

INTRODUCCIÓN

Los grandes avances de la tecnología han logrado darle al hombre una gran variedad de herramientas para alcanzar nuevos niveles en el desarrollo del conocimiento.

La tecnología ha hecho posible la globalización de la sociedad y la información se ha convertido en el eje promotor de cambios sociales, económicos y culturales. Como señala Castells (CASTELLS, M. (1999) - La era de la información. Economía, sociedad y cultura, Vol. 1. La sociedad red, p. 27.): "Una revolución tecnológica, centrada en torno a las tecnologías de la información, está modificando la base material de la sociedad a un ritmo acelerado.

La tecnología y el conocimiento constituyen los ejes centrales del progreso social. La sociedad actual no puede concebirse más que a partir del desarrollo que el conocimiento ha tenido durante estas últimas décadas. Se ha pasado de la era industrial a la de la información. En palabras de Carlota Solé (SOLE, C. (1987) - Ensayos de Teoría Sociológica, Madrid, Paraninfo, 25-26.): "El crecimiento y el desarrollo de una sociedad postindustrial, avanzada o tecnológica, es resultado de un conjunto complejo de factores sociales y no sólo de la acumulación de capital. La innovación, la creatividad, el cambio dependen mucho más directamente que antes del nivel de conocimientos. Información, educación, investigación científica y técnica, formación profesional, capacidad de programar y regular el cambio en las relaciones sociales de una empresa, las formas de management y organización, entre otros, forman parte de los factores de producción".

El conocimiento tecnológico se ha convertido en fuente de progreso económico y de productividad, ya que los modos de incrementar la producción definen, en cada momento, la estructura y dinámica del sistema económico. La tecnología ha

pasado a ser el principal factor de productividad y el instrumento imprescindible para la competitividad y rentabilidad.

La revolución científica y tecnológica y el alto ritmo de cambio tecnológico: nunca antes se había producido tanto conocimiento con capacidad para mejorar la condición humana en las siguientes áreas entre otras (*Biología molécula, Microelectrónica, Nuevas variedades, Plantas Transgénica, Biopesticidas y Agro, TICs y Productividad*). Sin embargo, los indicadores de desarrollo en la mayor parte del mundo no se han mejorado. Por el contrario se han deteriorado por señalar algunas áreas (*Desempleo, Segmento alimentario, Pobreza urbana y rural, Diversos indicadores de bienestar (salud) Desinstitucionalización: Pérdida Identidad Institucional*).

En el mundo se han dado cuenta que es urgente volcar todo este conocimiento en tecnología para desarrollar herramientas que apoyen el proceso educativo, de los estudiantes en cualquier nivel de educación (básica, media o superior), esto con la intención que se desarrollen nuevas habilidades haciendo énfasis en la capacidad de análisis y abstracción de información para poder llegar a soluciones integrales a los nuevos problemas que se enfrenta la sociedad en estos días.

Las entidades educativas han venido resaltando esta falencia en los estudiantes, el sistema educativo actual está llevando a los estudiantes a realizar tareas mecánicas y repetitivas sin llegar a ningún tipo de análisis sobre su proceso educativo.

Este proyecto busca llegar a una posible solución desarrollando un prototipo de software que ayude a identificar las fallas y brinde una nueva plataforma educativa a los estudiantes y profesores de ingeniería que estudien la materia de la estática de la Universidad San Buenaventura.

El proyecto está dividido en 4 fases para su desarrollo: levantamiento de información, análisis, diseño y desarrollo. En la primera etapa de levantamiento de información se debe conseguir un alto volumen de información o documentación sobre estática y el tema de cuerpos rígidos en equilibrio, esto con la intención de tener dominio del tema y asegurar un buen contenido y un buen banco de preguntas para el software.

En el análisis se debe entender cuáles son los alcances del proyecto, identificar las dificultades que se estén presentando durante el proceso de aprendizaje sobre el tema y de esta manera tratar de mejorarlo con el software. También se debe identificar qué tipo de plataforma usar, requerimientos de los equipos, laboratorios disponibles, herramientas de desarrollo.

En el diseño se involucra, el modelado UML, diseño de la base datos, el diseño de la interfaz, documentación y el diseño completo del prototipo con todos los requerimientos necesarios.

En la fase final se hará el desarrollo del prototipo, donde se harán los últimos ajustes del prototipo para su correcto funcionamiento mediante un laboratorio de simulación.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES

El Ministerio de Educación Nacional tiene el proyecto, **Colombia Aprende** el cual es el portal educativo del **Ministerio de Educación Nacional**, creado el 24 de mayo de 2004, como un proyecto estratégico dentro del Programa Nacional de Uso de Medios y Nuevas Tecnologías de la Información y la Comunicación del Plan Sectorial "La Revolución Educativa 2002-2006".

El programa de **uso de nuevas tecnologías para el desarrollo de competencias** es uno de los proyectos estratégicos para la competitividad que lidera el Ministerio de Educación Nacional. La estrategia del programa está fundamentada en esquemas colaborativos, de redes y alianzas estratégicas que se deben dar en diferentes niveles, y cada uno de estos son cruciales en el proceso de apropiación social del conocimiento.

El Programa establece las **líneas de acción** que ayudan en la construcción de una **1. Infraestructura tecnológica de calidad, en el 2. Desarrollo de contenidos de calidad, en la definición de 3. uso y apropiación de las tecnologías en la educación**, y en la consolidación de las comunidades educativas que apropien y pongan en marcha nuevos paradigmas apoyados con tecnología, para dar solidez a las políticas de cobertura, calidad y eficiencia de la **Revolución Educativa** a nivel de la educación preescolar, básica, media y superior.

Durante el 2007 el Gobierno Nacional, a través de los Ministerios de Educación y de Comunicaciones, las Secretarías de Educación, con el apoyo del sector privado, harán un aporte económico por \$349.196 millones, para el fortalecimiento

del Programa Nacional de Uso de Medios y Tecnologías de la Información y la Comunicación-TIC.

Esta inversión está contemplada para conectividad, con \$183.500 millones; dotación, por \$120.000 millones; formación de docentes, en \$40.135 millones; innovación, por \$2.815 millones; y desarrollo de contenidos, por \$2.746 millones.

En esta gran alianza figuran entidades públicas y privadas como la Federación Nacional de Cafeteros, ECOPETROL, Intel, Microsoft, Radio Televisión de Colombia-RTVC, Fundación Telefónica y la Red Latinoamericana de Portales Educativo -RELPE.

Este gran salto e inversión refuerza estrategias que La Revolución Educativa ha desarrollado desde el año 2002, tales como el Programa de Uso de Nuevas Tecnologías para el desarrollo de competencias. Precisamente esta iniciativa es uno de los proyectos estratégicos para la competitividad.

La Universidad San Buenaventura ha venido trabajando en una serie de proyectos pilotos involucrando a estudiantes de ingeniería, para desarrollar temas educativos.

Estos proyectos van dirigidos a desarrollar aplicaciones o soluciones prototipo que ayuden al sector educativo, para evaluar conocimientos, desarrollar nuevas habilidades, competencias y capacidad de análisis de los estudiantes.

Existe otro proyecto llamado **RIBIE**, este es La Red Iberoamericana de Informática Educativa (RIBIE), ha formado parte del Subprograma VII de Electrónica e Informática Aplicadas del CYTED. La integran instituciones y grupos que desarrollan o aplican tecnologías de la información a la solución de problemas educativos. Propicia la comunicación y la colaboración entre sus miembros alrededor de la gestión de proyectos, la formulación de políticas y el desarrollo de estrategias para el mejoramiento de la educación desde la perspectiva de las

ciencias de la información. El Programa Iberoamericano de Cooperación en Ciencia y Tecnológica para el Desarrollo (CYTED) fue creado en 1984, a través de un acuerdo suscrito por 19 países de América Latina, Portugal y España. Su ámbito de actuación es la investigación aplicada, el desarrollo tecnológico y la innovación. Su objetivo es la obtención de resultados transferibles a los sectores productivos y a las políticas sociales para facilitar la modernización y la calidad de vida en la Región. El punto de vinculación de RIBIE-Col con CYTED es COLCIENCIAS.

RIBIE tiene una estructura organizativa en estrella. Hay un nodo central que se integra con los nodos de los países miembros. En el nodo central hay un Coordinador que se encarga de organizar el programa de acción en coordinación con los nodos nacionales y de mantener las relaciones con CYTED. En cada nodo nacional hay una coordinación y un grupo de investigadores en tecnologías de la información aplicadas a la educación y de instituciones que desarrollan iniciativas en su campo de interés. Las coordinaciones en cada país se encargan de identificar los grupos de investigación y coordinar con ellos foros e iniciativas conjuntas y de divulgar los programas de toda la Red.

RIBIE ha representado un importante estímulo al desarrollo de la informática educativa en el país, a través del desarrollo de actividades científicas, cursos, talleres y proyectos de investigación y desarrollo que involucran a los grupos de los países Iberoamericanos.

Cada dos años se celebra un congreso Iberoamericano y en el caso Colombia, un congreso nacional y un foro de investigadores. Estos eventos han sido espacios para exponer y analizar el nivel de desarrollo alcanzado en el campo de la informática educativa, y para estimular iniciativas de trabajo científico. En su

orden, las sedes de los seis congresos Iberoamericanos han sido: Santo Domingo, Lisboa, Barranquilla, Brasilia, Viña del Mar, y Vigo. ¹

Estos proyectos tienen como objetivo analizar las relaciones entre tecnologías de información y comunicación (TICs), sociedad y educación, a partir de la discusión académica, la socialización de investigaciones, experiencias, productos, ambientes, innovaciones, aplicaciones y programas relacionados con la utilización de nuevas tecnologías de la información y la comunicación en procesos de enseñanza-aprendizaje en las diversas disciplinas y áreas curriculares y en programas comunitarios o del sector social.

De igual manera, el congreso también busca contribuir a la formación y consolidación de grupos de investigación, innovación y desarrollo en el área de TICs aplicadas a la educación y a fortalecer sus relaciones con el entorno social.

En este tema de investigación también se encuentra la Universidad Pedagógica Nacional de Colombia con el grupo de investigación **TECNICE**. Ellos han desarrollado el proyecto *“Agentes de software generadores de preguntas y el desarrollo de la competencia cognitiva de hacer inferencias: la comprensión y la inferencia en diferentes niveles de profundidad en ambientes hipermediales”*.

Este proyecto de investigación analiza el efecto del uso de un agente de software generador de preguntas durante la etapa de estudio de un hipertexto en el desarrollo de competencias cognitivas de hacer inferencias a diferentes niveles, por parte de estudiantes de educación media.

En julio de 2006 **RIBIECOL** y la universidad **ICESI** organizaron el *VIII Congreso Colombiano de Informática Educativa* donde se presentaron varios proyectos los cuales estaban clasificados así:

¹ http://ribiecol.org/congreso/menu_horizontal/ribie.jsp.htm, 15 de febrero de 2007, 4:30pm

- ⇒ Cognición, aprendizaje y Currículo
- ⇒ Educación virtual
- ⇒ Informática comunitaria
- ⇒ Instituciones educativas

Dentro de estas categorías se premiaron proyectos como:

⇒ En investigación:

1. **ATENCIÓN Y SENSIBILIDAD RÍTMICA EN NIÑOS CON LIMITACIÓN AUDITIVA.** *UNIVERSIDAD PEDAGÓGICA NACIONAL*

2. **SISTEMA DE CRÉDITOS ACADÉMICOS CON APOYO EN AMBIENTES EDUCATIVOS VIRTUALES PARA LA FORMACIÓN DE INGENIEROS - LECTURA.** *UNIVERSIDAD CENTRAL*

3. **RELACIÓN DE LOS AMBIENTES HIPERTEXTUALES DE APRENDIZAJE GRÁFICO Y SONORO, CON LOS ESTILOS DE APRENDIZAJE.** *UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA*

⇒ En experiencias en informática educativa:

1. **EL PERIÓDICO ESCOLAR VIRTUAL COMO ESTRATEGIA PEDAGÓGICA PARA MEJORAR LAS HABILIDADES COMUNICATIVAS.** *MANUEL JOSE GOMEZ SERNA*

2. INCORPORACIÓN DE LAS TIC EN EDUCACIÓN SUPERIOR.
UNIVERSIDAD DE LOS ANDES

**3. ENSEÑANZA DE LA LECTURA CON APOYO EN AMBIENTES
EDUCATIVOS VIRTUALES. *UNIVERSIDAD CENTRAL***

1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

La Universidad San Buenaventura junto con el grupo de investigación ***Pedagogía y Tecnología*** están trabajando en varios proyectos en donde buscan estudiar nuevas metodologías de aprendizaje, apoyándose en desarrollos de tecnologías de información.

Dentro de los estudios que se han hecho se ha encontrado que la mayoría de los estudiantes usa la memoria mecánica, es decir, memoriza conceptos, formulas, procesos. Las nuevas tendencias en educación tienen el concepto en desarrollar las competencias de los estudiantes para analizar, leer y comprender los problemas para llegar a una conclusión acertada sin la necesidad de memorizar información.

El grupo de investigación lleva otro proyecto que estudia *La complejidad en la solución de problemas*, donde se quiere determinar la incidencia de los niveles de complejidad en la etapa de entrenamiento sobre el aprendizaje en la solución de problemas.

En las investigaciones se han encontrado dificultades en los desarrollos de los prototipos anteriores, algunas de estas fueron, almacenar los datos de los resultados de los estudiantes para posteriormente generar informes para apoyar al docente con el análisis de estos, otra dificultad fue la plataforma que se escogió para trabajar los prototipos. En uno de ellos el docente debía hacer la instalación del prototipo en cada uno de los equipos donde trabajaban los estudiantes y posteriormente pasar nuevamente a recoger los resultados del trabajo lo cual dificulta el proceso de análisis por parte del docente. De acuerdo a esta descripción del problema se ha llegado a la siguiente pregunta:

¿Cómo desarrollar un prototipo de software educativo que cumpla con los requerimientos de recolectar, almacenar y generar informes para facilitar el análisis de los resultados obtenidos por los estudiantes de ingeniería que están trabajando el área de la estática, especialmente en el tema del cuerpo rígido en equilibrio, frente al desarrollo de habilidades de razonamiento deductivo ?

1.3 JUSTIFICACIÓN

Este proyecto busca apoyar al docente a recolectar, almacenar y generar informes sobre el tema del cuerpo rígido en equilibrio usando las tecnologías de la información. El prototipo de software educativo, tomando en cuenta las dificultades en los desarrollos anteriores definitivamente ayudará al docente a realizar los análisis de los resultados de los estudiantes más rápidamente.

Al usar el modelo cliente servidor se asegurará que la instalación del prototipo sea mucho más sencilla y rápida gracias a las características de este modelo, esto también garantiza que se recolectará y almacenarán los datos en tiempo real en una base de datos alojada en el servidor local.

