

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**“SISTEMA DE ADMINISTRACIÓN DE INSCRIPCIONES AL
PROGRAMA DE INGLÉS DE LA FACULTAD DE MEDICINA”**

INFORME DE ACTIVIDADES PROFESIONALES
QUE PARA OBTENER EL TÍTULO DE: INGENIERO EN COMPUTACIÓN

PRESENTA:

ESTEBAN ARRANGÓIZ ARECHAVALA

ASESOR: ING. DAVID LIMÓN CRUZ

MÉXICO, D.F., CIUDAD UNIVERSITARIA, AGOSTO DE 2010

AGRADECIMIENTOS

A mi Dios, creador de la cielos y la tierra, el mar, y todo lo que en ellos hay; quien es el motivo de mi existir y el que le ha dado un propósito a mi vida.

A mi esposa Mariadna, a quien amo y con quien he vivido y aprendido tantas cosas.

A mis padres Manuel y Verónica a quienes yo respeto, admiro y les agradezco infinitamente el amor con el que me criaron y siguen teniendo para conmigo.

A mis hermanos Pablo y Alejandro, quienes estimo enormemente, quienes son y serán mis verdaderos amigos.

A mi hermano del alma, Kike, quien ha sido y es una bendición en mi vida.

A mis amigos y compañeros de trabajo.

ÍNDICE

CAPÍTULO 1: OBJETIVO.....	1
CAPÍTULO 2: ANTECEDENTES DEL TEMA.....	2
2.1 PROGRAMA DE INGLÉS DE LA FACULTAD DE MEDICINA.....	2
2.2 MARCO TEÓRICO.....	4
2.2.1 Software libre.....	4
2.2.2 Lenguajes de programación.....	6
2.2.3 PHP.....	8
2.2.4 Sistema Manejador de Bases de Datos.....	10
2.2.5 Bases de datos relacionales.....	11
2.2.6 MySQL.....	13
2.2.7 Ingeniería de software.....	14
2.2.8 Método incremental.....	15
CAPÍTULO 3: DEFINICIÓN DEL PROBLEMA.....	17
CAPÍTULO 4: ANÁLISIS Y METODOLOGÍA EMPLEADA.....	19
4.1 SOLICITUD DE INSCRIPCIÓN.....	20
4.2 DISEÑO DE LA BASE DE DATOS.....	20
4.3 PÁGINA DEL PROGRAMA DE INGLÉS.....	22
4.4 PÁGINA DE ADMINISTRACIÓN.....	30
CAPÍTULO 5: PARTICIPACIÓN PROFESIONAL.....	38
CAPÍTULO 6: RESULTADOS Y APORTACIONES.....	40
CAPÍTULO 7: CONCLUSIONES.....	42
BIBLIOGRAFÍA.....	43
GLOSARIO.....	44

CAPÍTULO 1: OBJETIVO

Diseñar e implementar un sistema que administre las inscripciones de alumnos al Programa de Inglés de la Facultad de Medicina de la UNAM. Este sistema debe cubrir las siguientes necesidades:

- Solicitud de inscripción de alumnos en la cual se obtengan sus datos personales y el horario al cual se quieren inscribir.
- Autorización de la inscripción del alumno mediante el registro del comprobante de pago.
- Asignación automática del alumno a un grupo en el momento de la autorización de la inscripción si existen grupos con vacantes, de otra manera se debe crear automáticamente un grupo nuevo.
- Limitar las vacantes de los grupos a 30 en la creación automática de grupos.
- Creación de grupos nuevos en el momento que se desee.
- Asignación de profesores y aulas a los grupos creados.
- Cambios de grupo y de nivel para los alumnos inscritos.
- Obtención de listas de grupos.
- Obtención de reportes de pagos.
- Obtención de lista de alumnos inscritos al curso.
- Cambio de datos de los alumnos.
- Dar de baja del curso a alumnos.
- Sección para los profesores en la que llenen la lista de calificaciones y asistencias (únicamente de los grupos a los que estén asignados).
- Obtención de listas de calificaciones con promedios y estadísticas.
- Publicación de calificaciones para los alumnos.

CAPÍTULO 2: ANTECEDENTES DEL TEMA

2.1 PROGRAMA DE INGLÉS DE LA FACULTAD DE MEDICINA

En la Facultad de Medicina de la UNAM, uno de los requisitos de egreso para los alumnos es aprobar el examen de comprensión de lectura de inglés técnico médico. Para esto tienen la opción de presentar un examen o cursar tres niveles de inglés en el Programa de Inglés de la Facultad. He aquí la importancia de este programa y en consecuencia la administración de las inscripciones y su confiabilidad juegan un papel importante.

El Programa de Inglés pertenece a la Coordinación de Planes y Programas de Estudio, que a su vez pertenece a la Secretaría General de la Facultad de Medicina de la UNAM. El programa lo integran: la Jefa del Programa de Inglés, 2 secretarias y 13 profesores (número que puede variar dependiendo del número de grupos que se abran en el curso).

Las secretarias se encargan de realizar los procesos de inscripción y las operaciones que todo esto conlleva. Los exámenes parciales y finales son elaborados y revisados por un grupo de profesores y la jefa del programa. Estos exámenes se aplican a la mitad del curso y al final, de manera simultánea para todos los alumnos.

El programa consta de tres niveles de 60 horas cada uno, en total 180 horas. Se imparte solamente comprensión de lectura de textos médicos y está dirigido a personas relacionadas con la salud como son: estudiantes de medicina, médicos, enfermeras(os), odontólogos, etc.

El nivel de inglés al que se asigna un alumno depende del resultado de su examen de colocación. Para los alumnos de primer ingreso se toma el examen que presentan junto con el examen diagnóstico como su examen de colocación. Quienes no hayan presentado ese examen o si lo presentaron en un lapso mayor a dos años tienen que presentar un nuevo examen de colocación. Debido a esto los alumnos de nivel avanzado podrían solamente cursar el tercer nivel y al acreditarlo habrán cumplido con el requisito de titulación.

Los pasos que debe realizar el alumno para inscribirse a este programa (hasta antes de la implementación de este trabajo) son los siguientes:

1. Llenar una hoja de solicitud de inscripción con sus datos personales, información de estudios realizados, procedencia y horario preferido.
2. Pagar la cuota correspondiente al curso de inglés en las cajas de la Facultad de Medicina.
3. Entregar la hoja de solicitud de inscripción en la coordinación de inglés, presentando el comprobante de pago.

Las hojas de solicitudes recibidas en la coordinación se ordenan por horario y nivel, para así crear los grupos con un máximo de treinta alumnos por cada grupo (haciendo físicamente la agrupación de estas hojas para representar a los grupos). A cada grupo se le asigna un profesor y un aula. Además, los datos de cada alumno son capturados por las secretarías en una base de datos alojada en una computadora de la coordinación.

Durante el curso se realizan cambios en la inscripción, ya sea de datos personales de los alumnos, grupo, nivel o incluso dar de baja al alumno. También se lleva registro de la asistencia a clases y las calificaciones de los dos exámenes: uno parcial a medio semestre y el final al término del curso.

Para que los alumnos conozcan su calificación tienen que presentarse con su profesor o en su defecto ir a la coordinación para que ahí les informen. Al término de los cursos se les entrega una constancia de curso a todos los alumnos que aprobaron y cuando aprueban el tercer nivel se les entrega una constancia en la cual se informa que han acreditado la comprensión de lectura en inglés. Con esa constancia los alumnos de la Facultad de Medicina acreditan el idioma para poderse titular como médicos cirujanos.

2.2 MARCO TEÓRICO

2.2.1 Software libre

Uno de los principales promotores del software libre, Richard Stallman, tiene un lema que se entiende mejor en su idioma original:

*"Free software is a matter of liberty, not price. To understand the concept, you should think of free as in "free speech", not as in "free beer"."*¹

El texto anterior se puede traducir como sigue: *el "software libre" es un asunto de libertad, no de precio. Para entender el concepto, debe pensarse en "libre" como en "libertad de expresión", no como en "cerveza gratis"*. Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- ◆ Libertad de usar el programa, con cualquier propósito (libertad 0).
- ◆ Libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- ◆ Libertad de distribuir copias, con lo que puede ayudar a otros (libertad 2).
- ◆ Libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3). De igual forma que la libertad 1 el acceso al código fuente es un requisito previo.

Cualquier software que cumpla con estas libertades puede ser nombrado software libre, así los usuarios u organizaciones de éste pueden distribuir las copias con o sin modificaciones.

No debe confundirse software libre con "software de dominio público". Éste último es aquel software que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original.

¹ Williams, S. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software*

Ventajas de utilizar el software libre:

- ✓ Existen aplicaciones para todas las plataformas (Linux, Windows, Mac Os).
- ✓ El precio de las aplicaciones es mucho menor, la mayoría de las veces son gratuitas.
- ✓ Libertad de copia.
- ✓ Libertad de modificación y mejora.
- ✓ Libertad de uso con cualquier fin.
- ✓ Libertad de redistribución.
- ✓ Facilidad a la hora de traducir una aplicación en varios idiomas.
- ✓ Mayor seguridad y fiabilidad.
- ✓ El usuario no depende del autor del software.

