



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE CIENCIAS

Sistema de integración y unificación de  
encuestas para la obtención de datos  
estadísticos en la Dirección General de  
Administración Escolar de la UNAM:  
SIUNE-DGAE

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A :

ANGELA VICTORIA INCLÁN JASSO



DIRECTOR DE TESIS:  
MTRA. KARLA RAMÍREZ PULIDO

2012



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

## Hoja de datos del jurado

### 1. Datos del alumno

Apellido paterno Inclán  
Apellido materno Jasso  
Nombre(s) Angela Victoria  
Teléfono 56 37 91 58  
Universidad Universidad Nacional Autónoma de México  
Facultad Facultad de Ciencias  
Carrera Ciencias de la Computación  
Número de Cuenta 302136388

### 2. Datos del tutor

Grado M en A  
Nombre(s) Karla  
Apellido paterno Ramírez  
Apellido materno Pulido

### 3. Datos del sinodal 1

Grado M en C  
Nombre(s) María Guadalupe Elena  
Apellido paterno Ibarguengoitia  
Apellido materno González

### 4. Datos del sinodal 2

Grado Act  
Nombre(s) Carlos Ernesto  
Apellido paterno López  
Apellido materno Nataren

### 5. Datos del sinodal 3

Grado M en I  
Nombre(s) René Alejandro  
Apellido paterno Villeda  
Apellido materno Ruz

### 6. Datos del sinodal 4

Grado L en C C  
Nombre(s) Magali Odalinda  
Apellido paterno Morales  
Apellido materno Rojas

### 7. Datos del trabajo escrito

Título Sistema de integración y unificación de encuestas para la obtención de datos estadísticos en la Dirección General de Administración Escolar de la UNAM: SIUNE-DGAE  
Número de páginas 141 p  
Año 2011

## AGRADECIMIENTOS

*Este trabajo, que tanto tiempo me llevo terminar, es parte de muchas contribuciones de personas que me han ayudado a lo largo de mi estancia en la licenciatura y a quienes debo este logro.*

*A mí papa Miguel, quien ha sido mi inspiración para llegar más lejos en los estudios, es mi ejemplo a seguir y espero que también lo sea para sus hijos Franco y Vivi.*

*A mí abuelita Angela, que se ha encargado de velar por mí mientras mi mama se encargaba de llevar el sustento a la casa. A mis angelitos, mi abuelito Don Nacho y mí papá Pepe, se que siempre me cuidarán desde arriba y a quienes les hubiera gustado verme terminar mis estudios, a quienes siempre tendré presentes.*

*A mis primos Gerardo, Iván y Pames, quienes se que siempre han estado orgullosos de mí, aunque siempre me decian “ya merito pasabas”, espero que sepan que yo también estoy orgullosa de ellos.*

*A mi directora de tesis Karla, que sin su guia, su infinita paciencia, aliento y apoyo no hubiera podido realizar este logro, gracias por ayudarme a terminar la carrera y gracias por las enseñanzas que me brindó a lo largo de la carrera.*

*A Carlitos quien me ayudo infinitamente a solucionar mis dudas, quien me escucho y alento cuando ya no tenia ánimos de terminar, sin su apoyo no hubiera logrado terminar este proyecto.*

*A mi amiga Xoch y a Cris, quienes me acompañaron a lo largo de la carrera, quienes sufrieron conmigo para pasar las materias y con quienes me divertí conociendo nuevos lugares y viviendo nuevas experiencias. A Rafa, Omar y Carlos, ustedes y Xoch son los amigos que permenacieron constantes en este camino, varias personas fueron y vinieron, pero ustedes se quedaron. Gracias por recorrer conmigo este camino.*

*Por último, pero a quien más debo, a mí mama, quien siempre ha tratado de darme todo lo que necesito y aunque no tuvo mucho tiempo para ayudarme con mis tareas, siempre se preocupo por que pasara mis materias, ella es quien me dio los medios necesarios para culminar mis estudios y llegar hasta el lugar donde estoy ahora.*

*Gracias a todos.*

# Índice general

<b>1. Introducción.</b>	<b>1</b>
1.1. Dirección General de Administración Escolar (DGAE) . . . . .	3
<b>2. Estadística.</b>	<b>6</b>
2.1. Áreas de la estadística . . . . .	6
2.1.1. Poblaciones y muestras . . . . .	8
2.1.1.1. Muestreo . . . . .	8
2.1.1.2. Técnicas de muestreo. . . . .	8
2.2. Métodos estadísticos. . . . .	9
2.3. La estadística en la DGPL (UNAM). . . . .	10
2.3.1. Publicaciones de la DGPL . . . . .	11
2.4. Encuestas estadísticas. . . . .	17
<b>3. Bases de Datos.</b>	<b>19</b>
3.1. Definición y características de las bases de datos. . . . .	20
3.2. Sistema Manejador de Bases de Datos. (SMBD) . . . . .	21
3.3. Visión de los datos. . . . .	22
3.4. Arquitectura de los sistemas de bases de datos. . . . .	23
3.4.1. Arquitectura de los sistemas de bases de datos cliente- servidor. . . . .	23
3.4.2. Arquitecturas paralelas de sistemas de bases de datos. . . . .	24
3.4.3. Arquitectura de los sistemas de bases de datos distri- buidos. . . . .	25
3.5. Modelos de bases de datos. . . . .	26
3.5.1. Modelo Entidad Relación (E-R) . . . . .	27
3.5.2. Modelo Relacional. . . . .	29
3.5.3. Modelo orientado a objetos. . . . .	30
3.6. Lenguaje de Consulta Estructurado. (SQL) . . . . .	31

3.6.1. SQL incorporado . . . . .	32
<b>4. Sistemas de Información</b>	<b>33</b>
4.1. Información . . . . .	33
4.2. Definición Sistemas de Información. (S.I.) . . . . .	34
4.2.1. Estructura de un sistema de información. . . . .	35
4.2.2. Ciclo de vida de un sistema de información. . . . .	36
4.2.3. Lenguajes de programación estructurados . . . . .	37
4.2.3.1. Lenguaje C . . . . .	37
4.3. Sistemas de Información Web . . . . .	39
4.3.1. Interfaz de entrada común (CGI) . . . . .	40
4.3.2. Lenguajes de programación orientados a objetos. . . . .	41
4.3.3. Ruby . . . . .	41
4.3.3.1. Ruby on Rails . . . . .	42
4.3.3.2. Bibliotecas de Ruby on Rails . . . . .	44
<b>5. Metodologías de desarrollo de software</b>	<b>47</b>
5.1. Ingeniería de Software. . . . .	47
5.2. Metodologías de desarrollo de software. . . . .	52
5.2.1. Metodologías tradicionales . . . . .	52
5.2.2. Metodologías Ágiles . . . . .	55
5.3. Metodologías ágiles VS metodologías tradicionales. . . . .	60
5.4. Patrones de arquitectura de software. . . . .	61
<b>6. Análisis de Requerimientos</b>	<b>63</b>
6.1. Objetivo. . . . .	63
6.1.1. Requerimientos funcionales. . . . .	63
6.1.2. Requerimientos no funcionales. . . . .	66
6.2. Antecedentes del sistema SIUNE-DGAE. . . . .	66
<b>7. Diseño e implementación del sistema</b>	<b>68</b>
7.1. Diseño del algoritmo de codificación de respuestas. . . . .	69
7.2. Diseño del algoritmo de decodificación de respuestas. . . . .	73
7.3. Diseño de la base de datos SIUNE-DGAE. . . . .	77
7.3.1. Entidades y relaciones. . . . .	77
7.4. Diseño del sistema SIUNE-DGAE desarrollado en C . . . . .	81
7.4.0.1. Arquitectura del sistema. . . . .	83

7.5. Diseño del sistema SIUNE-DGAE desarrollado en Ruby. . . . .	84
7.5.1. Arquitectura del sistema. . . . .	86
7.5.2. Diseño de la Interfaz de usuario. . . . .	88
7.6. Implementación del sistema . . . . .	100
7.6.1. Optimización del Sistema implementado en C. . . . .	101
7.6.2. Desarrollo del sistema implementado en Ruby on Rails. . . . .	106
<b>8. Comparación del sistema desarrollado con C y Ruby. . . . .</b>	<b>110</b>
8.1. Comparativa de diseños de bases de datos del sistema . . . . .	111
8.2. Inicio de sesión. . . . .	116
8.3. Visualización de una encuesta. . . . .	117
8.4. Codificación de las respuestas. . . . .	121
<b>9. Conclusiones. . . . .</b>	<b>123</b>

# Capítulo 1

## Introducción.

La Universidad Nacional Autónoma de México (**UNAM**), considerada como la “Máxima Casa de Estudios” y con una historia nacional de más de 450 años, ha tenido un gran papel en la educación nacional, desde nivel de bachillerato, pasando por la licenciatura y hasta llegar a posgrado; asimismo, un considerable número de docentes, son egresados de esta institución. Además ocupó en el año 2010 la segunda posición en la lista de las mejores universidades de América latina<sup>1</sup>, el lugar 66 en la lista internacional de las mejores universidades<sup>2</sup> y el número 1 en la lista de las mejores universidades de México<sup>3</sup>.

Desde el año de 1910 y hasta nuestro tiempo presente, la Universidad Nacional Autónoma de México (UNAM), se ha consolidado como la institución educativa más importante a nivel nacional. La Ley orgánica[1] por la que se rige nuestra máxima casa de estudios, define las tres funciones sustantivas de la Universidad<sup>4</sup>: docencia, investigación y extensión de la cultura.

---

<sup>1</sup>SCImago Research Group de la Universidad de Granada.

[http://www.scimagoir.com/pdf/ranking\\_iberamericano\\_2010.pdf](http://www.scimagoir.com/pdf/ranking_iberamericano_2010.pdf) (15/06/2011 3:00 pm)

<sup>2</sup>Centro de Información y Documentación del Consejo Superior de Investigaciones Científicas (CSIC) de España.

<http://www.webometrics.info/top12000.asp?offset=50> (15/06/2011 3:00 pm)

<sup>3</sup>Estudio Comparativo de Universidades Mexicanas. ECUM.

<http://www.ecum.unam.mx/> (15/06/2011 5:00 pm)

<sup>4</sup>“Artículo 1.- La Universidad Nacional Autónoma de México es una corporación pública -organismo descentralizado del Estado- dotada de plena capacidad jurídica y que tiene por fines impartir educación superior para formar profesionistas, investigadores, profesores universitarios y técnicos útiles a la sociedad; organizar y realizar investigaciones, principalmente acerca de las condiciones y problemas nacionales, y extender con la mayor amplitud posible los beneficios de la cultura.” *Legislación Universitaria* <http://www.dgebu.unam.mx/m2.htm> (12/05/10 6:00 pm).

Dichas funciones, ya en la práctica, la han convertido en una universidad altamente competitiva colocándola a la par con varias universidades europeas y norteamericanas.

Diversos investigadores de las instituciones educativas de México y del mundo, han determinado el alto impacto que la UNAM ha establecido en la sociedad de nuestro país.

Sobre el particular, investigadores como Gastón García Cantú[2] o bien, Guillermo Sheridan, han dado cuenta de la cantidad de generaciones de profesionistas, intelectuales, artistas plásticos y sobre todo políticos, que luego de haber estudiado y egresado de esta casa de estudios, han contribuido a la transformación económica, cultural y política de México.

	Posgrado	Licenciatura	Bachillerato	Técnicos	Música
UNAM	23,875	172,444	107,848	1,064	738
IPN	5,937	76,225	54,501	***	***
UAM	966	80,084	***	***	***

Cuadro 1.1: Matrículas que solicitan ingresar anualmente de acuerdo al nivel académico.[4, 15, 16]

Del cuadro anterior se observa claramente que la UNAM atiende al mayor número de estudiantes mexicanos del país, tanto en sus opciones de bachillerato, licenciatura y posgrado, en comparación con el Instituto Politécnico Nacional y la Universidad Autónoma Metropolitana.

Actualmente la Universidad atiende, aproximadamente una población escolar de 349,490 alumnos, 108,699 de bachillerato, 179,052 de licenciatura, 25,036 de posgrado y 1,024 en nivel técnico, con una plantilla de 35,679 profesores<sup>5</sup> e investigadores.

La gran cantidad de información que se tiene de todos los aspirantes, alumnos y egresados aumenta considerablemente en tamaño cada ciclo escolar, por tal motivo es indispensable contar con un sistema de cómputo robusto, confiable y sin redundancia en el manejo de los datos y para garantizar la protección de la información que éste maneja debe ser óptimo y seguro, así como también es indispensable contar con una respuesta rápida del sistema por cada petición o acción que realice el usuario.

<sup>5</sup>De los académicos totales 11536 son de tiempo completo.

<http://www.estadistica.unam.mx/numeralia/> (12/09/21010 10:52 pm).

Del amplio universo de facultades, escuelas, institutos o unidades administrativas de la UNAM, la Dirección General de Administración Escolar (**DGAE**), es la dependencia responsable de dirigir, vigilar y coordinar las actividades de administración escolar de la Universidad en su conjunto.

El principal objetivo del presente trabajo es realizar un sistema de información para la Dirección General de Administración Escolar (DGAE) que ayude a manejar la información académica, socioeconómica, personal y de hábitos de estudios de los aspirantes de ingreso a la UNAM, obtenida en cada ciclo escolar, promoviendo la eficiencia de las actividades laborales y académicas en la DGAE, haciendo uso de la ingeniería de software, bases de datos y tecnologías de la información, para así obtener un sistema que cumpla con cada requisito solicitado y que además sea óptimo y mantenible. El Sistema de Encuestas (SIUNE-DGAE) tiene como principal objetivo recaudar los datos que las autoridades correspondientes explotarán para realizar las estadísticas concernientes a cada encuesta de aspirantes, alumnos y egresados de la UNAM.

## 1.1. Dirección General de Administración Escolar (DGAE)

Desde su creación, sus principales objetivos quedaron perfectamente definidos: mejorar la calidad de los servicios, dar validez y disminuir el tiempo en los procesos de la emisión de certificados, títulos y grados, recibir la documentación para concursos de selección de ingreso a la UNAM, responder a las necesidades de información en todos los niveles de la administración escolar y de la sociedad en general.

Su visión es fomentar, consolidar y cumplir siempre con una Administración Escolar Institucional que sea reconocida por toda la comunidad universitaria, como lo son: alumnos, académicos y trabajadores, así como la sociedad mexicana en su conjunto.

La DGAE está organizada en cinco Subdirecciones y dos Unidades (Unidad de Administración del Posgrado y Unidad Administrativa).

Las Subdirecciones son:

- **Subdirección de Coordinación, Desarrollo y Análisis de la Información (SCDAI)**, en la cual se realizan los procesos correspondientes con la información obtenida de los alumnos y aspirantes de ingreso a la UNAM.

- **Subdirección de Registro y Aplicación del Examen de Selección (SRAES)**, dentro de sus funciones se encuentra el atender a los aspirantes a los concursos de selección de nivel bachillerato y licenciatura, aplicar exámenes en las diferentes sedes y recibir la documentación correspondiente de los nuevos alumnos que no tienen antecedentes en la UNAM.
- **Subdirección de Certificación y Control Documental (SCyCD)**, la cual se encarga de vigilar que se cumpla la normatividad, la certificación de los estudios universitarios y la emisión de títulos y grados.
- **Subdirección de Sistemas de Registro Escolar (SSRE)**, la cual es la responsable de del registro y planes de estudio, registro de actividades en el ciclo de planeación, registro y autorización de la reinscripción , entre otros servicios.
- **Subdirección de Diseño de Proyectos (SDP)**, la cual es la responsable de los procesos relacionados con el primer ingreso a la UNAM, también se encarga de la administración de los recursos de cómputo y del desarrollo de nuevos sistemas, así como de la optimización de sistemas ya existentes para la administración escolar.

Las Unidades con las que cuenta son:

**Administración del Posgrado**, que es la unidad responsable del registro y seguimiento escolar de los programas de posgrado y especialización.

**Unidad Administrativa**, cuyas funciones son de apoyo y administración financiera (bienes/servicios), así como el realizar los procesos relacionados con los recursos humanos.

Para optimizar los recursos disponibles, la DGAE cuenta con diversos sistemas de cómputo como son:

- **Sistema para los Planes de Estudio** el cual administra la información de los planes, la lista de licenciaturas, objetivos, requisitos de ingreso y asignaturas de cada plan de estudio en la UNAM.
- **Sistema de Primer Ingreso** realizado por la Subdirección de Diseño de Proyectos (SDP), es el encargado del proceso de admisión a nivel bachillerato y licenciatura, así como del concurso de selección, pase reglamentado, asignación de plantel, turno y grupo a aspirantes aceptados en la UNAM, así como el validar la documentación necesaria para la inscripción a estos.

- **Sistema para el Registro y Seguimiento Escolar SIAE** (Sistema Integral de Administración Escolar), realizado por la Subdirección de Sistemas de Registro Escolar (SSRE), el cual se encarga de llevar el seguimiento de la historia académica de cada alumno.
- **Sistema de Egreso**, llevado a cabo por la Subdirección de Control y Certificación Documental (SCyCD), se encarga de la certificación, revisión documental y emisión de títulos y grados.
- **Sistema de Información**, es realizado por la Subdirección de Diseño de Proyectos (SDP) el cual está comprendido por varios sistemas para poder tener un fácil acceso a la información de toda la comunidad universitaria, uno de estos sistemas es el sistema de *Hojas de Datos Estadísticos* ó *Sistema de Encuestas*, que es el encargado de recabar la información de los aspirantes y alumnos de la UNAM.

Una vez expuestos los antecedentes referidos a la conformación de la Dirección General de Administración Escolar (DGAE), es oportuno precisar que el presente trabajo, tiene la finalidad de detallar la forma en que fue constituido el **Sistema de Encuestas**, los procesos de cambio que han ocurrido desde su creación a la fecha, así como las herramientas de trabajo con las que actualmente opera, a fin de exponer la utilidad real que ha ofrecido a las distintas áreas de la UNAM, y en última instancia, proponer modificaciones para su mejora y optimización.

## Capítulo 2

# Estadística.

Los seres humanos se han caracterizado por tener cierta curiosidad de conocer y descubrir nuevas cosas, que ayuden a hacer más cómoda su vida diaria; para lograr esto, los humanos han registrado sus vivencias en distintos materiales. Con el paso del tiempo, estos datos registrados se convierten en una cantidad enorme de información y como consecuencia, se han requerido herramientas y técnicas que faciliten la manipulación de los datos recopilados, para mejorar la toma de decisiones en el menor tiempo posible, tanto ellos mismos como por parte de aquellas personas que necesiten realizar una acción con base en los datos recopilados.

Las herramientas y técnicas creadas para la manipulación de los datos han sido de gran ayuda en la vida diaria de los seres humanos, ya que si no se tienen las técnicas para realizar acciones ó tomar decisiones de acuerdo a los datos, éstos no serían de gran ayuda para los seres humanos. De esta manera la estadística ha sido una parte fundamental para interpretar los datos recopilados.

*“La estadística es la ciencia que trata de la recolección, clasificación y presentación de los hechos sujetos a una apreciación numérica como base a la explicación, descripción y comparación de los fenómenos.”*[6]

Otros autores la definen así:

*“La estadística es el arte de aprender a partir de los datos. Está relacionada con la recolección de datos, su descripción subsiguiente y su análisis.”*[7]

### 2.1. Áreas de la estadística

La estadística se divide en dos áreas:

- **Estadística inferencial:** se dedica a inferir el resultado del comportamiento o características de las muestras, para lograr esto la estadística se basa en el análisis obtenido de los datos. Es utilizada comúnmente para crear patrones en los datos. La principal ventaja de ésta es que proporciona los modelos que probablemente seguirán los datos [7].
- **Estadística descriptiva:** desarrolla un conjunto de técnicas para la recolección de los datos, su presentación y resumen, con base en los datos observados. También estudia la dependencia que puede existir entre dos o más características observadas, para poder ordenarlos y presentarlos de manera legible [8] .

El siguiente ejemplo ilustra mejor la diferencia entre estas dos áreas:

La dependencia encargada de la educación de un determinado país, no puede realizar la construcción de escuelas si no dispone de los conocimientos previos necesarios, tales como: número de alumnos que necesitan ser escolarizados, edades de éstos, distribución geográfica, grado de conocimientos y situación económica de los alumnos. Estos datos necesarios para resolver el problema son proporcionados como resultado de las estadísticas del estado (censo de población), o bien por las estadísticas realizadas por la misma dependencia de educación, dichas estadísticas son elaboradas con las técnicas proporcionadas por la *estadística descriptiva*, ya que es la parte de la estadística que se encarga de la recolección de los datos.

Los datos obtenidos en las estadísticas mencionadas anteriormente, son el resultado de aplicarlas en un cierto momento, sin embargo es importante considerar que las escuelas estarán construidas en un futuro próximo, por lo que estos datos podrían variar. Las posibilidades de cambio deben de tenerse en cuenta, así como la probabilidad de éxito; este tipo de estudio depende de la *estadística inferencial* ya que ésta brindará las probabilidades de éxito ó fracaso que deben tomarse en cuenta para la construcción de una escuela.[8]

Ahora bien, la estadística para ser funcional no sólo se concentra en ser mera “recolección de datos”, lo que la hace particularmente valiosa es su descripción subsiguiente y su análisis. Para diversos autores, un factor importante acerca del análisis de la información, debe centrarse en, por y para la utilidad de los seres humanos. No hay que olvidar que la principal finalidad de la estadística es facilitar la mejor toma de decisiones, en el menor tiempo posible.

### 2.1.1. Poblaciones y muestras

La población (en algunos contextos es llamado universo) se define como el conjunto total de elementos a los cuales se refiere el análisis estadístico, y las muestras son un subconjunto de la población [8].

Al total de conjuntos de muestras que se pueden obtener de la población se denomina espacio muestral.

La razón por la que se realizan las muestras es porque el muestreo es más rápido que hacer el conteo completo, también es menos costoso realizar el estudio en la muestra que en la población completa [8].

Se procura que las muestras sean representativas de todos los elementos de la población (que se obtengan todas las características de la población en la misma proporción), ya que se pretende generalizar los resultados que se obtienen al aplicar alguna prueba a las muestras, de la misma manera se podrían aplicar éstos a diferentes poblaciones con las mismas características, pero en situaciones diferentes [9].

#### 2.1.1.1. Muestreo

Es la técnica utilizada para la selección de la muestra a partir de una población. Para verificar que la muestra sea representativa de la población se realiza el muestreo.

#### 2.1.1.2. Técnicas de muestreo.

Los métodos de selección de muestras son [9]:

- Muestreo aleatorio: consiste en seleccionar al azar los elementos de la población que conformarán la muestra. Este método es el más recomendable, sin embargo no se puede asegurar que todas las características deseadas se hayan obtenido al utilizar este método.
- Muestreo aleatorio sistemático: este tipo de muestreo consiste en realizar una lista con los elementos de la población, posteriormente en algún punto al azar cerca del principio de la lista se seleccionan intervalos fijos de elementos, esto se repite a lo largo de toda la lista.
- Muestreo estratificado: esta técnica de muestreo sirve cuando se desea conocer alguna característica específica de la población. Las características de la población que no se encuentren en la misma distribución y tengan alguna relación con lo que se desea analizar, deberán ser distribuidas uniformemente, si estas características no tienen alguna relación con lo que se desea analizar no importará que estén estratificadas.

- Muestreo no aleatorio: este método se basa en la experiencia de alguna persona con los elementos de la población, en general no se puede calcular la probabilidad de extracción de una determinada muestra. En este método la información proporcionada por la persona con experiencia sirve para la toma de decisiones.
- Muestreo no probabilístico: en este muestreo para ningún miembro de la población se conoce la probabilidad de ser seleccionado. Algunas de las estrategias de muestreo no probabilístico son:
  - Muestreo de conveniencia: esta estrategia selecciona la muestra de la población que le conviene realizar el experimento, por ejemplo un maestro puede seleccionar una muestra de la población que serían sus alumnos si a éstos al realizar el experimento se les brinda algo a cambio, como podrían ser créditos extra.
  - Muestreo por cuotas: se recomienda cuando se desea realizar un muestreo estratificado pero no es posible realizarlo de tal manera, en esta estrategia se seleccionan personas con las características deseadas, hasta llegar a la cuota (o cantidad esperada) de personas, se debe tomar cuenta que éstas no son seleccionadas aleatoriamente.

## 2.2. Métodos estadísticos.

La estadística, para obtener los datos necesarios cuenta con dos tipos de estudios estadísticos (también llamados métodos estadísticos), los cuales son [10]:

- Estudios experimentales: en este método se deben tomar mediciones del sistema, posteriormente modificarlo y volver a tomar mediciones, la toma de mediciones en ambos casos debe hacerse por medio del mismo procedimiento, los pasos a seguir de este procedimiento son:
  - Planeación de la investigación: seleccionar el material de consulta y plantear el problema que se estudiará.
  - Diseño del experimento: Se realiza el muestreo para la obtención de datos, se analizan los datos y se proporcionan los resultados utilizando la estadística descriptiva.
  - Usando un modelo estadístico se toman decisiones o se predicen acontecimientos futuros.

- Estudios observacionales: en este método los datos son obtenidos y posteriormente analizados. La observación puede ser periódica (que se lleva a cabo en períodos constantes, pueden ser semanas, meses, días, entre otros), continua (que se realiza de manera permanente) o circunstancial (se realiza de manera esporádica, atendiendo a una necesidad momentánea). De acuerdo a su cobertura los estudios observacionales pueden realizarse de la siguiente manera [8]:
  - Exhaustiva: esta observación se realiza sobre todos los elementos de la población.
  - Parcial: esta observación se realiza cuando es imposible observar a toda la población, entonces solo se observa una parte de ella.
  - Mixta: en esta observación se combinan la observación exhaustiva, que se realiza sobre algunas características de toda la población, y la observación parcial, se realiza sobre una muestra de la población.

Algunas técnicas de muestreo que se usan en los estudios observacionales son [8]:

- Censo: son observaciones exhaustivas, se realizan periódicamente en un plazo de 5 ó 10 años y por medio de éste se estudian las características más estáticas de la población.
- Estadísticas: son investigaciones previamente realizadas que muestran resultados sobre las características más dinámicas de la población, esta técnica se realiza con más frecuencia que el censo.
- Encuestas: regularmente son observaciones parciales y se realizan de manera ocasional.

### 2.3. La estadística en la DGPL (UNAM).

En nuestro país, y concretamente en la UNAM, la generación de estadísticas y su utilización, ocupan un lugar preponderante, no únicamente para el beneficio propio de nuestra casa de estudios, sino incluso para el entendimiento de diversos factores de la vida nacional, por lo que la Dirección General de Planeación (DGPL) utiliza la estadística como herramienta para hacer cumplir las funciones que le corresponden y que la normatividad de la UNAM establece. Por ello es pertinente revisarla.

“La normatividad vigente confiere a la Dirección General de Planeación (DGPL) la responsabilidad de construir los fundamentos y los instrumentos metodológicos necesarios para contribuir a que la planeación sea la forma habitual de trabajo en la UNAM, en su guía para las labores cotidianas, en el punto de referencia para la evaluación de su quehacer.”<sup>1</sup> En otras palabras, la DGPL vigila, establece y controla que las políticas, los fundamentos y las metodologías que empleará la UNAM para realizar sus actividades en todos sus ámbitos y niveles se logren según lo programado en los tiempos establecidos.

Las encuestas que conforman sistema SIUNE-DGAE son diseñadas por la Comisión de Encuestas del Grupo Técnico de Responsables de Estadística y Planeación Institucional, (Dirección General de Planeación DGPL), y a su vez la DGAE es la encargada de crear las aplicaciones necesarias para el funcionamiento de las mismas.

“La DGPL sirve a la Universidad mediante la elaboración y publicación de la Agenda Estadística, la Memoria de la UNAM, la serie de Cuadernos de Planeación, los perfiles de ingreso a la UNAM, la construcción de indicadores de desempeño y el levantamiento de encuestas diversas en el seno de la comunidad.”<sup>2</sup> La DGPL también tiene a su disposición el Portal de Estadística Universitaria para describir las actividades de la Universidad, apoyar actividades de planeación y evaluación, además de brindar al público en general información cuantitativa acerca de la comunidad universitaria.

### 2.3.1. Publicaciones de la DGPL

Dentro de las publicaciones de la DGPL se pueden encontrar:

- Agenda Estadística.
- Memoria de la Universidad Nacional Autónoma de México.
- Perfiles: que incluye los datos de los niveles bachillerato, técnico y licenciatura.
- Perfiles de alumnos egresados de nivel licenciatura.

---

<sup>1</sup>Dirección General de Planeación. <http://www.planeacion.unam.mx/> (12/08/2010 3:00 pm.)

<sup>2</sup>Dirección General de Planeación. <http://www.planeacion.unam.mx/> (12/08/2010 4:00 pm.)

- Cuadernos de Información Estadística Básica: muchos de los cuales contienen información obtenida con el apoyo<sup>3</sup> de la **DGAE**.

La DGPL publica anualmente el “*Perfil de aspirantes, así como los alumnos asignados a bachillerato y licenciatura en la UNAM*”, con el objetivo de que cada unidad académica obtenga un mayor conocimiento de su población de primer ingreso. A partir del año 2005 todos los aspirantes de concurso de selección de licenciatura, concurso de selección de bachillerato y pase reglamentado, que llenaron la encuesta (Hoja de Datos Estadísticos) lo hicieron de forma electrónica.

Esta información se encuentra segregada por el concepto de publicación por población aspirante<sup>4</sup> y el de población asignada<sup>5</sup>.

*“El perfil del aspirantes y los alumnos asignados a bachillerato y licenciatura de la UNAM (PAA-UNAM) es producto del análisis de la información recabada a través de la hoja de datos estadísticos, aplicada a los aspirantes y alumnos asignados desde el año 1988, su propósito es explorar los antecedentes socioeconómicos y académicos más relevantes de la población que desea ingresar a la institución (aspirantes) y la que de hecho ingresa<sup>6</sup>, y ser la base para estudios de demografía escolar tanto a nivel general de la Universidad como a niveles específicos por escuela o facultad, carrera, sexo, grupo de edad, etcétera.” [14]*

Los diferentes módulos del sistema (“Hoja de Datos estadísticos ó Encuestas”) que se encuentran disponibles en la página de la Dirección General de Administración Escolar ([www.dgae.unam.mx](http://www.dgae.unam.mx)) son las siguientes:

- Concurso de Selección de ingreso a nivel Bachillerato (CSB) aplicada a los aspirantes asignados a nivel bachillerato.
- Pase Reglamentado (PR) aplicada a la población de bachillerato que desea ingresar a licenciatura por pase reglamentado.
- Concurso de Selección de ingreso a nivel Licenciatura (CSL) aplicada a los aspirantes de ingreso a licenciatura de la UNAM por medio del concurso de selección.

---

<sup>3</sup>La DGAE proporciona los datos de la población estudiantil y académica que la DGPL necesita para realizar diversas estadísticas.

<sup>4</sup>Son todas aquellas personas registradas en algún concurso de selección a bachillerato o licenciatura a la UNAM o bien que tienen el pase reglamentado como ingreso a la UNAM.

<sup>5</sup>Subconjunto de aspirantes que fueron elegidos en el concurso de selección o que por medio del pase reglamentado, tienen un lugar asignado en la UNAM.

<sup>6</sup>La población que si fue asignada a la UNAM.

- Estudios de Equidad y Género (PUEG) aplicada a todos los usuarios del sistema, incluye los aspirantes de ingreso a nivel licenciatura y bachillerato, así como a los alumnos que tienen como opción el pase reglamentado.

Cada año se realizan cambios mínimos a las encuestas, como por ejemplo el agregar preguntas referentes a los pueblos y lenguas indígenas de México con el fin de conocer los antecedentes culturales; o bien el adecuar la cantidad del salario mínimo correspondiente al año actual, con el fin de obtener información acerca del sostén económico de los aspirantes y alumnos de la UNAM.

En la publicación del perfil de aspirantes y alumnos de la UNAM se pueden encontrar datos generales, tales como:

- Edad.
- Sexo:
  - Femenino.
  - Masculino.
- Estado civil:
  - Soltero.
  - Casado.
  - Divorciado.
  - Viudo.
- Tipo de primaria cursada:
  - Pública.
  - Privada.
  - Ambas.
- Tipo de secundaria cursada:
  - Pública.
  - Privada.
  - Ambas.

- Número de materias recursadas:
  - Ninguna.
  - 1 ó 2.
  - 3 ó más.
  
- Escuela de Procedencia:
  - Incorporada a la SEP.
  - Incorporada a la UNAM.
  - Colegio de Bachilleres.
  - Vocacional.
  - Nivel Superior.
  
