un mapa mental que presenta una vista conceptual de un problema dado a partir del cual es posible tomar decisiones.

Este software ayuda en el análisis de los modelos, permitiendo así trabajar de forma rápida y eficaz con muchos conceptos, enfocándose en mantener lo más simple posible la representación del problema o tema en discusión. Esto se logra estructurando las diferentes opiniones o posibles decisiones como un mapa mental, que posteriormente, es discutido para tomar una decisión.

Decision Explorer cuenta con una versión para estudiantes libre que se puede descargar desde la siguiente liga: <http://www.banxia.com/dexplore/editions-available/student-edition/>.

A continuación en la Figura 3.4 se muestra un ejemplo sobre la forma en que este DSS ayuda a explorar las decisiones que se toman. Retomando el caso de los automóviles y asumiendo que se le ha hecho caso a Decision Oven vemos una serie de consecuencias que podrían suceder si se asume comprar el BMW.

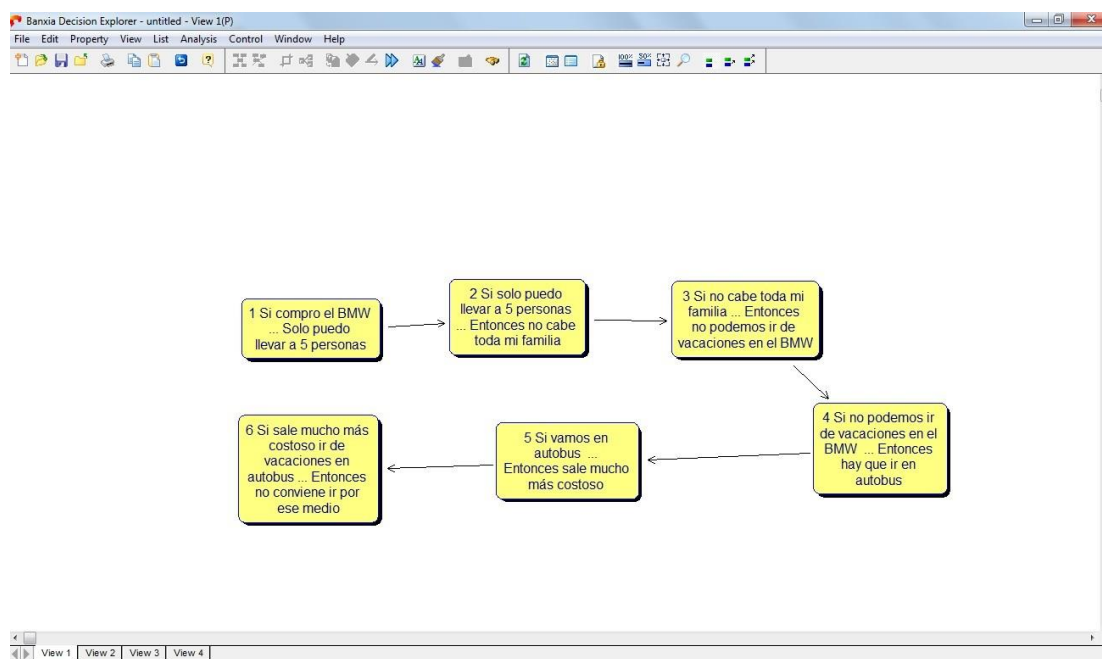


Figura 3.4. Captura de pantalla de Decision Explorer que muestra las posibles consecuencias de elegir comprar el BMW en el caso de los automóviles.

Este DSS provee de una herramienta que permite plasmar las distintas posibles decisiones y sus consecuencias. A partir de esta visualización es posible explorar las consecuencias de las decisiones que el usuario ha tomado.

LPA VisiRule: es una herramienta gráfica para el diseño, desarrollo y distribución de reglas de negocio y las aplicaciones de apoyo a las decisiones. Este sistema basa su funcionalidad en dibujar un diagrama de flujo que represente el procedimiento de decisión. Las principales características de VisiRule, de acuerdo con (LPA, 2013) incluyen:

- *“Herramienta de dibujo y diseño: permite generar diagramas significativos, verificando, en tiempo de edición, la sintaxis de las expresiones, evitando conexiones incompatibles entre elementos.*
- *Diseño de apoyo automático: representa un conjunto de funciones de edición para la alineación automática y el espaciado de los diagramas.*
- *Visualización de opciones configurables: permite suprimir la información cuando sea necesario para ayudar a producir gráficos compactos.*
- *Cajas Código: permite acceder a código de procedimientos y funciones externas escritas en Flex, Prolog o lenguajes tradicionales como Visual Basic, C y C + +”.*

En conclusión este sistema permite la toma de decisiones a partir de segmentar gráficamente las estrategias y alternativas que se presentan dentro de un caso de negocios, permitiendo construir y definir reglas de decisión de una manera que logre facilitar la toma de decisiones.

Este software es de paga y aunque muestra gráficamente el desarrollo de un caso de negocio y permite visualizar cómo se encuentra estructurado dicho caso de negocio, no permite identificar los detalles dentro de la información involucrada, tales como inferencia estadística o minería de datos. Una versión libre de este sistema puede descargarse de la siguiente liga: http://www.lpa.co.uk/dow_tri.htm

La Figura 3.5 muestra una captura de pantalla de LPA VisiRule en la cual se ha elaborado un árbol binario de decisión. Dentro de dicho árbol se muestra las distintas decisiones que se pueden tomar, así como las diversas alternativas sobre el caso de elección de profesores.

El criterio principal de decisión se encuentra en la parte superior del árbol, conforme éste desciende se van priorizando los criterios. El óvalo de color verde representa el inicio del árbol de decisión, los óvalos amarillos representan los puntos de decisión que se pueden presentar en el plan de negocio. Mientras que los óvalos grises representan las posibles direcciones hacia donde se encamina la decisión del usuario. Por último los óvalos rojos representan los valores posibles que finalmente terminaría por elegir el usuario.

Este software proporciona una herramienta muy flexible e interesante para la toma de decisiones. Sin embargo, para resolver el problema planteado en esta tesis, sería necesario construir un árbol de decisión cada vez que se quisiera elegir un profesor, por otro lado no es posible incluir la persistencia de datos con esta herramienta, por lo que dificultaría la administración de la información recabada mediante las evaluaciones de cada profesor.

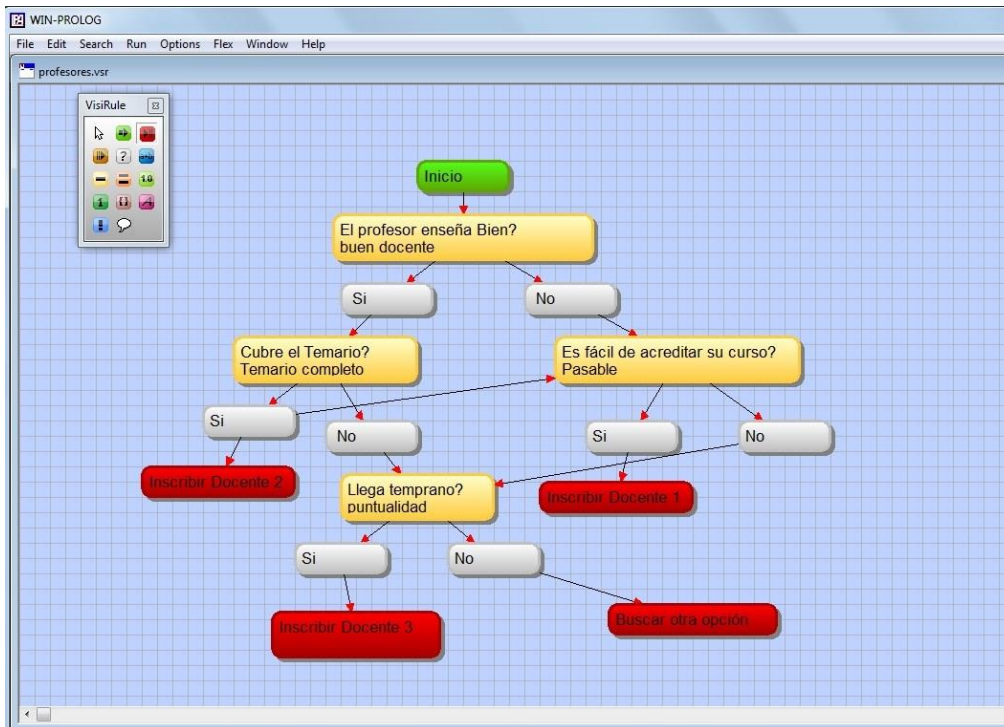


Figura 3.5. Captura de pantalla de LPA VisiRule que muestra el caso de decisión de la elección de profesores mediante un árbol de decisión binario

3.3 Ventajas de los DSS

Los DSS presentados en la sección anterior cuentan con varias ventajas. Después de analizar las referencias de los mismos, instalarlos y utilizarlos, a continuación presentamos algunas de ellas complementadas con las identificadas por (Calderón et al., 2009):

- Logran omitir la construcción física de prototipos de diversos proyectos, esto implica una disminución en tiempo, un menor costo en logística y esfuerzo de implementación.
- Son sistemas con una usabilidad muy flexible para que cualquier usuario logre aprender su operación.
- Permiten el manejo de los datos con mayor fluidez y en tiempo óptimo.
- Estructuran la información de manera que el usuario puede tomar decisiones con certidumbre respecto al caso de negocio planteado.
- Permiten la toma de decisiones aun cuando existen cambios drásticos e inesperados en los factores del caso de negocio.
- Optimizan el tiempo de ejecución de procesos complejos.
- Ayuda al usuario a liberar su creatividad para que pueda tomar decisiones de una manera más sencilla.
- Permite la detección de puntos cruciales para el desarrollo del caso de negocio.
- Pueden ser transformados en BI e incrementar su potencial, en algunos casos de una forma no tan costosa.

3.4 Desventajas de los DSS

Es evidente que un DSS tiene muchas ventajas, pero así mismo éstos, por su estructura sofisticada y por las tareas complejas que realizan, tienen un costo alto, lo que representa una desventaja. Como se explicó en el capítulo 2, los DSS forman parte de los BI, por esta razón comparten muchas de las desventajas ya mencionadas anteriormente. A continuación se retoman algunas de dichas desventajas:

- Necesidad de mayor conocimiento de uso por parte del tomador de decisiones.
- Herramientas costosas en implementación.
- Son extremos, ya que pueden ser demasiado específicos en cuanto a los problemas que resuelven o tan generales que resulta costoso adaptarlos a las necesidades particulares del usuario.

3.5 Aplicaciones de los DSS

Los DSS generalmente se utilizan en cualquier caso de negocios que proporcione beneficios a las organizaciones que las utilizan. Un DSS permite ahorrar recursos y tiempo, lo cual hace que en la actualidad sean requeridos para distintos fines. Algunas de las aplicaciones de los DSS son las siguientes:

- El control escolar de una entidad educativa de nivel superior puede usar un DSS para realizar los horarios de clases, al realizar la asignación de aulas disponibles.
- En empresas comerciales, un DSS es utilizado para la elaboración de reportes de ventas, pronósticos, inventarios y disponibilidades en la producción.
- En el área de inversiones, los planeadores financieros utilizan DSSs para diversificar los fondos de un cliente entre un grupo apropiado de opciones de inversión, esto para minimizar el riesgo y aun proporcionar una tasa de rendimiento adecuada sobre la inversión.

Entre las aplicaciones más representativas de los DSS, en (Ruíz, Hernández & Giraldo, 2009) se mencionan:

- *“El ejército estadounidense desarrolló un DSS del potencial humano del personal, con el propósito de ayudar con las decisiones de reclutamiento, entrenamiento, educación, reclasificación y ascensos. El DSS usa la optimización y la simulación para preparar modelos de las necesidades, requerimientos y requisitos del personal. Incluye características de "qué sucedería si" y puede interactuar con una base de datos en línea y con otros programas de análisis estadístico.*
- *Hewlett-Packard desarrolló Quality Decision Management para realizar funciones de control de la calidad. Puede ayudar con la inspección de las materias primas, las pruebas de productos y el análisis estadístico.*
- *Cinergy Corporation, una compañía de servicios públicos de electricidad con oficinas en Cincinnati, Ohio, EUA, desarrolló un DSS para disminuir el tiempo de espera y los esfuerzos requeridos para tomar decisiones al comprar carbón”.*

En cuestión de aplicaciones, un DSS puede ser construido para resolver problemas de toma de decisiones y hacer de éste una herramienta que ayude a dar solución un sin número de situaciones. Para esta tesis la aplicación del sistema va dirigida a la elección de profesores dentro de una entidad educativa donde existe un intercambio continuo de información entre el DSS y el alumno.

CAPÍTULO 4 Teoría de las Decisiones

En este capítulo se presentará una introducción a la Teoría de las Decisiones, los tipos de decisión que existen y algunos ejemplos de cómo se aplica a la práctica.

4.1 Introducción a la Teoría de las Decisiones

En la vida cotidiana, en el ámbito personal y profesional nos vemos obligados a tomar decisiones en diversas situaciones en las que tenemos que elegir entre varias alternativas. Lo que se busca instintivamente, cuando se presenta el momento de tomar una decisión, es elegir lo “mejor” entre lo “posible”, ya que ante cualquier situación de toma de decisiones las condiciones cambian constantemente, así mismo también podría variar nuestro objetivo. Lo anterior puede asemejarse a un problema de optimización⁶ en donde lo “mejor” está representado por la solución óptima, lo que dará la pauta para tomar la mejor decisión. Mientras que lo “posible” está representado por el conjunto de soluciones factibles que se acercan al objetivo, matemáticamente, dicho objetivo está representado por una función llamada *función objetivo*.

A continuación se presenta un ejemplo en donde se puede apreciar, de manera más clara, lo que se describió anteriormente. Supongamos que una persona quiere adquirir un automóvil y solo cuenta con un presupuesto de 250,000 unidades monetarias (um). La persona acude a una agencia de automóviles y se encuentra con que sólo le es posible elegir entre tres automóviles respetando el presupuesto con el que cuenta. Los automóviles son una camioneta, un auto deportivo y un auto compacto, ellos cuestan 250,000 um, 245,000 um y 225,000 um respectivamente.

Si su objetivo fuera solo comprar un automóvil tendríamos que el conjunto de decisiones posibles o factibles está formado por los tres automóviles, si su objetivo fuese conservar el mayor dinero posible, resulta claro que la decisión óptima sería comprar el automóvil más económico. Ahora bien si su objetivo fuera comprar el automóvil que le proporcione el mayor espacio para transportar objetos, debería decidirse por la camioneta, sin embargo este automóvil es el de mayor costo de los tres.

El ejemplo anterior podría volverse más complejo y sofisticado si se agregan más restricciones, por ejemplo, el color del automóvil, la velocidad máxima alcanzada, el tamaño, el número de puertas y todo aquello que forme parte de un objetivo mucho más específico. Con esto en cuenta, la toma de decisión se vuelve más difícil debido a que se tienen muchos criterios de por medio, y más aún cuando el espacio de lo “posible” se vuelve mucho más pequeño debido a los requisitos pedidos.

Lo mencionado anteriormente es todo un proceso, dicho proceso es llamado el “proceso de decisión”. La definición que asumiremos para esta tesis es la de (Romero, 1996):

"En su dimensión más básica un proceso de toma de decisión puede concebirse como la elección por parte de un centro decisor (un individuo o un grupo de individuos) de «lo mejor» entre «lo posible»".

De manera más detallada la descripción de este proceso puede verse como sigue según (Caballero et. al., 2008): *"El proceso de tomar una decisión se puede describir de la siguiente forma: planteado un problema se establece el conjunto de puntos factibles o admisibles, en nuestro caso el correspondiente conjunto que nos determina las restricciones del problema. Después se le asocia a cada alternativa, criterio u*

⁶ Un problema de optimización es aquel en el que se busca obtener la solución que mejor satisfaga ciertas condiciones o restricciones del problema entre un conjunto de soluciones posibles.

objetivo un grado de deseabilidad. Posteriormente, se busca, mediante cualquier técnica, una solución o un conjunto de posibles soluciones alternativas. Dichas soluciones posibles son aquellas que satisfacen las restricciones y los deseos o preferencias, las cuales se efectúan sobre los objetivos planteados. Este hecho nos lleva a distinguir entre los problemas tecnológicos, en los cuales sólo se realiza un proceso de planteamiento, medición y búsqueda, y los problemas denominados multiobjetivo (son aquellos problemas cuya solución busca satisfacer diversos objetivos, cada uno de ellos sujeto ciertas condiciones) los cuales implican de manera natural un proceso de toma de decisión."

4.2 Tipos de decisión

Existen diversos tipos de decisiones, siendo las más relevantes las decisiones programadas, las no programadas y las multicriterio. A continuación detallaremos cada una de ellas.

Las **decisiones programadas**, también conocidas como esquemas de ejecución, son procedimientos repetitivos y rutinarios. Se explican mediante un conjunto de reglas o procedimientos de decisión. Se reflejan en libros sobre reglas, tablas de decisión y reglamentaciones. Implican decisiones bajo certeza, en razón de que todos los resultados o consecuencias son conocidos de antemano (Borea & Vélez, 2003).

Un ejemplo simple de este tipo de decisiones es el juego de "memorama" en donde después de varios intentos y una vez que se tienen identificadas la cartas que forman un par se procede a tomar la decisión de voltear las cartas adecuadas.

Las **decisiones no programadas** se refieren a los problemas no estructurados o de gran importancia. A diferencia de las programadas, éstas no tienen reglas o procedimientos preestablecidos. Las decisiones programadas son factibles de ser delegadas a los niveles medios de la organización o se pueden automatizar, algo que no puede hacerse con las decisiones no programadas (Borea & Vélez, 2003).

En general es preferible trabajar con decisiones programadas, ya que éstas cuentan con procedimientos preestablecidos que permiten llegar a una toma de decisiones mucho más sencilla, aunque esto no siempre es así. En cuanto a las decisiones no programadas, en ocasiones es posible descomponerlas en decisiones programadas especificando todas las restricciones bajo condiciones normales que permitan definir reglas de decisión que, a su vez, se puedan programar para asemejarse a la decisión original. Cabe mencionar que se debe tener cuidado con las condiciones de decisión, ya que si dichas condiciones no son adecuadas podría sesgarse considerablemente la solución a un problema.

Al tratar de dar solución a decisiones no programadas se corren diferentes tipos de riesgos entre ellos, el más importante, la aplicación de reglas inapropiadas, que al aplicarlas pueden llevar a tomar una decisión muy alejada de lo que realmente se está buscando.

Las **decisiones multicriterio** son aquellas que consideran a más de un criterio para dar solución a un problema de toma de decisiones multiobjetivo. De ahora en adelante para alcanzar los objetivos de este proyecto, se abordaran únicamente las decisiones multicriterio, pues son el tipo de decisiones que se requieren para el desarrollo de este sistema.

4.3 Toma de Decisiones

Antes de hablar sobre la toma de decisiones es preciso dejar claros algunos conceptos basados en las definiciones de (Romero, 1996). El primero de ellos es el concepto de **atributo**, que "*se refiere a valores que el centro decisor enfrenta ante un*

problema de toma de decisiones. Para poder conceptualizar dichos valores como atributos, se necesita que éstos se puedan medir independientemente de los deseos del centro decisor. A su vez, se necesita que sean aptos para expresarse como una función de las correspondientes variables de decisión” (Romero, 1996). Retomando el ejemplo de los automóviles, el precio y la velocidad de éstos se pueden conceptualizar como atributos, pues es posible medirlos con independencia de los deseos del centro decisor.

Posteriormente otro concepto importante es el de **objetivo** que de acuerdo con (Romero, 1996), *“los objetivos representan direcciones que puedan mejorar a los atributos que se están considerando. Dicha mejora puede interpretarse en el sentido “más del atributo mejor” o bien “menos del atributo mejor”. “Más del atributo mejor” corresponde a un proceso de maximización y el caso “menos del atributo mejor” a un proceso de minimización”*. Un ejemplo de objetivo podría ser maximizar la utilidad de las ventas de una compañía productora de café, en este caso podemos ligar el objetivo con el atributo rendimiento para el cual entre más rendimientos tenga la compañía es mejor.

Ya que se tiene claro el concepto de objetivo podemos definir lo que es una **meta**. Retomando nuevamente la idea planteada en (Romero, 1996). *“La meta puede definirse a partir de un nivel de aspiración. Un nivel de aspiración representa un porcentaje aceptable de logro para el correspondiente atributo. La combinación de un atributo con un nivel de aspiración genera una meta”*. Así, retomando el ejemplo de los automóviles, si se desea comprar al que tenga un precio máximo de 150,000 un tendremos una meta.

Finalmente, se tiene al concepto **criterio**, que en general engloba los conceptos anteriores. Concretamente y basados en (Romero, 1996) *“los criterios son los atributos, objetivos o metas que se consideran relevantes en un cierto problema de toma de decisiones”*. El análisis de una decisión multicriterio conlleva, en general, a tratar con problemas de toma de decisiones que tienen diferentes atributos, objetivos o metas.

Otro de los puntos importantes dentro de esta teoría es el criterio de *optimalidad paretiana*, de acuerdo con (Romero, 1996), en 1896, Vilfredo Pareto propuso un criterio de optimalidad, el cual recibe su nombre y resulta ser de suma importancia para la teoría económica. En su primera formulación, Pareto expone que: *“una colectividad se encuentra en un estado óptimo si ninguna persona de dicha colectividad puede mejorar su situación sin que empeore la situación de alguna otra persona de la misma”*.

Lo anterior se conoce como eficiencia paretiana. El criterio de Pareto ha logrado un gran nivel de aceptación por parte de las personas involucradas en situaciones de toma de decisiones. Retomando la idea expuesta en (Romero, 1996), este criterio, de optimalidad paretiana, puede llevarse directamente del aspecto económico al análisis de decisión multicriterio. Para lograrlo solo basta intercambiar el concepto original de Pareto de «sociedad» o «colectividad» de personas por el de conjunto de criterios. Con ello cada uno de los criterios representa a una persona en esta nueva interpretación. Este traslado del concepto de optimalidad paretiana es crucial para el desarrollo de diferentes enfoques en el paradigma multicriterio.

En conclusión la eficiencia paretiana es una condición necesaria para poder garantizar soluciones racionales que se generan a partir de los diferentes enfoques multicriterio que existen. De acuerdo con (Romero, 1996), una definición formal del concepto de optimalidad paretiana dentro del campo multicriterio es:

“Un conjunto de soluciones es eficiente, o Pareto óptimas, cuando está formado por soluciones factibles (esto es, que cumplen las restricciones), tales que no existe otra solución factible que proporcione una mejora en un atributo sin producir un empeoramiento en al menos otro de los atributos”.

Una vez expuesto lo anterior se procede a describir los elementos que permiten tomar una decisión, dichos elementos requieren de una metodología que se presenta a continuación.

El primer elemento a tomar en cuenta es lo que se conoce como la normalización de los criterios. La normalización de los criterios es un proceso que no siempre es necesario, sin embargo, (Romero, 1996) sugiere que debe ser aplicado por las siguientes razones:

- **“Unidades de medición diferentes:** en la mayoría de los contextos en los que hay que tomar una decisión, las unidades en que están expresados los diferentes criterios suelen ser diferentes. Así, en el ejemplo de los automóviles los criterios están medidos en kilómetros/hora, metros cuadrados, unidades monetarias, etc. En este tipo de situación, una comparación o agregación de los diferentes criterios carece de significado.
- **Valores límite distintos:** la segunda razón es que en muchos problemas multicriterio, los valores alcanzables por los diferentes criterios pueden ser diferentes. En tales casos, sin una normalización previa de los criterios los métodos multicriterio pueden conducirnos a soluciones sesgadas hacia los criterios con valores alcanzables mayores, por ejemplo, la velocidad con respecto al número de pasajeros que caben en el automóvil.
- **Facilidad de interpretación:** por último, en gran parte de los casos, los centros decisores realizan con mayor facilidad las tareas comparativas y de interpretación entre criterios, cuando trabajan con valores normalizados de los mismos en vez de con sus correspondientes valores originales”.

A continuación se exponen brevemente los procedimientos de normalización de criterios más utilizados en la práctica.

Uno de los métodos más simples consiste en dividir los valores que alcanza el criterio por su valor “mejor”. El valor mejor es el máximo cuando el criterio consiste en un atributo del tipo “más es mejor” o el mínimo cuando se trata de un atributo del tipo “menos es mejor” (Romero, 1996). Considérese la tabla 4.1:

Tabla 4.1. Muestra 6 alternativas distintas de opciones en el ejemplo de los automóviles.