Garantizando estas características se verá muy beneficiado el docente, respecto al tiempo que tomará para realizar sus análisis sobre los informes que genera el prototipo tomando los datos recolectados en las sesiones de trabajo de los estudiantes y al mismo tiempo los estudiantes ya que el docente puede tomar acciones correctivas en su metodología de enseñanza en caso de tener resultados negativos en las sesiones de trabajo con el prototipo.

De esta forma la Universidad de San Buenaventura elevará sus indicadores de calidad, ya que sus estudiantes tendrán una mejor preparación y con unas habilidades más altas para realizar análisis, solución de problemas y toma de decisiones

1.4 OBJETIVOS DE LA INVETIGACIÓN.

1.4.1 Objetivo General

Desarrollar el prototipo de software educativo que presenta inferencias de varios niveles para estudiantes de ingeniería en el área de estática, en el tema específico de cuerpos rígidos en equilibrio.

1.4.2 Objetivos Específicos

- Identificar problemas, y sus causas que se pueden presentar durante el desarrollo del prototipo.
- Seleccionar una arquitectura mediante la cual permita la implementación del prototipo y se ajuste a los requerimientos de éste.
- Realizar el diseño de la Base de Datos para el prototipo de software educativo.
- Establecer métodos de validación del diseño para conseguir los resultados previstos por el grupo pedagogía y tecnología.
- Probar varias interfaces graficas para seleccionar la que mejor permita una navegación fácil y secuencial por todo el contenido del tema cuerpo rígido en equilibrio

- Programar pruebas del prototipo del sistema de evaluación con los estudiantes y docentes de la Universidad para encontrar nuevas fallas y sugerencias a la solución para su implementación

1.5 ALCANCES Y LIMITACIONES DEL PROYECTO

1.5.1 Alcances

Realizar el proceso de levantamiento de información, análisis y diseño para desarrollar el prototipo de software educativo. Este permitirá el entendimiento de cada proceso que realizará el prototipo para su funcionamiento, para localizar, entender y corregir los problemas que se están presentando en el desarrollo de estos.

Este aplicativo tendrá requerimientos específicos tanto para el estudiante, como para el docente.

- Requerimientos del estudiante: Evaluar al estudiante en los tres niveles diferentes de inferencia. Pero separando los niveles independientemente, se debe presentar al estudiante la documentación requerida de forma sencilla y fácil de leer durante la prueba.
- De forma confidencial el prototipo apoyará al docente en: controlar que el estudiante lea la documentación presentada para realizar la prueba, también podrá ver los resultados de los estudiantes discriminados por los diferentes niveles de inferencias.

- ☛ El aplicativo presentará inferencias de los diferentes niveles con base en la documentación previamente presentada al estudiante durante la prueba. El prototipo también controlará el tiempo, esto es, controlar la cantidad de tiempo que utiliza el estudiante en responder, como el tiempo que toma para hacer lectura de la documentación previa.

1.5.2 Limitaciones

Se tendrá en cuenta el tiempo para desarrollar este proyecto, ya que para llevarlo a cabo en su primera parte se debe hacer una investigación y un levantamiento de información profundo para tener un correcto dominio del área de la estática en especial el tema específico del cuerpo rígido en equilibrio, para poder desarrollar contenidos y bancos de inferencias coherentes de esta forma obtener resultados positivos en los laboratorios implementando este prototipo.

2. MARCO REFERENCIAL

2.1 MARCO CONCEPTUAL

Para esta investigación, se deben tener en claro varios conceptos, que darán el inicio conceptual de lo que se quiere y hacia donde se va. Primero se explicarán conceptos sobre metodologías de aprendizaje, también algunos conceptos técnicos para poder tener claridad para el desarrollo de esta investigación.

Las **ontologías como representación del conocimiento** pueden ser conceptualizadas como una taxonomía o como un espacio semántico. Sirve para definir las relaciones entre entidades que conocemos y el lenguaje que utilizamos. Cuando tenemos una ontología, podemos preguntar y dar explicaciones. Dicho de otra forma, la ontología define el conjunto de clases y relaciones jerárquicas, sistémicas, causales, necesarias para comunicarnos y para desarrollar escritos, diagramas o programas de computador.

Siguiendo esta idea debemos comprender qué es una **Inferencia**, la cual es una operación lógica mediante la cual, partiendo de uno o más juicios (premisas), se deriva la validez, la posibilidad o la falsedad de otro juicio. Por lo general los juicios en que se basa un razonamiento expresan conocimientos ya adquiridos o, por lo menos, postulados como hipótesis. Cuando la operación se realiza rigurosamente y el juicio se desprende con necesidad lógica de los juicios antecedentes, el razonamiento recibe el nombre de *inferencia*. Los juicios que sirven como punto de partida se denominan *premisas* y desempeñan la función de ser las condiciones de la inferencia. El resultado que se obtiene, o sea, el juicio inferido como consecuencia, es llamado conclusión.

El prototipo debe funcionar de tal forma que el estudiante después de leer el contenido sobre el equilibrio del cuerpo rígido, pueda hacer un razonamiento o

inferencia cuando se encuentre en el módulo. Dadas estas características este prototipo clasifica en un **software educativo**.

El software educativo generalmente presenta una secuencia (a veces establecida con técnicas de inteligencia artificial) de lecciones, o módulos de aprendizaje. También generalmente incluye métodos de evaluación automática, utilizando preguntas cerradas

2.2 MARCO LEGAL

Este proyecto también está soportado por las siguientes leyes de Derecho de Autor:

CIRCULAR Nro. 6 DERECHOS DE AUTOR EN EL ÁMBITO UNIVERSITARIO

CONSIDERACIONES GENERALES

El derecho de autor busca la protección del ingenio y el talento humano en los dominios literario y artístico, cualquiera que sea su modo o forma de expresión y cualquiera que sea su estilo, tales como:

“... los libros, folletos y otros escritos; las conferencias, alocuciones, sermones y otras obras de la misma naturaleza; las obras dramáticas o dramático _ musicales; las obras coreográficas y las pantomimas; las composiciones musicales con letra o sin ella; las obras cinematográficas, a las cuales se asimilan las obras expresadas por procedimiento análogo a la cinematografía, inclusive los videogramas; las obras de dibujo, pintura, arquitectura, escultura, grabado, litografía; las obras fotográficas a las cuales se asimilan las expresadas por procedimiento análogo a la fotografía; las obras de arte aplicadas; las ilustraciones, mapas, planos, croquis y obras plásticas relativas a la geografía, a la topografía, a la arquitectura o a las ciencias, y, en fin, toda producción del dominio científico, literario o artístico que

pueda reproducirse, o definirse por cualquier forma de impresión o de reproducción, por fonografía, radiotelefonía o cualquier otro medio conocido o por conocer.”²

Este reconocimiento hecho en Colombia a través de la adhesión de nuestro país al Convenio de Berna para la protección de las obras literarias y artísticas (Ley 33 de 1987) y al Tratado de la OMPI sobre derecho de autor (Ley 565 de 2000), la Decisión Andina 351 de 1993 y la Ley 23 de 1982 comporta para los autores de obras literarias y artísticas atributos de orden moral y patrimonial.

Los primeros facultan al autor para reivindicar la paternidad de su obra, oponerse a toda deformación, mutilación u otra modificación de aquella cuando tales actos puedan causar o causen perjuicio a su honor o reputación, o la obra se demerite, a conservar su obra inédita, a modificarla antes o después de su publicación o a retirarla de circulación. Estos derechos tienen un carácter de perpetuos, inalienables e irrenunciables.

En tanto que los segundos, bien llamados derechos patrimoniales o económicos, permiten al autor, mediante el ejercicio de un derecho exclusivo, realizar, autorizar o prohibir la reproducción, comunicación pública, distribución, transformación, o cualquier otra forma de explotación económica de la obra. Estos derechos son limitados en el tiempo y pueden ser restringidos en cuanto permiten, con fines de enseñanza, cultura e información, realizar ciertas utilidades sin que medie la expresa autorización del autor o titular del derecho, ni se tenga que efectuar el pago de remuneración alguna por ellas.

² Constitución Política de Colombia Ley 23 de 1982. Artículo 2º

Trabajos de Grado

Acorde con su competencia, cada institución de educación superior tiene, entre otras, previa aprobación del ICFES, la de expedir los correspondientes títulos de idoneidad en las diferentes ramas del conocimiento y del saber. A este efecto, y en la gran mayoría de los casos, los alumnos al final de su carrera profesional o técnica, deben acreditar los conocimientos adquiridos mediante un trabajo de grado que se puede considerar como una obra literaria o artística (tesis de grado, monografía, el documento que recopila el resultado de una investigación, un programa de computador, una escultura, una composición musical, un audiovisual, etc.). De tal manera, es indispensable tanto para los estudiantes como para las instituciones de educación superior, establecer quién se considera el titular de derechos patrimoniales sobre los trabajos de grado a la luz de lo dispuesto por nuestra legislación sobre derecho de autor.

El derecho de autor es un reconocimiento que el Estado hace a los autores, a través de la Constitución y la Ley, respecto de sus obras literarias y artísticas, al entregarles instrumentos que les permiten reivindicar su condición de titulares sobre las mismas.

Estos derechos surgen en favor del autor sin considerar el fin con cual fue creada la obra, siendo además irrelevante la calidad del creador, es decir, la ley no distingue si es un estudiante, un profesor o un investigador, así como tampoco es preciso establecer dónde tuvo lugar la creación o el tiempo que se haya utilizado, a efectos de esa misma protección.

Así, los derechos de autor sobre una obra literaria o artística, como lo sería un trabajo de grado, son de la persona que la realizó, quien la elaboró imprimiendo todo su ingenio e inteligencia. Es su expresión la que queda plasmada en lo producido, siendo por lo tanto el titular de los derechos morales y patrimoniales de

la creación. En consecuencia, si la obra es realizada por un estudiante, será él, a la luz de la legislación vigente en materia de derecho de autor, el titular de todas las prerrogativas y facultades que la misma concede.

El Director o coordinador del trabajo de Grado

El director de un trabajo de grado es por lo general un profesor de la institución de educación superior, a quien ésta le encomienda la tarea de brindar orientaciones o recomendaciones a uno o más estudiantes, quienes pretendiendo optar por su título profesional deben preparar un escrito o una expresión artística como un plano, una maqueta, una pintura, una composición musical, un audiovisual, etc. Su labor se concreta a señalar parámetros o líneas de investigación que inspiren al estudiante a fin de preparar finalmente su trabajo de grado. De tal forma, se considera como autor de la obra a la persona que expresó y plasmó sus ideas mediante dicho trabajo.

En consideración a ello, el autor único y exclusivo será el estudiante que organizó, recaudó y plasmó toda la información recopilada, incluidas las directrices e ideas planteadas por el director; así, cuando éste proporciona y presenta diferentes opciones al estudiante o corrige dicho trabajo, no hace otra cosa que cumplir con una obligación que le ha encomendado la institución de educación superior a la cual pertenece, sin realizar ninguna expresión literaria o artística.

Es menester señalar que el artículo 6º de la Ley 23 de 1982 y 7º de la Decisión Andina 351 de 1993, consagran el principio universal de “la no protección de las ideas”. Por lo tanto, aún cuando el director realiza una valiosa labor de apoyo al aportar ideas, dicha contribución no está protegida por el derecho de autor.

Obras colectivas o en colaboración

Se entiende por obra colectiva, aquélla realizada por un grupo de autores, por iniciativa y orientación de una persona natural o jurídica que la coordina, divulga o publica bajo su nombre. Por este hecho, la ley concede la titularidad de los derechos patrimoniales, que no son otros que la capacidad para disponer la explotación económica de la obra, a quien encargó o coordinó su realización, conservando los autores los derechos morales, vale decir aquellos que apuntan entre otros a asegurar la paternidad e integridad de la creación.

Es pertinente resaltar, que las obras colectivas creadas dentro de un contrato laboral o de prestación de servicios, en donde sea imposible identificar el aporte individual de cada uno de los autores, tendrán por titular al editor o persona jurídica o natural por cuya cuenta y riesgo se realizaron (Artículos 19, 83 y 92 de la Ley 23 de 1982).

Las obras en colaboración, son aquéllas creadas por dos o más personas naturales teniendo en cuenta una contribución común, donde sus aportes no pueden ser separados sin que la obra pierda su naturaleza (Artículos 18 y 82 de la Ley 23 de 1982).

Así las cosas, cuando la obra es creada por una pluralidad de estudiantes o de profesores, se mantiene el principio general, y todos y cada uno de ellos se consideraran autores de la misma, debiéndose precisar en cada caso si la obra es colectiva o en colaboración en los términos antes esbozados.

Ahora, cuando el director del trabajo de grado y el alumno concretan conjuntamente las ideas, escribiendo cada uno diferentes capítulos de la misma, la calidad de autor se predicará tanto del estudiante como del director, estando frente a la hipótesis de una obra en colaboración.

¿Cuándo la Intuición de educación superior es titular de los derechos patrimoniales sobre una obra?

Como se ha afirmado, el autor de la obra literaria o artística (trabajo de grado) es el titular originario de los derechos patrimoniales que la misma le otorga, y en tal virtud tiene la facultad para disponer de su creación como a bien tenga y a obtener una remuneración por su utilización. Así, para que la institución de educación superior pueda acreditarse como titular de los derechos patrimoniales que nacen de cualquier categoría de obra que realicen los estudiantes o profesores, es necesario que medie un acto de manifestación de la voluntad del autor a través del cual declare su disposición de transferir la titularidad de tales derechos a la correspondiente institución.

En consideración a lo anterior, la institución de educación superior podrá detentar la titularidad de los derechos patrimoniales sobre las obras literarias o artísticas que realicen sus alumnos o profesores, siempre que previa y expresamente se convenga con ellos que los derechos patrimoniales emanados de las mismas son de su propiedad por tratarse de obras por encargo³, en virtud de un contrato donde sea cedido el derecho de autor una vez finalizada la obra, ya sea de manera gratuita u onerosa, o de un contrato laboral que expresamente contenga la obligación para el empleado o trabajador de realizar obras literarias o artísticas para el empleador durante el tiempo de la relación laboral. En todo caso, es necesario que se cumpla con las formalidades previstas en el artículo 183 de la Ley 23 de 1982 que establece:

³ Artículo 20 de la Ley 23 de 1982: "Cuando uno o varios autores, mediante contrato de servicios, elaboren una obra según plan señalado por persona natural o jurídica y por cuenta y riesgo de ésta, solo percibirán, en la ejecución de ese plan, los honorarios pactados en el respectivo contrato. Por este solo acto, se entiende que el autor o autores transfieren los derechos sobre la obra, pero conservarán las prerrogativas consagradas en el artículo 30 de la presente Ley, en sus literales a) y b)".