Desventajas:

- x Algunas aplicaciones (de Linux) pueden llegar a ser algo complicadas de instalar.
- x Inexistencia de garantía por parte del autor.
- x Interfaces gráficas menos amigables.
- x Poca estabilidad y flexibilidad en el campo de multimedia y juegos.
- x Menor compatibilidad con el hardware.

SOFTWARE COMERCIAL

Ventajas

- ✓ Facilidad de adquisición (puede venir preinstalado con la compra del equipo, o encontrarlo fácilmente en las tiendas).
- ✓ Existencia de programas diseñados específicamente para desarrollar una tarea.
- ✓ Las empresas que desarrollan este tipo de software son por lo general grandes y pueden dedicar muchos recursos, sobre todo económicos, en el desarrollo e investigación.
- ✓ Interfases gráficas mejor diseñadas.
- ✓ Más compatibilidad en el terreno de multimedia y juegos.
- ✓ Mayor compatibilidad con el hardware.

Inconvenientes del software comercial:

- x No existen aplicaciones para todas las plataformas (Windows y Mac OS).
- x El usuario que adquiere software propietario depende totalmente de la empresa propietaria para el mantenimiento.
- x Imposibilidad de modificación.
- x Restricciones en el uso (especificadas en la licencia).
- x Imposibilidad de redistribución.
- x Imposibilidad de copia.

2.2.2 Lenguajes de programación

Los lenguajes utilizados para escribir programas de computadoras que puedan ser entendidos por ellas se denominan lenguajes de programación.

Los lenguajes de programación se clasifican en tres grandes categorías: *máquina*, *bajo nivel* y *alto nivel*.

Los **lenguajes máquina** son aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que el CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje máquina se expresan en términos de la unidad de memoria más pequeña, el *bit* (dígito binario 0, o bien 1), en esencia una secuencia de bits que especifican la operación y las celdas de memoria implicadas en una operación. Estas instrucciones serán fáciles de leer por la computadora y difíciles por un programador, y viceversa. Esta razón hace difícil escribir programas en código o lenguaje máquina y requiere buscar otro lenguaje para comunicarse con la computadora, pero que sea más fácil de escribir y leer por el programador. Para evitar la tediosa tarea de escribir programas en lenguaje máquina se han diseñado otros lenguajes de programación que facilitan la escritura y posterior ejecución de los programas. Estos lenguajes son los de bajo y alto nivel.

Los **lenguajes de bajo nivel** son generalmente dependientes de la máquina, es decir, dependen de un conjunto de instrucciones específicas de la computadora. Un

lenguaje típico de bajo nivel es el lenguaje ensamblador. En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como nemotécnicos (abreviaturas de palabras inglesas o españolas). Estas palabras nemotécnicas son mucho más fáciles de recordar que las secuencias de dígitos 0 y 1.

Después de que un programa ha sido escrito en lenguaje ensamblador se necesita un programa (llamado ensamblador) que lo traduzca a código máquina.

Los **lenguajes de programación de alto nivel** son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos (en general lenguaje inglés), lo que facilita la escritura y la fácil comprensión por el programador.

Los lenguajes de programación son generalmente "transportables". Esto significa que un programa escrito en un lenguaje de alto nivel se puede escribir con poca o ninguna modificación en diferentes tipos de computadoras. Otra propiedad de estos lenguajes es que son independientes de la máquina, esto es, las sentencias del programa no dependen del diseño hardware de una computadora específica.

Los programas escritos en lenguaje de alto nivel no son entendibles directamente por la máquina, necesitan ser traducidos a instrucciones en lenguaje máquina que entiendan las computadoras. Los programas que realizan esta traducción se llaman compiladores, y los programas escritos en un lenguaje de alto nivel se llaman programas fuente. El compilador traduce el programa fuente en un programa llamado programa objeto. Este programa objeto se utiliza en la fase de ejecución del programa. El proceso de traducción de un programa fuente se denomina compilación y tras la fase de enlace se obtiene un programa ejecutable directamente por la computadora.

Algunos ejemplos de lenguajes de programación son: C, Fortran, Java, PHP, Pascal, Visual Basic, Python, Lisp, SQL, Prolog, Html, etc.

2.2.3 PHP

PHP es el acrónimo de *Hypertext Preprocessor* (*preprocesador de hipertexto*). Es un lenguaje de programación de guiones (o *script*). Originalmente se trataba de un conjunto de "macros" concebidas para ayudar en el mantenimiento de páginas web. Desde entonces, sus características han ido creciendo hasta convertirse en un lenguaje de programación completo, capaz de manejar entornos que integran grandes bases de datos. Su popularidad se basa, en gran parte, a su sintaxis similar a la del lenguaje de programación C, su rapidez y simplicidad.

Un lenguaje del lado del servidor, como lo es PHP, es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución del PHP, es por esto que es compatible con todos los navegadores.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de software libre, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5.3.2 aunque en nuestro servidor la versión que utilizamos es la 5.2.4.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.

Algunas de las ventajas de PHP son:

- ✓ Es un lenguaje multiplataforma.
- ✓ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- ✓ Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.

- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Capacidad de expandir su potencial utilizando una enorme cantidad de módulos llamados extensiones.
- ✓ Posee una amplia documentación en su página oficial¹, en la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos.
- ✓ Biblioteca nativa de funciones sumamente amplia e incluida.
- ✓ No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- ✓ Tiene manejo de excepciones (a partir de PHP5).
- ✓ Generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz en servidores que utilicen Apache como servidor web.
- ✓ Es fácil de aprender, sobre todo para quienes están acostumbrados a programar en lenguaje C.

Inconvenientes:

- x La ofuscación de código es la única forma de ocultar los archivos fuente.
- x No existe IDE para los programadores.
- x El manejo de errores no es tan sofisticado como en ASP.
- x Existen muchas versiones de PHP con incompatibilidades entre sí.
- x Todo el trabajo lo realiza el servidor y no delega al cliente; por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.

2.2.4 Sistema Manejador de Bases de Datos

El Sistema Manejador de Bases de Datos (conocido también como Gestor de Base de Datos o DBMS por sus siglas en inglés) es la porción más importante del software de un sistema de base de datos. Es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las funciones principales de un DBMS son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que se pueda acceder a los datos rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos.
 - Esto a través de las sentencias en DML al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos.
- Respaldo y recuperación.
 - Consiste en contar con mecanismos implantados que permitan recuperar fácilmente los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia.
 - Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- Seguridad e integridad.
 - Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

En sí, un sistema manejador de base de datos es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

2.2.5 Bases de datos relacionales

El modelo relacional fue propuesto originariamente por E.F. Codd en un ya famoso artículo de 1970. Gracias a su coherencia y facilidad de uso, el modelo se ha convertido desde los años 80 en el más usado para la producción de DBMS.

Codd propuso que los sistemas de bases de datos deberían presentarse a los usuarios con una visión de los datos organizados en estructuras llamadas relaciones, definidas como conjuntos de tuplas (filas) y no como series o secuencias de objetos, con lo que el orden no es importante. Por tanto, detrás de una relación puede haber cualquier estructura de datos compleja que permita una respuesta rápida a una variedad de consultas. Codd hizo entonces énfasis en que el usuario de un sistema relacional sólo debía preocuparse por el qué consultar y no el cómo de las estructuras de almacenamiento.

La estructura fundamental del modelo relacional es precisamente esa, "relación", es decir una tabla bidimensional constituida por filas (tuplas) y columnas (atributos). Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad.

Las tuplas en una relación son un conjunto en el sentido matemático del término, es decir una colección no ordenada de elementos diferentes. Para distinguir una tupla de otra, se recurre al concepto de "llave primaria", o sea un atributo o conjunto de atributos que permiten identificar unívocamente una tupla en una relación. Naturalmente, en una relación puede haber más combinaciones de atributos que permitan identificar unívocamente una tupla ("llaves candidatas"), pero entre éstas se elegirá una sola para utilizar como llave primaria. Los atributos de la llave primaria no pueden asumir el valor nulo (que significa un valor no determinado), en tanto que ya no permitirían identificar una tupla concreta en una relación. Esta propiedad de las relaciones y de sus llaves primarias se conoce como integridad de las entidades.

Cada atributo de una relación se caracteriza por un nombre y por un dominio. El dominio indica qué valores pueden ser asumidos por una columna de la relación. El motor de datos se ocupará de controlar que en los atributos de las relaciones se incluyan

sólo los valores permitidos por sus dominios. Característica fundamental de los dominios de una base de datos relacional es que sean "atómicos"; es decir, que los valores contenidos en los atributos no se puedan separar en valores de dominios más simples. Más formalmente se dice que no es posible tener atributos con valores múltiples (multivaluados).

La normalización, o sea la razón y uso de las formas normales, es evitar la repetición innecesaria de datos (redundancia). Una solución a este problema es repartirlos en varias relaciones y utilizar referencias por valor entre ellas. Este procedimiento ahorra espacio de almacenamiento, optimiza el rendimiento y, al eliminar la redundancia, impide modificaciones parciales o incompletas que podrían dar lugar a inconsistencias. Existen hasta seis formas normales pero, en la práctica, se adopta generalmente la tercera forma normal.