ATRIBUTOS			
ALTERNATIVA	VELOCIDAD (km/h)	CAPACIDAD DEL AUTOMOVIL (unidades de personas)	PRECIO (unidades monetarias)
A	1400	4	500
B	1700	5	600
C	1400	7	500
D	1800	8	700
E	1500	2	600
F	1800	6	700

La tabla 4.1 describe varias alternativas involucradas en el proceso de decisión del ejemplo de los automóviles, en base a ella se procederá a normalizar los criterios de

velocidad y precio, columnas 2 y 4 de la tabla anterior, con el método antes descrito como se muestra a continuación.

$$\begin{aligned}
 \text{Velocidad} &= \left(\frac{1400}{1800} = .77, \frac{1700}{1800} = .94, \dots, \frac{1800}{1800} = 1 \right) \\
 \text{Precio} &= \left(\frac{500}{500} = 1, \frac{600}{500} = 1.20, \dots, \frac{700}{500} = 1.4 \right) \quad \dots \quad (1)
 \end{aligned}$$

Tabla 4.2. Muestra los valores normalizados con el método uno donde la velocidad toma como referencia al atributo de mayor valor y el precio al atributo de menor valor.

ATRIBUTOS			
ALTERNATIVA	VELOCIDAD (km/h)	CAPACIDAD DEL AUTOMOVIL (unidades de personas)	PRECIO (unidades monetarias)
A	.77	.5	1
B	.94	.625	1.2
C	.77	.875	1
D	1	1	1.4
E	.83	.25	1.2
F	1	.75	1.4

También pueden normalizarse los criterios, dividiendo los valores que alcanza el criterio por su recorrido, es decir, la diferencia entre el valor “mejor” y el valor “peor” alcanzado por cada criterio (Romero, 1996). Con este procedimiento los valores normalizados de los criterios velocidad y precio pasan a ser:

$$\begin{aligned}
 \text{Velocidad} &= \left(\frac{1400}{1800 - 1400} = 3.5, \frac{1700}{1800 - 1400} = 4.25, \dots, \frac{1800}{1800 - 1400} = 4.5 \right) \\
 \text{Precio} &= \left(\frac{500}{700 - 500} = 2.5, \frac{600}{700 - 500} = 3, \dots, \frac{700}{700 - 500} = 3.5 \right) \quad \dots \quad (2)
 \end{aligned}$$

La tabla 4.3 muestra los resultados del proceso antes descrito.

Tabla 4.3. Muestra los valores normalizados con el método antes mencionado.

ATRIBUTOS			
ALTERNATIVA	VELOCIDAD (km/h)	CAPACIDAD DEL AUTOMOVIL (unidades de personas)	PRECIO (unidades monetarias)
A	3.5	.66	2.5
B	4.25	.83	3
C	3.5	1.16	2.5
D	4.25	1.33	3.5
E	3.75	.33	3
F	4.5	1	3.5

En algunos métodos multicriterio resulta conveniente que los valores normalizados de los criterios queden acotados en el intervalo [0,1]. Este tipo de normalización puede conseguirse con facilidad restando al “mejor” valor el que realmente alcanza el criterio, dividiendo seguidamente dicha diferencia por el correspondiente rango (Romero, 1996). Operando de esta forma los valores normalizados de los criterios velocidad y precio son:

$$Velocidad = \left(\frac{1800 - 1400}{1800 - 1400} = 1, \frac{1800 - 1700}{1800 - 1400} = 0.25, \dots, \frac{1800 - 1800}{1800 - 1400} = 0 \right)$$

$$Precio = \left(\frac{500 - 500}{700 - 500} = 0, \frac{600 - 500}{700 - 500} = 0.5, \dots, \frac{700 - 500}{700 - 500} = 1 \right) \quad \dots \quad (3)$$

La tabla 4.4 muestra los resultados de la normalización llevada a cabo con el método descrito anteriormente.

Tabla 4.4. Muestra los valores normalizados con el método antes mencionado.

ATRIBUTOS			
ALTERNATIVA	VELOCIDAD (km/h)	CAPACIDAD DEL AUTOMOVIL (unidades de personas)	PRECIO (unidades monetarias)
A	1	.66	0
B	.25	.5	.5
C	1	.16	0
D	0	0	1
E	.75	1	.5
F	0	.33	1

Conviene indicar que con este sistema de normalización el valor normalizado del criterio es 0 cuando el criterio alcanza su “mejor” valor y por el contrario es 1 cuando el

criterio alcanza su “peor” valor (Romero, 1996). El segundo aspecto importante para tomar una decisión es la ponderación preferencial de los criterios.

El método consiste en pedir al centro decisor que clasifique los criterios por orden de importancia. Es decir, si tenemos n criterios se solicita al centro decisor que asigne el número 1 al criterio que considere más importante, el número 2 al criterio siguiente en importancia hasta asignar el número n al criterio que considera menos importante. Los pesos compatibles con dicha información pueden obtenerse a partir de la siguiente expresión:

$$W_i = \frac{\frac{1}{r_i}}{\sum_{j=1}^n \frac{1}{r_j}}$$

Donde r_j es el lugar o posición que ocupa el criterio j -ésimo en la clasificación establecida por el centro decisor (Romero, 1996).

Ya que se tienen determinados los pesos para cada criterio se procede a obtener el conjunto eficiente de soluciones. También existen diversos métodos para generar dicho conjunto, pero para este proyecto se utilizara el método de las *ponderaciones* que se describe a continuación.

El método de las ponderaciones consiste en multiplicar a cada objetivo por un peso o factor no negativo, procediendo a agregar todos los objetivos ponderados en una única función objetivo. La optimización de dicha función, ponderada y agregada, genera un elemento del conjunto eficiente, es decir, del conjunto de soluciones de Pareto. Por medio de la parametrización de los pesos asociados a los objetivos se va aproximando el conjunto eficiente (Romero, 1996).

Así todo lo anterior da la pauta para poder tomar una decisión al respecto de nuestro problema. Lo antes expuesto se verá ejemplificado conforme se avance en la definición del proceso de toma de decisiones para la elaboración de horarios de clases a partir de evaluaciones docentes, pues esto será parte de la implementación del sistema que se pretende elaborar.

CAPÍTULO 5 Construcción de la base de datos

En este capítulo se presentan los aspectos relacionados con la construcción de la base de datos del SATDEH. El capítulo aborda los supuestos del diseño y todo lo relacionado con la construcción, así como la versión final del modelo utilizado, representado mediante diagramas de Clases y Entidad/Relación, y finalmente el código generado para la implementación de ésta.

5.1 Supuestos en el diseño

A continuación se menciona el listado de los supuestos en el diseño de la base de datos del sistema de apoyo a la toma de decisiones de los alumnos para elaborar horarios de clases a partir de evaluaciones docentes.

- Se asume que la base de datos estará dirigida a una entidad educativa de nivel superior.
- Cada profesor debe contar con un número de identificación único dentro de la entidad educativa.
- El profesor puede laborar o no de tiempo completo dentro de la entidad educativa.
- Un profesor puede estar asociado a varios grupos y un grupo solo puede estar asociado a un solo profesor titular.
- Un profesor adjunto puede estar asociado a muchos grupos.
- El número de grupo debe ser único.
- Cada grupo debe tener asociado un horario y salón de clases.
- Cada materia impartida dentro de la entidad debe tener como características el semestre en que se imparte, una clave única de identificación y el número de créditos.
- Una materia puede estar asociada a muchos grupos pero un grupo solo puede estar asociado a una materia.
- Cada alumno debe contar con un número de identificación dentro de la entidad educativa, éste debe ser único para cada alumno.
- Un alumno puede estar inscrito en muchos grupos y un grupo puede tener a muchos alumnos inscritos.
- Un alumno no puede estar inscrito más de una vez en un mismo grupo
- Si un alumno o profesor se va de la entidad se asume que puede regresar a la entidad nuevamente en un futuro.
- Es un requisito para poder realizar comentarios sobre alguno de los grupos, ser alumno de la entidad educativa y haber estado inscrito en dicho grupo en semestres anteriores.
- Deberá especificarse a qué carrera y a qué generación pertenece cada alumno.
- Para fines de la evaluación de profesores se contará con una encuesta previamente definida.

5.2 Opinión de los alumnos como base de los supuestos

Como se ha mencionado en los supuestos del apartado anterior, mediante la aplicación de una encuesta se conoció qué es lo que generalmente un alumno busca de un profesor. La encuesta fue realizada en línea a un grupo de 87 alumnos de la Facultad de Ciencias de la UNAM, dicha encuesta estuvo integrada por los criterios que un alumno evaluaría de un profesor, tales criterios fueron seleccionados según lo propuesto en el trabajo de (Valenzuela, 2002). El listado de criterios original se presenta a continuación:

Lista de Reactivos

- R01. Inicia clase a la hora
 - R02. Usa materiales para exponer
 - R03. Termina la clase a la hora
 - R04. Propicia participación de alumnos
 - R05. Las clases siguen el orden del programa
 - R06. Comenta perspectivas sobre la profesión
 - R07. Deja ideas claras, no se enreda
 - R08. Al iniciar la clase presenta el objetivo de ésta
 - R09. Responde lo que se le pregunta
 - R10. Escucha sin interrumpir al alumno
 - R11. Mantiene disciplina sin rigidez
 - R12. Reconoce esfuerzo y logros
 - R13. Organiza equipos de trabajo
 - R14. Trata con respeto y cordialidad
 - R15. Forma equipos de investigación
 - R16. Ejemplifica lo aprendido en clase en el campo de la carrera
 - R17. Relaciona el tema con la siguiente clase
 - R18. Los temas tratados en clase corresponden con los evaluados en tareas y exámenes
 - R19. Responde concretamente
 - R20. Mantiene en actividad al grupo
-

-
- R21. Señala errores en la actividad profesional
 - R22. Aclara terminología
 - R23. Relaciona el tema con aplicaciones
 - R24. Cita fuentes bibliográficas al responder
 - R25. Entrega calificaciones el día prometido
 - R26. Acepta aportaciones de alumnos a la clase
 - R27. Utiliza ejemplos ilustrativos
 - R28. Al inicio del curso acuerda forma de evaluar
 - R29. Realiza evaluaciones periódicas
 - R30. Logra que los alumnos se interesen en la materia
 - R31. Trata todos los temas del programa
 - R32. Comunica con claridad sus conocimientos
 - R33. Propicia una actitud reflexiva en los alumnos
 - R34. Domina el tema de la materia que imparte
 - R35. Asiste a sus clases
 - R36. Aclara dudas fuera del horario de clases
 - R37. Deja tareas
 - R38. El nivel de los exámenes corresponde con el de las clases
 - R39. Integra Teoría y práctica
 - R40. Dificultad del curso
-

La encuesta realizada consistió en preguntar a cada alumno si consideraría o no cada uno de los reactivos anteriores para la elección de un profesor. Adicionalmente se le dio la opción de incluir cualquier otro criterio que no se encontrara entre los criterios expuestos pero que el alumno considerara importante para tomar su decisión.

Dicha encuesta se construyó utilizando la herramienta LimeSurvey (LimeSurvey, 2013). En la Figura 5.1 se muestra una vista de la misma.

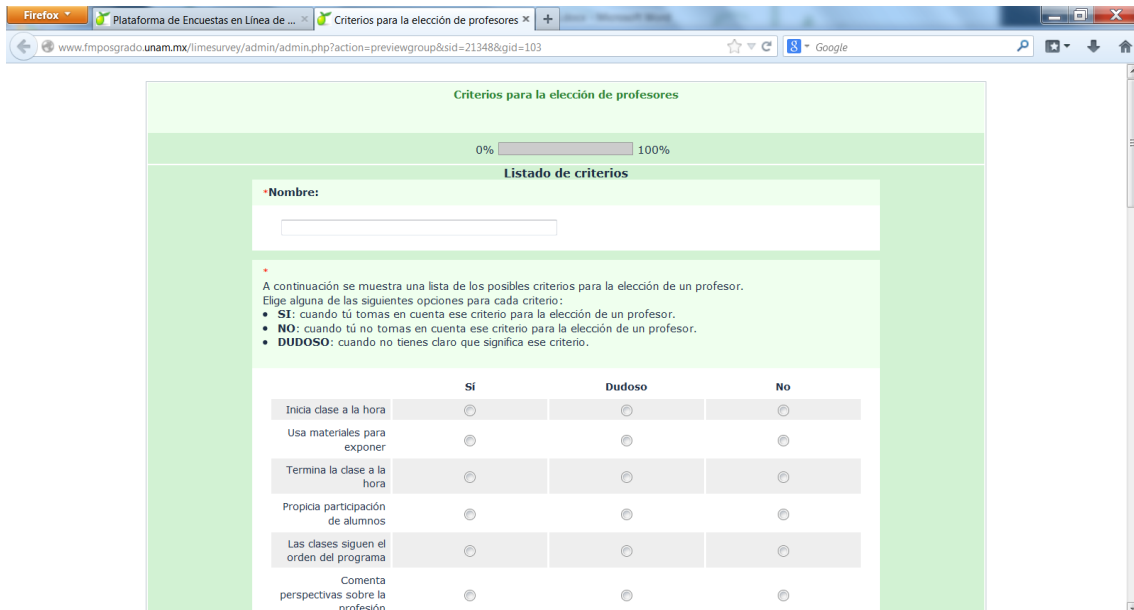


Figura 5.1. Captura de pantalla que muestra la herramienta LimeSurvey.

A continuación, a manera de ejemplo, en la Figura 5.2 se muestra la gráfica resultante del criterio “Inicia clase a la hora”, donde el 68% de los alumnos respondieron que **SÍ** tomarían en cuenta este criterio para la elección de un profesor, 25% **NO** lo tomaría en cuenta y para el 7% es **DUDOSO**, en el Anexo A pueden consultarse las gráficas de la totalidad de los criterios.



Figura 5.2. Resultados en porcentajes del criterio “Inicia clase a la hora”.

Enseguida se presenta una clasificación de los criterios anteriores según (Valenzuela, 2002).

Categorías del desempeño docente frente a grupo	Reactivos seleccionados como preguntas en la encuesta
(D) Dinamismo en clase	4, 13, 15, 20, 26, 33
(P) Preparación del tema de clase	2, 17, 22, 24, 27
(C) Criterios para calificar	18, 25, 28, 29, 37, 38, 40
(A) Puntualidad y asistencia	1, 3, 35
(R) Respeto a los alumnos	10, 14
(O) Dominio de la materia	7, 9, 32, 34
(M) Motivación al estudiante	8, 12, 30, 36
(E) Carácter y Estilo Interpersonal	11, 19
(N) Enfoque a la profesión	6, 16, 21, 23, 39
(L) Cumplimiento del programa	5, 31

La clasificación anterior fue utilizada para asegurar que la encuesta contuviera al menos un criterio de cada uno de los distintos aspectos del desempeño docente de un profesor.

Una vez que se realizó la encuesta a los alumnos, quienes pertenecían a una población entre el 5º, 6º y 8º semestre de las carreras de Actuaría y Matemáticas, todos ellos estudiantes de la Facultad de Ciencias de la UNAM, se obtuvieron los resultados mostrados en la Tabla 5.1.

Tabla 5.1. Resultados de la encuesta por criterio y los respectivos porcentajes en cada respuesta.

número de Criterio	Criterio	%Dudoso	%No	%Si
1	Inicia clase a la hora	7%	25%	68%
2	Usa materiales para exponer	5%	42%	53%
3	Propicia participación de alumno	8%	25%	67%
4	Las clases siguen el orden del programa	13%	23%	64%
5	Comenta perspectivas sobre la profesión	8%	15%	77%
6	Deja ideas claras, no se enreda	4%	1%	95%
7	Al iniciar la clase presenta el objetivo de ésta	18%	43%	39%
8	Responde lo que se le pregunta	2%	4%	94%
9	Escucha sin interrumpir al alumno	7%	15%	78%
10	Mantiene disciplina sin rigidez	13%	26%	61%
11	Reconoce esfuerzo y logros	8%	20%	72%
12	Organiza equipos de trabajo	8%	44%	48%
13	Trata con respeto y cordialidad	4%	9%	87%
14	Forma equipos de investigación	18%	45%	35%
15	Ejemplifica lo aprendido en clase en el campo de la carrera	8%	15%	77%
16	Relaciona el tema con la siguiente clase	11%	28%	61%
17	Los temas tratados en clase corresponden con los evaluados en tareas y exámenes	0%	0%	100%
18	Responde concretamente	4%	3%	93%
19	Mantiene en actividad al grupo	8%	16%	76%
20	Señala errores en la actividad profesional	14%	32%	54%
21	Aclara terminología	10%	15%	74%
22	Relaciona el tema con aplicaciones	3%	12%	85%
23	Cita fuentes bibliográficas al responder	15%	53%	32%
24	Entrega calificaciones el día prometido	9%	28%	63%
25	Acepta aportaciones de alumnos a la clase	1%	10%	89%
26	Utiliza ejemplos ilustrativos	1%	10%	89%
27	Al inicio del curso acuerda forma de evaluar	2%	5%	93%
28	Realiza evaluaciones periódicas	9%	18%	73%
29	Logra que los alumnos se interesen en la materia	2%	7%	91%
30	Trata todos los temas del programa	4%	24%	72%
31	Comunica con claridad sus conocimientos	1%	1%	98%
32	Propicia una actitud reflexiva en los alumnos	7%	25%	68%
33	Domina el tema de la materia que imparte	0%	0%	100%
34	Asiste a sus clases	0%	1%	99%
35	Aclara dudas fuera del horario de clases	7%	17%	76%
36	Deja tareas	9%	20%	71%
37	El nivel de los exámenes corresponde con el de las clases	9%	4%	87%
38	Integra Teoría y práctica	0%	3%	97%
39	Dificultad del curso	8%	25%	67%
40	¿Considerarías algún otro criterio para la elección de tus profesores que no se encontraba en la lista?	0%	89%	11%

A continuación se muestra una lista de los criterios que fueron elegidos por al menos el 70% de los alumnos que respondieron la encuesta.

Lista de Reactivos

- R06. Comenta perspectivas sobre la profesión
 - R07. Deja ideas claras, no se enreda
 - R09. Responde lo que se le pregunta
 - R10. Escucha sin interrumpir al alumno
 - R12. Reconoce esfuerzo y logros
 - R14. Trata con respeto y cordialidad
 - R16. Ejemplifica lo aprendido en clase en el campo de la carrera
 - R18. Los temas tratados en clase corresponden con los evaluados en tareas y exámenes
 - R19. Responde concretamente
 - R20. Mantiene en actividad al grupo
 - R22. Aclara terminología
 - R23. Relaciona el tema con aplicaciones
 - R26. Acepta aportaciones de alumnos a la clase
 - R27. Utiliza ejemplos ilustrativos
 - R28. Al inicio del curso acuerda forma de evaluar
 - R29. Realiza evaluaciones periódicas
 - R30. Logra que los alumnos se interesen en la materia
 - R31. Trata todos los temas del programa
 - R32. Comunica con claridad sus conocimientos
 - R34. Domina el tema de la materia que imparte
 - R35. Asiste a sus clases
 - R36. Aclara dudas fuera del horario de clases
 - R37. Deja tareas
 - R38. El nivel de los exámenes corresponde con el de las clases
 - R39. Integra Teoría y práctica
-

Enseguida se presenta la clasificación de los criterios que fueron elegidos por los alumnos según el porcentaje del 70% de aceptación en general.

Categorías del desempeño docente frente a grupo	Reactivos que quedaron después de la encuesta
(D) Dinamismo en clase	20, 26
(P) Preparación del tema de clase	22, 27
(C) Criterios para calificar	18, 28, 29, 37, 38
(A) Puntualidad y asistencia	35
(R) Respeto a los alumnos	10, 14
(O) Dominio de la materia	7, 9, 32, 34
(M) Motivación al estudiante	12, 30, 36
(E) Carácter y Estilo Interpersonal	19
(N) Enfoque a la profesión	6, 16, 23, 39
(L) Cumplimiento del programa	31

El proceso anterior arrojó como resultado los 25 criterios, de 40 iniciales, que serán utilizados en el sistema de apoyo a la toma de decisiones que se pretende construir en este trabajo.

5.3 Diagrama Entidad\Relación del diseño de la base de datos

Para modelar la base de datos siguiendo el Modelo Relacional y basados en los supuestos anteriormente descritos, se construyó, como primer paso, un diagrama Entidad-Relación que representara la solución propuesta. La Figura 5.3 muestra el diagrama Entidad-Relación que modela la base de datos del SATDEH.

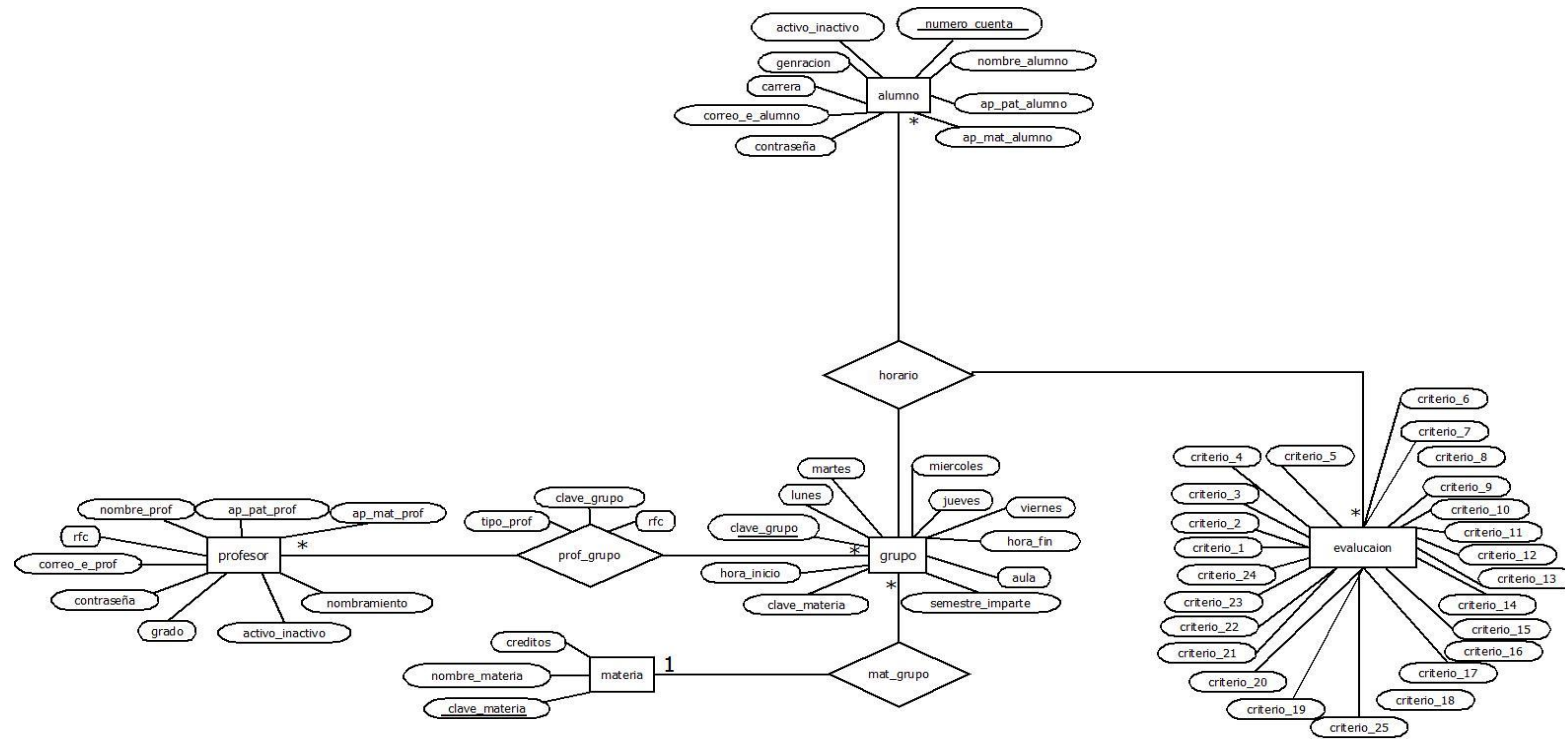


Figura 5.3. Diagrama entidad\relación de la base de datos del sistema.

