



UNIVERSIDAD DE
SOTAVENTO, A.C.



ESTUDIOS INCORPORADOS A LA UNIVERSIDAD NACIONAL AUTONOMA DE
MÉXICO

FACULTAD DE INFORMATICA

“CREACION DEL SISTEMA MEDICO DE LA UNIVERSIDAD DE SOTAVENTO”

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN INFORMATICA

PRESENTA:

LUCIA MORALES AMAYA

ASESOR DE TESIS

LIC. RAUL DE JESUS OCAMPO COLIN

COATZACOALCOS, VER.

ENERO 2010



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.

Con cariño:

Para mis maestros por abrirme el camino del conocimiento y darme las bases para ser un buen profesionalista

A mi padre, por estar siempre a mi lado, por darme su cariño, su amor y apoyarme siempre, por ser mi ejemplo a seguir.

A mi madre por que sin ella jamás habría podido llegar hasta donde estoy, por que es mi compañera, mi amiga, mi confidente, gracias por ser esa persona en la cual puedo confiar, la cual me ha apoyado siempre y a la cual le agradezco infinitamente haberme dado la vida y solo espero ser la mitad de buena madre de lo que ella fue conmigo.

A mi hermano que es la persona que mas quiero aparte de mis padres y mi hija, porque a pesar de las peleas siempre ha estado cuando lo necesito, gracias hermano por todo, te quiero.

A mi hija que es la persona mas importante en mi vida, el motor que me hace seguir día a día, doy gracias a dios por haberme regalado un tesoro tan grande como lo eres tú, te amo y te amare siempre, esto es por ti, solo espero poder darte el futuro que te mereces amor de mis amores...te amo, de aquí a la luna, las estrellas y de regreso...

Y por ultimo pero no menos importante gracias a todos mis amigos y amigas que creyeron en mí y me apoyaron, gracias Mario, Mayra, Daniela, y a todos aquellos que estuvieron conmigo, siempre los tendré muy presente en mi vida...

Gracias Dios por cada reto que me has impuesto, los cuales siempre he podido superar, es cierto no fácilmente, pero que seria de la vida si las cosas fueran fáciles, gracias por recorrer conmigo cada día, sin tu presencia yo no sería nada, por todo esto y más...

¡GRACIAS!

INDICE

CAPITULO 1

1.0 ESTRUCTURA DE LA TESIS	5
1.1 Principios filosóficos	7
1.2 Código de Ética y Valores	9
1.2.1 Misión	11
1.2.2 Visión	11
1.3 Filosofía Institucional	12
1.4 Objetivos Institucionales	12
1.5 La Dimensión Académica de la Universidad de Sotavento	14
1.6 Formación Profesional	14
1.7 La Práctica Educativa	16
1.8 Planes de Estudios	17
1.9 Reconstrucción histórica de la Universidad de Sotavento A.C	21

CAPITULO 2

2.0 PLANTEAMIENTO DEL PROBLEMA	23
2.1 Justificación	24
2.2 Hipótesis	24
2.3 Objetivo General	24
2.4 Objetivos Específicos	24
2.5 Alcances y Limitaciones	25

CAPITULO 3

3.0 MARCO TEORICO	27
3.1 Alta de Expedientes	27
3.2 Programación en Java	28
3.3 Filosofía	37
3.4 Críticas	41
3.5 Sintaxis	45
3.6 Recursos	48
3.7 Java en código abierto	51
3.8 ¿Qué es MySQL?	51

CAPITULO 4

4.0 DISEÑO DEL PROGRAMA PROPUESTO	58
4.1 DEFINICION DE LAS TABLAS DE LA BASE DE DATOS	59
4.2 CAPTURA DE PANTALLAS DEL SIAU - MEDI	61
4.3 Pantalla Principal	61
4.4 Menú Expedientes	62
4.4.1 Alta de Expediente Alumnos	63
4.4.2 Alta de Expediente Administrativos	64
4.5 Menú Inventarios	67
4.5.1 Inventario Medicamentos	67
4.5.2 Inventario Instrumental	68
4.6 Menú Reportes	69
4.6.1 Reporte Inventarios	70
4.6.2 Reporte Expedientes	71
4.6.3 Programación de Eventos	72
4.6.4 Reporte Bitácora	73
4.7 Perfil Medico	74
4.8 Salir del Sistema	74
BIBLIOGRAFIA	77
CONCLUSIÓN	75

CAPITULO 1

1.0 ESTRUCTURA DE LA TESIS

Nos toca vivir una época en la que cada vez la informática invade más campos de nuestra vida, estando el ciudadano medio cada vez más familiarizado con términos del mundo informático, entre ellos, como no, los lenguajes de programación. Los cuales nos ayudan a la creación de sistemas de información que faciliten el uso y almacenamiento de información dentro de las organizaciones.

Día con día la informática contribuye mas al alcance de logros que permiten a las organizaciones la automatización de muchos de los procesos que llevan a cabo cotidianamente, logrando con esto la reducción de tiempo en la realización de todos estos.

A lo largo de la presente, se muestra como ha presentado un avance significativo la informática en el campo de la educación. A continuación se expone una de las necesidades que se crean de manera especial para llevar a cabo las operaciones de la Universidad de Sotavento.

El Capitulo 1 corresponde la introducción del Tema, donde se describe la historia de la Universidad de Sotavento, sus objetivos, la misión, la visión y su oferta académica.

En el Capítulo 2 se realiza el Planteamiento del Problema, comprendiendo la justificación, objetivos tanto generales como específicos, así como los alcances y limitaciones de la presente investigación.

En el Capitulo 3 se presentan los lineamientos y requisitos necesarios para llevar a cabo las operaciones en la propia Universidad, así como los aspectos más relevantes del lenguaje de programación empleado para dar solución al problema.

El capitulo 4 expone el diseño del proceso que deberá seguirse para la realización de las operaciones y solución del problema, por otro lado se especifica la forma en que deberá operar el sistema, mediante una demostración de las pantallas, así como la función de los usuarios del sistema.

Al finalizar se presentan las conclusiones a las que se llegaron después de haber terminado el presente trabajo de investigación.

En general, lo que se presenta a continuación es una alternativa para el mejoramiento de las funciones administrativas de la Universidad de Sotavento, valiéndose de las herramientas informáticas, específicamente de los lenguajes de programación.

La Universidad de Sotavento, A.C. tiene como funciones sustantivas la docencia, la investigación, la difusión de la cultura y la extensión de los servicios, mismos que están en permanente vinculación con los diferentes sectores sociales.

Bajo la concepción dinámica que articula congruentemente el objetivo y la visión de la institución esta se orienta a las acciones a seguir en el proceso educativo, para asegurar el cumplimiento de la misión en un afán de mejorar y medir las interacciones con fines descriptivos y predictivos.

El Modelo educativo está articulado como un sistema que interactúan con un propósito común a partir de un proceso central denominado “proceso educativo” y que se apoyará con cinco procesos estratégicos que son: el académico, planeación, administración de recursos, vinculación y difusión de la cultura y el de innovación y calidad, en el que de manera fundamental gira entorno del ser humano y de su aprendizaje, desde la perspectiva de la construcción del conocimiento en sus diversas formas.

1.1 Principios filosóficos

Los principios que norman la vida de la Universidad de Sotavento, A.C., emanan de conformidad a la reglamentación interna y sustentada en el Artículo 3º de la Constitución Política de los Estados Unidos Mexicanos, y de la Ley General de Educación en su Artículo Séptimo, en la que de acuerdo con éstos, la educación en todos sus tipos y modalidades, debe basarse en los resultados de progreso científico, ser ajena a las doctrinas religiosas, ser democrática y contribuir al logro de un sistema de vida que busque el constante mejoramiento económico, social y cultural, promoviendo contenidos ecológicos que permitan la generación de conocimientos estratégicos acerca de la naturaleza, su aprovechamiento y conservación.

La Ley General de Educación en su artículo séptimo señala que los esfuerzos educativos deben promover el máximo desarrollo de las capacidades humanas, la observación, análisis y pensamiento crítico, fortalecer la conciencia de nacionalidad y soberanía así como el uso de un idioma común a todos los mexicanos; propiciar el conocimiento y práctica de la democracia y enfatizar el valor de la justicia, fomentar la investigación y la innovación científica y tecnológica, la difusión de los bienes y valores culturales, la solidaridad, el respeto a la libertad del individuo, el rechazo a los vicios y la protección del ambiente.

En consecuencia el Modelo Educativo de la Universidad de Sotavento, A.C., se enfoca en el ser humano y todas sus estrategias educativas se centran en el aprendizaje. Además de formar profesionistas competentes, el Modelo propicia la orientación de los proyectos de vida hacia la búsqueda de la autorrealización, en un escenario de cultivo permanente del humanismo, con fundamento en las consideraciones expuestas la Universidad establece los siguientes como principios filosóficos de su Modelo Educativo:

- a) El ser humano como persona es capaz de trascender; con identidad propia, pero al mismo tiempo interdependiente; un ser histórico, creador de su cultura, su lenguaje, consciente y por ende responsable de sus acciones, es un ser constante transformador y siempre en la búsqueda del conocimiento, la justicia, la felicidad y de su autorrealización.

- b) La educación integral como un proceso continuo de desarrollo de todas las potencialidades del ser humano, lo orienta hacia la búsqueda de su realización, en el aprender a ser, aprender a hacer, aprender a aprender, aprender a emprender y aprender a convivir, para que se convierta en un ciudadano consciente, responsable y solidario.
- c) La formación del ser humano comprende el desarrollo del espíritu, a través de la cultura; del intelecto, mediante la vida académica; de los sentimientos y emociones, por la convivencia y la vida artística, de la integridad física, por medio del deporte y la orientación para la salud y de la vida social mediante actividades cívicas.
- d) El ser humano como actor fundamental del proceso educativo. En su formación se promueve el aprendizaje significativo mediante la labor del docente como facilitador, a través de la reflexión y la participación apoyados con los instrumentos didácticos y tecnología de vanguardia, y asegurando la calidad, actualización y pertinencia del aprendizaje. Su compromiso será con el desarrollo sostenido e integral del país y como miembro de la comunidad de la Universidad de Sotavento, A.C., adoptará la filosofía y deberá estar consciente del privilegio y la responsabilidad que significa su ingreso, permanencia y egreso de la institución.
- e) La Universidad de Sotavento, A.C., promueve, reconoce, respeta y gestiona las aportaciones intelectuales originales, producto de los diversos ámbitos del trabajo profesional.
- f) Todos los recursos de la Universidad de Sotavento, A.C., se organizan para su ejercicio eficiente y eficaz con una clara pertinencia para el Proceso Educativo, bajo un esquema de mejora continua, todo ello, en un ambiente sociolaboral responsable, respetuosa y solidario.
- g) Los cuerpos colegiados y el trabajo en equipo favorecen la sinergia y potencian la obtención de resultados de gran alcance y calidad, por lo que se fomentan en el

Proceso Educativo, y se promueve el establecimiento de redes de colaboración entre departamentos.

- h) La Universidad de Sotavento, A.C., reconoce en sus funciones sustantivas vías importantes para el crecimiento humano y el desarrollo social; de manera que su personal, estudiantes y egresados conformen un valioso capital humano, con plena capacidad de comprender críticamente, con tolerancia, apertura y flexibilidad a las expresiones actuales del pensamiento universal, además de generar y aplicar conocimientos.
- i) La Universidad de Sotavento, A.C., concibe al liderazgo como la capacidad para integrarse en la conducción visionaria, participativa y comprometida con los procesos de innovación, calidad y desarrollo, por lo que promueve y lo impulsa.

1.2 Código de Ética y Valores

Considerando que la actualidad se caracteriza por un nuevo renacimiento de la humanidad, en el que se dan las condiciones para que el ser humano se reconstituya en su integridad histórica y social, bajo esta perspectiva basamos la convicción de que es posible construir esquemas y espacios de convivencia humana, en los que coincidan las bondades del avance científico y tecnológico, y del pensamiento social.

Los valores que guían el actual renacimiento son los que históricamente han sido base de los grandes logros de la sociedad, por lo que la Universidad de Sotavento, A.C., los hace suyos y los incorpora a su modelo educativo, estos valores son:

- a) Respeto a la persona

La persona, en toda su dignidad es el fin y razón de la Universidad de Sotavento, A.C., por lo tanto el proceso educativo se orienta a que los seres humanos, con el ejercicio de su libertad, responsabilidad e inteligencia, y con la fuerza de sus sentimientos y actitudes que los lleven a la plenitud de sus potencialidad en beneficio propio y de la sociedad a la que pertenecen.

b) Respeto a la diversidad

La Universidad de Sotavento, A.C., ofrece con equidad sus servicios educativos a la sociedad en general sin distinción de ninguna índole y desarrolla, entre otros, programas y estrategias para estudiantes que se encuentran en situaciones especiales

c) Desarrollo sostenible

El desarrollo social que se promueve desde los diversos espacios de la Universidad de Sotavento, A.C., se dirige al respeto y cuidado a los recursos naturales además de propiciar acciones que responsabilicen del cuidado que corresponda a su generación y a generaciones futuras.

d) Formación integral

La Universidad de Sotavento, A.C., promueve el desarrollo armónico de todas las potencialidades del ser humano, incluidas todas sus formas de pensar.

e) Responsabilidad y colaboración

Los retos que enfrentan las instituciones y las personas exigen un replanteamiento de la práctica educativa donde la participación responsable de todos los actores y el equilibrio dinámico entre el trabajo individual y el colaborativo son propósitos explícitos en los procesos formativos.

f) Creatividad e innovación

Los problemas inéditos, su complejidad y los escenarios cambiantes, que no se agotan en un campo disciplinario ni con una sola metodología, así como la urgencia de encontrar soluciones, exigiendo promover la actitud proactiva, la innovación y el pensamiento lateral; por consiguiente para la Universidad de Sotavento, A.C., la creatividad, la iniciativa y el desarrollo de competencias para plantear y resolver problemas son estrategias formativas prioritarias.

g) Calidad de vida

El sentido humano de la ciencia y la tecnología estriba en el por qué, para qué y para quién se usan y dirigen; como consecuencia, la Universidad de Sotavento, A.C., asume la

responsabilidad y el compromiso de promover la generación y aplicación del conocimiento en beneficio de la humanidad.

h) Identidad nacional y cultura universal

La interdependencia entre los pueblos es cada vez más necesaria, por esta razón y con el fin de fortalecer nuestra identidad nacional y, simultáneamente, fortalecer el sentido de la convivencia, es esencial promover, incrementar, transmitir y difundir los mejores logros de nuestra cultura regional, nacional y de la cultura universal.

i) Ética profesional

La Universidad de Sotavento, A.C., se rigen por códigos de conducta que nos comprometen con los integrantes de la institución a ejercer su función enalteciendo los valores universales en beneficio de ellos mismos, de los educandos y de la sociedad en general.

j) Conocimiento y comunidad

Los procesos formativos de la Universidad de Sotavento, A.C., dan respuesta a las necesidades individuales y colectivas de la realidad institucional y comunitaria, por lo que la actualización, capacitación, extensión y la formación permanentes constituyen pilares de la sociedad del conocimiento.