“Todo acto de enajenación del derecho de autor sea parcial o total, debe constar en escritura pública, o en documento privado reconocido ante notario, instrumentos que, para tener validez ante terceros, deberán ser registrados en la Oficina de Registro de Derechos de Autor, con las formalidades que se establecen en la presente Ley.”

Reprografía de obras literarias

El legislador colombiano mediante la Ley del Libro (Ley 98 de 1993), reiteró un principio tutelar del derecho de autor, consistente en la facultad exclusiva que le asiste al autor o titular del derecho, para controlar todas las formas de utilización de sus obras, conocidas o por conocer, incluida por supuesto la reproducción por cualquier medio.

En tal virtud, el artículo 26 de la disposición mencionada, consagra que: "Todo establecimiento que ponga a disposición de cualquier usuario aparatos para la reproducción de las obras de que trata esta Ley o que efectúe copias que sean objeto de utilización colectiva y/o lucrativa, deberá obtener autorización previa de los titulares de los derechos correspondientes a tales obras, bien sea directamente o bien mediante licencia otorgada por la entidad de gestión colectiva que designe para tal efecto la Cámara Colombiana del Libro".

Por su parte, el artículo 27 de la misma ley establece que " Los autores de obras literarias científicas o culturales conjuntamente con los editores de las mismas, tendrán derecho a participar de una remuneración compensatoria por la reproducción de tales obras al amparo del artículo anterior".

Es menester reiterar que por expresa disposición legal, la autorización deberá obtenerse directamente del titular del derecho o de la sociedad de gestión colectiva⁴ organizada con tal motivo.

Las limitaciones y excepciones al derecho de autor vinculadas con la enseñanza

Constituyen el mecanismo por el cual la legislación prevé la posibilidad de que ciertas utilidades puedan llevarse a cabo sin que medie la autorización del autor y sin que se efectúe por ello el pago de remuneración alguna.

Del conjunto de limitaciones y excepciones vigentes en Colombia a la luz de la Decisión Andina 351 de 1993 y la Ley 23 de 1982, se consideran de aplicación en el ámbito universitario, las siguientes:

1. Reproducir por medios reprográficos para la enseñanza o para la realización de exámenes en instituciones educativas, en la medida justificada por el fin que se persiga, artículos lícitamente publicados en periódicos o colecciones periódicas, o breves extractos de las obras lícitamente publicadas, a condición que tal utilización se haga conforme a los usos honrados y que la misma no sea objeto de venta u otra transacción a título oneroso, ni tenga directa o indirectamente fines de lucro. (Artículo 22 apartado b), Decisión 351 de 1993)

⁴ Las sociedades de gestión colectiva actúan en representación de sus miembros, negocian las tarifas y las condiciones de utilización con los usuarios, otorgan licencias y autorizaciones de uso, y recaudan y distribuyen las regalías, el titular del derecho no participa directamente en ninguna de esas tareas.

2. Reproducir en forma individual, una obra por una biblioteca o archivo cuyas actividades no tengan directa o indirectamente fines de lucro, cuando el ejemplar respectivo se encuentre en la colección permanente de la biblioteca o archivo, y dicha reproducción se realice con los siguientes fines:

- a) Preservar el ejemplar y sustituirlo en caso de extravío, destrucción o inutilización; o,
- b) Sustituir, en la colección permanente de otra biblioteca o archivo, un ejemplar que se haya extraviado, destruido o inutilizado. (Artículo 22 apartado c), Decisión 351 de 1993)

3. Realizar la representación o ejecución de una obra en el curso de las actividades de una institución de enseñanza por el personal y los estudiantes de tal institución, siempre que no se cobre por la entrada ni tenga algún fin lucrativo directo o indirecto, y el público esté compuesto exclusivamente por el personal y estudiantes de la institución o padres o tutores de los alumnos y otras personas directamente vinculadas con las actividades de la institución. (Artículo 22 apartado j), Decisión 351 de 1993).

4. Utilizar obras literarias o artísticas o parte de ellas, a título de ilustración en obras destinadas a la enseñanza, por medio de publicaciones, emisiones de radiodifusión o grabaciones sonoras o visuales, dentro de los límites justificados por el fin propuesto, o comunicar con propósitos de enseñanza la obra radiodifundida para fines escolares, educativos, universitarios y de formación profesional sin fines de lucro, con la obligación de mencionar el nombre del autor y el título de las obras así utilizadas. (Artículo 32 Ley 23 de 1982).

5. Anotar o recoger libremente por los estudiantes a quienes están dirigidas, las conferencias o lecciones dictadas en establecimientos de enseñanza

superior. Pero es prohibida su publicación o reproducción integral o parcial, sin la autorización escrita de quien las pronunció. (Artículo 40 Ley 23 de 1982).

2.3 MARCO TEÓRICO

2.3.1 Ontologías como representación del conocimiento: Una ontología puede ser conceptualizada como una taxonomía o como un espacio semántico. Sirve para definir las relaciones entre entidades que conocemos y el lenguaje que utilizamos. Cuando tenemos una ontología, podemos preguntar y dar explicaciones. Dicho de otra forma, la ontología define el conjunto de clases y relaciones jerárquicas, sistémicas, causales, necesarias para comunicarnos y para desarrollar escritos, diagramas o programas de computador.

Gruber, (1995) y Van Heijst *et al.* (1996) define una ontología como una especificación del nivel de conocimiento explícito en una conceptualización, el cual puede estar afectado por un dominio o campo particular del saber. En esta lógica, la representación formal de un cuerpo de conocimiento se basa en una conceptualización definida a partir de los objetos, los conceptos y relaciones en un campo de interés.

Para la Inteligencia Artificial, lo que existe es lo que puede ser representado. Cuando el conocimiento de un dominio es representado en forma declarativa, el conjunto de objetos que puede ser representado se denomina universo del discurso. Este conjunto de objetos y las relaciones descritas a través de ellos se reflejan en un vocabulario que es representable en un programa. De esta forma, la ontología de un programa se describe a partir de la definición de un conjunto de términos representables. La descripción del significado de los nombres y las

categorías que restringen la interpretación y utilidad de estos términos se construye a partir de definiciones asociadas a nombres de entidades en el universo del discurso, tales como clases, relaciones, funciones u otros objetos. Estos elementos están vinculados al texto leíble por los humanos.

En las ontologías se pueden utilizar tres modelos de representación: jerárquica, sistémica y causal.

2.3.1.1 Representación Jerárquica: Una organización jerárquica facilita la modularidad para permitir describir clases de conceptos utilizando representaciones simples de alto nivel. Las jerarquías son útiles en la medida en que representan y organizan conocimiento en términos de conceptos relacionados. Cada concepto de la jerarquía está relacionado con un concepto de nivel superior que se considera su padre.

2.3.1.2 Representación Sistémica: El pensamiento sistémico contempla el todo y las partes, así como las conexiones entre las mismas, y estudia el todo para poder comprender las partes. Todo sistema se fundamenta en sus interacciones. En consecuencia, las relaciones entre sus partes y su influencia mutua son más importantes que la cantidad de partes que lo constituyen o el tamaño de las mismas.

2.3.1.3 Representación Causal: “La habilidad para inferir relaciones causales es crucial para el proceso de razonamiento, y forma la base para aprender a actuar en el mundo. El conocimiento de relaciones causales nos da un sentido de entendimiento profundo de un sistema y un sentido de control sobre los estados del sistema. El conocimiento de relaciones causales proviene de la habilidad para predecir las consecuencias de acciones que no han sido realizadas (Pearl, 2000).

Las relaciones causa-efecto se infieren de señales observables. De la misma forma, para inferir la estructura de una red de múltiples relaciones causa-efecto, se debe comprender la interacción entre estas dos relaciones.”

En la representación causal, una relación está determinada por la influencia que tiene una entidad sobre otra. Este proceso se dificulta debido a que las relaciones causales no pueden ser observadas directamente, lo cual obliga a un individuo a hacer inferencias a partir de señales o con sus preconceptos. Para inferir la estructura de una red de múltiples relaciones causa-efecto, se considera que debe existir una comprensión de la manera como estas relaciones individuales interactúan.

Una forma de representar las relaciones causales es a través de los grafos dirigidos, donde los nodos representan el estado de las variables de un sistema, y las flechas representan las relaciones causales. La dirección de estos arcos va en el sentido causa-efecto. El origen indica la causa, y el destino el efecto. En los sistemas de representación conceptual, las relaciones causales son aquellas en las que un nodo A determina a otro nodo B, con una relación de causa-efecto.

Ahora se desarrollarán conceptos técnicos los cuales son necesarios para entender como se va desarrollar este proyecto y que herramientas de desarrollo se utilizarán.

2.3.2 Inferencias: El razonamiento es una operación lógica mediante la cual, partiendo de uno o más juicios (premisas), se deriva la validez, la posibilidad o la falsedad de otro juicio. Por lo general los juicios en que se basa un razonamiento expresan conocimientos ya adquiridos o, por lo menos, postulados como hipótesis.

Cuando la operación se realiza rigurosamente y el juicio se desprende con necesidad lógica de los juicios antecedentes, el razonamiento recibe el nombre de *inferencia*. Los juicios que sirven como punto de partida se denominan *premisas* y desempeñan la función de ser las condiciones de la inferencia. El resultado que se obtiene, o sea, el juicio inferido como consecuencia, es llamado *conclusión* (GORTARI, 1982)

Para establecer los niveles de una inferencia se tomará como referente la investigación *Agentes de software generadores de preguntas y el desarrollo de la competencia cognitiva de hacer inferencias: La comprensión y la inferencia en diferentes niveles de profundidad en ambientes hipermediales* desarrollada por MALDONADO, et. al. De allí se toman las siguientes categorías:

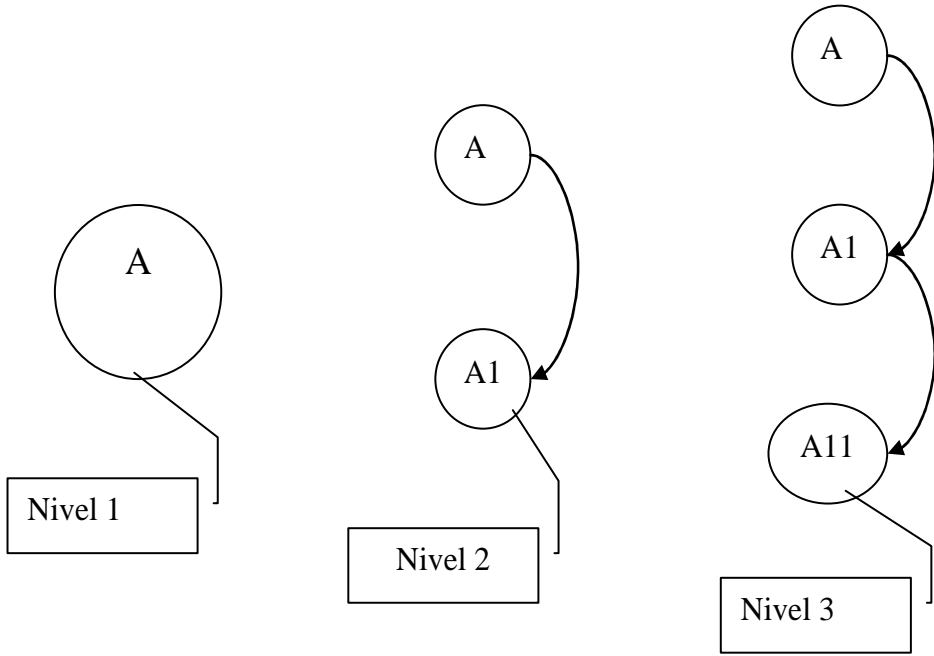
2.3.2.1 Nivel 1 de Inferencia: Para el nivel 1 de inferencia se procede a formular una premisa tomando información del nombre del nodo y de la ranura seleccionada. Luego se puede aplicar un cuantificador, un operador lógico y se pide al usuario que califique como verdadera o falsa la premisa. Para responder tendrá la opción de revisar la información en el nodo correspondiente.

2.3.2.2 Nivel 2 de Inferencia: En el nivel 2 se evalúa la información teniendo en cuenta la categoría superior de la cual el nodo es subcategoría. La premisa se genera a partir de la información del nodo seleccionado y la otra a partir del nodo padre o categoría inmediatamente superior

2.3.2.3 Nivel 3 de Inferencia: En el tercer nivel se selecciona un nodo y se evalúa si tiene un abuelo o categoría con dos niveles de superioridad; luego se

genera una premisa para cada uno de estos nodos y se pide al usuario que las califique como verdaderas o falsas. (Ver Figura 1)

Figura 1. Niveles de Inferencia



2.3.3 Didáctica de la física (mecánica): “Según Ausubel, los problemas generados por la enseñanza tradicional no se deben tanto a su enfoque expositivo como al inadecuado manejo de los procesos de aprendizaje de los alumnos, por lo que, para fomentar la comprensión, o en su terminología un *aprendizaje significativo*, no hay que recurrir tanto al descubrimiento como a mejorar la eficacia de las exposiciones. Para ello hay que considerar no sólo la lógica de las disciplinas sino también la lógica de los alumnos. De hecho para Ausubel el aprendizaje de la ciencia consiste en *transformar el significado lógico en significado psicológico*, es decir en lograr que los alumnos asuman como propios los significados científicos. Para ello la estrategia didáctica deberá consistir en un acercamiento progresivo de las ideas de los alumnos a los conceptos científicos...”⁵

2.3.4 Software Educativo: De acuerdo con Galvis⁶ hay tres tipos de entornos computarizados de aprendizaje: los que se basan en herramientas de “productividad” como Office, en los que los profesores innovadores transforman la función de la herramienta en un vehículo de desarrollo, constructorista, comunicacional, en el que se posibilita el aprendizaje colaborativo, etc. Otro tipo de entorno surge del trabajo en redes, sobre todo en la Internet, que puede enriquecer el aprendizaje a través del contacto con otros alumnos y profesores o profesionales de un campo de conocimiento determinado, y el acceso a bancos de información o sitios web especializados en las más diversas temáticas. Sin embargo, el tercer tipo de entorno es el que más se acerca al ideal de uso de esta tecnología. A decir de Galvis

⁵ POZO MUNICIO, Juan Ignacio. Aprender y enseñar ciencia. Ediciones Morata. Madrid. 1998. Pág 280.