En el diagrama anterior se han sustituido los nombres originales de los 25 criterios por “criterio n” n = 1,..., 25 y cada uno representa a los criterios seleccionados después de analizar los datos arrojados por la encuesta.

5.4 Diagrama de Clases UML del diseño de la base de datos

De acuerdo al diagrama Entidad-Relación mostrado anteriormente se construyó un diagrama de Clases UML que representará a cada una de las tablas, las asociaciones entre éstas y los atributos de la base de datos. La Figura 5.4 muestra el diagrama de Clases UML de la base de datos del sistema.

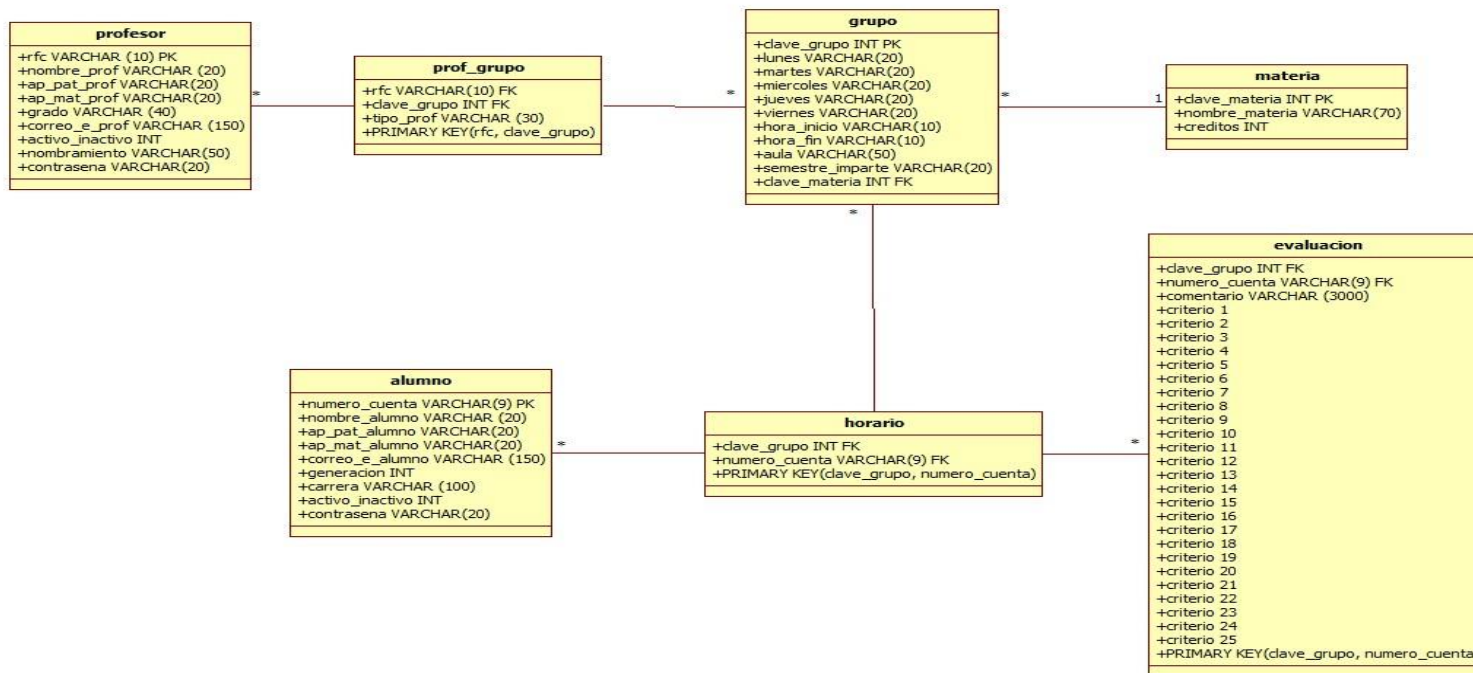


Figura 5.4. Diagrama de clases UML de la base de datos del sistema.

De igual forma, en el diagrama anterior se han sustituido los nombres originales de los 25 criterios por “criterio n” n = 1,..., 25 y cada uno representa uno de los criterios seleccionados después de la encuesta.

5.5 Entidades y atributos de la base de datos

Basados en los supuestos presentados, el diseño de la base de datos y los criterios obtenidos, a continuación se muestra una descripción de las entidades relaciones y atributos de la base de datos del SATDEH. En forma de tabla se muestran los elementos más importantes de cada entidad, indicando el nombre del atributo, su tipo de dato y las restricciones que pudiera tener.

profesor: Esta entidad modela a los profesores que imparten cursos dentro de la entidad educativa, ésta tendrá los siguientes atributos, ver Tabla 5.2.

Tabla 5.2. Descripción y conformación de la entidad profesor

Profesor		
Nombre Atributo	Tipo	Restricción
rfc	varchar(10)	<i>primary key</i>
Descripción Identificador de cada profesor		
nombre_prof	varchar (20)	
Descripción Almacena el nombre de cada profesor		
ap_pat_prof	varchar (20)	
Descripción Almacena el apellido paterno de cada profesor		
ap_mat_prof	varchar (20)	
Descripción Almacena el apellido materno de cada profesor		
correo_e_prof	varchar(150)	
Descripción Almacena el correo electrónico de cada profesor		
grado	varchar (40)	
Descripción Almacena el grado académico de cada profesor		
nombramiento	varchar(50)	
Descripción Representa el nombramiento que cada profesor tiene dentro de la entidad educativa		
activo_inactivo	int	<i>{0,1}</i>
Descripción Toma el valor 1 si el profesor se encuentra impartiendo algún curso dentro determinado ciclo escolar, y 0 se sucede lo contrario. Ningún otro valor es válido.		
contrasena	varchar(20)	<i>not null</i>
Descripción Almacena la contraseña que tendrá cada profesor para ingresar al sistema y es necesaria siempre.		

grupo: Esta entidad modela a los grupos disponibles en la entidad educativa, se encarga de almacenar toda aquella información correspondiente a dichos grupos, tales como: la hora, semestre y días a la semana en que se impartirá una materia. A continuación se muestran los atributos que la conforman, ver Tabla 5.3.

Tabla 5.3. Descripción y conformación de la entidad grupo

grupo		
Nombre Atributo	Tipo	Restricción
clave_grupo	Int	<i>primary key</i>
Descripción Asegura que no se repitan los grupos		
lunes	varchar(20)	
Descripción Almacena si en un grupo se imparte clase ese día		
martes	varchar(20)	
Descripción Almacena si en un grupo se imparte clase ese día		
miercoles	varchar(20)	
Descripción Almacena si en un grupo se imparte clase ese día		
jueves	varchar(20)	
Descripción Almacena si en un grupo se imparte clase ese día		
viernes	varchar(20)	
Descripción Almacena si en un grupo se imparte clase ese día		
hora_inicio	varchar(10)	
Descripción Es el encargado de almacenar la hora en que inicia la clase en un grupo.		
hora_fin	varchar(10)	
Descripción Almacena la hora en que finaliza la clase en un grupo.		
semestre_imparte	varchar (20)	<i>not null</i>
Descripción Recopila el semestre en que se imparte un curso.		
aula	varchar(50)	<i>not null</i>
Descripción Almacena el salón donde se impartirán los cursos.		
clave_materia	Int	<i>foreign key</i>
Descripción Almacena la clave de la materia a la que se le asocia un grupo y es la llave primaria en la entidad materia		

materia: Esta entidad modela la información referente a las asignaturas impartidas en la entidad educativa, ver Tabla 5.4.

Tabla 5.4. Descripción y conformación de la entidad materia

materia		
Nombre Atributo	Tipo	Restricción
clave_materia	int	<i>primary key</i>
Descripción Permite almacenar la clave de la materia.		
nombre_materia	varchar(70)	<i>not null</i>
Descripción Representa el nombre de la asignatura		
creditos	int	
Descripción Representa el peso en créditos que tiene cada materia		

alumno: Esta entidad modela y almacena la información requerida por la entidad educativa de los alumnos, la cual posee las siguientes características, ver Tabla 5.5.

Tabla 5.5. Descripción y conformación de la entidad alumno

alumno		
Nombre Atributo	Tipo	Restricción
numero_cuenta	varchar (9)	<i>primary key</i>
Descripción Identificador de cada alumno.		
nombre_alumno	varchar (20)	
Descripción Almacena el nombre de cada alumno.		
ap_pat_alumno	varchar (20)	
Descripción Almacena el apellido paterno de cada alumno.		
ap_mat_alumno	varchar(20)	
Descripción Almacena el apellido materno de cada alumno.		
correo_e_alumno	varchar(150)	
Descripción Almacena el correo electrónico de cada alumno.		
generacion	int	
Descripción Almacena el dato de la generación a la que pertenece cada alumno de la entidad educativa.		

carrera	varchar (100)	
Descripción		
Almacena el nombre del o las carreras a las que un alumno puede pertenecer.		
activo_inactivo	int	{0,1}
Descripción		
Permite identificar a los alumnos que se encuentran tomando cursos en la entidad educativa en determinado ciclo escolar, o bien si el alumno ha sido dado de baja de la entidad educativa. Solo toma los valores 0 o 1, 1 para activo y 0 para inactivo.		
contrasena	varchar(20)	<i>not null</i>
Descripción		
Almacena el código confidencial de cada alumno que necesita para ingresar a los servicios del sistema		

horario: Esta relación permite asociar a las entidades evaluación, alumno y grupo, para poder tener información de los alumnos inscritos en los diferentes grupos que hay en la entidad educativa, así como para ver las evaluaciones hechas a los grupos. La relación tiene los siguientes atributos además de las respectivas llaves primarias de las entidades asociadas, ver Tabla 5.6.

Tabla 5.6. Descripción y conformación de la relación mat_carrera

horario		
Nombre Atributo	Tipo	Restricción
clave_grupo	int	<i>foreign key</i>
Descripción		
Almacena la información correspondiente a la clave del grupo.		
numero_cuenta	varchar (9)	<i>foreign key</i>
Descripción		
Almacena el número de identificación de cada alumno.		

prof_grupo: Esta relación se encarga de asociar a la entidad profesor y a la entidad grupo, para conocer información, por ejemplo, del nombre del profesor que da clases en cierto grupo. Esta relación tiene como atributos a las llaves primarias de las respectivas entidades asociadas y al atributo tipo, ver Tabla 5.7.

Tabla 5.7. Descripción y conformación de la relación prof_grupo

prof_grupo		
Nombre Atributo	Tipo	Restricción
rfc	varchar (10)	<i>foreign key</i>
Descripción Identificador de cada profesor.		
clave_grupo	Int	<i>foreign key</i>
Descripción Identificador de cada grupo.		
tipo_prof	varchar (30)	<i>not null</i>
Descripción Almacena el tipo de participación en el curso del profesor, esto puede ser como titular o adjunto.		

evaluacion: Esta entidad contiene los criterios de evaluación de los profesores y los comentarios realizados por los alumnos hacia sus profesores. En esta entidad se encuentran los 25 criterios de evaluación y el atributo donde el alumno escribirá sus comentarios mencionados en la sección “Opinión de los alumnos como base de nuestros supuestos”, sus atributos además de las respectivas llaves primarias de dichas entidades se pueden ver en la Tabla 5.8.

Tabla 5.8. Descripción y conformación de la tabla evaluacion

evaluacion		
Nombre Atributo	Tipo	Restricción
clave_grupo	int	<i>foreign key</i>
Descripción Identificador de cada grupo		
numero_cuenta	varchar (9)	<i>foreign key</i>
Descripción Identificador de cada alumno.		
comentario	varchar(3000)	
Descripción Almacena los comentarios que un alumno haga sobre algún profesor.		
comenta_perspectivas	real	<i>{01,2,3,4,5}</i>
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor comenta perspectivas en clase.		

Tabla 5.8. Descripción y conformación de la tabla evaluación (continuación).

evaluación		
Nombre Atributo	Tipo	Restricción
deja_ideas	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor deja ideas claras en clase.		
responde_loque	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor responde lo que se le pregunta o no en clase.		
escucha	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor escucha al alumno en clase.		
reconoce	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor reconoce el esfuerzo y los logros del alumno en clase.		
trata_respeto	real	{01,2,3,4,5}
Descripción almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor trata con respeto al alumno en clase.		
ejemplifica	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor ejemplifica las ideas de la clase.		
temas_tratados	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor trata todos los temas de la clase.		

Tabla 5.8. Descripción y conformación de la tabla evaluación (continuación).

evaluación		
Nombre Atributo	Tipo	Restricción
responde_concretamente	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor responde concretamente a las preguntas que se le realizan en clase.		
mantiene_actividad	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor mantiene en actividad al grupo durante la clase.		
aclara_terminologia	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor escucha al alumno en clase.		
relaciona_tema_aplica	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor relaciona los temas vistos en clase con las aplicaciones de la vida real.		
acepta_aportaciones	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor acepta las aportaciones de los alumnos en la clase.		
utiliza_ejemplos	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor utiliza ejemplos ilustrativos en la clase.		
acuerda_forma_evaluar	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor acuerda la forma de evaluar desde el inicio del curso.		
evaluaciones_periodicas	real	{01,2,3,4,5}
Descripción Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor realiza evaluaciones periódicas durante todo el curso.		

Tabla 5.8. Descripción y conformación de la tabla evaluación (continuación).

Evaluación		
Nombre Atributo	Tipo	Restricción
alumnos_interesen	real	{01,2,3,4,5}
Descripción		
Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor logra que los alumnos se interesen por la materia.		
todos_temas_programa	real	{01,2,3,4,5}
Descripción		
Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor cubre todos los temas establecidos en el programa de estudio.		
comunica_claridad_conocimientos	real	{01,2,3,4,5}
Descripción		
Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor comunica al grupo sus conocimientos de una manera clara.		
domina_tema	real	{01,2,3,4,5}
Descripción		
Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor domina el tema de la materia que está impartiendo.		
asiste_clases	real	{01,2,3,4,5}
Descripción		
Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor asiste con regularidad a sus clases.		
aclara_dudas	real	{01,2,3,4,5}
Descripción		
Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor aclara dudas fuera del salón de clases.		
deja_tareas	real	{01,2,3,4,5}
Descripción		
Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor deja tareas durante todo el curso.		
integra_teoría_practica	real	{01,2,3,4,5}
Descripción		
Almacena una calificación del 1 al 5 respecto al criterio de evaluación de un profesor sobre si el profesor integra teoría y práctica durante las clases.		

5.6 Construcción de la base de datos y de los procedimientos almacenados (SP)

La construcción de la base de datos se realizó utilizando el sistema manejador de bases de datos PostgreSQL 9.1 (Postgresql, 2013). A continuación se presenta el código para crear la base de datos del sistema a partir del diagrama de Clases presentado en la Figura 5.4.

```
CREATE TABLE profesor (  
    rfc VARCHAR(10) PRIMARY KEY,  
    nombre_prof VARCHAR (20),  
    ap_pat_prof VARCHAR(20),  
    ap_mat_prof VARCHAR(20),  
    grado VARCHAR (40),  
    correo_e_prof VARCHAR (150),  
    activo_inactivo INT CHECK(activo_inactivo=0 OR activo_inactivo=1),  
    nombramiento VARCHAR(50),  
    contrasena VARCHAR(20)  
);  
  
CREATE TABLE alumno (  
    numero_cuenta VARCHAR(9) PRIMARY KEY,  
    nombre_alumno VARCHAR (20),  
    ap_pat_alumno VARCHAR(20),  
    ap_mat_alumno VARCHAR(20),  
    correo_e_alumno VARCHAR (150),  
    generacion INT,  
    carrera VARCHAR (100),  
    activo_inactivo INT CHECK(activo_inactivo=0 OR activo_inactivo=1),  
    contrasena VARCHAR(20)  
);  
  
CREATE TABLE materia (  
    clave_materia INT PRIMARY KEY,  
    nombre_materia VARCHAR(70),  
    creditos INT  
);  
  
CREATE TABLE grupo (  
    clave_grupo INT PRIMARY KEY,  
    lunes VARCHAR(20),  
    martes VARCHAR(20),  
    miercoles VARCHAR(20),  
    jueves VARCHAR(20),  
    viernes VARCHAR(20),  
    hora_inicio VARCHAR(10),  
    hora_fin VARCHAR(10),  
    aula VARCHAR(50),  
    semestre_imparte VARCHAR(20),  
    clave_materia INT NOT NULL,  
    FOREIGN KEY(clave_materia) REFERENCES materia(clave_materia)  
    MATCH simple ON DELETE CASCADE ON UPDATE CASCADE  
);  
  
CREATE TABLE prof_grupo (  
    rfc VARCHAR(10) NOT NULL,  
    clave_grupo INT NOT NULL,  
    tipo_prof VARCHAR (30),  
    PRIMARY KEY(rfc, clave_grupo),  
    FOREIGN KEY(rfc) REFERENCES profesor(rfc)  
    MATCH simple ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(clave_grupo) REFERENCES grupo(clave_grupo)  
    MATCH simple ON DELETE CASCADE ON UPDATE CASCADE  
);  
  
CREATE TABLE horario (  
    clave_grupo INT NOT NULL,  
    numero_cuenta VARCHAR(9) NOT NULL,  
    PRIMARY KEY(clave_grupo, numero_cuenta),  
    FOREIGN KEY(clave_grupo) REFERENCES grupo(clave_grupo)  
    MATCH simple ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(numero_cuenta) REFERENCES alumno(numero_cuenta)  
    MATCH simple ON DELETE CASCADE ON UPDATE CASCADE
```

```

);

CREATE TABLE evaluacion (
  clave_grupo INT NOT NULL,
  numero_cuenta VARCHAR(9) NOT NULL,
  PRIMARY KEY(clave_grupo, numero_cuenta),
  comentario VARCHAR (3000),
  comenta_perspectivas REAL,
  deja_ideas REAL,
  responde_loque REAL,
  escucha REAL,
  reconoce REAL,
  trata_respeto REAL,
  ejemplifica REAL,
  temas_tratados REAL,
  responde_concretamente REAL,
  mantiene_actividad REAL,
  aclara_terminologia REAL,
  relaciona_tema_aplica REAL,
  acepta_aportaciones REAL,
  utiliza_ejemplos REAL,
  acuerda_forma_evaluar REAL,
  evaluaciones_periodicas REAL,
  alumnos_interesen REAL,
  todos_temas_programa REAL,
  comunica_claridad_conocimientos REAL,
  domina_tema REAL,
  asiste_clases REAL,
  aclara_dudas REAL,
  deja_tareas REAL,
  nivel_exámenes_clases REAL,
  integra_teoría_práctica REAL,
  FOREIGN KEY(clave_grupo, numero_cuenta) REFERENCES horario(clave_grupo,
numero_cuenta)
MATCH simple ON DELETE CASCADE ON UPDATE CASCADE
);

```

5.7 Procedimientos Almacenados

Con el objetivo de proveer de un mecanismo particular de explotación y manipulación de la base de datos construida, se utilizaron Procedimientos Almacenados (SPs por sus siglas en inglés). Los SPs son funciones definidas por el usuario dentro del SMBD, que son utilizadas para facilitar el manejo de la información de una base de datos o bien para obtener información específica de la misma. Dichas funciones tienen una estructura similar a la de cualquier función en un lenguaje de programación. Éstos pueden recibir o no parámetros, utilizar ciclos o variables, sin embargo su principal entorno de trabajo son las tablas que contiene una base de datos.

En este trabajo la solución planteada a las necesidades del negocio identificadas, se ha concentrado en la base de datos y la herramienta que permite dar dicha solución son precisamente los SPs. Los SPs serán los encargados de generar el ranking que el sistema ofrece a los alumnos. No obstante los SPs son indispensables para llevar a cabo la administración de la base de datos de una manera eficiente y práctica para el administrador.

A continuación se presenta un listado de los SPs que se requieren para este sistema y algunos de sus códigos en PostgreSQL 9.1, el resto de los SPs se pueden encontrar en el **Anexo B**:

1. **SP de Inserción:** este tipo de procedimientos almacenados realizarán las inserciones de datos en las tablas de la BD así como en las relaciones si éstas estuvieran asociadas.

- Función que inserta un nuevo profesor en la tabla "profesor" verificando si el profesor ya existe y regresando 1 si la inserción fue exitosa y 0 si no fue así.

```

CREATE OR REPLACE FUNCTION sp_inserta_Profesor(
    rfc1 VARCHAR(10),
    nombre_prof1 VARCHAR (20),
    ap_pat_prof1 VARCHAR(20),
    ap_mat_prof1 VARCHAR(20),
    gradol VARCHAR (40),
    correo_e_prof1 VARCHAR (200),
    activo_inactivo1 int,
    nombramientol VARCHAR(50),
    contrasenal VARCHAR(20)
)
RETURNS INTEGER
AS
$$
DECLARE consulta_id INTEGER;
BEGIN

consulta_id= (SELECT rfc FROM profesor WHERE rfc=rfc1);

IF (consulta_id IS NULL) THEN
    INSERT INTO profesor VALUES
(rfc1,nombre_prof1,ap_pat_prof1,ap_mat_prof1,gradol,correo_e_prof1,activo_inactivo1,nombramientol,contrasenal);
    RETURN 1;
ELSE RETURN 0;
END IF;

END;
$$
LANGUAGE 'plpgsql';

IF (consulta_id IS NULL) THEN
    INSERT INTO grupo VALUES
(clave_grupo1,lunes1,martes1,miercoles1,jueves1,viernes1,hora_inicio1,hora_fin1,aula1,semestre_imparte1,clave_material);
    RETURN 1;
ELSE RETURN 0;
END IF;

END;
$$
LANGUAGE 'plpgsql';

```


- Función que asocia a un profesor con un grupo e inserta los "rfc", "clave_grupo" y el "tipo_prof" en la relación "prof_grupo" verificando que el profesor y el grupo existan en sus tablas respectivas y que no sea un registro ya existente, regresando 1 si la inserción fue exitosa y 0 si no fue así.

```

CREATE OR REPLACE FUNCTION sp_asocia_Profesor_Grupo(
    rfc1 VARCHAR(10),
    clave_grupo1 INTEGER,
    tipo_prof1 VARCHAR(200)
)
RETURNS INTEGER
AS
$$
DECLARE consulta_id_grupo INTEGER;
DECLARE consulta_id_profesor VARCHAR(10);
DECLARE consulta_id INTEGER;
BEGIN

consulta_id_grupo=(SELECT clave_grupo FROM grupo WHERE
clave_grupo=clave_grupo1);
consulta_id_profesor=(SELECT rfc FROM profesor WHERE rfc=rfc1);
consulta_id=(SELECT clave_grupo FROM prof_grupo WHERE rfc=rfc1 AND
clave_grupo=clave_grupo1);

IF (CAST(consulta_id_profesor AS TEXT)='') THEN
    RETURN 0;
END IF;
IF (CAST(consulta_id_grupo AS TEXT)='') THEN
    RETURN 0;
END IF;

IF (consulta_id IS NULL) THEN
    INSERT INTO prof_grupo VALUES (rfc1,clave_grupo1,tipo_prof1);
    RETURN 1;
END IF;

END;
$$
LANGUAGE 'plpgsql';

```

2. **SP de Actualización:** este tipo de procedimientos almacenados deberá realizar la actualización de datos en las tablas de la BD, así como en las relaciones si éstas estuvieran asociadas, es decir, debe hacerlo en cascada⁷ sobre todas las tablas asociadas.

- Función que Actualiza el registro de un profesor ya existente en la tabla "profesor" regresando una tupla que muestra la actualización.

```
CREATE OR REPLACE FUNCTION sp_Actualiza_Profesor(  
    rfc1 VARCHAR(10),  
    nombre_prof1 VARCHAR(20),  
    ap_pat_prof1 VARCHAR(20),  
    ap_mat_prof1 VARCHAR(20),  
    grad1 VARCHAR(40),  
    correo_e_prof1 VARCHAR(200),  
    activo_inactivo1 INT,  
    nombramiento1 VARCHAR(50),  
    contrasena1 VARCHAR(20)  
)  
RETURNS SETOF profesor  
AS  
$$  
DECLARE consulta_id INTEGER;  
BEGIN  
  
Update profesor set nombre_prof=nombre_prof1,  
    ap_pat_prof=ap_pat_prof1,  
    ap_mat_prof=ap_mat_prof1,  
    grado=grad1,  
    correo_e_prof=correo_e_prof1,  
    activo_inactivo=activo_inactivo1,  
    nombramiento=nombramiento1,  
    contrasena=contrasena1  
WHERE rfc=rfc1;  
  
RETURN QUERY(SELECT * FROM profesor);  
  
END;  
$$  
LANGUAGE 'plpgsql';
```

⁷ En el código de creación de la base de datos se declaró la eliminación y la actualización en cascada, acción que el mismo SMBD realiza automáticamente, por lo que ya no es necesario considerarlo para estos SP.