Junto con el modelo, el Dr. Codd también propuso el álgebra relacional, un lenguaje formal con una serie de operadores que trabajan sobre una o varias relaciones para obtener otra relación resultado, sin que cambien las relaciones originales. Tanto los operandos como los resultados son relaciones, por lo que la salida de una operación puede ser la entrada de otra operación. Esto permite anidar expresiones del álgebra del mismo modo que se pueden anidar las expresiones aritméticas. Codd originalmente propuso ocho operandos pero sólo cinco son fundamentales: *restricción*, *proyección*, *producto cartesiano*, *unión* y *diferencia*, que permiten realizar la mayoría de las operaciones de obtención de datos. Los operadores no fundamentales son la *concatenación (join)*, la *intersección* y la *división*, que se pueden expresar a partir de los cinco operadores fundamentales. La restricción y la proyección son operaciones unarias porque operan sobre una sola relación. El resto de las operaciones son binarias porque trabajan sobre pares de relaciones.

2.2.6 MySQL

MySQL es el Sistema Manejador de Bases de Datos de software libre más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad están haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle.

MySQL es un sistema de administración de bases de datos relacionales que soporta grandes cantidades de datos. Como base de datos relacional, se manejan los datos en tablas separadas en lugar de almacenar todos los datos en un archivo grande, lo que brinda velocidad y flexibilidad. También utiliza el lenguaje ANSI/ISO SQL estándar.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

Es muy destacable que MySQL se distribuye bajo licencia GNU/GPL, lo que hace que, en ciertas circunstancias, su uso sea gratuito e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

Otras características de MySQL son:

- Portabilidad. Está disponible en versiones para diferentes plataformas.
- Puede manejar conexiones en el núcleo de forma concurrente (es multithreaded o multihilos), esto significa que puede utilizar múltiples CPU si están disponibles.
- Provee APIs para clientes de diversos lenguajes de programación: C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y TCL.
- Sistema de direccionamiento de memoria rápido (basado en hilos).
- Consultas tipo join de alta velocidad utilizando el multi-join de una barrida optimizado.
- Uso de tablas temporales en memoria.

2.2.7 Ingeniería de software

Según la definición del IEEE: *"software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo"*². Según el mismo autor, *"un producto de software es un producto diseñado para un usuario"*. En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, que en palabras más cotidianas, se considera que *"la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software"*, es decir, *"permite elaborar consistentemente productos correctos, utilizables y costo-efectivos"*³.

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

Una metodología de desarrollo de software se refiere a un sistema que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad. Estos modelos se denominan frecuentemente paradigmas de la ingeniería del software y la elección de un paradigma se realiza básicamente de acuerdo al tipo del proyecto y de la aplicación, los controles y las entregas a realizar.

² Lewis G. 1994. "What is Software Engineering?" DataPro (4015). Feb 1994. pp. 1-10.

³ Cota A. 1994 "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994. pp. 5-13.

2.2.8 Método incremental

El Desarrollo iterativo y creciente (o incremental) es un proceso de desarrollo de software, creado en respuesta a las debilidades del modelo tradicional de cascada. La idea principal detrás de mejoramiento iterativo es desarrollar un sistema de programas de manera incremental, permitiendo al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema. Los pasos claves en el proceso son comenzar con una implementación simple de los requerimientos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

El proceso en sí mismo consiste de:

- Etapa de inicialización
- Etapa de iteración
- Lista de control de proyecto

En la etapa de inicialización se crea una versión del sistema. La meta de esta etapa es crear un producto con el que el usuario pueda interactuar, y por ende retroalimentar el proceso. Debe ofrecer una muestra de los aspectos claves del problema y proveer una solución lo suficientemente simple para ser comprendida e implementada fácilmente. Para guiar el proceso de iteración, una lista de control de proyecto se crea, y esta lista contiene un historial de todas las tareas que necesitan ser realizadas. Incluye cosas como nuevas funcionalidades para ser implementadas, y áreas de rediseño de la solución ya existente. Esta lista de control se revisa periódica y constantemente como resultado de la fase de análisis.

La etapa de iteración involucra el rediseño e implementación de una tarea de la lista de control de proyecto, y el análisis de la versión más reciente del sistema. La meta del diseño e implementación de cualquier iteración es ser simple, directa y modular, para poder soportar el rediseño de la etapa o como una tarea añadida a la lista de control de proyecto. El análisis de una iteración se basa en la retroalimentación del usuario y en el análisis de las funcionalidades disponibles del programa. Involucra el análisis de la estructura, modularidad, usabilidad, confiabilidad, eficiencia y eficacia (alcanzar las

metas). La lista de control del proyecto se modifica bajo la luz de los resultados del análisis.

Las guías primarias que dirigen la implementación y el análisis incluyen:

- Cualquier dificultad en el diseño, codificación y prueba de una modificación debería apuntar a la necesidad de rediseñar o volver a programar.
- Las modificaciones deben ajustarse fácilmente a los módulos fáciles de encontrar y a los aislados. Si no es así, entonces se requiere algún grado de rediseño.
- Las modificaciones a las tablas deben ser especialmente fáciles de realizar. Si dicha modificación no ocurre rápidamente, se debe aplicar algo de rediseño.
- Las modificaciones deben ser más fáciles de hacer conforme avanzan las iteraciones. Si no es así, hay un problema primordial usualmente encontrado en un diseño débil o en la proliferación excesiva de parches al sistema.
- Los parches normalmente deben permanecer solo por una o dos iteraciones. Se hacen necesarios para evitar el rediseño durante una fase de implementación.
- La implementación existente debe ser analizada frecuentemente para determinar que tan bien se ajusta a las metas del proyecto.
- Las facilidades para analizar el programa deben ser utilizadas cada vez para ayudar en el análisis de implementaciones parciales.
- La opinión del usuario debe ser solicitada y analizada para indicar deficiencias en la implementación referida por él.

CAPÍTULO 3: DEFINICIÓN DEL PROBLEMA

Hasta antes del año 2010 la inscripción de los alumnos al Programa de Inglés se realizaba siguiendo el procedimiento que fue señalado en el capítulo 3.1 de manera manual, esto es, llenando una solicitud con tinta o lápiz por cada uno de los alumnos para que después fuera vaciada esta información en una base de datos por las secretarías. Este proceso, además de ser muy lento y poco eficiente, acarreaba una gran cantidad de errores en los datos, ya que tenían que capturar los datos de los alumnos uno por uno y además se hacía dos veces el mismo trabajo de llenado de datos, primero por el alumno y después por las secretarías.

Por otro lado, para realizar su inscripción los alumnos tienen la libertad de hacerlo hasta el último viernes antes de que inicien las clases, lo cual reduce mucho el tiempo para preparar el inicio del curso por parte de la Coordinación, tanto para realizar todas las inscripciones de los alumnos como para crear y organizar los grupos asignando horarios, aulas y profesores.

Debido a estos factores la carga de trabajo en el inicio de cada curso sobrepasaba la capacidad de la gente que labora en el programa y no se tenía la información en el momento requerido ni con la veracidad necesaria. Esto tuvo un punto cumbre en el segundo semestre del año 2009, en el cual cerca de 700 alumnos quisieron inscribirse al curso. Como es común, muchos alumnos realizaron su inscripción los últimos días que tenían para hacerlo, esto es, la semana anterior al inicio del curso. Los primeros días del curso fueron muy problemáticos, no se tenían bien definidos los grupos, los alumnos no sabían con qué profesor tomarían clases ni en qué aula y tampoco se había capturado los datos de los alumnos. Para arrancar las clases se entregaron las mismas hojas de inscripción que llenaron los alumnos, en montones de aproximadamente 30 hojas, a los profesores para que los alumnos supieran con quién tomarían las clases y que los profesores pudieran tomar asistencia de alguna manera.

Conforme pasaban los días se llenaba la base de datos de alumnos inscritos y se hacían las operaciones que se requerían, como cambios de grupo para los alumnos, creación de listas de grupos, etc. Se tuvieron que hacer muchas correcciones y cuando

se requería tener cierta información fue muy difícil encontrarla, pero finalmente, después de muchos días de trabajo, se terminó de regularizar la información y el curso continuó y terminó de manera normal.

Ante esta situación la coordinadora de inglés, la Licenciada Silvia Hambleton, solicitó ayuda al Departamento de Informática Médica para que se le propusiera una alternativa al proceso de inscripción. Fue entonces que el Doctor Antonio Cerritos, jefe del Departamento de Informática Médica, asignó este proyecto al Laboratorio de Visualización en 3D y a su vez el Ingeniero David Limón Cruz lo asignó en concreto a mí.

El problema consiste en proponer un nuevo proceso de inscripción al Programa de Inglés de la Facultad de Medicina con el que se simplifique el trabajo relacionado con las inscripciones de los alumnos y además minimice los errores. Para este proceso se debe considerar que buscarán inscribirse más de 500 alumnos cada semestre a cualquiera de los nueve horarios que se ofrecen las clases de cada uno de los tres niveles de inglés. Además, la nueva propuesta debe abarcar las operaciones requeridas después de haber iniciado el curso, como son: cambios de datos, bajas de alumnos, asignación de aulas y profesores, creación de listas de alumnos y reportes de pagos recibidos, captura y publicación de calificaciones, entre otras. Esto es, crear todo un sistema de administración del Programa de Inglés de la Facultad de Medicina.