⁶ GALVIS PANQUEVA, Alvaro. “Micromundos lúdicos interactivos: aspectos críticos en su diseño y desarrollo”, en Tecnología y Comunicación Educativas, Año 12, Nº 28, pp. 29-39. (1998)

“...creo que los educadores tenemos mucho que decir en algo que, hasta ahora, ha sido casi un dominio comercial: ingeniería de software educativo. Estoy pensando en la construcción de ambientes educativos computarizados en los que hay mucho más que un buen uso educativo de soluciones informáticas genéricas. Me centro en los materiales educativos computarizados (MECs) que, por su misma naturaleza, están pensados para servir de entorno de aprendizaje y que han sido elaborados con propósito educativo específico. Mi experiencia me lleva a afirmar que este es un nuevo dominio, de carácter inter y transdisciplinario en el que no basta con saber hacer software y darle un toque educativo o tener grandes ideas educativas y hallarles un soporte informático. En este nuevo dominio de trabajo colaborativo entre educadores, expertos en informática y en comunicación y arte, toman importancia cada uno de los saberes de cada quien, integrando lo educativo, lo informático y lo comunicacional en micromundos interactivos, ojalá de carácter lúdico.”⁷

2.3.5 Sistema de Información: Un sistema de información es el conjunto de recursos que permiten recoger, gestionar, controlar y difundir la información de toda una empresa u organización.

Un sistema de información está formado por los siguientes componentes:

- La base de datos.
- Los programas de aplicación.
- Los dispositivos físicos (estaciones de trabajo, dispositivos de almacenamiento, etc.).
- El personal que utiliza y que desarrolla el sistema.

⁷ Ibid. Pag 32 - 33

La base de datos es un componente fundamental de un sistema de información. El ciclo de vida de un sistema de información está ligado al ciclo de vida del sistema de base de datos sobre el que se apoya. Al ciclo de vida de los sistemas de información también se le denomina ciclo de vida de desarrollo del software. Las etapas típicas del ciclo de vida de desarrollo del software son las descritas anteriormente para el desarrollo de este software de gestión. Este ciclo de vida hace énfasis en la identificación de las funciones que realiza la empresa y en el desarrollo de las aplicaciones que lleven a cabo estas funciones. Se dice que el ciclo de vida de desarrollo del software sigue un enfoque orientado a funciones, ya que los sistemas se ven desde el punto de vista de las funciones que llevan a cabo. Por esta razón, el análisis estructurado hace énfasis en los diagramas de flujo de datos, siguiendo el movimiento de los datos a través de una secuencia de transformaciones, y refinando éstas a través de una serie de niveles. Lo mismo ocurre en el diseño estructurado, que ve a un sistema como una función que se descompone sucesivamente en niveles o sub-funciones.

Concentrándose en las funciones se infravaloran los datos y, en especial, la estructura de los datos que son manipulados por las funciones. El resultado es que estos sistemas tienen valor durante poco tiempo en relación con las necesidades de los usuarios a largo plazo. Esto sucede debido a que al poco tiempo de haber instalado un sistema, las funciones implementadas son en realidad un subconjunto de las funciones que los usuarios realmente desean. Casi inmediatamente, los usuarios descubren una gran variedad de servicios adicionales que quisieran incorporar al sistema.

Estas necesidades causan problemas a los sistemas obtenidos con un diseño orientado a funciones, puesto que este diseño puede requerir una revisión importante para acomodar las funciones adicionales.

En contraste, el enfoque orientado a datos centra el foco de atención en el análisis de los datos utilizados por las funciones. Esto tiene dos ventajas. La primera es que los datos son una parte considerablemente más estable que las funciones. La segunda ventaja es que la propia estructura de un esquema de base de datos requiere de un análisis sofisticado de los datos y de sus relaciones. Una vez que se haya construido un esquema para la base de datos que sea lógico, podrían diseñarse tantas funciones como fuera necesario para sacar provecho del mismo. Sin embargo, sin un esquema tal, la base de datos sólo podría ser útil para una única aplicación. Por lo tanto, el enfoque orientado a funciones puede ser bueno para el desarrollo a corto plazo, pero pierde su valor real a largo plazo. Usando un enfoque orientado a datos, los datos pasan a ser los cimientos sobre los cuales se puede construir una gran variedad de funciones diferentes.

2.3.6 Fases del desarrollo de software⁸: En la ingeniería del software el término fases de desarrollo expresa cómo ha progresado el desarrollo de un software y cuánto desarrollo puede requerir. Cada versión importante de un producto pasa generalmente a través de una etapa en la que se agregan las nuevas características (etapa alfa), después una etapa donde se eliminan errores activamente (etapa beta), y finalmente una etapa en donde se han quitado todos los bugs importantes (etapa estable). Las etapas intermedias pueden también ser reconocidas. Las etapas se pueden anunciar y regular formalmente por los desarrolladores del producto, pero los términos se utilizan a veces de manera informal para describir el estado de un producto. Normalmente muchas compañías usan nombres en clave para las versiones antes del lanzamiento de un producto, aunque el producto y las características reales son raramente secretas.

⁸ Fases de desarrollo de Software, http://es.wikipedia.org/wiki/Fases_del_desarrollo_de_software, 05 de Mayo de 2007, 5:31 p.m.

2.3.6.1 Pre-alfa: La fase conocida como pre-alfa se publica a veces antes del lanzamiento de una versión alfa o beta. En contraste con la versión alfa y las versiones beta, la pre-alfa no tiene sus características completas. Los diseñadores todavía están determinando en esta etapa exactamente qué funcionalidades debe tener el producto. Tales etapas se pueden llamar también *development releases* o *nightly builds*.

2.3.6.2 Alfa: La versión alfa de un producto es la primera para la que el equipo de desarrollo decide que implementa todas las funcionalidades especificadas en los requisitos. Es la primera versión del programa que se envía a los verificadores para probarla.

Algunos equipos de desarrollo utilizan el término alfa informalmente para referirse a una fase donde un producto todavía es inestable, aguarda todavía a que se eliminen los errores o a la puesta en práctica completa de toda su funcionalidad, pero satisface la mayoría de los requisitos. El nombre se deriva de alfa, la primera letra en el alfabeto griego.

2.3.6.3 Beta: Una versión beta o lanzamiento beta representa generalmente la *primera versión completa* del programa informático o de otro producto, que es probable que sea inestable pero útil para que las demostraciones internas y las inspecciones previas seleccionen a clientes. Algunos desarrolladores se refieren a esta etapa como *inspección previa (preview)* o como una *inspección previa técnica (technical preview [TP])*. Esta etapa comienza a menudo cuando los desarrolladores anuncian una congelación de las características del producto, indicando que no serán agregadas más características a esta versión y que solamente se harán pequeñas ediciones o se corregirán errores. Las versiones beta están en un paso intermedio en el ciclo de desarrollo completo. Los desarrolladores las lanzan a un grupo de probadores beta o betatesters (a veces

el público en general) para una prueba de usuario. Los probadores divulgan cualquier error que encuentran y características, a veces de menor importancia, que quisieran ver en la versión final.

Cuando una versión beta llega a estar disponible para el público en general que a menudo es utilizada extensamente por los tecnológicamente expertos o familiarizados con versiones anteriores, como si el producto estuviera acabado. Generalmente los desarrolladores de las versiones betas del software gratuito o de código abierto los lanzan al público en general, mientras que las versiones beta propietarias van a un grupo relativamente pequeño de probadores.

2.3.6.4 Versión candidata a definitiva: El término candidata a definitiva o candidata para el lanzamiento (si traducimos más literalmente desde el término en inglés, *release candidate*) se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan. En esta fase el producto implementa todas las funciones del diseño y se encuentra libre de cualquier error que suponga un punto muerto en el desarrollo. Microsoft utiliza frecuentemente este término. Otros términos relacionados incluyen gamma, delta (y tal vez más letras griegas) para versiones que están prácticamente completas pero todavía en pruebas; y omega para versiones que se creen libres de errores y se hallan en el proceso final de pruebas. Gamma, delta y omega son, respectivamente, la tercera, cuarta y última letras del alfabeto griego.

2.3.6.5 Versión de disponibilidad general: La versión de disponibilidad general (también llamada "dorada") de un producto es su versión final. Normalmente es casi idéntica a la versión candidata final, con sólo correcciones de último momento. Esta versión es considerada muy estable y relativamente libre de errores con una calidad adecuada para una distribución amplia y usada por usuarios finales. En

versiones comerciales, puede estar también firmada (usado para que los usuarios finales verifiquen que el código no ha sido cambiado desde su salida. La expresión de que un producto "se ha dorado" significa que el código ha sido completado y que "está siendo producido masivamente y estará en venta próximamente".

El término "dorado" se refiere anecdóticamente al uso del "disco maestro de oro" que fue frecuentemente usado para enviar la versión final a los fabricantes que lo usan para producir las copias de venta al detalle. Esto puede ser una herencia de la producción musical. En algunos casos, sin embargo, el disco maestro está realmente hecho de oro, tanto por apariencia estética como por resistencia a la corrosión.

Microsoft y otros usan el término "release to manufacturing" (RTM) para referirse a esta versión (para productos comerciales como Windows XP, tal como "Build 2600 is the Windows XP RTM release"), y "release to Web" (RTW) para productos libremente descargables.

2.3.7 Bases de Datos⁹: Una **base** o **banco de datos** es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

⁹ Bases de Datos http://es.wikipedia.org/wiki/Base_de_datos, 05 de Marzo de 2007,7:30 a.m.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Tipos de bases de datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

2.3.7.1 Según la variabilidad de los datos almacenados

- **Bases de datos estáticas:** Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.
- **Bases de datos dinámicas:** Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, entre otros.

2.3.7.2 Según el contenido

- **Bases de datos bibliográficas:** Solo contienen un surrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo. Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

➔ **Bases de datos de texto completo:** Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

➔ **Bases de datos o "bibliotecas" de información Biológica:** Son bases de datos que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:

- Aquellas que almacenan secuencias de nucleótidos o proteínas.
- Las bases de datos de rutas metabólicas
- Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas
- Bases de datos clínicas
- Bases de datos bibliográficas (biológicas)

2.3.7.3 Modelos de bases de datos: Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de *base de datos*; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

- **Bases de datos jerárquicas:** Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

- **Base de datos de red:** Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

- **Base de datos relacional:** Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos

dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por *registros* (las filas de una tabla), que representarían las tuplas, y *campos* (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

Durante los años '80 (1980-1989) la aparición de dBASE produjo una revolución en los lenguajes de programación y sistemas de administración

de datos. Aunque nunca debe olvidarse que dBase no utilizaba SQL como lenguaje base para su gestión.

- **Bases de datos orientadas a objetos:** Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

Se está trabajando en **SQL3**, que es el estándar de SQL92 ampliado, que soportará los nuevos conceptos orientados a objetos y mantendrá compatibilidad con SQL92.

- ⇒ **Gestión de bases de datos distribuida:** La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas.

2.3.8 Arquitectura o topología de red: Es la disposición física en la que se conectan los nodos de una red de ordenadores o servidores, mediante la combinación de estándares y protocolos.

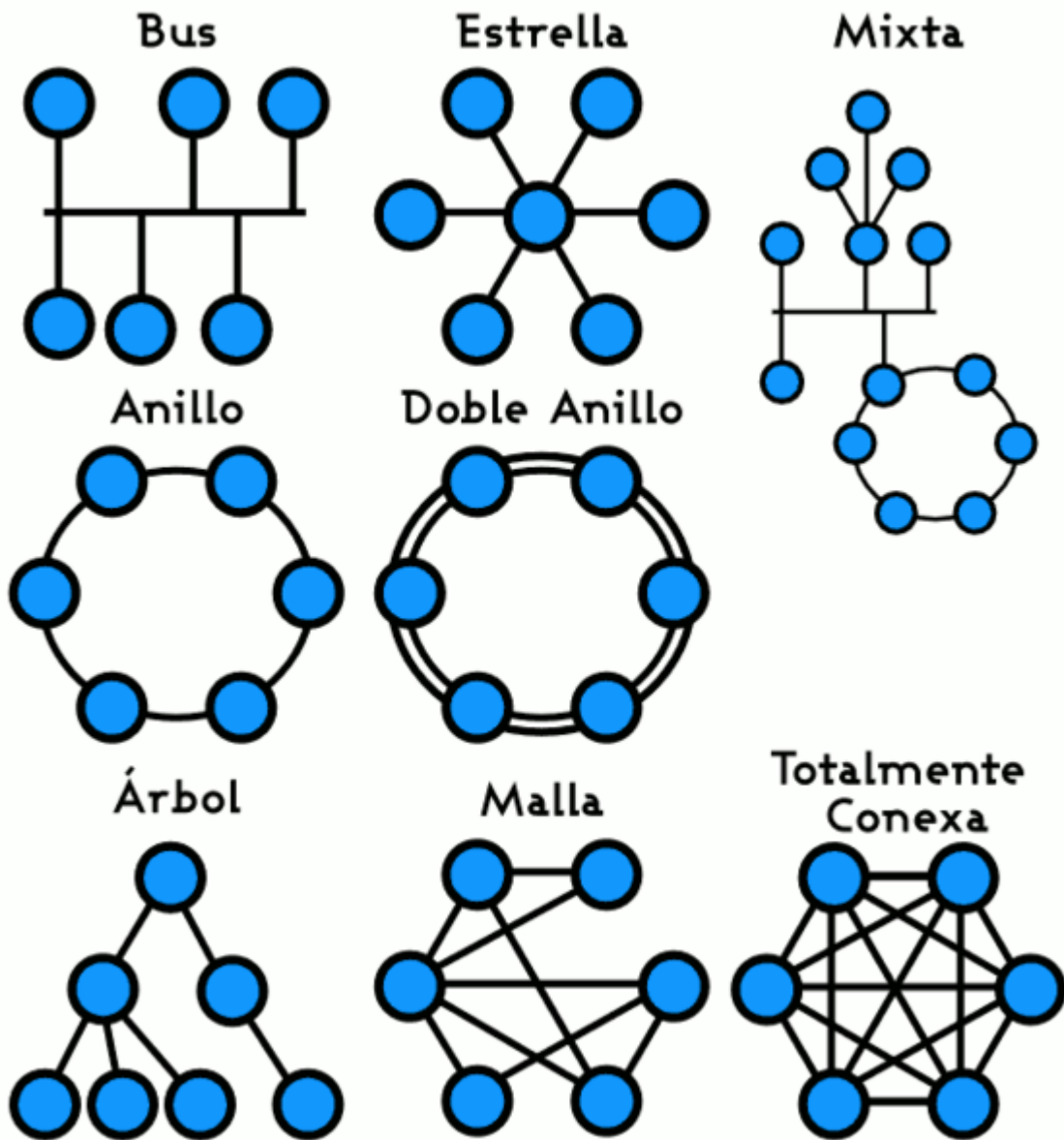
Define las reglas de una red y cómo interactúan sus componentes. Estos equipos de red pueden conectarse de muchas y muy variadas maneras. La conexión más simple es un enlace unidireccional entre dos nodos. Se puede añadir un enlace de retorno para la comunicación en ambos sentidos. Los cables de comunicación modernos normalmente incluyen más de un cable para facilitar esto, aunque redes muy simples basadas en buses tienen comunicación bidireccional en un solo cable.

En casos mixtos se puede usar la palabra arquitectura en un sentido relajado para hablar a la vez de la disposición física del cableado y de como el protocolo considera dicho cableado. Así, en un anillo con una MAU podemos decir que tenemos una topología en anillo, o de que se trata de un anillo con topología en estrella.