3. **SP de Eliminación:** este tipo de procedimientos almacenados deberá realizar un borrado lógico de registros en las tablas de la BD así como en las relaciones si éstas estuvieran asociadas, es decir, debe hacerlo en cascada sobre todas las tablas asociadas.

- Función que activa o desactiva a un profesor en la tabla "profesor" buscando el rfc del profesor que se quiere dar de baja o alta y regresando 1 si se tuvo éxito y 0 si no fue así.

```
CREATE OR REPLACE FUNCTION sp_Alta_Baja_Profesor(
    rfc1 VARCHAR (10)
)
RETURNS INTEGER
AS
$$
DECLARE consulta INTEGER;
BEGIN
consulta= (SELECT activo_inactivo FROM profesor WHERE rfc=rfc1);
IF (consulta =1) THEN
UPDATE profesor SET activo_inactivo=0
WHERE rfc=rfc1;
RETURN 1;
END IF;
IF( consulta=0)THEN
UPDATE profesor SET activo_inactivo=1
WHERE rfc=rfc1;
RETURN 1;
ELSE RETURN 0;
END IF;

END;
$$
LANGUAGE 'plpgsql';
```

4. **SP de Búsqueda:** este tipo de procedimientos almacenados deberá realizar la búsqueda de registros en las tablas de la BD mediante el identificador del registro y devolver el registro encontrado junto con todos los atributos a los que está asociado, aun cuando éstos pertenezcan a varias tablas.

- Función que busca un profesor en la tabla profesor y demás tablas asociadas, regresando la consulta buscada.

```
CREATE OR REPLACE FUNCTION sp_consulta_Profesor(
    rfc1 VARCHAR(9)
)
RETURNS SETOF profes_grupos_materias
AS
$$
BEGIN

RETURN QUERY(SELECT * FROM profes_grupos_materias WHERE rfc=rfc1);

END;
$$
LANGUAGE 'plpgsql';
```

5. **SP que genera el ranking:** se encargará de calcular el peso y aplicar el método de las ponderaciones según lo visto en el capítulo 3, de la calificación que tiene en promedio cada profesor en cada uno de los 25 criterios por materia y luego ordenar el ranking a presentar al usuario. Cabe destacar que los criterios no necesitan ser normalizados ya que el rango de valores se ha establecido dentro de una escala estándar del 1 al 5 para todos los criterios. Los SPs que se encargan de generar el ranking son 2, uno para aplicar el método de las ponderaciones y otro para ordenar por orden de preferencia. Para ello se ha establecido que cada alumno elegirá solo 5 criterios de los 25 totales en forma de más preferido al menor preferido. El código es el siguiente:

```

CREATE OR REPLACE FUNCTION sp_pesos_rank(
    hora_inicio1 VARCHAR(255),
    semestre_imparte1 VARCHAR(255),
    nombre_material VARCHAR(255),
    mas_preferido_1 VARCHAR(255),
    mas_preferido_2 VARCHAR(255),
    mas_preferido_3 VARCHAR(255),
    mas_preferido_4 VARCHAR(255),
    mas_preferido_5 VARCHAR(255)
)
RETURNS TABLE (
    rfc VARCHAR(255),
    comenta_perspectivas NUMERIC,
    deja_ideas NUMERIC,
    responde_loque NUMERIC,
    escucha NUMERIC,
    reconoce NUMERIC,
    trata_respeto NUMERIC,
    ejemplifica NUMERIC,
    temas_tratados NUMERIC,
    responde_concretamente NUMERIC,
    mantiene_actividad NUMERIC,
    aclara_terminologia NUMERIC,
    relaciona_tema_aplica NUMERIC,
    acepta_aportaciones NUMERIC,
    utiliza_ejemplos NUMERIC,
    acuerda_forma_evaluar NUMERIC,
    evaluaciones_periodicas NUMERIC,
    alumnos_interesen NUMERIC,
    todos_temas_programa NUMERIC,
    comunica_claridad_conocimientos NUMERIC,
    domina_tema NUMERIC,
    asiste_clases NUMERIC,
    aclara_dudas NUMERIC,
    deja_tareas NUMERIC,
    nivel_exámenes_clases NUMERIC,
    integra_teoría_práctica NUMERIC
)
AS
$$
DECLARE tupla promedio%ROWTYPE;
DECLARE suma NUMERIC(5,3);
DECLARE j INTEGER;
DECLARE contador INTEGER;
DECLARE contador1 INTEGER;
DECLARE contador2 INTEGER;
DECLARE contador3 INTEGER;
DECLARE contador4 INTEGER;
DECLARE contador5 INTEGER;
DECLARE Lista_pesos NUMERIC(5,3) ARRAY [25];
DECLARE Lista_criterios VARCHAR ARRAY [25];
DECLARE Lista_preferencias NUMERIC(5,3) ARRAY [25];
BEGIN

Lista_criterios = ARRAY
['comenta_perspectivas','deja_ideas','responde_loque','escucha','reconoce','trata_respet
o','ejemplifica','temas_tratados','responde_concretamente','mantiene_actividad',

```

```

    'aclara_terminologia','relaciona_tema_aplica','acepta_aportaciones','utiliza_ejem
plos','acuerda_forma_evaluar','evaluaciones_periodicas','alumnos_interesen','todos_temas
_programa',
    'comunica_claridad_conocimientos','domina_tema','asiste_clases','aclara_dudas','d
eja_tareas','nivel_examenés_clases','integra_teoría_practica'];

```

```

contador=5;
FOR i in 1..25
  loop
    IF Lista_criterios[i]= mas_preferido_1 THEN
      contador1=i;
    end IF;
    i=i+1;
  end loop;

FOR i in 1..25
  loop
    IF Lista_criterios[i]= mas_preferido_2 THEN
      contador2=i;
    end IF;
    i=i+1;
  end loop;

FOR i in 1..25
  loop
    IF Lista_criterios[i]= mas_preferido_3 THEN
      contador3=i;
    end IF;
    i=i+1;
  end loop;

FOR i in 1..25
  loop
    IF Lista_criterios[i]= mas_preferido_4 THEN
      contador4=i;
    end IF;
    i=i+1;
  end loop;

FOR i in 1..25
  loop
    IF Lista_criterios[i]= mas_preferido_5 THEN
      contador5=i;
    end IF;
    i=i+1;
  end loop;

Lista_preferencias[contador1]=1;
Lista_preferencias[contador2]=2;
Lista_preferencias[contador3]=3;
Lista_preferencias[contador4]=4;
Lista_preferencias[contador5]=5;

FOR i in 1..25
  loop
    IF Lista_criterios[i]= null THEN
      Lista_criterios[i]=0;
    end IF;
    i=i+1;
  end loop;

j=contador;

suma = 0;

FOR i IN 1..j
  loop
    suma=suma+(1.0/i);
    i=i+1;
  end loop;

FOR k IN 1..25
  loop
    IF Lista_preferencias[k]!=0 THEN
      Lista_pesos[k]=(1.0/Lista_preferencias[k])/suma;

```

```

        end IF;
        k=k+1;
    end loop;

    FOR tupla IN
        SELECT * FROM (SELECT W.rfc,AVG(W.comenta_perspectivas)as a,AVG(W.deja_ideas)as
b,AVG(W.responde_loque)as c,AVG(W.escucha)as d,AVG(W.reconoce) as e
,AVG(W.trata_respeto) as f,AVG(W.ejemplifica) as g,
        AVG(W.temas_tratados) as h,AVG(W.responde_concretamente) as
i,AVG(W.mantiene_actividad) as j,AVG(W.aclara_terminologia) as
k,AVG(W.relaciona_tema_aplica) as l,AVG(W.acepta_aportaciones) as m,
        AVG(W.utiliza_ejemplos) as n,AVG(W.acuerda_forma_evaluar) as
o,AVG(W.evaluaciones_periodicas) as p,AVG(W.alumnos_interesen) as
q,AVG(W.todos_temas_programa) as r,AVG(W.comunica_claridad_conocimientos) as s,
        AVG(W.domina_tema) as t,AVG(W.asiste_clases) as u,AVG(W.aclara_dudas) as
v,AVG(W.deja_tareas) as qq,AVG(W.nivel_examenes_clases) as
y,AVG(W.integra_teoría_practica) as z FROM vista_rank AS W WHERE
W.hora_inicio=hora_inicio1 AND W.semestre_imparte=semestre_imparte1 AND
W.nombre_materia=nombre_material GROUP BY W.rfc) AS R

    LOOP

        rfc := tupla.rfc;
        comenta_perspectivas := tupla.a*Lista_pesos[1];
        deja_ideas := tupla.b*Lista_pesos[2];
        responde_loque := tupla.c*Lista_pesos[3];
        escucha := tupla.d*Lista_pesos[4];
        reconoce := tupla.e*Lista_pesos[5];
        trata_respeto := tupla.f*Lista_pesos[6];
        ejemplifica := tupla.g*Lista_pesos[7];
        temas_tratados := tupla.h*Lista_pesos[8];
        responde_concretamente := tupla.i*Lista_pesos[9];
        mantiene_actividad := tupla.j*Lista_pesos[10];
        aclara_terminologia := tupla.k*Lista_pesos[11];
        relaciona_tema_aplica := tupla.l*Lista_pesos[12];
        acepta_aportaciones := tupla.m*Lista_pesos[13];
        utiliza_ejemplos := tupla.n*Lista_pesos[14];
        acuerda_forma_evaluar := tupla.o*Lista_pesos[15];
        evaluaciones_periodicas := tupla.p*Lista_pesos[16];
        alumnos_interesen := tupla.q*Lista_pesos[17];
        todos_temas_programa := tupla.r*Lista_pesos[18];
        comunica_claridad_conocimientos := tupla.s*Lista_pesos[19];
        domina_tema := tupla.t*Lista_pesos[20];
        asiste_clases := tupla.u*Lista_pesos[21];
        aclara_dudas := tupla.v*Lista_pesos[22];
        deja_tareas := tupla.qq*Lista_pesos[23];
        nivel_examenes_clases := tupla.y*Lista_pesos[24];
        integra_teoría_practica := tupla.z*Lista_pesos[25];

        RETURN NEXT;
    END LOOP;

END;
$$
LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION sp_sort_rank(
    hora_inicio1 VARCHAR(255),
    semestre_imparte1 VARCHAR(255),
    nombre_material VARCHAR(255),
    mas_preferido_1 VARCHAR(255),
    mas_preferido_2 VARCHAR(255),
    mas_preferido_3 VARCHAR(255),
    mas_preferido_4 VARCHAR(255),
    mas_preferido_5 VARCHAR(255)

)
RETURNS TABLE (
    rfc VARCHAR(255),
    comenta_perspectivas NUMERIC,
    deja_ideas NUMERIC,
    responde_loque NUMERIC,
    escucha NUMERIC,
    reconoce NUMERIC,

```

```

        trata_respeto NUMERIC,
        ejemplifica NUMERIC,
        temas_tratados NUMERIC,
        responde_concretamente NUMERIC,
        mantiene_actividad NUMERIC,
        aclara_terminologia NUMERIC,
        relaciona_tema_aplica NUMERIC,
        acepta_aportaciones NUMERIC,
        utiliza_ejemplos NUMERIC,
        acuerda_forma_evaluar NUMERIC,
        evaluaciones_periodicas NUMERIC,
        alumnos_interesen NUMERIC,
        todos_temas_programa NUMERIC,
        comunica_claridad_conocimientos NUMERIC,
        domina_tema NUMERIC,
        asiste_clases NUMERIC,
        aclara_dudas NUMERIC,
        deja_tareas NUMERIC,
        nivel_exámenes_clases NUMERIC,
        integra_teoría_práctica NUMERIC
    )
AS
$$
DECLARE tupla promedio%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT
            sp_pesos_rank(hora_inicio1, semestre_imparte1, nombre_material, mas_preferido_1, mas_preferido_2, mas_preferido_3, mas_preferido_4, mas_preferido_5)
            ORDER BY mas_preferido_1, mas_preferido_2, mas_preferido_3, mas_preferido_4, mas_preferido_5 DESC
        FROM
    LOOP
        rfc := tupla.rfc;
        comenta_perspectivas := tupla.a;
        deja_ideas := tupla.b;
        responde_loque := tupla.c;
        escucha := tupla.d;
        reconoce := tupla.e;
        trata_respeto := tupla.f;
        ejemplifica := tupla.g;
        temas_tratados := tupla.h;
        responde_concretamente := tupla.i;
        mantiene_actividad := tupla.j;
        aclara_terminologia := tupla.k;
        relaciona_tema_aplica := tupla.l;
        acepta_aportaciones := tupla.m;
        utiliza_ejemplos := tupla.n;
        acuerda_forma_evaluar := tupla.o;
        evaluaciones_periodicas := tupla.p;
        alumnos_interesen := tupla.q;
        todos_temas_programa := tupla.r;
        comunica_claridad_conocimientos := tupla.s;
        domina_tema := tupla.t;
        asiste_clases := tupla.u;
        aclara_dudas := tupla.v;
        deja_tareas := tupla.qq;
        nivel_exámenes_clases := tupla.y;
        integra_teoría_práctica := tupla.z;
    RETURN NEXT;
END LOOP;

END;
$$
LANGUAGE 'plpgsql';

```

6. **SP que obtenga la evaluación global de un profesor:** es el encargado de obtener todas las evaluaciones realizadas a un profesor durante toda su historia como académico en la institución.

- Esta función calcula la evaluación global promedio de un profesor recibiendo como parámetro el rfc del mismo.

```

CREATE OR REPLACE FUNCTION sp_evaluaciones_Profesor_global(
    rfc1 VARCHAR(255)
)
RETURNS TABLE (
    rfc VARCHAR(255),
    comenta_perspectivas NUMERIC,
    deja_ideas NUMERIC,
    responde_loque NUMERIC,
    escucha NUMERIC,
    reconoce NUMERIC,
    trata_respeto NUMERIC,
    ejemplifica NUMERIC,
    temas_tratados NUMERIC,
    responde_concretamente NUMERIC,
    mantiene_actividad NUMERIC,
    aclara_terminologia NUMERIC,
    relaciona_tema_aplica NUMERIC,
    acepta_aportaciones NUMERIC,
    utiliza_ejemplos NUMERIC,
    acuerda_forma_evaluar NUMERIC,
    evaluaciones_periodicas NUMERIC,
    alumnos_interesen NUMERIC,
    todos_temas_programa NUMERIC,
    comunica_claridad_conocimientos NUMERIC,
    domina_tema NUMERIC,
    asiste_clases NUMERIC,
    aclara_dudas NUMERIC,
    deja_tareas NUMERIC,
    nivel_exámenes_clases NUMERIC,
    integra_teoría_práctica NUMERIC
)
AS
$$
DECLARE tupla promedio%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT * FROM promedio AS R WHERE R.rfc=rfc1

LOOP

    rfc := tupla.rfc;
    comenta_perspectivas := tupla.a;
    deja_ideas := tupla.b;
    responde_loque := tupla.c;
    escucha := tupla.d;
    reconoce := tupla.e;
    trata_respeto := tupla.f;
    ejemplifica := tupla.g;
    temas_tratados := tupla.h;
    responde_concretamente := tupla.i;
    mantiene_actividad := tupla.j;
    aclara_terminologia := tupla.k;
    relaciona_tema_aplica := tupla.l;
    acepta_aportaciones := tupla.m;
    utiliza_ejemplos := tupla.n;

```



```

        acuerda_forma_evaluar := tupla.o;
        evaluaciones_periodicas := tupla.p;
        alumnos_interesen := tupla.q;
        todos_temas_programa := tupla.r;
        comunica_claridad_conocimientos := tupla.s;
        domina_tema := tupla.t;
        asiste_clases := tupla.u;
        aclara_dudas := tupla.v;
        deja_tareas := tupla.qq;
        nivel_examenes_clases := tupla.y;
        integra_teoría_practica := tupla.z;
    RETURN NEXT;
END LOOP;

END;
$$
LANGUAGE 'plpgsql';

```

7. SP que obtenga todos los comentarios: obtiene todos los comentarios que se han hecho de un profesor durante toda su historia como académico en la institución.

- Esta función muestra los comentarios hechos a un profesor durante toda su historia como docente en la entidad educativa.

```

CREATE OR REPLACE FUNCTION sp_comentarios_Profesor_global(
    rfc1 VARCHAR(255)
)
RETURNS TABLE (
    comentario VARCHAR(255)
)
AS
$$
DECLARE tupla vista_rank%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT * FROM vista_rank as A WHERE A.rfc=rfc1

    LOOP
        comentario := tupla.comentario;

    RETURN NEXT;
    END LOOP;

END;
$$
LANGUAGE 'plpgsql';

SELECT * FROM sp_comentarios_Profesor_global('MOTM840404');

```

8. **SP que obtenga todos los comentarios por materia:** a diferencia del SP anterior, éste recupera los comentarios que se han hecho de un profesor durante toda su historia como académico en la institución pero agrupados por materia.

- Esta función muestra los comentarios por materia hechos a un profesor durante toda su historia como docente en la entidad educativa.

```
CREATE OR REPLACE FUNCTION sp_comentarios_Profesor_materia(
    rfc1 VARCHAR(255)
)
RETURNS TABLE (
    nombre_materia VARCHAR(255),
    comentario VARCHAR(255)
)
AS
$$
DECLARE tupla vista_rank%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT * FROM vista_rank AS A WHERE A.rfc=rfc1 ORDER BY A.nombre_materia
    LOOP
        nombre_materia := tupla.nombre_materia;
        comentario := tupla.comentario;

        RETURN NEXT;
    END LOOP;

END;
$$
LANGUAGE 'plpgsql';
```

9. **SP que obtenga el promedio de las evaluaciones por materia:** obtiene el promedio por materia de todas las evaluaciones que se han hecho de un profesor durante toda su historia como académico en la institución.

- Esta función muestra las evaluaciones promedio de un profesor por materia.

```
CREATE OR REPLACE FUNCTION sp_evaluaciones_Profesor_global_materias(
    rfc1 VARCHAR(255)
)
RETURNS TABLE (
    nombre_materia VARCHAR(255),
    comenta_perspectivas NUMERIC,
    deja_ideas NUMERIC,
    responde_loque NUMERIC,
    escucha NUMERIC,
    reconoce NUMERIC,
    trata_respeto NUMERIC,
    ejemplifica NUMERIC,
    temas_tratados NUMERIC,
    responde_concretamente NUMERIC,
    mantiene_actividad NUMERIC,
    aclara_terminologia NUMERIC,
    relaciona_tema_aplica NUMERIC,
    acepta_aportaciones NUMERIC,
    utiliza_ejemplos NUMERIC,
    acuerda_forma_evaluar NUMERIC,
    evaluaciones_periodicas NUMERIC,
    alumnos_interesen NUMERIC,

```

```

        todos_temas_programa NUMERIC,
        comunica_claridad_conocimientos NUMERIC,
        domina_tema NUMERIC,
        asiste_clases NUMERIC,
        aclara_dudas NUMERIC,
        deja_tareas NUMERIC,
        nivel_exámenes_clases NUMERIC,
        integra_teoría_práctica NUMERIC
    )
AS
$$
DECLARE tupla promedio_materias%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT
            W.nombre_materia,AVG(W.comenta_perspectivas)as
a,AVG(W.deja_ideas)as      b,AVG(W.responde_loque)as      c,AVG(W.escucha)as
d,AVG(W.reconoce) as e ,AVG(W.trata_respeto) as f,AVG(W.ejemplifica) as g,
            AVG(W.temas_tratados)      as      h,AVG(W.responde_concretamente)      as
i,AVG(W.mantiene_actividad)      as      j,AVG(W.aclara_terminologia)      as
k,AVG(W.relaciona_tema_aplica) as l,AVG(W.acepta_aportaciones) as m,
            AVG(W.utiliza_ejemplos)      as      n,AVG(W.acuerda_forma_evaluar)      as
o,AVG(W.evaluaciones_periodicas)      as      p,AVG(W.alumnos_interesen)      as
q,AVG(W.todos_temas_programa) as r,AVG(W.comunica_claridad_conocimientos) as s,
            AVG(W.domina_tema) as t,AVG(W.asiste_clases) as u,AVG(W.aclara_dudas) as
v,AVG(W.deja_tareas)      as      qq,AVG(W.nivel_exámenes_clases)      as
y,AVG(W.integra_teoría_práctica) as z FROM vista_rank AS W WHERE W.rfc=rfc1 GROUP
BY W.nombre_materia
    LOOP
        nombre_materia := tupla.nombre_materia;
        comenta_perspectivas := tupla.a;
        deja_ideas := tupla.b;
        responde_loque := tupla.c;
        escucha := tupla.d;
        reconoce := tupla.e;
        trata_respeto := tupla.f;
        ejemplifica := tupla.g;
        temas_tratados := tupla.h;
        responde_concretamente := tupla.i;
        mantiene_actividad := tupla.j;
        aclara_terminologia := tupla.k;
        relaciona_tema_aplica := tupla.l;
        acepta_aportaciones := tupla.m;
        utiliza_ejemplos := tupla.n;
        acuerda_forma_evaluar := tupla.o;
        evaluaciones_periodicas := tupla.p;
        alumnos_interesen := tupla.q;
        todos_temas_programa := tupla.r;
        comunica_claridad_conocimientos := tupla.s;
        domina_tema := tupla.t;
        asiste_clases := tupla.u;
        aclara_dudas := tupla.v;
        deja_tareas := tupla.qq;
        nivel_exámenes_clases := tupla.y;
        integra_teoría_práctica := tupla.z;
    RETURN NEXT;
END LOOP;

END;
$$
LANGUAGE 'plpgsql';

```

Una vez que se tienen los elementos necesarios que dan soporte al SATDEH, en el siguiente capítulo se verá en dónde se utiliza todo lo construido en este capítulo y la justificación de lo que fue refactorizado y el porqué.

CAPÍTULO 6 Proceso de desarrollo del DSS

En este capítulo se abordará la construcción del SATDEH, desde los requerimientos a implementar, la construcción, la integración y hasta las pruebas realizadas después de haber concluido con el proceso de desarrollo.

Como se describió en el capítulo 2, el desarrollo de un BI sigue 6 etapas comunes que son: Justificación, Planeación, Análisis del sistema, Diseño, Construcción e Implementación. La Figura 6.1 muestra de manera esquemática el proceso de desarrollo del SATDEH desde la justificación hasta la implementación.

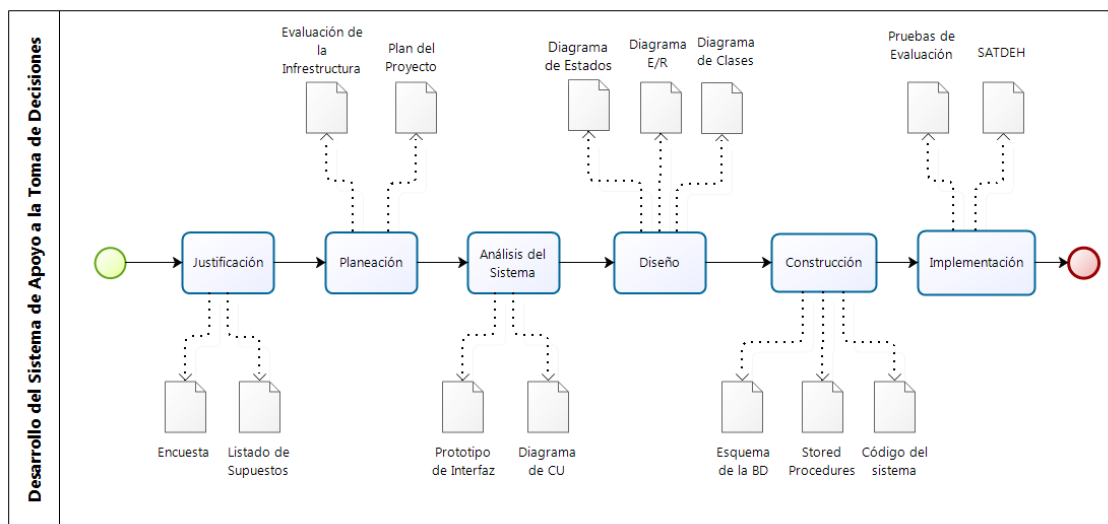


Figura 6.1. Proceso de desarrollo para construir el sistema.