- Promedio de calificaciones:
  - De 8.1 a 8.5.
  - De 8.6 a 9.0.
  - De 9.1 a 9.5.
  - De 9.6 a 10.
  
- Año de egreso de la escuela de procedencia:
  - Antes de 1990.
  - 1991.
  - Hasta el año actual, en este caso, 2011.
  
- Número de hermanos:
  - 1.
  - 3.
  - 5.
  - 7.
  - 9 ó más.
  
- Costumbres de estudio:

- Primero leer todo el tema.
- Subrayar las ideas principales.
- Elaborar resúmenes del material.
- Hacer síntesis tipo "acordeón".

■ Material de consulta:

- Revistas.
- Libros.
- Internet.
- Enciclopedias.
- Atlas y mapas.

Situación socioeconómica:

■ Si se habla alguna lengua indígena:

- Si.
- No.

■ Nivel máximo de estudio de los padres y hermanos:

- Secundaria.
- Escuela superior.
- Carrera técnica.
- Vocacional.
- Licenciatura.
- Posgrado.

■ Principal sostén económico:

- Padre.
- Madre.
- Ambos padres.
- Cónyuge o pareja.
- Otra persona.

- Bienes y servicios:
  - Lavadora.
  - Secadora.
  - Computadora.
  - Automóvil.
  - Teléfono celular.
  - Dvd.
  - Aspiradora.
  - Personal de servicio.
  
- Ingreso familiar mensual:
  - Menos de 2 salarios (Menos de \$3,448).
  - De 2 a 4 (De \$3448 á \$6,895).
  - De 4 a 6 (De \$6,895 a \$10,343).
  - De 6 a 8 (De \$10,343 a \$13,790).
  - De 8 a 10(de \$13,790 a \$17,238).
  - Mas de 10 (Mas de \$17,238).

*“La hoja de datos estadísticos no se ha modificado sustancialmente desde 1995, con el propósito de mantener la comparatividad de la información y facilitar la obtención de conclusiones, así como la elaboración de estudios que apoyen la planeación, evaluación y toma de decisiones a partir de información cuantificable.” [14]*

También se planea obtener los perfiles de los egresados de licenciatura de la misma manera, (utilizando la hoja de datos estadísticos), así como también obtener el perfil de los alumnos que solicitan becas (hoja de estadísticos de becas)<sup>7</sup> .

De esta forma cada módulo del sistema sirve para obtener datos relevantes para la planeación en distintos niveles y aspectos de la educación.

---

<sup>7</sup> *El procesamiento de la información obtenida por el sistema de encuestas se llevó a cabo en los paquetes estadísticos SPSS y Visual Fox Pro para Windows. La edición y presentación final del reporte se realizó en Microsoft-Word, Microsoft-Excel y Adobe Indesign CS3. Publicaciones <http://www.planeacion.unam.mx/Publicaciones/> (03/05/2010 1:05 am).*

Por ejemplo con el módulo de **Género y Equidad**, se puede conocer que porcentaje de la población estudiantil de la UNAM tiene hijos y cuál es la edad de dichos hijos; por otro lado con los módulos correspondientes a concursos de selección y pase reglamentado se puede obtener el salario mensual promedio.

## 2.4. Encuestas estadísticas.

La encuesta es el acopio de datos obtenidos mediante consulta o interrogatorio, se refieren a opiniones, nivel económico, nivel académico, o algún otro aspecto de la vida diaria.

La información recabada por las encuestas se obtiene de forma estructurada, mostrando a las personas que contestarán las encuestas siempre en el mismo orden de las preguntas, conservando así la estructura de la encuesta.

*“Las encuestas, como toda investigación, procuran alcanzar la objetividad y se preocupa por la exactitud de los resultados; cada indagación individual es provisoria, mas la totalidad de las mismas resulta acumulativa”*<sup>8</sup>.

Las principales etapas en el desarrollo de una encuesta son:

- **Formulación del problema:** esta etapa implica definir previamente el problema que se desea resolver con los datos que son recopilados con la encuesta.
- **Definición de objetivos:** en esta etapa para obtener un mejor diseño de la encuesta se definen los objetivos que está deberá cumplir, así como son los datos que brindará, los resultados que se obtendrán de la encuesta, el alcance de la encuesta en la población, entre otras cosas.
- **Delimitar el tamaño de la muestra:** el tamaño de la población que será analizada por medio de la encuesta puede variar dependiendo del tiempo y recursos que se dispongan para aplicar la encuesta. Pueden llegar a presentarse algunos errores relacionados con el tamaño de la muestra; cuanto mayor sea la muestra, más precisos serán los resultados, si la muestra es muy pequeña en ocasiones no será posible generalizar los resultados.
- **Diseño de la encuesta:** se recomienda que la encuesta esté conformada por bloques y las preguntas se encuentren ordenadas, evitar la ambigüedad en las preguntas, para esto deben ser cortas y bien definidas.

---

<sup>8</sup>Hyman Herbert, **Diseño y análisis de las encuestas sociales**, Amorrortu editores, 2001, pág 52.

También es recomendable motivar al usuario a contestar la encuesta, esto puede realizarse por medio de una agradecimiento por contestar la encuesta.

- Aplicación de la encuesta: se debe tener en cuenta el espacio donde será aplicada la encuesta, este espacio debe ser neutral para el encuestado, no debe influir en sus respuestas. La aplicación de la encuesta sirve para recopilar los datos.
- Análisis de datos: en esta etapa se analizan los datos que son obtenidos con la encuesta. No existe un procedimiento único para realizar el análisis de los datos, esto depende de cada problema que se desea resolver. Después del análisis de los datos se realiza un informe con los resultados de tal análisis.

## Capítulo 3

# Bases de Datos.

En la UNAM se almacena una gran cantidad de datos de alumnos, académicos, aspirantes a alumnos y trabajadores. Cada año, con cada convocatoria de ingreso a la UNAM se registran más de 300 000 aspirantes, y cada uno de éstos brinda información valiosa a nuestra Máxima Casa de Estudios, como lo son los datos personales, opciones de ingreso, antecedentes escolares, entre otros. Tener toda esta información de manera ordenada y de fácil acceso es una necesidad importante. Cada movimiento que se realiza de cada estudiante o de cada académico, es almacenado tanto para realizar futuras aclaraciones como para mejorar los diferentes servicios ofrecidos a través de sistemas de información, estos últimos encargados de mostrar, actualizar o borrar los datos de toda la comunidad universitaria.

Desde sus inicios la UNAM ha sido una universidad con una gran demanda de estudiantes ávidos de ingresar y ser parte de ella, desde entonces la matrícula de la Universidad se ha incrementado año con año, con el fin de dar cabida a la mayor cantidad de solicitudes de ingreso.

<b>Aspirantes</b>	<b>(2008)</b>	<b>(2009)</b>	<b>(2010)</b>
UNAM	348,283	374,175	381,745
IPN	71,952	60,232	76,225
UAM	65,195	67,988	76,991

Cuadro 3.1: Número de aspirantes a ingresar en las principales universidades públicas en los últimos tres años.

En el cuadro anterior se puede observar que la UNAM atiende un número

ro mayor de aspirantes en comparación con el Instituto Politécnico Nacional (IPN) y la Universidad Autónoma Metropolitana (UAM). En cada ciclo escolar la UNAM atiende aproximadamente a 370,000 aspirantes, 200,000 alumnos de reingreso a nivel bachillerato y licenciatura, así como 20,000 alumnos de posgrado.

La UNAM preocupada por la siempre creciente matrícula ha tenido que ir incrementando todos sus servicios e infraestructura, brindando siempre una excelente educación a cada uno de sus estudiantes, es por ello que para sustentar la principal tarea de la Universidad es necesario que los servicios tanto académicos como administrativos sean siempre realizados de manera eficiente, en particular, se ha requerido conocer el perfil de los estudiantes de primer ingreso desde varios ámbitos como lo son: su situación económica, su situación familiar, entre otras, para así poder recibirlos y atenderlos de acuerdo a sus necesidades previamente establecidas por los datos expuestos anteriormente.

Para almacenar toda esta información, procesarla, y finalmente analizarla; se han utilizado las bases de datos, las cuales forman una parte fundamental de los sistemas de información, ya que es posible aumentar el rendimiento de los sistemas, debido a que se facilita la manipulación de los datos, obteniendo así con una simple consulta los datos deseados, sin tener que buscar en miles de archivos, como antes se solía hacer, también se tiene una independencia de las bases de los datos con los sistemas de información y una menor redundancia de datos, así como una mayor seguridad en éstos.

### 3.1. Definición y características de las bases de datos.

Algunos autores la definen como: “Una base de datos es una colección de datos interrelacionados”,<sup>1</sup> otros autores la definen como: “un conjunto grande de datos estructurados almacenados dentro de una computadora”<sup>2</sup>.

Algunas de las características más importantes de las bases de datos son:

- Independencia lógica y física de los datos: la independencia lógica permite cambiar la estructura lógica de la base de datos, sin alterar la estructura de los sistemas que acceden a la base de datos. La independencia física permite que la distribución física de los datos sea inde-

---

<sup>1</sup>Silberschatz Abraham, F. Korth Henry, Sudarshan S., **Fundamentos de Bases de Datos**, 2002, España, pág 25.

<sup>2</sup>Carlo Batini, **Diseño conceptual de bases de datos: un enfoque de entidades relacionales**, Ediciones Díaz de Santos, 1994, pág 4

pendiente de la estructura lógica de la base de datos y de los sistemas que los manipulan.

- **Minima redundancia:** cuando se tiene redundancia de datos, es muy probable que existan problemas de inconsistencia, ya que muchas veces al actualizar algún dato, no se realiza la actualización en todas las repeticiones de los datos.
- **Capacidad de respuesta:** las bases de datos deben responder a los procesos o sistemas que accedan a ésta de manera eficiente, en un tiempo de respuesta razonable.
- **Integridad de los datos:** cuando se realiza alguna modificación en los datos, la integridad de los datos se preserva, es decir, no se modificarán con un valor incorrecto, ni se perderán.
- **Seguridad de acceso:** se refiere a evitar el acceso o modificación de datos a usuarios que no tengan permiso de realizar la función solicitada.
- **Atomicidad de los datos:** cuando ocurre alguna falla en la computadora que está modificando los datos de la base de datos debe asegurarse que los datos se restauren al estado de consistencia que existía antes del fallo. Por tal razón las operaciones realizadas en la base de datos son atómicas.
- **Acceso de multiples usuarios:** aprovechando el tiempo de respuesta de las bases de datos se ha logrado permitir accesos de mas de un usuario al mismo tiempo a la base de datos sin presentar errores de consistencia.

### 3.2. Sistema Manejador de Bases de Datos. (SMBD)

El sistema manejador de bases de datos es el software que permite realizar la comunicación entre la base de datos física y los usuarios del sistema. Éste realiza operaciones tales como agregar o eliminar tablas o registros, recuperar o almacenar datos desde archivos, solicitar un dato específico de la base, entre otras.

Los principales lenguajes que contiene el sistema manejador de bases de datos son:

1. Descripción de los datos: para lograr esto utiliza el lenguaje de definición de datos (por sus siglas en inglés, Data Definition Language DDL),

este lenguaje permite definir los elementos que conforman la base de datos, la estructura de cada uno y las relaciones que hay entre éstos.

2. Manipulación de los datos: llamado lenguaje de manipulación de datos (por sus siglas en inglés, Data Manipulation Language DML), permite a los usuarios añadir, modificar, eliminar y seleccionar los registros de la base de datos que cumplen con ciertas características requeridas.
3. Control de datos: esta funcionalidad permite que los diferentes usuarios se comuniquen con la base de datos, permite controlar los accesos no autorizados, además de las capacidades de los archivos, esto se logra utilizando el lenguaje de control de datos (por sus siglas en inglés, Data Control Language DCL).

### 3.3. Visión de los datos.

La arquitectura de los sistemas de bases de datos más utilizada (ANSI/SPARC<sup>3</sup>) se compone de tres niveles que son [25]:

- Nivel físico: también es conocido como nivel interno, el cual indica cómo son almacenados físicamente los datos. Este nivel corresponde a una visión más real del almacenamiento de éstos, por ejemplo los datos se pueden ver como registros.
- Nivel lógico o conceptual: en este nivel describe donde se almacenan los datos y las relaciones que existe entre ellos. En este nivel todos los datos de la base de datos se describen en términos de estructuras simples.
- Nivel de vistas: en este nivel muestra solo una parte de los datos de toda la base de datos completa, corresponde a la forma en que los usuarios ven la información individualmente. En este nivel cada usuario puede observar diferentes datos, por ejemplo un estudiante puede ver sus calificaciones, mientras que un maestro puede observar las calificaciones de todos sus alumnos en una asignatura.

---

<sup>3</sup>Es el diseño estandar propuesto en 1975 por el Instituto Nacional Americano de Estándares/Comité de Normas de Planificación y Requerimientos, por sus siglas en inglés, American National Standards Institute, Standards Planning And Requirements Committee. [http://en.wikipedia.org/wiki/ANSI-SPARC\\_Architecture](http://en.wikipedia.org/wiki/ANSI-SPARC_Architecture) (01/07/11 02:41 pm.)

### 3.4. Arquitectura de los sistemas de bases de datos.

La arquitectura del sistema de una base de datos depende en gran medida del sistema que tiene acceso a la base de datos, ya que éste es el que solicita, modifica o borra los datos de la base. Algunas de las arquitecturas más utilizadas se explican a detalle a continuación:

- Arquitectura cliente-servidor.
- Arquitectura paralela.
- Arquitectura distribuida.

#### 3.4.1. Arquitectura de los sistemas de bases de datos cliente-servidor.

Las bases de datos pueden ser vistas como un sistema con arquitectura cliente-servidor [25]. El servidor puede ser el sistema que tiene acceso a la base de datos, el cual cumple con las funciones de definición, manipulación, seguridad e integridad de los datos, entre otros. Los clientes son los usuarios que observan la parte visible de la base de datos (formularios, informes, interfaces de usuario gráficas o sistemas que muestran los datos de la base).

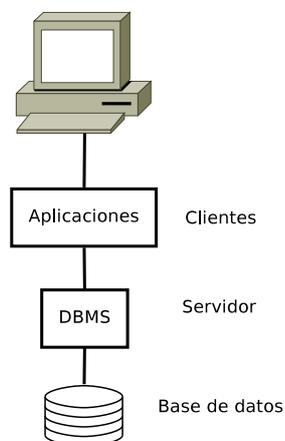


Figura 3.1: Arquitectura Cliente-Servidor de bases de datos

En la figura 3.1 se pueden observar los componentes de esta arquitectura de bases de datos. Un ejemplo de esta arquitectura es el sistema manejador

de bases de datos que funciona como el servidor, y donde los clientes son las aplicaciones que realizan algún acceso u operación sobre los datos de ésta.

### 3.4.2. Arquitecturas paralelas de sistemas de bases de datos.

Los grandes avances en la tecnología, han permitido que se procese la información en paralelo en una computadora, logrando así que las solicitudes de transacciones de datos se realicen en menor tiempo.

Existen diferentes arquitecturas de bases de datos paralelas, las cuales difieren del número de procesadores ó del número de memorias que se tengan, estos modelos son:

- Memoria compartida: en esta arquitectura solo se tiene una memoria y todos los procesadores tienen acceso a ésta. Esta arquitecturas presentan gran eficiencia respecto a la comunicación entre procesadores. Una desventaja es que a partir de cierto número de procesadores la comunicación entre éstos y la memoria puede convertirse en un cuello de botella.
- Disco compartido: esta arquitectura cuenta con un grupo de discos que están disponibles para los todos los procesadores utilizando una red de interconexión, además cada procesador tiene su propia memoria a la cual no puede acceder otro procesador. También tiene tolerancia a los fallos, ya que si algún procesador tiene un error los procesadores restantes se hacen cargo de las funciones que desempeñaba el procesador que sufrió el daño sin ningún problema, ya que la base de datos se encuentra en los discos. Esta arquitectura soporta más procesadores interconectados que el modelo de memoria compartida, pero la comunicación entre éstos es más lenta.
- Sin compartimiento de disco, ni memoria: cada procesador en esta arquitectura tiene su propia memoria así como sus propios discos y cada procesador es visto como un nodo. Los nodos pueden tener comunicación entre sí y cada nodo es como un servidor de datos almacenados en sus propios discos, así cada nodo responde a las peticiones de datos con referencias locales y solo se transmiten por la red las peticiones a los accesos de discos remotos. El principal inconveniente de esta arquitectura es que depende del software para las comunicaciones entre los nodos para el envío y la recepción de los datos.
- Jerárquico: esta arquitectura combina las arquitecturas anteriores, el sistema está formado por nodos que no comparten memoria, ni dis-

cos, pero cada nodo puede ser un sistema de arquitectura de memoria compartida, de disco compartido ó sin compartimiento.

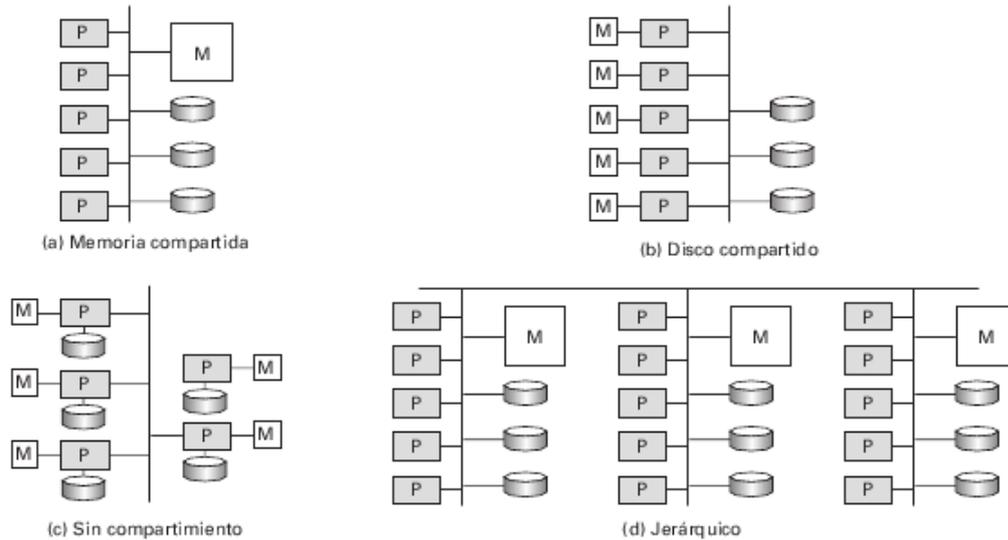


Figura 3.2: Arquitectura paralela de bases de datos [23].

En la figura 3.2 se puede observar las diferentes arquitecturas paralelas de bases de datos anteriormente explicadas<sup>4</sup>.

### 3.4.3. Arquitectura de los sistemas de bases de datos distribuidos.

En esta arquitectura la base de datos es almacenada en varias computadoras, las cuales se comunican entre ellas por medio de redes de área local o de área amplia, sin embargo no comparten memoria, ni discos. Las computadoras que contienen a la base de datos pueden estar en diferentes lugares geográficos, se pueden administrar de forma distinta, pero la comunicación entre ellas es más lenta que en las otras arquitecturas. Una de las ventajas de esta arquitectura es que los usuarios pueden acceder a los datos de una computadora remota desde otra computadora más cercana a la ubicación del usuario, dicho de otra forma, desde la computadora en que se encuentra en usuario se puede acceder a los datos de cualquier otra computadora remota

<sup>4</sup>Donde “P” significa procesador y “M” memoria.

sin tener que trasladarse hasta ellas. Si alguna computadora falla las demás computadoras pueden seguir funcionando sin problemas y en el caso de que los datos de la computadora que tuvo el error estén replicados en el resto de las computadoras, las operaciones que requieran esos datos pueden ser respondidas.

### 3.5. Modelos de bases de datos.

Para representar las bases de datos en el mundo real, se utilizan modelos, y especificaciones que indican cómo debe estar estructurada una base de datos. Un modelo de datos es una definición lógica, abstracta e independiente de los objetos.

Los modelos conceptuales de las bases de datos deben cumplir con algunas características para asegurar que representan la realidad del problema y el dominio de los datos de éste, las características más importantes son [24]:

- **Expresividad:** se refiere a que entre más extenso sea el modelo conceptual mejor se podrá representar la realidad, por lo que el modelo conceptual será más expresivo.
- **Simplicidad:** entre más simple sea el modelo más fácil será de entender para los diseñadores y usuarios de las aplicaciones de bases de datos.
- **Minimalidad:** se refiere a que ningún concepto del modelo pueda ser expresado mediante otros conceptos.
- **Formalidad:** indica que cada concepto del modelo debe tener una interpretación única y precisa. Si tiene un concepto formal se puede manipular de forma matemática.

Aunque en la actualidad el modelo de bases de datos más usado es el modelo relacional, existen otros modelos, algunos de ellos son los siguientes [23]:

- **Modelo de red:** Este modelo está basado en la estructura de red lineal; en ésta, cada registro hijo puede tener más de un registro padre. Los registros padre son los propietarios del conjunto de registros, el registro hijo es un miembro del conjunto (por ejemplo “alumno” sería un registro hijo de “persona”, por lo que éste sería el registro padre).
- **Modelo jerárquico:** Los sistemas manejadores de bases de datos jerárquicos para la representación lógica de la base de datos utilizan árboles, los registros están representados como un árbol, por lo que no pueden

haber ciclos en ellos. Las únicas relaciones que puede existir entre los registros son, uno a uno y uno a muchos, y al estar representado en forma de árbol si un registro padre es borrado todos sus registros hijos también los serán.

- Modelo orientado a objetos: este modelo representa los datos como objetos, estos modelos están diseñados para trabajar con lenguajes de programación orientados a objetos. Las variables del objeto contienen las propiedades o atributos de éstos. Los métodos del objeto sirven para manipular los datos que contiene dicho objeto. Este modelo se explica con mas detalle en la sección 3.5.3.
- Modelo entidad-relación: es un modelo de alto nivel, consiste en la percepción del mundo real como entidades, que son los objetos así como las relaciones que hay entre estas entidades.
- Modelo relacional: es un modelo de menor nivel que el modelo entidad-relación, en este modelo la representación de los datos y las relaciones que existe entre éstos se representa como tablas. Un registro o tupla es una fila en la tabla, cada columna de la tabla es llamada atributo y no debe haber tuplas repetidas en ésta. Para manipular las tablas existen operaciones como unión, intersección, división y diferencia entre otras.
- Modelo relacional orientado a objetos: este modelo combina características del modelo relacional y el modelo orientado a objetos.

### 3.5.1. Modelo Entidad Relación (E-R)

En este modelo los objetos o cosas que son distinguibles de otras cosas son llamadas entidades, las cuales se describen en la base utilizando atributos (las cuales son características que poseen los objetos). Una entidad puede ser concreta (por ejemplo un animal o una computadora) o abstracta (por ejemplo: un concepto). Para cada atributo hay un conjunto de valores permitidos, este es llamado dominio.

Un conjunto de entidades son varias entidades que comparten los mismos atributos. Las relaciones son asociaciones entre las entidades y éstas también pueden tener atributos.

En las relaciones binarias (que involucran sólo dos entidades) existen diferentes cardinalidades, por ejemplo si se toman dos entidades A y B distintas, las cardinalidades de las relaciones binarias que puede haber entre ellas son:

- Uno a uno: en este tipo de relación binaria cada elemento de A solo puede relacionarse con un elemento de la B.
- Uno a varios: un elemento de A se puede relacionar con cualquier número de elementos en B.
- Varios a uno: en este tipo de relación cualquier cantidad de elementos de A puede relacionarse solo con un elemento de B.
- Varios a varios: un elemento de la entidad A se puede asociar con cualquier cantidad de elementos en la entidad B y viceversa.

La estructura lógica completa de la base de datos se puede representar mediante un diagrama E-R, el cual tiene como componentes más comunes [23]:

- Rectángulos: representan las entidades de la base de datos.
- Elipses: representan los atributos o características de las entidades.
- Rombos: representan las relaciones entre las entidades.
- Líneas: representan las uniones de los atributos con las entidades y las uniones de las relaciones con las entidades.

Sin embargo existe una gran variedad de componentes de los diagramas E-R. A continuación se muestra la figura 1.3 con algunos componentes.

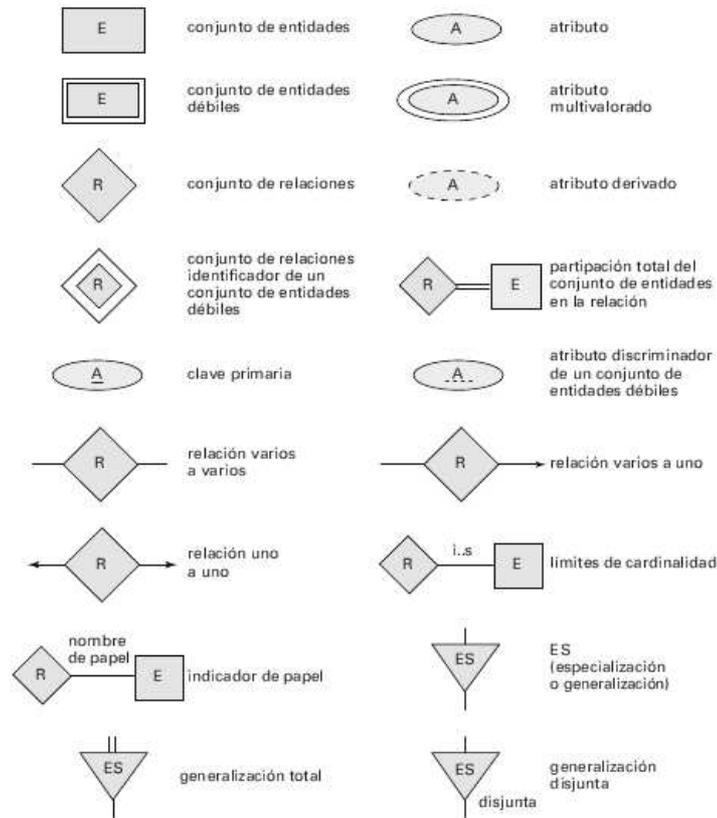


Figura 3.3: Componentes de diagrama E-R [23].

### 3.5.2. Modelo Relacional.

Este modelo es actualmente el más utilizado por las aplicaciones que procesan datos y necesitan manejar una base de datos.

Consiste en un conjunto de tablas que en este modelo son llamadas relaciones, cada una de estas contiene columnas que son llamadas atributos y cada uno de éstos tiene un dominio, que especifica los valores que pueden estar contenidos en los atributos.

Los renglones de las tablas son llamados tuplas y los datos contenidos en la base de datos en un determinado momento son llamados ejemplares.

Para que los usuarios puedan solicitar información a la base de datos existe el lenguaje de consulta, este puede ser de dos tipos[70]:

- Lenguajes procedurales: en este tipo el usuario especifica las instrucciones que se deben seguir para obtener los datos solicitados.
- Lenguajes no procedurales: el usuario sólo indica la información requerida, sin importar como se obtenga está.

En este modelo de bases de datos se pueden realizar operaciones como eliminación de datos, inserción, actualización y consulta de datos. También se pueden crear vistas (relaciones virtuales) para que los usuarios tengan una visión de los datos individualizada.

### 3.5.3. Modelo orientado a objetos.

Este modelo surgió para resolver las limitaciones que ha enfrentado el modelo relaciones con respecto a algunas aplicaciones. Es ocupado con mayor frecuencia con el paradigma de la programación orientada a objetos.

En un sistema en el que se desea obtener todos los detalles de una entidad de la base de datos, (por ejemplo la dirección de una persona), si se utiliza el modelo relacional se tendría que escribir la consulta con todos los atributos de la entidad.

En el modelo orientado a objetos, cada objeto es un ejemplar de una entidad de la base de datos y cada objeto se puede asociar con<sup>5</sup>:

- Conjunto de variables: son los atributos de las entidades y contienen los datos del objeto.
- Conjunto de métodos: son funciones implementadas en un lenguaje de programación orientado a objetos, estos pueden o no devolver un valor como respuesta.
- Conjunto de mensaje: son las llamadas que se realizan entre los objetos y éstas pueden o no tener parámetros.

Los métodos de los objetos pueden ser sólo de lectura o actualización. Los objetos que tienen las mismas variables, los mismos métodos y los mismos mensajes se agrupan en clases, evitando así repetir el código utilizado para éstos.

Las entidades de la base de datos son representadas por las clases en el modelo orientado a objetos. Algunas relaciones que existen entre las entidades de las bases de datos se pueden representar en el modelo orientado a objetos como lo es la herencia entre las clases.

---

<sup>5</sup>Silberschatz Abraham, F. Korth Henry, Sudarshan S., **Fundamentos de Bases de Datos**, 2002, España, pág. 194.

Cada objeto de una clase tiene identidad propia, la cual esta definida por los siguientes aspectos:

- Valor: este valor puede ser la llave primaria de alguna tupla de la base de datos.
- Nombre: es un identificador proporcionado por el usuario, de esta manera cuando el usuario desee manipular la información que contiene el objeto podrá hacerlo mediante su nombre y los métodos de este.
- Identificador: cada objeto al momento de ser creado recibe un identificador asignado por el sistema.

### 3.6. Lenguaje de Consulta Estructurado. (SQL)

Para realizar consultas a las bases de datos, los sistemas manejadores de bases de datos utilizan un lenguaje llamado Lenguaje de Consulta Estructurado (Structured Query Language, SQL). Fue creado a principios de 1970 por IBM, su primer nombre fue Sequel y actualmente es el lenguaje estándar de las bases de datos relacionales.

Los principales componentes del lenguaje de consulta estructurado son:

- Lenguaje de definición de datos: con este componente se puede realizar la definición del esquema de la base, la eliminación o modificaciones de esquemas.
- Lenguaje interactivo de manipulación de datos: proporciona los medios necesarios para realizar las consultas a la base de datos, se puede también manipular los datos.
- Control de transacciones: este componente contiene las instrucciones necesarias para definir el comienzo y el final de una transacción.
- Autorización: este componente de SQL permite especificar los derechos de acceso a las relaciones y a las vistas.

La estructura básica de una expresión en SQL consta de las siguientes cláusulas:

1. Selección (Select): esta cláusula se utiliza para listar los atributos deseados de una consulta.
2. De (From): por medio de esta cláusula se listan las relaciones que serán utilizadas en una consulta. Pueden ser sólo una o varias relaciones.

3. Donde (Where): esta cláusula incluye los atributos de las relaciones que serán analizadas en la sentencia SQL.

Las consultas a la base de datos por medio de SQL se pueden realizar desde lenguajes anfitriones, usando SQL incorporado o dinámico.

SQL permite actualizar, borrar e insertar información en la base de datos. Para asegurar que la integridad de los datos se cumpla SQL proporciona dentro de su lenguaje las instrucciones necesarias para realizar transacciones, con las cuales las secuencias de instrucciones que pertenezcan a la transacción deben ser atómicas, es decir todas deben realizarse o ninguna de ellas.

### 3.6.1. SQL incorporado

Cuando se realizan consultas a la base de datos desde un sistema creado con un lenguaje de programación de propósito general, a menudo son más complicadas que si se realizan en el sistema manejador de la base de datos.

Debido a que las consultas a la base de datos por medio de un lenguaje de propósito general son necesarias, por el procesamiento de la información que esta requiere (mostrarla en pantalla, imprimirla, analizarla, modificarla, entre otras), se llama SQL incorporado a las estructuras de SQL que se utilizan en el lenguaje de propósito general.

En un sistema que contiene SQL incorporado la consulta se realiza a través del sistema manejador de bases de datos y el resultado de la consulta es procesado por el sistema desarrollado en el lenguaje de propósito general.

## Capítulo 4

# Sistemas de Información

Las computadoras y las telecomunicaciones han estado cambiando radicalmente las formas de trabajo, la comunicación y la mayor parte de las actividades diarias en la última década. También han ayudado a mejorar la calidad de vida; sin embargo, también han creado algunas desventajas para el ser humano, como son la violación de la privacidad de los datos personales, ataques a la información con el objetivo de modificar o eliminar cierta información y la gran dependencia que sigue en aumento de recursos computacionales. Tal es el valor de la información en la actualidad que existen autores que llaman a esta época *“la era de la información”*[72, 28].

Los sistemas de información brindan una mejor seguridad de la información, mejor servicio, mayor eficiencia y costos menores.

### 4.1. Información

La cantidad de datos almacenada en las computadoras alrededor de todo el mundo en la actualidad es incalculable, sin embargo cabe aclarar que no todos esos datos se convierten en información, ya que únicamente los datos son transformados en información al ser procesados por algún sistema de información.

Para que la información pueda ser manipulada y así se pueda obtener conclusiones útiles, es necesario que la información sea obtenida de manera selectiva, requiriendo información primaria y secundaria.