1.2.1 Misión

Formar profesionales de excelencia académica, capaces de comprometerse con el desarrollo de su entorno con un amplio sentido analítico, creativo y humanista, que le otorgue el liderazgo nacional e internacional, mediante la docencia, el conocimiento científico, la investigación y la difusión de la cultura y el deporte.

1.2.2 Visión

“Ser una institución educativa líder con reconocimiento nacional e internacional que otorgue prestigio a sus egresados en su campo profesional con el mas alto nivel de calidad y excelencia académica”.

1.3 Filosofía Institucional

- I. La enseñanza se basará en los resultados obtenidos por la ciencia y el conocimiento, aplicando nuevos métodos y técnicas pedagógicas;
- II. Los egresados serán capaces de aplicar los conocimientos adquiridos con acierto a los problemas inherentes al ámbito propio del ejercicio de su profesión;
- III. Crear espacios que posibiliten el desarrollo de las personas y de las comunidades sociales;
- IV. Promover la capacitación de su personal como actitudes básicas;
- V. Fomentar, organizar y coordinar la investigación científica, tecnológica y humanística como fuente del desarrollo;
- VI. Organizar y desarrollar las actividades de docencia a nivel superior en sus diversas modalidades siendo congruentes a los requerimientos de la sociedad;
- VII. Promover, organizar y coadyuvar en todas aquellas actividades encaminadas a la actualización de los profesionales en ejercicio y vigilar el correcto desempeño de su función profesional y social;

1.4 Objetivos Institucionales

- 1) Conservar, crear y transmitir la cultura en beneficio de la sociedad con el más alto nivel de calidad académica a través de las funciones sustantivas de docencia, investigación, difusión de la cultura y extensión de los servicios.
- 2) Estar vinculada permanentemente con la sociedad, para incidir en la solución de sus problemas y en el planteamiento de alternativas para el desarrollo, sustentadas en el avance de la ciencia y la tecnología, proporcionándole los beneficios de la cultura y obteniendo de ella en reciprocidad, los apoyos necesarios para su fortalecimiento.

- 3) Formar profesionistas de una excelencia académica, capaces de asumir el compromiso que tienen con la sociedad, de conservar, generar y transmitir el conocimiento científico, humanístico, artístico y tecnológico, mediante la docencia, la investigación, las actividades deportivas y la difusión de la cultura.
- 4) Fomentar en los integrantes de la comunidad universitaria la actitud crítica, humanística, científica y el espíritu emprendedor y de participación.
- 5) Promover que los programas educativos y acciones de extensionismos institucional, sean congruentes a los requerimientos de la sociedad veracruzana y mexicana, e impliquen soluciones a los mismos.
- 6) Apoyar, estimular y generar en el alumno la capacidad de crítica reflexiva y realista, como expresión básica del desarrollo y de su forma de pensar.
- 7) Orientar al alumno en la estructuración consciente y responsable de su jerarquía de valores ético-morales que le permitan modelar y moderar su relación y comportamiento personal.
- 8) Transformar el aula en laboratorio de búsqueda constante, de estrategia y metodología, que propicien el mejor aprovechamiento académico.
- 9) Proporcionar a los alumnos un ambiente de estabilidad, tranquilidad y reflexión, que le permita el óptimo y continuo desarrollo educacional, así como el análisis de los acontecimientos económicos, políticos y culturales de la sociedad que lo rodea.
- 10) Estructurar una planta docente con elementos de amplia experiencia, reconocida y solvencia moral, que fomente el libre desarrollo de la personalidad humana, alentando la búsqueda de la verdad, mediante la formación de un espíritu científico y una conciencia social y liberal.
- 11) Mantener la capacitación continua del personal docente y la implantación de nuevas técnicas educativas.

- 12) Impulsar en sus programas académicos, los principios, valores y práctica de la democracia, la justicia, la libertad, la igualdad, la solidaridad y el respeto a la dignidad humana.
- 13) Realizar acciones conjuntas con asociaciones y organizaciones de docencia, investigación, difusión de la cultura y extensión de los servicios, en el ámbito nacional e internacional.
(Reglamento Interno de la UIA. 2003)

1.5 La Dimensión Académica de la Universidad de Sotavento

La dimensión académica es otro de los ejes fundamentales del Modelo Educativo de la Universidad de Sotavento, A.C., en ella se engloban los parámetros de referencia para la formación profesional, la concepción del aprendizaje y sus condiciones, así como los estándares de la práctica educativa.

1.6 Formación Profesional

La formación profesional de que es objeto el ser humano que participa en el Proceso Educativo de la Universidad de Sotavento, A.C., busca desarrollarse como un actor ético, asertivo y exitoso en su campo de acción con una clara identidad organizacional y compromiso social, así como en una persona íntegra en constante búsqueda de su autorrealización.

En consecuencia, una de las finalidades fundamentales del Proceso Educativo de la Universidad de Sotavento, A.C., busca desarrollarse en un actor ético, asertivo y exitoso en su campo de acción con una clara identidad organizacional y compromiso social, así como una persona íntegra en constante búsqueda de su autorrealización a través de un proceso continuo de desarrollo en todas sus potencialidades del ser humano, que lo orienta a la búsqueda de su plenitud en el aprender a ser, a hacer, a aprender a emprender y a convivir, lo cual lo convierte en un ciudadano y un profesionista consciente, responsable y solidario.

La formación profesional que impulsa y fomenta el Modelo propicia lo siguiente:

- a) El humanismo, como el entendimiento respetuoso tolerante y crítico de todas las manifestaciones de la cultura universal, sin menoscabo alguno ni exclusivismos, con el propósito de ampliar los horizontes de la educación integral del ser humano.
- b) El desarrollo de un conocimiento amplio y objetivo de la problemática contemporánea del ámbito profesional, desde las perspectivas regional, nacional y mundial.
- c) El desarrollo de las diferentes formas de competencias que permitan la certificación y el reconocimiento de clase mundial.
- d) La investigación como una forma de generar conocimientos que por su pertinencia y actualidad enriquezcan el acervo, así como una estrategia de fortalecimiento de la vinculación de la Universidad de Sotavento, A.C., con su entorno regional y nacional e internacional, y que tiene como propósito mejorar las condiciones de vida del ser humano.
- e) Un sentido de identidad profesional y pertenencia institucional.
- f) El desarrollo de las capacidades y habilidades para obtener, analizar, interpretar y aplicar información, generar conocimientos, así como para identificar, plantear, resolver problemas y tomar decisiones.
- g) El cultivo de habilidades y la capacidad de liderazgo para gestionar, emprender, negociar y dirigir.
- h) El crecimiento cualitativo de las capacidades y habilidades para el trabajo colegiado, en equipo, en situaciones cambiantes y en ambientes multiculturales.
- i) La formación de un profesionalista que se actualiza constantemente y es competente en la comunicación oral y escrita, así como en otro idioma.
- j) La búsqueda permanente del conocimiento y la corresponsabilidad en el proceso de formación, como características inherentes al modo de vida del ser humano.

1.7 La Práctica Educativa

A través de la práctica educativa la Universidad de Sotavento, A.C., busca asegurar la formación integral del ser humano. Por ello se considera que, en todo proceso de aprendizaje, son esenciales las acciones del facilitador y de la persona que construye su conocimiento, desde la perspectiva que el proceso es un escenario de aprendizaje significativos para ambos, bajo la concepción de este nuevo paradigma, en el sentido que exige actitudes nuevas y propicias para la construcción del conocimiento, su facilitación, el trabajo colaborativo y la comunicación asertiva.

La práctica educativa está orientada por la convicción de que el trabajo educativo en la Universidad de Sotavento, A.C., asegura el desarrollo de la región así como de la entidad y la cual se caracteriza por lo siguiente:

- a) Impulsa la formación integral del ser humano.
- b) Busca el aprendizaje significativo de los actores involucrados en el Proceso Educativo.
- c) Se funda en el interés intelectual de los actores del Proceso Educativo.
- d) Fomenta el trabajo colegiado y colaborativo, y el desarrollo de la comunicación asertiva.
- e) Se alimenta de conocimientos actuales, vigentes y pertinentes para el desarrollo sostenible.
- f) Se orienta hacia el desarrollo de habilidades para el planteamiento y la solución de problemas.
- g) Concibe a la evaluación como una estrategia para asegurar e impulsar la construcción del conocimiento.
- h) Estimula la continua actualización profesional.

- i) Se nutre con la filosofía humanista y cultiva el espíritu de colaboración y compromiso social.
- j) Crea un ambiente propicio para el aprendizaje y la generación de conocimientos a los que se dedican los recursos necesarios para asegurar las condiciones que permitan el éxito del Proceso Educativo.

1.8 Planes de Estudios

Las licenciaturas de la Universidad de Sotavento, A. C., conciben procesos integrados de generación de conocimientos y aprendizajes. Las habilidades profesionales que se buscan crear en los estudiantes comprenden:

1.8.1 Licenciatura en Administración de Empresas.

Formar profesionistas que propicien la satisfacción de necesidades y el logro de objetivos de las instituciones públicas, privadas y sociales a través del aprovechamiento adecuado de sus recursos, mediante la planeación, organización, dirección y control, en las distintas áreas funcionales: Personal, finanzas, mercadotecnia, producción e informática.

1.8.2 Licenciatura en Arquitectura.

Formar profesionistas capacitados para proyectar y construir los espacios habituales que la sociedad necesita para vivir, partiendo tanto de condiciones de índole cultural, como físicas, en un proceso de interrelación y mutua determinación entre estos dos campos propios del ser humano.

1.8.3 Licenciatura en Contaduría.

Formar profesionistas capacitados para satisfacer necesidades de las organizaciones tales como obtener y comprobar su información financiera, llevar a cabo una debida

administración y protección de los intereses de los organismos, cumplir con sus obligaciones, derechos y patrimonios. Aportar los elementos de juicio para la toma de decisiones.

1.8.4 Licenciatura en Ciencias de la Comunicación

Estudiar los orígenes, la escritura, el desarrollo y los cambios de los procesos de comunicación humana en sus distintos niveles y facetas tanto interpersonales como grupales, organizacionales, colectivas o masivas, para contribuir de manera crítica a la difusión y análisis de la problemática social a través de la elaboración de mensajes y contenidos informativos escritos y audiovisuales.

1.8.5 Licenciatura en Derecho

Promover la Justicia Social e Individual por medio de la educación de los ordenamientos legales a la realidad. Procurar la defensa y aplicación de los valores nacionales surgidos y consagrados a lo largo de nuestra historia y plasmados en la Constitución Política de los Estados Unidos Mexicanos. Aplicar con honestidad y lealtad los principios éticos del derecho en beneficio de la comunidad.

1.8.6 Licenciatura en Informática

Formar profesionales capaces de diseñar e implantar soluciones basadas en sistemas de información, que faciliten la toma de decisiones y agilicen las operaciones propias de una organización. Esta labor debe adaptarse a los avances tecnológicos y cambios socioeconómicos en el ámbito nacional e internacional.

1.8.7 Ingeniería en Sistemas Computacionales

Formar profesionistas capaces de operar con éxito los sistemas de computo de la empresas de todo tipo, así como el diseño de sistemas de información. Comprometidos con la sociedad en la que viven, dotados de una sólida ética profesional y con un amplio criterio de su entorno social.

1.8.8 Ingeniería Industrial

La industria nacional e internacional, requiere egresados del ramo, quienes mediante su labor, impulsen y modernicen las estructuras existentes dentro de los sectores productivo, comercial y de servicios. Así en México, esta disciplina debe contribuir a renovar la organización productiva y apoyar a la industria en su conjunto, a fin de colocarla en un nivel competitivo, tanto en el mercado interno como en el externo.

Aunque en un principio la Ingeniería Industrial se inició con un Análisis sobre el Trabajo y la Administración Científica, posteriormente, profundizó en Estudios de Métodos, Planeación y Control de la Producción, Investigación de Operaciones y Control de la Calidad. En las últimas décadas ha rebasado el ámbito de la industria, aplicándose también a las empresas mexicanas y extranjeras o instituciones de servicios de salud, transporte, comercio, finanzas, seguridad industrial y de ecología entre otros.

El campo de acción de este profesionista abarca tanto los sistemas operativos y productivos, así como los financieros y administrativos. Dentro de estas ramas sus principales actividades consisten en la integración, diseño, control, desarrollo e innovación de procesos y sistemas.

Su ejercicio profesional adopta diversas modalidades, desempeñándose como: Ingeniero de todo el amplio espectro de la Producción, Ejecutivo, Investigador, Consultor y Generador de Empresas. Estas funciones las lleva a cabo dentro de las áreas de Planeación, Producción, Sistemas, Calidad, Materiales, Procesos Industriales, Capacitación, Proyectos, Envase y Embalaje, Logística, Reingeniería, Administración y Finanzas, Localización, Distribución y Mantenimiento de Plantas Industriales, dentro del marco ecológico contribuyendo al desarrollo sustentable, en la referencia de la productividad y de la calidad.

Para realizar su tarea, utiliza métodos y modelos matemáticos, físicos, químicos y computacionales, además de técnicas y tecnologías de Ingeniería, fundamentos sólidos de Economía, Administración, Finanzas y Dirección de Empresas, a fin de optimizar los procesos industriales, comerciales y de servicios, ostentando así, una competitividad internacional.

Sus actividades profesionales le demandan adaptarse con Creatividad, Conocimientos,

Actitudes, Aptitudes y Habilidades y alto Sentido Ético y de Responsabilidad Social, a los grupos interdisciplinarios y multidisciplinarios para contender con éxito en los cambios que se presenten en su vida laboral.

1.8.9 Licenciatura en Pedagogía

Formar profesionales capaces de analizar problemas, proponer soluciones relacionadas con el campo educativo y atender a diversos sectores de la población que requieren ayuda pedagógica. El pedagogo emplea técnicas como la psicometría, la estadística descriptiva, programas de computación, dinámica de grupos y otras, de acuerdo con el método de enseñanza o investigación adoptado.

1.8.10 Licenciatura en Psicología

Formar profesionistas capaces de estudiar el comportamiento de los organismos para dar solución a los problemas en los que interviene como dimensión principal el comportamiento humano, ya sea a nivel individual o de pequeños grupos, aún cuando los efectos de su acción se expandan con frecuencia a grupos numerosos.