La topología de red la determina únicamente la configuración de las conexiones entre nodos. La distancia entre los nodos, las interconexiones físicas, las tasas de transmisión y/o los tipos de señales no pertenecen a la topología de la red, aunque pueden verse afectados por la misma¹⁰. Existen varias topologías de red (ver Figura 2).

¹⁰ Arquitectura o Topología de Red http://es.wikipedia.org/wiki/Topolog%C3%ADa_de_red, 13 de marzo de 2007, 9:20 p.m.

Figura 2. Topologías de Red



2.3.9 Modelo Cliente | Servidor¹¹: Esta arquitectura presenta las siguientes características para implementar el prototipo que nos ocupa en este caso.

1. El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
2. Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.
3. Un servidor da servicio a múltiples clientes en forma concurrente.
4. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
5. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
6. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente

¹¹ Modelo Cliente | Servidor <http://es.wikipedia.org/wiki/Cliente-servidor>, 22 de Marzo de 2007, 11:10 a.m.

separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.

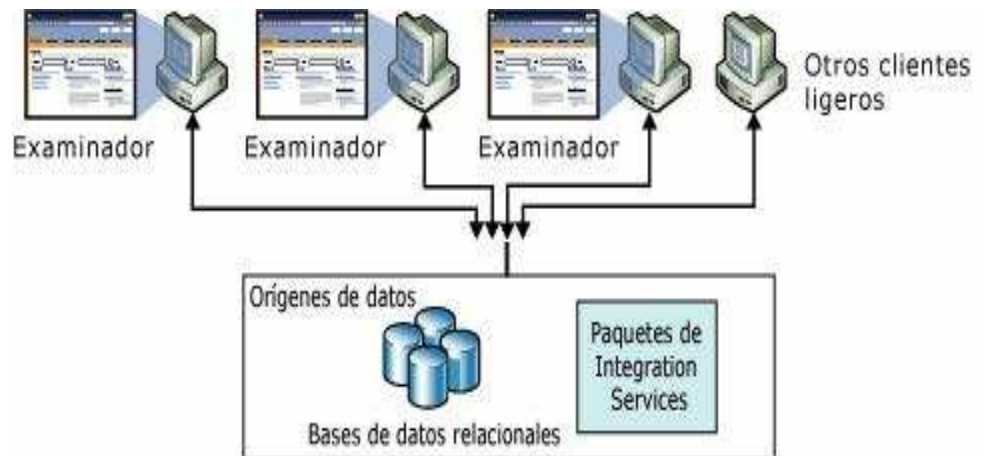
También es importante hacer notar que las funciones cliente/servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.

7. Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea.
8. Además se constituye como el nexo de unión mas adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.
9. Designa un modelo de construcción de sistemas informáticos de carácter distribuido.

La arquitectura cliente|servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. Por lo tanto, su implantación involucra diferentes tipos de estándares: APPC, TCP/IP, OSI, NFS, DRDA corriendo sobre DOS, OS/2, Windows o PC UNIX, en TokenRing, Ethernet, FDDI o medio coaxial, sólo por mencionar algunas de las posibilidades. (Ver figura 3)

Diagrama: Arquitectura Cliente | Servidor.

Figura 3. Modelo Cliente | Servidor



Combinación tecnológica exitosa

El modelo Cliente servidor y sus características frente a las bondades en las redes LAN, se ajusta al aplicativo ya que la infraestructura tecnológica de la universidad favorece mucho para hacer las pruebas en los laboratorios existentes.

2.3.10 UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de

programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante remarcar que UML es un "lenguaje" para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.¹²

2.3.10.1 Diagramas: En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente, como se muestra en la figura de la derecha.

2.3.10.1.1 Diagramas de estructura: Enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases
- Diagrama de componentes
- Diagrama de objetos
- Diagrama de estructura compuesta (UML 2.0)
- Diagrama de despliegue
- Diagrama de paquetes

¹² UML http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado, 07 de Abril de 2007, 12:45 a.m.

2.3.10.1.2 Diagramas de comportamiento: Enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

2.3.10.1.3 Diagramas de Interacción: Un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación
- Diagrama de tiempos (UML 2.0)
- Diagrama de vista de interacción (UML 2.0)

2.3.11 Aplicación Web: Una **aplicación web** es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero. La habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los webmails, wikis, weblogs, MMORPGs, tiendas en línea y la Wikipedia misma son ejemplos bien conocidos de aplicaciones web.

Historia

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, como parte de la aplicación, requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal.

En contraste, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

Interfaz

Las interfaces web tienen ciertas limitantes en la funcionalidad del cliente. Métodos comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar-y-soltar no están soportadas por las tecnologías web estándar. Los desarrolladores web comúnmente utilizan lenguajes interpretados del lado del cliente para añadir más funcionalidad, especialmente para crear una experiencia interactiva que no requiera recargar la página cada vez (cosa que suele molestar a los usuarios). Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías del lado del servidor, como por ejemplo PHP. JSP, AJAX, es una técnica de desarrollo web que usa una combinación de varias tecnologías.

Consideraciones técnicas

Una ventaja significativa en la construcción de aplicaciones web que soporten las características de los browsers estándar es que deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y es mostrada casi en todos lados. Sin embargo, aplicaciones inconsistentes de HTML, CSS, DOM y otras especificaciones de browsers pueden causar problemas en el desarrollo y soporte de aplicaciones web. Adicionalmente, la habilidad de los usuarios a personalizar muchas de las características de la interfaz (como tamaño y color de fuentes, tipos de fuentes, inhabilitar Javascript) puede interferir con la consistencia de la aplicación web.

Otra (poco común) aproximación es utilizar Macromedia Flash o Java applets para producir parte o toda la interfaz de usuario. Como casi todos los browsers incluyen soporte para estas tecnologías (usualmente por medio de plug-ins), aplicaciones basadas en Flash o Java pueden ser implementadas con aproximadamente la misma facilidad. Como hacen caso omiso de las configuraciones de los browsers estas tecnologías permiten más control sobre la interfaz, aunque incompatibilidad entre implementaciones de Flash o Java puedan traer nuevas complicaciones. Por las similitudes con una arquitectura cliente-servidor, con un cliente un poco “especializado”, hay disputas sobre si llamar a estos sistemas “aplicaciones web”; un término alternativo es “aplicación enriquecida de Internet”.

Estructura

Aunque muchas variaciones son posibles, una aplicación web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web es la primera capa, un motor usando alguna tecnología web

dinámica (ejemplo: CGI, PHP, Java Servlets o ASP) es la capa de en medio, y una base de datos como última capa. El navegador web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

Uso en negocios

Una estrategia que está emergiendo para las empresas proveedoras de software, es proveer acceso vía web al software. Para aplicaciones previamente distribuidas como de escritorio, esto puede requerir el desarrollo de una aplicación totalmente nueva o simplemente adaptar la aplicación para usar una interfaz web. Estos programas permiten al usuario pagar una cuota mensual o anual para usar la aplicación, sin necesidad de instalarla en la computadora del usuario. Las compañías que siguen esta estrategia son llamadas Proveedores de Aplicaciones de Servicio (ASP por sus siglas en inglés), este modelo de negocios está atrayendo la atención de la industria del software.

Lenguajes de programación

Existen numerosos lenguajes de programación utilizados para el desarrollo de Aplicaciones Web, entre los que destacan:

- ⇒ PHP
- ⇒ ASP/ASP.NET
- ⇒ Java, con sus tecnologías Java Servlets y JavaServer Pages(JSP)
- ⇒ Perl
- ⇒ Ruby
- ⇒ Python

Aunque ciertamente ASP no es un lenguaje de programación, sino una arquitectura de desarrollo web en la que se pueden usar por debajo distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET, o VBScript/JScript para ASP).¹³

2.3.12 Servidor Web: Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que suele conocerse como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

Sobre el servicio web clásico se puede disponer de aplicaciones web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP.

Es necesario distinguir entre las aplicaciones del cliente y las del servidor.

Aplicaciones en el lado del cliente, donde el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript, el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en

¹³ Aplicación Web http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web, 22 de Abril de 2007, 2:00 p.m.

lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.¹⁴

¹⁴ Servidor Web http://es.wikipedia.org/wiki/Servidor_web, 01 de Mayo de 2007, 5:40 p.m.

3. METODOLOGÍA

3.1 ENFOQUE DE LA INVESTIGACIÓN

El enfoque a trabajar es Empírico – analítico ya que debe pasar por la experiencia del levantamiento de información más el análisis de ésta para poder entender las fallas y fortalezas que se estén presentando y proceder al desarrollo de la solución.

3.2 LÍNEA DE INVESTIGACIÓN

- **Línea:** Tecnologías Actuales y Sociedad.
- **Sub-línea:** Sistemas de Información y Comunicaciones.
- **Campo:** Gestión Informática.

3.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

Para hacer el levantamiento de información se usarán las siguientes técnicas con el fin de obtener los datos necesarios e identificar las fallas y fortalezas que se están presentando actualmente.

- **Reuniones:** estas se realizarán directamente con el asesor de la investigación y el grupo **Pedagogía y Tecnología**, para tener el correcto dominio sobre la información respecto al tema del cuerpo rígido en equilibrio, cambios en la navegación de los temas, presentación de las inferencias y presentación de los informes.

3.4 POBLACIÓN Y MUESTRA

La población que empleará el prototipo de software educativo inicialmente serán los estudiantes que cursen durante el semestre II de 2007 las asignaturas: mecánica vectorial de tercer semestre del programa de Ingeniería Mecatrónica y estática de segundo semestre del programa de Ingeniería Aeronáutica de la Universidad de San Buenaventura.

3.5 HIPÓTESIS

Las habilidades de razonamiento deductivo desarrolladas por los estudiantes que utilizarán el prototipo de software educativo en sus diferentes niveles de complejidad, serán más altas frente a los estudiantes que no utilizarán el prototipo.

3.6 VARIABLES

3.6.1 Variables Independientes

El ambiente computacional empleado en las sesiones de trabajo, puede tomar los siguientes valores ({AA}únicamente con inferencias de tercer nivel [3]; {BB} con todos los niveles de inferencias en este orden [1,2,3]; {CA} con todos los niveles de inferencias en este orden [2,1,3]; {CB} con todos los niveles de inferencias en este orden [2,3,1]; {CC} con todos los niveles de inferencias en este orden [3,2,1]; {CD} con todos los niveles de inferencias en este orden [3,1,2])

3.6.2 Variables Dependientes

Las habilidades de razonamiento deductivo desarrolladas por los estudiantes reflejadas mediante el desempeño académico en el post-test

4. DESARROLLO INGENIERIL

4.1. PROTOCOLO DE TRABAJO

CICLO DE VIDA DE LOS SISTEMAS DE INFORMACIÓN.

1. **Primera Fase:** Levantamiento de información. (Planificación ¿Qué hacer?)

⇒ **Objetivos.**

1. Estudio de viabilidad.
2. Conocer los requisitos que debe satisfacer el sistema (funciones y limitaciones de contexto).
3. Asegurar que los requisitos son alcanzables.
4. Formalizar el acuerdo con los usuarios.
5. Realizar una planificación detallada.

2. **Segunda Fase:** Análisis.

⇒ **Objetivos.**

1. Identificar actividades.
2. Entender cómo se realizan.
3. Volumen de transacciones o decisiones.
4. Identificar los problemas.
5. Identificar los alcances de los problemas.
6. Identificar las causas de los problemas.

3. **Tercera Fase:** Diseño (¿Cómo hacerlo? Soluciones en costo, tiempo y calidad).

➔ **Objetivos.**

1. Identificar soluciones tecnológicas para cada una de las funciones del sistema.
2. Asignar recursos materiales para cada una de las funciones.
3. Establecer métodos de validación del diseño.
4. Ajustar las especificaciones del producto.

4. **Cuarta Fase:** Construcción.

➔ **Objetivos.**

1. Generar el producto o servicio pretendido con el proyecto.
2. Integrar los elementos subcontratados o adquiridos externamente.
3. Validar que el producto obtenido satisface los requisitos de diseño previamente definidos y realizar, si es necesario, los ajustes necesarios en dicho diseño para corregir posibles lagunas, errores o inconsistencias.

4.2 ANÁLISIS

4.2.1 Modelado UML

Actores

Estudiante y/o Usuario: Es quien interactúa con todos los casos de uso y el beneficiario del prototipo del software educativo. (Ver Tabla 1)

Tabla1. Actor Estudiante

Actor	Estudiante
Caso de Uso	Registrar Estudiante, validar Estudiante, Consultar contenido, sesión de inferencias
Tipo	Primario
Descripción	Es el actor principal, es cualquier estudiante y/o usuario que desee utilizar el prototipo de software educativo

Base de datos local: Interactúa con el caso de uso consultar información; contiene toda la información para desplegar todo el contenido sobre los cuerpos rígidos en equilibrio y todo el banco de preguntas para la sección de evaluación. (Ver tabla 2)

Tabla 2. Actor Base de datos local

Actor	Base de datos local
Caso de Uso	Consultar información, informes
Tipo	Secundario
Descripción	Representa la base de datos local; donde se encuentra alojada toda la información sobre el tema de cuerpos rígidos en equilibrio, el banco de inferencias e informes

Profesor: Es quien interactúa con todos los casos de uso y el beneficiario del prototipo del software educativo. (Ver tabla 3)

Tabla 3. Actor Profesor

Actor	Profesor
Caso de Uso	Registrar Profesor, validar profesor, Consultar contenido, consultar informes, consultar sección de preguntas
Tipo	Primario
Descripción	Es el actor que se encarga de consultar los informes que genera el prototipo con base a los resultados que obtienen los estudiantes, también puede consultar la información para verificar su veracidad y la sesión de inferencias para verificar la dificultad de estas.