El desglose de dichas etapas bajo el contexto de este proyecto y que según sugiere (Moss & Atre, 2003) se presenta a continuación.

6.1 Justificación

En este punto se presentan aquellos argumentos que justifican la construcción del SATDEH que es el caso de negocio que se aborda. Llevados por la necesidad de proveer al alumno de una herramienta que le permita elaborar su horario de clases, se llega a la propuesta de un sistema de apoyo a la toma de decisiones. La implementación de este sistema ha resultado ser de bajo costo y proporciona al usuario una alternativa para obtener un horario de clases.

A partir de ponderar los criterios que describen los intereses que tiene un alumno, el sistema proporciona un ranking de los profesores que cumplen con los criterios facilitados, así como los comentarios hechos por otros alumnos sobre dichos profesores. En concreto los beneficios que se espera proporcione el SATDEH son los siguientes:

- Bajo costo en implementación
- Proporcionará información específica sobre los profesores
- Permitirá la libre opinión sobre el personal docente de la institución educativa
- Permitirá la elaboración de un horario de clases de manera simple y práctica

Algo muy importante para la justificación de este desarrollo son los supuestos asumidos en el diseño de la base de datos, los cuales son un producto de trabajo que se muestra en la Figura 6.2.



LISTADO DE SUPUESTOS

A continuación se menciona el listado de los supuestos en el diseño de la base de datos del Sistema de apoyo a la toma de decisiones de los alumnos para elaborar horarios de clases a partir de evaluaciones docentes.

- Se asume que la base de datos estará dirigida a una entidad educativa de nivel superior.
- Cada profesor debe contar con un número de identificación único dentro de la entidad educativa.
- El profesor puede laborar o no de tiempo completo dentro de la entidad educativa.
- Un profesor puede ser titular y adjunto a la vez pero no del mismo grupo.
- Un profesor puede estar asociado a varios grupos, tanto como titular como de adjunto, y un grupo solo puede estar asociado a un solo profesor titular.
- Un profesor adjunto puede estar asociado a muchos grupos y un grupo puede tener varios profesores adjuntos.
- El número de grupo debe ser único.
- Cada grupo debe tener asociado un horario y salón de clases.
- Cada materia impartida dentro de la entidad debe tener como características el semestre en que se imparte, una clave única de identificación y el número de créditos.
- Una materia puede estar asociada a muchos grupos pero un grupo solo puede estar asociado a una materia.
- Cada alumno debe contar con un número de identificación dentro de la entidad educativa, éste debe ser único para cada alumno.
- Un alumno puede estar inscrito en muchos grupos y un grupo puede tener a muchos alumnos inscritos.
- Un alumno no puede estar inscrito más de una vez en un mismo grupo.
- Si un alumno o profesor se va de la entidad se asume que puede regresar a la entidad nuevamente en un futuro.
- Es un requisito para poder realizar comentarios sobre alguno de los grupos, ser alumno de la entidad educativa y haber estado inscrito en dicho grupo en semestres anteriores.
- Deberá especificarse a qué carrera y a qué generación pertenece cada alumno.
- Para fines de la evaluación de profesores se contará con una encuesta previamente definida.

Figura 6.2. Supuestos en el diseño de la base de datos.

Otro aspecto a considerar que justifica el desarrollo es la encuesta realizada a los alumnos, cuyos detalles fueron mostrados en el capítulo 5.

6.2 Planificación

La planificación en este proyecto consistió en dos partes: la evaluación de la infraestructura con la que se cuenta y la distribución temporal de todo el proceso de desarrollo. De lo anterior se generaron dos productos de trabajo que se presentan a continuación.

6.2.1 Evaluación de la Infraestructura

La infraestructura con la que se cuenta para este proyecto es la considerada en la Figura 6.3 asumiendo que el SATDEH debe ser de bajo costo en implementación.



EVALUACIÓN DE LA INFRAESTRUCTURA

Dado que este proyecto no está al servicio de ninguna organización en particular la infraestructura con la que contamos es la siguiente:

La infraestructura técnica es el conjunto de software con el que se llevará a cabo el desarrollo y es el siguiente:

- Sistema Manejador de Bases de Datos
 - PostgreSQL 9.1 (Postgresql, 2013)
- Lenguaje de Programación de propósito general
 - Groovy 2.1.7 (Groovy, 2013)
 - Grails 2.3.0 (Grails, 2013)
- Entorno de Desarrollo Integrado
 - NetBeans 7.3 (Netbeans, 2013)
- Herramientas para diagramación con distintos objetivos
 - StarUML 5.0.2.1570 (Staruml, 2013)
 - Dia 0.97.2 (Dia, 2013)
 - Bizagi Process Modeler 2.6 (Bizagi, 2013)
 - Pencil 2.0.3 (Pencil, 2013)
- Sistema de encuestas en línea
 - LimeSurvey 1.91 (Limesurvey, 2013)
- Procesador de textos
 - Notepad++ 6.3.2 (Notepad, 2013)

Figura 6.3. Listado de software que conforma la infraestructura técnica, todos son de carácter libre.

En cuanto a la infraestructura no técnica, ésta recae en su totalidad en la base de datos donde estará ubicado el repositorio de datos, el modelo lógico, los procedimientos de prueba, prototipos y el mismo diseño de la base de datos.

6.2.2 Planificación del proyecto

El proyecto busca satisfacer las necesidades de los alumnos que pretenden realizar un horario de clases. El primer paso fue obtener los requerimientos que busca un alumno sobre un profesor (ver capítulo 5), con ello se procedió al diseño y elaboración de la base de datos que da solución a las necesidades obtenidas.

Una vez almacenados los datos, el SATDEH los utiliza para elaborar un ranking de los profesores que se adecúen a los requerimientos del alumno. Este ranking es construido a partir de criterios proporcionados por el estudiante, mismos que describen sus intereses, y pueden ser modificados en consultas posteriores. Estos criterios están asociados principalmente a la evaluación del desempeño docente de los profesores.

El paso siguiente fue crear una interfaz web con la que el usuario del SATDEH interactúe, mediante un lenguaje de programación, en este caso JAVA y utilizando el framework de Groovy on Grails (Groovy, Grails, 2013). A través de ella podrá llevar a cabo la elaboración de su horario de clases, tomando en cuenta los rankings generados a partir de la información recolectada por el sistema.

La decisión de utilizar Grails fue porque es un framework que facilita bastante la labor de programación, arquitectura e interfaz, además de que permitió omitir el uso de varios de los SPs, como los de inserción, actualización, borrado y algunos de consulta

debido a que es el framework el que establece el cómo se interaccionará con el SMBD. Todo lo descrito anteriormente representa un plan de proyecto que se llevó a cabo bajo un tiempo determinado. La distribución temporal de las actividades anteriores se muestra en la Figura 6.4.

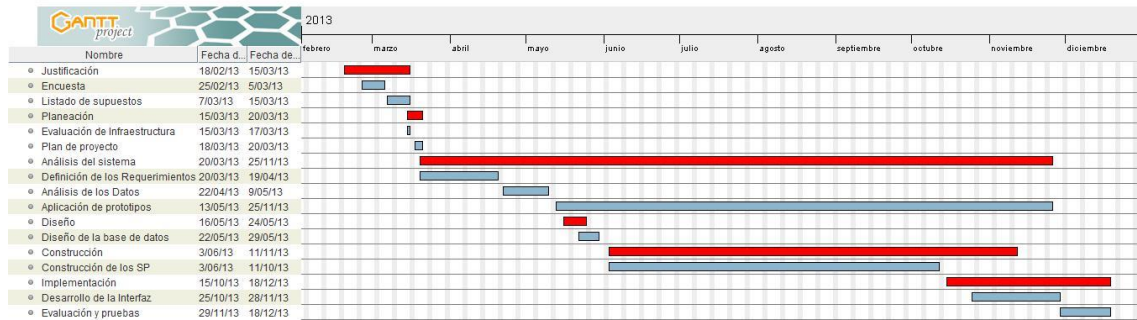


Figura 6.4. Distribución temporal de las fases de construcción del sistema.

6.3 Análisis del Sistema

Este punto consiste en presentar la definición de los requerimientos, el análisis de los datos y el diseño de prototipos, para ello se construyeron dos productos de trabajo, un prototipo sobre las vistas del sistema y un diagrama de casos de uso que se presentan a continuación.

6.3.1 Definición de los requerimientos

El sistema debe dar solución al problema que se busca resolver, es decir, proporcionar al usuario la información suficiente y concreta que le permita tomar decisiones al elaborar su horario de clases. Todo lo anterior debe llevarse a cabo con un bajo costo de implementación, ser robusto, fácil de operar y ser sensible a cambios en el aspecto del manejo de la información.

Las limitaciones tecnológicas deberán restringirse al software utilizado en el sistema. Las necesidades que rigieron la definición de los requerimientos son los definidos en el producto de trabajo mostrado en la Figura 6.2. A partir de ellos se construyó un Diagrama General de Casos de Uso que modela las funcionalidades y el alcance de SATDEH, ver Figura 6.5

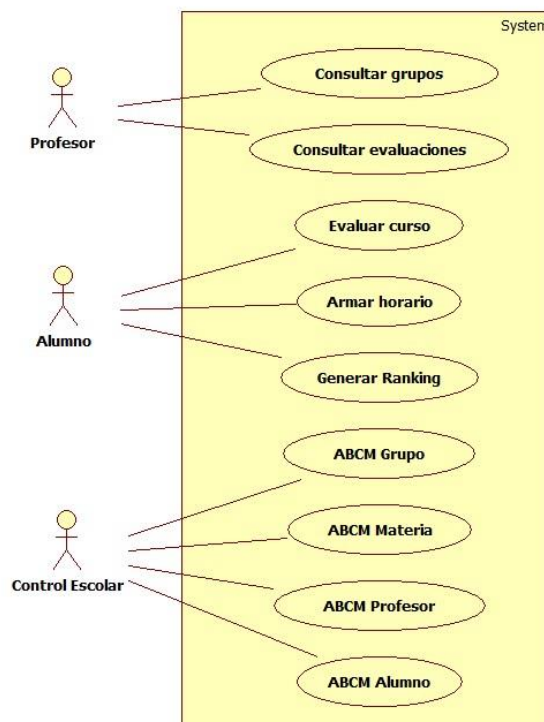


Figura 6.5. Diagrama general de casos de uso del SATDEH

6.3.2 Análisis de los datos

Para tener un buen análisis de los datos, la base de datos fue estructurada con los principios de buen diseño aprendidos en el curso de Bases de Datos. Dicho análisis se llevó a cabo dentro del mismo SMBD de la base de datos por medio de procedimientos almacenados (SP), por lo que la logística de cada SP es muy precisa y clara. La realización de este análisis optimiza en su gran mayoría el tiempo de implementación y de consulta, es aquí donde la arquitectura *in-memory* influyó en el SATDEH.

6.3.3 Aplicación de prototipos

Los prototipos del SATDEH están en la base de datos, ya que la solución de problema se concentra ahí, las pruebas necesarias han podido realizarse en ausencia de la interfaz de usuario pues toda la complejidad el sistema está concentrada en los SP.

En cuanto a los prototipos de la interfaz de usuario sólo fue realizado uno por medio de la herramienta Pencil. La Figura 6.6 muestra la vista de búsqueda de grupos del prototipo del SATDEH, la cual fue modificada para la versión final del sistema.

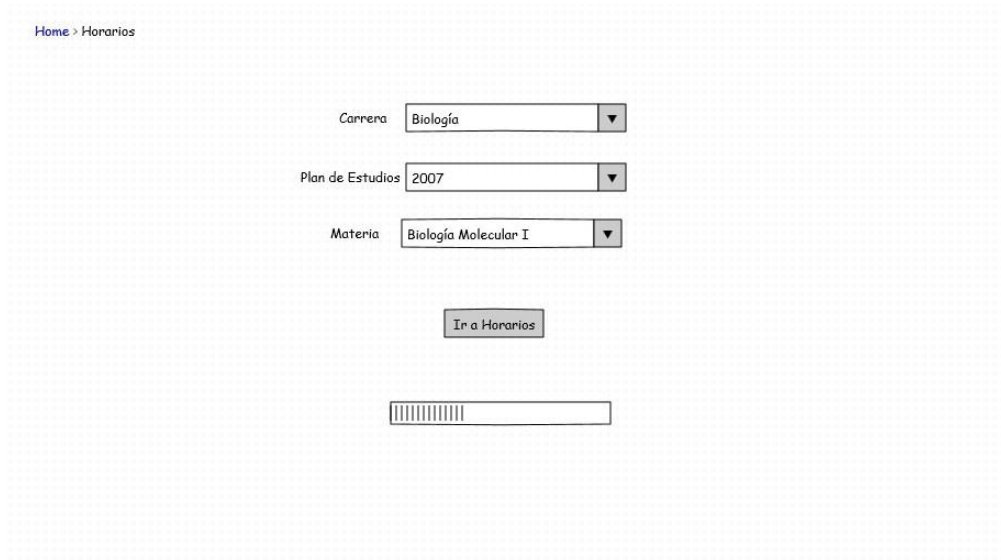


Figura 6.6. Vista de búsqueda de grupos del prototipo del SATDEH.

6.4 Diseño

En este punto se logró la creación del diseño de la base de datos, el diagrama de estados y de las herramientas ETL del SATDEH.

6.4.1 Diseño de la base de datos

Para el diseño de la base de datos se utilizó el Modelo Relacional, el capítulo 5 muestra detalladamente todo el proceso de diseño llevado a cabo, así como el diagrama Entidad/Relación (Figura 5.3.) y el diagrama de Clases UML (Figura 5.4.).

Para modelar la navegación del sistema se utilizaron diagramas de estados de UML para cada actor del SATDEH. Según sus casos de uso, los diagramas de estado generados para Alumno, Control Escolar (Administrador) y Profesor se muestran en las Figuras 6.7, 6.8 y 6.9 respectivamente.

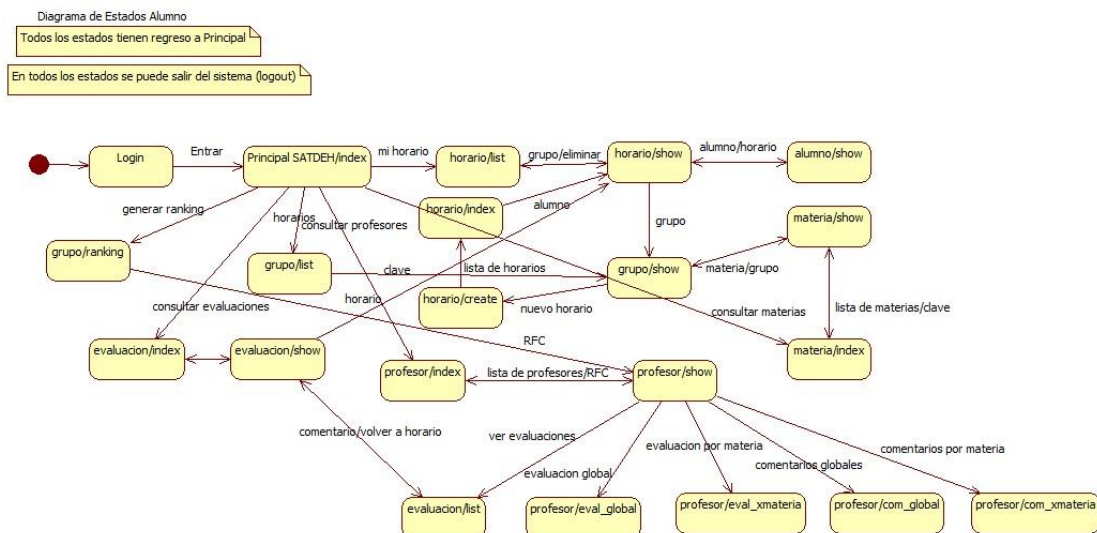


Figura 6.7 Diagrama de Estados para el usuario Alumno

Extracción. La extracción de datos se llevará a cabo por medio del SMBD que permite la extracción de los datos. Dicha extracción se lleva a cabo mediante el uso de SPs, la limpieza de los datos se asegura desde la inserción de los mismos mediante el uso de validaciones de restricciones de la interfaz de usuario. En la Figura 6.10 se muestra una captura de pantalla de la aplicación de validaciones, en ella se puede apreciar que el SATDEH vigila la integridad en los tipos de datos solicitando al usuario ingrese el dato del atributo Hora Inicio en un formato correcto.

The screenshot shows a web form titled "Crear Grupo" with the instruction: "Ingresa los siguientes datos del grupo que se quiere dar de alta, recuerda que los campos con * son obligatorios." The form contains the following fields and values:

- Clave *: 3312
- Hora inicio *: hh (highlighted in red with a tooltip: "Por favor, coincide con el formato solicitado.")
- Hora fin *: (empty)
- Aula *: Magna II
- Lunes:
- Martes:
- Miércoles:
- Jueves:
- Viernes:
- Semestre impartido *: 2014-II
- Materia *: 575 Ingeniería de Software
- Profesor *: MOTM840404 Miguel Ehecatl Morales Trujillo

At the bottom left, there is a "Crear" button with a lock icon.

Figura 6.10. Vista de validación de datos para la creación de un nuevo grupo dentro del SATDEH.

Transformación. La transformación la dará la interfaz quien procesará consultas en lenguaje natural predefinidas a lenguaje SQL y viceversa para permitir la transición de la información a una presentación acorde a las necesidades del usuario. La Figura 6.11 muestra el ranking hecho por el SATDEH a un usuario, en ella se puede apreciar el efecto de la transformación, es decir, la información generada por el SP de ranqueo arrojada por el SMBD es transformada y mostrada de manera adecuada al usuario.

Aquí puedes realizar el ranqueo de grupos por cualquier combinación de los siguientes criterios.

Semestre en que se imparte:

Materia:

Hora de inicio:

Criterio 1:

Criterio 2:

Criterio 3:

Criterio 4:

Criterio 5:

Ranking

RFC	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5
BSHY951585	1.75200	0.74460	0.43800	0.17600	0.21120
MOZE179593	1.31400	0.65700	0.14600	0.11000	0.44000
NJZK942931	0.70080	0.65700	0.49640	0.33000	0.33440

Figura 6.11. Pantalla de un ranking hecho por el SATDEH.

Carga. La carga recae principalmente en la interfaz ya que en ella se muestra al usuario la información de manera organizada para que este pueda tomar decisiones, tal como se puede apreciar en la figura anterior.

6.6 Construcción

Lo que se pretende lograr en este punto del desarrollo es construir la base de datos según lo establecido en el diseño, los SPs y la construcción del SATDEH en un lenguaje de programación. El código para la creación de la base de datos es el mostrado en el capítulo 5 al igual que el código con el que fueron creados los SPs en el SMBD.

Lo realizado a través de JAVA y Groovy on Grails es la base de la presentación del SATDEH con la que el usuario interactuará. A continuación se describen los elementos de mayor importancia dentro del desarrollo.

El árbol de directorios del proyecto, el cual se muestra en la Figura 6.12, dicho árbol contiene los elementos necesarios para que el sistema opere y es generado automáticamente por el framework. El ambiente de desarrollo utilizado fue Netbeans.

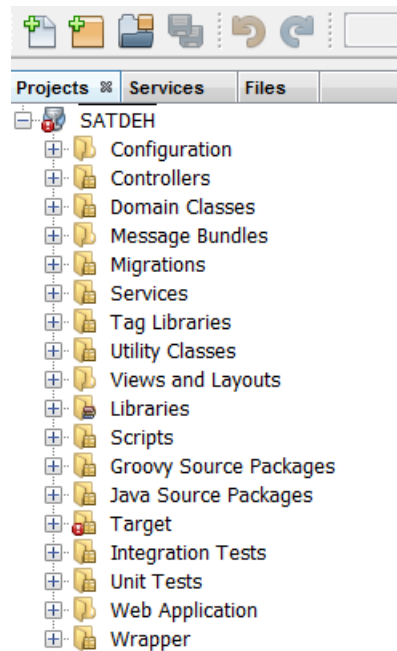


Figura 6.12. Árbol de directorios del proyecto SATDEH.

Los principales elementos del directorio y con los que más se trabajó para el desarrollo del sistema fueron las carpetas de *Controllers*, *Domain Classes* y *Views and Layouts*. La carpeta de *Controllers* contiene los controladores del proyecto, los controladores son los encargados de realizar las acciones que ligan las vistas con el modelo del sistema, por ejemplo la inserción de registros a la base de datos del SATDEH. En la carpeta *Controllers*, grails genera los módulos de inserción, actualización y borrado, así como las ligas entre las vistas y las clases. Se debe crear un controlador para cada clase creada en el proyecto.

La Figura 6.13 muestra la estructura de la carpeta *Controllers*, todos sus elementos con la extensión *groovy* que es el lenguaje empleado por grails, se puede observar que existen tantos controladores como clases, y existen clases como al menos el número de tablas que hay en el diseño de la base de datos del SATDEH.

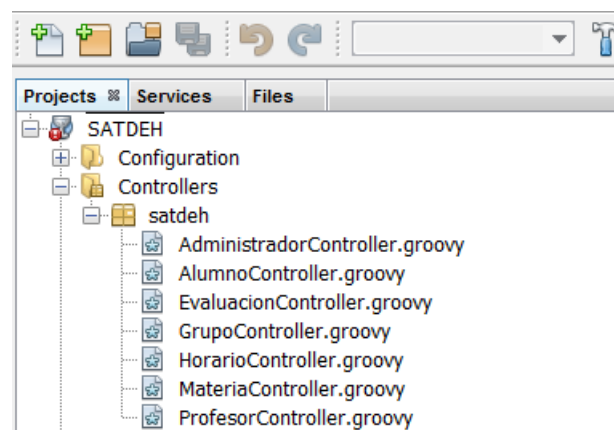


Figura 6.13. Elementos que conforman la carpeta *Controllers*.

La carpeta *Domain Classes* contiene las clases del proyecto, cada clase representa una de las tablas del diseño de la base de datos, en cada elemento de la carpeta lo único que se hizo fue mapear cada entidad del diseño que ya se tenía y traducirla al lenguaje correspondiente junto con sus restricciones y tipo de dato.

En la Figura 6.14 se muestra la estructura de la clase profesor, en ella se ve como se declaran los atributos que posee la clase y qué tipo de dato tiene cada uno. La instrucción *hasMany* indica que existe una relación de muchos a 1 entre la clase profesor y la clase grupo. En la parte de las *Constraints* se indican las restricciones de cada atributo de la clase profesor, por ejemplo *rfc*, es un dato no nulo, con un tamaño de 10 caracteres, se pueden ingresar todas las letras del abecedario y los números del 0 al 9, prácticamente se vigila la integridad de los datos para que el sistema pueda tener compatibilidad entre el SMDB y la interfaz de usuario.

```

1 package satdeh
2
3 class Profesor {
4     String rfc
5     String nombre_prof
6     String ap_pat_prof
7     String ap_mat_prof
8     String grado
9     String correo_e_prof
10    Integer activo_inactivo
11    String nombramiento
12    String contrasena
13
14    static hasMany = [grupo : Grupo]
15
16    String toString() {
17        return " ${rfc} ${nombre_prof} ${ap_pat_prof} ${ap_mat_prof}"
18    }
19
20    static constraints = {
21        rfc (nullable: false, blank: false, minSize: 10, matches: "[A-Za-z0-9]+")
22        nombre_prof (nullable: false, blank: false, matches: "[A-Za-zÁÉÍÓÚáéíóúÑ ]+")
23        ap_pat_prof (nullable: false, blank: false, matches: "[A-Za-zÁÉÍÓÚáéíóúÑ ]+")
24        ap_mat_prof (nullable: true, blank: false, matches: "[A-Za-zÁÉÍÓÚáéíóúÑ ]+")
25        grado (inList: ["Licenciatura", "Maestria", "Doctorado"])
26        correo_e_prof (nullable: false, blank: false, email: true)
27        activo_inactivo (inList: [0, 1])
28        nombramiento (inList: ["Profesor de Asignatura A", "Profesor de Asignatura B", "Profesor de Carrera"])
29        contrasena (nullable: false, blank: false, password: true)
30    }
31 }

```

Figura 6.14. Código de la clase de domino de profesor.