1.9 Reconstrucción histórica de la Universidad de Sotavento A.C

La Universidad de Sotavento A.C., es una Institución de Educación Superior, cuya sede principal, se encuentra en la ciudad de Coatzacoalcos, Veracruz.

La Universidad de Sotavento A.C., inicia actividades en la ciudad de Coatzacoalcos, Ver., el año de 1994, con dos licenciaturas: Lic. En Derecho y Lic. En Contaduría, con planes y programas de estudio incorporados a la UNAM.

En el año de 1995 la Universidad de Sotavento A.C., abre tres nuevas licenciaturas, como son: Lic. En Administración, Lic. En Ciencias de la Comunicación y Lic. En Informática, también con estudios incorporados a la UNAM.

En el año de 1997 la Universidad de Sotavento A.C., abre tres nuevas licenciaturas, como son: Lic. En Arquitectura, Ingeniería Industrial y Lic. En Pedagogía; en ese año también abre la Maestría en administración, todas estas nuevas opciones de estudios, con incorporación a la máxima casa de estudios del país como es la UNAM.

En el año 2000 la Universidad de Sotavento A.C., recibe la autorización de la Secretaria de Educación Publica para impartir la carrera de Ingeniería en Sistemas Computacionales, la que se ofrece en la modalidad escolarizada.

En el año de 2001, nuevamente la Universidad de Sotavento A.C., apertura dos opciones más a su ya amplia gama de oferta educativa, dichas licenciaturas son: Licenciatura en Psicología en modalidad escolarizada y Licenciatura en Derecho en modalidad abierta ambas carreras con incorporación a la UNAM.

Para el año 2002 la Universidad de Sotavento A.C., decide agregar a su oferta educativa la Licenciatura en Educación Física, esta con el reconocimiento de Validez oficial de estudios por parte de la Secretaría de Educación y Cultura (hoy Secretaría de Educación de Veracruz).

Actualmente la Universidad de Sotavento A.C., cuenta con una población de 1925 alumnos del nivel de licenciatura y 201 de maestrías, en el plantel Coatzacoalcos.

CAPITULO 2

2.0 PLANTEAMIENTO DEL PROBLEMA

La Universidad de Sotavento es un organismo dedicado a la oferta de servicios educativos a la sociedad en general sin distinción de ninguna índole y desarrolla, entre otros, programas y estrategias para estudiantes que se encuentran en situaciones especiales

Actualmente la Universidad cuenta con 11 Licenciaturas y más de 1900 alumnos, a los cuales, además de su formación educativa también ofrecen cursos y congresos así como apoyos económicos a los alumnos mediante becas como parte de su responsabilidad social.

Debido al crecimiento tan importante que ha presentado desde el inicio de sus operaciones en el 1994, es necesario tener información constante sobre la salud tanto física como mental de los estudiantes, por tanto la Universidad a implementado un Servicio medico, el cual es totalmente gratuito y ofrece consultas medicas llevadas a cabo por personal profesional, el cual cuenta con el instrumental y medicamentos necesarios para llevar realizar dicha tarea.

2.1 Justificación

El servicio médico de la Universidad de Sotavento, A.C. se encarga entre otras cosas de mantener la salud de los estudiantes en optimo estado, mediante consultas medicas, las cuales son realizadas por un medico contratado exclusivamente para ello.

Debido al incremento de alumnos es necesario llevar un expediente detallado sobre la salud de estos, el cual hasta la fecha se realizaba a mano y requería de constante papelería, y actualización manual de los mismos, lo cual en la actualidad y contando con los sistemas de información que maneja la universidad, es totalmente innecesario, y poco viable.

2.2 Hipótesis

“La implementación de un sistema con una interfaz agradable al usuario, adicionara a un mejor funcionamiento al servicio médico, contribuirá a un ahorro de tiempo en los procedimientos de control médico así como a tener información más precisa y detallada sobre la salud de cada alumno, logrando con esto mejorar el servicio que se ofrece en la actualidad”

2.3 Objetivo General

En un esfuerzo por solucionar los problemas ya mencionados, así como la parte Administrativa del Sistema, el objetivo general del presente trabajo de tesis consiste en la creación de un sistema para control médico, así como la ordenación lógica y aprovechamiento de la Base de Datos que será utilizada en todo proceso del sistema.

2.4 Objetivos Específicos

- ❖ Llevar un expediente médico de los alumnos, con la finalidad de estar enterados de cualquier eventualidad que se presente durante el horario escolar.
- ❖ Llevar un control sobre los medicamentos que se tienen en el consultorio, por medio de un inventario el cual se mantendrá actualizado constantemente.

- ❖ Llevar un control sobre el instrumental medico, por medio de un inventario el cual se mantendrá actualizado constantemente.
- ❖ Llevar un control sobre la programación de eventos médicos durante el ciclo escolar requerido.
- ❖ Crear un Sistema de control medico, entendible y de fácil uso.
- ❖ Diseñar una interfaz agradable y que facilite la interacción entre Usuario-Sistema

2.5 Alcances y Limitaciones

Concretando la información que se presenta en este proyecto, se abarcan los siguientes puntos:

1. Definición del esquema de la base de datos para el sistema de control médico usando el manejador de Base de Datos MySQL 5.0.26
2. Diseño funcional de la interfaz del sistema, de una manera agradable a la vista y de forma simple y entendible al usuario.

Respecto a las limitantes presentadas las principales fueron las siguientes:

- La falta de equipo adecuado, ya que la computadora del departamento médico no presentaba el hardware adecuado para la instalación del sistema.
- La falta de tiempo, con respecto al horario del doctor, ya que la planeación de la interfaz grafica, fue hecho de acuerdo a las necesidades que el doctor planteo y las cuales fueron seguidas al pie de la letra.
- La falta de red adecuada, ya que el departamento se encuentra en otra ala de la institución, donde se maneja una red distinta a la que se necesitaba para la conexión con la base de datos de la Universidad.

CAPITULO 3

3.0 MARCO TEORICO

En este capítulo, se analiza de una manera práctica y general el procedimiento que se lleva a cabo en la Universidad de Sotavento para la alta de un expediente médicos, así como el procedimiento que se realiza para la programación de eventos médicos llevados a cabo por dicha institución.

Así mismo, se redactan los orígenes del lenguaje de programación a utilizar (java) además de sus principales ventajas sobre otros lenguajes de programación existentes. De igual manera se detalla la información más relevante del gestor de Base de Datos MySQL, como lo son las estructuras que todas las tablas deben contener.

3.1 Alta de Expedientes

Cuando un alumno, de la institución se siente mal, es llevado al servicio médico que brinda la institución. El cual se encuentra en la planta baja de edificio dos de la universidad.

- El doctor, toma los datos del alumno de forma manual, es decir en hojas blancas, y son guardados en un folder y archivados.
- Si dicho alumno vuelve a sentirse mal durante las clases, se tiene que buscar entre el expediente en archivo, lo cual es bastante tardado ya que son muchos los archivos que se guardan.
- Del mismo modo, la programación de eventos se realiza de forma manual en hojas blancas y pasadas a la paquetería Excel de office, de la PC.
- El inventario tanto de medicamentos como de Instrumental, se lleva a cabo del mismo modo, en hojas blancas que son archivadas y sin una fecha de actualización establecida.

3.2 Programación en Java

3.2.1 ¿Qué es Java?

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los 90. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en un bytecode que es interpretado (usando normalmente un compilador JIT), por una máquina virtual Java.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos mucho más simple y elimina herramientas de bajo nivel como punteros.

La plataforma Java y el lenguaje Java empezaron como un proyecto interno de Sun Microsystems en diciembre de 1990. Patrick Naughton, ingeniero de Sun, estaba decepcionado con el estado de C++ y la API de C y sus herramientas. Mientras consideraba migrar a NeXT, Naughton recibió la oferta de trabajar en una nueva tecnología, y así comenzó el proyecto *Stealth*.

3.2.2 Orígenes de Java

El Proyecto Stealth fue rebautizado como *Green Project* (o *Proyecto Verde*) cuando James Gosling y Mike Sheridan se unieron a Naughton. Con la ayuda de otros ingenieros, empezaron a trabajar en una pequeña oficina en Sand Hill Road en Menlo Park, California. Intentaban desarrollar una nueva tecnología para programar la siguiente generación de *dispositivos inteligentes*, en los que Sun veía un campo nuevo a explotar.

El equipo pensó al principio usar C++, pero se descartó por varias razones. Al estar desarrollando un sistema empotrado con recursos limitados, C++ no es adecuado por necesitar mayor potencia además de que su complejidad conduce a errores de desarrollo. La ausencia de un recolector de basura (*garbage collector*) obligaba a los programadores a manejar manualmente el sistema de memoria, una tarea peligrosa y proclive a fallos. El equipo también se encontró con problemas por la falta de herramientas portables en cuanto a seguridad, programación distribuida, y programación concurrente. Finalmente abogaban por una plataforma que fuese fácilmente portable a todo tipo de dispositivo.

Bill Joy había concebido un nuevo lenguaje que combinase lo mejor de *Mesa* y C. En un escrito titulado *Further* (más lejos), proponía a Sun que sus ingenieros crearan un entorno *orientado a objetos* basado en C++. Al principio Gosling intentó modificar y ampliar C++, a lo que llamó C++ ++ --, pero pronto descartó la idea para crear un lenguaje completamente nuevo, al que llamó *Oak*, en referencia al roble que tenía junto a su oficina.

El equipo dedicó largas horas de trabajo y en el verano de 1992 tuvieron listas algunas partes de la plataforma, incluyendo el Sistema Operativo Verde, el lenguaje Oak, las librerías y el hardware. La primera prueba, llevada a cabo el 3 de Septiembre de 1992, se centró en construir una PDA (*Personal Digital Assistant* o Asistente Digital Personal) llamada *Star7*, que contaba con una interfaz gráfica y un asistente apodado "Duke" para guiar al usuario.

En noviembre de ese mismo año, el Proyecto Verde se convirtió en **FirstPerson, Inc**, una división propiedad de Sun Microsystems, y el equipo se trasladó a Palo Alto (California). El interés se centró entonces en construir dispositivos interactivos, hasta que Time Warner publicó una solicitud de oferta para un adaptador de televisión. Es decir, un aparato que se sitúa entre la televisión y una fuente de señal externa y que adapta el contenido de ésta (video, audio, páginas Web, etc.) para verse en la pantalla. Entonces, Firstperson cambió de idea y envió a Warner una propuesta para el dispositivo que deseaban. Sin embargo, la industria del cable consideró que esa propuesta daba demasiado control al usuario, con lo que FirstPerson perdió la puja a favor de Silicon Graphics Incorporated. Un trato con la empresa 3DO para el mismo tipo de dispositivo tampoco llegó a buen puerto. Viendo que no había muchas posibilidades en la industria de la televisión, la compañía volvió al seno de Sun.

3.2.3 Características de Java

A continuación se hace una pequeña redacción de las características del lenguaje, que ayudarán a ver para que tipo de problemas está pensado Java:

3.2.3.1 Simple

Es un lenguaje sencillo de aprender. Su sintaxis es la de C++ "simplificada". Los creadores de Java partieron de la sintaxis de C++ y trataron de eliminar de este todo lo que resultase complicado o fuente de errores en este lenguaje.

3.2.3.2 Orientado a Objetos

Posiblemente sea el lenguaje más orientado a objetos de todos los existentes; en Java todo, a excepción de los tipos fundamentales de variables (int, char, long...) es un objeto.

3.2.3.4 Distribuido

Java está muy orientado al trabajo en red, soportando protocolos como TCP/IP, UDP, HTTP y FTP. Por otro lado el uso de estos protocolos es bastante sencillo comparandolo con otros lenguajes que los soportan.

3.2.3.5 Robusto

El compilador Java detecta muchos errores que otros compiladores solo detectarían en tiempo de ejecución o incluso nunca. (ej: `if(a=b) then ...` el compilador Java no nos dejaría compilar este código).

3.2.3.6 Seguro

Sobre todo un tipo de desarrollo: los Applet. Estos son programas diseñados para ser ejecutados en una página web. Java garantiza que ningún Applet puede escribir o leer de nuestro disco o mandar información del usuario que accede a la página a través de la red (como, por ejemplo, la dirección de correo electrónico). En general no permite realizar cualquier acción que pudiera dañar la máquina o violar la intimidad del que visita la página web.

3.2.3.7 Portable

En Java no hay aspectos dependientes de la implementación, todas las implementaciones de Java siguen los mismos estándares en cuanto a tamaño y almacenamiento de los datos. Esto no ocurre así en C++, por ejemplo. En éste un entero, por ejemplo, puede tener un

tamaño de 16, 32 o más bits, siendo lo única limitación que el entero sea mayor que un short y menor que un long int. Así mismo C++ bajo UNIX almacena los datos en formato little endian, mientras que bajo Windows lo hace en big endian. Java lo hace siempre en little edian para evitar confusiones.

3.2.3.8 Arquitectura Neutral

El código generado por el compilador Java es independiente de la arquitectura: podría ejecutarse en un entorno UNIX, Mac o Windows. El motivo de esto es que el que realmente ejecuta el código generado por el compilador no es el procesador del ordenador directamente, sino que este se ejecuta mediante una máquina virtual. Esto permite que los Applets de una web pueda ejecutarlos cualquier máquina que se conecte a ella independientemente de que sistema operativo emplee (siempre y cuando el ordenador en cuestión tenga instalada una máquina virtual de Java).

3.2.3.9 Rendimiento medio

Actualmente la velocidad de procesado del código Java es semejante a la de C++, hay ciertos pruebas estándares de comparación (benchmarks) en las que Java gana a C++ y viceversa.

Esto es así gracias al uso de compiladores *just in time*, compiladores que traduce los bytecodes de Java en código para una determinada CPU, que no precisa de la máquina virtual para ser ejecutado, y guardan el resultado de dicha conversión, volviéndolo a llamar en caso de volverlo a necesitar, con lo que se evita la sobrecarga de trabajo asociada a la interpretación del bytecode.

No obstante por norma general el programa Java consume bastante más memoria que el programa C++, ya que no sólo ha de cargar en memoria los recursos necesario para la ejecución del programa, sino que además debe simular un sistema operativo y hardware virtuales (la máquina virtual). Por otro lado la programación gráfica empleando las librerías Swing es más lenta que el uso de componentes nativos en las interfaces de usuario.

En general en Java se ha sacrificado el rendimiento para facilitar la programación y sobre todo para conseguir la característica de neutralidad arquitectural, si bien es cierto que los avances en las máquinas virtuales remedian cada vez más estas decisiones de diseño.