Administrador: Es quien se encarga de de la administración y mantenimiento del prototipo. (Ver tabla 4)

Tabla 4. Actor Administrador

Actor	Administrador
Caso de Uso	Registrar administrador, validar administrador, consultar información, ingresar información, actualizar información
Tipo	Primario
Descripción	Es el actor que se encarga de actualizar el sistema (contenido, consultas, informes, usuarios)

Base de Datos Actualizada: Es quien almacena toda la información actualizada del prototipo (contenidos, consultas, informes, usuarios), es la base de datos que el administrador actualiza. (Ver tabla 5)

Tabla 5. Actor Base de datos actualizada

Actor	Base de datos actualizada
Caso de Uso	Consultar información, informes
Tipo	Secundario
Descripción	Representa la base de datos actualizada; donde se encuentra alojada toda la información sobre el tema de cuerpos rígidos en equilibrio, el banco de inferencias e informes

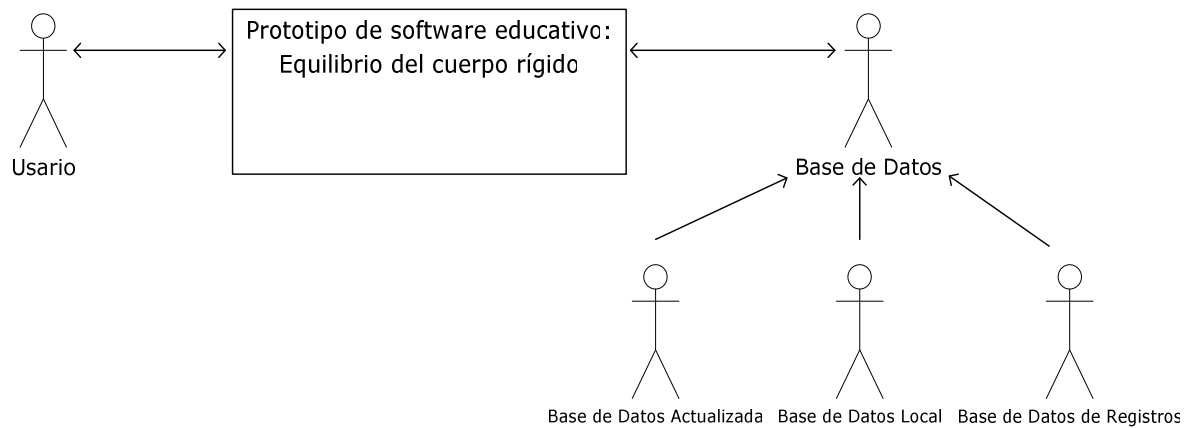
Bases de Datos de Registros: Registra y valida al administrador para permitir el ingreso y la actualización del prototipo; al profesor para permitir el ingreso al modulo de informes y al estudiante para entrar al modulo de consulta y la sesión de preguntas. (Ver tabla 6)

Tabla 6. Actor Base de datos de registros

Actor	Base de datos de registros
Caso de Uso	Registrar, validar (administrador, profesor, estudiante)
Tipo	Secundario
Descripción	Representa la base de datos de registros; donde se guarda la información relacionada con el administrador, profesor y estudiante.

Ilustración Diagrama de Actores (ver Figura 4)

Figura 4. Actores



Formato Caso de uso: Registrar Usuario (Ver Tabla 7)

Tabla 7. Caso de uso: Registrar Usuario

Caso de Uso	Registrar usuario
Actores	Usuario, Base de datos de Registros
Tipo	Inclusión
Propósito	Registrar al usuario, para ingresar al prototipo y consultar el contenido sobre estática e iniciar la sesión de inferencias
Resumen	El usuario se registra mediante su código USB y un

	<p>grupo previamente asignado por el profesor, de esta forma accede a toda la información y a una determinada sesión de inferencias que se determina por el grupo asignado por el profesor en el momento que el usuario va iniciar su sesión</p>
Precondiciones	<p>El usuario debe tener un código USB y un grupo que le debe proveer el profesor.</p>
Flujo Principal	<p>Se presenta la pantalla principal (Log-in) al usuario, donde se debe ingresar el código USB y escoger un grupo el cuál le indica el docente.</p> <p>Los opciones de grupos son:</p> <p>Grupo AA Grupo BB Grupo CA Grupo CB Grupo CC Grupo CD</p> <p>Se registra al usuario con su código USB y el grupo que se le asigna el usuario. Después de leer el contenido (equilibrio del cuerpo rígido) llega a una sesión de inferencias determinada por el grupo que se le asigno</p>

	al momento del Log-in
Sub-flujos	Ninguno
Excepciones	Si el usuario no posee un código USB y/o la contraseña, debe realizar el proceso de pedir su código USB en registro académico o que el profesor le asigne códigos provisionales para utilizar el prototipo

Formato Caso de uso: Validar Profesor (Ver Tabla 8)

Tabla 8. Caso de uso: Validar Profesor

Caso de Uso	Validar Profesor
Actores	Profesor, base de datos de Registros
Tipo	Inclusión
Propósito	Validar al profesor ya registrado o creado en la base de datos, para ingresar al prototipo.
Resumen	El profesor inicia con un Log-in donde ingresa su nombre y cedula, estos se validan con la información de la base de datos para conceder el acceso al modulo de informes de resultados

Precondiciones	Se requiere que anteriormente el administrador del prototipo, cree al profesor en la base de datos, es decir que el administrador realice el caso de uso crear profesor
Flujo Principal	<p>Se presenta la pantalla principal (Log-in) al profesor, donde se debe ingresar el nombre y la cedula.</p> <p>La opción es “entrar”.</p> <p>Se valida al profesor con su nombre y la cedula. Después de la validación puede acceder al menú de informes de resultados</p>
Sub-flujos	Ninguno
Excepciones	Si la validación no es exitosa, el administrador debe verificar los datos del profesor y actualizarlos o sino existe el administrador debe crearlo en la base de datos

Formato Caso de uso: Consultar información (Ver Tabla 9)

Tabla 9. Caso de uso: Consultar Información

Caso de Uso	Consultar Información
Actores	Usuario, Profesor, base de datos de local
Tipo	Inclusión
Propósito	Permitir al usuario consultar toda la información sobre estática y el equilibrio del cuerpo rígido
Resumen	El usuario consulta todo los temas relacionados con la estática y el equilibrio del cuerpo rígido.
Precondiciones	Se requiere que anteriormente que el usuario haya ingresado al prototipo registrándose al mismo
Flujo Principal	Se presenta una pantalla principal don encuentra un mapa de navegación. Este esta compuesto por botones que los llevan a el contenido especifico que desea consultar
Sub-flujos	Consultar preguntas
Excepciones	Ninguna

Formato Caso de uso: Consultar Informes (Ver Tabla 10)

Tabla 10. Caso de uso: Consultar Informes

Caso de Uso	Consultar Informes
Actores	Profesor, base de datos de Actualizada
Tipo	Inclusión
Propósito	Permitir al profesor consultar los informes que crea el prototipo a partir de las sesiones de trabajo de los usuarios
Resumen	El profesor consulta los diferentes informes que genera el prototipo, los informes contienen los resultados de las sesiones de trabajos que realizaron los usuarios
Precondiciones	Se requiere que anteriormente que el profesor haya ingresado al prototipo con su respectiva validación
Flujo Principal	Se presenta al profesor la pantalla principal donde encuentra un menú para seleccionar el informe que desea consultar. Las opciones son: Grupo AA

	<p>Grupo BB</p> <p>Grupo CA</p> <p>Grupo CB</p> <p>Grupo CC</p> <p>Grupo CD</p> <p>Al seleccionar alguna de las opciones se presenta una pantalla donde encontrara todos los resultados y la grafica global del resultado del grupo</p>
Sub-flujos	Consultar los detalles de la sesión de cada usuario
Excepciones	Solo se puede generar informes si ya sea realizado al menos una sesión de trabajo

Formato Caso de uso: Consultar Inferencias (Ver Tabla 11)

Tabla 11. Caso de uso: Consultar Inferencias

Caso de Uso	Consultar Inferencias
Actores	Usuario, Profesor, base de datos de local
Tipo	Inclusión
Propósito	Permitir al usuario consultar el banco de inferencias que tiene el prototipo sobre estática y equilibrio del cuerpo rígido

Resumen	El usuario después de consultar el contenido sobre estática y equilibrio del cuerpo rígido pasa a la sesión de inferencias las cuales están definidas por el grupo que selecciona al inicio de la sesión de trabajo
Precondiciones	Se requiere que el usuario haya ingresado al prototipo, sin embargo, no es obligatorio consultar la información puede empezar la sesión de preguntas sin haber leído nada.
Flujo Principal	<p>Se presenta una pantalla principal en donde va a encontrar la inferencia la cual consta de dos frases más una conclusión.</p> <p>Las opciones son:</p> <p>Verdadero</p> <p>Falso</p> <p>Depuse de escoger una opción automáticamente pasa la siguiente inferencia. La sesión de inferencias es de 4</p>
Sub-flujos	Ninguno
Excepciones	El usuario debe terminar la sesión para poder salir del prototipo

Ilustración Caso de uso (Consultar información) (ver Figura 5)

Figura 5. Caso de uso (Consultar información)

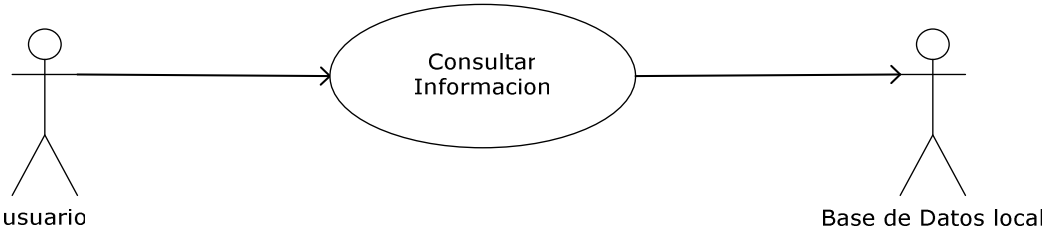


Ilustración Caso de uso (Consultar inferencias) (ver Figura 6)

Figura 6. Caso de uso (consultar inferencias)

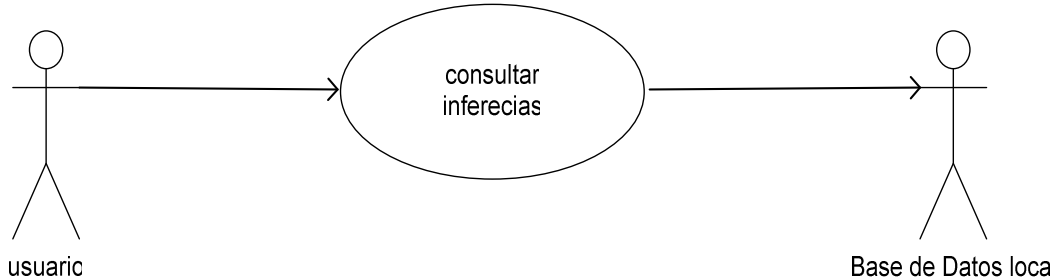
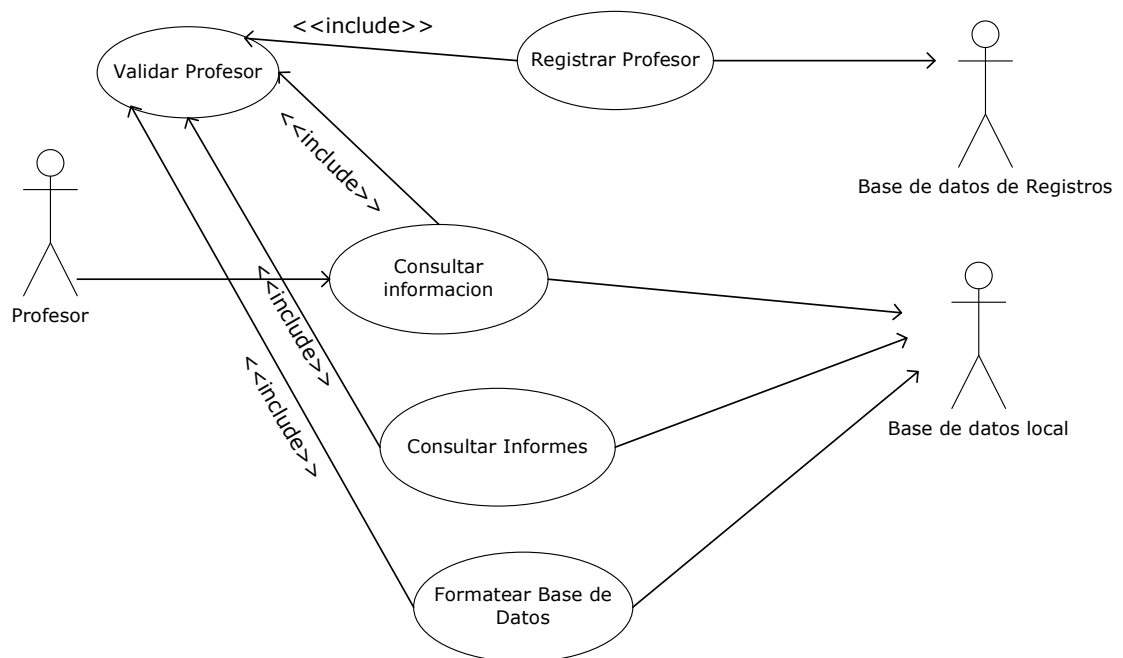


Ilustración Caso de uso (profesor) (ver Figura 7)

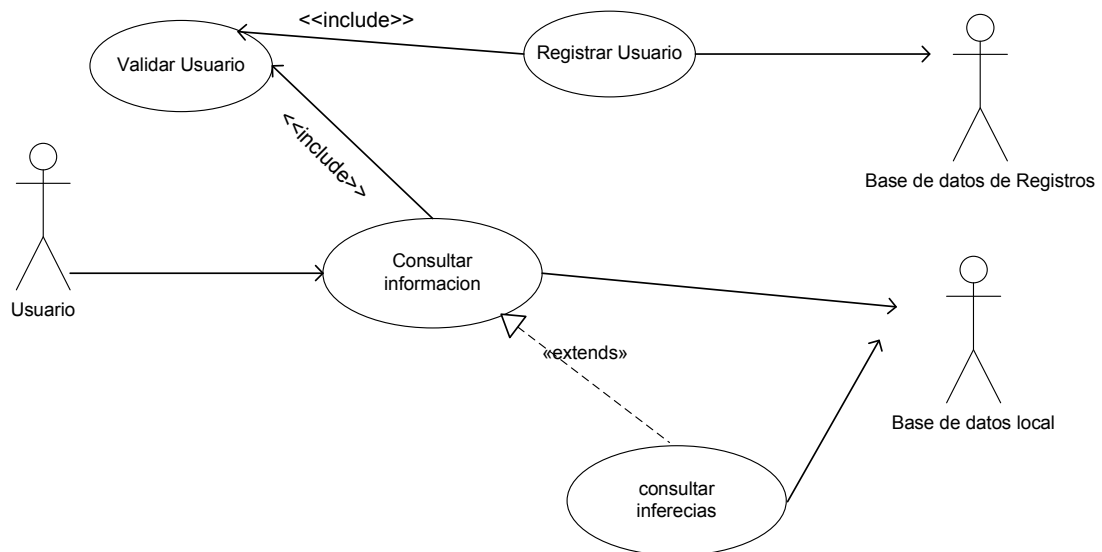
Figura 7. Caso de uso (Profesor)



Resumen: Para ingresar al prototipo el profesor primero debe ser validado ingresando su nombre y cedula. Después de realizar el ingreso esta en libertad de consultar los informes o consultar el contenido del tema el cuerpo rígido en equilibrio y las inferencias que hay en los diferentes grupos

Ilustración Caso de uso (usuario) (ver Figura 8)