CAPÍTULO 4: ANÁLISIS Y METODOLOGÍA EMPLEADA

El desarrollo de este proyecto fue siguiendo la metodología incremental. En un principio no se tenían todos los requerimientos ni tampoco todas las necesidades. Los alcances del proyecto fueron creciendo conforme se fue viendo el avance de éste.

En un inicio se buscaba solamente hacer un cambio en el proceso de solicitud de inscripción de los alumnos, para hacerlo más rápido y con menos complicaciones para el personal del Programa de Inglés. Al tener listo y funcionando se ampliaron los requisitos y se buscó continuar ese proceso para poder primero autorizar las inscripciones y después tener los registros y datos de los alumnos ya inscritos. Después de esta etapa volvieron a cambiar las peticiones para tener un sistema completo de administración de inscripciones de los alumnos.

Es por esto que el proyecto se realizó siguiendo el método incremental descrito en el capítulo 2.2.8. El proyecto fue creciendo en cuanto a alcances y requisitos. Las versiones del sistema se fueron desarrollando con una frecuente retroalimentación del cliente (jefa del Programa de Inglés y en cierta medida las secretarías), quienes mientras utilizaban el sistema mencionaban puntos específicos que son requeridos para los controles que llevan administrativamente y funcionalmente, como pueden ser los reportes de alumnos inscritos, reportes de ingresos por concepto de inscripción al Programa de Inglés, proceso de publicación de calificaciones, etc.

Esta manera de trabajar me resultó muy cómoda y también propició que se llegara a un sistema en el cual el cliente se sintió satisfecho y familiarizado con el producto, ya que lo fue evaluando y utilizando durante todo el desarrollo.

4.1 SOLICITUD DE INSCRIPCIÓN

Para este problema observé primero que el primer paso en el proceso de solicitud de inscripción de los alumnos, que implica el llenado de una hoja con sus datos y después vaciar esa la información en la base de datos por medio de secretarías, es muy poco adecuado, lento y propenso a cometer errores o pérdida de información. Todo este trabajo se puede simplificar si se capturan correcta y completamente los datos necesarios de cada alumno desde el llenado de la solicitud de inscripción. Por eso propuse que esta captura de información se realizara por medio de un formulario en una página de internet, el cual debe tener los filtros y candados necesarios para asegurar que se cuente con toda la información requerida. De esta manera también se beneficia a los alumnos ya que tienen mayor libertad para iniciar su proceso de inscripción a la hora y el lugar que más les convenga y sin tener que estar sujetos a disponibilidad del personal del Programa de Inglés. Una vez que los alumnos terminen de llenar correctamente el formulario, la información será registrada en la base de datos.

4.2 DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos fue siguiendo el modelo relacional. Para iniciar identifiqué las entidades que conforman los datos que se van a manejar. La primera entidad es el curso, el cual contiene principalmente las fechas que marcan los periodos dentro del curso como es el inicio de clases. Ahora bien, no podemos tener un curso por sí solo, necesitamos alumnos, profesores y aulas con sus respectivos datos. Al recordar el proceso de inscripción de los alumnos saltan a la vista las entidades: solicitudes de inscripción, pagos y bajas. Pero también cada alumno puede estar inscrito en más de una ocasión, así que también consideramos la inscripción como una entidad. Por otro lado, el personal que manejará el sistema de administración necesita estar registrado para tener acceso, por lo cual necesitamos otra entidad: usuarios. Por último, los horarios en los que se pueden dar clases son establecidos por la entidad horarios.

De esta manera tenemos diez entidades: curso, alumno, solicitud, grupo, horario, inscripción, baja, pago, profesor y usuario. Cada una de estas entidades tiene diferentes

atributos y representan las tablas que se manejan en MySQL.

El esquema de la base de datos queda de la siguiente manera:

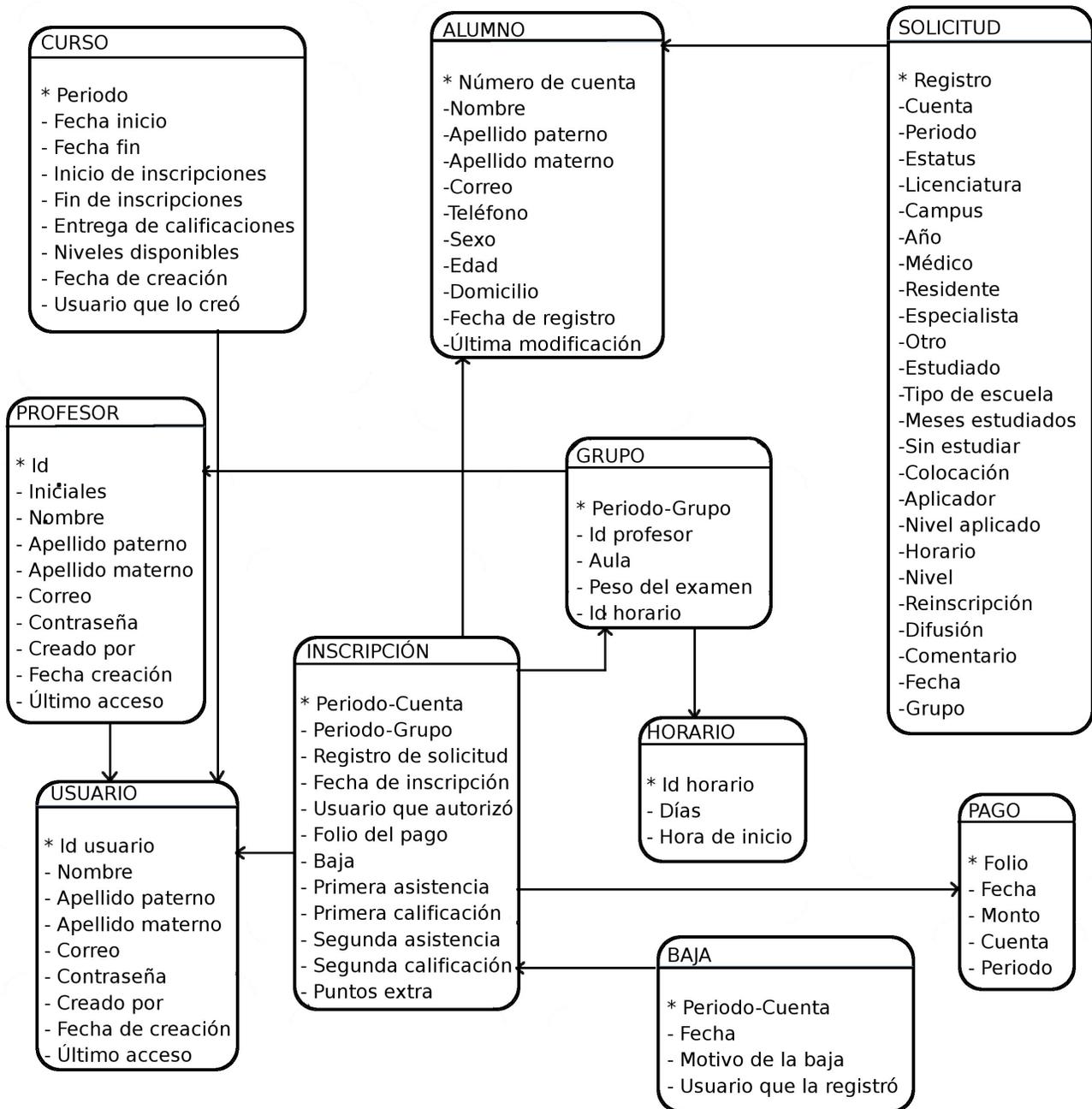


Fig. 4.1 Esquema de la base de datos.

4.3 PÁGINA DEL PROGRAMA DE INGLÉS

Como el Programa de Inglés no contaba con una página en internet con la cual se difundiera la información de los cursos se creó la página del Programa de Inglés (http://lab3d.facmed.unam.mx/programa_ingles/).



Fig. 4.2 Página principal del portal de internet creado para el Programa de Inglés.

Esta página fue creada por medio del sistema de gestión de contenido: Drupal versión 6.15. Toda la información que se quiere difundir se inserta como páginas en esta instalación de Drupal, pero la parte del formulario se programó por separado en archivos de PHP.