La información primaria puede ser obtenida por medio de[30]:

- Entrevistas personales: es la técnica que permite conocer y evaluar la personalidad, motivación, actitudes o estudios de las personas. Trata de recopilar la información relacionada con determinada finalidad.

Consiste en una serie de preguntas que realiza un entrevistador y de este depende la información obtenida. La mayor parte de las veces esta técnica se usa como complemento de otra técnica .

- **Observación:** es una técnica que consiste en observar personas, fenómenos, hechos, acciones o situaciones, por lo general los sujetos del estudio no están conscientes de que son observados. Esta técnica permite obtener información precisa a bajo costo y además es fácil de aplicar.
- **Encuestas:** es una técnica que puede ser aplicada a un gran número de personas, se requiere de una planeación y diseño de las preguntas que conforman la encuesta, y se determinan las posibles variables de las respuestas, facilitando así el análisis de los resultados.
- **Muestreo:** esta técnica es utilizada principalmente cuando no se dispone del tiempo o recursos económicos deseados para entrevistar o encuestar a un gran número de personas, consiste en seleccionar una muestra de la población, logrando realizar observaciones al azar.

La información secundaria es aquella que ya se encuentra almacenada y disponible para su consulta, como puede ser: publicaciones, información bibliográfica, artículos, entre otros.

## 4.2. Definición Sistemas de Información. (S.I.)

Un **sistema de información (SI)** es un conjunto de componentes relacionados entre sí, para manipular, analizar y procesar datos e información con el objetivo de obtener retroalimentación útil y de esta manera facilitar el cumplimiento de los objetivos de las organizaciones.

A diario interactuamos con sistemas de información, como son los cajeros automáticos, las cajas del supermercado, entre otros; por esta razón actualmente varias compañías invierten mucho dinero en tecnologías de la información, pues cada día los sistemas de cómputo están más inmersos en la vida cotidiana.

Los individuos participantes en el desarrollo de un sistema de información son[27]:

- **Propietarios:** son aquellas personas que promueven el desarrollo de los sistemas de información, se encargan de definir el presupuesto y el plazo para el desarrollo.

- Usuarios: son las personas a las que está dirigido el sistemas de información, son aquellos que capturan, validan, introducen, seleccionan y almacenan datos e información.
- Diseñadores: son los encargados de solucionar las restricciones y necesidades de los usuarios mediante tecnología (tecnologías de bases de datos, de comunicación, Web o de seguridad y privacidad).
- Desarrolladores: son los encargados de construir el sistema de información de acuerdo a las especificaciones del diseño.
- Administrador del proyecto: es el encargado de supervisar, planificar y controlar el desarrollo del sistema de información.

#### 4.2.1. Estructura de un sistema de información.

La estructura de un sistema de información esta conformada por [28]:

- Información: son los datos ya procesados por el sistema de información y transformados en información relevante.
- Hardware: son los componentes físicos de la máquina; es decir, los periféricos de entrada (ratón (mouse), teclado, micrófono, entre otros) y salida (monitor, impresora, entre otros), el CPU, las unidades de almacenamiento, entre otras.
- Redes de comunicaciones: son los elementos que permiten mantener unidos o con comunicaciones a todos los componentes del sistema que desean manipular, crear o borrar información. Un medio de comunicación puede ser: fibra óptica, infrarrojo, señales de radio, cable de cobre, entre otros.
- Personas: conforma a los participantes del desarrollo de software, como son los individuos que desarrollan y mantienen el sistema, el administrador del sistema y los usuarios de éste.
- Software: “*Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación*”<sup>1</sup>. Se tratará con más detalle este tema en el capítulo 5.

---

<sup>1</sup>Lewis G., **What is Software Engineering?**, DataPro (4015), 1994, pág 1.

- Bases de datos: “Una base de datos es una colección de datos interrelacionados”<sup>2</sup>, dicho tema se describió en el capítulo 3.

#### 4.2.2. Ciclo de vida de un sistema de información.

Para el desarrollo de un *sistema de información* se puede reemplazar o crear un sistema diseñado para lograr los objetivos o las necesidades de un grupo de usuarios [30].

Durante el desarrollo del *sistema de información* se deben tener en cuenta los siguientes principios:

- Utilizar una estrategia de resolución de problemas, (para lo cual se pueden utilizar las metodologías de desarrollo de software)
- Incluir a los usuarios del sistema.
- Gestionar los procesos y el proyecto.
- Diseñar sistemas que puedan ser mantenidos fácilmente y que puedan crecer: durante la vida del SI pueden ocurrir cambios y aparecer nuevos requerimientos o necesidades, si el sistema no está diseñado para crecer, la nueva necesidad se resuelve mediante parches, o duplicación de código, lo cual es una mala práctica de programación.

El proceso general de desarrollo de un *sistema de información* consta de:

- Conocer los requerimientos del sistema (estos en ocasiones cambian durante el desarrollo del sistema), los datos que manejará, cómo se manipulará estos datos y quién recibirá la información.
- Una vez que se ha decidido que el sistema es viable, se deben establecer los costos, la infraestructura y las condiciones legales.
- Interactuar con los usuarios, así se conocerá si las funciones que realiza el SI necesita mejoras.
- Diseñar los componentes del *sistema de información*, como pueden ser la interfaz, la lógica del programa, la interacción con la base de datos, entre otros.

---

<sup>2</sup>Silberschatz Abraham, F. Korth Henry, Sudarshan S., **Fundamentos de Bases de Datos**, 2002, España, pág 25.

- Seleccionar el lenguaje en que será desarrollado el sistema, para esto debe tomarse en cuenta las funcionalidades del *sistema de información*, los componentes que necesita y las ventajas del lenguaje con respecto a las funciones del *sistema de información*.
- Posteriormente se construye el sistema, y se programan las funciones definidas por el *sistema de información*.
- Por último se brinda mantenimiento a éste, para que de esta manera el sistema siga cumpliendo con su funcionalidad.

### 4.2.3. Lenguajes de programación estructurados

Los lenguajes estructurados surgieron en los años 60, debido a una necesidad de resolver los problemas a los que se enfrentó la programación lineal, como fueron la creación de programas más complejos en esa época y fáciles de mantener.

Estos lenguajes están limitados en palabras y construcciones, por lo cual se tiene más precisión y claridad, evitando ambigüedades. Sus principales estructuras son la secuencia, la instrucción condicional y la iteración. Si se requiere de alguna estructura más compleja está puede ser construida con las tres estructuras principales.

La estructura de estos programas es jerárquica lo que facilita leer, hacer seguimiento de fallos y entender los programas estructurados. Debido a la estructura en bloques de código de los programas desarrollados con estos lenguajes se tiene como ventaja la fácil documentación de estos, sin embargo también se tiene como desventaja el tamaño de los bloques ya que si estos son de gran tamaño se complica el entendimiento y seguimiento del programa.

#### 4.2.3.1. Lenguaje C

Fue creado en 1969 por Ken Thompson y Denis Ritchie. Este lenguaje ha evolucionado de manera paralela al sistema UNIX y ahora es un lenguaje de propósito general.

C es un lenguaje de medio nivel, por lo tanto tiene independencia de la arquitectura del sistema, a su vez es un lenguaje portable, ya que los programas desarrollados en C se pueden adaptar de una computadora a otra[32].

Es uno de los lenguajes más populares<sup>3</sup> para crear software en sistemas

---

<sup>3</sup>El índice TIOBE, que se encarga de medir el crecimiento de los lenguajes de programación, lo ubica en el lugar 2.

y es uno de los lenguajes más antiguos, por tal motivo muchos sistemas de décadas anteriores se encuentran implementados en este lenguaje.

Es muy útil para desarrollar compiladores y sistemas operativos, ya que tiene características de los lenguajes de bajo nivel y dispone de las estructuras de los lenguajes de alto nivel. Los tipos de datos con los que trabaja C son tratados directamente por el hardware de las computadoras, por lo que el código generado es muy eficiente. Otras ventajas que ofrece son control de flujo (con estructuras como *if-else*, *while*, *for*, *do*, *break*) y una amplia variedad de estructuras de datos. Los mecanismos para tratar tipos de datos que no sean los básicos (caracteres, números enteros y flotantes) deben ser implementados por el programador, lo cual se compensa con las bibliotecas disponibles en este lenguaje.

Las funciones implementadas en C pueden regresar como resultado valores de los tipos básicos de datos o apuntadores, además pueden ser llamadas o invocadas recursivamente. Por si mismo C no cuenta con un recolector de basura, este tiene que ser llamado explícitamente.

Por la razón que C es un lenguaje relativamente pequeño es fácil de aprender y entender, logrando así implementar las funciones que sean necesarias para realizar tareas más complicadas. El polimorfismo<sup>4</sup> y el encapsulamiento<sup>5</sup> de los datos se puede simular en C utilizando los punteros a funciones y las variables estáticas.

---

**Algoritmo 4.1** Programa “hola mundo” en C.

---

1. `#include "<stdio.h>"`
2. `int main() {`
3. `printf("Hola mundo");`
4. `return 0;`
5. `}`

---

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> (09/06/2011 11:30 am)

<sup>4</sup>Es la habilidad donde diferentes clases o programas con diferentes jerarquias contienen métodos conceptualmente similares, pero con diferentes operaciones, para cada programa. [69]

<sup>5</sup>Es el ocultamiento del estado de los datos contenidos en un programa, logrando así conocer su estado o alterarlo solo con los métodos definidos por el programa. [http://es.wikipedia.org/wiki/Encapsulamiento\\_informática](http://es.wikipedia.org/wiki/Encapsulamiento_informática) (16/07/2011 14:54)

El ejemplo anterior es un programa clásico, debido a que no es realmente complicado programarlo, en este se puede observar que la sintaxis de C es sencilla, delimitando las funciones con llaves y las instrucciones con punto y coma, aunque se puede realizar programas muy sencillos con C, también se pueden desarrollar programas complejos y eficientes.

### 4.3. Sistemas de Información Web

Desde la aparición del World Wide Web (WEB) y el Lenguaje de Marcado de Hipertexto (HyperText Markup Language, por sus siglas en inglés HTML), el Internet ha tenido un papel importante en los negocios, la educación, la investigación y en muchos otros campos, ya que ha agilizado tramites bancarios y de negocios, ha facilitado la comunicación entre personas y la investigación de temas específicos para la educación y la investigación.

El World Wide Web ha servido para la comunicación global y para compartir información desde cualquier punto del mundo a otro, el crecimiento de éste ha sido sorprendente al igual que la cantidad de sus usuarios, como consecuencia, se han creado una inmensidad de sistemas de información web, para satisfacer las necesidades de información de estos usuarios. Los sistemas de información para Web multiplican los canales de comunicación, por lo que transmiten una mayor cantidad de información, en menor tiempo y en mayor escala geográficamente en comparación a otros sistemas de información.

Los sistemas de información Web al igual que otros sistemas de información que no son para Web deben estar formados por:

- Usuarios.
- Mecanismos de entrada y salida de información.
- Bases de datos.
- Mecanismos de recuperación de la información.

Para desarrollar sistemas para el WEB, se comenzó a introducir lenguajes como Java, C++, C#, ASP, PHP, Perl, Javascript y Ruby los cuales permiten interactuar con el lenguaje HTML con facilidad, y con ayuda de estos lenguajes se obtuvieron sistemas de información más complejos y mantenibles, que pueden ser accedidos desde muchos lugares en el mundo. Estos sistemas de información para el Web han permitido que el comercio global sea más eficiente, que los sistemas se puedan comunicar con más flexibilidad y en un tiempo óptimo.

En la actualidad los sistemas computacionales son mucho más grandes, complejos y útiles, además cumplen con un mayor número de necesidades de los usuarios, por lo que las necesidades de programación actuales son mayores respecto a las necesidades existentes en los años 60's, por esta razón las técnicas de programación lineal y estructurada ya no son suficientes. Como consecuencia de esto se han creado nuevas técnicas, como la programación orientada a objetos y marcos de trabajo que facilitan el desarrollo de sistemas, específicamente existen marcos de trabajo enfocados al desarrollo de aplicaciones Web.

### 4.3.1. Interfaz de entrada común (CGI)

Una interfaz de entrada común por sus siglas en inglés Common Gateway Interface, fue desarrollado conjuntamente por el Centro Nacional de Aplicaciones de Supercómputo (por sus siglas en inglés National Center for Supercomputing Applications, NCSA) y el Laboratorio europeo para partículas físicas (European Laboratory for Particle Physics, CERN).

*“CGI es una tecnología WEB que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. Es un mecanismo de comunicación entre el servidor web y una aplicación externa”<sup>6</sup>.*

*“CGI busca proporcionar garantías razonables de que la entrada del usuario, en especial los envíos de formularios, no se perderán debido a las limitaciones del sistema operativo del servidor”<sup>7</sup>.*

La creación de CGI's fueron las primeras prácticas para crear contenido dinámico en los sistemas web. El flujo de un CGI es el siguiente:

1. El servidor web, realiza una petición del cliente (el usuario solicita abrir una página por medio de la dirección URL), a un programa externo. El programa externo esta escrito en cualquier lenguaje de programación que sea soportado por el servidor, algunos de los más utilizados son Perl y C.
2. La salida del programa externo es enviada al cliente en un archivo estático.

---

<sup>6</sup>Revuelta Domínguez Francisco Ignacio, Pérez Sánchez Lourdes, **Interactividad en los entornos de formación on-line**, Editorial UOC, 2009, 179.

<sup>7</sup>Thomas Boutell, **CGI programming in C & Perl**, Addison-Wesley Professional, 1996, 402 páginas.

3. La aplicación o CGI realiza sus funciones, mediante estas funciones se crea un objeto MIME<sup>8</sup> en el que se escribe la salida.
4. Posteriormente si la aplicación ha terminado sus funciones, el servidor envía la información producida al cliente, para que este pueda observar la página solicitada.

### 4.3.2. Lenguajes de programación orientados a objetos.

Los lenguajes de programación orientados a objetos surgieron para resolver las complicaciones enfrentadas por los lenguajes de programación estructurados al intentar crear sistemas más grandes y complejos, que cumplieran con las nuevas necesidades de los usuarios, las cuales han ido adaptándose a las nuevas tecnologías, como lo es la Web.

Este paradigma de programación está basado en la división del sistema en pequeñas unidades lógicas de código, agrupando estas unidades con características parecidas.

Los principales componentes de un lenguaje de programación orientado a objetos son:

- **Objetos:** son los objetos del mundo real que se desean describir, tienen características y comportamientos. Son las unidades más pequeñas de los sistemas, independientes y se comunican mediante mensajes.
- **Clases:** es la agrupación de los objetos que tienen las mismas características.
- **Mensajes:** son las acciones que pueden realizar los objetos de una determinada clase.

Algunas ventajas de este paradigma de programación son la abstracción y el encapsulamiento de los datos.

### 4.3.3. Ruby

*“Un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad. Su elegante sintaxis se siente natural al leerla y fácil al escribirla.” [35]*

---

<sup>8</sup>Extensiones multipropósito de correo de internet, por sus siglas en inglés Multipurpose Internet Mail Extensions, es una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario.

Este lenguaje de programación se dio a conocer en 1995, sin embargo fue hasta el 2006 que empezó a tener un crecimiento masivo<sup>9</sup>. Este lenguaje esta formado por mezclas de Perl, Smalltalk, Eiffel, Ada, y Lisp. En Ruby todo es visto como un objeto, por esta razón se le puede agregar propiedades y acciones a todo el código.

Ruby es un lenguaje flexible ya que le permite a sus usuarios modificarlo,[40] (permite introducir a las clases nuevos métodos en tiempo de ejecución, lo cual no es posible en otros lenguajes como Java y C++), otra característica interesante de Ruby es que las clases pueden incluir módulos debido a la herencia entre clases que se tiene en este lenguaje, este tipo de herencia es vista por algunos usuarios como herencia múltiple<sup>10</sup>.

Como en otros lenguajes orientados a objetos, en Ruby también se cuenta con un manejador de excepciones (errores), un recolector de basura y además se puede incrustar Ruby en programas escritos en otros lenguajes, cuenta con bibliotecas para el manejo de hilos.

Como el lenguaje C, Ruby también es portable y los sistemas desarrollados en este lenguaje son altamente mantenibles.

Un ejemplo del clásico “hola mundo” en Ruby, es decir, el algoritmo 4.2 imprime en pantalla el mensaje “Hola mundo”, lo más destacado en este programa es la sencillez con la que se logra imprimir el mensaje deseado.

---

**Algoritmo 4.2** El programa “Hola Mundo” en Ruby.

---

1. *print "Hola Mundo"*

---

#### 4.3.3.1. Ruby on Rails

Es un marco de trabajo muy popular de Ruby para desarrollo WEB. En el 2006 se publicaron más libros de Rails que libros de marcos de trabajo para Java [44]. Este marco de trabajo ha incrementado el uso de Ruby, así como el número de programadores que usan este lenguaje. Rails permite a los programadores tener un modelo de base de datos con solo un par de líneas, así Rails se ocupa de los detalles tediosos.

Otras características importantes de Rails son[39]:

- No necesita mucha configuración, en comparación con otros marcos de

---

<sup>9</sup>El índice TIOBE lo ubica en el lugar 12. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> (09/06/2011 11:35 am)

<sup>10</sup>Sitio oficial, Acerca de Ruby. <http://www.ruby-lang.org/es/about/> (08/08/2011 16:06 pm)

desarrollo para .Net o Java que obligan a los programadores a escribir archivos de configuración.

- Crea pruebas simples y automáticas para el sistema mientras se va desarrollando, estas pruebas pueden ser extendidas.
- Tiene tres ambientes principales: desarrollo, pruebas y producción. Estos ambientes tiene pequeñas diferencias, las cuales hacen más fáciles el ciclo de desarrollo del software.

Los componentes más importantes de Rails son [69]:

- Active Record, este componente brinda la base para los modelos (los objetos que se guardan en la base de datos). Provee la funcionalidad CRUD<sup>11</sup>.
- Action Controller, es el componente que se encarga de controlar los controladores de la aplicación, se encarga también de la sesión y despliega las vistas.
- Action View, se encarga de controlar las vistas de la aplicación.
- Action Dispatch, se encarga del direccionamiento de las peticiones hacia el controlador.
- Active Model, permite a Rails utilizar otros marcos de trabajo en lugar de Active Record.
- Active Resource, brinda un marco de trabajo que mapea recursos WEB a objetos locales con las propiedades crear, leer, actualizar y borrar (por si siglas en inglés CRUD).
- Active Support, es un conjunto de clases y bibliotecas estándar para Ruby y Rails.

En general los componentes antes mencionados de Rails permiten que el desarrollador se concentre en programar la funcionalidad y la interfaz del sistema, dejando la configuración y el manejo de recursos (como son las conexiones a la base de datos, el direccionamiento de las acciones y el despliegue de las vistas) al marco de trabajo.

---

<sup>11</sup> Crear, leer, actualizar y borrar (por sus siglas en inglés Create, Read, Update y Delete), son las cuatro funciones básicas en las bases de datos.

### 4.3.3.2. Bibliotecas de Ruby on Rails

A continuación se explica el propósito de algunas bibliotecas (gemas) muy conocidas en la comunidad de desarrollo en Ruby:

- **devise**: esta biblioteca sirve para la autenticación del sistema, permitiendo tener múltiples roles (diferentes modelos o formas de inicio de sesión en el sistema) al mismo tiempo, para el acceso de los usuarios a la base de datos encripta y almacena la contraseña, también permite mandar correos para confirmar instrucciones para la autenticación o para el restablecimiento de contraseñas.
- **Slim**: es un motor de plantilla rápido y ligero con el cual se logra reducir la sintaxis de las partes esenciales. Otra biblioteca parecida a ésta es **haml**, sin embargo Slim es más rápido y comprensible para el usuario. Tiene un estilo minimalista y se basa en el siguiente pensamiento:

*“¿Qué es lo mínimo que se requiere para hacer este trabajo?”[36]*

Código 4.1: Ejemplo archivo HTML con Slim.

---



---

```

1 doctype html
2   html
3   head
4     title Ejemplo de Slim
5     meta name="claves" content="plantilla "
6   body
7     h1 id="titulo" Hola mundo

```

---



---

En el ejemplo correspondiente al Código 4.1 se puede observar las líneas necesarias para crear una página web con el contenido “*Hola mundo*”, simplificando la escritura de las partes HTML.

- **haml**: (lenguaje abstracto de marcado en HTML, por sus siglas en inglés, HTML abstracción Markup Language) esta biblioteca sirve para expresar la estructura de los documentos XHTML o XML de una manera sencilla utilizando sangría en lugar de las etiquetas de cierre de HTML y permitiendo que las líneas de Ruby se incorporen fácilmente.

En el Código 4.2 se muestra una sección del documento HTML, en el cual se imprimen en la pantalla en la columna izquierda la fecha (línea 2), la dirección del usuario, y en la columna derecha se imprime el correo y la biografía del usuario actual.

Código 4.2: Ejemplo archivo HTML con haml.

---

```

1 #perfil
2   .columna.izquierda
3     #fecha=imprime_fecha
4     #direccion=usuario_actual.direccion
5   .columna.derecha
6     #email=usuario_actual.email
7     #biografia=usuario_actual.biografia

```

---

- **show\_for**: esta biblioteca sirve para mostrar rápidamente la información de un modelo. Cada dato del modelo puede ser configurado mediante etiquetas de HTML como *content\_html*<sup>12</sup> y *wrapper\_html*<sup>13</sup>. Un ejemplo de esto es el siguiente:

Código 4.3: Ejemplo archivo HTML con show-for.

---

```

1 <%= show_for @admin do |a| %>
2 <%= a.attribute :nombre %>
3 <%= a.attribute :confirmado? %>
4 <% a.value :biografia %>
5 <% end %>

```

---

El Código 4.3 es un ejemplo de como se muestra el nombre y la biografía en una instancia de la clase administrador, una vez que éste ya ha sido confirmado por algún método, en la línea 1 con `show_for` se inicia la construcción del formulario para mostrar los datos y en las líneas 2-4 se puede observar que se imprimen los datos del administrador.

- **compass**: esta gema es un marco de trabajo para lograr una fácil creación de estilos, los archivos creados con **compass** tienen extensión `sass` en lugar de `css`.
- **simple\_form**: el objetivo de esta biblioteca es no tocar o modificar la manera de definir el diseño de la vista del sistema, ayuda a crear los formularios de manera automática, también permite añadir, desactivar o configurar etiquetas de HTML. Un ejemplo de esto es:

---

<sup>12</sup>Esta etiqueta indica el tipo del contenido del documento, por lo general en un documento HTML el tipo de contenido es "text/html". [38]

<sup>13</sup>Permite incluir objetos externos dentro de otros documentos HTML. [38]

---

Código 4.4: Ejemplo archivo HTML com simple-form.

---

```
1 <%= simple_form_for @usuario do |f| %>
2 <%= f.input :usuario %>
3 <%= f.input :password %>
4 <%= f.input :descripcion, :as => :text %>
5 <%= f.button :submit %><% end %>
```

---

En la línea 1 del código anterior se observa que se crea el formulario para una instancia de la clase usuario y en las líneas 2-4 se crean los campos de texto donde se recibirán los datos del usuario, para posteriormente enviar los datos.

- **cancan**: esta gema es una biblioteca de autorización con la cual se limitan los permisos de cada usuario para acceder al sistema. Se definen en un solo archivo y son leídos por el sistema.

## Capítulo 5

# Metodologías de desarrollo de software

Los costos del software a menudo tienen una gran influencia en el costo de una computadora personal, e incluso si el software está desarrollado con el propósito de tener una larga vida, el costo del mantenimiento es aún más alto. La ingeniería de software trata de disminuir la imprecisión en la planificación de un proyecto, estimar los costos y facilitar el mantenimiento de las aplicaciones usando *metodologías de desarrollo de software*.

Antes de definir y centrarnos específicamente en las metodologías de desarrollo de software, es importante mencionar que en la actualidad debido al amplio uso de la computadora, la ingeniería de software ha tenido un gran impacto en la sociedad, ya que aplicaciones de cómputo como el correo electrónico y las redes sociales (Facebook, Hi5, entre otras) permiten a las personas interactuar entre ellas de nuevas formas, por la razón anterior resulta útil definir en principio a la ingeniería de software.

### 5.1. Ingeniería de Software.

Al área de la ingeniería de software le concierne cada aspecto en la producción de software (análisis de requerimientos, diseño, implementación y pruebas), además ofrece principios y métodos, así como la documentación necesaria para tener el control sobre los avances o mejoras en el desarrollo de la aplicación y así comprender el funcionamiento y uso del software con

el fin de que éste sea rentable<sup>1</sup> y de calidad<sup>2</sup>.

Los principios de la ingeniería de software son[42, 44]:

- *Generalidad*: descubrir los aspectos más importantes y generales de un problema.
- *Abstracción*: identificar las propiedades esenciales para brindar una descripción simplificada de un sistema, enfatizando las propiedades del sistema más importantes.
- *Modularidad*: dividir el problema en partes más pequeñas.
- *Incrementabilidad*: el software se construye aumentando sus funcionalidades en periodos, de tal forma que en cada periodo se le agreguen funcionalidades al sistema.

Las actividades que se realizan en el proceso de desarrollo de software varían dependiendo de la organización y del tipo de sistema a desarrollarse. Algunas de las etapas de la ingeniería de software son[41, 44]:

- **Especificación de requerimientos**: en esta etapa se establecen todos los requisitos funcionales y no funcionales del software, así como las restricciones del software y los criterios para validarlo, ya que muchas veces el cliente cree que sabe exactamente lo que quiere (en otras palabras cómo funcionará el software), sin embargo en la mayoría de los casos estos requisitos resultan ambiguos o bien incompletos y por lo tanto el software no es el deseable por el usuario final. Los requerimientos se clasifican en[41]:
  - *Requerimientos funcionales*: son las entradas, salidas, cálculos y funciones, es decir, describen el comportamiento interno del software, son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares, y a su vez describen la funcionalidad o los servicios que se espera que éste provea.
  - *Requerimientos no funcionales*: son las restricciones y los atributos o características del software. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Son aquellos requerimientos que se refieren a las propiedades emergentes de éste como

---

<sup>1</sup> Que proporcione beneficios y ganancias a largo o corto plazo. [52]

<sup>2</sup> Es la totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas. [61]

la fiabilidad<sup>3</sup>, la portabilidad<sup>4</sup>, la respuesta en el tiempo y la capacidad de almacenamiento. Definen las restricciones del sistema así como la capacidad de los dispositivos de entrada y salida, así como la representación de los datos que se utilizan en la interfaz del sistema.

- **Diseño del sistema:** consiste en el diseño de los componentes de la aplicación, muchas veces se utilizan patrones de arquitectura de software (monolítica, cliente-servidor, arquitectura de tres niveles, entre otras), en esta etapa se pueden utilizar diagramas de bases de datos, así como diagramas de secuencia y en ocasiones diagramas de clases.
- **Implementación del sistema:** se realiza la construcción del sistema, la duración de esta etapa puede variar dependiendo del lenguaje de programación utilizado, el diseño, la complejidad y amplitud del sistema.
- **Pruebas:** consiste en comprobar que el software haya resuelto correctamente el o los problemas indicados en la especificación.
- **Documentación:** en esta etapa se realizan los manuales de instalación, de usuario, de diseño, entre otros; los cuales ayudarán a entender el software.

Algunos de los atributos o características, más comunes, que los clientes solicitan para el funcionamiento del software son[42]:

1. Mantenimiento: posteriormente a la entrega del sistema, las actualizaciones y optimizaciones seguirán cumpliendo las especificaciones, también deberá prevenir y corregir los errores, una consecuencia del buen mantenimiento es la facilidad de comprensión. La documentación generada en el proceso de desarrollo del sistema ayuda a lograr la facilidad de comprensión.

*“El mantenimiento del software es la modificación de un producto software después de la entrega para corregir fallos, para mejorar el rendimiento u otros atributos, o para adaptar el producto a un entorno modificado.”[62]*

---

<sup>3</sup> Grado en el que un programa se espera que realice su función con una precisión requerida.

<sup>4</sup> Esfuerzo requerido para transferir un programa de una configuración hardware o entorno software a otro.

2. Confiable: debe tener tolerancia a fallos y recuperación de éstos. Un fallo es cualquier comportamiento no esperado por el software, la generación de fallos depende del código, de su tamaño, de los procesos que realiza, de las tecnologías y herramientas de ingeniería de software usadas.

*“Es la probabilidad de que un programa realice su objetivo satisfactoriamente (sin fallos) en un determinado periodo de tiempo y en un entorno concreto (denominado perfil operacional).”[52]*

3. Eficiente: se puede medir por el número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema, el software no debe desperdiciar los recursos del sistema lo cual se incrementa al tener un buen diseño[51].

La eficiencia en código depende de los algoritmos usados y del estilo de programación, se puede hacer más eficiente el código evitando ciclos anidados, no mezclando tipos de datos, entre otros. Respecto a la eficiencia en memoria, ésta puede ser consecuencia de la eficiencia del código en tiempo de ejecución, ya que al limitar el uso de estructuras como arreglos o simplificando operaciones se reduce el tiempo de acceso a los elementos en memoria.

4. Facilidad de aprendizaje: el sistema debe contar con una interfaz de usuario adecuada y documentación, como para ser capaz de realizar correctamente la tarea que se desea llevar a cabo. Si el programa es claro y sencillo se facilitará su mantenimiento posterior. Se pretende requerir el mínimo esfuerzo necesario para aprender, operar y predecir las salidas del programa de acuerdo a las entradas que reciba[59].

*“Es la facilidad con la que nuevos usuarios desarrollan una interacción efectiva con el sistema o producto. Está relacionada con la predicibilidad, sintetización, familiaridad, la generalización de los conocimientos previos y la consistencia.”[60]*

5. Interfaz de usuario: es una descripción de las características del software, que utiliza elementos de hardware y software para apoyar al usuario en la interacción con el sistema. La interfaz incluye las pantallas, ventanas, controles, menús, metáforas, la ayuda en línea, la documentación y el entrenamiento[59].

*“Las interfaces básicas de usuario son aquellas que incluyen cosas como menús, ventanas, teclado, ratón, los “beeps” y algunos otros sonidos que*

*la computadora hace, en general, todos aquellos canales por los cuales se permite la comunicación entre el hombre y la computadora.” [63]*

Existen diferentes tipos de interfaces de usuario según la forma de interactuar con el usuario, éstas son:[59]

- a) Interfaz alfanumérica: en estas interfaces sólo se muestra texto.
  - b) Interfaz gráfica de usuario: en éstas se representan gráficamente los elementos de control, usando botones, ventanas, entre otras cosas.
  - c) Interfaz táctil: representa gráficamente a los elementos, los cuales se pueden seleccionar al tocar la pantalla del dispositivo.
6. Portabilidad: es la facilidad de transferencia de un ambiente de ejecución a otro, ya sea hardware o software. La portabilidad es la cuestión clave para reducir el costo en el desarrollo de software si se piensan en varias plataformas, ya que no es necesario modificar o desarrollar un nuevo software para adaptar el sistema a un nuevo ambiente de ejecución[51, 44].

*“El código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad menor es la dependencia del software con respecto a la plataforma.”[55, 56]*

En el Cuadro 5.1 se muestra la clasificación del software, así como algunas de las características principales de cada categoría:

Software de sistema	Software de programación	Software de aplicación
Ofrece interfaz de alto nivel.	Ofrece interfaz diseñada solo para el programador.	Ofrece interfaz diseñada para el usuario.
Son herramientas que permiten el mantenimiento de la computadora.	Tiene soporte para varios lenguajes de programación.	Contiene herramientas que ayudan a realizar las tareas deseadas.
Se puede realizar el control y administración de los recursos de la computadora.		Realiza tareas específicas

Cuadro 5.1: Clasificación del software.[28]

## 5.2. Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son marcos de trabajo usados para la estructuración, planeación y desarrollo de sistemas. Existen metodologías tradicionales que se centran en el control del proceso, estableciendo tanto las actividades como las herramientas involucradas y la notación usada en la codificación y la documentación (por ejemplo JSD, SSADM, SADT, entre otras)[42].

Por otro lado tenemos a las metodologías ágiles que se centran en el producto y en la colaboración con el cliente; es decir, con éstas se reduce el tiempo de producción del software manteniendo una calidad alta, esto se logra usando el desarrollo incremental con iteraciones cortas y realizando la más mínima documentación necesaria, además están preparadas para adaptarse a los cambios que se presenten durante el proceso de desarrollo del software (por ejemplo la programación extrema).