3.2.3.10 Multithread

Soporta de modo nativo los threads, sin necesidad del uso de de librerías específicas (como es el caso de C++). Esto le permite además que cada Thread de una aplicación java pueda correr en una CPU distinta, si la aplicación se ejecuta en una máquina que posee varias CPU. Las aplicaciones de C++ no son capaces de distribuir, de modo transparente para el programador, la carga entre varias CPU.

3.2.4 Java y las Bases de Datos

Muchos programadores quizás tengan mayor interés en realizar programación basada conjunto a Bases de Datos, pues Java no se queda atrás, Java no implementa Bases de Datos, ya que solo es un lenguaje de programación, pero implementa funciones que permiten al programador realizar conexiones entre la interfaz de usuario y el Gestor de Base de Datos.

Java permite conectarse por medio de puentes JDBC o a través de Driver's a programas gestores de bases de datos, su independencia entre ambos permite al usuario mantener siempre un enfoque, separando el diseño de la Base de Datos y el de la interfaz en dos mundos de pensamientos diferentes el mundo de los datos y el mundo de las interfaces.

Java es orientado a objetos por ende da solidez a la aplicación evitando cortes bruscos del programa y permitiendo continuar de esta manera con la aplicación, lo que permite montar cualquier aplicación con Bases de Datos a través de la red de forma segura y sólida.

3.2.4.1 Java En el PC de escritorio

Aunque cada vez la tecnología Java se acerca más y más al PC de sobremesa, las aplicaciones Java han sido relativamente raras para uso doméstico, por varias razones.

- ❖ Las aplicaciones Java pueden necesitar gran cantidad de memoria física.

- ❖ La Interfaz Gráfica de Usuario (GUI) no sigue de forma estricta la Guía para Interfaces Humana (Human Interface Guidelines), así como tampoco aquella a la que estamos habitualmente acostumbrados. La apariencia de las fuentes no tiene las opciones de optimización activadas por defecto, lo que hace aparecer al texto como si fuera de baja calidad.
- ❖ Las herramientas con que cuenta el JDK no son suficientemente potentes para construir de forma simple aplicaciones potentes. Aunque el uso de herramientas como Eclipse, un IDE con licencia libre de alta calidad, facilita enormemente las tareas de desarrollo.
- ❖ Hay varias versiones del Entorno en Tiempo de Ejecución de Java, el JRE. Es necesario tener instalada la versión adecuada.
- ❖ Las aplicaciones basadas en la Web están tomando la delantera frente a aquellas que funcionan como entidades independientes. Las nuevas técnicas de programación producen aplicaciones basadas en un modelo en red cada vez más potentes.

Sin embargo hay aplicaciones Java cuyo uso está ampliamente extendido, como los NetBeans, el entorno de desarrollo (IDE) Eclipse, y otros programas como LimeWire y Azureus para intercambio de archivos. Java también es el motor que usa MATLAB para el renderizado de la interfaz gráfica y para parte del motor de cálculo. Las aplicaciones de escritorio basadas en la tecnología Swing y SWT (Standard Widget Toolkit) suponen una alternativa a la plataforma .Net de Microsoft.

3.2.4.2 Disponibilidad del JRE de Java

Una versión del JRE (Java Runtime Environment) está disponible en la mayoría de equipos de escritorio. Sin embargo, Microsoft no lo ha incluido por defecto en su sistema operativo, ya que Windows XP fue lanzado en 2001. En el caso de Apple, éste incluye una versión propia del JRE en su sistema operativo, el Mac OS. También es un producto que por defecto aparece en la mayoría de las distribuciones de Linux. Debido a incompatibilidades entre distintas versiones del JRE, muchas aplicaciones prefieren instalar su propia copia del JRE antes que confiar su suerte a la aplicación instalada por defecto. Los desarrolladores de applets de Java o bien deben insistir a los usuarios en la actualización del JRE, o bien desarrollar bajo una versión antigua de Java y verificar el correcto funcionamiento en las versiones posteriores.

3.2.4.3 Historial de versiones

Java ha experimentado numerosos cambios desde la versión primigenia, JDK 1.0, así como un enorme incremento en el número de clases y paquetes que componen la librería estándar.

Desde J2SE 1.4, la evolución del lenguaje ha sido regulada por el JCP ([Java Community Process](#)), que usa *Java Specification Requests* (JSRs) para proponer y especificar cambios en la plataforma Java. El lenguaje en sí mismo está especificado en la *Java Language Specification* (JLS), o Especificación del Lenguaje Java. Los cambios en los JLS son gestionados en [JSR 901](#).

- ❖ **JDK 1.0** ([23 de enero de 1996](#)) — Primer lanzamiento. [comunicado de prensa](#)
- **JDK 1.1** ([19 de febrero de 1997](#)) — Principales adiciones incluidas: [comunicado de prensa](#)
 - una reestructuración intensiva del modelo de eventos AWT (Abstract Windowing Toolkit)
 - clases internas (inner classes)
 - [JavaBeans](#)
 - [JDBC](#) (Java Database Connectivity), para la integración de bases de datos
 - [RMI](#) (Remote Method Invocation)
- **J2SE 1.2** ([8 de diciembre de 1998](#)) — Nombre clave *Playground*. Esta y las siguientes versiones fueron recogidas bajo la denominación **Java 2** y el nombre "J2SE" (Java 2 Platform, Standard Edition), reemplazó a JDK para distinguir la plataforma base de J2EE (Java 2 Platform, Enterprise Edition) y J2ME (Java 2 Platform, Micro Edition). Otras mejoras añadidas incluían: [comunicado de prensa](#)
 - la palabra reservada (keyword) [strictfp](#)
 - [reflexión en la programación](#)
 - la API gráfica ([Swing](#)) fue integrada en las clases básicas
 - la máquina virtual (JVM) de Sun fue equipada con un [compilador JIT](#) (Just in Time) por primera vez
 - [Java Plug-in](#)
 - [Java IDL](#), una implementación de IDL (Interfaz para Descripción de Lenguaje) para la interoperabilidad con [CORBA](#)
 - Colecciones ([Collections](#))
- **J2SE 1.3** ([8 de mayo de 2000](#)) — Nombre clave *Kestrel*.

Los cambios más notables fueron: [comunicado de prensa](#) [lista completa de cambios](#)

- - la inclusión de la máquina virtual de HotSpot JVM (la JVM de HotSpot fue lanzada inicialmente en abril de 1999, para la JVM de J2SE 1.2)
 - RMI fue cambiado para que se basara en CORBA
 - JavaSound
 - se incluyó el Java Naming and Directory Interface (JNDI) en el paquete de librerías principales (anteriormente disponible como una extensión)
 - Java Platform Debugger Architecture (JPDA)

- **J2SE 1.4** (6 de febrero de 2002) — Nombre Clave *Merlin*. Este fue el primer lanzamiento de la plataforma Java desarrollado bajo el Proceso de la Comunidad Java como JSR 59. Los cambios más notables fueron: comunicado de prensalista completa de cambios
 - Palabra reservada assert (Especificado en JSR 41.)
 - Expresiones regulares modeladas al estilo de las expresiones regulares Perl
 - Encadenación de excepciones Permite a una excepción encapsular la excepción de bajo nivel original.
 - non-blocking NIO (New Input/Output) (Especificado en JSR 51.)
 - Logging API (Specified in JSR 47.)
 - API I/O para la lectura y escritura de imágenes en formatos como JPEG o PNG
 - Parser XML integrado y procesador XSLT (JAXP) (Especificado en JSR 5 y JSR 63.)
 - Seguridad integrada y extensiones criptográficas (JCE, JSSE, JAAS)
 - Java Web Start incluido (El primer lanzamiento ocurrió en Marzo de 2001 para J2SE 1.3) (Especificado en JSR 56.)

- **J2SE 5.0** (30 de septiembre de 2004) — Nombre clave: *Tiger*. (Originalmente numerado 1.5, esta notación aún es usada internamente.[4]) Desarrollado bajo JSR 176, Tiger añadió un número significativo de nuevas características comunicado de prensa
 - Plantillas (genéricos) — provee conversion de tipos (type safety) en tiempo de compilación para colecciones y elimina la necesidad de la mayoría de conversion de tipos (type casting). (Especificado por JSR 14.)
 - Metadatos — también llamados anotaciones, permite a estructuras del lenguaje como las clases o los métodos, ser etiquetados con datos adicionales, que

puedan ser procesados posteriormente por utilidades de proceso de metadatos. (Especificado por [JSR 175](#).)

- [Autoboxing/unboxing](#) — Conversiones automáticas entre [tipos primitivos](#) (Como los int) y [clases de envoltura primitivas](#) (Como [Integer](#)). (Especificado por [JSR 201](#).)
- [Enumeraciones](#) — la palabra reservada enum crea una [typesafe](#), lista ordenada de valores (como Dia.LUNES, Dia.MARTES, etc.). Anteriormente, esto solo podía ser llevado a cabo por constantes enteras o clases construidas manualmente (enum pattern). (Especificado por [JSR 201](#).)
- Varargs (número de argumentos variable) — El último parámetro de un método puede ser declarado con el nombre del tipo seguido por tres puntos (e.g. void drawtext(String... lines)). En la llamada al método, puede usarse cualquier número de parámetros de ese tipo, que serán almacenados en un array para pasarlos al metodo.
- [Bucle for](#) mejorado — La sintaxis para el bucle for se ha extendido con una sintaxis especial para iterar sobre cada miembro de un array o sobre cualquier clase que implemente [Iterable](#), como la clase estándar [Collection](#), de la siguiente forma:

```
void displayWidgets (Iterable<Widget> widgets) {  
    for (Widget w : widgets) {  
        w.display();  
    }  
}
```

Este ejemplo itera sobre el objeto Iterable widgets, asignando, en orden, cada uno de los elementos a la variable w, y llamando al método display() de cada uno de ellos. (Especificado por JSR 201.)

Java SE 6 ([11 de diciembre de 2006](#)) — Nombre clave [Mustang](#). Estuvo en desarrollo bajo la [JSR 270](#). En esta versión, Sun cambió el nombre "J2SE" por **Java SE** y eliminó el ".0" del número de versión.[\[5\]](#). Está disponible en <http://java.sun.com/javase/6/>. Los cambios más importantes introducidos en esta versión son:

Incluye un nuevo marco de trabajo y APIs que hacen posible la combinación de Java con lenguajes dinámicos como PHP, Python, Ruby y JavaScript.

Incluye el motor Rhino, de Mozilla, una implementación de Javascript en Java.

Incluye un cliente completo de Servicios Web y soporta las últimas especificaciones para Servicios Web, como JAX-WS 2.0, JAXB 2.0, STAX y JAXP.

Mejoras en la interfaz gráfica y en el rendimiento.

- **Java SE 7** — Nombre clave *Dolphin*. En el año 2006 aún se encontraba en las primeras etapas de planificación. Se espera que su desarrollo dé comienzo en la primavera de 2006, y se estima su lanzamiento para 2008.
 - Soporte para XML dentro del propio lenguaje
 - Un nuevo concepto de superpaquete
 - Soporte para closures
 - Introducción de anotaciones estándar para detectar fallos en el software

Además de los cambios en el lenguaje, con el paso de los años se han efectuado muchos más cambios dramáticos en la librería de clases de Java (*Java class library*) que ha crecido de unos pocos cientos de clases en JDK 1.0 hasta más de tres mil en J2SE 5.0. APIs completamente nuevas, como Swing y Java2D, han sido introducidas y muchos de los métodos y clases originales de JDK 1.0 están desaprobados.

3.3 Filosofía

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Para conseguir la ejecución de código remoto y el soporte de red, los programadores de Java a veces recurren a extensiones como CORBA (Common Object Request Broker Architecture), Internet Communications Engine o OSGi respectivamente.

3.3.1 Orientado a Objetos

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que use estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y librerías de objetos.

3.3.2 Independencia de la plataforma

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Es lo que significa ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run everywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode)—instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (VM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran librerías adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como [GCJ](#). Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

La licencia sobre Java de Sun insiste que todas las implementaciones sean “compatibles”. Esto dio lugar a una disputa legal entre Microsoft y Sun, cuando éste último alegó que la implementación de Microsoft no daba soporte a las interfaces RMI y JNI además de haber añadido características “dependientes” de su plataforma. Sun demandó a Microsoft y ganó por daños y perjuicios (unos 20 millones de dólares) así como una orden judicial forzando la acatación de la licencia de Sun. Como respuesta, Microsoft no ofrece Java con su versión de sistema operativo, y en recientes versiones de Windows, su navegador Internet Explorer no admite la ejecución de applets sin un conector (o plugin) aparte. Sin embargo, Sun y otras fuentes ofrecen versiones gratuitas para distintas versiones de Windows.

Las primeras implementaciones del lenguaje usaba una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase

una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad. Otra técnica, conocida como compilación JIT (Just In Time, o "compilación al vuelo"), convierte el bytecode a código nativo cuando se ejecuta la aplicación. Otras máquinas virtuales más sofisticadas usan una "recompilación dinámica" en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura, el gran número de estas con pequeños errores o inconsistencias llevan a que a veces se parodie el eslogan de Sun, "Write once, run anywhere" como "Write once, debug everywhere" (o "Escríbelo una vez, ejecútalo en todas partes" por "Escríbelo una vez, depúralo en todas partes")

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

3.3.3 El recolector de basura

Un argumento en contra de lenguajes como C++ es que los programadores se encuentran con la carga añadida de tener que administrar la memoria de forma manual. En C++, el desarrollador debe asignar memoria en una zona conocida como *heap* (montículo) para crear cualquier objeto, y posteriormente desalojar el espacio asignado cuando desea borrarlo. Un olvido a la hora de desalojar memoria previamente solicitada, o si no lo hace en el instante

oportuno, puede llevar a una *fuga de memoria*, ya que el sistema operativo piensa que esa zona de memoria está siendo usada por una aplicación cuando en realidad no es así. Así, un programa mal diseñado podría consumir una cantidad desproporcionada de memoria. Además, si una misma región de memoria es desalojada dos veces el programa puede volverse inestable y llevar a un eventual *cuelgue*.

En Java, este problema potencial es evitado en gran medida por el recolector automático de basura (o *automatic garbage collector*). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante una referencia a éste (que, desde un punto de vista de bajo nivel es una dirección de memoria). Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado). Aún así, es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios—es decir, pueden aún ocurrir, pero en un nivel conceptual superior. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos, mayor seguridad y frecuentemente más rápida que en C++.

La recolección de basura de Java es un proceso prácticamente invisible al desarrollador. Es decir, el programador no tiene conciencia de cuándo la recolección de basura tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente.