El formato de solicitud de inscripción se implementó en nueve páginas consecutivas, utilizando un formulario de HTML y programación en PHP. Las preguntas se elaboraron con base en el formato impreso que se utilizaba anteriormente para recabar la información de los alumnos, eliminando solamente los datos que ya no se requerían en el Programa de Inglés y conservando todos aquellos que consideraron necesarios. Cada uno de estos pasos contiene filtros o candados que impiden que continúe el alumno

(mandando un aviso en la página) si la información que está insertando no está completa o se detecta algún error. A continuación se describen estos pasos y el código que revisa que la información sea correcta y completa en cada paso:

1. Iniciar la solicitud de inscripción.

En la página se muestra un botón para iniciar la inscripción solamente si se encuentra dentro de las fechas especificadas por la Jefa del Programa. De lo contrario se muestra un aviso de que las inscripciones están cerradas sin la posibilidad de iniciar el proceso.

```
<?php
    $fechaActual = date("YmdHi");
    $fechaInicio = substr($cursoActual['FechaInicioInscripciones'],0,4)*100000000 +
        substr($cursoActual['FechaInicioInscripciones'],5,2)*1000000 +
        substr($cursoActual['FechaInicioInscripciones'],8,2)*10000 + 0;
    $fechaFin = substr($cursoActual['FechaFinInscripciones'],0,4)*100000000 +
        substr($cursoActual['FechaFinInscripciones'],5,2)*1000000 +
        substr($cursoActual['FechaFinInscripciones'],8,2)*10000 + 2359;
    if( ($fechaInicio<$fechaActual) and ($fechaFin>$fechaActual) )
    { /* Aquí se presenta el botón para iniciar el proceso y la información de lo que se
        deben hacer los alumnos */ }
    else
        echo 'Las inscripciones al curso '.$cursoActual['Periodo'].' se han cerrado.';
?>
```

2. Número de cuenta.

El número de cuenta se captura como un número entero, es decir, sin espacios guiones ni otro carácter que no sea número. Si es una persona que no está inscrita en la UNAM y no tiene número de cuenta, se da la opción de que por medio de su fecha de nacimiento se le asigne un número que le servirá como número de cuenta.

```
<?php
    $cuenta = $_POST['cuenta'];
    if( !filter_var($cuenta, FILTER_VALIDATE_INT) )
        error('<p align="center">El número de cuenta <b>'. $cuenta. '</b> no es
            válido. <br>Debe contener únicamente números.</p>');
    $cuenta = intval($cuenta);
```

```

if( ($cuenta<70000000) or ($cuenta>1000000000) )
    error('<p align="center">El número de cuenta <b>'. $_POST['cuenta'].'</b>
        no es válido. <br>');
?>

```

3. Datos personales

Nombre, apellidos, sexo, edad, domicilio, teléfono y correo electrónico.

Facultad de Medicina

Programa de Inglés
FACULTAD DE MEDICINA UNAM

Número de cuenta: **612345678**

Apellido paterno:

Apellido materno:

Nombre(s):

Teléfono (casa):

Sexo: Masculino
 Femenino

Edad:

Domicilio:

Correo electrónico:

[Regresar a la página anterior](#)
2/8

Fig. 4.3 Captura de datos.

```

<?php
if(!preg_match("#^[-A-Za-zÁ-Úá-ú ]*$#", $_POST['nombre']) or $_POST['nombre']=="")
    error('<p align="center">El nombre <b>'. $_POST['nombre'].'</b> no es
        válido.</p>');
if( !preg_match("#^[-A-Za-zÁ-Úá-ú ]*$#", $_POST['apellidoP']) or
    $_POST['apellidoP']=="" )
    error("<p align=\"center\">El apellido paterno <b>". $_POST['apellidoP'].'</b> no
        es válido. </p>");
if( !preg_match("#^[-A-Za-zÁ-Úá-ú ]*$#", $_POST['apellidoM']) or
    $_POST['apellidoM']=="" )
    error('<p align="center">El apellido materno <b>'. $_POST['apellidoM'].'</b> no

```

```

        es válido. </p>');
$telefono = intval($_POST['telefono']);
if( ($telefono<10000) or ($telefono>10000000000) )
    error('<p align="center">El teléfono <b>'.$_POST['telefono'].'</b> no es
        válido.</p>');
if( $_POST['edad']!="" )
{
    $edad = intval($_POST['edad']);
    if( ($edad<10) )
        error('<p align="center">La edad especificada <b>'.$edad.'</b> no es
            válida. </p>');
}
else
    error('<p align="center">Te faltó escribir tu edad. </p>');

if( $_POST['domicilio']=="" )
    error('<p align="center">Hace falta que escribas tu domicilio. </p>');
if( $_POST['correo']!="" )
{
    if( !filter_var($_POST['correo'], FILTER_VALIDATE_EMAIL) )
        error('<p align="center">El correo electrónico <b>'.$_POST['correo'].'
            "</b> no es válido. <br></p>');
}
else
    error('<p align="center">Te faltó escribir tu correo electrónico.
        </p>');
?>

```

4. Seleccionar estatus académico/laboral

Se selecciona una opción de las siguientes:

- Alumno de licenciatura
- Médico sin estudios de posgrado
- Alumno residente de maestría
- Médico especialista maestro, doctor
- Otro

```

<?php
if( $_POST['estatus']=="" )
    error("<p align=\"center\">Debes especificar una opción. <br></p>");
?>

```

5. Especificar estatus académico

Aquí se especifica el estatus académico dependiendo de la opción elegida en el paso anterior; por ejemplo, si es un alumno de licenciatura se debe escribir la carrera, campus y año o si es un médico especialista se debe especificar su especialidad.

```
<?php
switch( $_POST['estatus'] )
{
    case 'alumno':
        if( $_POST['licenciatura']=="" )
            error("Debes especificar la licenciatura.");
        if( $_POST['campus']=="" )
            error("Debes especificar el campus.");
        if( $_POST['ano']=="" )
            error("Debes especificar el año.");
        break;
    case 'medico':
        if( $_POST['medico']=="" )
            error("Debes especificar el nombre de tu carrera.");
        break;
    case 'residente':
        if( $_POST['residente']=="" )
            error("Debes especificar el estatus académico/laboral de residente.");
        break;
    case 'especialista':
        if( $_POST['especialista']=="" )
            error("Debes especificar la especialidad.");
        break;
    case 'otro':
        if( $_POST['otro']=="" )
            error("Debes especificar tu estatus académico/laboral (otro).");
        break;
}
?>
```

6. Antecedentes de inglés

Información sobre el tiempo que ha estudiado inglés y el lapso que ha pasado desde la última vez que lo estudió.

```

<?php
    if( $_POST['estudiado']=="" )
        error("Debes especificar si ya has estudiado ingl&eacute;s.");
    if( $_POST['colocacion']=="" )
        error("Debes especificar si presentaste un examen de colocaci&oacute;n.");
    if( $_POST['estudiado']==1 )
    {
        if( $_POST['tipoEscuela']=="" )
            error("Debes especificar el tipo de escuela en la que estudiaste el
            ingl&eacute;s.");
        if( $_POST['mesesEstudiados']=="" )
            error("Debes especificar los meses que estudiaste el ingl&eacute;s.");
        if( $_POST['sinEstudiar']=="" )
            error("Debes especificar los meses que dejaste de estudiar el ingl&eacute;s.");
    }
    if( $_POST['colocacion']==1 )
    {
        if( $_POST['aplicador']=="" )
            error("Debes especificar qui&eacute;n te aplic&oacute; el examen de
            colocaci&oacute;n.");
    }
?>

```

7. Horario y nivel deseados

Se presentan los horarios disponibles (que no han sido cerrados por la Jefa del Programa) para inscripción al curso.

```

<?php
    if( $_POST['reinscripcion']=="" )
        error('Debes especificar si te est&aacute;s reinscribiendo.');
```

if(VerificaGrupoCerrado(\$periodoActual."-".\$_POST['nivel'].\$_POST['horario']."_"))

```

        error('El horario que elegiste ha sido cancelado para el nivel <b>'
            .$_POST['nivel'].'</b>. <br> Por favor elige otro horario.');
```

?>

8. Medios de difusión

Seleccionar los medios por medio de los cuales se enteró del Programa de Inglés y un espacio para dejar comentarios o sugerencias.

9. Confirmar datos

Se presenta toda la información recibida y se pide su confirmación de que es correcta para que pueda ser registrada en el sistema. Al confirmar se presenta la opción de imprimir el comprobante de solicitud de inscripción.

Confirme sus datos

Revise muy bien sus datos:

Número de cuenta: **612345678**
Nombre: **Arrangóiz Arechavala Esteban**
Periodo al que se inscribe: **2010-1**
Teléfono: **32435**
Sexo: **masculino**
Edad: **27**
Domicilio: **bajada al río 111**
Correo: **esteban@lab3d.facmed.unam.mx**
Estatus académico/laboral: **alumno (Médico Cirujano, campus Ciudad Universitaria, año 1)**
¿Ha estudiado inglés?: **sí**, en escuela **privada**, durante **más de 18** meses.
Dejó de estudiar durante **1-5** meses.
Presentó examen de colocación: **no**
Nivel al que quiere inscribirse: **3**
Horario solicitado: **Martes y jueves, 15:00 - 17:00**
Se enteró por **2** medio(s): **personas internet**

Si los datos son correctos dé click en finalizar

[Regresar a la página anterior](#)

8/8

Fig. 4.3 Resumen de datos del alumno que se van a registrar en la solicitud de inscripción en línea. El registro se crea al momento en que seleccionan "Finalizar".