### 5.2.1. Metodologías tradicionales

Las metodologías tradicionales, casi todas propuestas antes de los años 80, se centran en llevar una documentación exhaustiva de todo el proyecto y

cumplir con el plan del proyecto. Estas metodologías son guiadas por casos de uso<sup>5</sup>. Existen diversas metodologías tradicionales como son:

- **Modelado en cascada [44][50]:**

También llamado Método Lineal Secuencial o Ciclo de vida Clásico, es uno de los modelos más usados para el desarrollo de software, en éste el flujo es secuencial entre las etapas, las cuales tienen superposición, retroalimentación hacia las capas anteriores (por esta razón las fallas pueden prevenirse en gran medida), una entrada (son requisitos de cada etapa) y una salida (resultado del trabajo realizado en la etapa anterior).

El proceso debe ser planeado, disciplinado y cada objetivo debe cumplirse. La documentación es muy importante y se debe realizar durante todo el proceso, por esta razón el sistema entero es descrito y registrado en los documentos generados, los cuales pueden llegar a ser cientos de páginas. Este método asume que una vez especificados los requerimientos éstos no cambiarán más, lo cual casi nunca sucede. La evolución del sistema posterior a la entrega de éste debe considerarse dentro del ciclo de vida, ya que en general los sistemas de software casi siempre se encuentran más tiempo en etapas de servicio que en desarrollo.

- **Prototipo evolutivo[47]:**

Es un modelo iterativo, en el cual se desarrollan versiones del software, donde cada nueva versión es más completa y con más funcionalidades. Se da por hecho que los requerimientos se encuentran en un cambio continuo durante el proceso de desarrollo. El objetivo radica en desarrollar un software flexible y expandible, así si los requerimientos cambian, con un mínimo de esfuerzo y tiempo se lograrán hacer los cambios. Este modelo busca reemplazar el viejo sistema con uno nuevo, bajo la perspectiva de satisfacer los nuevos requerimientos lo más pronto posible. Las etapas de este modelo tienen por objetivo extender el producto de software operacional en cada iteración.

- **Prototipos [44, 46]:**

Este modelo pone énfasis en la etapa de especificación de requerimientos a través de la construcción de prototipos que aproximan al usuario

---

<sup>5</sup>Son las funciones que puede realizar el software, en estas se identifican sus entradas, salidas y los actores que participan en cada función [47].

con la idea del sistema final, lo anterior se logra con iteraciones sucesivas y por lo general el equipo de desarrollo es reducido. Este modelo es útil cuando el cliente conoce los objetivos pero no conoce los requisitos de entrada y salida. Con este modelo se reduce el costo y el riesgo de construir un sistema que no satisfaga las necesidades de los usuarios. En la etapa de diseño se tienen dos fases principales que son: la producción de documentación del diseño, en la cual se especifican y describen la estructura del sistema, el flujo del mismo y la interfaz de usuario, por otro lado en la segunda fase la producción de lo requerido para el mantenimiento del sistema, si eventualmente se requiere de más mantenimiento entonces el método de prototipos se repite y se definen nuevamente los requerimientos. El prototipo es modificado y evaluado repetidamente hasta que los requerimientos del sistema han sido satisfechos.

Algunas desventajas de este modelo son:

- Para que realmente sea efectivo el modelo debería de ser infinito al iterar, lo cual lo hace poco efectivo .
- El usuario puede crearse expectativas al ver el prototipo, pero por ser un prototipo rápido muchas veces no se toma en cuenta la calidad.
- El mantenimiento puede ser de largo plazo antes de estar terminado.

■ **Incremental [42]:**

La forma de reducir los riesgos en este modelo es construir sólo una parte del sistema, dejando así algunos requerimientos del sistema para los niveles posteriores del proceso de desarrollo, de esta forma también se determina si los requerimientos son correctos, e iterativamente se mejora la secuencia evolutiva de versiones hasta que el sistema se encuentra completo. Si una falla es encontrada solo es necesario desechar la última iteración y retomar la versión anterior del sistema. Con este modelo el tiempo de desarrollo del sistema se reduce y decrecen las posibilidades de que los requerimientos del sistema cambien en el proceso. Este modelo es muy usado con la programación extrema. El análisis de la iteración está basado en la retroalimentación del usuario y el análisis de las funcionalidades del sistema. Una de las desventajas de este modelo es que al igual que el modelo prototipos, al tener el contacto directo con el cliente en la iteración, se puede crear una expectativa previa del sistema.

**■ Espiral [42]:**

Este modelo agrupa las mejores características del modelo en cascada y el modelo prototipado, agrega objetivos de calidad y el análisis de riesgos, está basado en los requisitos iniciales, que es donde se centra la atención. También permite iteraciones y si se presenta una falla solo se toma en cuenta la iteración anterior. En cada iteración se identifican los objetivos correspondientes, las alternativas y las restricciones del sistema. Las reducciones de los riesgos se logran dividiendo el sistema en segmentos pequeños y así se obtiene mayor facilidad de cambio en el proceso de desarrollo. Este modelo plantea realizar al inicio del desarrollo varias iteraciones con el propósito de mitigar los riesgos más críticos del proyecto, mediante la realización de prototipos o simulaciones.

Las etapas de este modelo son:

- Determinar objetivos y alternativas.
- Evaluar alternativas y resolver riesgos.
- Desarrollar y verificar resultados.
- Planear la próxima iteración.

En la actualidad, después de más de 20 años de la aparición de las metodologías de desarrollo de software tradicionales, la innovación tecnológica ha sido uno de los grandes avances de la ciencia, por lo que se han creado nuevas metodologías de desarrollo que cumplan con los requisitos, las necesidades y los problemas actuales, así como también se pretende que usen las nuevas herramientas y lenguajes creados en este tiempo, logrando así que sean más ágiles.

**5.2.2. Metodologías Ágiles**

De la programación estructurada se ha pasado a la programación orientada a objetos y programación en Web, de lenguajes de programación como Cobol a lenguajes como C, Java y Ruby. La manera de pensar de la gente va cambiando y en el área de sistemas se ha incrementado la demanda en entregas de software cada vez más rápidas. Las comunicaciones son ahora casi instantáneas y la información fluye en tiempo real, esto se traduce en la implementación de las **metodologías ágiles**. Por lo que a principios de la década de los 90 se creó un *Manifiesto Ágil* donde las principales ideas contenidas en éste son [66]:

- *“Las interacciones entre los individuos son más importantes que las herramientas y los procesos usados”.*
- *“El producto de software funcionando es la medida principal de progreso”.*
- *“La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua del software”.*
- *“Debe asegurarse la capacidad de respuesta ante un cambio”.*
- *“El cliente y los desarrolladores deben trabajar juntos diariamente a lo largo de la duración del proyecto”.*
- *“Realizar entregas del avance del desarrollo del software, de un par de semanas a un par de meses, dándole preferencia a los plazos cortos”.*
- *“Construir proyectos alrededor de individuos motivados, darles el ambiente y el soporte que necesitan, así como confiar en ellos para realizar el trabajo”.*
- *“Los procesos ágiles promueven el desarrollo sustentable. Los patrocinadores, los desarrolladores y los usuarios deben poder mantener un paso constante indefinidamente”.*
- *“La continua atención a la excelencia técnica y un buen diseño realza la agilidad”.*
- *“La mejor arquitectura, diseño y requerimientos emergen de la auto-organización del equipo”.*
- *“En intervalos regulares, el equipo refleja cómo hacer más eficaz el software, entonces ajustan el comportamiento del software a estos intervalos”.*

Retrasar las decisiones tanto como sea posible y la planificación adaptiva son el eje en el que gira la metodología ágil. Las principales ventajas de retrasar decisiones son:

- Reducir el número de decisiones de alta inversión.
- Reducir el número de cambios en el proyecto.
- Reduce los costos de los cambios.

Entre los principales métodos ágiles tenemos el XP (eXtreme Programming), Scrum, Iconix, Cristal Methods, AUP entre otras.

■ **Programación extrema (*eXtreme Programming “XP”*)** [42]:

Esta metodología da más énfasis a la adaptabilidad que a la previsibilidad. Los autores de XP consideran que los cambios de los requisitos durante el desarrollo del software son inevitables, por lo que la adaptabilidad es una mejor opción que intentar definir todos los requisitos al comienzo del proyecto y posteriormente controlar los cambios de los mismos. Sus principales características son[42, 58]:

- *“Desarrollo iterativo e incremental (pequeñas mejoras, una tras otra)”.*
- *“Pruebas unitarias continuas”.*
- *“Programación por parejas (el código escrito de esta manera es revisado y discutido mientras se escribe)”.*
- *“Frecuente interacción del equipo de programación con el cliente o el usuario”.*
- *“Corrección de todos los errores antes de añadir nuevas funcionalidades”.*
- *“Reescribir las partes del código que se pueda mejorar su legibilidad y mantenibilidad pero sin modificar su comportamiento”.*
- *“Todos los programadores pueden corregir y extender cualquier parte del proyecto”.*
- *“Mientras más simple es el proyecto más se reduce el equipo de desarrollo”.*

Algunas de las ventajas de esta metodología son[42]:

- La presión está a lo largo de todo el proyecto y no solo en la entrega final.
- Permite definir cuales son los objetivos en cada iteración.
- Permite tener una retroalimentación de los usuarios.

■ **(SCRUM):**

Surge de un artículo de 1986 de la Harvar Business Review, titulado *“The New New Product Development Game”* de Takeuchi y Nonaka, pero fue formalizado hasta el año 1995 por Ken Schwaber y Mike Beedle.

En esta metodología el desarrollo se realiza en forma iterativa e incremental. Cada ciclo o iteración termina con un software que incorpora una nueva funcionalidad.

Las iteraciones duran entre 2 ó 4 semanas y cada una de estas tiene su propio subconjunto de requerimientos. Los desarrolladores se centran en entregar un software de calidad y la gestión del proyecto se centra en remover cualquier obstáculo que pudiera entorpecer las tareas de los desarrolladores. Esta metodología está basada en los principios de inspección continua, autogestión e innovación. El líder del proyecto lleva a cabo la gestión de convocar diario a una reunión con una duración máxima de 15 minutos, con el propósito de la retroalimentación de las tareas y los obstáculos. Al principio de cada iteración se realiza una reunión de planificación para decidir los requerimientos a desarrollar con sus respectivas prioridades y al finalizar cada iteración se realiza un encuentro retrospectivo aproximadamente de 4 horas con el fin de recabar información acerca de las lecciones aprendidas y datos útiles para las futuras iteraciones.

■ **ICONIX:**

Esta metodología maneja casos de uso, no desecha el análisis y el diseño como lo hace XP. El resultado es concreto, específico y de uso entendible. Por medio de los diagramas de casos de uso se toman técnicas para codificar rápida y eficazmente. Usa en gran medida los diagramas de UML y utiliza las herramientas de este para complementar las herramientas básicas. Una característica importante de esta metodología es el uso de un análisis robusto, lo que reduce el espacio que hay entre el análisis y el diseño, ya que reduce las ambigüedades en las descripciones de casos de uso. Esta metodología se divide en cuatro etapas, en cada nueva etapa es revisada la etapa anterior y actualizada, estas etapas son:

- Análisis de requerimientos: se realiza el análisis de los requerimientos del sistema y se identifican los casos de uso a partir de este. Se pueden realizar los prototipos de interfaces del usuario.
- Diseño preliminar: Una vez identificadas las entidades, se describe la manera en que interactúa el usuario y el sistema. Se realiza un análisis exhaustivo para detectar posibles errores en los casos de uso.
- Diseño detallado: En esta etapa se utiliza el diseño preliminar

para realizar el diseño de clases del sistema, con el que se puede comenzar a desarrollar el sistema.

- Implementación: Se definen las pruebas necesarias para comprobar que el sistema cumple con sus casos de uso y secuencia, finalmente se desarrolla el sistema basándose en estos diagramas.

■ **CRYSTAL CLEAR:**

Creada por Alistair Cockburn, tiene un gran énfasis en la comunicación. Se realizan iteraciones cortas con retroalimentación frecuente por parte de los usuarios/clientes y se dispone de un usuario real para realizar las validaciones sobre la interfaz de usuario y definir los requerimientos funcionales y no funcionales del sistema. Los desarrolladores involucrados en este proceso escogen aquellas fases que les resultan efectivas y mediante un consenso deciden agregar o remover fases.

■ **Proceso Ágil Unificado (*Agile Unified Process “AUP”*)<sup>6</sup>:**

Está basado en disciplinas y entregas incrementables con el tiempo. El ciclo de vida de los proyectos grandes que usan esta metodología es serial y en los pequeños proyectos es iterativo. Las disciplinas de esta metodología son:

- Modelado.
- Implantación.
- Prueba.
- Despliegue.
- Administración de la configuración.
- Administración del proyecto.
- Entorno.

■ **Desarrollo rápido de aplicaciones (*Rapid Application Development “RAD”*) [42]:**

Esta metodología está basada en el desarrollo iterativo y la construcción de prototipos. Se pretende entregar un sistema de alta calidad a costo bajo. El plazo de entrega es de 90 a 120 días máximo. El sistema se parte en pequeños segmentos, con lo cual se logran reducir los riesgos y facilitar cambios durante el proceso de desarrollo. Promueve

---

<sup>6</sup>The Agile Unified Process (AUP). <http://www.ambysoft.com/unifiedprocess/agileUP.html>  
(04/06/2010 11:10 pm)

el uso de herramientas de desarrollo por medio de una computadora, como lo son: constructores de interfaz gráfica de usuario, sistemas manejadores de bases de datos, generadores de código, entre otras. Otra de las características de este método es que el grupo de programadores es pequeño. La atención se centra más en las necesidades comerciales que en las necesidades de alta calidad. Los usuarios participan en gran medida en el diseño del sistema y se realiza la documentación necesaria para el futuro mantenimiento. Este método tiene 3 fases principales, las cuales son:

- **Requisitos:** en esta etapa los usuarios deben establecer las funcionalidades con las que deberá cumplir el sistema, para esto se organizan juntas de planificación de requisitos.
- **Diseño:** en esta etapa los usuarios participan en talleres en los cuales se descomponen las funcionalidades del sistema y se crean las entidades correspondientes a éstos. Una vez identificadas las entidades se crean los diagramas que definen las acciones entre las funcionalidades y los datos para posteriormente realizar el prototipo del sistema.
- **Construcción:** en esta etapa se desarrolla el sistema y se realizan las pruebas necesarias. También se crea la documentación necesaria para manejar el sistema.

Estas son algunas de las metodologías de desarrollo de software más conocidas, y se utilizan de acuerdo a las necesidades de cada proyecto.

En conclusión cada metodología, ya sea ágil o tradicional, tiene su propia perspectiva de cómo llevar a cabo el proceso de desarrollo de software, muchas de éstas utilizan algún patrón para facilitar esta perspectiva. Las metodologías de desarrollo de software ayudan a establecer el orden en que se realizarán las tareas en el proyecto e indican los requisitos de entrada y de salida de cada una de las tareas a realizar. La comunicación es la base de los patrones, esto se refiere a todas las personas del equipo de desarrollo, el cliente y a todas las personas que sean afectadas por la ejecución del desarrollo de software.

### **5.3. Metodologías ágiles VS metodologías tradicionales.**

Algunas conclusiones del tema anterior, ventajas y desventajas de las metodologías tradicionales y las ágiles se pueden expresar en el siguiente

cuadro.

Metodologías Ágiles	Metodologías Tradicionales
El equipo esta preparado para cambios del sistema durante el proceso del desarrollo del software.	Se tiene una mayor resistencia a los cambios en el sistema.
El proceso de desarrollo es menos controlado, por lo tanto tiene menos principios.	El proceso de desarrollo es controlado por las normas que se establecen al principio del proyecto.
El usuario es parte del equipo de desarrollo.	El usuario o cliente interactua con el equipo de desarrollo.
Se necesitan menos desarrolladores en el equipo.	El equipo de desarrollo es grande y el trabajo es distribuido entre estos.
El diseño es sencillo, con la mínima documentación.	El diseño es flexible y extensible, con documentación exhaustiva.
El desarrollo de los módulos del sistema se realiza en parejas, con transferencia de conocimientos.	El desarrollo de los módulos del sistema se realiza de forma individual, con roles y responsabilidades.

Cuadro 5.2: Metodologías ágiles VS metodologías tradicionales.

#### 5.4. Patrones de arquitectura de software.

“Cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y después describe la base de la solución a dicho problema de una forma que se puede utilizar dicha solución un millón de veces sin realizarla dos veces de la misma manera”<sup>7</sup>.

Aunque existe una gran cantidad de patrones de arquitectura de software a continuación solo se mencionan algunos:

- **Modelo Vista Controlador (*Model View Controller* “MVC”):**  
Este patrón enfatiza la escalabilidad del sistema, permite separar la interfaz de usuario de la lógica, ya que ésta es lo que más cambia en un sistema, basándose en tres componentes principales que son:

<sup>7</sup>Alexander C., **The timeless way of Building**, The Oxford University Press, 1979

- Modelo: es el dominio de los datos del sistema, contiene el comportamiento de éstos, brinda información específica acerca de las entidades del sistema. Maneja el registro de las vistas y sus correspondientes controladores.
- Vista: muestra la información de los modelos al usuario en un formato adecuado permitiendo la interacción de éstos con el sistema.
- Controlador: responde a los eventos realizados por el usuario, dándole gestión a los datos que brinda el usuario, puede invocar al modelo o la vista.

■ **Modelo Vista Adaptador (*Model-View-Adapter* “MVA”):**

Es un patrón de arquitectura de software en niveles, este patrón es utilizado en sistemas donde se presentan grandes cantidades de datos de los usuarios, en el cual se separan el modelo (los datos) de la vista (la interfaz de usuario), en este patrón no hay comunicación directa entre el modelo y la vista, como lo hay en el patrón MVC.

Pueden existir varios adaptadores que se encargan de la comunicación entre el modelo y la vista, esto quiere decir que puede haber diferentes lógicas de negocio (comportamiento del sistema).

■ **Arquitecturas Orientadas a Servicios (*SOA*):**

En este patrón la arquitectura del sistema se define mediante la utilización de servicios para cumplir con los requisitos del negocio. Las principales capas del sistema son:

- Aplicaciones básicas: son los sistemas que son desarrollados con cualquier arquitectura o lenguaje de programación, geográficamente en diferentes lugares.
- De exposición de funcionalidades: donde las funcionalidades de la capa de aplicaciones básicas son mostradas como servicios, por lo general como sistemas Web.
- De integración de servicios: esta capa facilita el intercambio de los datos entre las dos capas anteriores.
- De composición de procesos: en esta capa se define el proceso en términos del negocio y sus necesidades.
- De entrega: es la capa final, en donde los servicios son mostrados a los usuarios.

## Capítulo 6

# Análisis de Requerimientos

### 6.1. Objetivo.

El objetivo del sistema es recaudar los datos estadísticos personales, académicos, socioeconómicos y de opinión de los aspirantes, alumnos y egresados de la UNAM, con el fin de conocer los perfiles de los aspirantes de primer ingreso a la UNAM, así como de los alumnos inscritos tanto a nivel bachillerato como en licenciatura.

#### 6.1.1. Requerimientos funcionales.

El sistema esta dividido en diferentes módulos, dependiendo del tipo de usuario, se tiene la siguiente categorización<sup>1</sup>:

1. **Concurso de Selección de Ingreso a Licenciatura:** este módulo está dirigido a los aspirantes por concurso de selección a licenciatura. Comprende el primer concurso realizado en febrero, el segundo concurso realizado en el mes de junio, concurso al sistema de universidad abierta y educación a distancia efectuado también en junio y concurso para el sistema de universidad abierta y educación a distancia realizado en noviembre (modalidad abierta).
2. **Concurso de Selección de Ingreso a Bachillerato:** este módulo se encuentra dirigido a los aspirantes del concurso de ingreso a la UNAM a nivel bachillerato.

---

<sup>1</sup>Las categorías 5-8 aún no se encuentran disponibles en la página de la DGAE.

3. **Pase Reglamentado:** dirigido a todos los alumnos de la UNAM en nivel bachillerato que están por ingresar a nivel licenciatura<sup>2</sup>.
4. **Equidad y Género:** este módulo está dirigido a todos los aspirantes de ingreso a la UNAM en cualquier nivel de estudios y a los alumnos de la UNAM.
5. **Egresados:** este módulo está dirigido a los alumnos egresados a nivel licenciatura de la UNAM.
6. **Opinión de Servicios de Bachillerato:** este módulo está dirigido a todos los alumnos de la UNAM en el nivel bachillerato.
7. **Opinión de Servicios de Licenciatura:** módulo dirigido a los alumnos que actualmente están inscritos en el nivel de licenciatura en la UNAM.
8. **Ingreso de Programas de Becas:** dirigido a todos los alumnos de la UNAM en nivel bachillerato o licenciatura que han solicitado una beca.

Cada módulo tendrá las siguientes características:

- Período: corresponde al periodo o ciclo escolar del concurso de ingreso a la UNAM, consta de cuatro caracteres, dos de estos para indicar el año inicial y los otros dos para indicar el año final del ciclo escolar. (Por ejemplo el ciclo escolar actual es: "1112", que corresponde al ciclo escolar comprendido por los años 2011-2012).
- Concurso: consta de un caracter, los tipos de concurso pueden ser:
  - Pase reglamentado (0).
  - Primer concurso de selección a nivel licenciatura (1).
  - Segundo concurso de selección a nivel licenciatura (2).
  - Tercer concurso de selección a nivel licenciatura (4).
  - Concurso de selección a nivel licenciatura en el sistema universidad abierta y educación a distancia (S)<sup>3</sup>.
  - Concurso de selección a nivel bachillerato (3).

---

<sup>2</sup>Cuando los alumnos de bachillerato de la UNAM terminan el registro de la carrera a la cual desean ingresar a nivel licenciatura, se dice que liberan el pase reglamentado.

<sup>3</sup>Este concurso se realiza en el mes de noviembre, en el cual se ofrece la modalidad de universidad a distancia.

- Concurso de selección a nivel licenciatura en el sistema universidad abierta y educación a distancia (s)<sup>4</sup>.
  - Equidad y Género (P).
  - Ingreso al Programa de Becas (V).
  - Opinión de servicios de nivel licenciatura (O).
  - Opinión de servicios de nivel Bachillerato (H).
- Clave del módulo (Tipo de encuesta): consta de 5 ó 6 caracteres que indican la abreviatura del módulo, ésta se utilizará para distinguir las respuestas en la base.
- PR-08 Pase reglamentado.
  - OSL-08 Opinión de servicios de nivel licenciatura.
  - EGL Egresados.
  - BEC-10 Ingreso al Programa de Becas.
  - CSL-08 Concurso de selección a nivel licenciatura<sup>5</sup>.
  - CSB-08 Concurso de selección a nivel bachillerato.
  - OSB-08 Opinión de servicios de nivel bachillerato.
  - PUEG-10 Equidad y Género.

Todos los módulos tienen secciones comunes, sin embargo, éstas no son homogéneas.

El sistema debe verificar los períodos de vigencia de cada módulo para que puedan ingresar los usuarios al sistema, estos períodos estarán indicados en archivos separados de acuerdo a cada módulo y cada concurso de ingreso a la UNAM, de no cumplirse lo anterior, se mostrará un mensaje en la pantalla indicando que el período expiró.

Se podrá iniciar sesión de manera directa, desde la página de la DGAE o por medio del sistema de entrega de documentación<sup>6</sup>.

Para el ingreso al sistema desde la página de DGAE:

---

<sup>4</sup>Este concurso de ingreso se realiza en el mes de enero y se ofrece el sistema abierto.

<sup>5</sup>Esta clave incluye el primer, segundo y tercer concurso de selección a licenciatura, así como los concursos de selección a licenciatura sistema universidad abierta y educación a distancia (modalidades abierta y a distancia).

<sup>6</sup>Este sistema se encarga de entregar las cartas de aceptación de ingreso a la UNAM y los únicos usuarios que podrán utilizar esta forma de inicio de sesión son los aspirantes seleccionados por concurso, ya sea a nivel licenciatura o a nivel bachillerato o por pase reglamentado.

- Se tendrá una página de inicio de sesión, la cual está clasificada de la siguiente manera, de acuerdo al módulo al que se ingrese; para los módulos de Egresados (EGL), Pase reglamentado (PR) y Opinión de servicios de licenciatura (OSL) se solicitará el número de cuenta y la fecha de nacimiento; para las encuestas de Concurso de selección de licenciatura (CSL) e Ingreso a becas (BEC) se solicitará el folio asignado al registrarse en el concurso de selección y la fecha de nacimiento; para las encuestas de Concurso de selección a bachillerato (CSB) y Opinión de servicios de bachillerato (OSB) se deberán ingresar los datos de folio *COMIPEMS*<sup>7</sup> o número de cuenta y fecha de nacimiento.

Por último para el módulo Equidad y género (PUEG) no se solicitará ningún dato para iniciar sesión, ya que éste será la segunda parte de todos los demás módulos, de esta manera los datos personales ya se habrán solicitado de acuerdo al módulo previamente contestado.

### 6.1.2. Requerimientos no funcionales.

El sistema debe poder visualizarse en al menos los exploradores *Mozilla* e *Internet Explorer*, sin importar el sistema operativo que tenga la computadora (Windows, Linux o Mac OS). Respecto a las necesidades de hardware, por ser un sistema de información para WEB, sólo se requiere que el sistema se encuentre en un servidor al cual se pueda acceder desde cualquier computadora en cualquier punto geográfico con acceso a Internet.

Para almacenar los datos de los usuarios, contemplando que se tendrá una gran cantidad de usuarios se requiere una base de datos capaz de almacenar grandes cantidades de información.

## 6.2. Antecedentes del sistema SIUNE-DGAE.

El Instituto Politécnico Nacional (IPN), que ocupa el lugar 35 en la lista de mejores universidades de Iberoamérica<sup>8</sup>, tiene una deserción del 10 % estudiantes debido a problemas económicos, para disminuir este índice, el Politécnico brinda becas a sus alumnos a través del Programa Nacional de

---

<sup>7</sup>Comisión Metropolitana de Instituciones Públicas de Educación Media Superior. Esta institución asigna un folio a cada alumno al registrarse en el concurso de ingreso a nivel bachillerato. Sitio Oficial COMIPEMS. <http://www.comipems.org.mx/> (008/08/2011 17:04 pm)

<sup>8</sup>SCImago Research Group de la Universidad de Granada.  
[http://www.scimagoir.com/pdf/ranking\\_iberamericano\\_2010.pdf](http://www.scimagoir.com/pdf/ranking_iberamericano_2010.pdf) (15/06/2011 3:00 pm)

Becas (PRONABES) realizando encuestas socioeconómicas las cuales consisten en llenar un formulario impreso<sup>9</sup>. Cabe señalar que los perfiles de los aspirantes resultantes de estas encuestas no están disponibles al público en general. Las estadísticas que se generan con estos datos, no están disponibles en el portal de transparencia del IPN.

La Universidad Autónoma Metropolitana (UAM) realiza desde el 2002 un *cuestionario-tipo* a sus aspirantes, con el fin de realizar estadísticas con los datos recopilados. Los resultados de tales estadísticas se encuentran disponibles en su página Web<sup>10</sup>. La UAM al igual que la UNAM realiza los perfiles de los aspirantes de ingreso, con la única diferencia de que la UAM solo brinda este servicio para el nivel de licenciatura.

La UNAM desde 1988 le aplica a sus aspirantes de ingreso una encuesta socioeconómica, con el fin de desarrollar los perfiles de aspirantes y mejorar los servicios que ofrece la Máxima Casa de Estudios. Esta encuesta se realizaba en papel impreso hasta el año 2005, posteriormente se creó un sistema Web para agilizar el análisis de los datos, así como para almacenar de una manera más eficiente la información, sin embargo dado el aumento en el número de aspirantes, los diferentes niveles escolares de ingreso que brinda la UNAM y el avance de las tecnologías de información, el sistema antecesor del SIUNE-DGAE actualmente no cubre las necesidades requeridas.

Encuestados	(2008)	(2009)	(2010)
UNAM	178,981	182,023	212,437
IPN	47,469	50,383	70,419
UAM	11,704	11,850	12,647

Cuadro 6.1: Cantidad de aspirantes que contestaron la encuesta socioeconómica en las diferentes universidades públicas en los tres últimos años.

En el cuadro 6.1 se puede observar la demanda que tiene el sistema de encuestas en la UNAM en comparación con el IPN y la UAM, lo cual denota una gran diferencia entre el número de usuarios de cada ciclo escolar, ya que el número de personas que contestaron la encuesta socioeconómica en la UNAM sobrepasa a la UAM y el IPN aproximadamente por más de cien mil.

<sup>9</sup>Sitio Oficial. Reglas de operación del programa nacional de becas y financiamiento. <http://web.seycc.gob.mx/pronabes/menu.asp> (15/062011 3:00 pm)

<sup>10</sup>Sitio Oficial. Acciones de Transparencia. <http://www.transparencia.uam.mx/> (16/062011 1:00 pm)

## Capítulo 7

# Diseño e implementación del sistema

En este capítulo se especifican los detalles de diseño del sistema, como lo son la plataforma de desarrollo y los componentes del sistema, así como la comunicación que existe entre éstos; para describir lo anterior se usan diagramas que muestran la arquitectura tanto del sistema como de la base de datos, también se incluye la especificación a detalle de las clases o módulos y métodos utilizados en la implementación del sistema.

Se incluye también el prototipo de la interfaz de usuario y para concluir el capítulo se muestran ejemplos del funcionamiento del sistema.

Debido a diversas causas, entre ellas que el SIUNE-DGAE es un sistema de información Web y que se realizó una migración de tecnologías en la DGAE, el sistema se desarrolló en dos versiones:

- La primera usando el lenguaje de programación C por ser un lenguaje robusto, muy conocido para desarrollar programas y sistemas de información, obteniendo así un sistema multiplataforma, portable, fácil de mantener y con un acceso a memoria más rápido.
- La segunda versión usando el lenguaje de programación Ruby, utilizando el marco de trabajo de Rails, un lenguaje flexible, multiplataforma, orientado a objetos ya que Rails es utilizado en el desarrollo de aplicaciones Web, además de contar con una gran variedad de bibliotecas (gemas) que facilitan, en gran medida, el proceso de desarrollo de sistemas de este tipo.

### 7.1. Diseño del algoritmo de codificación de respuestas.

Una de las partes más importantes del sistema de encuestas es la manipulación y el almacenamiento de las respuestas del usuario, este sistema pretende utilizar un número menor de registros en la base de datos, dado que el número de registros utilizados desde ciclos escolares anteriores por el sistema antecesor excedía de la capacidad de almacenamiento de los servidores de bases de datos de la DGAE, estos tenían que ser almacenados en discos, evitando esto se logra que la diferencia entre el presente sistema y su antecesor sea significativa; para lograr este objetivo se desarrolló el siguiente algoritmo, con el cual se tratan las respuestas de los usuarios para obtener una representación que utilice menos espacio en la base de datos, a continuación se presenta dicho algoritmo en pseudocódigo.

Código 7.1: Algoritmo para la codificación de las respuestas.

---

```

1  j=0
2  entero=1
3  caracter=''
4  contador=0
5  Desde i=0 hasta i=n //n es la posición máxima de las
   opciones
6   cadena[i]='0'
7 //Para formar la cadena con las respuestas
8 Mientras existan respuestas hacer{
9   opcion=pregunta.posicionInicial + respuesta.cve -1
10  cadena[opcion]='1'
11 }
12 Mientras j < cadena.longitud{
13  entero << 1
14  si cadena[j] == '1'
15   entero= entero | 1
16  si contador=7 ó j+1 = cadena.longitud{
17   caracter=char(entero)
18   concatena(cadena, caracter)
19   entero=1
20   contador=0
21  }
22 }
23 regresar cadena

```

---

En el Código 7.1 anterior se observa de las líneas 1-4 que se declaran e inicializan las variables necesarias para el algoritmo, de la línea 8 a la 10 se muestra como se forma una cadena con *0's* y *1's* según las respuestas del usuario y las posiciones correspondientes a cada pregunta<sup>1</sup>. Posteriormente esta cadena es transformada en enteros de las líneas 13 a 15, esto se realiza con operaciones como el corrimiento de bits, los cuales a su vez son transformados a un caracter, en la línea 17, de acuerdo a su representación en el código ASCII y posteriormente se añaden a una cadena.

Para comprender mejor el algoritmo anterior se muestra un ejemplo:

Primero se define los datos necesarios para el ejemplo, como son el número de preguntas y las respuestas del usuario, supóngase que se tienen cinco respuestas que provienen de las siguientes preguntas:

- Pregunta 1: con `posiciónIncial`<sup>2</sup>=0 y `posicionFinal`=2. Lo que significa que esa pregunta tiene tres opciones como respuesta, por su parte, en este ejemplo supongamos que el usuario seleccionó la opción 2.
- Pregunta 2: con `posiciónIncial`=3 y `posicionFinal`=4, en esta pregunta se tienen 2 opciones, de las cuales el usuario seleccionó la opción 1, lo cual significa que en la posición 3 de la cadena se colocará un '1' como se explica mas adelante.
- Pregunta 3: con `posiciónIncial`=5 y `posicionFinal`=8, por lo que esta pregunta tiene 4 opciones, de donde el usuario seleccionó la opción 4.
- Pregunta 4: con `posiciónIncial`=9 y `posicionFinal`=11, en esta pregunta hay 11 posibles opciones, de las cuales el usuario seleccionó la opción 3.
- Pregunta 5: con `posiciónIncial`=12 y `posicionFinal`=14, donde se tienen 3 opciones y el usuario seleccionó la opción 2.