Debe tenerse en cuenta que la memoria es sólo uno de los muchos recursos que deben ser gestionados.

3.4 Críticas

En 1995 se dijo que Java fue creado para abrir una nueva vía en la gestión de software complejo, y es por regla general aceptado que se ha comportado bien en ese aspecto. Sin embargo no puede decirse que Java no tenga grietas, ni que se adapta completamente a todos los estilos de programación, todos los entornos, o todas las necesidades.

3.4.1 General

- Java no ha aportado capacidades estándares para aritmética en punto flotante. El estándar [IEEE 754](#) para “Estándar para Aritmética Binaria en Punto Flotante” apareció en 1985, y desde entonces es el estándar para la industria. Y aunque la aritmética flotante de Java (*cosa que cambió desde el 13 de Noviembre de 2006, cuando se abrió el código fuente y se adoptó la licencia GPL, aparte de la ya existente*) se basa en gran medida en la norma del IEEE, no soporta aún algunas características. Más información al respecto puede encontrarse en la sección final de enlaces externos.
- A raíz de la naturaleza propietaria de Java, la supuesta inflexibilidad para cambiar, y un creciente estrechamiento de líneas alrededor del sector corporativo, se dice que Java es “el nuevo [COBOL](#)”. Aunque puede ser una afirmación exagerada, hace referencia a una preocupación real sobre el panorama de futuro de Java.
- El recolector de basura de Java sólo gestiona la memoria, pero el instante en que tiene lugar su tarea no puede controlarse manualmente. Por tanto, aquellos objetos que reservan recursos externos deben ser desalojados de memoria a mano (usando el mecanismo del lenguaje “finally”), o deben mantenerse hasta que acabe la ejecución del programa.

3.4.2 El lenguaje

- En un sentido estricto, Java no es un lenguaje absolutamente orientado a objetos, a diferencia de, por ejemplo, [Ruby](#) o [Smalltalk](#). Por motivos de eficiencia, Java ha relajado en cierta medida el paradigma de orientación a objetos, y así por ejemplo, no todos los valores son objetos.
- Por el contrario, los programadores de [C++](#) pueden caer en la confusión con Java, porque en éste los tipos primitivos son siempre variables automáticas, y los objetos siempre residen en el montículo (heap), mientras que en C++ ambos casos están en manos del programador, usando el operador new.
- El código Java puede ser a veces redundante en comparación con otros lenguajes. Esto es en parte debido a las frecuentes declaraciones de tipos y conversiones de tipo manual (casting). También se debe a que no se dispone de operadores sobrecargados, y a una sintaxis relativamente simple. Sin embargo, J2SE 5.0 introduce elementos para tratar de reducir la redundancia, como una nueva construcción para los bucles “foreach”.

- A diferencia de C++, Java no dispone de operadores de sobrecarga definidos por el usuario. Sin embargo esta fue una decisión de diseño que puede verse como una ventaja, ya que esta característica puede hacer los programas difíciles de leer y mantener.
- Java es un lenguaje basado en un solo paradigma.
- Java no permite herencia múltiple como otros lenguajes. Sin embargo el mecanismo de los interfaces de Java permite herencia múltiple de tipos y métodos abstractos.
- El soporte de Java para patrones de texto y la manipulación de éste no es tan potente como en lenguajes como Perl, Ruby o PHP, aunque J2SE 1.4 introdujo las expresiones regulares.

3.4.3 Apariencia

La apariencia externa (el “look and feel”) de las aplicaciones GUI (Graphical User Interface) escritas en Java usando la plataforma Swing difiere a menudo de la que muestran aplicaciones nativas. Aunque el programador puede usar el juego de herramientas AWT (Abstract Windowing Toolkit) que genera objetos gráficos de la plataforma nativa, el AWT no es capaz de funciones gráficas avanzadas sin sacrificar la portabilidad entre plataformas; ya que cada una tiene un conjunto de APIs distinto, especialmente para objetos gráficos de alto nivel. Las herramientas de Swing, escritas completamente en Java, evitan este problema construyendo los objetos gráficos a partir de los mecanismos de dibujo básicos que deben estar disponibles en todas las plataformas. El inconveniente es el trabajo extra requerido para conseguir la misma apariencia de la plataforma destino. Aunque esto es posible (usando GTK+ y el Look-and-Feel de Windows), la mayoría de los usuarios no saben cómo cambiar la apariencia que se proporciona por defecto por aquella que se adapta a la de la plataforma. Mención merece la versión optimizada del Java Runtime que ha desarrollado Apple y que incluye en su sistema operativo, el Mac OS X. Por defecto implementa su propio look-and-feel (llamado Aqua), dando a las aplicaciones Swing ejecutadas en un Macintosh una apariencia similar a la que tendría si se hubiese escrito en código nativo.

3.4.4 Rendimiento

El rendimiento de una aplicación está determinado por multitud de factores, por lo que no es fácil hacer una comparación que resulte totalmente objetiva. En tiempo de ejecución, el

rendimiento de una aplicación Java depende más de la eficiencia del compilador, o la JVM, que de las propiedades intrínsecas del lenguaje. El bytecode de Java puede ser interpretado en tiempo de ejecución por la máquina virtual, o bien compilado al cargarse el programa, o durante la propia ejecución, para generar código nativo que se ejecuta directamente sobre el hardware. Si es interpretado, será más lento que usando el código máquina intrínseco de la plataforma destino. Si es compilado, durante la carga inicial o la ejecución, la penalización está en el tiempo necesario para llevar a cabo la compilación.

Algunas características del propio lenguaje conllevan una penalización en tiempo, aunque no son únicas de Java. Algunas de ellas son el chequeo de los límites de arrays, chequeo en tiempo de ejecución de tipos, y la indirección de funciones virtuales.

El uso de un recolector de basura para eliminar de forma automática aquellos objetos no requeridos, añade una sobrecarga que puede afectar al rendimiento, o ser apenas apreciable, dependiendo de la tecnología del recolector y de la aplicación en concreto. Las JVM modernas usan recolectores de basura que gracias a rápidos algoritmos de manejo de memoria, consiguen que algunas aplicaciones puedan ejecutarse más eficientemente.

El rendimiento entre un compilador JIT y los compiladores nativos puede ser parecido, aunque la distinción no está clara en este punto. La compilación mediante el JIT puede consumir un tiempo apreciable, un inconveniente principalmente para aplicaciones de corta duración o con gran cantidad de código. Sin embargo, una vez compilado, el rendimiento del programa puede ser comparable al que consiguen compiladores nativos de la plataforma destino, inclusive en tareas numéricas. Aunque Java no permite la expansión manual de llamadas a métodos, muchos compiladores JIT realizan esta optimización durante la carga de la aplicación y pueden aprovechar información del entorno en tiempo de ejecución para llevar a cabo transformaciones eficientes durante la propia ejecución de la aplicación. Esta recompilación dinámica, como la que proporciona la máquina virtual HotSpot de Sun, puede llegar a mejorar el resultado de compiladores estáticos tradicionales, gracias a los datos que sólo están disponibles durante el tiempo de ejecución.

Java fue diseñado para ofrecer seguridad y portabilidad, y no ofrece acceso directo al hardware de la arquitectura ni al espacio de direcciones. Java no soporta expansión de código

ensamblador, aunque las aplicaciones pueden acceder a características de bajo nivel usando librerías nativas (JNI, Java Native Interfaces)

3.5 Sintaxis

La sintaxis de Java se deriva en gran medida de C++. Pero a diferencia de éste, que combina la sintaxis para programación genérica, estructurada y orientada a objetos, Java fue construido desde el principio para ser completamente orientado a objetos. Todo en Java es un objeto (salvo algunas excepciones), y todo en Java reside en alguna clase.

Para que esto quede explicado detalladamente, presentare el siguiente ejemplo, el famoso **“Hola Mundo”** que no es mas que un programa que imprime el texto «¡Hola, mundo!» en un dispositivo de visualización (generalmente una pantalla de monitor). Se suele usar como introducción al estudio de un lenguaje de programación, siendo un primer ejercicio típico.

El *Hola Mundo* se caracteriza por su sencillez, especialmente cuando se utiliza en programas de línea de comandos. En interfaces gráficas este programa suele tener una complejidad mayor.

3.5.1 Aplicaciones autónomas

```
// Hola.java
public class Hola
{
    public static void main(String[] args) {
        System.out.println("Hola, mundo!");
    }
}
```

Este ejemplo necesita una pequeña explicación.

- Todo en Java está dentro de una clase, incluyendo programas autónomos.
- El código fuente se guarda en archivos con el mismo nombre que la clase que contienen y con extensión “.java”. Una clase (class) declarada pública (public) debe seguir este convenio. En el ejemplo anterior, la clase es *Hola*, por lo que el código fuente debe guardarse en el fichero “*Hola.java*”

- El compilador genera un archivo de clase (con extensión “.class”) por cada una de las clases definidas en el archivo fuente. Una clase anónima se trata como si su nombre fuera la concatenación del nombre de la clase que la encierra, el símbolo “\$”, y un número entero.
- Los programas que se ejecutan de forma independiente y autónoma, deben contener el método “main()”.
- La palabra reservada “void” indica que el método main no devuelve nada.
- El método main debe aceptar un array de objetos tipo String. Por acuerdo se referencia como “args”, aunque puede emplearse cualquier otro identificador.
- La palabra reservada “static” indica que el método es un método de clase, asociado a la clase en vez de a instancias de la misma. El método main debe ser estático o “de clase”.
- La palabra reservada **public** significa que un método puede ser llamado desde otras clases, o que la clase puede ser usada por clases fuera de la jerarquía de la propia clase. Otros tipos de acceso son “private” o “protected”.
- La utilidad de impresión (en pantalla por ejemplo) forma parte de la librería estándar de Java: la clase “System” define un campo público estático llamado “out”. El objeto out es una instancia de “PrintStream”, que ofrece el método “println(String)” para volcar datos en la pantalla (la salida estándar).
- Las aplicaciones autónomas se ejecutan dando al entorno de ejecución de Java el nombre de la clase cuyo método main debe invocarse. Por ejemplo, una línea de comando (en Unix o Windows) de la forma `java -cp . Hola` ejecutará el programa del ejemplo (previamente compilado y generado “Hola.class”). El nombre de la clase cuyo método main se llama puede especificarse también en el fichero “MANIFEST” del archivo de empaquetamiento de Java (.jar).

3.5.2 Aplicaciones con ventanas

Swing es la librería para la interfaz gráfica de usuario avanzada de la plataforma Java SE.

```
// Hola.java
import javax.swing.*;

public class Hola extends JFrame {
    Hola() {
```

```

        setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        add(new JLabel("Hola, mundo!"));
        pack();
    }

    public static void main(String[] args) {
        new Hola().setVisible(true);
    }
}

```

Las sentencias **import** indican al compilador de Java que las clases e interfaces del paquete **javax.swing** se incluyan en la compilación.

La clase **Hola** extiende (**extends**) la clase **javax.swing.JFrame**, que implementa una ventana con una barra de título y un control para cerrarla.

El constructor **Hola()** inicializa el marco o frame llamando al método **setDefaultCloseOperation(int)** heredado de **JFrame** para establecer las operaciones por defecto cuando el control de cierre en la barra de título es seleccionado al valor **WindowConstants.DISPOSE_ON_CLOSE**. Esto hace que se liberen los recursos tomados por la ventana cuando es cerrada, y no simplemente ocultada, lo que permite a la máquina virtual y al programa acabar su ejecución. A continuación se crea un objeto de tipo **JLabel** con el texto "Hola, mundo!", y se añade al marco mediante el método **add(Component)**, heredado de la clase **Container**. El método **pack()**, heredado de la clase **Window**, es invocado para dimensionar la ventana y distribuir su contenido.

El método **main()** es llamado por la JVM al comienzo del programa. Crea una instancia de la clase **Hola** y hace la ventana sea mostrada invocando al método **setVisible(boolean)** de la superclase (clase de la que hereda) con el parámetro a true. Véase que, una vez el marco es dibujado, el programa no termina cuando se sale del método **main()**, ya que el código del que depende se encuentra en un hilo de ejecución independiente ya lanzado, y que permanecerá activo hasta que todas las ventanas hayan sido destruidas.

3.6 Recursos

3.6.1 JRE

El JRE (Java Runtime Environment, o Entorno en Tiempo de Ejecución de Java) es el software necesario para ejecutar cualquier aplicación desarrollada para la plataforma Java. El usuario final usa el JRE como parte de paquetes software o plugins (o conectores) en un navegador Web. Sun ofrece también el SDK de Java 2, o JDK (Java Development Kit) en cuyo seno reside el JRE, e incluye herramientas como el compilador de Java, Javadoc para generar documentación o el depurador. Puede también obtenerse como un paquete independiente, y puede considerarse como el entorno necesario para ejecutar una aplicación Java, mientras que un desarrollador debe además contar con otras facilidades que ofrece el JDK.

3.6.2 Componentes

- Bibliotecas de Java, que son el resultado de compilar el código fuente desarrollado por quien implementa la JRE, y que ofrecen apoyo para el desarrollo en Java. Algunos ejemplos de estas librerías son:
 - Las bibliotecas centrales, que incluyen:
 - Una colección de bibliotecas para implementar estructuras de datos como listas, arrays, árboles y conjuntos.
 - Bibliotecas para análisis de XML.
 - Seguridad.
 - Bibliotecas de internacionalización y localización.
 - Bibliotecas de integración, que permiten la comunicación con sistemas externos. Estas librerías incluyen:
 - La API para acceso a bases de datos JDBC (Java DataBase Connectivity).
 - La interfaz JNDI (Java Naming and Directory Interface) para servicios de directorio.
 - RMI (Remote Method Invocation) y CORBA para el desarrollo de aplicaciones distribuidas.
 - Bibliotecas para la interfaz de usuario, que incluyen:
 - El conjunto de herramientas nativas AWT (Abstract Windowing Toolkit), que ofrece componentes GUI (Graphical User Interface), mecanismos para usarlos y manejar sus eventos asociados.

- Las Bibliotecas de Swing, construidas sobre AWT pero ofrecen implementaciones no nativas de los componentes de AWT.
- APIs para la captura, procesamiento y reproducción de audio.
- Una implementación dependiente de la plataforma en que se ejecuta de la máquina virtual de Java (JVM), que es la encargada de la ejecución del código de las librerías y las aplicaciones externas.
- Plugins o conectores que permiten ejecutar applets en los navegadores Web.
- Java Web Start, para la distribución de aplicaciones Java a través de Internet.
- Documentación y licencia.