Cuando el alumno confirma sus datos se envía la información a la base de datos y por un lado se registran los datos personales del alumno (si no estaba registrado) y por otro lado se crea el registro de su solicitud de inscripción mediante el siguiente código:

```
<?php
    $cuenta = $_POST['cuenta'];
```

```

$encuentraSolicitud = VerificaSolicitud($cuenta, $periodoActual);
$encuentraAlumno = VerificaAlumno($cuenta);

if( ($encuentraSolicitud==0) and ($encuentraAlumno==0) )
    RegistraAlumno($cuenta, $_POST['nombre'], $_POST['apellidoP'], $_POST['apellidoM'],
        $_POST['correo'], $_POST['telefono'], $_POST['sexo'], $_POST['edad'],
        $_POST['domicilio']);

if( $encuentraSolicitud )
{
    $solicitud = VerificaSolicitud($cuenta, $periodoActual);
    error('<p>El numero de cuenta ya ha sido registrado en la solicitud de
        inscripciones para el periodo '.$periodoActual.'<br>con el n&uacute;mero de
        registro: <b>'.$solicitud['Registro'].'</b></p> <p>Para cualquier
        aclaraci&oacute;n o cambio acude a la coordinaci&oacute;n de ingl&eacute;s.</p>
        <form method="POST" action="comprobanteSolicitud.php"> <input type="submit"
        name="finalizar" value="Imprimir comprobante"> <input type="hidden"
        name="pagina" value="inscribeAlumno"> <input type="hidden" name="cuenta"
        value="'.$cuenta.'"> <input type="hidden" name="periodo" value="'.
        $periodoActual.'">'.</form>');
}

$registro = RegistraSolicitud($cuenta, $_POST['periodo'], $_POST['estatus'],
    $licenciatura, $campus, $ano, $medico, $residente, $especialista, $otro,
    $_POST['estudiado'], $tipoEscuela, $mesesEstudiados, $sinEstudiar,
    $_POST['colocacion'], $aplicador, $_POST['horario'], $_POST['nivel'],
    $_POST['reinscripcion'], $difusion, $_POST['comentario']);

if( $registro > 0 )
{
?>

<p align="center"> Listo, ha completado el primer paso para su inscripci&acirc;n al programa
de ingl&eacute;s.
    <br>El n&uacute;mero de registro de su solicitud es: <b> <?=$registro?> </b>
    <br>Gu&aacute;rdelo para cualquier aclaraci&oacute;n.
</p>
<p> RECUERDA IMPRIMIR TU COMPROBANTE
    <form method="POST" action="comprobanteSolicitud.php">^M
        <input type="submit" name="finalizar" value="Obtener comprobante">^M
        <input type="hidden" name="pagina" value="inscribeAlumno">
        <input type="hidden" name="cuenta" value="<?=$cuenta?>">
        <input type="hidden" name="periodo" value="<?=$periodoActual?>">
    </form>
    <small>(puedes guardar el archivo si no te es posible imprimir en este
momento)</small>
</p>
<p>
    Siguiete paso: pagar <b>$550</b> en las cajas de la facultad por concepto de
<b>curso de ingl&eacute;s</b>. <br>
    Posteriormente deber&aacute; presentar los <b>comprobantes</b>, tanto de <b>pago</b>
como el de esta <b>solicitud de inscripci&oacute;n</b> en <br>
    la coordinaci&oacute;n de ingl&eacute;s para hacer oficial su inscripci&oacute;n y
que le sea asignado un grupo.
</p>

<?php
}
else
    echo 'No se pudo registrar la solicitud.</p>';
?>

```

4.4 PÁGINA DE ADMINISTRACIÓN

La sección de administración del curso fue implementada en una página por separado, en la cual los usuarios (jefa y secretarias) necesitan identificarse con un nombre de usuario y una contraseña para poder tener acceso. Una vez autenticados pueden hacer movimientos dentro del curso.

La página está organizada en seis secciones, las cuales contienen distintas páginas para realizar operaciones específicas:

1. Alumnos: Autorizar inscripciones, ver y modificar datos, cambios de grupo o nivel, bajas de alumnos.
2. Grupos: Ver los grupos existentes del curso, ver alumnos de cierto grupo, crear un grupo nuevo, borrar un grupo, asignar un profesor o un aula a un grupo.
3. Profesores: Ver profesores registrados, registrar un nuevo profesor, ver o cambiar datos de un profesor
4. Listas: Obtener listas de grupos, listas de calificaciones, lista general de alumnos o reporte de ingresos.
5. Cursos: Abrir un nuevo curso, ver todos los cursos, mostrar detalles de un curso, cambiar especificaciones del curso actual, estadísticas de los cursos.
6. Usuarios: Crear usuario, ver o cambiar datos de un usuario, borrar un usuario, cambiar la contraseña propia.

Para continuar con su inscripción los alumnos deben presentarse en el Programa de Inglés con sus comprobantes. Las secretarias verifican que la información que presentan esté correcta y completa; si todo está en orden proceden a autorizar la inscripción del alumno. Para esto tienen que entrar al sistema y en la sección de alumnos seleccionar "Buscar un alumno". Introducen su número de cuenta y si el alumno hizo una solicitud de inscripción para el curso actual entonces aparece la opción de autorizar la inscripción, especificando también los datos del recibo de pago.

Usuario: **esteban**

Alumnos Grupos Profesores Listas Cursos Usuarios Salir

Inscripciones Buscar un alumno Solicitudes Cambios de grupo o nivel Bajas

Inscripción de alumnos.

Nombre del alumno: **Esteban Arrangóiz Arechavala.**
 Número de cuenta: **401054503**
 Escribir **número de folio** del recibo de pago.

Folio:

Fecha de pago: 28 / mayo / 2010

Monto del pago: \$

Programa de Inglés Facultad de Medicina UNAM Lab3D Página creada por Esteban Arrangóiz

Fig. 4.4 Autorización de la inscripción del alumno, en la página de administración de inscripciones, mediante el registro del comprobante de pago correspondiente (folio).

Una vez que se hayan proporcionado estos datos el sistema busca una vacante para el horario que seleccionó el alumno en la solicitud y lo inscribe al grupo en el que encuentre. Si no hay grupos disponibles y no se ha cerrado ese horario (en la sección de cursos y detalles del curso actual) se crea un nuevo grupo para ese horario y se inscribe al alumno. Se presenta un mensaje de confirmación de la inscripción y se especifica a cual grupo fue inscrito el alumno, con lo que termina el proceso de inscripción para el alumno.

Estas son las funciones involucradas en este proceso:

```
function ConectaDB($db="")
{
    $conexion = @mysql_connect("localhost", "inglesadmin", "mariadnaingles")
    or die("No es posible conectar a la base de datos");
    if($db!="" and !@mysql_select_db($db))
        die("La base de datos no est&aacute; disponible.");
    return $conexion;
}
```

```

function error($msg)
{
    IniciaBloqueRedondeado("", "");
    echo '<font color="#001010">'. $msg. '</font>';
    TerminaBloqueRedondeado("", "");
    echo '<p align="center"><a href="javascript:history.back(1);"><font color="#FFFFFF">
        Regresar</a></p>';
    include ('ingles/pieDePagina.php');
    exit;
}

function EscribeLog($curso, $archivo, $usuario, $msg)
{
    if( !file_exists("/var/log/ingles/$curso") )
    { mkdir("/var/log/ingles/$curso"); }
    $nombreArchivo = "/var/log/ingles/$curso/$archivo.log";
    $manejadorArchivo = fopen($nombreArchivo, 'a', 1) or die('no fue posible abrir el
        archivo "'. $nombreArchivo. '"');
    $texto = date('Y-m-d H:i:s')." - ". $usuario. " - $msg\n";
    fwrite($manejadorArchivo, $texto);
    fclose($manejadorArchivo);
}

function AsignaGrupo( $cuenta, $periodo, $nivel, $horario )
{
    $encontrado = 0;
    ConectaDB('inglesdb');
    $query = "SELECT PeriodoGrupo FROM `grupos` WHERE PeriodoGrupo LIKE '". $periodo. "-".
    $nivel. $horario. "_'";
    $busqueda = mysql_query($query);
    if( !$busqueda )
    {
        error('Ocurri&ocute; un error con la base de datos al buscar los grupos existentes.
            Error 51.<br>El error fue: '.mysql_error(). '<br>Query: '. $query);
    }
    while( $grupos = mysql_fetch_array($busqueda) )
    {
        if( $encontrado==0 )
        {
            $alumnosContados = CuentaAlumnos($grupos['PeriodoGrupo']);
            if( $alumnosContados<30 )
            {
                $encontrado = 1;
                $grupo = $grupos['PeriodoGrupo'];
            }
        }
    }
    if( $encontrado==0 )
    {
        if( !VerificaGrupoCerrado($periodo. "-". $nivel. $horario) )
        {
            $grupo = CreaGrupo($horario, $nivel, $periodo);
            $msg = "Se crea autom&acuticamente el grupo: '". $grupo. "' al inscribir al alumno: '".
                $cuenta. "'";
            EscribeLog($periodoActual, 'operaciones', $_SESSION['usuario'], $msg);
        }
        else
            $grupo = 0;
    }
    mysql_close();
    return $grupo;
}