Figura 8. Caso de Uso (usuario)



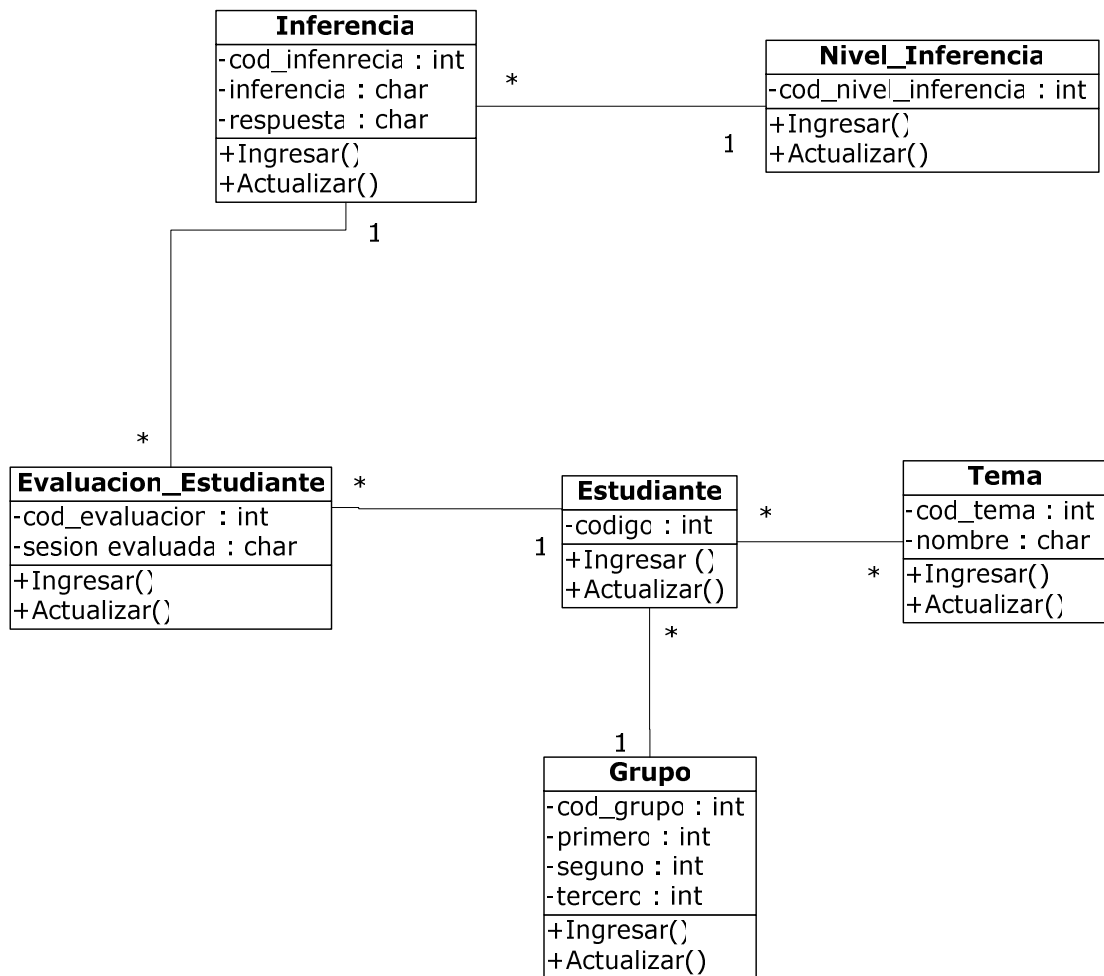
Resumen: el usuario debe ingresar su código USB y seleccionar un grupo de trabajo para la sesión de inferencias. Después del ingreso el usuario puede empezar a consultar toda la información del tema cuerpo rígido en equilibrio después de leer los temas de su elección puede pasar al sesión de trabajo de inferencias. Para pasar a la sesión de trabajo no es obligatorio q consulte alguna información lo puede hacer directamente.

4.3 DISEÑO

4.3.1 Diagrama de Clases

Ilustración Diagrama de clases (ver Figura 9)

Figura 9. Diagrama de clases



Asociaciones de Clases

Asociación de Clases
Un nivel de inferencia tiene muchas inferencias
Una inferencia esta en varias evaluaciones de estudiantes
Un estudiante tiene varias evaluaciones
Un estudiante puede ver muchos temas
Un tema puede ser visto por muchos estudiantes
Un grupo tiene muchos estudiantes

Diccionario de Clases

Nivel_Inferencia: Se identifica con un código. Un nivel de inferencia puede tener muchas inferencias

Inferencia: Se identifica con un código, una inferencia, una respuesta a la inferencia. Una inferencia pertenece a un nivel de inferencia.

Evaluación_Estudiante: Se identifica con un código y una sesión evaluada. Una evaluación de estudiante tiene muchas inferencias

Estudiante: Se identifica con un código. Un estudiante puede ver muchos temas

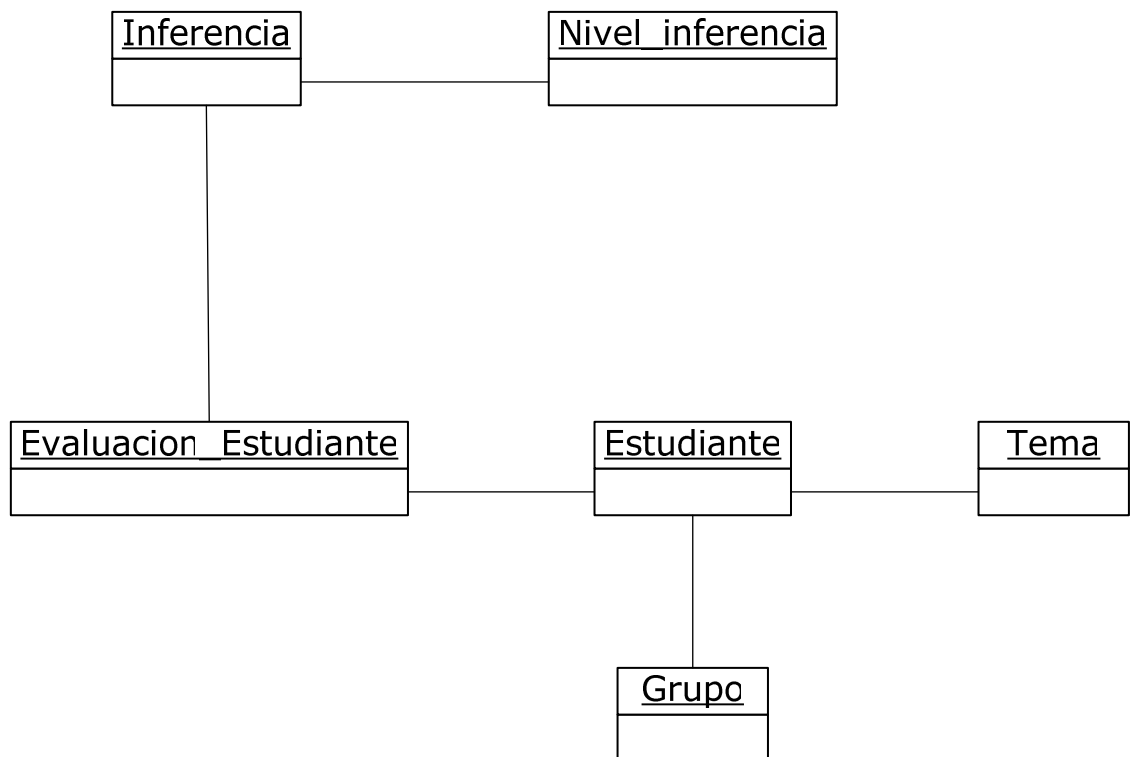
Tema: Se identifica con código y un nombre. Un tema puede tener muchos estudiantes

Grupo: Se identifica con un código, primero, segundo y tercero. Un grupo puede tener varios estudiantes.

4.3.2 Diagrama de Objetos

Ilustración Diagrama de objetos (ver Figura 10)

Figura 10. Diagrama de Objetos

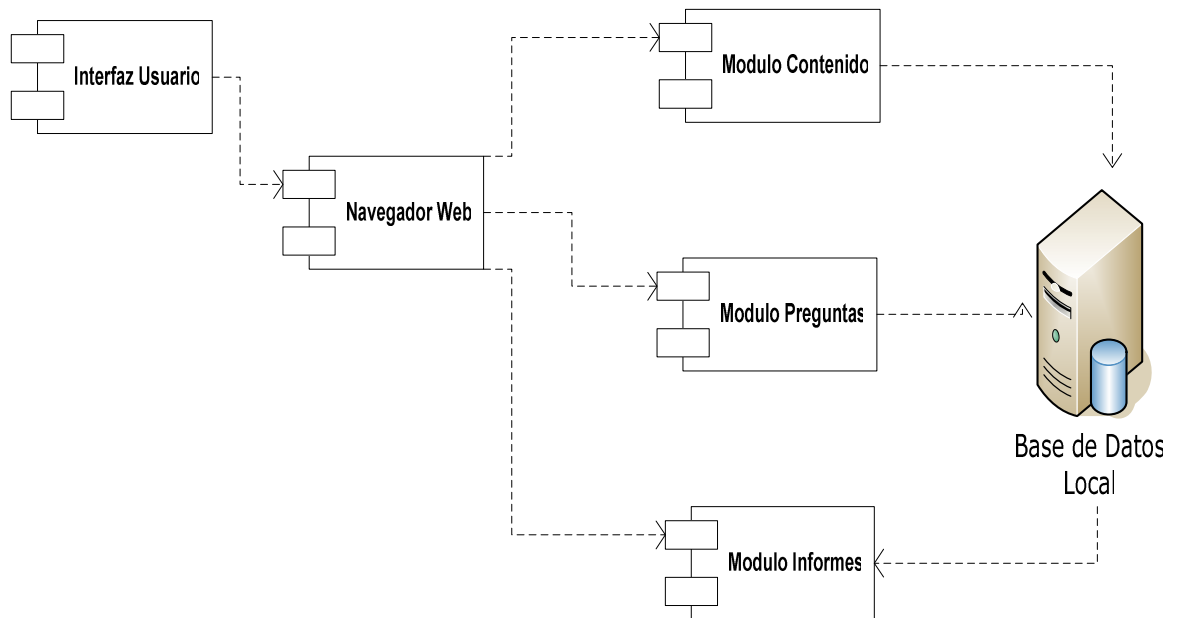


4.3.3 Diagrama de Componentes

Aplicación

Este diagrama muestra como se comporta la aplicación unas veces dado inicio a una sesión, el usuario debe ingresar su código y escoger el grupo que se visualiza en la interfaz. Los módulos se en cargan del funcionamiento de la aplicación recolectando y almacenando la información en la base de datos local. Posteriormente el profesor inicia una sesión ingresando su nombre y cedula que se visualiza en la interfaz para trabajar el modulo de informes.(Ver Figura11)

Figura 11. Diagrama de Componentes (Aplicación)

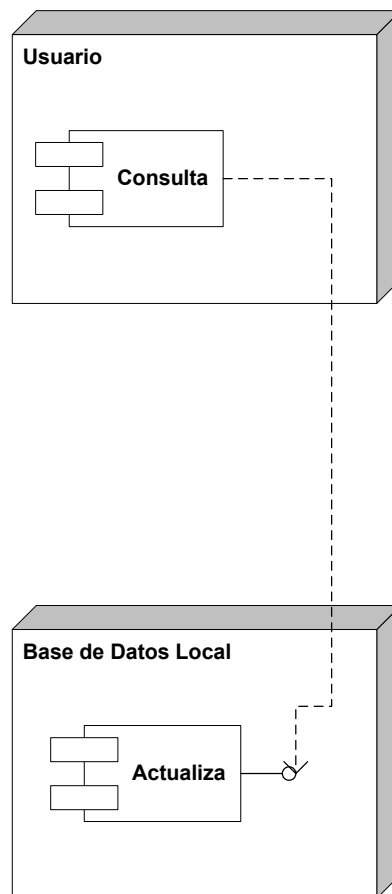


4.3.4 Diagrama de Despliegue

Aplicación

El usuario consulta la información libremente y la aplicación se encarga de de recolectar y almacenar los datos en la Base de Datos local, de esta forma se pueden generar los informes de consulta para el docente. (Ver Figura 12)

Figura 12. Diagrama de Despliegue (Aplicación)

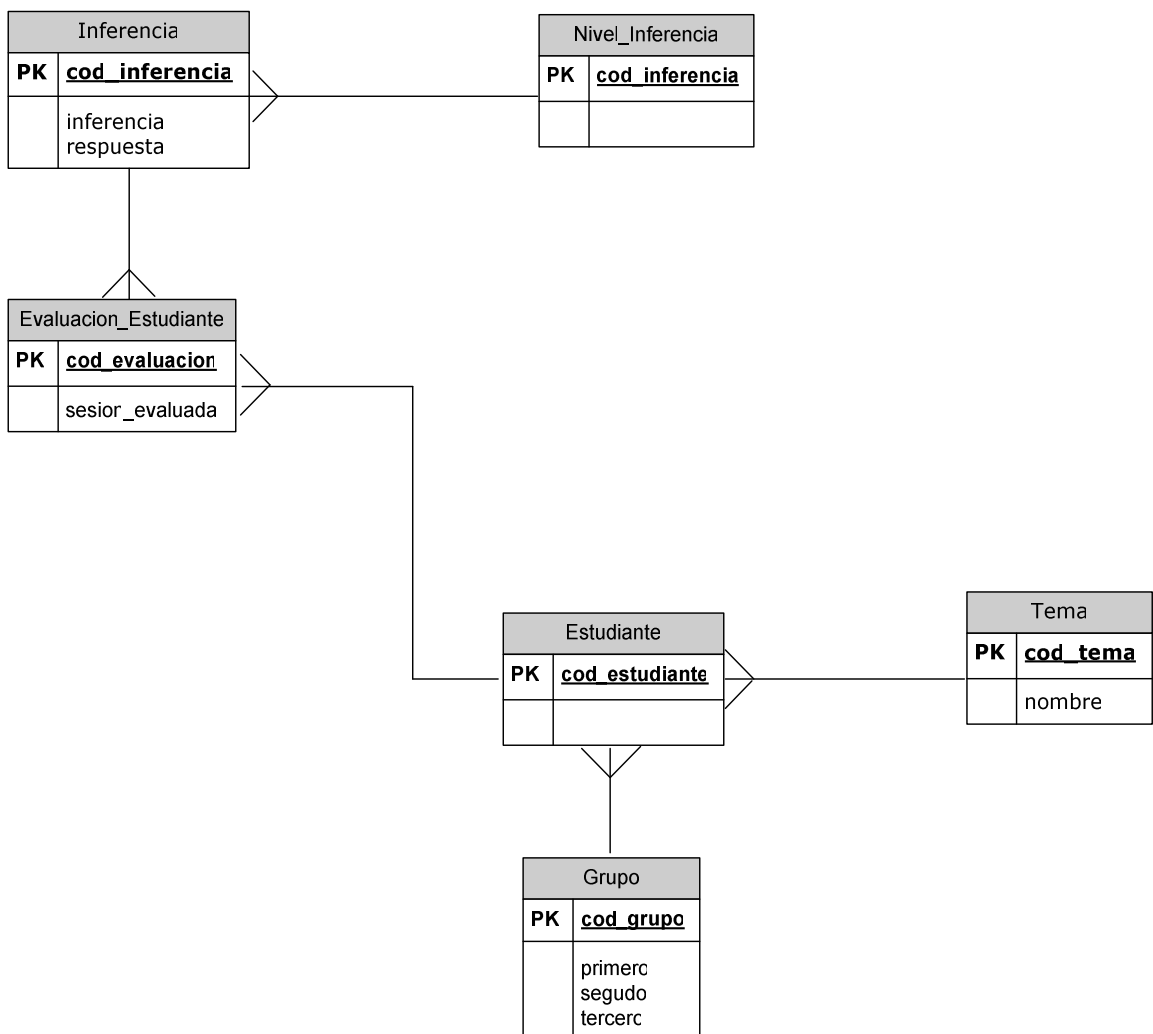


4.3.5 Base de Datos

4.3.5.1 Modelo Conceptual de la Base de Datos

Ilustración Modelo Conceptual (ver Figura 13)