3.6.3 APIs

Sun define tres plataformas en un intento por cubrir distintos entornos de aplicación. Así, ha distribuido muchas de sus APIs (Application Program Interface) de forma que pertenezcan a cada una de las plataformas:

- Java ME (Java Platform, Micro Edition) o J2ME — orientada a entornos de limitados recursos, como teléfonos móviles, PDAs (Personal Digital Assistant), etc.
- Java SE (Java Platform, Standard Edition) o J2SE — para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario medio en un PC de escritorio.
- Java EE (Java Platform, Enterprise Edition) o J2EE — orientada a entornos distribuidos empresariales o de Internet.

Las clases en las APIs de Java se organizan en grupos disjuntos llamados paquetes. Cada paquete contiene un conjunto de interfaces, clases y excepciones relacionadas. La información sobre los paquetes que ofrece cada plataforma puede encontrarse en la documentación de ésta.

El conjunto de las APIs es controlado por Sun Microsystems junto con otras entidades o personas a través del programa JCP (Java Community Process). Las compañías o individuos participantes del JCP pueden influir de forma activa en el diseño y desarrollo de las APIs, algo que ha sido motivo de controversia.

En 2004, IBM y BEA apoyaron públicamente la idea de crear una implementación de código abierto (open source) de Java, algo a lo que Sun, a fecha de 2006, se ha negado.

3.6.4 Extensiones y arquitecturas relacionadas

Las extensiones de Java están en paquetes que cuelgan de la raíz `javax`: `javax.*`. No se incluyen en la JDK o el JRE.

Algunas de las extensiones y arquitecturas ligadas estrechamente al lenguaje Java son:

- [Java EE](#) (Java Platform, Enterprise Edition; antes J2EE) —para aplicaciones distribuidas orientadas al entorno empresarial
- [Java ME](#) (Java Platform, Micro Edition; antes J2ME)—para dispositivos de recursos limitados como teléfonos móviles y PDAs
- [JMF](#) (Java Media Framework)
- [JNDI](#) (Java Naming and Directory Interface)
- [JSML](#) (Java Speech API Markup Language)
- [JDBC](#) (Java Database Connectivity)
- [JDO](#) (Java Data Objects)
- [JAI](#) (Java Advanced Imaging)
- [JAIN](#) (Java API for Integrated Networks)
- [JDMK](#) (Java Dynamic Management Kit)
- [Jini](#) (una arquitectura de red para la construcción de sistemas distribuidos)
- [Jiro](#)
- [Java Card](#)
- [JavaSpaces](#)
- [JML](#) (Java Modeling Language)
- [JMI](#) (Java Metadata Interface)
- [JMX](#) (Java Management Extensions)
- [JSP](#) (JavaServer Pages)
- [JSF](#) (JavaServer Faces)
- [JNI](#) (Java Native Interface)
- [JXTA](#) (Protocolos abiertos para redes virtuales Peer-to-Peer o P2P)
- [Java 3D](#) (Una API de alto nivel para programación gráfica en 3D)
- [JOGL](#) (Java OpenGL—Una API de bajo nivel para programación gráfica usando [OpenGL](#))
- [LWJGL](#) (Light Weight Java Game Library—Una API de bajo nivel para acceso a [OpenGL](#), [OpenAL](#) y varios dispositivos de entrada)
- [OSGi](#) (Dynamic Service Management and Remote Maintenance)

3.7 Java en código abierto

3.7.2 ¿Hasta donde Java es Software libre?

En diciembre de 2006, Sun está en pleno relanzamiento de su plataforma Java bajo la licencia GPL de GNU. Cuando este cambio de licencia haya terminado, esperamos que Java siga siendo un recurso amigable para todo tipo de usuarios.

3.7.3 Compromiso de Sun Microsystems con el código abierto

La importancia del código abierto en relación con Java puede verse entre otras cosas, en que el presidente y CEO de Sun, Jonathan Schwartz, ha retado a la compañía a que ofrezca código abierto para todo el software que produce Sun (fuentes), Sun ya hace mucho tiempo que empezó a apostar por el código abierto cuando liberó StarOffice (Llamado a ahora OpenOffice).

Sun ha aportado más líneas de código abierto que cualquier otra organización (fuentes) También Richard Stallman opina eso mismo (fuentes)

- El éxito del código abierto - Artículo puesto en el sitio oficial en Español de Sun Microsystems, en el que habla de la apuesta de Sun Microsystems por el código abierto.

3.8 ¿Qué es MySQL?

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

3.8.1 Historia de MySQL

MySQL surgió como un intento de conectar el gestor MySQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

Se comenta en el manual [[MySQL Manual](#)] que no se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

La versión estable de este gestor a días de hoy es la 3.23.49. Se puede encontrar más información sobre este gestor en el manual [[MySQL Manual](#)]

3.8.2 Características de MySQL

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
 2. Soporta gran cantidad de tipos de datos para las columnas.
 3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
 4. Gran portabilidad entre sistemas.
 5. Soporta hasta 32 índices por tabla.
 6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- MySQL es un sistema de gestión de bases de datos

Una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compra a una galería de pintura o las más vastas cantidades de información en una red corporativa. Para añadir, acceder, y procesar los datos almacenados en una base de datos, necesita un sistema de gestión de base de datos como MySQL Server. Al ser los computadores muy buenos en tratar grandes cantidades de datos, los sistemas de gestión de bases de datos juegan un papel central en computación, como aplicaciones autónomas o como parte de otras aplicaciones.

- MySQL es un sistema de gestión de bases de datos relacionales

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1986 y existen varias versiones. En este manual, "SQL-92" se refiere al estándar del 1992, "SQL:1999" se refiere a la versión del 1999, y "SQL:2003" se refiere a la versión actual del estándar. Usamos la frase "el estándar SQL" para referirnos a la versión actual de SQL.

- MySQL software es Open Source.

Open Source significa que es posible para cualquiera usar y modificar el software. Cualquiera puede bajar el software MySQL desde internet y usarlo sin pagar nada. Si lo desea, puede estudiar el código fuente y cambiarlo para adaptarlo a sus necesidades. El software MySQL usa la licencia GPL (GNU General Public License), <http://www.fsf.org/licenses/>, para definir lo que puede y no puede hacer con el software en diferentes situaciones. Si no se encuentra cómodo con la GPL o necesita añadir código MySQL en una aplicación comercial, puede comprarnos una licencia comercial.

El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar.

- Seguridad
 - Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.
- Escalabilidad y límites
 - Soporte a grandes bases de datos. Usamos MySQL Server con bases de datos que contienen 50 millones de registros. También conocemos usuarios que usan MySQL Server con 60.000 tablas y acerca de 5.000.000 de registros.

- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna **CHAR**, **VARCHAR**, **BLOB**, o **TEXT**.
- Conectividad
 - Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT,2000,XP, o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros socket Unix.
 - En MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción **--shared-memory**. Los clientes pueden conectar a través de memoria compartida usando la opción **--protocol=memory**.
 - La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity). Por ejemplo, puede usar MS Access para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows o Unix. El código fuente de MyODBC está disponible. Todas las funciones para ODBC 2.5 están soportadas, así como muchas otras. Consulte Sección 25.1, "MySQL Connector/ODBC".
 - La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible. Consulte Sección 25.4, "MySQL Connector/J".
- Localización
 - El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas. Consulte Sección 5.9.2, "Escoger el idioma de los mensajes de error".
 - Soporte completo para distintos conjuntos de caracteres, incluyendo **latin1** (ISO-8859-1), **german**, **big5**, **ujis**, y más. Por ejemplo, los caracteres escandinavos 'â', 'ä' y 'ö' están permitidos en nombres de tablas y columnas. El soporte para Unicode está disponible

- Todos los datos se guardan en el conjunto de caracteres elegido. Todas las comparaciones para columnas normales de cadenas de caracteres son case-insensitive.
- La ordenación se realiza acorde al conjunto de caracteres elegido (usando colación Sueca por defecto). Es posible cambiarla cuando arranca el servidor MySQL. Para ver un ejemplo de ordenación muy avanzada, consulte el código Checo de ordenación. MySQL Server soporta diferentes conjuntos de caracteres que deben ser especificados en tiempo de compilación y de ejecución.

A partir de MySQL 5.0, se usa el motor de almacenamiento **MyISAM**, el máximo tamaño de las tablas es de 65536 terabytes ($256^7 - 1$ bytes). Por lo tanto, el tamaño efectivo máximo para las bases de datos en MySQL usualmente los determinan los límites de tamaño de ficheros del sistema operativo, y no por límites internos de MySQL.

El motor de almacenamiento **InnoDB** mantiene las tablas en un espacio que puede ser creado a partir de varios ficheros. Esto permite que una tabla supere el tamaño máximo individual de un fichero. Este espacio puede incluir particiones de disco, lo que permite tablas extremadamente grandes. El tamaño máximo del espacio de tablas es 64TB.

La siguiente tabla lista algunos ejemplos de límites de tamaño de ficheros de sistemas operativos. Esto es sólo una burda guía y no pretende ser definitiva. Para la información más actual, asegúrese de consultar la documentación específica de su sistema operativo.

Sistema operativo	Tamaño máximo de fichero
Linux 2.2-Intel 32-bit	2GB (LFS: 4GB)
Linux 2.4	(usando sistema de ficheros ext3) 4TB
Solaris 9/10	16TB
Sistema de ficheros NetWare w/NSS	8TB
win32 w/ FAT/FAT32	2GB/4GB
win32 w/ NTFS	2TB (posiblemente mayor)
MacOS X w/ HFS+	2TB

Por defecto, MySQL crea tablas **MyISAM** con una estructura interna que permite un tamaño máximo de unas 4GB.

CAPITULO 4

4.0 DISEÑO DEL PROGRAMA PROPUESTO

Todo departamento dentro de una organización es un punto importante para el buen funcionamiento de la misma. Por ello se necesita que cada uno de los departamentos trabaje de una manera eficaz y sea eficiente; una herramienta para lograrlo es la automatización de la información, conocido como sistema de información. Por ello, en la universidad de Sotavento se ha desarrollado un Sistema Médico.

El sistema médico es una herramienta de gestión para el departamento de servicio médico de la Universidad de Sotavento A.C. Las funciones del sistema médico son las de llevar un control sobre los medicamentos e instrumental, registrar las consultas e historias clínicas de los alumnos que soliciten consultas con el fin de tener una bitácora de consultas; por otra parte tiene la utilidad de programar eventos que sean organizados por dicho departamento y contar con un historial de actividades realizadas o por realizar. Todo esto con el fin de optimizar los procesos que se desarrollan en el departamento antes mencionado.

El Sistema Médico cuenta con los siguientes módulos:

- 1) Expedientes
 - a. Alta de expediente Alumnos
 - b. Alta de expediente Administrativos
 - c. Consulta Médica
- 2) Inventarios
 - a. Inventario de medicamentos
 - b. Inventario de instrumental
- 3) Reportes
 - a. Reporte Inventarios
 - b. Reporte expedientes
 - c. Programación de eventos
 - d. Reportes bitácoras

4.1 DEFINICION DE LAS TABLAS DE LA BASE DE DATOS

La definición de la Base de Datos del Sistema para el control medico es a través de cinco tablas que muestran la relación de los alumnos y administrativos con el consultorio medico. A continuación se presenta la estructura de las tablas utilizadas por SIAU-MEDI:

4.1.1

1) Tabla de expedientes:

<i>Nombre del campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
med_id	Int	Clave del registro, autoincrementable y llave primaria.
med_expediente	int	Numero de expediente del paciente.
med_matricula	Varchar	Matricula del alumno
med_nombre	Varchar	Nombre del paciente.
med_edad	Varchar	Edad del paciente
med_peso	Varchar	Peso del paciente
med_presion_art	Varchar	Presion arterial del paciente
med_alergia	Varchar	Opcional si tiene alergias o no.
med_tipo_alergia	Varchar	El tipo de alergia que padece.
med_tipo_enfermedad	Varchar	Que tipo de enfermedad especial padece.
med_recomendaciones	Varchar	Recomendaciones para tratar al paciente.
med_tipo	Varchar	Si es tipo alumno, administrativo o catedratico.

4.1.2

2) Tabla de consultasmedicas:

<i>Nombre del campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
Consulta_id	Int	Clave del registro, autoincrementable y llave primaria.
Consulta_expediente	Varchar	Numero de expediente del paciente

Consulta_diagnostico	Varchar	Diagnostico de la consulta
Consulta_medicamentos	Varchar	Nombre y cantidad de medicamentos recetados
Consulta_fecha_consulta	Varchar	Fecha de la consulta
Consulta_numero	Varchar	Numero de la consulta.

4.1.3

3) eventos_medico:

<i>Nombre del campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
Eventos_id	Int	Clave del registro, autoincrementable y llave primaria.
Eventos_ciclo	Varchar	Ciclo escolar del evento
Eventos_fecha	Date	Fecha del evento
Eventos_nombre	Varchar	Nombre del evento
Eventos_lugar	Varchar	Lugar del evento
Eventos_hora	Varchar	Hora del evento

4.1.4

4) Inventario_instrumental

<i>Nombre del campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
Instrumental_id	Int	Clave del registro, autoincrementable y llave primaria.
Instrumental_nombre	Varchar	Nombre del Instrumental
Instrumental_tipo	Varchar	Tipo de Instrumental
Instrumental_estado	Varchar	Estado del Instrumental
Instrumental_existencias	int	Numero de existencias del instrumental
Instrumental_fecha_ingreso	date	Fecha en la cual se ingreso el instrumental

4.1.5

5) Inventario_medicinas

<i>Nombre del campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
medicina_id	Int	Clave del registro, autoincrementable y llave primaria.
medicina_nombre	Varchar	Nombre del medicamento
medicina_presentacion	Varchar	Presentacion del medicamento
medicina_tipo_medicina	Varchar	Tipo de Medicamento
medicina_existencias	int	Numero de existencias del medicamento

4.2 CAPTURA DE PANTALLAS DEL SIAU - MEDI

En este apartado se presentan las pantallas con las cuales se lleva a cabo la operación del SIAU, mismas que fueron creadas en JAVA, y están ordenadas de acuerdo a la importancia y el uso que siguen durante la operación del sistema.

4.3 Pantalla Principal

Para iniciar el sistema médico sólo basta con dar doble clic en el acceso directo localizado en el escritorio del ordenador en que se encuentra instalado el sistema médico.

Aparecerá la siguiente ventana:



Figura 1.1. Pantalla inicial del sistema medico

Como se muestra en la figura 1.1, primero se tiene que ingresar el nombre de usuario y contraseña, y a continuación pulsar el botón iniciar sesión.

4.3.1 Problemas que se pueden suscitar al inicio del sistema

Es probable que se presenten problemas al inicio del sistema; estos pueden ser ocasionados por problemas de conexión.