A continuación se muestra como se obtiene la cadena de respuestas siguiendo las líneas 8-10 del algoritmo anterior, cabe mencionar que dicha cadena es de longitud  $n$  (suponiendo que solo hay 5 preguntas en la base de datos, por lo que la posición final máxima de las opciones es  $n=14$ ).

- Para la respuesta de la pregunta 1, de acuerdo a la línea 9 se tiene `opcion=0+2-1= 1` entonces se coloca `cadena[opcion]='1'`, obteniendo `cadena="01000000000000"`.

<sup>1</sup>La posición inicial, como la final de cada pregunta fue asignada de acuerdo al orden de la clave de la pregunta y el número de opciones de cada una de estas.

<sup>2</sup>Esto quiere decir que en la cadena de todas las respuestas del usuario, a la pregunta 'X' le corresponden las opciones que estan entre [`posicionIncial...` `posicionFinal`].

- Ahora para la respuesta de la pregunta 2, de acuerdo a la línea 9 se tiene  $\text{opcion}=3+1-1=3$ , entonces se coloca  $\text{cadena}[3]='1'$ , obteniendo  $\text{cadena}="0101000000000"$ .
- Tomando ahora para su analisis la pregunta 3, de acuerdo a la línea 9 se tiene  $\text{opcion}=5+4-1=8$ , de acuerdo a la línea 10 se coloca  $\text{cadena}[8]='1'$ , obteniendo  $\text{cadena}="01010000100000"$ .
- En la respuesta de la pregunta 4, de acuerdo a la línea 9 se tiene  $\text{opcion}=9+3-1=11$ , de acuerdo a la línea 10 se coloca  $\text{cadena}[11]='1'$ , obteniendo  $\text{cadena}="01010000100100"$ .
- En la pregunta 5, de acuerdo a la línea 9,  $\text{opcion}=12+2-1=13$ , verificando la instrucción de la línea 10 se coloca  $\text{cadena}[13]='1'$ , obteniendo  $\text{cadena}="01010000100101"$ .

Como resultado del proceso anterior se obtiene la  $\text{cadena}="01010000100101"$ . Ahora se procede a verificar cada caracter de ésta, de acuerdo a las líneas 12-22, de la siguiente manera:

1. Se realiza (línea 13)  $\text{entero} \ll 1$ , posteriormente se comprueba si  $\text{cadena}[0] = '1'$ , en este caso no es verdad ya que se tiene  $\text{cadena}[0]=0$  entonces, los bits del entero hasta este paso están representados como "10".
2. De acuerdo a la línea 13, se realiza  $\text{entero} \ll 1$ , después de esto se comprueba si  $\text{cadena}[1] = '1'$ , en la posición actual de la cadena se tiene que  $\text{cadena}[1]=1$  entonces se realiza  $\text{entero} \ll 1$  y  $\text{entero} | 1$ , por lo que ahora los bits del entero en este paso son "101".
3. Realizando lo indicado en la línea 13,  $\text{entero} \ll 1$ , después se comprueba si  $\text{cadena}[2] = '1'$ , en esta posición se tiene  $\text{cadena}[2]=0$  entonces se realiza  $\text{entero} \ll 1$  y los bits del entero son "1010".
4. Nuevamente se realiza (línea 13)  $\text{entero} \ll 1$ , posteriormente se comprueba si  $\text{cadena}[3] = '1'$ , se tiene que  $\text{cadena}[3]=1$  entonces se realiza  $\text{entero} \ll 1$  y  $\text{entero} | 1$ . Los bits del entero en este paso son "10101".
5. Según lo indicado en la línea 13,  $\text{entero} \ll 1$ , después se comprueba si  $\text{cadena}[4] = '1'$ , en esta posición se tiene  $\text{cadena}[4]=0$  entonces se realiza  $\text{entero} \ll 1$ . Los bits del entero hasta este paso son "101010".

6. Se realiza la instrucción de la línea 13 `entero <<1`, posteriormente se comprueba si `cadena[5] = '1'`, nuevamente se tiene `cadena[5]=0` entonces se realiza `entero <<1`. Los bits del `entero` en este paso son "1010100".
7. Siguiendo la línea 13, `entero <<1`, después se comprueba si `cadena[6] = '1'`, en esta posición se tiene `cadena[6]=0` entonces se realiza `entero <<1`. Los bits del `entero` en este paso son "10101000", los cuales representan al `entero 168`.
8. Ahora se tiene que el contador es igual a 7 entonces se realiza `caracter=char(entero)` y se concatena el caracter a una cadena que será regresada como salida de el algoritmo, por lo que se tiene `cadena="i"` y se reinicia tanto el `contador` como el `entero` 0 y 1 respectivamente.
9. Realizando lo indicado en la línea 13 `entero <<1`, posteriormente se comprueba si `cadena[7] = '1'`, en esta posición de la cadena se tiene el valor "0 " entonces se realiza `entero <<1`. Los bits del `entero` en este paso son "10".
10. De acuerdo a la línea 13, `entero <<1`, posteriormente se comprueba si `cadena[8] = '1'`, se tiene `cadena[8]=1` por lo que se realizan las operaciones `entero <<1` y `entero | 1`. Los bits del `entero` en este paso son "101".
11. En la línea 13 se indica que se debe realizar `entero <<1`, posteriormente se comprueba si `cadena[9] = '1'`, en esta posición se tiene `cadena[9]=0`, debido a esto solo se realiza la operación `entero <<1`. Los bits del `entero` en este paso son "1010".
12. De acuerdo a la línea 13 `entero <<1`, posteriormente se comprueba si `cadena[10] = '1'`, nuevamente se tiene `cadena[10]=0`, entonces se realiza `entero <<1`. Los bits del `entero` en este paso son "10100".
13. Ahora se realiza lo indicado en la línea 13 `entero <<1`, posteriormente se comprueba si `cadena[11] = '1'`, ahora se tiene `cadena[11]=1` por lo que se realiza `entero <<1` y `entero | 1`. Los bits del `entero` en este paso son "101001".
14. Realizando la instrucción de la línea 13 `entero <<1`, posteriormente se comprueba si `cadena[12] = '1'`, lo cual no se cumple en la posición actual de la cadena ya que `cadena[12]=0`, así que solo se realiza `entero <<1`. Los bits del `entero` en este paso son "1010010".

15. Siguiendo la línea 13 `entero <<1`, posteriormente se comprueba si `cadena[13] = '1'`, se puede observar que se cumple para esta posición que `cadena[13]=1` por lo que se realizan las operaciones `entero <<1` y `entero | 1`. Los bits del `entero` en este paso son "10100101"=165.
16. Ahora se tiene que `j+1=cadena.longitud`, por lo que se cumple la condición requerida en la línea 16, así que se realiza `caracter=char(entero)`, es decir, el `entero` es transformado en `caracter`, de lo cual se obtiene el carácter " $\tilde{N}$ ", por último es concatenada a la cadena final, obteniendo `cadena="¡ $\tilde{N}$ "`.

Como puede observarse en el procedimiento anterior, se codificaron las respuestas del usuario, quedando la cadena que representa a éstas como " $\tilde{N}$ ", es decir, se logró reducir la cadena original la cual era "01010000100101" y como puede observarse se reduce en gran medida el tamaño de ésta; esto se nota aún más en una cadena con mayor longitud, pero en el ejemplo dado en esta sección se obtiene una diferencia de 12 caracteres.

Lo cual reduce en gran medida el espacio requerido para almacenar las respuestas, ya que solo se necesitará un registro para almacenar las respuestas de las cinco preguntas de la encuesta, recordando que esto no se podía realizar en el sistema antecesor. Lo anterior se explica en la sección 7.3 del presente capítulo.

## 7.2. Diseño del algoritmo de decodificación de respuestas.

En la sección anterior se mostró el diseño del algoritmo de codificación de las respuestas, sin embargo este algoritmo no serviría de mucho si no se pueden recuperar estos datos, por tal motivo es necesario un algoritmo de decodificación, con el que se puedan recuperar los datos originales.

De igual manera que en la sección anterior, se realiza un ejemplo con la cadena resultante en la codificación de las respuestas.

El algoritmo de decodificación de respuestas es similar al algoritmo de codificación, ya que en éste se trata de realizar las operaciones inversas al de codificación, para ilustrar mejor esta idea se muestra a continuación el algoritmo en pseudocódigo<sup>3</sup>.

Del código 7.2 se puede observar en las líneas 1-2 que se declaran las variables necesarias, en la línea 3 se tiene la llamada a un método denominado

---

<sup>3</sup>El método `respuestasUsuario()` regresa la cadena de respuestas del usuario almacenada en la base de datos.

Código 7.2: Algoritmo para la decodificación de las respuestas.

---

```

1  j=0
2  caracter=''
3  cadena=
4  cadenaCodificada= respuestasUsuario()
5  //Para formar la cadena con las respuestas
6  Mientras j< cadenaCodificada.longitud{
7      cadenaAux=
8      entero=int(cadena[j])
9      Mientras entero > 1{
10         caracter= entero & 1
11         entero= entero >> 1
12         concatena(cadenaAux, caracter)
13     }
14         invierte(cadenaAux)
15         concatena(cadena, cadenaAux)
16     }
17 regresar cadena

```

---

“*respuestasUsuario*” que regresa en una cadena las respuestas codificadas de algún usuario, posteriormente en la línea 6 se convierte cada caracter de la cadena en entero, por último a cada entero se le realizan las operaciones correspondientes a las líneas 8-12 (se convierte el caracter a entero y se realiza la operación binaria “and” sobre el entero), estas líneas se utilizan para saber si se debe agregar a la cadena de respuestas un ‘1’ o un ‘0’, esto se puede observar con mayor claridad en el siguiente ejemplo.

La cadena obtenida en el ejemplo de codificación de las respuestas fue “¿Ñ”, ahora siguiendo el algoritmo de decodificación se obtiene:

1. En la línea 6 se tiene que  $j=0$  y  $\text{cadena}[0]=\text{“¿”}$ , siguiendo la línea 8 se tiene  $\text{entero}=\text{int}(\text{“¿”})=168$ .
2. Ahora siguiendo la línea 10, se realiza  $\text{caracter} = \text{entero} \& 1=0$  y se recorre un bit en el número que representa al caracter de la siguiente forma  $\text{entero} = \text{entero} \gg 1$ , por lo que ahora se tiene el número  $\text{entero}=84$ , y se le agrega el caracter a la cadena, quedando de la siguiente forma:  $\text{cadenaAux}=\text{“0”}$ .
3. Nuevamente se sigue la línea 10 del código, se realiza  $\text{caracter} = \text{entero} \& 1=0$ , se recorre un bit en el número que representa al caracter de la

siguiente forma `entero = entero >> 1`, por lo que ahora se tiene el número `entero=42`, con la cadena hasta este momento `cadenaAux="00"`.

4. Ahora según lo indicado en la línea 10, se realiza la operación `caracter = entero & 1=0`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1`, como resultado de esto se obtiene `entero=21`, con la cadena hasta este momento `cadenaAux="000"`.
5. Siguiendo la línea 10, se realiza la operación `caracter = entero & 1=1`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1` obteniendo `entero=10`, con la cadena hasta este momento `cadenaAux="0001"`.
6. De acuerdo a la línea 10, se realiza la operación `caracter = entero & 1=0`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1` obteniendo `entero=5`, con la cadena hasta este momento `cadenaAux="00010"`.
7. Ejecutando la línea 10, se realiza la operación `caracter = entero & 1=1`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1` obteniendo `entero=2`, con la cadena hasta este momento `cadenaAux="000101"`.
8. Realizando lo indicado en la línea 10, se realiza la operación `caracter = entero & 1=0`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1` obteniendo `entero=1`, con la cadena hasta este momento `cadenaAux="0001010"`.
9. Dado que no se cumple que `entero > 1`, se verifica el siguiente caracter de la cadena con `j=1`, el cual se tiene `cadena[1]="Ñ"`, este caracter se convierte a número obteniendo `entero=165`, posteriormente se realiza lo indicado en la línea 14 obteniendo `cadenaAux="0101000"` y agregando esta a la cadena que será devuelta por el algoritmo.
10. Siguiendo la línea 10, se realiza la operación `caracter = entero & 1=1`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1`, dando como resultado `entero=82`, ahora se agrega el caracter a la cadena decodificada obteniendo: `cadenaAux="1"`.
11. De acuerdo a la línea 10, se realiza la operación `caracter = entero & 1=0`, se recorre un bit en el número que representa al caracter de la siguiente

forma: `entero = entero >> 1`, dando como resultado `entero = 41`, se agrega el caracter a la cadena decodificada obteniendo: `cadenaAux = "10"`.

12. Según la instrucción de la línea 10, se realiza la operación `caracter = entero & 1 = 1`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1`, dando como resultado `entero = 20`, se agrega el caracter a la cadena decodificada obteniendo: `cadenaAux = "101"`.
13. Ahora la línea 10 indica que se realiza la operación `caracter = entero & 1 = 0`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1`, dando como resultado `entero = 10`, se agrega el caracter a la cadena decodificada obteniendo: `cadenaAux = "1010"`.
14. En la línea 10, se realiza la operación `caracter = entero & 1 = 0`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1`, dando como resultado `entero = 5`, se agrega el caracter a la cadena decodificada obteniendo: `cadenaAux = "10100"`.
15. Posteriormente en la línea 10, se realiza la operación `caracter = entero & 1 = 1`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1`, dando como resultado `entero = 2`, se agrega el caracter a la cadena decodificada obteniendo: `cadenaAux = "101001"`.
16. Nuevamente en la línea 10, se realiza `caracter = entero & 1 = 0`, se recorre un bit en el número que representa al caracter de la siguiente forma: `entero = entero >> 1`, dando como resultado `entero = 1`, se agrega el caracter a la cadena decodificada obteniendo: `cadenaAux = "1010010"`.
17. Nuevamente se llega a `entero = 1`, por lo que no se cumple la condición de la línea 9, así que se ejecuta la línea 14 quedando `cadenaAux = "0100101"`, por último se concatena ésta a la cadena que se tenía anteriormente obteniendo `cadena = "01010000100101"`.

Al terminar este ejemplo se obtiene la cadena "01010000100101", la cual es la misma que se había generado con las respuestas del usuario antes de codificarla, por lo tanto se puede observar que los datos originales de las respuestas se recuperaron con el algoritmo de decodificación de respuestas.

### 7.3. Diseño de la base de datos SIUNE-DGAE.

El almacenamiento de los datos que se proporcionan en las encuestas del sistema se realiza en los servidores de bases de datos de la UNAM que se encuentran en la dependencia Dirección General de Administración Escolar (DGAE), las respuestas de las encuestas son los únicos datos que se almacenan cuando el sistema es ejecutado, ya que los datos necesarios para crear las encuestas ya se encuentran almacenados en la base de datos, éstos se guardan en servidores SUN Fire V880, con las siguientes características:

- 1050 Mhz de velocidad.
- 12 DD de 72 GB.
- 6 DD de 36 GB.
- 10 DD de 415 GB.
- 16 GB de memoria RAM.
- 8 procesadores UltraSPARC-III.

#### 7.3.1. Entidades y relaciones.

La base de datos utilizada para el sistema de encuestas antecesor al sistema presente, consta de 7 tablas o entidades, con sus respectivas relaciones entre ellas, éstas se muestran en la figura a continuación<sup>4</sup>.

---

<sup>4</sup>Las llaves o claves primarias de cada tabla están identificadas como “<PK>atributo”, que por sus siglas en inglés significa Primary Key.

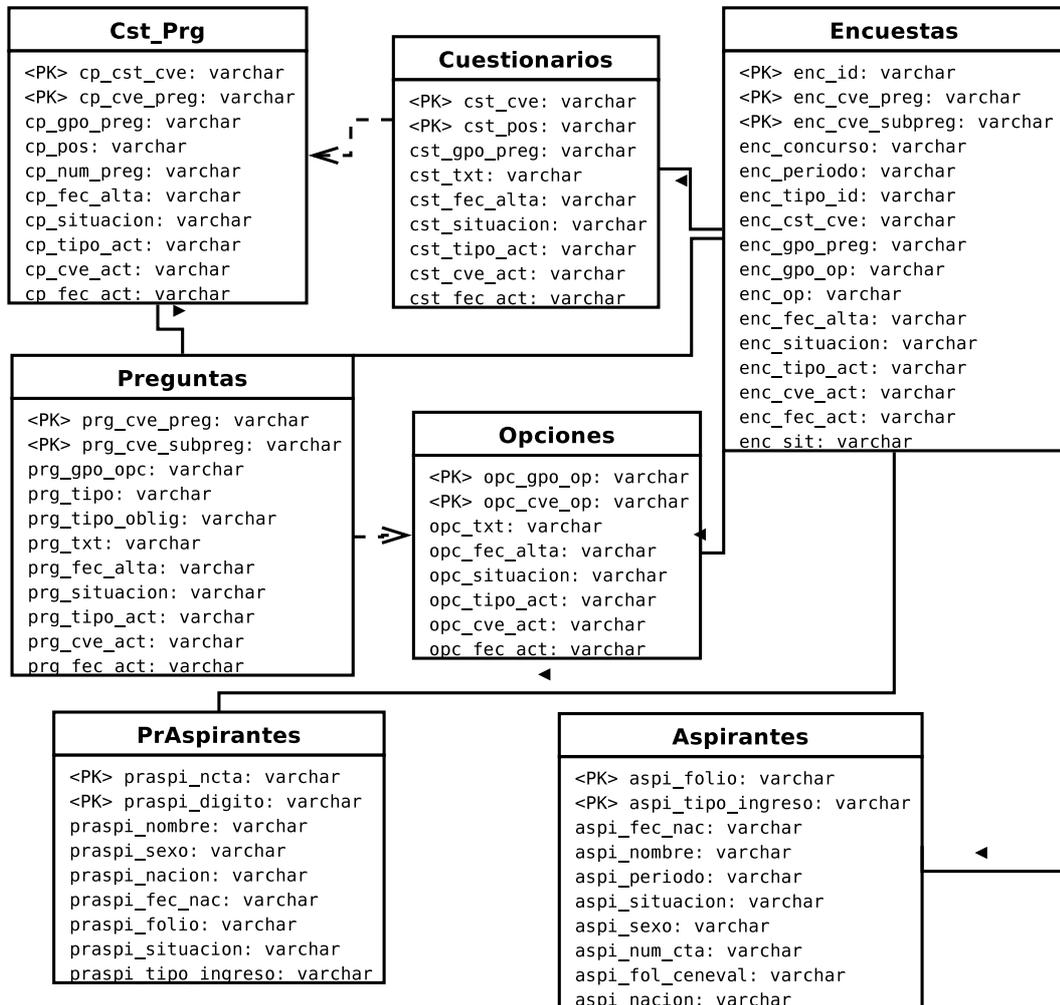


Figura 7.1: Diseño de la base de datos del sistema predecesor.

La tabla de **Aspirantes**, mostrada en la figura 7.1, contiene más de 150 atributos, ya que contiene los datos personales del aspirantes como lo son el domicilio, la CURP, las opciones de ingreso (carreras de preferencia, campus de preferencia y los sistemas de las carreras<sup>5</sup>), entre otras, pero para fines del sistema SIUNE-DGAE estos atributos no son utilizados, por lo que se omiten en el diagrama de la base de datos mostrada en la Figura 7.1.

La tabla **Cuestionarios** contiene la definición de las encuestas (clave

<sup>5</sup>Los posibles sistemas de las carreras son: escolarizado, abierto, a distancia.

de cada encuesta, texto con el objetivo de la encuesta, entre otras), con sus respectivas secciones (título de la sección, clave de la sección), las cuales están definidas en la tabla **Cst\_Prg**, en la que también se definen las preguntas que contiene cada sección.

En la tabla **Preguntas** se definen las claves de las preguntas, las claves de las subpreguntas, los textos correspondientes a cada una de éstas y la clave del grupo de opciones correspondiente.

En la tabla **Opciones** de acuerdo a cada grupo de opciones (**op\_gpo\_opc**), se definen las claves y los textos de las posibles opciones de respuesta de cada pregunta.

En la tabla **Aspirantes** se encuentran los datos necesarios de cada aspirante de ingreso la UNAM en los niveles licenciatura y bachillerato. Por otro lado, en la tabla **PrAspirantes** se encuentran los datos personales de los alumnos de bachillerato que liberan pase reglamentado.

Por último, en la tabla de **Encuestas** se almacenan las respuestas de los usuarios utilizando como clave primaria el identificador del usuario, la clave de la pregunta y la llave de la subpregunta de la encuesta.

En una primera versión del sistema de encuestas optimizado el diseño de la base de datos se modificó de tal manera que la única tabla afectada fue la de **Encuestas**, quedando esta tabla de la siguiente manera:

Encuestas	
<PK>	enc_cst_cve
<PK>	enc_id
	enc_período
	enc_concurso
	enc_tipo_id
	enc_resp
	enc_fec_alta
	enc_situacion
	enc_tipo_act
	enc_cve_act
	enc_fec_act
	enc_situacion

Figura 7.2: Primer modificación en la tabla Encuestas.

Posteriormente, de acuerdo a las especificaciones del sistema se realizó un nuevo diseño, en este diseño se modificó únicamente dos tablas (**Preguntas** y **Encuestas**) con el propósito de alterar en lo menor posible el diseño del sistema antecesor. En la Figura 7.3 se muestra el diseño de la base de datos del actual sistema SIUNE-DGAE.

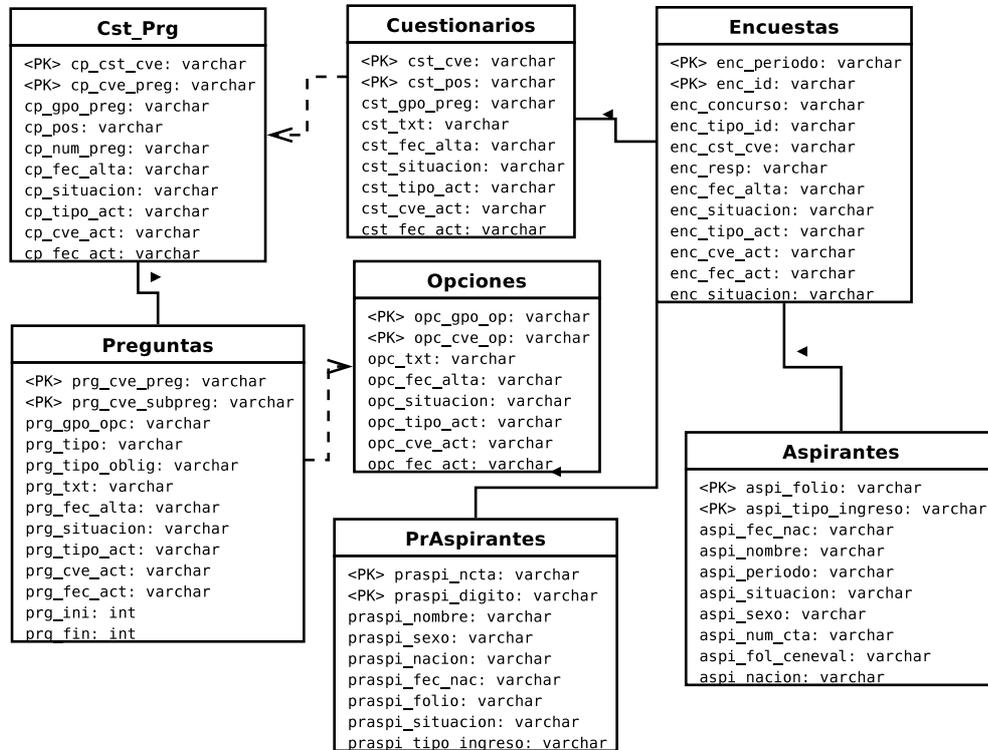


Figura 7.3: Nuevo diseño de la base de datos.

A la tabla de **Preguntas** se le añadió los atributos **prg\_ini** y **prg\_fin**, que indican las posiciones inicial y final de las posibles respuestas de cada pregunta.

La tabla de **Encuestas** fue modificada cambiando los siguientes atributos: **enc\_cve\_preg**, **enc\_cve\_subpreg**, **enc\_gpo\_op** y **enc\_op** por el atributo **enc\_resp**, ya que en el presente sistema de encuestas (SIUNE-DGAE) se modificó la forma de almacenar las respuestas de tal manera que se unificaron las respuestas de todas las encuestas por usuario<sup>6</sup>.

<sup>6</sup>Se explica más a detalle el cambio en la manipulación de las respuestas en la sección 7.1

## 7.4. Diseño del sistema SIUNE-DGAE desarrollado en C

El diseño del sistema desarrollado en C, consiste en mantener la arquitectura del sistema predecesor, modificando únicamente la manera de almacenar las respuestas de las encuestas en la base de datos y añadiendo a todas las encuestas el módulo de *Equidad y Género* como segunda parte de las demás encuestas (ya mencionado con anterioridad en la sección 6.1.1).

El diseño del sistema se presenta en la Figura 7.4, debido a que éste es un lenguaje de programación estructurado, no existe el concepto de clase como tal, por lo que el diagrama de clases del sistema está representado por los módulos del sistema (`funciones_bit.h`, `aux.c`, `sybase.c`, `aux3.c`, `cabecera.h`, `encuesta_g.c`, `enc.h` y `enc.c`), mostrados en la Figura 7.4.

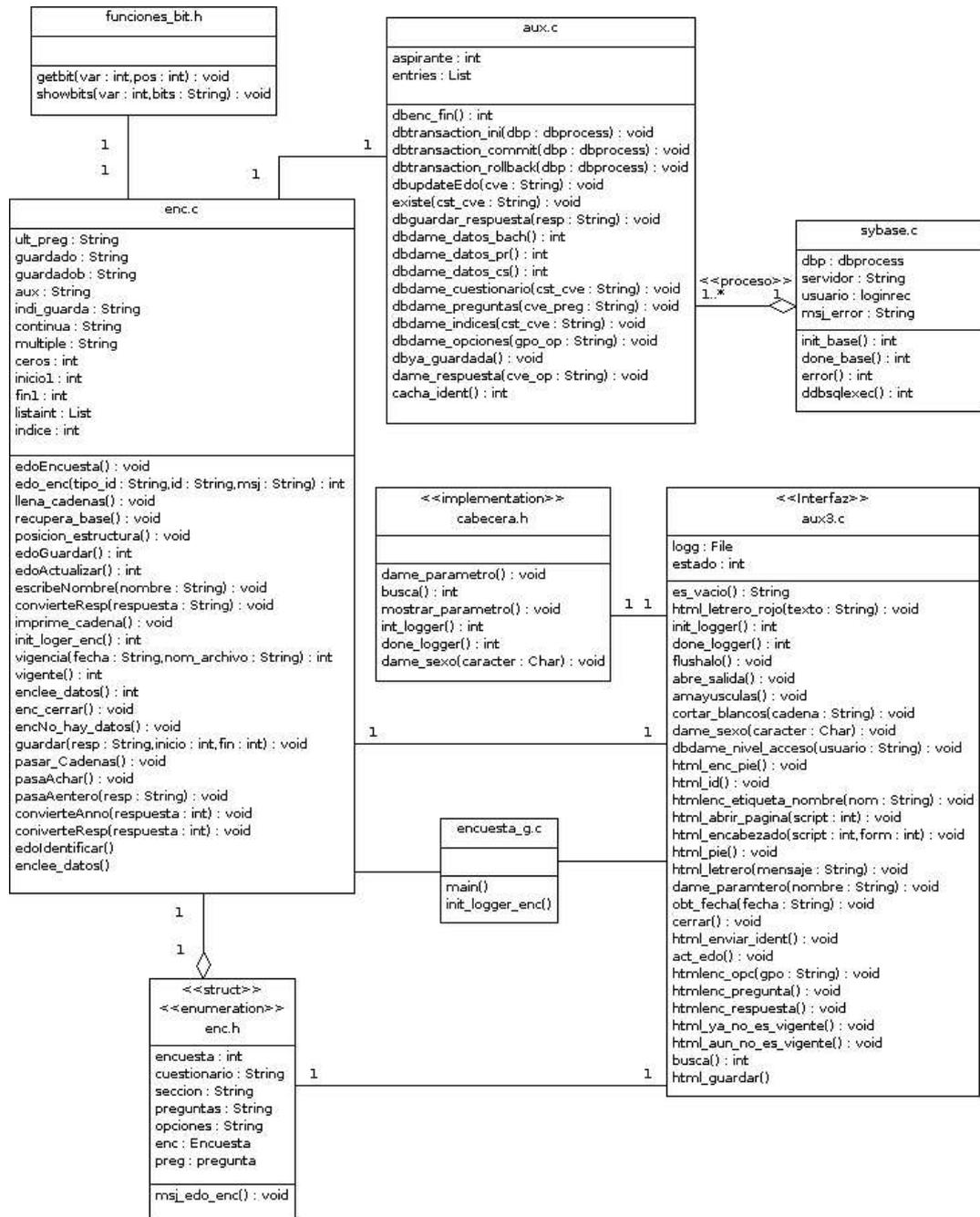


Figura 7.4: Diagrama de clases del sistema desarrollado en C.

#### 7.4.0.1. Arquitectura del sistema.

La arquitectura del sistema de esta versión está implementado como un CGI<sup>7</sup>, el cual cumple con las características del patrón de arquitectura de software *cliente-servidor*, ya que para la comunicación utiliza dos sockets<sup>8</sup>, un *socket servidor* y un *socket cliente*. Cuando un usuario entra al sistema usando el Internet se realiza una petición (socket cliente) al sistema, el cual responde de acuerdo a los datos proporcionados (socket servidor).

En la Figura 7.5 se muestra la arquitectura del sistema SIUNE-DGAE desarrollado en C:

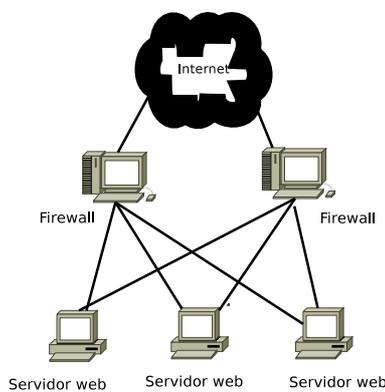


Figura 7.5: Arquitectura cliente-servidor

Los equipos que se usan como servidores web (donde se localiza el socket cliente) son equipos SUNFire V240, con las siguientes características:

- 2 GB de memoria RAM.
- 2 DD de 72 GB.
- 2 procesadores UltraSPARC-III.
- 1500 Mhz de velocidad.

<sup>7</sup>Interfaz de entrada común, por sus siglas en inglés Common Gateway Interfce, explicado en la sección 4.3.1.

<sup>8</sup>Mecanismo para la entrega de paquetes de datos provenientes de diferentes computadoras, posiblemente ubicadas en distintos lugares geográficos. Se definen con dos direcciones IPs, un protocolo de transporte y dos números de puerto, uno local y el otro remoto. <http://es.wikipedia.org/wiki/Socket> (16/07/2011 6:36 pm).

Los equipos donde se desarrolló el sistema y donde se localizan los sockets servidores tiene las mismas características que los equipos descritos en la sección 7.3.

### **7.5. Diseño del sistema SIUNE-DGAE desarrollado en Ruby.**

El diseño del sistema SIUNE-DGAE desarrollado con el lenguaje de programación Ruby esta basado en la estructura que por omisión genera el marco de trabajo Rails, debido a que Ruby es un lenguaje de programación orientado a objetos, en éste sí existe el concepto de clases. El diseño pretende que las clases sean lo más sencillas posibles, aunque esto implique aumentar el número de clases ya que entre más sencillas sean éstas, más fáciles de comprender serán para el desarrollador, lo que a su vez facilita el mantenimiento.