```

```

function CreaGrupo($horario, $nivel, $periodo)
{
    ConectaDB('inglesdb');
    $resultado = mysql_query("SELECT * FROM `grupos` WHERE PeriodoGrupo LIKE '".
        $periodo."-".$nivel.$horario."_'");
    $gruposAbiertos = mysql_num_rows($resultado);
    $nuevoGrupo = ($nivel*100) + ($horario*10) + $gruposAbiertos + 1;
    $periodoGrupo = $periodo."-".$nuevoGrupo;
    $query = "INSERT INTO `grupos` (PeriodoGrupo) VALUES ( '$periodoGrupo' );";
    if( !mysql_query($query) )
    {
        if( mysql_error() )
        {
            error('Ocurrió un error con la base de datos al crear un nuevo grupo. Error 98.
                <br>Error: '.mysql_error().'</b> <br>El query fue: <b>'.$query.'</b>');
        }
    }
    mysql_close();
    return $periodoGrupo;
}

function VerificaGrupoCerrado( $periodoGrupo )
{
    $cerrado = 0;
    $grupo = substr($periodoGrupo,7,2);
    ConectaDB('inglesdb');
    $query = "SELECT GruposCerrados FROM `cursos` WHERE Periodo =
        '".substr($periodoGrupo,0,6)."'";
    $busqueda = mysql_query($query);
    if( !$busqueda )
        error('Ocurrió un error con la base de datos al buscar grupos cerrados en la
            tabla de cursos. Error 25. <br> <br>Error: '.mysql_error().'</b> <br>El query
            fue: <b>'.$query.'</b>');
    $gruposCerrados = mysql_fetch_array($busqueda, MYSQL_NUM);
    $gruposCerrados = explode( " ", $gruposCerrados[0] );

    if( mysql_num_rows($busqueda) )
    {
        while( list(, $temp) = each($gruposCerrados))
        {
            if( $grupo==$temp)
                $cerrado++;
        }
    }
    else
        $cerrado = 0;
    mysql_close();
    return $cerrado;
}

```

De una manera similar se realizan las operaciones que se requieran. Dentro de cada categoría se encuentran las ligas a las páginas que realizan alguna función en específico. Estas funciones se implementaron de una manera en la que el usuario dé la información mínima necesaria y si es correcta se ejecuta y presenta los resultados obtenidos.

Las secciones y sus funciones se listan a continuación:

1. Alumnos

- Inscripciones

Autorización de inscripción de alumnos. Se requiere que el alumno haya completado correctamente la solicitud de inscripción en línea y que muestre su comprobante de pago con folio.

- Buscar un alumno

Se presentan los datos personales del alumno y el historial de sus inscripciones. También se presenta una liga para ver el comprobante de solicitud de inscripción y para autorizar su inscripción si es que realizó la solicitud de inscripción para el periodo en curso.

- Solicitudes

Se permite hacer cambios en las solicitudes ya realizadas por si tienen errores o se busca hacer un cambio de horario o de nivel antes de que el alumno quede inscrito.

- Cambios de grupo o nivel

Se pide el número de cuenta del alumno y si el alumno está inscrito al periodo actual se presentan las posibles opciones de cambios tanto de grupo como de nivel.

- Bajas

Se da de baja al alumno especificando el motivo.

2. Grupos

- Mostrar grupos

Se listan los grupos abiertos del periodo en curso con los datos de horario, nivel, profesor, aula y alumnos inscritos.

- Ver un grupo

Se listan los alumnos del grupo seleccionado con su número de cuenta, nombre completo, correo electrónico y teléfono

- Crear un grupo

Se abre un nuevo grupo seleccionando el nivel y el horario.

- Borrar un grupo

Se borra uno de los grupos existentes. Es requisito que no haya ningún alumno inscrito a este grupo, de otra manera no se realiza la operación.

- Asignar profesores

Se listan los profesores que están dados de alta y se debe seleccionar un grupo de los existentes al cual será asignado.

- Asignar aula

Se especifica un aula para un grupo existente.

3. Profesores

- Lista de profesores

Se listan los profesores que están registrados con sus iniciales, nombre completo y correo electrónico.

- Agregar un nuevo profesor

Se registra un profesor con su nombre completo, correo y especificando sus iniciales.

- Borrar un profesor

Se borra el registro del profesor seleccionado

- Ver/cambiar datos de un profesor

Editar la información de un profesor que ya haya sido registrado.

4. Listas

- Listas de grupos

Se genera un archivo PDF que contiene la información de horario, profesor y aula del grupo, así como la lista de los alumnos inscritos a ese grupo en orden alfabético, mostrando su nombre, número de cuenta, correo electrónico, teléfono y número de registro de su solicitud de inscripción.

- Lista de calificaciones

Se genera un archivo PDF que contiene las calificaciones y asistencias que hayan sido registradas por el profesor para un grupo específico, además de los promedios individuales y grupales calculados por el sistema. Al final se presenta un espacio para las observaciones del profesor, su firma y un

renglón que muestra el conteo de alumnos aprobados, reprobados, bajas y los que no tienen calificación.

- Reporte de pagos

Se genera un archivo PDF que lista todos los recibos de pago registrados por el Programa de Inglés durante el periodo actual. Se da la opción de escoger que el ordenamiento de los renglones sea por un campo específico, ya sea por folio de pago, fecha de pago o nombre del alumno.

- Lista general de alumnos

Se genera un archivo PDF con la información de todos los alumnos a los que se les autorizó la inscripción en el periodo actual. Se ordenan primero por inscritos y dados de baja y luego por apellidos, mostrando también el grupo, folio de pago y número de registro de la solicitud de inscripción.

5. Cursos

- Abrir un nuevo curso

Esta opción se utiliza una vez por curso, cuando se van a especificar las fechas del nuevo periodo. Esto es posible hacerlo solamente cuando el periodo actual se haya terminado, según las fechas especificadas, ya que se cerrará el periodo para abrir uno nuevo.

- Ver todos los cursos

Se listan los cursos de los cuales se tiene registros, tanto el actual como los anteriores, con una breve información de cada uno y con la opción de ver todos los detalles al seleccionar uno en particular.

- Cambiar detalles del curso actual

Se cambian las fechas del curso, que son: inicio y fin de inscripciones, inicio y fin de clases y la fecha límite para subir calificaciones de los profesores. También se especifican los horarios disponibles para creación automática de grupos por cada nivel.

- Estadísticas

Se presenta una tabla con la información relativa al número de alumnos inscritos o dados de baja en el periodo que se seleccione, haciendo un desplegado por grupo y año de la carrera del alumno, así como los totales de

cada renglón y columna.

6. Usuarios

- Crear usuario

Se registra un nuevo usuario del sistema de administración de inscripciones.

- Ver/cambiar datos de un usuario

Editar información de un usuario existente.

- Borrar usuario

Borrar uno de los usuarios existentes.

- Cambiar mi contraseña

Cambiar la contraseña personal.

CAPÍTULO 5: PARTICIPACIÓN PROFESIONAL

El problema a resolver fue solicitado por parte de la Jefa del Programa de Inglés, la Lic. Silvia Hambleton, al Departamento de Informática Médica, del cual el Dr. Antonio Cerritos es el jefe. Él fue quien lo asignó al Laboratorio de Visualización en 3D y a su vez el Ingeniero David Limón Cruz lo asignó a mí específicamente.

El proyecto fue muy enriquecedor para mí porque puso a prueba diversos conocimientos adquiridos en la carrera de Ingeniero en Computación. Además de que requerí de creatividad, habilidades en programación y entendimiento con el cliente o usuario final, con quien llevé un diálogo cercano y frecuente para lograr un producto que cumpliera con todas sus necesidades y que además fuera entendible y manejable por usuarios.

Considerando que las personas que van a utilizar el sistema, que en este caso son la jefa del programa y las secretarias y que evidentemente no cuentan con conocimientos profundos de informática, realicé una interfaz de usuario simple, que fuera entendible y que pidiera el mínimo de información requerida para las operaciones, pero que a su vez cubriera todos los procedimientos que realiza el personal.

También se hizo un cambio importante en cuanto al repositorio de la información de los alumnos. Anteriormente se hacían carpetas con todas las hojas impresas o escritas con la información de los alumnos, de los profesores, las calificaciones, reportes de pago, etc. Además de ocupar mucho espacio físicamente, es muy difícil, complicado y tardado para encontrar información y para obtener respaldos sobre esta. Es por esto que se propuso que la información sería guardada en una base de datos alojado en el servidor del Lab3D de la Facultad de Medicina. Esto elimina el problema del espacio físico requerido y también simplifica enormemente la obtención de copias que respalden la información con la que se cuenta.

Referente a esto, quisiera mencionar que en la etapa final del semestre en que se puso en marcha este proyecto, se realizaron trabajos de remodelación en el área donde labora el personal del Programa de Inglés, bloqueando así el acceso a la información que se tenía impresa en hojas y organizada en carpetas.

Fue muy oportuna la implementación del sistema porque esto no causó ningún problema para el curso que se llevaba a cabo y que no había concluido todavía. Los profesores pudieron llenar las listas de calificación a través de internet y la jefa del programa pudo verificar las listas de calificación dentro del sistema. Las listas que fueron necesarias se pudieron imprimir nuevamente sin mayor esfuerzo y los alumnos también pueden consultar sus calificaciones por medio de la página de internet, evitando así cualquier inconveniente o problema que hubiera podido acarrear las obras de remodelación mencionadas.