Como ya se ha mencionado el SATDEH cuenta con tres actores referidos en la Figura 6.5, el alumno, el profesor y control escolar. En la carpeta *Views and Layouts* se encuentran todas las vistas de las clases que generan la interfaz de usuario mostrada a cada actor. La Figura 6.15 muestra la estructura de dicha carpeta, en ella se pueden encontrar tres vistas que competen a los tres actores, ellas son *error*, *index* y *login*. La primera es la vista creada para ser mostrada en caso de que exista cualquier error en el sistema, la vista *index* es la vista de menú principal y que varía según el actor que entre al sistema, mientras que la de *login* es la vista que se muestra para que cualquier usuario se identifique y pueda ingresar al SATDEH según su rol.

Dentro de la interacción del usuario con el sistema, hay diversos estados a los que puede ir, ver Figura 6.7, 6.8 y 6.9, cada estado representa una vista, por ejemplo la carpeta profesor que puede apreciarse en la Figura 6.15 está conformada por 5 elementos, *_form*, *create*, *edit*, *index* y *show*. El primero *_form* es la base de las demás vistas, y en cada acción las vistas de esa clase consultan en *_form* si es posible la acción solicitada; *create* es la vista mostrada para dar de alta nuevos registros, en ella se realizan las inserciones. Las actualizaciones y el borrado corresponden a la vista *edit*; *index* muestra el menú general de la clase y sus instancias y *show* muestra el detalle de una instancia seleccionada.

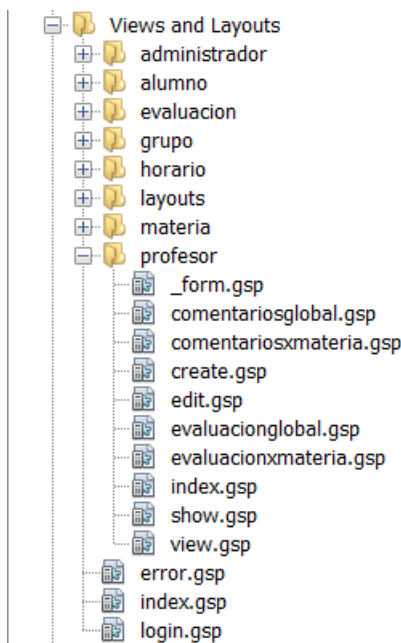


Figura 6.15. Clase profesor dentro de la capeta Views and Layouts.

La Figura 6.16 muestra la vista *index* de profesor en donde se tiene el listado de profesores que hay en una institución. Mientras que la Figura 6.17 muestra la vista *show* de la vista mostrada en la Figura 6.16, es decir el detalle de un profesor mostrado después de dar click al respectivo RFC.

SATDEH

Sistema de Apoyo a la Toma de Decisiones en la Elaboración de Horarios

Hola Pablo[Logout]

Principal
Nuevo Profesor

Lista de Profesor

RFC	Nombre	Apellido paterno	Apellido materno	Grado	Correo electrónico
YONU714385	Erich	Hansen	Bullock	Doctorado	et.arcu.imperdiet@loremlorem.co.uk
NLVL818906	Uriel	Ellison	Mendoza	Licenciatura	et@metusurnaconvallis.ca
MTLL957842	Ashton	Scott	Hopper	Licenciatura	senectus.et@nisl.com
QECE967760	Hadley	Shields	Thompson	Licenciatura	sit.amet@utlacus.org
ZNYV537656	Coby	Mitchell	Foreman	Doctorado	et.risus.Quisque@ligulaNullam.net
WZJP102529	Lacota	Navarro	Jensen	Doctorado	eu.odio@nequeNullam.co.uk
VYLH645258	Ali	Monroe	Hooper	Doctorado	molestie.tortor.nibh@molestietellus.ca
TBBS378223	Fritz	Hanson	Underwood	Maestria	imperdiet.nec@augueeu.org
HQPV632891	Tobias	Soto	Little	Doctorado	non.nisi@elit.net
QKDU212096	Noelle	Patton	Roach	Licenciatura	habitant@eliterat.co.uk

Anterior
1
2
3
4
5
6
7
8
9
10
Siguiente

Figura 6.16. Vista index de la lista de profesores del SATDEH.

Principal Lista de Profesor Nuevo Profesor

Mostrar Profesor

RFC MOTM840404

Nombre Miguel Ehecatl

Apellido paterno Morales

Apellido materno Trujillo

Grado Doctorado

Correo electrónico migmor@ciencias.unam.mx

Status 1

Nombramiento Profesor de Asignatura B

Grupo [3312 2014-II](#)
[4656 2014-I](#)
[1478 2014-I](#)

Editar Eliminar Ver Evaluaciones

Sección de apoyo a la toma de decisiones

Evaluación Global Evaluación por Materia Comentarios Globales Comentarios por Materia

Figura 6.17. Vista show con los detalles de un profesor dentro del SATDEH.

6.6.1 ETL Extraer / Transformar / Cargar de la Construcción

Como ya se ha mencionado anteriormente la ETL estará implementada tanto en la base de datos como en la interfaz, con ello no se requiere de ninguna otra herramienta fuera de las antes mencionadas. Las ETL forman parte de la construcción de los SP y la interfaz de usuario, que en conjunto componen al SATDEH.

6.6.1.1 Desarrollo de aplicaciones

Las aplicaciones son principalmente los SP encargados de generar el ranking a partir de los criterios proporcionados por el usuario y la interfaz que permitirá la transición de la información entre el usuario y la base de datos.

6.6.1.2 Minería de datos

Parte de la minería de datos será realizada por el sistema tras la elaboración del ranking, sin embargo la otra parte de la minería de los datos le corresponde al usuario, quien con un uso adecuado del sistema, podrá inferir, de los datos proporcionados por el SATDEH, información y así tomar decisiones más sólidas.

Una vez que se ha concluido con el desarrollo de la base de datos, los SPs y se han probado los prototipos del SATDEH, se procederá a implementar la interfaz que permita al usuario tener acceso al sistema. Tras esta fase, el SATDEH quedará completo.

6.7 Implementación

En este punto se llevaron a cabo la implementación, liberación y las pruebas de evaluación del SATDEH.

Una vez concluido el SATDEH se procedió a probar si cumple con las expectativas que se esperaban, si realmente resuelve el problema planteado. Para ello se probó con usuarios que cuenten con intereses en el problema que pretende resolver el sistema.

La liberación del SATDEH consistió en dejar listo el sistema para ser utilizado, accesible desde cualquier navegador web. Esta liberación pudo ser hecha montando el sistema en un servidor de aplicaciones, en este caso Tomcat7. La Figura 6.18 Muestra una vista del home de usuario del SATDEH ingresando como alumno al sistema.

Cabe mencionar que para poblar la base de datos y para fines de prueba del sistema se utilizó el software GenerateData (GenerateData, 2014). Esta aplicación permitió generar entre 100 y 800 tuplas para cada una de las tablas de la base de datos de SATDEH.



Figura 6.18. Vista del *home* de usuario del SATDEH ingresando como alumno al sistema.

Tomando en cuenta la opinión de un grupo de usuarios del sistema, se reconoce que el SATDEH requiere de mejoras estéticas, así como de módulos que permitan la fácil administración del sistema. Lejos de eso la opinión de dichos usuarios fue positiva en cuanto al soporte de decisiones que ofrece el SATDEH para la elaboración de horarios de clases.

Entre las opiniones recopiladas, los usuarios creen que sería una buena idea implementar dicho sistema dentro de alguna institución de nivel superior, en particular en la Facultad de Ciencias de la UNAM.

CONCLUSIONES

En conclusión el trabajo realizado cumple con las expectativas esperadas y ofrece al alumno de la Facultad de Ciencias una herramienta capaz de proporcionarle información suficiente para tomar una decisión al elegir los profesores que conformarán su horario de clases. Tras mi estancia en la Facultad de Ciencias y hasta el momento no existen evidencias que respalden lo que se dice de un profesor pero a mi parecer este proyecto puede llegar a ser trascendente si se llegará a implementar en la Facultad en cuestiones de proporcionar información fidedigna y bien fundamentada para que cada semestre los alumnos cuenten con más elementos en la toma de decisiones para elaborar sus horarios de clases.

Respecto a la opinión de los usuarios del SATDEH se concluye que se está de acuerdo en mejorar el SATDEH en cuestiones de vistas, seguridad y administración si este decide ser implementado en alguna institución de nivel superior. Este proyecto queda abierto para quien decida mejorarlo y darle un mayor alcance o simplemente para ser implementado por la sección escolar de la Facultad de Ciencias de la UNAM con las mejoras necesarias para su uso.

En cuanto a mí, este proyecto me permitió extender mis conocimientos en bases de datos y programación, ponerlos en práctica en la solución de un problema de la vida real, además de obtener conocimientos en Ingeniería de Software, en el uso de herramientas como Bizagi, Pencil, Decision Oven y demás. Gracias a este proyecto me ilustre en los Business Intelligence lo cual deja muchos beneficios para mí en el ámbito profesional y en mi carrera resulta ser un plus tener conocimientos en el desarrollo de sistemas que apoyen a la toma de decisiones. Lejos de ser un logro para obtener mi título, el resultado obtenido con esta tesis ha logrado saciar mi curiosidad al preguntarme en el pasado si algún día alguien se animaría a realizar algo de esta índole para hacer mucho más sencilla la elaboración de los horarios de clases, al final termine siendo yo quien decidió echar a andar esta idea lo cual me deja muy satisfecho.

Trabajo Futuro

El desarrollo del SATDEH se hizo pensando en que solo se utilizaría localmente, por lo que los niveles de seguridad no son tan robustos, pero si este sistema llega a ser implementado en una entidad educativa se debe de reforzar dicha seguridad. En cuestiones de administración el SATDEH no cuenta con módulos que realicen el cambio de semestre por lo que debe crearse dicho módulo para facilitar las labores del administrador. El modelo relacional planteado en el capítulo 5 establece que hay una relación de muchos a muchos entre la entidad profesor y la entidad grupo, la implementación del SATDEH dentro de grails no respeta este supuesto, pero sí lo hace la base de datos, debido a que en la Facultad de Ciencias en la mayoría de los casos no se toma en cuenta la evaluación de los ayudantes, sin embargo se tiene conciencia de que hay más de un profesor asociado a un grupo dentro de la Facultad, debido a esto en un futuro el SATDEH puede mejorarse para lograr cumplir con dicho supuesto.

En cuestiones de vistas el SATDEH puede mejorar bastante, desde la presentación de la información hasta los colores de la interfaz, este hecho puede hacer un poco más accesible la información y mucho más agradable el uso del mismo.

Referencias

Capítulo 2

BI fácil, <http://www.businessintelligence.info/definiciones/que-es-business-intelligence.html> 15/03/2013

Crystalis, http://www.crystalisconsulting.com.mx/pagina.php?clave=BOB_J&seccion=2&nav=39 18/03/2013

Gartner, <http://www.gartner.com/technology/reprints.do?id=1-1DZLPF2&ct=130207&st=sb> 05/02/2013

IBM, <http://www-03.ibm.com/software/products/en/category/business-intelligence> 19/03/2014

Kimball, R., Caserta, J.: *The Data Warehouse ETL Toolkit*. Wiley Publishers, pp. 29-32 (2004)

MicroStrategy, <http://www.microstrategy.com/platforms/analytics/business-intelligence> 19/03/2014

Moss, L., Atre, S.: *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*. Addison Wesley, EUA, pp. 27-54 (2003)

PMBok: Guía de los Fundamentos de la Administración de Proyectos. Project Management Institute. 3era edición, p. 5 (2004)

Preguntia, <http://www.preguntia.com/para-que-sirve-un-sistema-de-business-intelligence.html> 15/03/2013

SAS, http://www.sas.com/offices/latinamerica/mexico/noticias/notas_prensa/Visual_Analytics_analizar_visualizar_BigData.html 18/03/2013

White, K.: *Microsoft Business Intelligence for Dummies*. Wiley Publishing, Inc., Indianapolis, Indiana, EUA, pp. 9-56 (2010)

Wikipedia, http://es.wikipedia.org/wiki/Extract,_transform_and_load 20/03/2013

Capítulo 3

Alter, S.: *Decision support systems: Current practice and continuing challenges*. Addison-Wesley Publishing, p. 22 (1980)

Banxia Decision Explorer, <http://www.banxia.com/dexplore/> 02/04/2013

Calderón, J., Muñoz, J., Ramos, Y., Reque, J., Tumbajulca, K.: *TI en el Control de la Gestión*. Curso de Ingeniería de la Información, p. 42 (2009)

LPA, http://www.lpa.co.uk/vsr_det.htm 02/04/2013

Ruiz, A., Hernández, L., Giraldo, W.: *Aplicación de los Sistemas de soporte a la decisión en el comercio electrónico*. Revista Ingeniería e Investigación, Vol. 29, No. 2, agosto, pp. 94-99 (2009)

Capítulo 4

Borea, F., Vélez, I.: *Teoría y técnica de la Decisión*. Departamento de Ciencias Económicas, Universidad Nacional de la Matanza, Argentina, pp. 4-5 (2003)

Caballero, R., Gómez, T., González, M., Hernández, M., Miguel, F., Molina, J., Muñoz, M., Rey, L., Ruiz, F.: *Programación Matemática para Economistas*. Notas de clase. UMA España, pp. 1-2 (2008)

Romero, C.: *Análisis de las Decisiones Multicriterio*. Isdefe, Madrid, España, pp. 14-32 (1996)

Capítulo 5

Limesurvey, <http://www.limesurvey.com/> 25/03/2013

Postgresql, <http://www.postgresql.org.es/> 04/06/2013

Valenzuela-Medina J.: *Evaluación del Desempeño Docente a partir de la Opinión de los Alumnos*, Tesis de Maestría en Psicología, Universidad de Sonora, Hermosillo, México, pp. 5-45 (2002)

Capítulo 6

Bizagi, <http://www.bizagi.com/index.php/es/productos/bizagi-process-modeler> 03/06/2013

Dia, <https://projects.gnome.org/dia/> 13/05/2013

GenerateData, <http://www.generatedata.com/> 08/01/2014

Grails, <http://grails.org/> 03/08/2013

Groovy, <http://groovy.codehaus.org/> 03/08/2013

LimeSurvey, <http://www.limesurvey.com/> 25/03/2013

Moss, L., Atre, S.: *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*. Addison Wesley, EUA, pp 27-54 (2003)

Netbeans, <https://netbeans.org/> 03/08/2013

Notepad, <http://notepad-plus-plus.org/> 02/02/2013

Pencil, <http://pencil.evolus.vn/> 23/03/2013

Postgresql, <http://www.postgresql.org.es/> 04/06/2013

StarUML, <http://staruml.sourceforge.net/en/> 16/05/2013

Bibliografía

Alter, S.: *Decision support systems: Current practice and continuing challenges*. Addison-Wesley Publishing (1980)

Date, C.: *Introducción a los Sistemas de Bases de Datos*. Prentice Hall (2003)

Elmasri, R., Navathe, S.: *Fundamentos de Sistemas de Bases de Datos*. Addison Wesley Iberoamericana (2010)

García-Molina, H., Ullman, J., Widom, J.: *Database Systems: The Complete Book*. Prentice Hall (2009)

Kimball, R., Caserta, J.: *The Data Warehouse ETL Toolkit*. Wiley Publishers (2004)

Moss L, Atre S.: *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*. Addison Wesley, EUA (2003)

Romero, C.: *Análisis de las Decisiones Multicriterio*. Isdefe, Madrid, España (1996)

Ruiz, A., Hernández, L., Giraldo, W.: *Aplicación de los Sistemas de soporte a la decisión en el comercio electrónico*. Revista Ingeniería e Investigación, Vol. 29 No. 2, agosto (2009)

Silberschatz, S., Korth, H., Sudarshan, S.: *Fundamentos de Bases de Datos*. Mc Graw-Hill (2010)

Vitoriano, B.: *Teoría de la Decisión: Decisión con Incertidumbre, Decisión Multicriterio y Teoría de Juegos*. Universidad Complutense Madrid, España (2007)

Wang, R., Reddy, M.P., Kon, H.: *Toward quality data: An attribute-based approach*. Massachusetts Institute of Technology, Cambridge, MA, EUA (1995)

Yacuzzi, E., Rodríguez, V.: *Diseño e Implementación de un Sistema de Apoyo a las Decisiones Basado en el Modelo de Transporte*. Universidad de CEMA, Buenos Aires, Argentina (2005)

Anexo A

En este anexo se encuentra la información sobre la encuesta realizada con LimeSurvey y los resultados obtenidos.

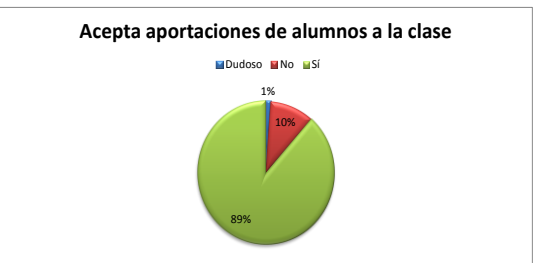
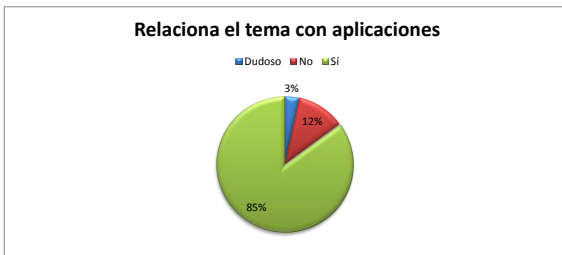
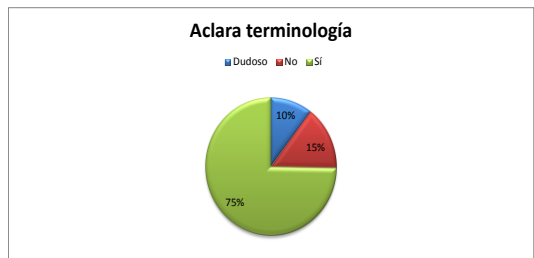
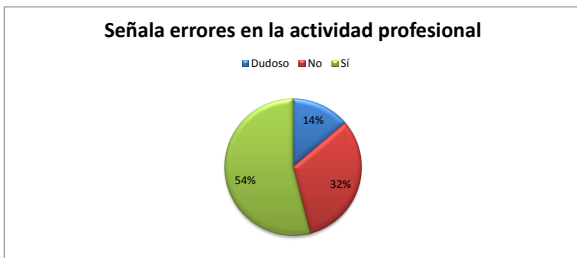
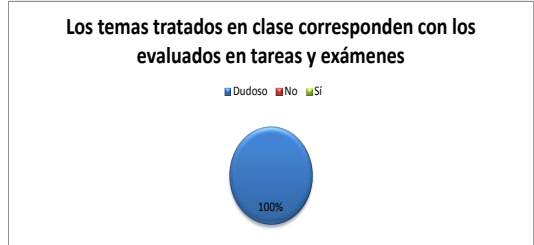
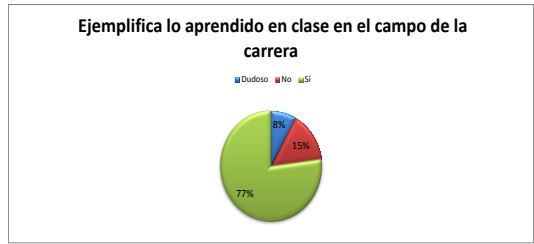
La siguiente es la encuesta que se aplicó a los alumnos de la Facultad de Ciencias a través de la herramienta LimeSurvey.

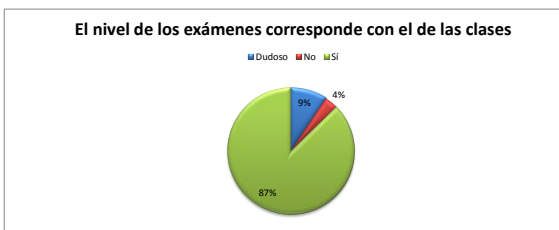
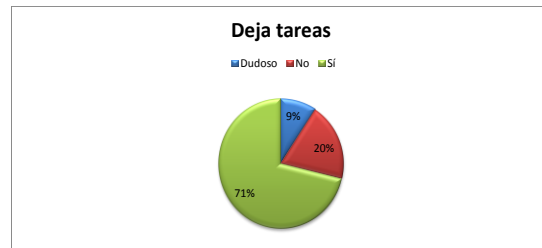
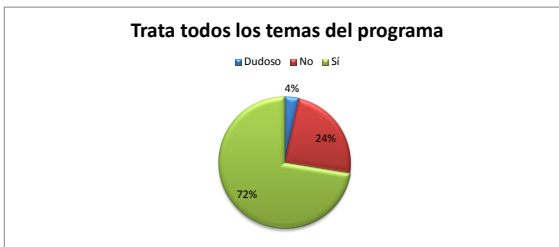
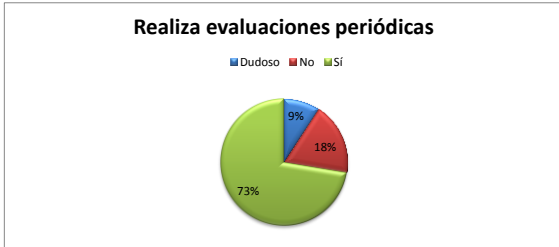
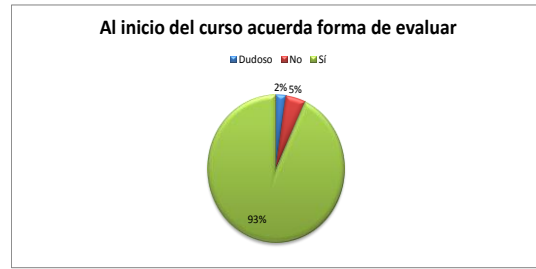
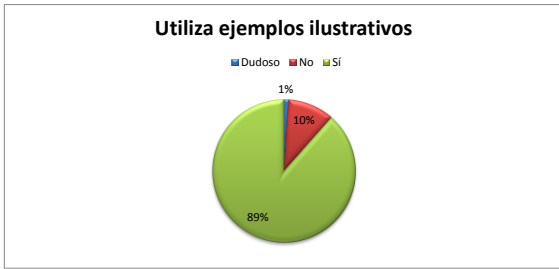
<i>Criterios para la elección de profesores</i>			
Hay 4 preguntas en esta encuesta			
Listado de criterios			
1. Nombre: *			
Por favor, escriba su respuesta aquí:			
2. A continuación se muestra una lista de los posibles criterios para la elección de un profesor.			
Elige alguna de las siguientes opciones para cada criterio:			
• SÍ: cuando tú tomas en cuenta ese criterio para la elección de un profesor.			
• NO: cuando tú no tomas en cuenta ese criterio para la elección de un profesor.			
• DUDOSO: cuando no tienes claro que significa ese criterio.*			
Por favor, seleccione la respuesta apropiada para cada concepto:			
	Sí	Dudoso	No
Inicia clase a la hora	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Usa materiales para exponer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Termina la clase a la hora	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Propicia participación de alumnos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Las clases siguen el orden del programa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Comenta perspectivas sobre la profesión	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Deja ideas claras, no se enreda	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Al iniciar la clase presenta el objetivo de ésta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Responde lo que se le pregunta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Escucha sin interrumpir al alumno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mantiene disciplina sin rigidez	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reconoce esfuerzo y logros	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Organiza equipos de trabajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Trata con respeto y cordialidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Forma equipos de investigación	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ejemplifica lo aprendido en clase en el campo de la carrera	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

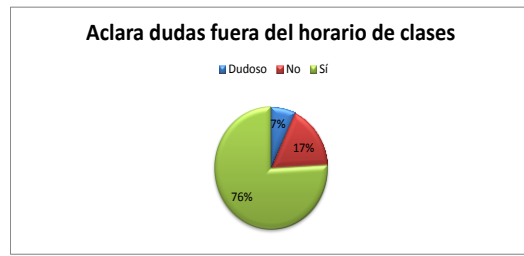
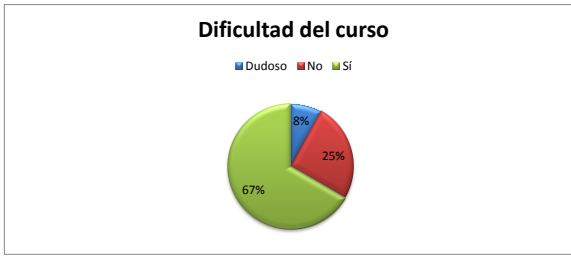
Relaciona el tema con la siguiente clase	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Los temas tratados en clase corresponden con los evaluados en tareas y exámenes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Responde concretamente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mantiene en actividad al grupo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Señala errores en la actividad profesional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aclara terminología	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Relaciona el tema con aplicaciones	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cita fuentes bibliográficas al responder	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Entrega calificaciones el día prometido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acepta aportaciones de alumnos a la clase	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Utiliza ejemplos ilustrativos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Al inicio del curso acuerda forma de evaluar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Realiza evaluaciones periódicas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Logra que los alumnos se interesen en la materia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Trata todos los temas del programa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Comunica con claridad sus conocimientos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Propicia una actitud reflexiva en los alumnos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Domina el tema de la materia que imparte	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Asiste a sus clases	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aclara dudas fuera del horario de clases	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Deja tareas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
El nivel de los exámenes corresponde con el de las clases	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Integra Teoría y práctica	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dificultad del curso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3. ¿Considerarías algún otro criterio para la elección de tus profesores que no se encontraba en la lista? *			
Por favor seleccione sólo una de las siguientes opciones:			
<input type="radio"/> Sí			
<input type="radio"/> No			
4. ¿Cuáles? *			
Sólo conteste esta pregunta si se cumplen las siguientes condiciones:			
° La respuesta fue 'Sí' en la pregunta '3' (¿Considerarías algún otro criterio para la elección de tus profesores que no se encontraba en la lista?)			
Por favor, escriba su respuesta aquí:			
Enviar su encuesta.			
Gracias por completar esta encuesta.			