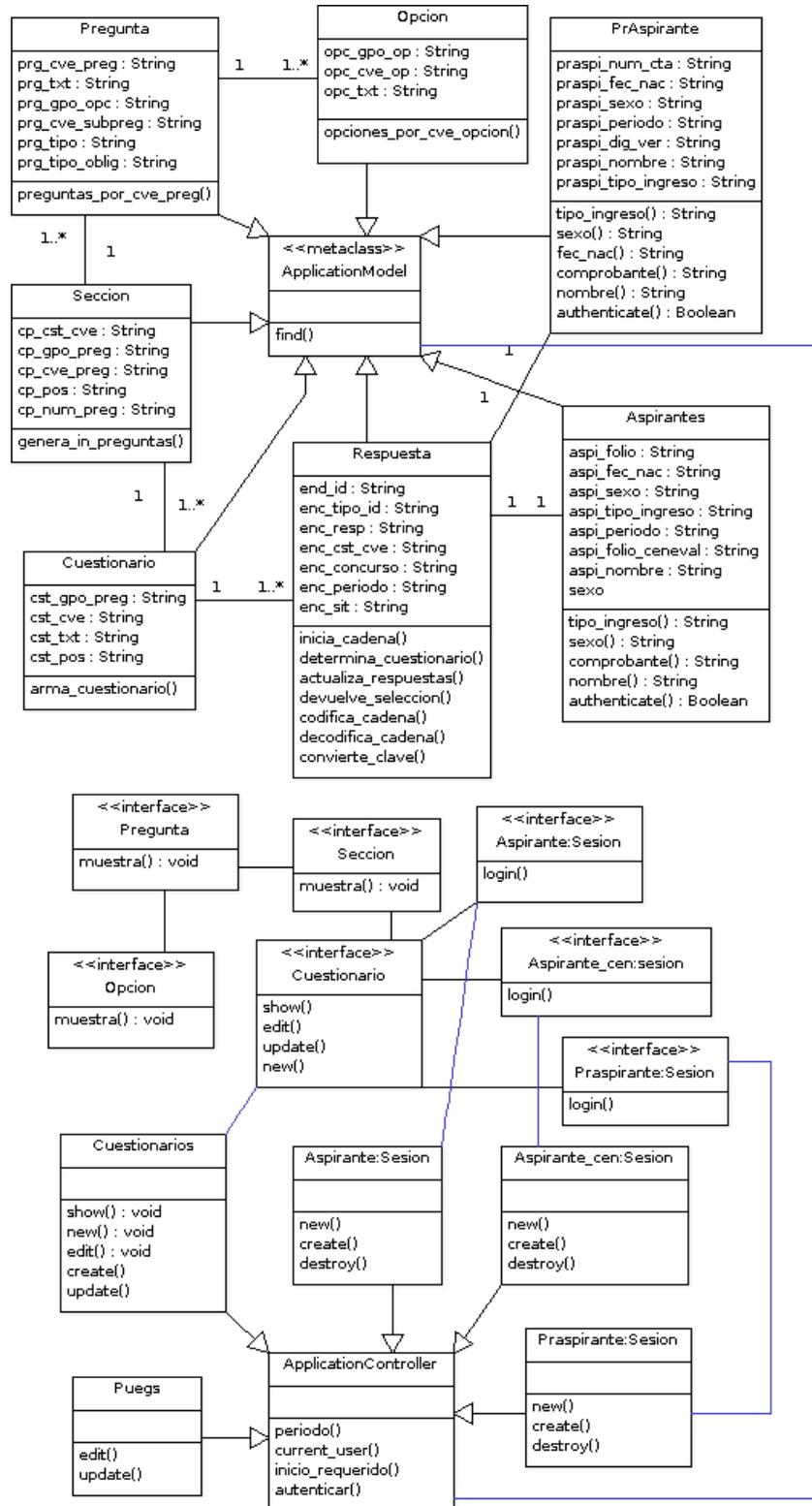


Figura 7.6: Diagrama de clases del sistema desarrollado en Ruby on Rails.

En la Figura 7.6 se muestran las clases `Pregunta`, `Opcion`, `PrAspirante`, `Seccio`, `Cuestionario`, `Respuesta`, `ApplicationModel`, `Aspirantes`, las interfaces `Pregunta`, `Seccion`, `Aspirante.Session`, `Cuestionario`, `Opcion`, `Aspirante_cen.Session`, `PrAspirante.Session` y las clases que corresponden a la parte del controlador `Cuestionario`, `Aspirante.Session`, `Puegs`, `Aspirante_cen.Session`, `PrAspirante.Session` y `ApplicationController`, todas estas clases componen el sistema desarrollado en Ruby y más adelante se muestra la clasificación que cada una de ellas tiene.

### 7.5.1. Arquitectura del sistema.

La arquitectura de este sistema cumple con las características del patrón *Modelo-Vista-Controlador* (MVC), ya que es la estructura de organización de sistemas desarrollados con Ruby on Rails. Al utilizar este marco de trabajo, cuando se crea el proyecto también se generan los paquetes del modelo, de los controladores y el paquete de vistas, este último con los archivos correspondientes a cada acción de los controladores.

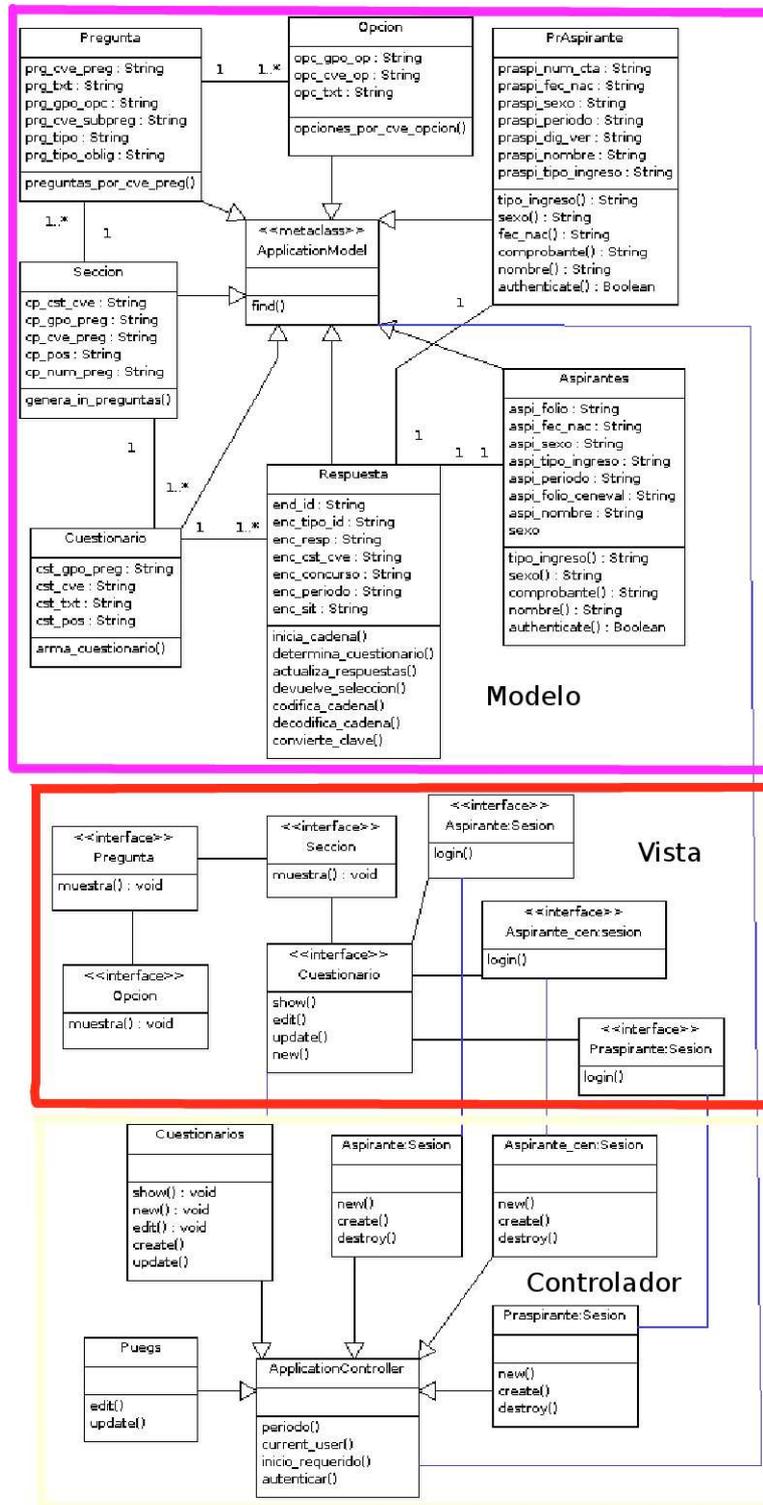


Figura 7.7: Diagrama MVC del sistema SIUNE-DGAE.

En la Figura 7.7 se muestra la estructura del sistema dividida en clases y para mostrar los componentes del patrón MVC se dibujaron cuadrados, que se interpretan de la siguiente manera:

- Cuadrado 1 color **lila** representa la parte del modelo del sistema, que contiene la clase `AppModel`, de la cual heredan las clases `Cuestionario`, `Sección`, `Pregunta`, `Opcion`, `Respuesta`, `Aspirante` y `Praspirante`, en estas clases están los métodos para obtener los atributos de cada clase.
- Cuadrado 2 color **amarillo** representa la parte del controlador, este cuadro contiene la clase `AppControl`, de ésta heredan `Cuestionarios`, `Aspirantes.Sesion`, `Aspirante_cen.Sesion`, `Puegs` y `Praspirantes.Sesion`, en estas clases se realizan las modificaciones a los datos del modelo que se representaran en la vista.
- Cuadrado 3 color **rojo** representa la parte de la vista del sistema, contiene los archivos que serán mostrados a los usuarios, entre ellas se encuentran `Cuestionario`, `Sección`, `Pregunta`, `Opcion`, `Aspirante.Sesion`, `Aspirante_cen.Sesion` y `Praspirante.Sesion`.
- Las líneas de color **azul** representan la relación que existe entre las clases de los diferentes componentes de la arquitectura del sistema.

Aunque en la estructura del sistema desarrollado en C se pretende separar los módulos que realizan operaciones en la base de datos, tanto de los módulos que imprimen en pantalla la interfaz del usuario como de los componentes del sistema que realizan las operaciones de control, no se podría asegurar que el sistema tuviera una arquitectura MVC, ya que los módulos no están completamente definidos y divididos por funciones, esto debido a que en los componentes del sistema SIUNE-DGAE desarrollado en C puede encontrarse tanto acciones que son parte del controlador como acciones que son parte del modelo del sistema.

### 7.5.2. Diseño de la Interfaz de usuario.

Desde el año pasado (2010) se pretende que las páginas que proporcionan información general de los aspirantes y alumnos debieran tener una interfaz de usuario lo más semejante posible, para lograr tener un diseño institucional. Por esta razón el sistema SIUNE-DGAE toma como referencia el diseño de la página principal de la DGAE, que a continuación en la Figura 7.8 se muestra:



Figura 7.8: Menú página principal DGAE.

En la figura anterior se muestran algunas opciones de ingreso al presente sistema de encuestas, éstas se encuentran divididas por concurso de ingreso a la UNAM, por sistema y por modalidad, ya sea “Escolarizado”, “Abierto” ó “A distancia”.<sup>9</sup>

Una vez que el usuario selecciona la encuesta que contestará se le muestra el formulario con los campos necesarios para iniciar la sesión.

<sup>9</sup>En la página principal de la DGAE, solo están disponibles tres módulos del sistema, que son “Concurso de selección a nivel licenciatura”, “Pase reglamentado” y “Concurso de selección a nivel bachillerato”, los módulos restantes aún no están disponibles para el público en general.

UNAM Universidad Nacional Autónoma de México

**IDENTIFICACION**  
**PARA ACCESAR TU HOJA DE DATOS ESTADISTICOS**

Número de Comprobante:   
(en tu bolsa credencial)

Fecha de Nacimiento:  Día (2) Mes (2) Año (4)  
Ejemplo: 10/01/1978

Por favor verifica tus datos antes de enviarlos.

Enviar Limpieza

Principal Regresar

Figura 7.9: Inicio de sesión, módulo “Concurso de selección a nivel Licenciatura”.

Debido a que en el concurso de selección a nivel licenciatura no todos los aspirantes tienen número de cuenta de la UNAM, el dato solicitado para ingresar al sistema es el folio que se les proporciona al inscribirse al concurso, por otro lado para el acceso a la encuesta de Pase reglamentado, Opinión de servicios de licenciatura y bachillerato, así como la encuesta de Egresados y la de Becas se aplican a alumnos de la UNAM, por lo que ya tienen asignado un número de cuenta, el cual se les solicita para ingresar al sistema, lo cual se puede observar en la Figura 7.9 y 7.10 respectivamente.



Universidad Nacional Autónoma de México

IDENTIFICACION  
PARA ACCESAR TU HOJA DE DATOS ESTADISTICOS

No. Cuenta:  -   
Si tu número de cuenta es de 8 dígitos, antepón un 0 (CERO)

Fecha de Nacimiento:  Día (2) Mes (2) y Año (4)  
Ejemplo: 15/01/1978

Por favor verifica tus datos antes de enviarlos.

[Principal](#) [Regresar](#)

Figura 7.10: Inicio de sesión, módulo “Pase Reglamentado”.

En los diferentes tipos de inicio de sesión se les pide además de su identificador, la fecha de nacimiento, como se observa en las Figuras 7.9, 7.10 y 7.11.

El último tipo de inicio de sesión del sistema se trata del módulo de “*Concurso de selección a nivel bachillerato*”, los usuarios de este módulo no tienen número de cuenta, por esta razón se solicita el folio brindado por la COMIPEMS a cada usuario de este módulo, sin embargo debido a que la encuesta puede ser contestada aún después de haber ingresado a la UNAM, así el usuario podría contar en ese momento con un número de cuenta y probablemente no recordará el folio que la COMIPEMS le asignó, por lo que este módulo también tiene la opción de ingresar con el número de cuenta, como se muestra en la Figura 7.11.

UNAM Universidad Nacional Autónoma de México

**IDENTIFICACION**  
**PARA ACCESAR TU HOJA DE DATOS ESTADISTICOS**

Proporciona sólo alguno de los siguientes dos datos  
(no olvides marcar tu selección):

Folio COMIPEMS:

Número de cuenta UNAM:  -

y también

Fecha de Nacimiento:  Día (2), Mes (2) y Año (4). Ejemplo: 10/11/1989.

Por favor verifica tus datos antes de enviarlos.

[Principal](#) [Regresar](#)

Figura 7.11: Inicio de sesión, módulo “Concurso de selección a nivel Bachillerato”.

Una vez que el sistema haya realizado las verificaciones necesarias se le muestra al usuario la encuesta solicitada, con el objetivo del módulo solicitado, los títulos de cada sección y las preguntas correspondientes de cada una de éstas.

Para mostrar las preguntas, se verificará si el usuario ya ha contestado preguntas del módulo con anterioridad, de ser así se mostrarán sus respuestas anteriores en forma de texto, como se observa en las Figuras 7.12, 7.13 y 7.14, en caso de que no se han respondido las preguntas, se mostrará una lista con las posibles opciones de respuesta<sup>10</sup> correspondientes a cada una de éstas.

<sup>10</sup>Para lograr esto se hace uso del tag `<select>` de HTML el cual muestra una lista de opciones.



**Universidad Nacional  
Autónoma de México**



---

**DATOS ESTADÍSTICOS DEL CONCURSO DE SELECCIÓN A LA LICENCIATURA**

**I. Esta encuesta tiene por objetivo conocer las características de la población que ingresará a la licenciatura por medio del concurso de selección. La información que nos proporciona es confidencial y sólo con fines estadísticos.**

**Es fundamental que llenes la encuesta con información verídica y de forma completa, ya que los datos recogidos a través de ella nos permitirán tomar decisiones que redundarán en el mejoramiento de los diversos programas de apoyo académico y económico (becas).**

Muchas gracias por tu cooperación.

---

**DATOS GENERALES**

AP. PATERNO AP. MATERNO NOMBRE(S)  
 NOMBRE

1. NÚMERO DE COMPROBANTE:

2. FECHA DE NACIMIENTO:

3. GÉNERO: MASCULINO

4. ESTADO CIVIL : Soltero

5. ¿TIENES HIJOS? : No

6. TIPO DE ESCUELA A LA QUE ASISTISTE  
 Primaria : Sólo privada  
 Secundaria : Sólo privada  
 Bachillerato : Sólo privada

7. EL BACHILLERATO LO TERMINASTE EN : El Edo. de México

**DATOS DE ESTUDIOS INMEDIATOS ANTERIORES**

8. ESCUELA DE PROCEDENCIA : Incorporada a la UNAM

9. ¿EN QUÉ AÑO INICIASTE EL BACHILLERATO? : 2008

10. ¿EN QUÉ AÑO TERMINASTE COMPLETO EL BACHILLERATO? : 2011

11. ¿REALIZASTE EL BACHILLERATO EN TRES AÑOS? : Si

12. TURNO EN QUE HICISTE EL BACHILLERATO : Matutino

13. PROMEDIO DE CALIFICACIONES EN EL BACHILLERATO : De 7.0 a 7.5

14. ¿CUÁNTOS EXÁMENES EXTRAORDINARIOS PRESENTASTE? : Ninguno

15. ¿CUÁNTAS MATERIAS VOLVISTE A CURSAR? : Ninguna

16. LA PREPARACIÓN QUE RECIBISTE FUE ... : Excelente

**DATOS PERSONALES**

17. ¿CUÁNTOS HERMANOS TIENES SIN CONSIDERAR MEDIOS HERMANOS? : 1

18. ¿CUAL ES EL LUGAR QUE OCUPAS ENTRE TUS HERMANOS? : 1o

19. ¿QUÉ TANTO INSISTEN TUS PADRES PARA QUE SIGAS ESTUDIANDO? : Mucho

20. CALIFICA EL ÉXITO QUE HAS TENIDO COMO ESTUDIANTE : Bueno

21. ¿QUÉ MATERIAL DE CONSULTA USAS EN TU CASA?  
 (Puedes marcar más de una opción pulsando la tecla control y las respuestas que desees) : Libros de texto

Figura 7.12: Primera parte del cuestionario de concurso de selección a nivel licenciatura, con respuestas de un usuario.

En la Figura 7.12 se puede observar la primera parte del módulo CSL-08, mostrando las respuestas del usuario, previamente almacenadas<sup>11</sup>.

<sup>11</sup> Por motivos de confidencialidad no se muestran en la Figura 7.12 el nombre, el número de comprobante y la fecha de nacimiento del aspirante.

21. ¿QUE MATERIAL DE CONSULTA USAS EN TU CASA?  
(Puedes marcar más de una opción puliendo la tecla control y las respuestas que deseas): Libros de texto

22. AL ESTUDIAR ACOSTUMBRAS ...  
(Elige una respuesta para cada renglón)

Primero leer todo el tema : Siempre (+ del 90% de las veces)  
 Subrayar las ideas principales : Siempre (+ del 90% de las veces)  
 Elaborar resúmenes del material : Siempre (+ del 90% de las veces)  
 Hacer síntesis tipo "acordeón" : Siempre (+ del 90% de las veces)  
 Resolver ejercicios para reafirmar el tema : Siempre (+ del 90% de las veces)

23. ESTUDIAS ...  
(Elige una respuesta para cada renglón)

Solo : Siempre (+ del 90% de las veces)  
 En equipo : Siempre (+ del 90% de las veces)  
 En la casa : Siempre (+ del 90% de las veces)  
 En la biblioteca : Siempre (+ del 90% de las veces)  
 En el parque : Siempre (+ del 90% de las veces)  
 Escuchando radio o televisión : Siempre (+ del 90% de las veces)  
 Frecuentemente (Del 70% al 90%  
 de las veces)  
 En la cama : Siempre (+ del 90% de las veces)  
 En escritorio o mesa : Siempre (+ del 90% de las veces)  
 En el transporte colectivo : Siempre (+ del 90% de las veces)  
 Sin distracción : Siempre (+ del 90% de las veces)

**SITUACIÓN SOCIOECONÓMICA**

24. ¿Hablas alguna lengua indígena? : Si

25. ¿Tus padres hablan alguna lengua indígena? : Si

26. ¿Te consideras miembro de algún pueblo indígena? : Si

27. SI UNA O VARIAS DE LAS TRES PREGUNTAS ANTERIORES FUERON AFIRMATIVAS,  
 ESPECIFICA CUÁL LENGUA O PUEBLO INDÍGENA

Hablas la lengua : Aguatécatco  
 Tus padres hablan la lengua : Chatino (Ch'ol'ña)  
 Eres miembro de un pueblo que habla la lengua : Aguatécatco

28. ¿CUÁL ES EL MÁXIMO NIVEL DE ESTUDIOS DE TUS PADRES Y HERMANOS MAYORES DE  
 12 AÑOS?  
 (AÚN CUANDO YA HAYAN FALLECIDO)

Madre : Sin instrucción  
 Padre : Sin instrucción  
 Hermano A : Sin instrucción  
 Hermano B : Sin instrucción  
 Hermano C : Sin instrucción  
 Hermano D : Sin instrucción  
 Hermano E : Sin instrucción

29. DE LAS PERSONAS QUE VIVEN EN TU CASA, ¿CUÁNTAS TRABAJAN O HACEN ALGUNA  
 ACTIVIDAD POR LA CUAL GANEN DINERO? : 1

30. ¿CUÁL ES LA PRINCIPAL OCUPACIÓN REMUNERADA QUE TIENEN TUS PADRES  
 ACTUALMENTE?

Figura 7.13: Continuación del cuestionario de concurso de selección a nivel licenciatura, con respuestas de un usuario.

En la Figura 7.13 se muestra la segunda parte o continuación del módulo CSL-08.

30 . ¿CUÁL ES LA PRINCIPAL OCUPACIÓN REMUNERADA QUE TIENEN TUS PADRES ACTUALMENTE?  
 Madre : No trabaja actualmente  
 Padre : No trabaja actualmente

31 . ¿QUIÉN SOSTIENE TUS ESTUDIOS? : Madre

32 . ¿CUÁNTAS PERSONAS DEPENDEN ECONÓMICAMENTE DE QUIEN SOSTIENE TUS ESTUDIOS?  
 (inclúyete tú y a tu sostén económico) : 1

33 . LA CASA DONDE HABITAS ACTUALMENTE ES : Propia

34 . CUÁNTOS CUARTOS HAY EN LA CASA DONDE VIVES?  
 (No cuentes cocina y baños) : 1

35 . ¿CON QUIÉN VIVES ACTUALMENTE? : Padre y/o madre y/o hermanos

36 . ¿CUÁNTAS PERSONAS HABITAN DONDE TÚ VIVES?  
 (inclúyete tú) : 1

37 . ¿DE QUÉ BIENES Y SERVICIOS DISFRUTAS EN TU CASA?  
 Cuarto de baño : No  
 Lavadora de ropa : Si  
 Horno de microondas : Si  
 Teléfono celular : Si  
 Televisión de paga : Si  
 Computadora personal : Si  
 Automóvil : Si  
 Calentador de agua : Si  
 Secadora de ropa : No  
 Línea telefónica : Si  
 Videograbadora : Si  
 Reproductor de películas en DVD : Si  
 Aspiradora : Si  
 Tostador de pan : Si  
 Personas de servicio de planta y/o entrada por salida : Si

38 . ¿CUÁNTOS FOCOS, CONTANDO LÁMPARAS DE TECHO, MESA Y PISO, HAY EN TU CASA? : 3 ó menos

39 . ¿CUÁL ES EL MATERIAL PREDOMINANTE EN LOS PISOS DE TU CASA? : Tierra

40 . ¿A CUÁNTO ASCIENDEN LOS INGRESOS MENSUALES DE TU FAMILIA, SIN CONSIDERAR IMPUESTOS?  
 (en términos de salarios mínimos) : Menos de 2 (Menos de \$3,500)  
 Si, tengo trabajo permanente con plaza o contrato

41 . ¿TRABAJAS? : plaza o contrato

42 . CONTESTA LA SIGUIENTE PREGUNTA SÓLO SI TRABAJAS  
 ¿Cuántas horas trabajas a la semana? : Menos de 16 horas

Guardar Datos (Si no has concluido podrás hacerlo después)

Figura 7.14: Tercera parte del cuestionario de concurso de selección a nivel licenciatura, con respuestas de un usuario.

En la Figura 7.14 se muestra la última parte del módulo ya mencionado, con las respuestas del usuario.

Si el alumno no ha terminado de responder el módulo, la siguiente vez que inicie su sesión en el sistema, éste podrá continuar respondiéndolo. Si se ha

terminado de contestar el módulo, pero el usuario desea ver sus respuestas, se podrá iniciar sesión y todas las respuestas se mostrarán como texto. En cada pregunta se podrá elegir solo una opción con la excepción de una pregunta (“¿Qué material de consulta usas en tu casa?”), donde se podrá elegir tantas opciones como se desee.

**UNAM** Universidad Nacional Autónoma de México

**DATOS ESTADÍSTICOS DEL CONCURSO DE SELECCIÓN A LA LICENCIATURA**

Esta encuesta tiene por objetivo conocer las características de la población que ingresará a la licenciatura por medio del concurso de selección. La información que nos proporcionas es confidencial y solo con fines estadísticos. Es fundamental que tengas la encuesta con información verídica y de forma completa, ya que los datos recogidos a través de ella nos permitirán tomar decisiones que redundarán en el mejoramiento de los diversos programas de apoyo académico y económico (becas).

Muchas gracias por tu cooperación.

**DATOS GENERALES**

NOMBRE	AP. PATERNO	AP. MATERNO	NOMBRE(S)
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

1. NÚMERO DE COMPROBANTE:

2. FECHA DE NACIMIENTO:

3. GÉNERO: FEMENINO

4. ESTADO CIVIL:

5. ¿TIENES HIJOS?:

6. TIPO DE ESCUELA A LA QUE ASISTISTE

Primaria:

Secundaria:

Bachillerato:

7. EL BACHILLERATO LO TERMINASTE EN:

**DATOS DE ESTUDIOS INMEDIATOS ANTERIORES**

6. ESCUELA DE PROCEDENCIA:

9. ¿EN QUÉ AÑO INICIASTE EL BACHILLERATO?:

10. ¿EN QUÉ AÑO TERMINASTE COMPLETO EL BACHILLERATO?:

11. ¿REALIZASTE EL BACHILLERATO EN TRES AÑOS?:

12. TURNO EN QUE HICISTE EL BACHILLERATO:

13. PROMEDIO DE CALIFICACIONES EN EL BACHILLERATO:

14. ¿CUÁNTOS EXÁMENES EXTRAORDINARIOS PRESENTASTE?:

15. ¿CUÁNTAS MATERIAS VOLVISTE A CURSAR?:

16. LA PREPARACIÓN QUE RECIBISTE FUE ...:

**DATOS PERSONALES**

17. ¿CUÁNTOS HERMANOS TIENES SIN CONSIDERAR MEDIOS HERMANOS?:

18. ¿CUÁL ES EL LUGAR QUE OCUPAS ENTRE TUS HERMANOS?:

19. ¿QUÉ TANTO INSISTEN TUS PADRES PARA QUE SIGAS ESTUDIANDO?:

Figura 7.15: Primera parte del cuestionario de concurso de selección a nivel licenciatura.

20. CALIFICA EL ÉXITO QUE HAS TENIDO COMO ESTUDIANTE :

21. ¿QUÉ MATERIAL DE CONSULTA USAS EN TU CASA?  
(Puedes marcar más de una opción pulsando la tecla control y las respuestas que deseas) :

Libros de texto

Revistas culturales

Otros libros

Enciclopedias

22. AL ESTUDIAR ACOSTUMBRAS ...  
(Elige una respuesta para cada renglón)

Primero leer todo el tema :

Subrayar las ideas principales :

Elaborar resúmenes del material :

Hacer síntesis tipo "acordeón" :

Resolver ejercicios para reafirmar el tema :

23. ESTUDIAS ...  
(Elige una respuesta para cada renglón)

Solo :

En equipo :

En la casa :

En la biblioteca :

En el parque :

Escuchando radio o televisión :

En la cama :

En escritorio o mesa :

En el transporte colectivo :

Sin distracción :

**SITUACIÓN SOCIOECONÓMICA**

24 . ¿Hablas alguna lengua indígena? :

25 . ¿Tus padres hablan alguna lengua indígena? :

26 . ¿Te consideras miembro de algún pueblo indígena? :

27 . SI UNA O VARIAS DE LAS TRES PREGUNTAS ANTERIORES FUERON AFIRMATIVAS, ESPECIFICA CUÁL LENGUA O PUEBLO INDÍGENA

Hablas la lengua :

Figura 7.16: Segunda parte (continuación) del cuestionario de concurso de selección a nivel licenciatura.

32. ¿CUÁNTAS PERSONAS DEPENDEN ECONÓMICAMENTE DE QUIEN SOSTIENE TUS ESTUDIOS? (inclúyete tú y a tu sostén económico):

33. LA CASA DONDE HABITAS ACTUALMENTE ES:

34. CUÁNTOS CUARTOS HAY EN LA CASA DONDE VIVES? (No cuentes cocina y baños):

35. ¿CON QUIÉN VIVES ACTUALMENTE?:

36. ¿CUÁNTAS PERSONAS HABITAN DONDE TÚ VIVES? (inclúyete tú):

37. ¿DE QUÉ BIENES Y SERVICIOS DISFRUTAS EN TU CASA?

Cuarto de baño:

Lavadora de ropa:

Horno de microondas:

Teléfono celular:

Televisión de paga:

Computadora personal:

Automóvil:

Calentador de agua:

Secadora de ropa:

Línea telefónica:

Videograbadora:

Reproductor de películas en DVD:

Aspiradora:

Tostador de pan:

Personas de servicio de planta y/o entrada por salida:

38. ¿CUÁNTOS FOCOS, CONTANDO LÁMPARAS DE TECHO, MESA Y PISO, HAY EN TU CASA?:

39. ¿CUÁL ES EL MATERIAL PREDOMINANTE EN LOS PISOS DE TU CASA?:

40. ¿A CUÁNTO ASCIENDEN LOS INGRESOS MENSUALES DE TU FAMILIA, SIN CONSIDERAR IMPUESTOS? (en términos de salarios mínimos):

41. ¿TRABAJAS?:

42. CONTESTA LA SIGUIENTE PREGUNTA SÓLO SI TRABAJAS  
¿Cuántas horas trabajas a la semana?:

Guardar Datos (Si no has concluido podrás hacerlo después)

Figura 7.17: Tercera parte del cuestionario de concurso de selección a nivel licenciatura.

En las Figuras 7.15, 7.16 y 7.17 se muestra el módulo de *Concurso de Selección a Nivel Licenciatura* cuando el usuario aún no ha respondido ninguna pregunta, sólo se muestran en texto fijo los datos personales del usuario, como son: nombre, identificador, fecha de nacimiento y género del usuario.

Posteriormente el usuario elige las respuestas de la encuesta<sup>12</sup> y el sis-

<sup>12</sup>El diseño de la interfaz es el mismo para todos los módulos, sólo varían las secciones

tema le mostrará en pantalla un mensaje indicándole los números de las preguntas que falta por contestar, como se observa en la Figura 7.18, en el caso de haber terminado de responder, el mensaje que se mostrará será un agradecimiento por haberlo hecho, posteriormente se podrá continuar con el módulo de PUEG (ver Figura 7.19).



Figura 7.18: Mensaje que indica las preguntas faltantes.

Si el usuario contestó con anterioridad alguna pregunta que se encuentre repetida deberá mostrarse en cada módulo al que pertenezca dicha pregunta, como se muestra en la Figura 7.19.

de cada uno y sus preguntas.


**Universidad Nacional Autónoma de México**  
 UNAM

II. Esta información se solicita sólo con fines estadísticos para realizar análisis de las trayectorias escolares de las y los estudiantes de la UNAM con perspectiva de género y promover acciones por la equidad. Su uso es confidencial, por lo que no se identificarán nombres.  
 Gracias por tu valiosa comprensión y colaboración.

**DATOS GENERALES**

1. SELECCIONA TU ESTADO CIVIL : .....  
 2. ¿TIENES HIJOS? : No  
 3. ¿CUÁNTOS HIJAS/HIJOS TIENES? : .....  
 4. POR FAVOR, MENCIONA LA FECHA DE NACIMIENTO -MES Y AÑO- DE CADA UNO DE TUS HIJAS Y/O HIJOS

HIJA (O) 1 MES : .....  
 HIJA (O) 1 AÑO : .....  
 HIJA (O) 2 MES : .....  
 HIJA (O) 2 AÑO : .....  
 HIJA (O) 3 MES : .....  
 HIJA (O) 3 AÑO : .....  
 HIJA (O) 4 MES : .....  
 HIJA (O) 4 AÑO : .....  
 HIJA (O) 5 MES : .....  
 HIJA (O) 5 AÑO : .....  
 HIJA (O) 6 MES : .....  
 HIJA (O) 6 AÑO : .....

[Guardar Datos \(Si no has concluido podrás hacerlo después\)](#)

[Regresar](#)
[Buzón](#)

Figura 7.19: Segunda parte de las encuestas, módulo “Equidad y Género (PUEG)”.