CAPÍTULO 6: RESULTADOS Y APORTACIONES

El primer curso del Programa de Inglés para el año 2010 fue administrado satisfactoriamente por este proyecto, puesto en marcha desde la solicitud de inscripciones en línea en la segunda semana de enero.

Al regreso a clases en la Facultad de Medicina se le informó a los alumnos que para inscribirse al Programa de Inglés debían ingresar a la página: <http://lab3d.facmed.unam.mx/ingles/> y ahí comenzar su inscripción completando el formulario de solicitud de inscripción. Hubo algunas dudas en los alumnos en cuanto a los pasos y las preguntas del formulario en línea debido a la novedad o falta de información, pero esto se fue aclarando con el paso de los días y las solicitudes de inscripción se completaron con éxito.

Al terminar el periodo de inscripciones se registraron 758 solicitudes de inscripción, de las cuales 693 culminaron en la inscripción de los alumnos, el resto de ellos no continuó con el proceso. Este número de alumnos inscritos fue más alto de lo que se venía registrando en años anteriores para el segundo semestre del ciclo escolar, que eran menos de 500 alumnos, lo cual indica que fue un buen momento para cambiar el proceso de inscripción y hacerlo más eficiente para brindar un mejor servicio.

La creación automática de grupos y asignación de alumnos a éstos se realizó de manera satisfactoria. Se crearon treinta y dos grupos en total, de los cuales diecisiete tuvieron cupo lleno con treinta o más alumnos y diez grupos finalmente no se abrieron por contar con menos de cinco alumnos.

Para los alumnos que se encontraban en un grupo que no se abrió se les dio la opción de hacer su cambio a otro horario o de guardar sus datos y pago para el siguiente semestre, el cual quedaría registrado en el sistema, si es que no tienen la posibilidad de asistir en otro horario.

Todos estos movimientos habrían causado muchos problemas para las listas de grupos en semestres anteriores pero en esta ocasión fue muy fácil para las secretarías, como así lo mencionaron, ya que los cambios se ven reflejados inmediatamente en las

listas y pueden ser consultadas en línea en cualquier momento.

En cuanto a las calificaciones también se obtuvieron muy buenos resultados ya que fue muy bien aceptada la nueva forma de reportar las calificaciones por parte de los profesores. El vaciado de las calificaciones en la página de internet les da mayor flexibilidad y comodidad para hacerlo en tiempo y lugar a su gusto y también hacer correcciones sin mayor problema. La información que tienen que proporcionar al sistema es la asistencia y calificación de cada alumno por cada una de las dos partes en las que se divide el curso, además el porcentaje de peso que tiene el examen parcial en la calificación final por cada grupo. Habiendo hecho esto el profesor no se tiene que preocupar ni detener por el cálculo de calificaciones finales ni llenado de listas de calificaciones, ya que se realiza de manera automática.

Para que los alumnos puedan conocer sus calificaciones ya no es necesario que busquen contactar al profesor o se presenten en la coordinación. Para esto agregué una sección a la página del Programa de Inglés, en la cual con tan solo su número de cuenta se les informará de sus calificaciones si es que están inscritos al curso y el profesor ha llenado las listas de calificaciones.

CAPÍTULO 7: CONCLUSIONES

El producto de este proyecto es un sistema confiable, eficaz y completo que realiza las operaciones necesarias para administrar la inscripción de los alumnos al Programa de Inglés de la Facultad de Medicina, desde las solicitudes de inscripción y organización de grupos hasta el manejo de calificaciones y listas de reportes.

La interfaz de usuario que emplea el sistema es muy sencilla de usar y no requiere de instalaciones adicionales, únicamente un navegador de internet y por supuesto una conexión a Internet.

La combinación del uso de programación en PHP y el manejador de bases de datos MySQL brinda una amplia gama de opciones para programar un sistema basado en páginas web y bases de datos. Siendo estas dos herramientas de software libre, dan la facilidad de instalarla y utilizarla sin tener que incrementar el costo de nuestro proyecto, además de ser muy confiables y eficientes.

Por otro lado, en el primer semestre que se utilizó el sistema se mejoró mucho el proceso de inscripción, tanto por la rapidez como por organización. El personal que labora en el Programa de Inglés quedó muy satisfecho y motivado con las bondades que este sistema les brinda, además de que se minimizaron los errores.

También, la mayoría de los alumnos expresaron su agrado por la nueva forma de inscribirse a través de Internet ya que mencionan que lo pueden realizar desde el lugar y a la hora que mejor les conviene, sin tener que estar sujetos a los horarios de trabajo en la coordinación.

GLOSARIO

- Apache: Acrónimo de "*a patchy server*". Servidor web de distribución libre y de código abierto, siendo el más popular del mundo desde abril de 1996.
- Código fuente: Todas las etiquetas e instrucciones contenidas en un archivo de texto utilizado para componer una página Web, un programa, etc.
- Compilador: es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar. Este proceso de traducción se conoce como compilación.
- CPU (*Central Processing Unit*): Unidad Central de Procesamiento. Es la parte de la computadora que interpreta las instrucciones contenidas en los programas y procesa la información. En la CPU se encuentran: la unidad de control, unidad aritmético lógica o ALU y la memoria principal.
- Enlace (*link*): Es una palabra, una frase o un gráfico de un hipertexto que conecta con otra información. En internet los enlaces (o vínculos) de una página web conectan con otras páginas, a menudo muy alejadas físicamente, pero relacionadas por sus temática.
- GUI (*Graphic User Interface*): Interfaz Gráfica de Usuario. Es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.
- Hipertexto: Documento que reúne imágenes, textos, sonidos o vídeos relacionados entre sí por medio de enlaces, de tal modo que al señalar una palabra o gráfico se pasa de uno a otro. La *World Wide Web* es una forma de usar la Internet por medio de hipertextos conectados entre sí.
- HTML: Siglas de *HyperText Markup Language* (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.
- IDE (*Integrated Development Environment*): Entorno de Desarrollo Integrado. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).
- Interfaz de usuario: Son aquellas que incluyen aspectos como menús, ventanas, teclado, ratón, en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora.

- **Intérprete:** Es un programa informático capaz de analizar y ejecutar otros programas, escritos en un lenguaje de alto nivel. Los intérpretes difieren de los compiladores en que mientras estos traducen un programa desde su descripción en un lenguaje de programación al código de máquina del sistema, los intérpretes sólo realizan la traducción a medida que sea necesaria, comúnmente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción.
- **Macro:** Abreviatura de macroinstrucción. Es una serie de instrucciones que se almacenan para que se puedan ejecutar de forma secuencial mediante una sola llamada u orden de ejecución.
- **Multiplataforma:** Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas. Por ejemplo, una aplicación multiplataforma podría ejecutarse en Windows en un procesador x86, en GNU/Linux en un procesador x86, y en Mac OS X en uno x86 (solo para equipos Apple) o en un PowerPC.
- **Ofuscación:** Se refiere a encubrir el significado de una comunicación haciéndola más confusa y complicada de interpretar. En computación, la ofuscación se refiere al acto deliberado de realizar un cambio no destructivo, ya sea en el código fuente de un programa informático o código máquina cuando el programa está en forma compilada o binaria, con el fin de que no sea fácil de entender o leer.
- **PDF (*Portable Document Format*):** Formato de archivo que reproduce un documento tal y como éste se despliega en pantalla o en papel manteniendo su apariencia original.
- **Script (guión):** En informática, un *script* es un conjunto de instrucciones. Permite la automatización de tareas, creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Éstos son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto.
- **Servidor Web:** Es el programa que, utilizando el protocolo de comunicaciones HTTP, es capaz de recibir peticiones de información de un programa cliente (navegador), recuperar la información solicitada y enviarla al programa cliente para su visualización por el usuario.
- **Software:** Se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas; en contraposición a los componentes físicos del sistema. Tales componentes lógicos incluyen, entre muchos otros, aplicaciones informáticas como procesador de textos, que permite al usuario realizar todas las tareas concernientes a edición de textos; software de sistema, tal como un sistema operativo, que, básicamente, permite al resto de los programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de las aplicaciones, también provee una interfaz para el usuario.

BIBLIOGRAFÍA

L. Joyanes. *Programación en Turbo Pascal.* McGraw-Hill, 1990.

J. Tackett, S. Burnett. *Edición Especial Linux.* Prentice Hall, 2000.

M. Maslakowski, T. Butcher. *Aprendiendo MySQL en 21 Días.* Prentice Hall, 2001.

H. Williams, D. Lane. *Web database applications with PHP and MySQL.* O'Reilly, 2004.

Sklar, D. *Introducción a PHP 5: Construcción de sitios web dinámicos de forma fácil y sencilla.* Anaya Multimedia, 2005.

Ullman, L. *PHP and MySQL for dynamic web sites.* Peachpit, 2003.

C. Musciano, B. Kennedy. *HTML, the definitive guide.* O'Reilly, 1997.

Widenius, M. *MySQL reference manual.* O'Reilly, 2003.

Quiroz, J. *El modelo relacional de bases de datos.* Boletín de Política Informática Núm. 6, 2003

<http://www.php.net>