A continuación se muestran los resultados obtenidos de la encuesta realizada a través de gráficas por criterio, en ellas se puede apreciar el resultado en porcentaje de las tres posibles respuestas, Sí, No y Dudoso, además de los criterios propuestos por algunos de los alumnos que decidieron expresar su punto de vista.









¿Cuáles?	Total
Actividad Laboral y Proyectos de Investigación	1
Checar si el horario en que es impartida la materia me conviene.	1
El Horario en que imparten la clase y que se acomode a mis necesidades.	1
FORMA DE EVALUACION	1
la hora a la cual imparten la clase	1
que sus tareas tengan que ver con cosas que en el futuro usaremos	1
QUE TAN PESADO EVALUA	1
Recomendaciones de amigos que ya han cursado la materia con el profesor.	1
-Recomendaciones de compañeros que tomaron clases con los profesores. -Consideraba las evaluaciones que se hacían en la pagina de la facultad.	1
Si se acomoda a un horario accesible a la hora que imparte la clase Si deja tareas semanales (que tanta y dependiendo cuantas otras materias tenga así aunque esto ya es mas subjetivo)	1
(en blanco)	0
Total general	10

Anexo B

Este anexo contiene el código de todos los SPs que se construyeron para uso del SATDEH, como se explicó en el capítulo 6 muchos de ellos no fueron requeridos gracias al uso de Grails. Los códigos son los siguientes:

- Función que inserta un nuevo profesor en la tabla "profesor" verificando si el profesor ya existe y regresando 1 si la inserción fue exitosa y 0 si no fue así.

```
CREATE OR REPLACE FUNCTION sp_inserta_Profesor(
    rfc1 VARCHAR(10),
    nombre_prof1 VARCHAR (20),
    ap_pat_prof1 VARCHAR(20),
    ap_mat_prof1 VARCHAR(20),
    gradol VARCHAR (40),
    correo_e_prof1 VARCHAR (200),
    activo_inactivo1 int,
    nombramiento1 VARCHAR(50),
    contraseñal VARCHAR(20)
)
RETURNS INTEGER
AS
$$
DECLARE consulta_id INTEGER;
BEGIN
consulta_id= (SELECT rfc FROM profesor WHERE rfc=rfc1);
IF (consulta_id IS NULL) THEN
    INSERT INTO profesor VALUES
(rfc1,nombre_prof1,ap_pat_prof1,ap_mat_prof1,gradol,correo_e_prof1,activo_inactivo1,nomb
ramiento1,contraseñal);
    RETURN 1;
ELSE RETURN 0;
END IF;

END;
$$
LANGUAGE 'plpgsql';
```

- Función desactiva a un profesor en la tabla "profesor" buscando el RFC del profesor que se quiere dar de baja o alta y regresando 1 si se tuvo éxito y 0 si no fue así.

```
CREATE OR REPLACE FUNCTION sp_Alta_Baja_Profesor(
    rfc1 VARCHAR (10)
)
RETURNS INTEGER
AS
$$
DECLARE consulta INTEGER;
BEGIN
consulta= (SELECT activo_inactivo FROM profesor WHERE rfc=rfc1);
IF (consulta =1) THEN
UPDATE profesor SET activo_inactivo=0
WHERE rfc=rfc1;
RETURN 1;
END IF;
IF( consulta=0)THEN
UPDATE profesor SET activo_inactivo=1
WHERE rfc=rfc1;
RETURN 1;
ELSE RETURN 0;
END IF;
END;
$$
LANGUAGE 'plpgsql';
```

- Función que Actualiza el registro de un profesor ya existente en la tabla "profesor" regresando una tupla que muestra la actualización.

```

CREATE OR REPLACE FUNCTION sp_ActualizaProfesor(
    rfc1 VARCHAR(10),
    nombre_prof1 VARCHAR(20),
    ap_pat_prof1 VARCHAR(20),
    ap_mat_prof1 VARCHAR(20),
    gradol VARCHAR(40),
    correo_e_prof1 VARCHAR(200),
    activo_inactivo1 INT,
    nombramiento1 VARCHAR(50),
    contraseñal VARCHAR(20)
)
RETURNS SETOF profesor
AS
$$
DECLARE consulta_id INTEGER;
BEGIN
Update profesor set nombre_prof=nombre_prof1,
    ap_pat_prof=ap_pat_prof1,
    ap_mat_prof=ap_mat_prof1,
    grado=gradol,
    correo_e_prof=correo_e_prof1,
    activo_inactivo=activo_inactivo1,
    nombramiento=nombramiento1,
    contraseña=contraseñal
    WHERE rfc=rfc1;
RETURN QUERY(SELECT * FROM profesor);
END;
$$
LANGUAGE 'plpgsql';

```

- Función que busca un profesor y muestra que materias ha impartido, los grupos y todos los detalles.

```

CREATE OR REPLACE FUNCTION sp_consulta_Profesor(
    rfc1 VARCHAR(9)
)
RETURNS SETOF profes_grupos_materias
AS
$$
BEGIN
RETURN QUERY(SELECT * FROM profes_grupos_materias WHERE rfc=rfc1);
END;
$$
LANGUAGE 'plpgsql';

```

- Función que inserta un nuevo grupo en la tabla "grupo" verificando si el grupo ya existe y regresando 1 si la inserción fue exitosa y 0 si no fue así.

```

CREATE OR REPLACE FUNCTION sp_inserta_Grupo(
    clave_grupo1 INTEGER,
    lunes1 VARCHAR(20),
    martes1 VARCHAR(20),
    miercoles1 VARCHAR(20),
    jueves1 VARCHAR(20),
    viernes1 VARCHAR(20),
    hora_inicio1 VARCHAR(10),
    hora_fin1 VARCHAR(10),
    aula1 VARCHAR(50),
    semestre_imparte1 VARCHAR(20),
    clave_material INTEGER
)
RETURNS INTEGER
AS
$$
DECLARE consulta_id INTEGER;
DECLARE consulta_id1 INTEGER;
BEGIN

```

```

consulta_id1= (SELECT clave_materia FROM materia WHERE clave_materia=clave_material);
IF (consulta_id1 IS NULL) THEN
RETURN 0;
END IF;
consulta_id= (SELECT clave_grupo FROM grupo WHERE clave_grupo=clave_grupo1);
IF (consulta_id IS NULL) THEN
INSERT INTO grupo VALUES
(clave_grupo1,lunes1,martes1,miercoles1,jueves1,viernes1,hora_inicio1,hora_fin1,aula1,se
mestre_imparte1,clave_material);
RETURN 1;
ELSE RETURN 0;
END IF;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que Elimina a un grupo en la tabla "grupo" buscando el id del grupo que se quiere eliminar y regresando 1 si la eliminación fue exitosa y 0 si no fue así.

```

CREATE OR REPLACE FUNCTION sp_Elimina_Grupo(
clave_grupo1 INT
)
RETURNS INTEGER
AS
$$
DECLARE consulta integer;
BEGIN
DELETE FROM grupo WHERE clave_grupo=clave_grupo1;
consulta= (SELECT clave_grupo FROM grupo WHERE clave_grupo=clave_grupo1);
IF( consulta IS NULL)THEN
RETURN 1;
ELSE RETURN 0;
END IF;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que Actualiza el registro de un grupo ya existente en la tabla "grupo" regresando una tupla que muestra la actualización.

```

CREATE OR REPLACE FUNCTION sp_Actualiza_Grupo(
clave_grupo1 INTEGER,
lunes1 VARCHAR(20),
martes1 VARCHAR(20),
miercoles1 VARCHAR(20),
jueves1 VARCHAR(20),
viernes1 VARCHAR(20),
hora_inicio1 VARCHAR(10),
hora_fin1 VARCHAR(10),
aula1 VARCHAR(50),
semestre_imparte1 VARCHAR(20),
clave_material INTEGER
)
RETURNS SETOF grupo
AS
$$
DECLARE consulta_id INTEGER;
BEGIN
Update grupo set
lunes=lunes1,
martes=martes1,
miercoles=miercoles1,
jueves=jueves1,
viernes=viernes1,
hora_inicio=hora_inicio1,
hora_fin=hora_fin1,
aula=aula1,
semestre_imparte=semestre_imparte1,
clave_material=clave_material
WHERE clave_grupo=clave_grupo1;
RETURN QUERY(SELECT * FROM grupo);

```

```
END;
$$
LANGUAGE 'plpgsql';
```

- Función que asocia a un profesor con un grupo e inserta los "rfc", "clave_grupo" y el "tipo_prof" en la relación "prof_grupo" verificando que el profesor y el grupo existan en sus tablas respectivas y que no sea un registro ya existente, regresando 1 si la inserción fue exitosa y 0 si no fue así.

```
CREATE OR REPLACE FUNCTION sp_asocia_Profesor_Grupo(
    rfc1 VARCHAR(10),
    clave_grupo1 INTEGER,
    tipo_prof1 VARCHAR(200)
)
RETURNS INTEGER
AS
$$
DECLARE consulta_id_grupo INTEGER;
DECLARE consulta_id_profesor VARCHAR(10);
DECLARE consulta_id INTEGER;
BEGIN
consulta_id_grupo= (SELECT clave_grupo FROM grupo WHERE clave_grupo=clave_grupo1);
consulta_id_profesor= (SELECT rfc FROM profesor WHERE rfc=rfc1);
consulta_id= (SELECT clave_grupo FROM prof_grupo WHERE rfc=rfc1 AND
clave_grupo=clave_grupo1);
IF (CAST(consulta_id_profesor AS TEXT)='') THEN
RETURN 0;
END IF;
IF (CAST(consulta_id_grupo AS TEXT)='') THEN
RETURN 0;
END IF;
IF (consulta_id IS NULL) THEN
INSERT INTO prof_grupo VALUES (rfc1,clave_grupo1,tipo_prof1);
RETURN 1;
END IF;
END;
$$
LANGUAGE 'plpgsql';
```

- Función que inserta un nuevo Alumno en la tabla "alumno" verificando si el alumno ya existe y regresando 1 si la inserción fue exitosa y 0 si no fue así.

```
CREATE OR REPLACE FUNCTION sp_inserta_Alumno(
    numero_cuenta1 VARCHAR(9),
    nombre_alumno1 VARCHAR (20),
    ap_pat_alumno1 VARCHAR(20),
    ap_mat_alumno1 VARCHAR(20),
    correo_e_alumno1 VARCHAR(150),
    generacion1 INTEGER,
    carrera1 VARCHAR (100),
    activo_inactivo1 INTEGER,
    contrasena1 VARCHAR(20)
)
RETURNS INTEGER
AS
$$
DECLARE consulta_id INTEGER;
BEGIN
consulta_id= (SELECT numero_cuenta FROM alumno WHERE numero_cuenta=numero_cuenta1);
IF (consulta_id IS NULL) THEN
INSERT INTO alumno VALUES
(numero_cuenta1,nombre_alumno1,ap_pat_alumno1,ap_mat_alumno1,correo_e_alumno1,generacion
1,carrera1,activo_inactivo1,contrasena1);
RETURN 1;
ELSE RETURN 0;
END IF;
END;
$$
LANGUAGE 'plpgsql';
```

- Función desactiva a un alumno en la tabla "alumno" buscando el número de cuenta del alumno que se quiere dar de baja o alta y regresando 1 si se tuvo éxito y 0 si no fue así.

```
CREATE OR REPLACE FUNCTION sp_Alta_Baja_Alumno(
    numero_cuental VARCHAR (9)
)
RETURNS INTEGER
AS
$$
DECLARE consulta INTEGER;
BEGIN
consulta= (SELECT activo_inactivo FROM alumno WHERE numero_cuenta=numero_cuental);
IF (consulta =1) THEN
UPDATE alumno SET activo_inactivo=0
WHERE numero_cuenta=numero_cuental;
RETURN 1;
END IF;
IF( consulta=0)THEN
UPDATE alumno SET activo_inactivo=1
WHERE numero_cuenta=numero_cuental;
RETURN 1;
ELSE RETURN 0;
END IF;
END;
$$
LANGUAGE 'plpgsql';
```

- Función que Actualiza el registro de un alumno ya existente en la tabla "alumno" regresando una tupla que muestra la actualización.

```
CREATE OR REPLACE FUNCTION sp_Actualiza_Alumno(
    numero_cuental VARCHAR(9),
    nombre_alumno1 VARCHAR (20),
    ap_pat_alumno1 VARCHAR(20),
    ap_mat_alumno1 VARCHAR(20),
    correo_e_alumno1 VARCHAR(150),
    generacion1 INTEGER,
    carreral VARCHAR (100),
    activo_inactivo1 INTEGER,
    contrasena1 VARCHAR(20)
)
RETURNS SETOF alumno
AS
$$
DECLARE consulta_id INTEGER;
BEGIN
Update alumno set
    nombre_alumno=nombre_alumno1,
    ap_pat_alumno=ap_pat_alumno1,
    ap_mat_alumno=ap_mat_alumno1,
    correo_e_alumno=correo_e_alumno1,
    generacion=generacion1,
    carrera=carreral,
    activo_inactivo=activo_inactivo1,
    contrasena=contrasena1
    WHERE numero_cuenta = numero_cuental;
RETURN QUERY(SELECT * FROM alumno);
END;
$$
LANGUAGE 'plpgsql';
```

- Función que inserta una nueva materia en la tabla "materia" verificando si la materia ya existe y regresando 1 si la inserción fue exitosa y 0 si no fue así.

```
CREATE OR REPLACE FUNCTION sp_inserta_Materia(
    clave_material INTEGER,
    nombre_material VARCHAR(70),
```



```

        creditos1 INTEGER
    )
    RETURNS INTEGER
    AS
    $$
    DECLARE consulta_id INTEGER;
    BEGIN
    consulta_id= (SELECT clave_materia FROM materia WHERE clave_materia=clave_material);
    IF (consulta_id IS NULL) THEN
        INSERT INTO materia VALUES (clave_material,nombre_material, creditos1);
        RETURN 1;
    ELSE RETURN 0;
    END IF;
    END;
    $$
    LANGUAGE 'plpgsql';

```

- Función que Elimina a una materia en la tabla "materia" buscando el la clave de la materia que se quiere eliminar y regresando 1 si la eliminación fue exitosa y 0 si no fue así.

```

CREATE OR REPLACE FUNCTION sp_Elimina_Materia(
    clave_material INTEGER
)
    RETURNS INTEGER
    AS
    $$
    DECLARE consulta INTEGER;
    BEGIN
    DELETE FROM materia WHERE clave_materia=clave_material;
    consulta= (SELECT clave_materia FROM materia WHERE clave_materia=clave_material);
    IF( consulta IS NULL)THEN
        RETURN 1;
    ELSE RETURN 0;
    END IF;
    END;
    $$
    LANGUAGE 'plpgsql';

```

- Función que Actualiza el registro de una materia ya existente en la tabla "materia" regresando una tupla que muestra la actualización.

```

CREATE OR REPLACE FUNCTION sp_Actualiza_Materia(
    clave_material INTEGER,
    nombre_material VARCHAR(70),
    creditos1 INTEGER
)
    RETURNS SETOF materia
    AS
    $$
    DECLARE consulta_id INTEGER;
    BEGIN
    Update materia set
        nombre_materia=nombre_material,
        creditos=creditos1
        WHERE clave_materia=clave_material;
    RETURN QUERY(SELECT * FROM materia);
    END;
    $$
    LANGUAGE 'plpgsql';

```

- Función que asocia un grupo con un alumno e inserta la clave del grupo y el número de cuenta en la relación "horario" verificando que el alumno y el grupo existan en sus tablas respectivas, regresando 1 si la asociación fue exitosa y 0 si no fue así.

```

CREATE OR REPLACE FUNCTION sp_Asocia_Alumno_Grupo(
    clave_grupo1 INTEGER,
    numero_cuenta1 VARCHAR(9)
)
RETURNS INTEGER
AS
$$
DECLARE consulta_id_grupo INTEGER;
DECLARE consulta_id_alumno VARCHAR(9);
DECLARE consulta_id INTEGER;
BEGIN
consulta_id_grupo= (SELECT clave_grupo FROM grupo WHERE clave_grupo=clave_grupo1);
consulta_id_alumno= (SELECT numero_cuenta FROM alumno WHERE
numero_cuenta=numero_cuenta1);
consulta_id= (SELECT clave_grupo FROM horario WHERE numero_cuenta=numero_cuenta1 AND
clave_grupo=clave_grupo1);
IF (CAST(consulta_id_alumno AS TEXT)='') THEN
RETURN 0;
END IF;
IF (CAST(consulta_id_grupo AS TEXT)='') THEN
RETURN 0;
END IF;
IF (consulta_id IS NULL) THEN
INSERT INTO horario VALUES (clave_grupo1,numero_cuenta1);
RETURN 1;
END IF;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que inserta una evaluación a la tabla "evaluacion" verificando que el alumno este asociado a un grupo, regresando 1 si la asociación fue exitosa y 0 si no fue así.

```

CREATE OR REPLACE FUNCTION sp_Inserta_Evaluacion(
    clave_grupo1 INT,
    numero_cuenta1 VARCHAR(9),
    comentario1 VARCHAR (3000),
    comenta_perspectivas1 REAL,
    deja_ideas1 REAL,
    responde_loque1 REAL,
    escuchal REAL,
    reconoce1 REAL,
    trata_respeto1 REAL,
    ejemplificacional REAL,
    temas_tratados1 REAL,
    responde_concretamente1 REAL,
    mantiene_actividad1 REAL,
    aclara_terminologia1 REAL,
    relaciona_tema_aplicacional REAL,
    acepta_aportaciones1 REAL,
    utiliza_ejemplos1 REAL,
    acuerda_forma_evaluar1 REAL,
    evaluaciones_periodicas1 REAL,
    alumnos_interesen1 REAL,
    todos_temas_programa1 REAL,
    comunica_claridad_conocimientos1 REAL,
    domina_temal REAL,
    asiste_clases1 REAL,
    aclara_dudas1 REAL,
    deja_tareas1 REAL,
    nivel_examenes_clases1 REAL,
    integra_teoría_practical REAL
)
RETURNS INTEGER
AS

```

```

$$
DECLARE consulta_id INTEGER;
BEGIN
consulta_id= (SELECT clave_grupo FROM horario WHERE clave_grupo=clave_grupo1 AND
numero_cuenta=numero_cuenta1);
IF (CAST(consulta_id AS TEXT)='') THEN
RETURN 0;
END IF;
INSERT INTO evaluacion VALUES
(clave_grupo1,numero_cuenta1,comentariol,comenta_perspectivas1,deja_ideas1,responde_loqu
e1,escuchal,reconoce1,
trata_respeto1,
ejemplifical,
temas_tratados1,
responde_concretamentel,
mantiene_actividadl,
aclara_terminologial,
relaciona_tema_aplical,
acepta_aportaciones1,
utiliza_ejemplos1,
acuerda_forma_evaluar1,
evaluaciones_periodicas1,
alumnos_interesen1,
todos_temas_programa1,
comunica_claridad_conocimientos1,
domina_tema1,
asiste_clases1,
aclara_dudas1,
deja_tareas1,
nivel_exámenes_clases1,
integra_teoría_practical);
RETURN 1;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que Elimina a una evaluación en la tabla "evaluacion" buscando la clave del grupo y el número de cuenta y elimina la evaluación de esa tupla, regresando 1 si la eliminación fue exitosa y 0 si no fue así.

```

CREATE OR REPLACE FUNCTION sp_Elimina_Evaluacion(
clave_grupo1 INTEGER,
numero_cuenta1 VARCHAR(9)
)
RETURNS INTEGER
AS
$$
DECLARE consulta INTEGER;
BEGIN
DELETE FROM evaluacion WHERE clave_grupo=clave_grupo1 AND numero_cuenta=numero_cuenta1;
consulta= (SELECT clave_grupo FROM evaluacion WHERE clave_grupo=clave_grupo1 AND
numero_cuenta=numero_cuenta1);
IF( consulta IS NULL)THEN
RETURN 1;
ELSE RETURN 0;
END IF;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que muestra la lista de los horarios de la institución educativa según el semestre que se quiera.

```

CREATE OR REPLACE FUNCTION sp_horarios(
semestre_se_imparte VARCHAR(255)
)
RETURNS TABLE (
rfc1 VARCHAR(10),
nombre_prof1 VARCHAR(20),
ap_pat_prof1 VARCHAR(20),

```

```

        ap_mat_prof1 VARCHAR(20),
        activo_inactivo1 INTEGER,
        tipo_prof1 VARCHAR (30),
        clave_grupo1 INTEGER,
        lunes1 VARCHAR(20),
        martes1 VARCHAR(20),
        miercoles1 VARCHAR(20),
        jueves1 VARCHAR(20),
        viernes1 VARCHAR(20),
        hora_inicio1 VARCHAR(10),
        hora_fin1 VARCHAR(10),
        aula1 VARCHAR(50),
        semestre_imparte1 VARCHAR(20),
        clave_material INTEGER,
        nombre_material VARCHAR(255),
        comentario1 VARCHAR(255),
        creditos1 INTEGER
    )
AS
$$
DECLARE tupla profes_grupos_materias%ROWTYPE;
BEGIN
    FOR tupla IN
    SELECT * FROM profes_grupos_materias WHERE semestre_imparte = semestre_se_imparte
    LOOP
        rfc1 := tupla.rfc;
        nombre_prof1 := tupla.nombre_prof;
        ap_pat_prof1 := tupla.ap_pat_prof;
        ap_mat_prof1 := tupla.ap_mat_prof;
        activo_inactivo1 := tupla.activo_inactivo;
        tipo_prof1 := tupla.tipo_prof;
        clave_grupo1 := tupla.clave_grupo;
        lunes1 := tupla.lunes;
        martes1 := tupla.martes;
        miercoles1 := tupla.miercoles;
        jueves1 := tupla.jueves;
        viernes1 := tupla.viernes;
        hora_inicio1 := tupla.hora_inicio;
        hora_fin1 := tupla.hora_fin;
        aula1 := tupla.aula;
        semestre_imparte1 := tupla.semestre_imparte;
        clave_material := tupla.clave_materia;
        nombre_material := tupla.nombre_materia;
        creditos1 := tupla.creditos;
    RETURN NEXT;
    END LOOP;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que realiza la búsqueda de un profesor por nombre aproximado y regresa la tabla de las tuplas semejantes.

```

CREATE OR REPLACE FUNCTION sp_busca_Profesor_nombre(
    nombre VARCHAR(255)
)
RETURNS TABLE (
    rfc1 VARCHAR(10),
    nombre_prof1 VARCHAR(20),
    ap_pat_prof1 VARCHAR(20),
    ap_mat_prof1 VARCHAR(20),
    activo_inactivo1 INTEGER,
    tipo_prof1 VARCHAR (30),
    clave_grupo1 INTEGER,
    lunes1 VARCHAR(20),
    martes1 VARCHAR(20),
    miercoles1 VARCHAR(20),
    jueves1 VARCHAR(20),
    viernes1 VARCHAR(20),
    hora_inicio1 VARCHAR(10),
    hora_fin1 VARCHAR(10),
    aula1 VARCHAR(50),
    semestre_imparte1 VARCHAR(20),
    clave_material INTEGER,
    nombre_material VARCHAR(255),

```

```

        comentario1 VARCHAR(255),
        creditos1 INTEGER
    )
AS
$$
DECLARE tupla profes_grupos_materias%ROWTYPE;
BEGIN
    FOR tupla IN
    SELECT * FROM profes_grupos_materias WHERE nombre_prof LIKE nombre || '%'
LOOP
    rfc1 := tupla.rfc;
    nombre_prof1 := tupla.nombre_prof;
    ap_pat_prof1 := tupla.ap_pat_prof;
    ap_mat_prof1 := tupla.ap_mat_prof;
    activo_inactivo1 := tupla.activo_inactivo;
    tipo_prof1 := tupla.tipo_prof;
    clave_grupo1 := tupla.clave_grupo;
    lunes1 := tupla.lunes;
    martes1 := tupla.martes;
    miercoles1 := tupla.miercoles;
    jueves1 := tupla.jueves;
    viernes1 := tupla.viernes;
    hora_inicio1 := tupla.hora_inicio;
    hora_fin1 := tupla.hora_fin;
    aula1 := tupla.aula;
    semestre_imparte1 := tupla.semestre_imparte;
    clave_material := tupla.clave_material;
    nombre_material := tupla.nombre_material;
    creditos1 := tupla.creditos;
RETURN NEXT;
END LOOP;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que realiza la búsqueda de un profesor por nombre de la materia aproximado y regresa la tabla de las tuplas semejantes.