A continuación se presenta una lista con problemas y soluciones:

Tabla 1. Problemas de conexión

Problema	Descripción	Solución
No puede entrar al sistema	No acepta el nombre de usuario y/o contraseña	Asegúrese que el nombre de usuario y/o contraseña este escrito debidamente, ya que el sistema es sensible a letras mayúsculas y minúsculas
Aparece un mensaje de error	Mensaje de error al tratar de conectar con el servidor de base de datos... Contacte al administrador	<ol style="list-style-type: none"> 1. Verifique que esté conectado el cable de red de su equipo, y de su equipo al concentrador. 2. Asegúrese que el concentrador este encendido. 3. Puede que la dirección (host) a la que hace referencia no sea la correcta, para esto tiene que contactar con el administrador de red. 4. Es probable que el servidor este apagado o tenga problemas, tiene que contactar con el administrador de red

4.4 Menú Expedientes

Para acceder al menú expedientes, se tiene que dar un clic en el botón “Bitácora” situado en la parte superior derecha de la ventana principal.

Se abrirá el siguiente menú:



Figura. 2.1. Menú expedientes

El menú está conformado de las siguientes opciones:

- 1) Alta de Expediente Alumnos
- 2) Alta de Expediente Administradores
- 3) Consulta Médica.

Los cuales serán explicados a continuación.

4.4.1 Alta de Expediente Alumnos

Esta opción sirve para generar el expediente de los alumnos que vayan a ser atendidos por el médico de la institución. Al acceder a la opción se visualizará el siguiente formulario:

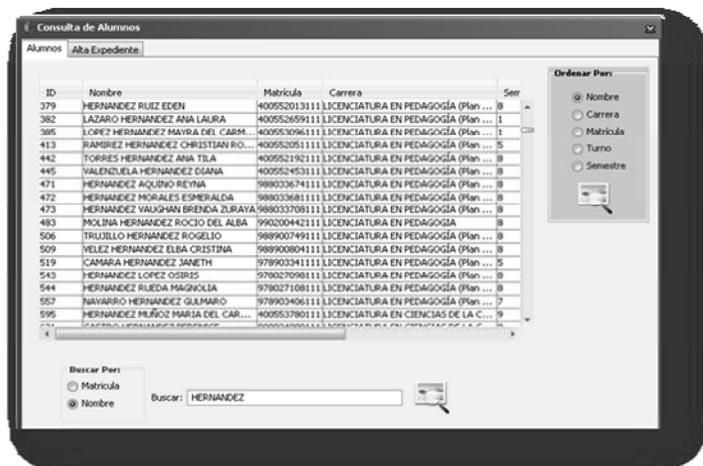


Figura. 2.1.1. Buscar Alumno

1. Seleccionar el método de búsqueda (Matrícula o Nombre) en el recuadro ubicado en la parte inferior derecha de la figura 2.1.1.
2. Ingresar el nombre o matrícula del alumno que sea requerido.
3. Pulsar el botón buscar. Puede ordenar la búsqueda por nombre, carrera, matrícula, turno, o semestre; indicándolo en el recuadro gris ubicado en la parte superior izquierda de la figura 2.1.1.
4. Seleccionar el alumno requerido; dando un clic en el nombre que aparece en la tabla superior.
5. Para generar el expediente, dar clic en la pestaña "Alta Expediente" ubicada en la parte superior derecha de la figura 2.1.1. Se visualizará el siguiente formulario.

Figura 2.1.1 Formulario de alta de expediente medico

6. Se llenan los campos en blanco (si es necesario), y al final se pulsa en botón guardar, ubicado en la parte inferior del formulario.

Nota: Si se guardaron los datos correctamente, tiene que enviar el siguiente mensaje:



Fig. 2.1.3. Mensaje de éxito en generar el expediente

4.4.2 Alta de Expediente Administrativos

Esta opción es idéntica a la anterior, con la diferencia que accede a la base de datos del personal administrativo y profesores que labora en la institución.

Para utilizarlo ver el punto “2.1. Alta de expediente alumnos”, ya que son los mismos pasos.

Consulta Médica

La opción de consulta médica es para realizar una consulta a pacientes que se les haya generado un expediente médico (puntos 2.1, y 2.2). Se pueden hacer varias consultas a la misma persona. Al ingresar a dicha opción se visualiza el siguiente formulario:

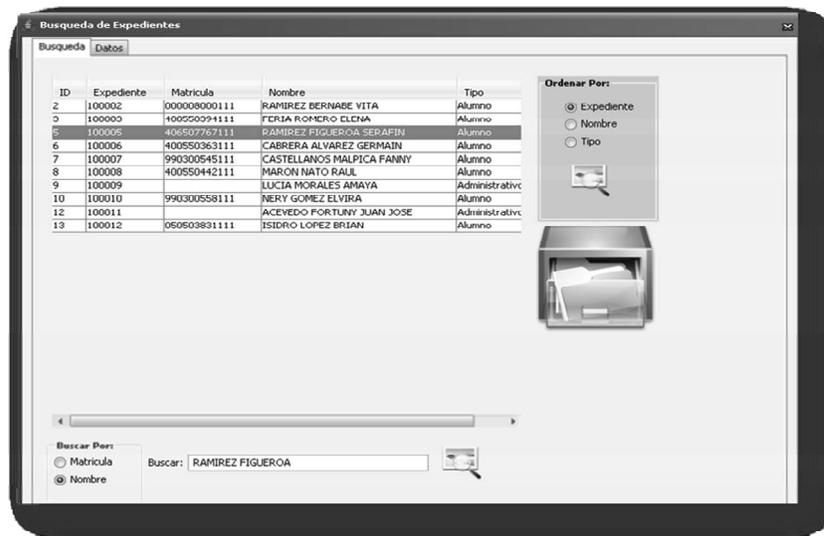


Figura. 2.3.1. Formulario de búsqueda de expedientes

El principio de búsqueda es similar al que se han explicado con anterioridad:

1. Seleccionar el tipo de búsqueda (nombre o matrícula)
2. Ingresar el nombre o matrícula del paciente
3. Pulsar el botón buscar (se puede ordenar la búsqueda por expediente, nombre o tipo de paciente).
4. Seleccionar el nombre del paciente que requiera la consulta
5. Dar clic en la pestaña "Datos" situada en la parte superior derecha de la figura

Nota: Recuerde que en ésta opción sólo aparecen los datos del personal dado de alta previamente en los puntos 2.1 y 2.2. En caso que no se encuentre la persona, debe generar su expediente.

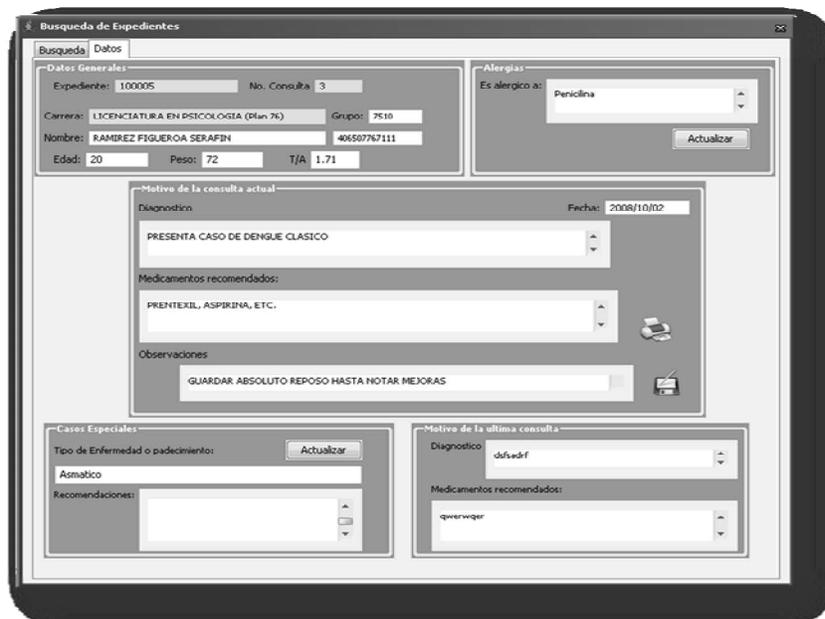


Figura. 2.3.2. Datos de la consulta

El formulario se subdivide en cinco grupos de datos:

- Datos Generales: Contiene los datos del paciente
- Alergias: Muestra las alergias que tiene el paciente, en caso de que se añada otra alergia se pulsa el botón “actualizar”.
- Motivo de la consulta actual:
 - a. Se ingresan el diagnóstico, medicamentos y recomendaciones (Observaciones).
 - b. Guardar datos, dando un click en el botón guardar
 - c. Imprimir receta, dando click en el botón impresora (a lado del cuadro de texto “medicamentos recomendados”.
- Casos especiales: Si se modifican esos datos, o se añaden más se pulsa el botón “Actualizar”.
- Motivo de la última consulta: Aparece el diagnóstico y medicamentos recomendados de la consulta previa (si es primera consulta, los campos estarán en blanco).

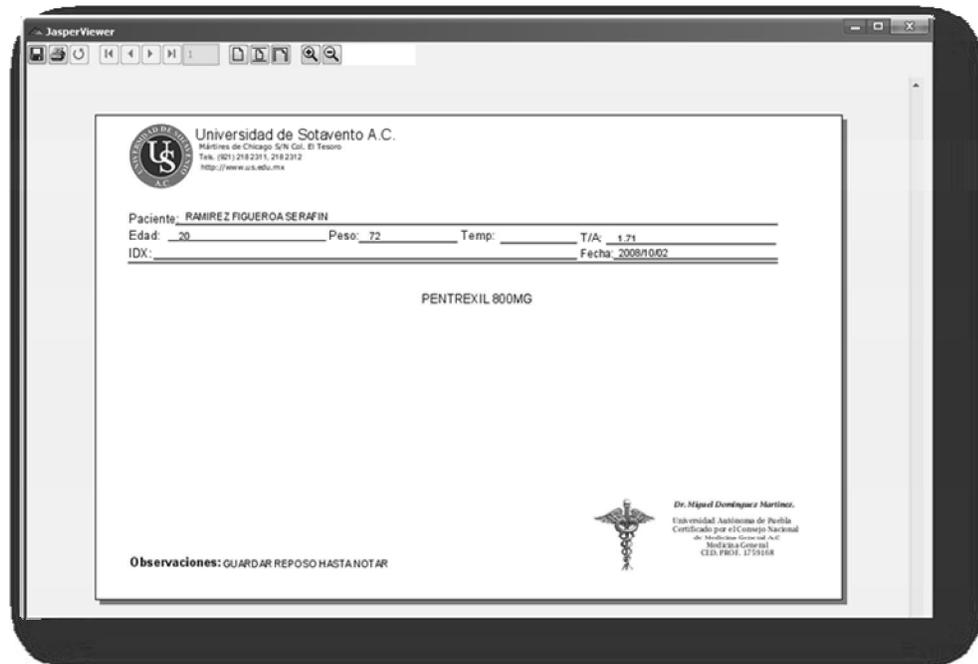


Figura. 2.3.3. Vista de impresión de una receta

4.5 Menú Inventarios



Para acceder al menú inventarios, se tiene que dar un clic en el botón “Inventarios” situado en la parte superior derecha de la ventana principal

Se abrirá el siguiente menú:



Figura. 3.1. Menú Inventario

El menú está conformado de las siguientes opciones:

- 1) Inventario de medicamentos
- 2) Inventario de instrumental.

4.5.1 Inventario Medicamentos

Esta opción sirve para tener un control de los medicamentos existentes o agotados en el departamento de servicios médicos.

Al acceder a esta opción, se visualizará un formulario compuesto de dos pestañas (consulta e ingreso de medicamentos).

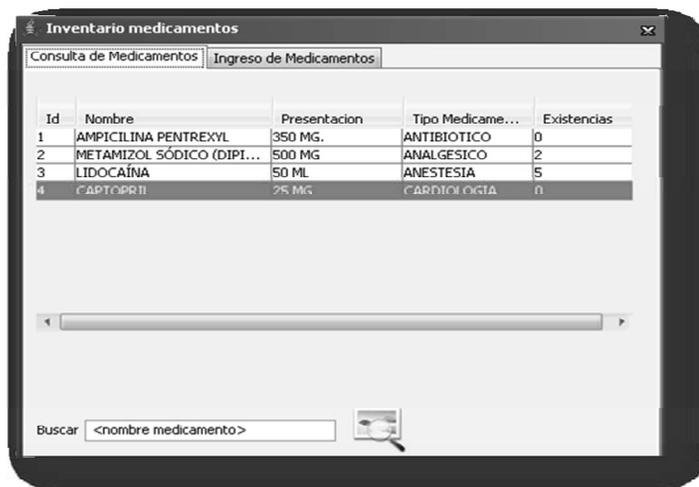


Figura. 3.1.1. Buscar medicamentos

La primera parte (consulta de medicamentos) sirve para buscar los medicamentos dados de alta en el inventario, y poder modificar las existencias del mismo. Para hacer es necesario seguir los siguientes pasos:

1. En la caja de texto “Buscar” (parte inferior derecha del formulario) se escribe el nombre del medicamento. Y enseguida el botón “buscar” (se encuentra a un lado).
2. Seleccionar dando un clic en el nombre del medicamento, que se desea modificar
3. Dar clic en la pestaña “Ingreso de Medicamentos” (parte superior del formulario)

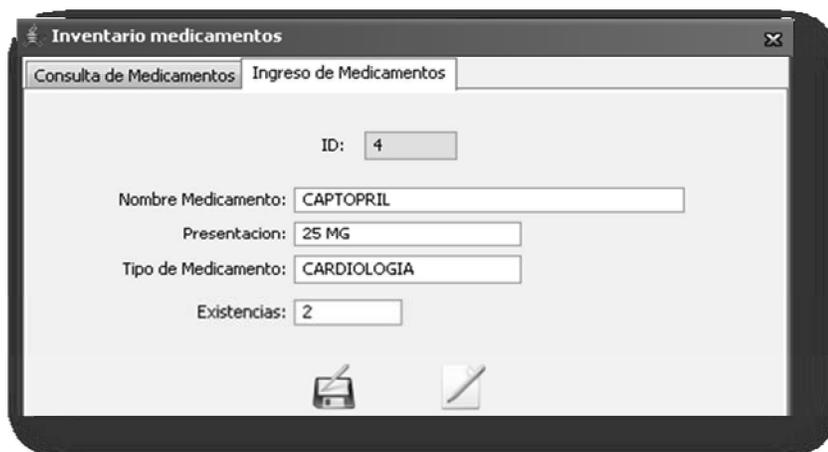


Fig. 3.1.2. Ingreso o modificación de Medicamentos

4. Aumentar o disminuir existencias del medicamento. O modificar los datos que requiera.
5. Pulsar el botón guardar (parte inferior, lado derecho).