La segunda parte de todas las encuestas es el módulo de PUEG el cual no tiene acceso directo, por lo que siempre se deberá de acceder primero a cualquier otro módulo, antes de contestar la segunda parte del sistema.

## 7.6. Implementación del sistema

Como ya se mencionó anteriormente, la implementación del sistema SIUNE-DGAE se realizó en dos versiones, con el fin de realizar comparaciones entre

ambos sistemas, así como de la migración del mismo:

- La primera versión se encuentra implementada en C, un lenguaje de programación estructurado de propósito general. En esta versión ya se tenía un antecedente del sistema, por lo cual sólo se optimizó su diseño, así como las operaciones necesarias para almacenar las respuestas y el espacio que utiliza. Esta versión fue desarrollada utilizando la metodología de desarrollo de software en espiral.
- La segunda versión implementada en Ruby con el marco de trabajo Rails, un lenguaje de programación orientado a objetos, de propósito general. En esta versión se utilizó la metodología de desarrollo XP (eXtreme Programming), debido a que en la versión anterior se definieron a detalle en la especificación de requisitos, así como el flujo del sistema, en otras palabras, ya se conocía el trabajo a realizar.

### 7.6.1. Optimización del Sistema implementado en C.

Para dar una visión más detallada del sistema a continuación se describe el objetivo general de los módulos que lo componen:

- El módulo *sybase.c*, en este módulo se define la conexión a la base de datos y los atributos de ésta, como son, el usuario, la contraseña, el nombre de la conexión, entre otros.
- El módulo *aux.h*, este contiene los métodos que realizan alguna operación en la base de datos.
- El módulo *enc.c*, éste se encarga del control del sistema, se realizan las operaciones necesarias para convertir las respuestas proporcionadas por el usuario, en datos que serán almacenados en la base de datos, y a su vez se determina la pantalla que será mostrada al usuario, y finalmente se comprueba que los datos recibidos sean válidos.
- El módulo *funciones\_bit.h* contiene acciones auxiliares para codificar las respuestas de los usuarios en datos que requieran menos espacio para su almacenamiento.
- El módulo *enc.h* contiene las estructuras de datos necesarias (como son: aspirantes, cuestionario, respuesta, entre otras) para el módulo *enc.c*, con las cuales se realizan las operaciones sobre los datos.
- El módulo *aux3.c* se encarga de las vistas de los usuarios, el cual después de ser llamado por *enc.c* envía el código HTML que será mostrado.

- El módulo *cabecera.h* es la interfaz de *aux3.c*, donde se definen los métodos (*dame\_parametro()*, *mostrar\_parametro()*, entre otros) que utilizará *aux3.c*

El siguiente diagrama de secuencias (Figura 7.20) muestra el flujo del programa para iniciar sesión en el sistema, en éste se muestra la interacción que existe entre los diferentes módulos o programas del sistema. Para iniciar con esta acción, el usuario proporciona los datos que se solicitan en las Figuras 7.9, 7.10 y 7.11 según sea el módulo al que se ingresará, posteriormente el sistema por medio de un método en el módulo *encuesta\_g.c* se dirige al programa *enc.c* el cual a su vez verifica que los datos sean correctos a través de un método del módulo *aux.h*, el cual es el encargado de realizar las peticiones a la base de datos, con las conexiones definidas en el módulo *sybase.c*

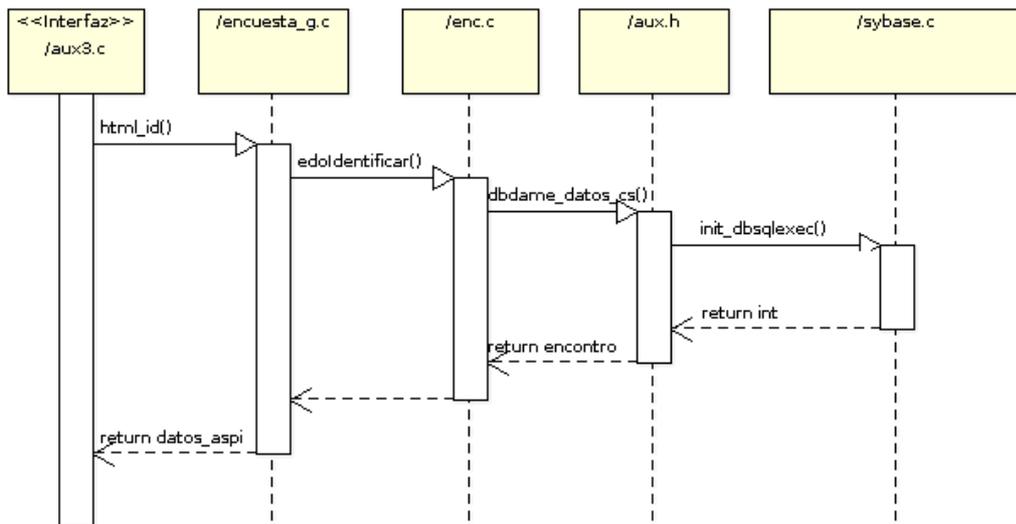


Figura 7.20: Diagrama de secuencia para iniciar sesión.

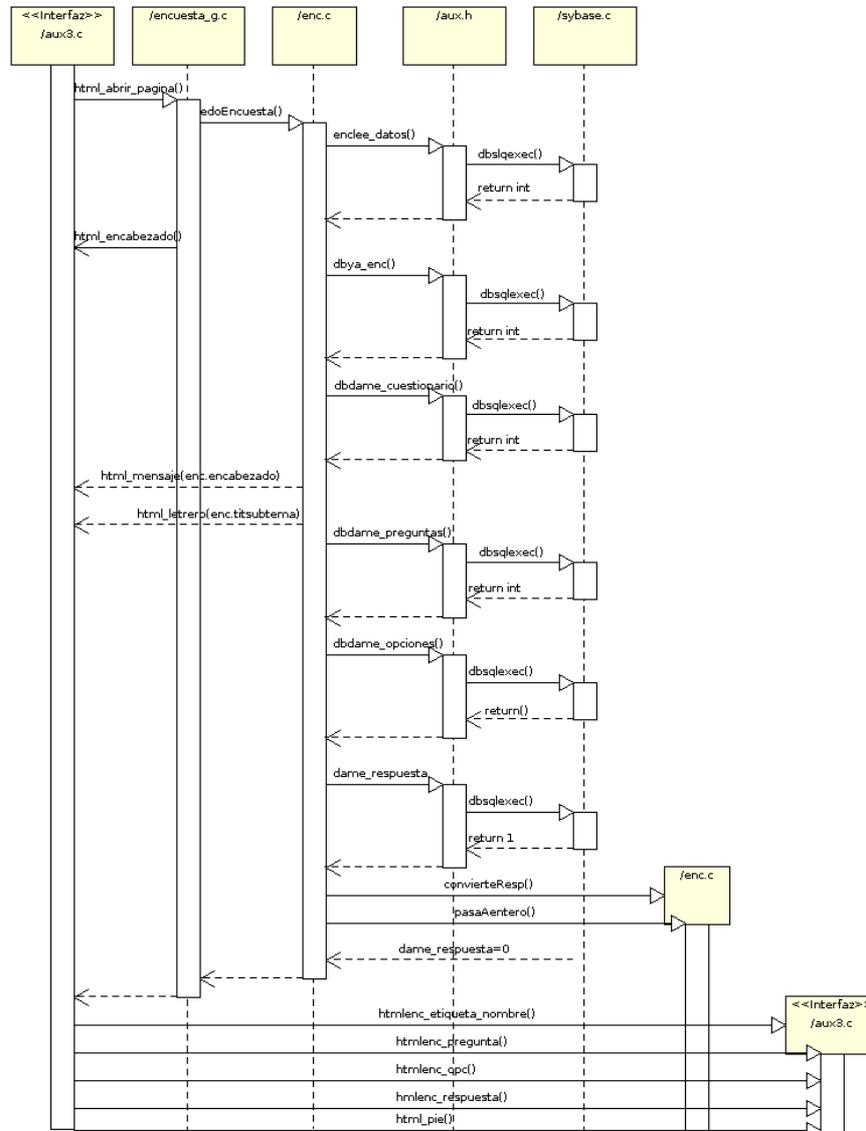


Figura 7.21: Diagrama de secuencia para mostrar una encuesta.

En la Figura 7.21 se puede observar las acciones que realiza el sistema implementado en C, para mostrar una encuesta en la pantalla, desde solicitar la página HTML, leer los datos del usuario, verificar si ya existen respuestas del mismo y obtener cada elemento de la encuesta, como lo son las secciones,

las preguntas y sus opciones o sus correspondientes respuestas, también se observa las operaciones que realiza a la base de datos en cada acción de este diagrama, como lo son `enclee_datos()` en la cual se buscan los datos del usuario en la base e datos, `dbya_enc()` en este método se verifica si ya se terminó de responder la encuesta de este usuario, el método `dbdame_Cuestionario()` busca en la base de datos las secciones de la encuesta y por cada una de éstas se realiza la búsqueda de sus correspondientes preguntas con el método `dbdame_preguntas()`, y para cada pregunta contestada con anterioridad se muestran sus respuestas haciendo uso del método `dame_respuesta()`, de lo contrario se realiza una búsqueda de sus correspondientes opciones con `dbdame_opciones()`.

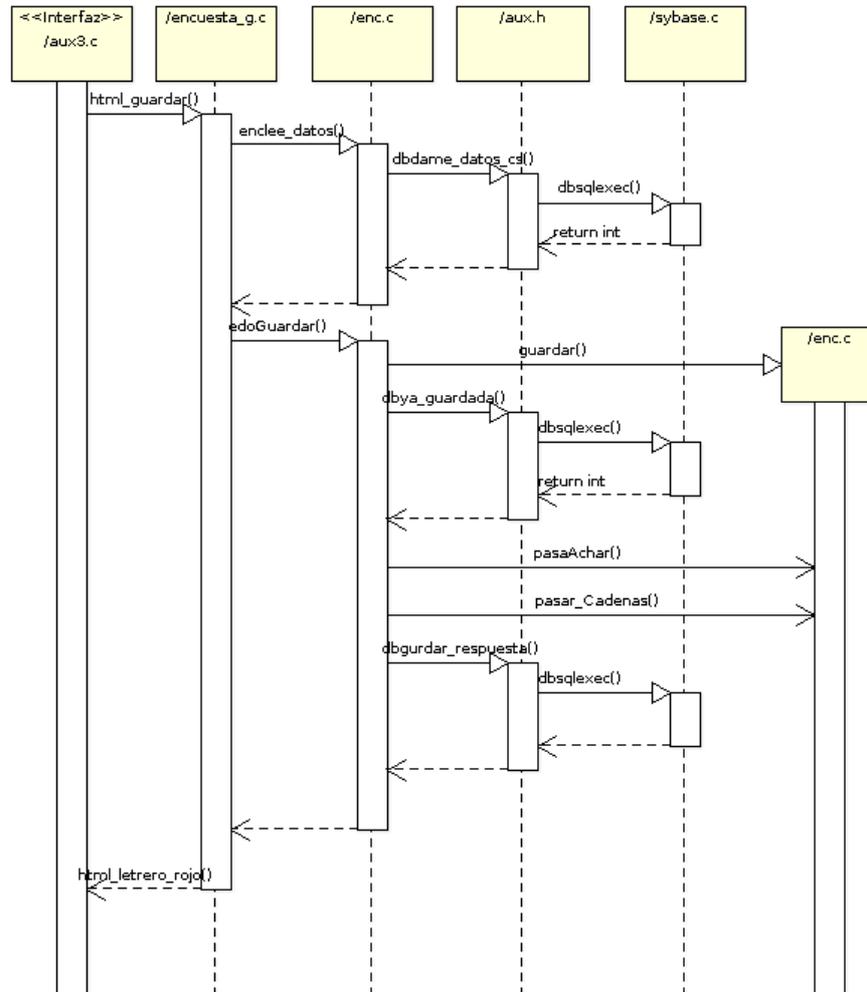


Figura 7.22: Diagrama de secuencia para almacenar las respuestas.

La Figura 7.22 muestra el flujo del sistema al almacenar las respuestas de un usuario, para realizar esta opción se solicita la opción de **guardar** desde la página Web, dependiendo del resultado de la verificación de respuestas existentes que se realiza con el método `dbya_guardada()`, se tienen dos opciones:

- Crear un nuevo registro si no existen datos anteriores del usuario en la base de datos.

- Actualizar los datos del usuario almacenados en la base de datos.

Después de estas verificaciones se realizan las operaciones necesarias (convertir la cadena a números respecto a su representación binaria y posteriormente convertir estos números en caracteres) para codificar las respuestas en lo que será almacenado en la base de datos.

### 7.6.2. Desarrollo del sistema implementado en Ruby on Rails.

La segunda versión del sistema implementado en Ruby, tiene una estructura fácil de entender, como ya se mencionó anteriormente, sigue el patrón MVC, en el que cada clase contenida en el modelo es una entidad en la base de datos, a la que le corresponde una clase en el *Controlador* (mencionado en la sección 7.5.1), que se encarga de recibir las acciones solicitadas en la *Vista* y solicitarle al *Modelo* la información de la base de datos necesaria para realizar las funciones del sistema de encuestas, por último cada clase en el módulo de la *Vista* muestra los datos que el *Controlador* les envía.

Para comprender mejor esto, a continuación se muestran algunos diagramas de secuencia, en los que se representa el flujo del sistema en determinada acción.

En la Figura 7.23 se observa como los componentes MVC del sistema interactúan para iniciar la sesión en el sistema, primero la parte de la *Vista* del aspirante envía los datos necesarios al *Controlador*, el cual por medio del *Modelo* del aspirante verifica que el usuario exista en la base de datos, en seguida regresa la respuesta al *Controlador*, el cual redirecciona la salida a una nueva *Vista*.

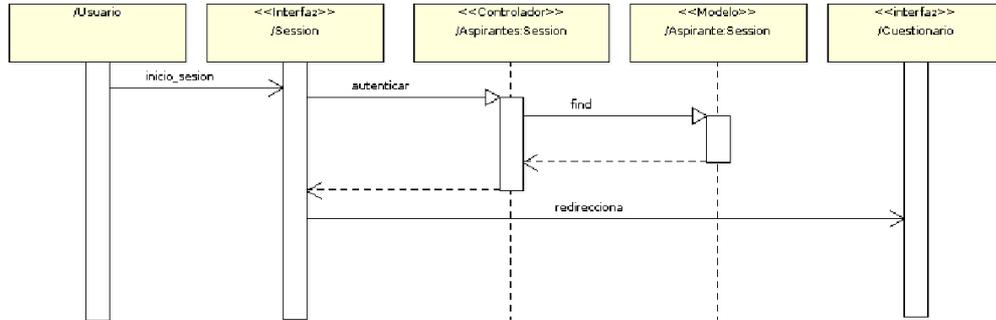


Figura 7.23: Diagrama de secuencia de inicio de sesión.

En la Figura 7.24 se muestra la interacción de los componentes del programa desarrollado en Ruby para mostrar una encuesta, se puede observar en ésta como se realizan las llamadas de la vista del sistema a componentes en el controlador y éstos a su vez a los componentes del modelo.

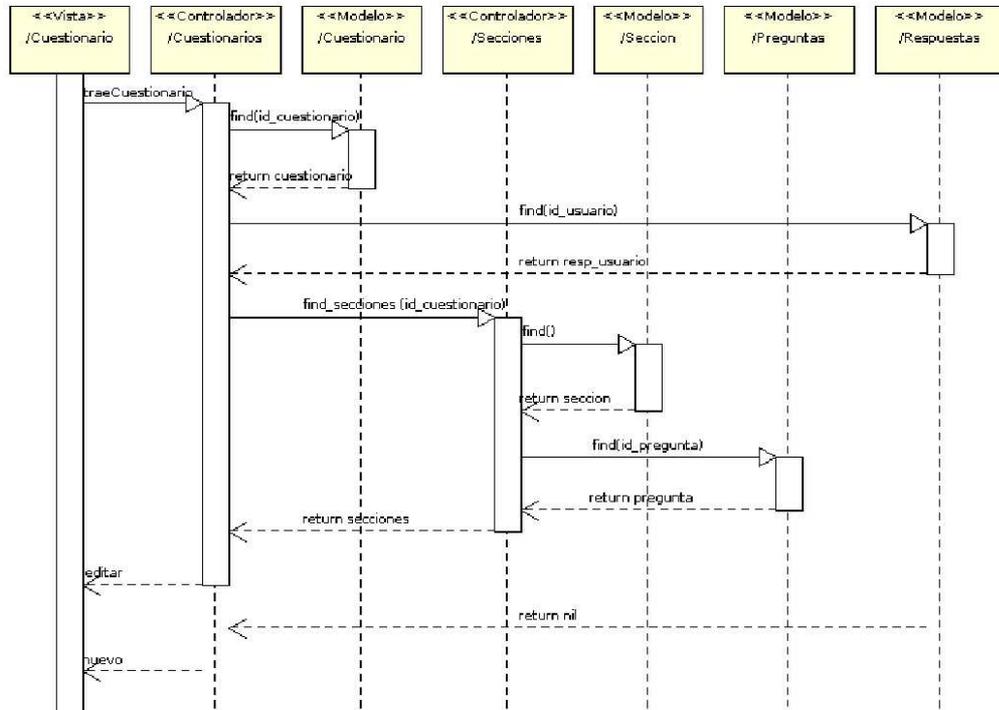


Figura 7.24: Diagrama de secuencia para mostrar una encuesta.

En la siguiente Figura 7.25 se muestra el flujo del programa desarrollado en Ruby para el almacenamiento de las respuestas. En este se puede observar que los datos de la *Vista* son enviados al *Controlador*, después de esto, se transforman las respuestas del usuario en la cadena que será almacenada, tal y como se mostró en la sección 7.2, lo cual es realizado por el *Modelo* de *Respuestas* y posteriormente se crea el objeto *respuesta*, el cual es almacenado en la base de datos, como lo señala el diagrama de secuencias.

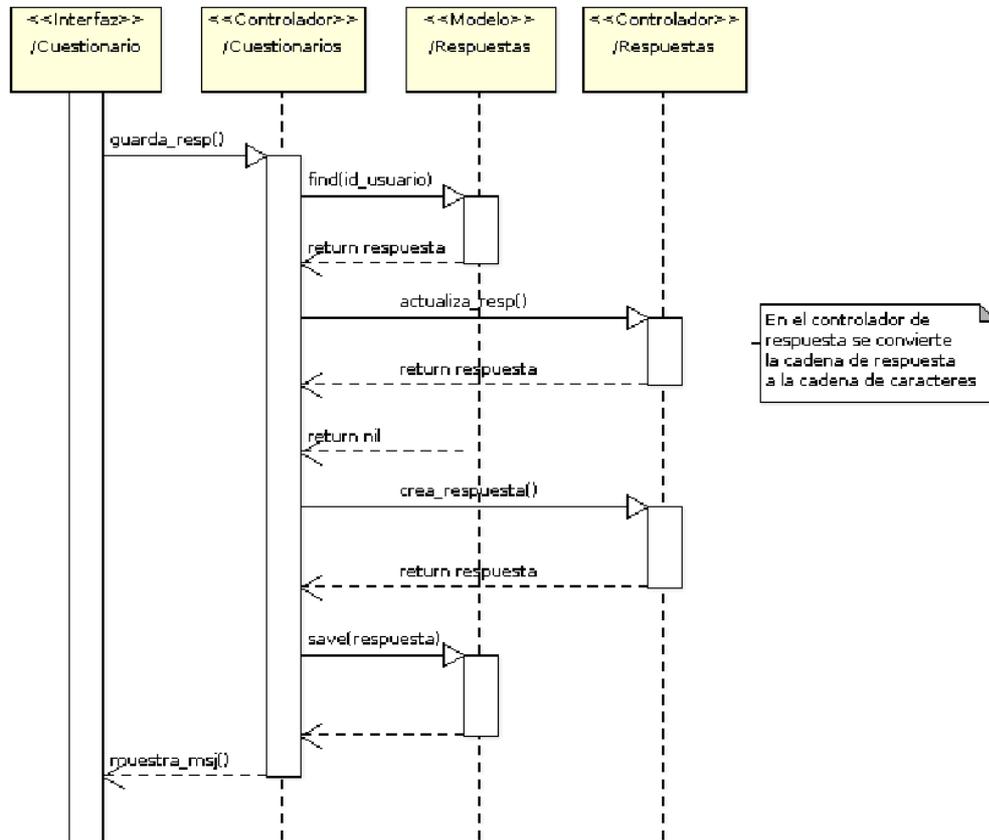


Figura 7.25: Diagrama de secuencia para almacenar las respuestas.

## Capítulo 8

# Comparación del sistema desarrollado con C y Ruby.

En este capítulo se mostrará la comparativa entre el diseño del sistema predecesor y el sistema del presente trabajo, las comparaciones de ambos sistemas se realizan basándose en el mantenimiento y legibilidad de éstos, destacando los beneficios y las diferencias entre los lenguajes de programación utilizados para el desarrollo del sistema.

Entre las principales diferencias que se puede encontrar entre éstos es el paradigma de programación al que pertenece cada uno, por su parte C corresponde al paradigma estructurado y aunque es un lenguaje de propósito general, el código HTML que se utiliza en este sistema es más legible en la versión desarrollada en Ruby, ya que utiliza el marco de trabajo Rails, el cual está diseñado para desarrollo de sistemas WEB, lo que facilita al programador realizar las plantillas HTML para las vistas del sistema.

Otra gran diferencia es que debido a que Ruby está dentro del paradigma de programación orientado a objetos manipula las entidades de la base de datos como objetos, cada uno con su clase correspondiente.

Por otro lado C trata las entidades de la base de datos como estructuras, a las cuales se puede acceder sin necesidad de utilizar métodos que regresen el valor de algún atributo de la estructura, como es el caso de Ruby, en donde se utilizan métodos que encapsulan el valor de los atributos de los objetos.

A continuación se muestra el cuadro 8.1 como resumen de las diferencias mencionadas anteriormente.

<b>Características de C y Ruby</b>						
	Lenguaje estructu- rado	Lenguaje orientado a objetos	Maneja estructu- ras	Maneja clases	Es de propósito general	Soporta marcos de traba- jo para web
<b>C</b>	√		√		√	
<b>Ruby</b>		√		√	√	√

Cuadro 8.1: Comparativa del desarrollo del sistema utilizando C VS Ruby.

En las siguientes secciones de este capítulo se muestran los métodos de ambas versiones del sistema, los cuales corresponden a los diagramas de secuencia antes mostrados.

## 8.1. Comparativa de diseños de bases de datos del sistema

Para mostrar la principal diferencia entre los diseños de la base de datos del sistema SIUNE-DGAE y su predecesor, se muestra las cantidades de registros utilizados por cada uno de éstos. Para demostrarlo se toma un ejemplo en el que 100,000 usuarios contestan alguna encuesta en ambos sistemas.

En el sistema predecesor del SIUNE-DGAE para almacenar las respuestas de un usuario se crea un registro por cada pregunta de la encuesta previamente respondida, de tal manera que:

$$\#registrosXUsuario = \#preguntasDeEncuesta$$

En la primera versión del sistema SIUNE-DGAE se crea un registro por cada usuario y cada encuesta, por lo que si el usuario contesta todas las encuestas se tendrían  $\#registrosXUsuario = \#encuestas$ , en cambio con el sistema predecesor se tendría:

$$\#registrosXUsuario = \sum_i \#preguntasDeEncuesta^1$$

---

<sup>1</sup>Donde i hace referencia al numero de encuestas desde i=1 hasta i=n, donde n es el número de encuestas definidas en la base de datos.

Con el diseño del presente sistema se logró reducir en gran medida la cantidad de registros utilizados para almacenar las respuestas de cada usuario sin importar el número de encuestas a las cuales accedió, quedando de la siguiente manera:

$$\#registrosXUsuario = 1$$

Es decir, si la encuesta respondida contiene 46 preguntas, entonces el número de registros utilizados por cada sistema con un total de 100,000 usuarios se puede observar en el Cuadro 8.2, en donde se puede observar que la fórmula para obtener los registros utilizados en cada sistema es:  $\#registrosUtilizados = \#registrosXUsuario \times \#usuarios$

	Sist. predecesor	SIUNE-DGAE
	46*100,000	1*100,000
Total	4,600,000	100,000

Cuadro 8.2: Ejemplo de registros utilizados para almacenar respuestas.

En el Cuadro 8.2 se muestra la diferencia por ciclos escolares contando el número de registros utilizados para almacenar las respuestas de los usuarios del sistema en el modelo anterior de la base de datos, mismos en el que se desarrolló el nuevo sistema.

Para resaltar aún más esta diferencia, se tomaron las respuestas de los usuarios entre los periodos “0405-0910”<sup>2</sup> que comprenden 6 ciclos escolares, del año 2004 al año 2009 y se pasaron al actual modelo de base de datos del sistema obteniendo las cantidades mostradas en el Cuadro 8.3, en el cual se puede observar por ejemplo en el renglón correspondiente al periodo **1011** con la clave del módulo **CSB-08** que en el sistema antecesor de encuesta se utilizaron **2,290,317** registros para almacenar las respuestas de los usuarios (mostrado en la columna “**Cant. Anterior**” del Cuadro 8.3) en cambio con el sistema de encuestas SIUNE-DGAE se utilizaron **33,193** registros (columna “**Cant. Actual**” del cuadro 8.3) obteniendo una diferencia de **2,257,124** registros tan solo en ese periodo, con el módulo **CSB-08**.

---

<sup>2</sup>Se explica con más detalle como se forman los periodos en la sección 6.1.1

Periodo	Cve modulo	Cant. Anterior	Cant. Actual	Diferencia
0405	CSL-03	3,378,745	49,829	3,328,916
0405	PR-03	996,082	14,931	981,151
0405	CSB-05	234,434	5,717	228,717
0506	CSL-03	531	17	514
0506	PR-03	856	22	834
0506	CSB-05	416	10	406
0607	CSL-03	7,243,946	105,926	7,138,020
0607	PR-03	1,778,354	26,092	1,752,262
0607	CSB-05	268	5	263
0708	CSL-03	5,381,601	78,630	5,3029,71
0708	CSL-08	2,818,894	39,980	2,779,514
0708	PR-08	1,765,519	24,832	1,740,687
0708	CSB-08	2,255,769	32,323	2,223,446
0809	CSL-08	9,174,740	128,480	9,046,260
0809	PR-08	1,852,873	26,083	1,826,790
0809	CSB-08	2,336,533	33,453	2,303,080
0910	CSL-08	10,204,773	142,862	10,061,911
0910	PR-08	1,994,639	28,023	1,966,911
0910	CSB-08	2,321,558	33,279	2,288,279
1011	CSL-08	10,506,650	150,095	10,356,555
1011	PR-08	2,039,520	29,136	2,010,384
1011	CSB-08	2,290,317	33,193	2,257,124
<b>Total</b>		<b>53,506,097</b>	<b>764,177</b>	<b>67,365,983</b>

Cuadro 8.3: Comparación de registros usados por los diferentes modelos.

Para hacer notar aún más la diferencia de registros utilizados mencionada en el Cuadro 8.3 se muestran a continuación en las Figuras 8.1, 8.2 y 8.3 la diferencia entre los registros usados por el modelo anterior de base de datos

contra el modelo actual, por ejemplo en la Figura 8.1 se puede observar en el período 0405 en el módulo **CSL-03** que la barra correspondiente al modelo anterior de la base de datos utiliza mas registros, por lo que es más alta que la barra correspondiente al modelo actual.

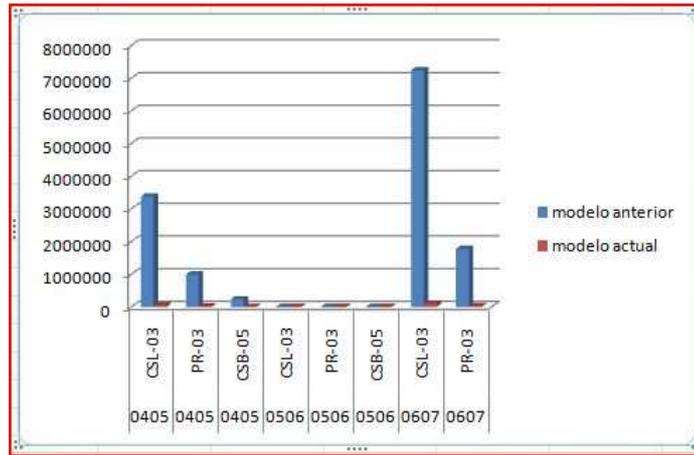


Figura 8.1: Gráfica con los períodos 0405 a 0607, que comprenden los años (2004-2006)

La Figura 8.1 muestra los registros usados en los modelos de bases de datos de los periodos 0405-0607, es decir se muestra los registros usados en los años 2004-2006.

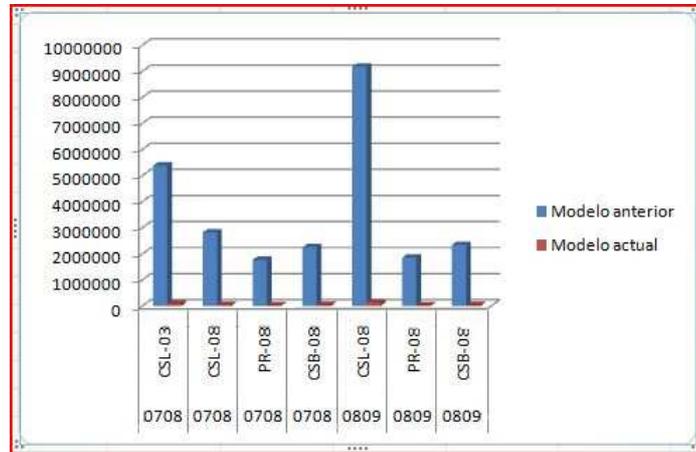


Figura 8.2: Gráfica comparativa de los períodos 0708 y 0809, que comprenden los años (2007,2008)

La Figura 8.2 muestra los registros usados en los modelos de bases de datos de los períodos 0708-0809, como puede observarse en la gráfica la tendencia de los registros usados por el modelo anterior (las barras de color azul) crecen rápidamente comparadas con las del modelo actual (barras de color vino), lo cual indica que con el modelo anterior se utiliza una cantidad mayor de registros para almacenar la información de los usuarios.

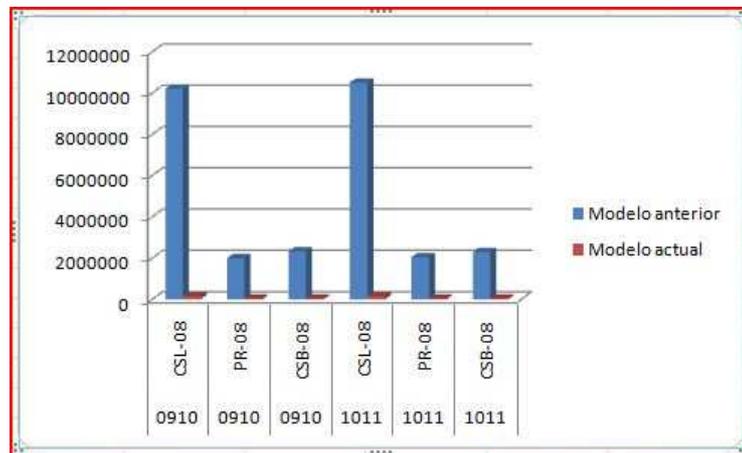


Figura 8.3: Gráfica comparativa de los períodos 0910 y 1011, que comprenden los años (2009,2010)

La Figura 8.3 muestra los registros usados en los modelos de bases de datos de los periodos “0910”-“1011”<sup>3</sup>, puede observarse en ésta que los registros usados en el modelo actual son considerablemente menores que el número de registros usados en el modelo anterior de la base de datos.

## 8.2. Inicio de sesión.

El Código 8.1 es una fracción del método que muestra el formulario para iniciar sesión en el sistema, como puede observarse en la línea 4, en la variable “nomid” se asigna el tipo de identificador con el que ingresará el usuario, posteriormente en la línea 5, se imprime a otra variable llamada “asal” el código HTML que será mostrada. Lo que se muestra en pantalla con este código puede observarse en la Figura 7.9 (Sección 7.5.2).

Código 8.1: Método que imprime en pantalla el formulario para inicio de sesión en el explorador.