```

CREATE OR REPLACE FUNCTION sp_busca_Profesor_materia(
    nombre VARCHAR(255)
)
RETURNS TABLE (
    rfc1 VARCHAR(10),
    nombre_prof1 VARCHAR(20),
    ap_pat_prof1 VARCHAR(20),
    ap_mat_prof1 VARCHAR(20),
    activo_inactivo1 INTEGER,
    tipo_prof1 VARCHAR(30),
    clave_grupo1 INTEGER,
    lunes1 VARCHAR(20),
    martes1 VARCHAR(20),
    miercoles1 VARCHAR(20),
    jueves1 VARCHAR(20),
    viernes1 VARCHAR(20),
    hora_inicio1 VARCHAR(10),
    hora_fin1 VARCHAR(10),
    aula1 VARCHAR(50),
    semestre_imparte1 VARCHAR(20),
    clave_material INTEGER,
    nombre_material VARCHAR(255),
    comentario1 VARCHAR(255),
    creditos1 INTEGER
)
AS
$$
DECLARE tupla profes_grupos_materias%ROWTYPE;
BEGIN
    FOR tupla IN
    SELECT * FROM profes_grupos_materias WHERE nombre_materia LIKE nombre || '%'
LOOP
    rfc1 := tupla.rfc;
    nombre_prof1 := tupla.nombre_prof;
    ap_pat_prof1 := tupla.ap_pat_prof;
    ap_mat_prof1 := tupla.ap_mat_prof;
    activo_inactivo1 := tupla.activo_inactivo;

```

```

        tipo_prof1 := tupla.tipo_prof;
        clave_grupo1 := tupla.clave_grupo;
        lunes1 := tupla.lunes;
        martes1 := tupla.martes;
        miercoles1 := tupla.miercoles;
        jueves1 := tupla.jueves;
        vienes1 := tupla.viernes;
        hora_inicio1 := tupla.hora_inicio;
        hora_fin1 := tupla.hora_fin;
        aula1 := tupla.aula;
        semestre_imparte1 := tupla.semestre_imparte;
        clave_material := tupla.clave_material;
        nombre_material := tupla.nombre_material;
        creditos1 := tupla.creditos;
RETURN NEXT;
END LOOP;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que realiza la búsqueda de un profesor por nombre de la materia aproximado y regresa la tabla de las tuplas semejantes.

```

CREATE OR REPLACE FUNCTION sp_busca_Profesor_grupo(
    grupo INTEGER
)
RETURNS TABLE (
    rfcl VARCHAR(10),
    nombre_prof1 VARCHAR(20),
    ap_pat_prof1 VARCHAR(20),
    ap_mat_prof1 VARCHAR(20),
    activo_inactivo1 INTEGER,
    tipo_prof1 VARCHAR(30),
    clave_grupo1 INTEGER,
    lunes1 VARCHAR(20),
    martes1 VARCHAR(20),
    miercoles1 VARCHAR(20),
    jueves1 VARCHAR(20),
    vienes1 VARCHAR(20),
    hora_inicio1 VARCHAR(10),
    hora_fin1 VARCHAR(10),
    aula1 VARCHAR(50),
    semestre_imparte1 VARCHAR(20),
    clave_material INTEGER,
    nombre_material VARCHAR(255),
    comentario1 VARCHAR(255),
    creditos1 INTEGER
)
AS
$$
DECLARE tupla profes_grupos_materias%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT * FROM profes_grupos_materias WHERE clave_grupo = grupo
    LOOP
        rfcl := tupla.rfc;
        nombre_prof1 := tupla.nombre_prof;
        ap_pat_prof1 := tupla.ap_pat_prof;
        ap_mat_prof1 := tupla.ap_mat_prof;
        activo_inactivo1 := tupla.activo_inactivo;
        tipo_prof1 := tupla.tipo_prof;
        clave_grupo1 := tupla.clave_grupo;
        lunes1 := tupla.lunes;
        martes1 := tupla.martes;
        miercoles1 := tupla.miercoles;
        jueves1 := tupla.jueves;
        vienes1 := tupla.viernes;
        hora_inicio1 := tupla.hora_inicio;
        hora_fin1 := tupla.hora_fin;
        aula1 := tupla.aula;
        semestre_imparte1 := tupla.semestre_imparte;
        clave_material := tupla.clave_material;
        nombre_material := tupla.nombre_material;
        creditos1 := tupla.creditos;
RETURN NEXT;

```

```

END LOOP;
END;
$$
LANGUAGE 'plpgsql';

```

- Función que calcula la tabla de pesos según la Teoría de las Decisiones.

```

CREATE OR REPLACE FUNCTION sp_pesos_rank(
    hora_inicio1 VARCHAR(255),
    semestre_imparte1 VARCHAR(255),
    nombre_material VARCHAR(255),
    mas_preferido_1 VARCHAR(255),
    mas_preferido_2 VARCHAR(255),
    mas_preferido_3 VARCHAR(255),
    mas_preferido_4 VARCHAR(255),
    mas_preferido_5 VARCHAR(255)
)
RETURNS TABLE (
    rfc VARCHAR(255),
    comenta_perspectivas NUMERIC,
    deja_ideas NUMERIC,
    responde_loque NUMERIC,
    escucha NUMERIC,
    reconoce NUMERIC,
    trata_respeto NUMERIC,
    ejemplifica NUMERIC,
    temas_tratados NUMERIC,
    responde_concretamente NUMERIC,
    mantiene_actividad NUMERIC,
    aclara_terminologia NUMERIC,
    relaciona_tema_aplica NUMERIC,
    acepta_aportaciones NUMERIC,
    utiliza_ejemplos NUMERIC,
    acuerda_forma_evaluar NUMERIC,
    evaluaciones_periodicas NUMERIC,
    alumnos_interesen NUMERIC,
    todos_temas_programa NUMERIC,
    comunica_claridad_conocimientos NUMERIC,
    domina_tema NUMERIC,
    asiste_clases NUMERIC,
    aclara_dudas NUMERIC,
    deja_tareas NUMERIC,
    nivel_exámenes_clases NUMERIC,
    integra_teoría_práctica NUMERIC
)
AS
$$
DECLARE tupla promedio%ROWTYPE;
DECLARE suma NUMERIC(5,3);
DECLARE j INTEGER;
DECLARE contador INTEGER;
DECLARE contador1 INTEGER;
DECLARE contador2 INTEGER;
DECLARE contador3 INTEGER;
DECLARE contador4 INTEGER;
DECLARE contador5 INTEGER;
DECLARE Lista_pesos NUMERIC(5,3) ARRAY [25];
DECLARE Lista_criterios VARCHAR ARRAY [25];
DECLARE Lista_preferencias NUMERIC(5,3) ARRAY [25];
BEGIN

Lista_criterios = ARRAY
['comenta_perspectivas','deja_ideas','responde_loque','escucha','reconoce','trata_respeto',
'o','ejemplifica','temas_tratados','responde_concretamente','mantiene_actividad',
'aclara_terminologia','relaciona_tema_aplica','acepta_aportaciones','utiliza_ejem
plos','acuerda_forma_evaluar','evaluaciones_periodicas','alumnos_interesen','todos_temas
_programa',
'comunica_claridad_conocimientos','domina_tema','asiste_clases','aclara_dudas','d
eja_tareas','nivel_exámenes_clases','integra_teoría_práctica'];

contador=5;
FOR i in 1..25

```

```

loop
  IF Lista_criterios[i]= mas_preferido_1 THEN
    contador1=i;
  end IF;
  i=i+1;
end loop;

FOR i in 1..25
loop
  IF Lista_criterios[i]= mas_preferido_2 THEN
    contador2=i;
  end IF;
  i=i+1;
end loop;

FOR i in 1..25
loop
  IF Lista_criterios[i]= mas_preferido_3 THEN
    contador3=i;
  end IF;
  i=i+1;
end loop;

FOR i in 1..25
loop
  IF Lista_criterios[i]= mas_preferido_4 THEN
    contador4=i;
  end IF;
  i=i+1;
end loop;

FOR i in 1..25
loop
  IF Lista_criterios[i]= mas_preferido_5 THEN
    contador5=i;
  end IF;
  i=i+1;
end loop;

Lista_preferencias[contador1]=1;
Lista_preferencias[contador2]=2;
Lista_preferencias[contador3]=3;
Lista_preferencias[contador4]=4;
Lista_preferencias[contador5]=5;

FOR i in 1..25
loop
  IF Lista_criterios[i]= null THEN
    Lista_criterios[i]=0;
  end IF;
  i=i+1;
end loop;

j=contador;

suma = 0;

FOR i IN 1..j
loop
  suma=suma+(1.0/i);
  i=i+1;
end loop;

FOR k IN 1..25
loop
  IF Lista_preferencias[k]!=0 THEN
    Lista_pesos[k]=(1.0/Lista_preferencias[k])/suma;
  end IF;
  k=k+1;
end loop;

FOR tupla IN
  SELECT * FROM (SELECT W.rfc,AVG(W.comenta_perspectivas)as a,AVG(W.deja_ideas)as
b,AVG(W.responde_loque)as c,AVG(W.escucha)as d,AVG(W.reconoce) as e
,AVG(W.trata_respeto) as f,AVG(W.ejemplifica) as g,

```



```

        AVG(W.temas_tratados) as h,AVG(W.responde_concretamente) as
i,AVG(W.mantiene_actividad) as j,AVG(W.aclara_terminologia) as
k,AVG(W.relaciona_tema_aplica) as l,AVG(W.acepta_aportaciones) as m,
        AVG(W.utiliza_ejemplos) as n,AVG(W.acuerda_forma_evaluar) as
o,AVG(W.evaluaciones_periodicas) as p,AVG(W.alumnos_interesen) as
q,AVG(W.todos_temas_programa) as r,AVG(W.comunica_claridad_conocimientos) as s,
        AVG(W.domina_tema) as t,AVG(W.asiste_clases) as u,AVG(W.aclara_dudas) as
v,AVG(W.deja_tareas) as qq,AVG(W.nivel_examenes_clases) as
y,AVG(W.integra_teoría_practica) as z FROM vista_rank AS W WHERE
W.hora_inicio=hora_inicio1 AND W.semestre_imparte=semestre_imparte1 AND
W.nombre_materia=nombre_material GROUP BY W.rfc) AS R

```

LOOP

```

        rfc      := tupla.rfc;
comenta_perspectivas := tupla.a*Lista_pesos[1];
deja_ideas      := tupla.b*Lista_pesos[2];
responde_loque  := tupla.c*Lista_pesos[3];
escucha        := tupla.d*Lista_pesos[4];
reconoce        := tupla.e*Lista_pesos[5];
trata_respeto  := tupla.f*Lista_pesos[6];
ejemplifica    := tupla.g*Lista_pesos[7];
temas_tratados := tupla.h*Lista_pesos[8];
responde_concretamente := tupla.i*Lista_pesos[9];
mantiene_actividad := tupla.j*Lista_pesos[10];
aclara_terminologia := tupla.k*Lista_pesos[11];
relaciona_tema_aplica := tupla.l*Lista_pesos[12];
acepta_aportaciones := tupla.m*Lista_pesos[13];
utiliza_ejemplos := tupla.n*Lista_pesos[14];
acuerda_forma_evaluar := tupla.o*Lista_pesos[15];
evaluaciones_periodicas := tupla.p*Lista_pesos[16];
alumnos_interesen := tupla.q*Lista_pesos[17];
todos_temas_programa := tupla.r*Lista_pesos[18];
comunica_claridad_conocimientos := tupla.s*Lista_pesos[19];
domina_tema := tupla.t*Lista_pesos[20];
asiste_clases := tupla.u*Lista_pesos[21];
aclara_dudas := tupla.v*Lista_pesos[22];
deja_tareas := tupla.qq*Lista_pesos[23];
nivel_examenes_clases := tupla.y*Lista_pesos[24];
integra_teoría_practica := tupla.z*Lista_pesos[25];
RETURN NEXT;
END LOOP;

END;
$$
LANGUAGE 'plpgsql';

```

- Esta función ordena según el orden de preferencia las tuplas arrojadas por la función `sp_rank`.

```

CREATE OR REPLACE FUNCTION sp_sort_rank(
        hora_inicio1 VARCHAR(255),
        semestre_imparte1 VARCHAR(255),
        nombre_material VARCHAR(255),
        mas_preferido_1 VARCHAR(255),
        mas_preferido_2 VARCHAR(255),
        mas_preferido_3 VARCHAR(255),
        mas_preferido_4 VARCHAR(255),
        mas_preferido_5 VARCHAR(255)
)
RETURNS TABLE (
        rfc VARCHAR(255),
        comenta_perspectivas NUMERIC,
        deja_ideas NUMERIC,
        responde_loque NUMERIC,
        escucha NUMERIC,
        reconoce NUMERIC,
        trata_respeto NUMERIC,
        ejemplifica NUMERIC,
        temas_tratados NUMERIC,
        responde_concretamente NUMERIC,
        mantiene_actividad NUMERIC,
        aclara_terminologia NUMERIC,

```

```

relaciona_tema_aplica NUMERIC,
acepta_aportaciones NUMERIC,
utiliza_ejemplos NUMERIC,
acuerda_forma_evaluar NUMERIC,
evaluaciones_periodicas NUMERIC,
alumnos_interesen NUMERIC,
todos_temas_programa NUMERIC,
comunica_claridad_conocimientos NUMERIC,
domina_tema NUMERIC,
asiste_clases NUMERIC,
aclara_dudas NUMERIC,
deja_tareas NUMERIC,
nivel_exámenes_clases NUMERIC,
integra_teoría_práctica NUMERIC
)
AS
$$
DECLARE tupla promedio%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT
            *
            FROM
            sp_pesos_rank(hora_inicio1, semestre_imparte1, nombre_material, mas_preferido_1, mas_preferido_2, mas_preferido_3, mas_preferido_4, mas_preferido_5)
            ORDER BY
            mas_preferido_1, mas_preferido_2, mas_preferido_3, mas_preferido_4, mas_preferido_5 DESC
        LOOP
            rfc := tupla.rfc;
            comenta_perspectivas := tupla.a;
            deja_ideas := tupla.b;
            responde_loque := tupla.c;
            escucha := tupla.d;
            reconoce := tupla.e;
            trata_respeto := tupla.f;
            ejemplifica := tupla.g;
            temas_tratados := tupla.h;
            responde_concretamente := tupla.i;
            mantiene_actividad := tupla.j;
            aclara_terminología := tupla.k;
            relaciona_tema_aplica := tupla.l;
            acepta_aportaciones := tupla.m;
            utiliza_ejemplos := tupla.n;
            acuerda_forma_evaluar := tupla.o;
            evaluaciones_periodicas := tupla.p;
            alumnos_interesen := tupla.q;
            todos_temas_programa := tupla.r;
            comunica_claridad_conocimientos := tupla.s;
            domina_tema := tupla.t;
            asiste_clases := tupla.u;
            aclara_dudas := tupla.v;
            deja_tareas := tupla.qq;
            nivel_exámenes_clases := tupla.y;
            integra_teoría_práctica := tupla.z;
            RETURN NEXT;
        END LOOP;

    END;
    $$
    LANGUAGE 'plpgsql';

```

- Esta función calcula la evaluación global promedio de un profesor recibiendo como parámetro el RFC del mismo.

```

CREATE OR REPLACE FUNCTION sp_evaluaciones_Profesor_global(
    rfc1 VARCHAR(255)
)
RETURNS TABLE (
    rfc VARCHAR(255),
    comenta_perspectivas NUMERIC,
    deja_ideas NUMERIC,
    responde_loque NUMERIC,
    escucha NUMERIC,
    reconoce NUMERIC,
    trata_respeto NUMERIC,

```

```

ejemplifica NUMERIC,
temas_tratados NUMERIC,
responde_concretamente NUMERIC,
mantiene_actividad NUMERIC,
aclara_terminologia NUMERIC,
relaciona_tema_aplica NUMERIC,
acepta_aportaciones NUMERIC,
utiliza_ejemplos NUMERIC,
acuierda_forma_evaluar NUMERIC,
evaluaciones_periodicas NUMERIC,
alumnos_interesen NUMERIC,
todos_temas_programa NUMERIC,
comunica_claridad_conocimientos NUMERIC,
domina_tema NUMERIC,
asiste_clases NUMERIC,
aclara_dudas NUMERIC,
deja_tareas NUMERIC,
nivel_examenes_clases NUMERIC,
integra_teoría_practica NUMERIC
)
AS
$$
DECLARE tupla promedio%ROWTYPE;
BEGIN
FOR tupla IN
SELECT * FROM promedio AS R WHERE R.rfc=rfc1
LOOP
rfc := tupla.rfc;
comenta_perspectivas := tupla.a;
deja_ideas := tupla.b;
responde_loque := tupla.c;
escucha := tupla.d;
reconoce := tupla.e;
trata_respeto := tupla.f;
ejemplifica := tupla.g;
temas_tratados := tupla.h;
responde_concretamente := tupla.i;
mantiene_actividad := tupla.j;
aclara_terminologia := tupla.k;
relaciona_tema_aplica := tupla.l;
acepta_aportaciones := tupla.m;
utiliza_ejemplos := tupla.n;
acuierda_forma_evaluar := tupla.o;
evaluaciones_periodicas := tupla.p;
alumnos_interesen := tupla.q;
todos_temas_programa := tupla.r;
comunica_claridad_conocimientos := tupla.s;
domina_tema := tupla.t;
asiste_clases := tupla.u;
aclara_dudas := tupla.v;
deja_tareas := tupla.qq;
nivel_examenes_clases := tupla.y;
integra_teoría_practica := tupla.z;
RETURN NEXT;
END LOOP;
END;
$$
LANGUAGE 'plpgsql';

```

- Esta función calcula la evaluación global promedio por materia de un profesor recibiendo como parámetro el RFC del mismo.

```

CREATE OR REPLACE FUNCTION sp_evaluaciones_Profesor_global_materias(
rfc1 VARCHAR(255)
)
RETURNS TABLE (
nombre_materia VARCHAR(255),
comenta_perspectivas NUMERIC,
deja_ideas NUMERIC,
responde_loque NUMERIC,
escucha NUMERIC,
reconoce NUMERIC,
trata_respeto NUMERIC,
ejemplifica NUMERIC,
temas_tratados NUMERIC,

```

```

    responde_concretamente NUMERIC,
    mantiene_actividad NUMERIC,
    aclara_terminologia NUMERIC,
    relaciona_tema_aplica NUMERIC,
    acepta_aportaciones NUMERIC,
    utiliza_ejemplos NUMERIC,
    acuerda_forma_evaluar NUMERIC,
    evaluaciones_periodicas NUMERIC,
    alumnos_interesen NUMERIC,
    todos_temas_programa NUMERIC,
    comunica_claridad_conocimientos NUMERIC,
    domina_tema NUMERIC,
    asiste_clases NUMERIC,
    aclara_dudas NUMERIC,
    deja_tareas NUMERIC,
    nivel_examenes_clases NUMERIC,
    integra_teoría_practica NUMERIC
)
AS
$$
DECLARE tupla promedio_materias%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT W.nombre_materia,AVG(W.comenta_perspectivas)as a,AVG(W.deja_ideas)as
b,AVG(W.responde_loque)as c,AVG(W.escucha)as d,AVG(W.reconoce) as e
,AVG(W.trata_respeto) as f,AVG(W.ejemplifica) as g,
        AVG(W.temas_tratados) as h,AVG(W.responde_concretamente) as
i,AVG(W.mantiene_actividad) as j,AVG(W.aclara_terminologia) as
k,AVG(W.relaciona_tema_aplica) as l,AVG(W.acepta_aportaciones) as m,
        AVG(W.utiliza_ejemplos) as n,AVG(W.acuerda_forma_evaluar) as
o,AVG(W.evaluaciones_periodicas) as p,AVG(W.alumnos_interesen) as
q,AVG(W.todos_temas_programa) as r,AVG(W.comunica_claridad_conocimientos) as s,
        AVG(W.domina_tema) as t,AVG(W.asiste_clases) as u,AVG(W.aclara_dudas) as
v,AVG(W.deja_tareas) as qq,AVG(W.nivel_examenes_clases) as
y,AVG(W.integra_teoría_practica) as z FROM vista_rank AS W WHERE W.rfc=rfc1 GROUP BY
W.nombre_materia
LOOP
    nombre_materia := tupla.nombre_materia;
    comenta_perspectivas := tupla.a;
    deja_ideas := tupla.b;
    responde_loque := tupla.c;
    escucha := tupla.d;
    reconoce := tupla.e;
    trata_respeto := tupla.f;
    ejemplifica := tupla.g;
    temas_tratados := tupla.h;
    responde_concretamente := tupla.i;
    mantiene_actividad := tupla.j;
    aclara_terminologia := tupla.k;
    relaciona_tema_aplica := tupla.l;
    acepta_aportaciones := tupla.m;
    utiliza_ejemplos := tupla.n;
    acuerda_forma_evaluar := tupla.o;
    evaluaciones_periodicas := tupla.p;
    alumnos_interesen := tupla.q;
    todos_temas_programa := tupla.r;
    comunica_claridad_conocimientos := tupla.s;
    domina_tema := tupla.t;
    asiste_clases := tupla.u;
    aclara_dudas := tupla.v;
    deja_tareas := tupla.qq;
    nivel_examenes_clases := tupla.y;
    integra_teoría_practica := tupla.z;
RETURN NEXT;
END LOOP;
END;
$$
LANGUAGE 'plpgsql';

```

- Esta función muestra todos los comentarios hechos a un profesor en toda su historia como académico.

```

CREATE OR REPLACE FUNCTION sp_comentarios_Profesor_global(
    rfc1 VARCHAR(255)
)

```

```

RETURNS TABLE (
    comentario VARCHAR(255)
)
AS
$$
DECLARE tupla vista_rank%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT * FROM vista_rank as A WHERE A.rfc=rfc1
    LOOP
        comentario := tupla.comentario;
    RETURN NEXT;
    END LOOP;
END;
$$
LANGUAGE 'plpgsql';

```

- Esta función muestra todos los comentarios por materia hechos a un profesor en toda su historia como académico.

```

CREATE OR REPLACE FUNCTION sp_comentarios_Profesor_materia(
    rfc1 VARCHAR(255)
)
RETURNS TABLE (
    nombre_materia VARCHAR(255),
    comentario VARCHAR(255)
)
AS
$$
DECLARE tupla vista_rank%ROWTYPE;
BEGIN
    FOR tupla IN
        SELECT * FROM vista_rank AS A WHERE A.rfc=rfc1 ORDER BY A.nombre_materia
    LOOP
        nombre_materia := tupla.nombre_materia;
        comentario := tupla.comentario;
    RETURN NEXT;
    END LOOP;
END;
$$
LANGUAGE 'plpgsql';

```