Para agregar un nuevo medicamento al inventario:

1. Pulsar el botón “Nuevo” (parte inferior del formulario, lado izquierdo).
2. Ingresar datos correspondientes (Nombre, presentación, tipo y existencia).
3. Guardar cambios, pulsando el botón guardar (parte inferior, lado derecho).

4.5.2 Inventario Instrumental.

Esta opción sirve para controlar los instrumentos médicos con los que cuentas en el departamento de servicios médicos.

Al ingresar en esta opción, se visualiza el siguiente formulario, el cual contiene dos pestañas (Consulta e ingreso de Instrumental).



Id	Nombre	Tipo	Estado
1	ESTETOSCOPIO	QUIRURGICO	SEMINIEVO

Figura. 3.2.1. Lista de instrumentos médicos inventariados

Para modificar los datos de algún instrumento:

1. Seleccionar el instrumento requerido de la lista de inventarios, dando un clic en el nombre del instrumento. Fig. 3.2.1.
2. Dar clic en la pestaña “Ingreso de Instrumental” (parte superior del formulario)

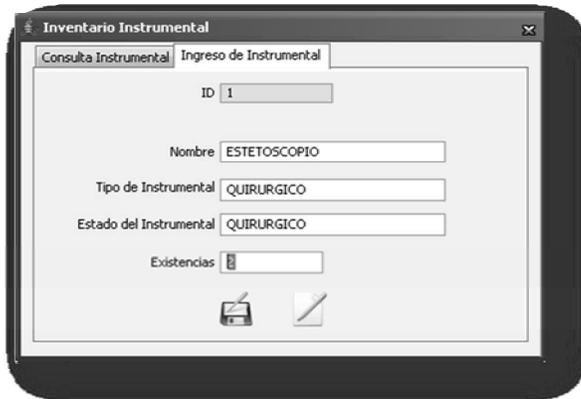


Figura. 3.2.2. Ingreso o modificación del inventario de instrumental

3. Añadir o quitar existencias de instrumentos, o modificar la información deseada.
4. Pulsar el botón guardar.

Para añadir un nuevo instrumento médico:

1. Pulsar el botón “Nuevo” (parte inferior)
2. Ingresar los datos en los campos en blanco.
3. Pulsar el botón guardar.

4.6 Menú Reportes



Para acceder al menú reportes, se tiene que pulsar el botón “Reportes” situado en la parte superior derecha de la ventana principal.

Se abrirá el siguiente menú:



Fig. 4.1. Menú Reportes

El menú reportes contiene las siguientes opciones:

- 1) reporte inventarios
- 2) reporte expedientes
- 3) Programación de eventos
- 4) reportes bitácora

4.6.1 Reporte Inventarios.

La opción de reportes de inventarios es para imprimir el inventario de instrumentos y/o medicamentos que se tienen o no en el departamento de servicios médicos.



Figura. 4.1.1. Formulario de reportes de inventarios.

Para imprimir el inventario de medicamentos, se elige una de las opciones (existentes, agotados o todos) y pulsar el botón “imprimir”.

Para imprimir el inventario de instrumental, sólo pulsar el botón “imprimir”

A continuación ejemplo de impresión de reporte

Nombre del Medicamento	Presentación	Tipo Medicamento	Existencias
1 AMFICILINA PENTREYL	300 MG	ANTIBIOTICO	0

Nombre del Medicamento	Presentación	Tipo Medicamento	Existencias
2 METARAZOL SÓDICO DIFENOXIL	500 MG	ANALGESICO	2
3 LIDOCAINA	50 ML	ANESTESIA	5
4 CAPTOPRIL	25 MG	CARDIOLOGIA	2

Nombre del Medicamento	Presentación	Tipo Medicamento	Existencias
1 AMFICILINA PENTREYL	300 MG	ANTIBIOTICO	0
2 METARAZOL SÓDICO DIFENOXIL	500 MG	ANALGESICO	2
3 LIDOCAINA	50 ML	ANESTESIA	5
4 CAPTOPRIL	25 MG	CARDIOLOGIA	2

Nombre del Instrumental	Tipo	Estado	Existencias
1 ESTETOSCOPIO	QUIRURGICO	SEMINUEVO	2

4.6.2 Reporte Expedientes.

Sirve para imprimir los expedientes médicos de los pacientes:

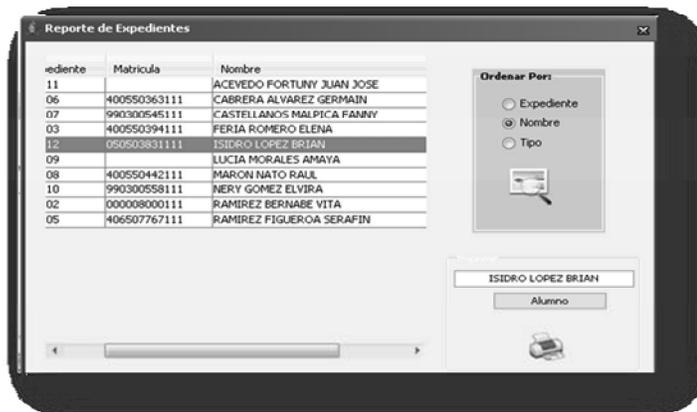


Figura 4.2.1. Reporte de expedientes

Seleccionar al paciente que se desea imprimir (se puede ordenar por expediente, nombre o tipo), luego pulsar el botón imprimir.

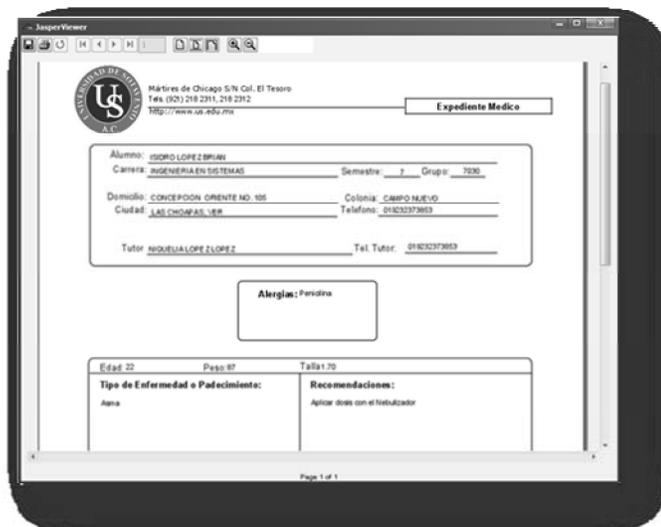


Figura. 4.2.2. Vista de impresión del reporte

4.6.3 Programación de Eventos

La opción programación de eventos está desarrollada para candelarizar eventos que correspondan al departamento de servicios médicos.

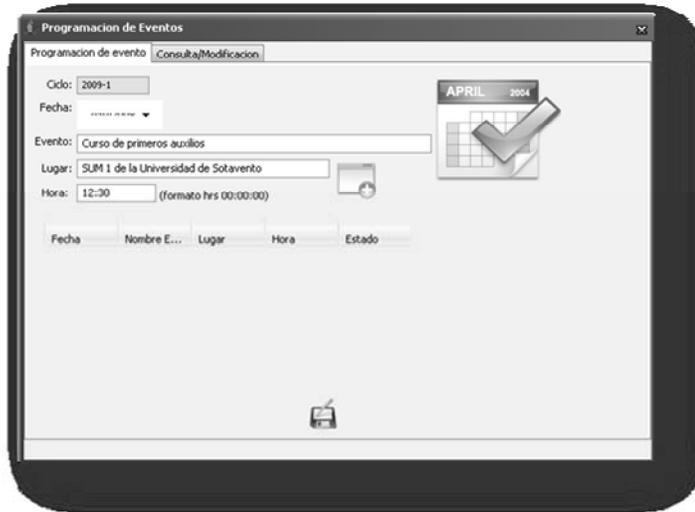


Figura. 4.3.1. Formulario de programación de eventos

El formulario contiene dos pestañas, programación de eventos y Consulta/Modificación. Para programar un evento:

1. Seleccionar la fecha
2. Llenar los datos requeridos
3. Pulsar el botón guardar (parte inferior del formulario).

Para consultar o modificar un evento:

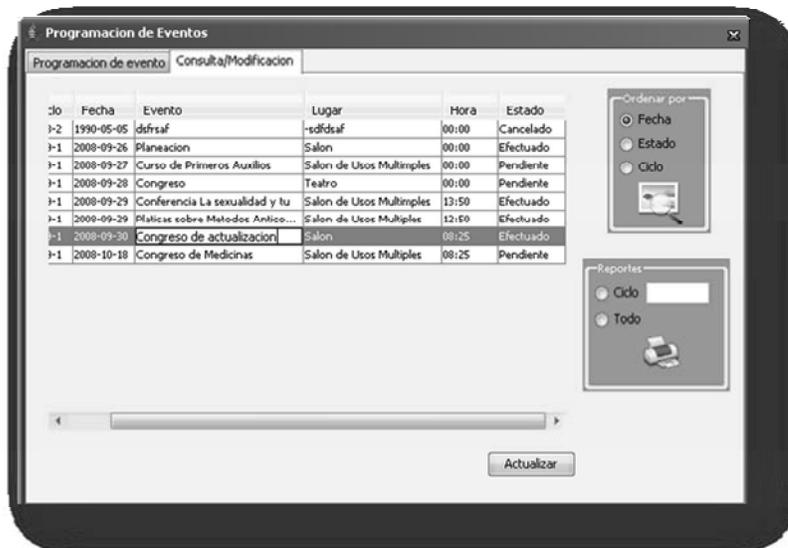


Figura. 4.3.2. Consulta o modificar eventos

1. Seleccionar el evento de la lista fig. 4.3.2. (se pueden ordenar por fecha, estado o ciclo)
2. Hacer cambios pertinentes
3. Pulsar el botón actualizar.

Se puede imprimir los programas de eventos guardados, puede ser, por ciclo o todos.

Ciclo	Fecha	Nombre del Evento	Lugar del Evento	Estado
2008-2	05-may-90	dfrsaf	-sdfdsaf	Cancelado
2009-1	26-sep-08	Planeacion	Salon	Efectuado
2009-1	28-sep-08	Congreso	Teatro	Pendiente
2009-1	27-sep-08	Curso de Primeros Auxilios	Salon de Usos Multiples	Pendiente
2009-1	29-sep-08	Conferencia La sexualidad y tu	Salon de Usos Multiples	Efectuado
2009-1	29-sep-08	Platicas sobre Metodos	Salon de Usos Multiples	Efectuado
2009-1	30-sep-08	Congreso	Salon	Efectuado
2009-1	18-oct-08	Congreso de Medicinas	Salon de Usos Multiples	Pendiente

Figura. 4.3.4. Reporte de programación de eventos

4.6.4 Reporte Bitácora

Esta opción es útil para imprimir el historial de consultas que se han realizado, ya sea por fecha específica o por periodo de fechas.

ID	Expediente	Fecha de Cons...	No. de Cons.
5	100005	2008-09-22	1
7	100002	2008-09-21	1
8	100002	2008-09-22	2
9	100007	2008-09-28	1
10	100007	2008-09-29	2
11	100007	2008-09-29	3
23	100005	2008-10-01	2
25	100010	2008-10-01	1
26	100010	2008-10-01	2
27	100005	2008-10-02	3

Total de Consultas: 10

Figura. 4.4.1. Formulario de reporte de bitácoras

1. Indicar si quieres imprimir por fecha o por periodo.
2. Seleccionar la fecha o periodo de fechas, según sea el caso.
3. Pulsar el botón consultar (situado en la parte superior derecha, de colores).
4. Indicar la opción fecha o periodo.
5. Pulsar el botón imprimir para obtener el reporte.

UNIVERSIDAD DE SONORA
Mártires de Chicago S/N Col. El Tesoro
Tels. (921) 218 2311, 218 2312
http://www.us.edu.mx

Bitacora por Fechas

Periodo comprendido entre: **2008-07-01 y 2008-11-03**

ID	Expediente	Nombre	Fecha de Consulta	No. de Consulta
7	100002	RAMIREZ BERNABE VITA	09/21/2008	1
8	100002	RAMIREZ BERNABE VITA	09/22/2008	2
6	100005	RAMIREZ FIGUEROA SERAFIN	09/22/2008	1
23	100005	RAMIREZ FIGUEROA SERAFIN	10/01/2008	2
27	100005	RAMIREZ FIGUEROA SERAFIN	10/02/2008	3
9	100007	CASTELLANOS MALPICA	09/28/2008	1
10	100007	CASTELLANOS MALPICA	09/29/2008	2
11	100007	CASTELLANOS MALPICA	09/29/2008	3
25	100010	NERY GOMEZ ELVIRA	10/01/2008	1
26	100010	NERY GOMEZ ELVIRA	10/01/2008	2

Total de Consultas: 10

Page 1 of 1

Figura. 4.4.2. Reporte de historial de consultas.

4.7 Perfil Medico

El perfil médico muestra la información general del médico encargado del departamento de servicios médicos de la universidad.

Muestra la siguiente ventana:



4.8 Salir del Sistema

Para salir del sistema sólo basta pulsar el botón



CONCLUSIÓN

La Universidad de Sotavento, A.C. se encarga de prestar servicios de educación y cuenta con un número importante de alumnos matriculados, con los cuales tiene una gran responsabilidad.

Con la finalidad de mejorar la calidad en su servicio, surge la necesidad de apoyarse en las tecnologías de información implementando sistemas que faciliten sus operaciones, llevando registros de las operaciones administrativas que realizan los alumnos.

El hecho de utilizar estos sistemas trae muchos beneficios, ya que al tener la información disponible, esta puede emplearla de maneras que ayuden a realizar su trabajo de una manera más eficiente, ya que cuenta con la posibilidad de presentarla de diversas formas lo cual facilitara su interpretación.

Es por esto que surge la necesidad de proponer la mejora de estos sistemas con la finalidad de seguir aprovechando los recursos ya que con la propuesta de mejora que el presente hace se lograra garantizar el mejoramiento de una de las operaciones administrativas más importantes dentro de la Universidad lo cual lleva implícito el mejoramiento en la oferta de los servicios que esta ofrece.

BIBLIOGRAFIA

<http://java.sun.com>
<http://www.mysql.com>

JAVA como programar
Deitel Harvey M.
Prentice Hall 2004

Análisis y Diseño de Sistemas de información
Senn James A.
McGraw Hill 1992

Java
Deitel Harvey M.
Prentice Hall 1998