---

```

1  switch(conc[0]) {
2  ...
3  case CONC_CS1:
4      strcpy(nomid, "<b>Número de Comprobante:</b><br> (en tu
          boleto credencial)");
5      fprintf(asal,
6          "\n <br><table class=\"estilotable\">"
7          "\n <tr id=\"azuloscuro\">"
8          "\n <td><b>%s</b></td>"
9          "\n <td colspan=\"3\">"
10         "\n <input type=\"text\" name=\"id\">"
11         "\n </td></tr> <tr id=\"azuloscuro\">"
12         "\n <td><b> Fecha de Nacimiento:</b></td>"
13         "\n <td colspan=\"3\"><br> "
14         "\n <input type=\"text\" name=\"fn\"> Día (2), Mes (2)
          y" "Año (4). <BR>Ejemplo: <em>16011978.</em>"
15         "</td> </tr> </table>", nomid);
16         break;
17     ...
18 }
```

---

<sup>3</sup>El período 1011 es el ciclo escolar que empieza en el segundo semestre del año 2010 y termina en el primer semestre del año 2011.

El Código 8.2 muestra el contenido de la página para iniciar sesión en el sistema implementado con Ruby, esta página es `sesion.html.haml`, en la cual se muestra el formulario para iniciar la sesión en el sistema implementado en Ruby. En este código se puede observar que las etiquetas de HTML son precedidas por el símbolo `%` (líneas 2, 3, 5) y los campos del formulario son especificados con el símbolo `=`, esto es porque se utiliza la biblioteca “`haml`” de Ruby, que con anterioridad se ha mencionado.

La jerarquía de las etiquetas HTML se indica con la indentación, de tal manera que no es necesario escribir el cierre de las etiquetas como se hace comúnmente en HTML, el resultado de este código siguiente se puede observar al igual que el Código 8.1 en la Figura 7.9 (Sección 7.5.2).

Código 8.2: Página que imprime el formulario para inicio de sesión.

---

---

```
1 .titulosseccion
2 - form_for @aspi, :url => {:action => :create} do |f|
3   %br
4   %table{:class => "estilotabla"}
5     %tr{:id => 'azuloscuro'}
6       %td
7         %b Número de Comprobante:
8         %br (en tu boleta credencial)
9       %td
10        = f.text_field :aspi_folio
11      %tr{:id => 'azuloscuro'}
12        %td
13          %b Fecha de Nacimiento:
14          %td
15            = f.text_field :aspi_fec_nac Día (2), Mes (2) y Año
16              (4).
17            %br Ejemplo:
18            %em 16011978.
```

---

---

### 8.3. Visualización de una encuesta.

En esta sección se muestran los códigos correspondientes en C y Ruby para mostrar las Figuras 7.15, 7.16 y 7.17, las cuales se pueden observar en la sección 7.5.2.

Se muestran los Códigos 8.3 y 8.4, para el sistema desarrollado en C y Ruby respectivamente, ambos códigos son utilizados para mostrar en la pan-

talla del explorador las preguntas de una encuesta, con sus correspondientes respuestas, si es que éstas existen previamente.

En la versión del sistema implementado en Ruby, antes de mostrarse el cuestionario se inicializan los objetos `cuestionario` y `respuestas`, para obtener los datos que se mostrarán en el navegador, el código 8.3 muestra como se realiza esto, en la línea 2 se puede observar que utilizando un objeto de la clase `Cuestionario` se realiza una búsqueda a la base de datos en donde la clave del módulo o encuesta sea algún valor especificado, lo mismo sucede con las respuestas del usuario en la línea 3.

Código 8.3: Método que inicializa la encuesta y las respuestas del usuario.

---

```
1 def edit
2   @cuestionarios = Cuestionario.where(:cst_cve => session[:
      cuestionario])
3   @respuesta = Respuesta.find_by_enc_id(session[: usuario].
      comprobante)
4 end
```

---

La página que se muestra en el explorador (el módulo con todos sus elementos), está representada en el Código 8.4, en el que se puede observar en la línea 1 que se llama a otra vista “parcial” o auxiliar que es llamada `cuestionario_cabecera`, en las siguientes líneas se puede observar la impresión del formulario para enviar las respuestas del usuario. Usando otra página auxiliar llamada “cuestionario” a la que se le envía una colección con los datos del cuestionario y la respuesta del usuario.

Código 8.4: Código para mostrar en pantalla una encuesta.

---

```
1 =render :partial => 'cuestionarios_shared/
      cuestionario_cabecera', :locals => {:cuestionario =>
      @cuestionarios.first}
2   -form_tag :cuestionario, :method => "PUT" do
3     %table{:id => "preguntas"}
4   =render :partial=>'cuestionarios_shared/cuestionario', :
      collection=> @cuestionarios[1..@cuestionarios.size-1],
      :locals => {:resp => @respuesta}
5     %hr
6     = submit_tag "Guardar_Datos_(Si_no_has_concluido_
      podrás_hacerlo_despues)", :id =>"commit"
```

---

El Código 8.5 ilustra el contenido del archivo “cuestionario\_cabecera”, en el que se imprime en un nuevo renglón el objetivo de cada sección (línea 4) y este a su vez llama al archivo auxiliar “seccion”.

Código 8.5: Código del archivo "partial"del cuestionario.

---

```
1 % tr
2   % td
3   . titulosseccion
4   =raw(cuestionario.cst_txt) = render :partial => '
      cuestionarios_shared/seccion' , :collection => Seccion
      .secciones_por_cst_y_gpo_preg(cuestionario.cst_cve ,
      cuestionario.cst_gpo_preg) , :locals => {:resp => resp}
```

---

Por último para mostrar la encuesta en pantalla, se muestra la vista auxiliar “seccion”, en la que se imprime cada pregunta, con sus correspondientes opciones y su número de pregunta de acuerdo a la encuesta y sección a la que corresponde, esto se puede observar en el Código 8.6.

Código 8.6: Código del archivo "partial"de secciones.

---

```
1 =render :partial => 'cuestionarios_shared/pregunta' , :
      collection => Pregunta.preguntas_por_cve_preg(seccion.
      cp_cst_cve_preg) , :locals =>{:num => seccion.
      cp_cst_num_preg , :resp => resp}
```

---

En el Código 8.7 se pueden observar las acciones que realiza el sistema implementado en C, para mostrar una encuesta en la pantalla, lo cual se hace al solicitar partes de la página HTML (línea 7), leer los datos del usuario (línea 4), verificar si ya existen respuestas del mismo (línea 3) y obtener cada elemento de la encuesta (líneas 6-30), como lo son las secciones, las preguntas y sus opciones o sus correspondientes respuestas. Las letras en rojo que corresponden a las líneas 17, 19 y 22 son comentarios del método.

Código 8.7: Código implementado en C para mostrar una encuesta en pantalla

---

```

1 void edoEncuesta() {
2     int ok, i, j, edo_enc;
3     ok= enclee_datos();
4     if(ok){
5         edo_enc= dbya_enc(cst, tipo_id, id);
6         dbdame_Cuestionarios(cst,&enc);
7         html_mensaje(enc.encabezado);
8         for(i=0, j= 0; i< enc.subtemas; i++){
9             html_letrero(enc.titsubte[i]);
10            while (dbdame_Preguntas(cst,enc.cvesubte[i])){
11                dbdame_Respuesta(preg.op);
12                if(edo_enc!=2 && !strcmp(a[1].val,"FIN"))
13                    htmlenc_pregunta(j++);
14                else
15                    if(edo_enc==2 && !strcmp(a[1].val,"FIN")){
16                        if(strcmp(preg.op,"000"))
17                            //muestra etiqueta de pregunta
18                            htmlenc_etiqueta(preg.subpreg, preg.num_preg,
19                                preg.txt);
20                            else //muestra sexo, fecha de nacimiento
21                                htmlenc_pregunta(j++);
22                            }else //preguntas no obligatorias
23                                if(strcmp(a[1].val,"FIN") && !strcmp(cst,"
24                                    PUEG-10"))
25                                    htmlenc_respuesta(j++, 1);
26                                else
27                                    htmlenc_respuesta(j++, 0);
28                            }
29                            }
30                            }
31                            }
32                            }
33                            }
34                            }
35                            }
36                            }
37                            }
38                            }
39                            }
40                            }
41                            }
42                            }
43                            }
44                            }
45                            }
46                            }
47                            }
48                            }
49                            }
50                            }
51                            }
52                            }
53                            }
54                            }
55                            }
56                            }
57                            }
58                            }
59                            }
60                            }
61                            }
62                            }
63                            }
64                            }
65                            }
66                            }
67                            }
68                            }
69                            }
70                            }
71                            }
72                            }
73                            }
74                            }
75                            }
76                            }
77                            }
78                            }
79                            }
80                            }
81                            }
82                            }
83                            }
84                            }
85                            }
86                            }
87                            }
88                            }
89                            }
90                            }
91                            }
92                            }
93                            }
94                            }
95                            }
96                            }
97                            }
98                            }
99                            }
100                           }

```

---

## 8.4. Codificación de las respuestas.

Para codificar las respuestas se utilizó el algoritmo explicado en la sección 7.1, a continuación se muestra el Código 8.8, correspondiente a este algoritmo desarrollado en Ruby. Cabe mencionar que éste cuenta con la cadena de respuestas con 0's y 1's formada por las respuestas del usuario, colocando los 1's de acuerdo a las posiciones inicial y final de cada pregunta. En este código se puede observar (línea 5) el recorrido de toda la cadena, posteriormente se genera una subcadena de longitud 7 (línea 6), para cada caracter de esta subcadena se revisa si el caracter es igual a '1' y se realiza la acción correspondiente, ya sea un corrimiento de bits ó un OR lógico (líneas 8-13), después de analizar cada caracter de la subcadena el entero obtenido de las operaciones en los bits de éste, es convertido en tipo caracter (línea 14), el cual será concatenado a una cadena para ser regresada por dicho método.

Código 8.8: Códificando la cadena de respuestas en Ruby.

---

```
1 def codifica_cadena
2   inicial = 0
3   final = 6
4   cadena_final=""
5   while(inicial < self.enc_resp.size)
6     subcad = self.enc_resp[inicial..final]
7     entero=1
8     subcad.each_char do |c|
9       entero = entero << 1
10      if c=='1'
11        entero = entero | 1
12      end
13    end
14    cadena_final+=entero.chr
15    inicial=final+1
16    final+=7
17  end
18  self.enc_resp = cadena_final
19 end
```

---

Para lograr observar las diferencias entre las dos versiones del sistema, también se añade el Código 8.9, correspondiente al algoritmo de codificación de respuestas desarrollado en C. En éste código se observa en las líneas 2-5 la declaración de las variables necesarias para el desarrollo del método, cabe mencionar que al igual que en la versión anterior se recorre toda la cadena

Código 8.9: Códificando la cadena de respuestas en C.

---

```
1 void pasar_bits() {
2     int inicial=0, final=6, j=0;
3     int unsigned entero=1;
4     char aux1[300];
5     char *aux2;
6     while(inicial<strlen(aux)){
7         entero=1;
8         strncpy(aux2, aux, inicial, final);
9         while(j<strlen(aux2)){
10            entero=entero<<1;
11            if(aux2[j]=='1')
12                entero=entero|1;
13            }
14            sprintf(aux1, "%s%c", aux1, (char)entero);
15            inicial=final+1;
16            final+=7;
17            j=0;
18        }
19    } // fin while
20    sprintf(aux1, "%s%c", aux1, '\\0');
21    return aux1;
22 }
```

---

(línea 6), y se genera una subcadena de longitud 7 (línea 8), posteriormente se verifica si el caracter en el que se encuentra posicionado es igual a '1' y se realizan las operaciones necesarias sobre los bits de un entero (líneas 10-13), al terminar de recorrer la subcadena se convierte el entero en caracter y finalmente se concatena a la cadena que será regresada por el método (línea 14).

## Capítulo 9

# Conclusiones.

Tanto el Instituto Nacional Politécnico (IPN) como la Universidad Autónoma Metropolitana (UAM) realizan encuestas socioeconómicas a los aspirantes de ingreso y a sus alumnos para crear sus correspondientes perfiles, los cuales como se ha explicado en este trabajo conllevan grandes beneficios.

Por su parte el IPN realiza dichas encuestas utilizando un formulario impreso y la UAM las realiza a través de su sitio Web, (cabe mencionar que únicamente cuenta con encuestas a nivel licenciatura). Ahora haciendo la comparación con la Universidad Nacional Autónoma de México (UNAM) resulta que nuestra Universidad (UNAM) realiza estas encuestas para los niveles de bachillerato y licenciatura tal y como puede observarse en el Cuadro 6.1, la UNAM atiende a un mayor número de personas que contestan dichas encuestas.

Debido a los cambios mencionados a lo largo de este trabajo como lo son: el incremento en el número de aspirantes en cada ciclo escolar para los diferentes niveles escolares de ingreso que brinda la UNAM, así como también debido al avance de las Tecnologías de Información y Comunicación (TIC), resulta entonces indispensable contar con un sistema novedoso y mantenible para la UNAM para que a través de éste se logre automatizar y agilizar el proceso; así mismo mediante el uso de nuevas tecnologías y métodos de desarrollo de software enfocados en la funcionalidad, usabilidad y mantenibilidad utilizados en el sistema SIUNE-DGAE se propone que sean los pasos iniciales para impulsar a la DGAE a través del desarrollo de sistemas de información actualizados, innovadores y a la vanguardia de las tecnologías de la información logrando así que cumplan con las necesidades actuales de este sistema como lo son almacenar, manipular y acceder de una manera más sencilla a la información brindada por miles de usuarios y que está en

constante demanda por parte de distintas dependencias de la UNAM; y que en un futuro el sistema SIUNE-DGAE podrá ser la referencia a una nueva etapa en los sistemas de información de la DGAE.

Por medio de este sistema de encuestas, se puede realizar estudios socio-económicos, así como obtener la información necesaria para tomar decisiones acerca de proyectos académicos o de apoyo económico, los cuales beneficiarán a la población estudiantil de la UNAM, así como proponer posibles acciones que enriquezcan el aspecto académico, como lo son propuestas para mejorar la enseñanza de académicos y alumnos, creación de talleres que ayuden a los alumnos a obtener un mejor aprendizaje, así como agilizar la atención a los aspirantes y alumnos de la UNAM.

Respecto a las dos versiones en las que se desarrolló el presente sistema, ambas cumplen con los requisitos especificados desde un principio; sin embargo, el sistema desarrollado en Ruby resultó ser más sencillo de mantener que la versión desarrollada en C, debido a que utiliza el marco de trabajo Rails y la arquitectura MVC, los cuales proporcionan una organización de los componentes del sistema mejor distribuida y de fácil comprensión, obteniendo una manera sencilla de identificar cada componente, así como las partes que conforman cada uno de éstos, además Ruby cuenta con manejo de excepciones y recolector de basura.

Para este sistema en particular se puede concluir que el lenguaje de programación Ruby, con el marco de trabajo Rails, resultó ser el más adecuado ya que está diseñado para ayudar en el desarrollo de sistemas WEB, brindándole a los programadores las herramientas antes mencionadas (recolector de basura, manejo de excepciones, entre otras). Cabe remarcar que C es un lenguaje de programación con gran alcance y desempeño en tiempo, ya que el sistema de información Web aquí presentado logró cumplir con todos los requerimientos establecidos desde un principio, sin embargo, éste requiere de un número mayor de configuraciones (como son los sockets), así como de una arquitectura más compleja que el sistema en la otra versión, por lo que el sistema desarrollado con Ruby resultó ser más rentable, esto debido a que en la DGAE se requiere de constantes modificaciones a los sistemas de información por lo que la mantenibilidad de éstos es un importante factor tomado en cuenta en el proceso de desarrollo de los mismos.

Como se observó en el Cuadro 8.3 al utilizar el sistema de encuestas SIUNE-DGAE se logró obtener un número menor de registros para almacenar las respuestas de los aspirantes y alumnos de los ciclos escolares 2004 hasta 2010 (períodos 0405-1011), lo que permitió tener toda la información de estos períodos en la base de datos, lo cual anteriormente no se podía realizar, ya que al utilizar el sistema antecesor de encuestas, el número de

registros almacenados tan solo en un ciclo escolar comprendía una cantidad de millones de registros, con lo cual la base de datos alcanzaba su máxima capacidad de almacenamiento (se llenaba la memoria del servidor de bases de datos, por lo que solo se podía tener almacenada la información de un solo ciclo escolar en la base de datos), teniendo que realizar respaldos en cintas de esta información para no saturar la base de datos, provocando así que cuando se requería conocer algún dato de esta información se tenía que restaurar éstos en una base de datos para tener acceso a ella, incrementando el tiempo de respuesta dependiendo del tiempo que tardará la restauración. Sin embargo se ve afectada la explotación de la información de las respuestas del ciclo actual, ya que antes de conocer la información almacenada esta debe ser decodificada, por lo que se requiere un tiempo mayor para esto.

Algunas de las dependencias de la UNAM que son beneficiadas con este sistema son la Dirección General de Planeación (DGPL), ya que con esta información se crean los perfiles de los aspirantes y alumnos de la UNAM con el fin de ayudar a conocer los antecedentes socioeconómicos y académicos más relevantes de éstos; y la Dirección General de Orientación y Servicios Educativos (DGOSE) que a través de los perfiles creados puede influenciar en una mejor toma de decisiones, para crear fondos de apoyo a los estudiantes, como son las becas, proyectos de orientación y mejorar los servicios de ingreso y permanencia brindados a los estudiantes de la UNAM.

Es importante mencionar que tener disponible toda esta información es de gran importancia para la DGAE, ya que permite dar respuesta a peticiones de distintas dependencias de la UNAM, que involucren los datos obtenidos en las encuestas socioeconómicas, las cuales son constantemente solicitados en reportes que involucran la información de dichas encuestas (desde datos personales y hábitos de estudios de diferentes deportistas de la UNAM hasta brindar los datos académicos de todos los aspirantes de ingreso a la UNAM comprendidos en cierto ciclo escolar a la fecha, o bien para realizar estudios demográficos o sociales).

En futuras actualizaciones del software se proponen mejoras como lo son eliminar los casos especiales, unificando así el comportamiento del sistema y evitando poner “parches” en los componentes del sistema, así como revisar el diseño de la base de datos y renovarlo de ser necesario.

En conclusión este sistema proporciona los datos necesarios para la DGPL y la DGOSE, así como para estudios de alguna área en particular o estudios estadísticos de la comunidad estudiantil de la UNAM en general. Por medio de este sistema se logró acceder, manipular y almacenar información que anteriormente requería de un número mayor de procesos y tiempo que en la actualidad.

# Índice de cuadros

1.1. Matrículas que solicitan ingresar anualmente de acuerdo al nivel académico.[4, 15, 16] . . . . .	2
3.1. Número de aspirantes a ingresar en las principales universidades públicas en los últimos tres años. . . . .	19
5.1. Clasificación del software.[28] . . . . .	52
5.2. Metodologías ágiles VS metodologías tradicionales. . . . .	61
6.1. Cantidad de aspirantes que contestaron la encuesta socioeconómica en las diferentes universidades públicas en los tres últimos años. . . . .	67
8.1. Comparativa del desarrollo del sistema utilizando C VS Ruby.	111
8.2. Ejemplo de registros utilizados para almacenar respuestas. . .	112
8.3. Comparación de registros usados por los diferentes modelos. .	113

# Índice de figuras

3.1. Arquitectura Cliente-Servidor de bases de datos . . . . .	23
3.2. Arquitectura paralela de bases de datos [23]. . . . .	25
3.3. Componentes de diagrama E-R [23]. . . . .	29
7.1. Diseño de la base de datos del sistema predecesor. . . . .	78
7.2. Primer modificación en la tabla Encuestas. . . . .	79
7.3. Nuevo diseño de la base de datos. . . . .	80
7.4. Diagrama de clases del sistema desarrollado en C. . . . .	82
7.5. Arquitectura cliente-servidor . . . . .	83
7.6. Diagrama de clases del sistema desarrollado en Ruby on Rails. . . . .	85
7.7. Diagrama MVC del sistema SIUNE-DGAE. . . . .	87
7.8. Menú página principal DGAE. . . . .	89
7.9. Inicio de sesión, módulo “Concurso de selección a nivel Licenciatura”. . . . .	90
7.10. Inicio de sesión, módulo “Pase Reglamentado”. . . . .	91
7.11. Inicio de sesión, módulo “Concurso de selección a nivel Bachillerato”. . . . .	92
7.12. Primera parte del cuestionario de concurso de selección a nivel licenciatura, con respuestas de un usuario. . . . .	93
7.13. Continuación del cuestionario de concurso de selección a nivel licenciatura, con respuestas de un usuario. . . . .	94
7.14. Tercera parte del cuestionario de concurso de selección a nivel licenciatura, con respuestas de un usuario. . . . .	95
7.15. Primera parte del cuestionario de concurso de selección a nivel licenciatura. . . . .	96
7.16. Segunda parte (continuación) del cuestionario de concurso de selección a nivel licenciatura. . . . .	97
7.17. Tercera parte del cuestionario de concurso de selección a nivel licenciatura. . . . .	98

---

7.18. Mensaje que indica las preguntas faltantes. . . . .	99
7.19. Segunda parte de las encuestas, módulo “Equidad y Género (PUEG)”. . . . .	100
7.20. Diagrama de secuencia para iniciar sesión. . . . .	102
7.21. Diagrama de secuencia para mostrar una encuesta. . . . .	103
7.22. Diagrama de secuencia para almacenar las respuestas. . . . .	105
7.23. Diagrama de secuencia de inicio de sesión. . . . .	107
7.24. Diagrama de secuencia para mostrar una encuesta. . . . .	108
7.25. Diagrama de secuencia para almacenar las respuestas. . . . .	109
8.1. Gráfica con los períodos 0405 a 0607, que comprenden los años (2004-2006) . . . . .	114
8.2. Gráfica comparativa de los períodos 0708 y 0809, que com- prenden los años (2007,2008) . . . . .	115
8.3. Gráfica comparativa de los períodos 0910 y 1011, que com- prenden los años (2009,2010) . . . . .	115

# Bibliografía

- [1] Sitio Oficial de la Legislación universitaria. <http://www.dgelu.unam.mx/m2.htm> (12/05/10 6:00 pm).
- [2] García Cantú Gastón, **Historia en voz alta**, Editorial J. Mortiz, 1988, 115 páginas.
- [3] García Cantú Gastón, **Años críticos: la UNAM 1968-1987**, Coordinación de Difusión Cultural, Dirección de Literatura, UNAM, 1987, 485 páginas.
- [4] Sitio Oficial de la Dirección General de Planeación. Datos Estadísticos. [http://www.ipn.mx/wps/wcm/estadistica\\_20093.PDF](http://www.ipn.mx/wps/wcm/estadistica_20093.PDF) (14/05/10 8:00 pm).
- [5] Sitio Oficial de la UNAM <http://www.unam.mx> (14/05/10 16:00 pm).
- [6] Sarabia Alegría, José María, Pascual Sáez, José María, **Curso básico de estadística**, Universidad de Canabria, 2005, 378 páginas (Yale y Kendal, 1954).
- [7] M. Ross Sheldon, **Introducción a la estadística**, Reverte, 2007, 809 páginas.
- [8] Fernandez Fernandez Santiago, **Estadística Descriptiva**, ESIC Editoriales, 566 páginas.
- [9] J. Salkind Neil, **Métodos de investigación**, Pearson Educación, 1998, pág 103.
- [10] Coscolluela Mas Antoni, Jaume Turbany I Oset , **Estadística descriptiva**, Editorial UOC, 37 páginas.
- [11] Hyman Herbert, **Diseño y análisis de las encuestas sociales**, Amorrortu editores, 2001, 529 páginas.

- [12] Sitio Oficial de la UNAM en números. <http://www.estadistica.unam.mx/numeralia/> (12/09/10 10:52 pm).
- [13] Sitio Oficial de la Dirección General de Planeación. <http://www.planeacion.unam.mx/> (12/08/2010 3:00 pm.)
- [14] Sitio Oficial de la Dirección General de Planeación. Perfil de Aspirantes. <http://www.planeacion.unam.mx/Publicaciones/pdf/perfiles/aspirantes> (12/08/2010 5:00 pm.)
- [15] Sitio Oficial. Acciones de transparencia. Universidad Autónoma Metropolitana <http://www.transparencia.uam.mx/> (30/06/11 10:37 pm)
- [16] Sitio Oficial del Portal alternativo de las acciones de transparencia en el IPN 2011 [www.transparencia.ipn.mx/](http://www.transparencia.ipn.mx/) (30/06/11 11:37 pm)
- [17] Manuel García Ferrando, **Socioestadística. Introducción a la estadística en sociología**, Alianza Editorial, 2001, 560 páginas.
- [18] Elena Abascal, Ildefonso Grande Esteban, **Análisis de encuestas**, ESIC Editorial, 2005, 291 páginas.
- [19] Estadística descriptiva. [http://es.wikipedia.org/wiki/Estadistica\\_descriptiva](http://es.wikipedia.org/wiki/Estadistica_descriptiva) (30/11/10 12:00 pm).
- [20] Sitio Oficial. Publicaciones digitales. <http://www.planeacion.unam.mx/Publicaciones/> (03/05/2010 1:05 am).
- [21] Sitio Oficial. Datos Estadísticos. <http://www.ipn.mx/connect/estadisticabcd.pdf> (14/05/10 8:00 pm).
- [22] Sitio Oficial de la Dirección General de Administración Escolar. <http://www.dgae.unam.mx> (14/07/10 3:00 pm).
- [23] Silberschatz Abraham, F. Korth Henry, Sudarshan S., **Fundamentos de Bases de Datos**, España, 2002, 787 páginas.
- [24] Batini Carlo, **Diseño conceptual de bases de datos: un enfoque de entidades relacionales**, Ediciones Díaz de Santos, 1994, 546 páginas.
- [25] C.J. Date, **Introducción a los sistemas de bases de datos**, Pearson Educación, México, 2001, 936 páginas.

- 
- [26] Tamayo Alzate Alonso, **Sistemas de información**, Universidad Nacional de Colombia, 2001, 121 páginas.
- [27] Ralph M. Stair, George W. Reynolds, **Principios de sistemas de información: enfoque administrativo**, Cengage Learning Editores, 2000.
- [28] Giner de la Fuente Fernando, **Los sistemas de información en la sociedad del conocimiento**, Artegraft, 2004 España, 212 páginas.
- [29] De Castro Valeria, Marcos Esperanza, **Desarrollo de sistemas de información web orientados a servicios**, kybele consulting, México, 8 páginas.
- [30] Morales Domingo Antonio, **Análisis y desarrollo de sistemas de información**, Bib. Orton IICA / CATIE, 1976, 20 páginas.
- [31] Kenneth C. Loudon, **Lenguajes de programación: principios y práctica**, Cengage Learning Editores, 2004, pág 238.
- [32] W. Kernighan Brian, M. Ritchie Dennis, **El lenguaje de programación C**, Pearson Educación, 1991, 294 páginas.
- [33] Sitio Oficial. Índice TIOBE.  
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>  
(09/06/2011 11:30 am)
- [34] Osorio Rivera, Fray León, **Lógica y programación orientada a los objetos: un inicio al desarrollo de software**, ITM, 2005, 480 páginas.
- [35] Sitio Oficial. Lenguaje de Programación Ruby. <http://www.ruby-lang.org/es/> (11/4/11 13:13 pm).
- [36] Sitio Oficial. Slim. <http://slim-lang.com/> (12/04/11 12:41 pm).
- [37] Sitio Oficial. Haml. <http://haml-lang.com/> (13/04/11 11:10 pm).
- [38] Günter Born, **Compendium HTML: con XHTML, DHTML, CSS, XML, XSL y WML**, Marcombo, 2001, 974 páginas.
- [39] Bruce Tate, Curt Hibbs, **Ruby on Rails: up and running**, O'Reilly Media, Inc., 2006, 167 páginas.

- 
- [40] Christian Hellsten, Jarkko Laine, **Beginning Ruby on Rails e-commerce: from novice to professional**, Apress, 2006, 424 páginas.
- [41] Pressman, Roger S., **Ingeniería del Software, un Enfoque Práctico (Quinta edición)**, Mc Graw Hill, 2002, 642 páginas.
- [42] Ian Sommerville, **Ingeniería del software**, Pearson Educación, 2005, 687 páginas.
- [43] Khaled Elleithy, **Advanced Techniques in Computing Sciences and Software Engineering**, Springer, 2010, 599 páginas.
- [44] F. Alonso Amo, Loïc Martínez Normand, **Introducción a la ingeniería del software**, Delta Publicaciones, 2005, 542 páginas.
- [45] Metodología de desarrollo de software.  
<http://es.wikipedia.org/wiki/Metodologiadesarrollodesoftware>  
(12/4/11 11:36 am).
- [46] Modelo de prototipos.  
[http://es.wikipedia.org/wiki/Modelo\\_de\\_prototipos](http://es.wikipedia.org/wiki/Modelo_de_prototipos) (12/4/11 13:25 pm).
- [47] Benet Campderrich Falgueras, **Ingeniería del software**, Editorial UOC, 2003, 320 páginas.
- [48] José Salvador Sánchez Garreta, **Ingeniería de proyectos informáticos: actividades y procedimientos**, Universidad Jaume I, 2003, 165 páginas
- [49] June Jamrichoja Parsons, **Conceptos de Computación: Nuevas Perspectivas**, Cengage Learning Editores, 2008, 848 páginas.
- [50] Modelo cascada.  
[http://es.wikipedia.org/wiki/Desarrollo\\_en\\_cascada](http://es.wikipedia.org/wiki/Desarrollo_en_cascada) (13/4/11 10:13 am).
- [51] Coral Moraga Ma Angeles, Piattini Velthius Mario G, **Calidad Del Producto Y Proceso Software**, Editorial Ra-Ma, 2010, 665 páginas.
- [52] Thomas H. Davenport, **Motivar, retener y crear valor en la era del conocimiento**, Deusto, 2006, 197 páginas.

- 
- [53] Prudencio Óscar Mochi Alemán, **La industria del software en México en el contexto internacional y latinoamericano**, UNAM, 2006, 261 páginas.
- [54] Manuel Ortega Cantero, José Bravo Rodríguez, Julián Ruiz Hernández, **Informática industrial**, Univ de Castilla La Mancha, 1997, 262 páginas.
- [55] Mooney, **Bringing Portability to the Software Process**, West Virginia University. Dept. of Statistics and Computer Science, 1997, pág 9.
- [56] Garey Ken, **Software Portability: Weighing Options, Making Choices**, The CPA Journal 77,11, 2007, pág 3.
- [57] Andres Gomez De Silva Garza, Ignacio De Jesus Ania Briseno, **Introducción a la computación**, Cengage Learning Editores, 2008, 522 páginas.
- [58] Kent Beck, **Extreme programming explained: embrace change**, Addison-Wesley Professional, 2000 - 190 páginas
- [59] Catherine Plaisant, Ben Shneiderman, **Diseño de interfaces de usuario: estrategias para una interacción persona-computadora efectiva**, Pearson Education, 2006, 722 páginas
- [60] Sitio Oficial. Formación Master en interacción ordenador-persona. <http://griho2.udl.es/aipo/> (03/04/10 10:00 am).
- [61] Esteban Leyva Cortés, Ma. Luisa Garzón Villar, José Prieto Tinoco, **Informática. Temario A. Volumen Iv**, MAD-Eduforma,2003, 182 páginas.
- [62] Mamone Salvatore, **IEEE Standard for Software Maintenance**, ACM SIGSOFT Software Engineering Notes, 1994.
- [63] Lewis Clayton, Rieman John, **Task-Centered User Interface Design**, University of Colorado, 1993, 170 páginas.
- [64] Andres Gomez De Silva Garza, Ania Briseno Ignacio De Jesus, **Introducción a la Ciencia de la computación**, Cengage Learning Editores, 2008.
- [65] Lewis G., **What is Software Engineering?**, DataPro (4015), 1994, pág 10.

- 
- [66] Sitio Oficial. Manifiesto Ágil. <http://www.agilemanifesto.org/> (12/09/2010 23:31 pm).
- [67] Alexander C., **The timeless way of Building**, The Oxford University Press, 1979, 552 páginas.
- [68] Metodología de diseño rápido de aplicaciones vs. Metodología de diseño tradicional. <http://members.fortunecity.com/miadinellie/TrabajoFinal.htm> (26/04/11 13:52 pm).
- [69] Ann L. Winblad, **Software orientado a objetos**, Ediciones Díaz de Santos, 1993, 338 páginas.
- [70] Francisco Ignacio Revuelta Domínguez, Lourdes Pérez Sánchez, **Interactividad en los entornos de formación on-line**, Editorial UOC, 2009, 179 páginas.
- [71] Javier Tuya, Isabel Ramos Román, Javier Dolado Cosín, **Técnicas cuantitativas para la gestión en la ingeniería del software**, Netbiblo, 2007, 373 páginas.
- [72] Thomas Boutell, **CGI programming in C & Perl**, Addison-Wesley Professional, 1996, 402 páginas.
- [72] José Antonio Mañas, **Mundo IP**, Ediciones Nowtilus S.L., 2004, 367 páginas.
- [73] Sitio oficial del COMIPEMS. <http://www.comipems.org.mx/> (008/08/2011 17:04 pm)