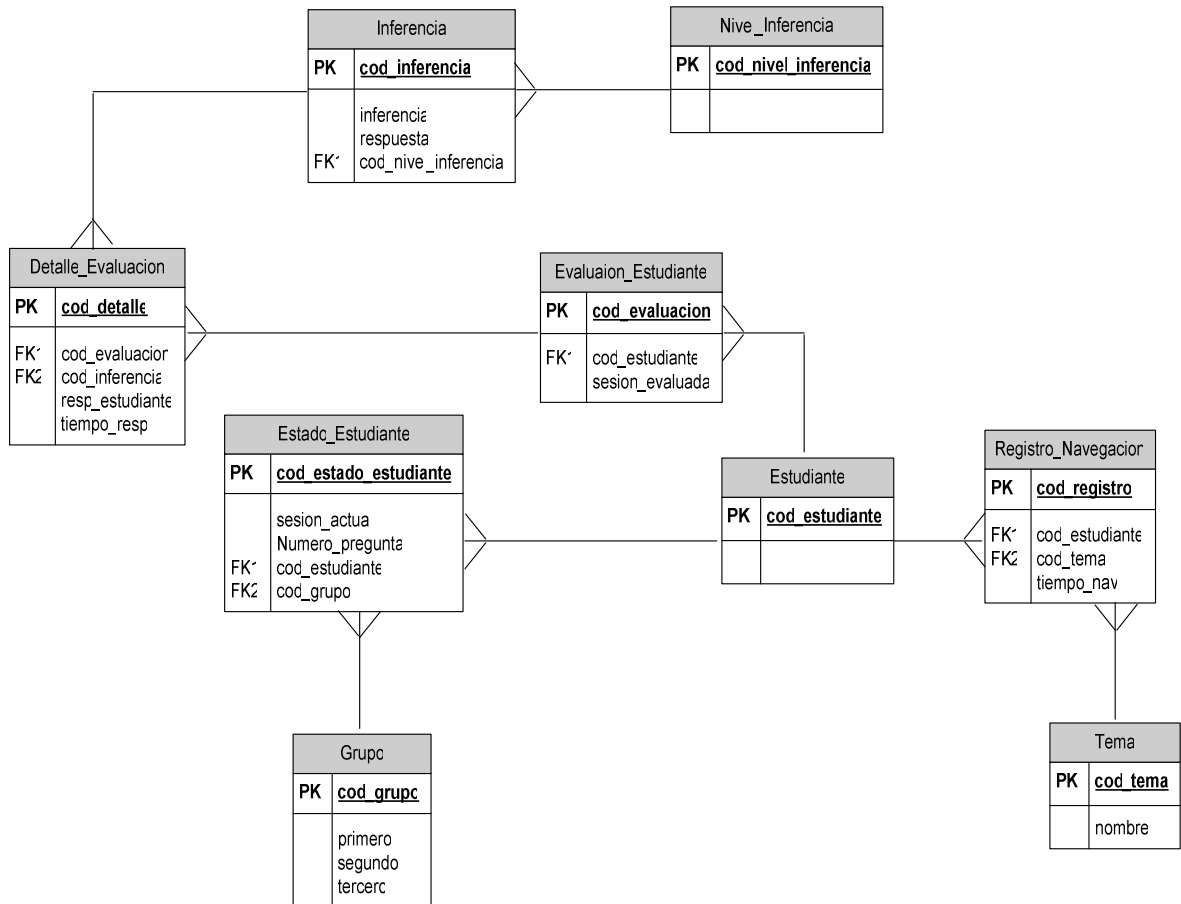
Figura 13. Modelo Conceptual



4.3.5.2 Modelo Lógico

Ilustración Modelo Lógico (ver Figura 14)

Figura 14. Modelo Lógico



5. REQUERIMIENTOS DEL SISTEMA

5.1. Requerimientos Mínimos Servidor

Hardware

- **Pocesador:** Pentium 4 1.66Ghz
- **Ram:** 128MB
- **Disco Duro:** 80GB

Software

- Macromedia 8
- MySql
- Tomcat
- j2re 1.4.1

5.2. Requerimientos Mínimos Clientes

Hardware

- **Pocesador:** Pentium III 1.66Ghz
- **Ram:** 64MB
- **Disco Duro:** 60GB

Software

- j2re 1.4.1
- Navegador Web

5.3 Herramientas de Desarrollo

- Macromedia 8
- MySql 3.23
- Tomcat 4.x
- j2re 1.4.1
- Microsoft Visio
- SmartDraw suite 7

6. CONCLUSIONES

En el transcurso del desarrollo de esta investigación se encontraron varios aspectos importantes a mencionar, y que resultan de la experiencia al analizar los impactos positivos del uso de la tecnología para el desarrollo del conocimiento y la facilidad para realizar análisis de resultados.

- Este prototipo de software educativo, proporciona otro punto de partida para seguir, investigando y desarrollando, nuevas formas para el desarrollo del conocimiento apoyadas en la tecnología.
- Este prototipo, ayuda a los estudiantes a desarrollar su nivel de análisis, partiendo de la condición de que no se debe trabajar únicamente la memorización de conceptos aislados, pues el prototipo lo guía a analizar las inferencias partiendo de documentación previamente presentada durante la totalidad de las sesiones de trabajo, es decir, que ya no debe memorizar conceptos, sino leer, analizar y comprender, de esta forma pueda resolver las inferencias, pero analizando el contenido para llegar a una conclusión empleando el razonamiento deductivo
- El sistema provee transparencia al usuario que es indiferente a la lógica del mismo. Haciendo fácil la utilización del servicio.

- Definitivamente para el futuro de la educación esta en seguir apoyándose en desarrollos tecnológicos y de esta manera mejorar las metodologías de enseñanza

- Implementar el modelo cliente servidor proporciona claras ventajas para el desempeño del prototipo y de esta forma recolectar, almacenar y generar informes eficazmente para los análisis del profesor.

BIBLIOGRAFIA

Libros

BEDFORD, Anthony, FOWLER, Wallace. Estática Mecánica para ingeniería. Addison Wesley, Wilmington, Delaware, E.U.A. 1996. 606 Págs.

BEER, Ferdinand P., JOHNSTON, Russell E. Jr. Mecánica vectorial para ingenieros. Mc Graw Hill, Sexta edición. México. 1997. 599 Págs.

CONSTITUCIÓN POLITICA DE COLOMBIA Ley 23 de 1982. Artículo 2°.

GALVIS PANQUEVA, Alvaro. "Micromundos lúdicos interactivos: aspectos críticos en su diseño y desarrollo", en *Tecnología y Comunicación Educativas*, Año 12, N° 28, (1998). 450 Págs.

POZO MUNICIO, Juan Ignacio. Aprender y enseñar ciencia. Ediciones Morata. Madrid. 1998. 350 Págs.

PRESSMAN, Roger S. Ingeniería del software un enfoque practico. Mc Graw Hill, quinta edición. Madrid. 2002. 601 Págs.

Fuentes Secundarias

Aplicación Web, http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web, 22 de Abril de 2007, 2:00 p.m.

Arquitectura o Topología de Red,
http://es.wikipedia.org/wiki/Topolog%C3%ADa_de_red, 13 de marzo de 2007, 9:20 p.m.

Bases de Datos, http://es.wikipedia.org/wiki/Base_de_datos, 05 de Marzo de 2007, 7:30 a.m.

Fases de desarrollo de Software,
http://es.wikipedia.org/wiki/Fases_del_desarrollo_de_software, 05 de Mayo de 2007, 5:31 p.m.

Modelo Cliente | Servidor, <http://es.wikipedia.org/wiki/Cliente-servidor>, 22 de Marzo de 2007, 11:10 a.m.

RIBIECOL, http://ribiecol.org/congreso/menu_horizontal/ribie.jsp.htm, 15 de Febrero de 2007, 4:30 p.m.

Servidor Web, http://es.wikipedia.org/wiki/Servidor_web, 01 de Mayo de 2007, 5:40 p.m.

UML, http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado, 07 de Abril de 2007, 2:00 p.m.

ANEXO A

MANUAL DE INSTALACIÓN

1. Instalar el Gestor de Bases de Datos Mysql. Tener presente el nombre del usuario y la contraseña del administrador

Figura 1: Inicio de la instalación del gestor de la base de datos

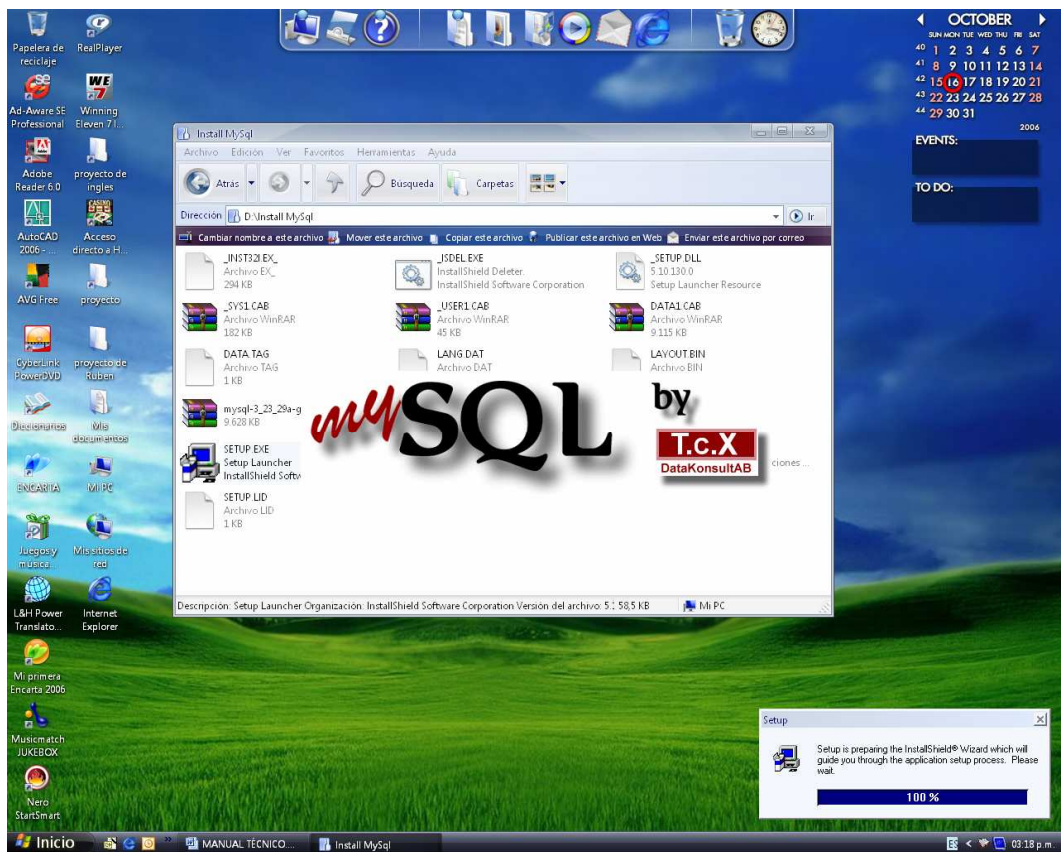


Figura 2: Mensaje de bienvenida para la instalación del gestor de la base datos

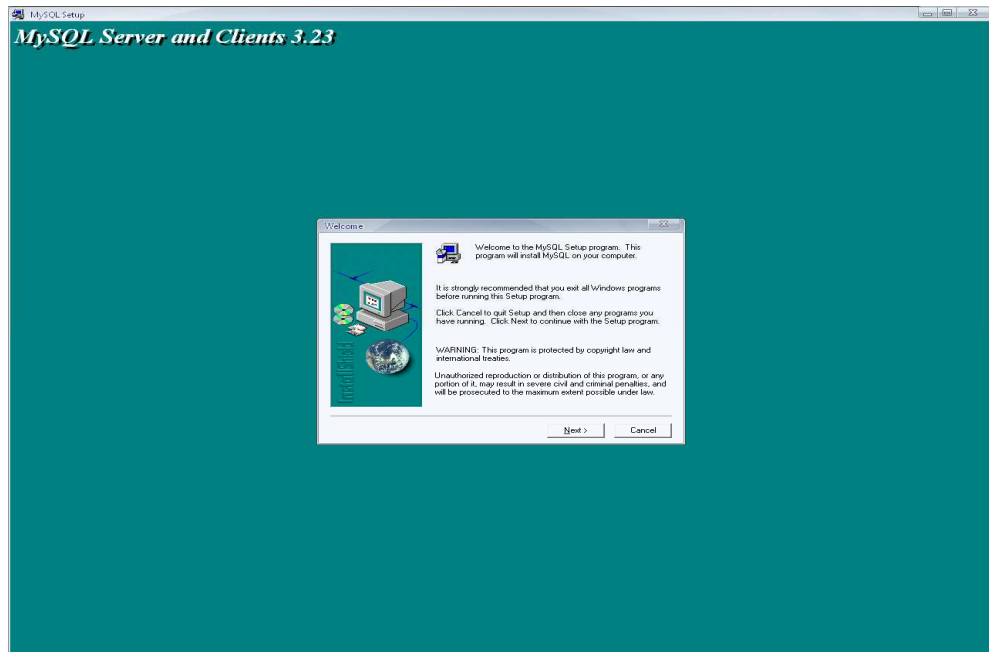


Figura 3: Mensaje inicial de las condiciones legales para instalar el gestor de la base de datos

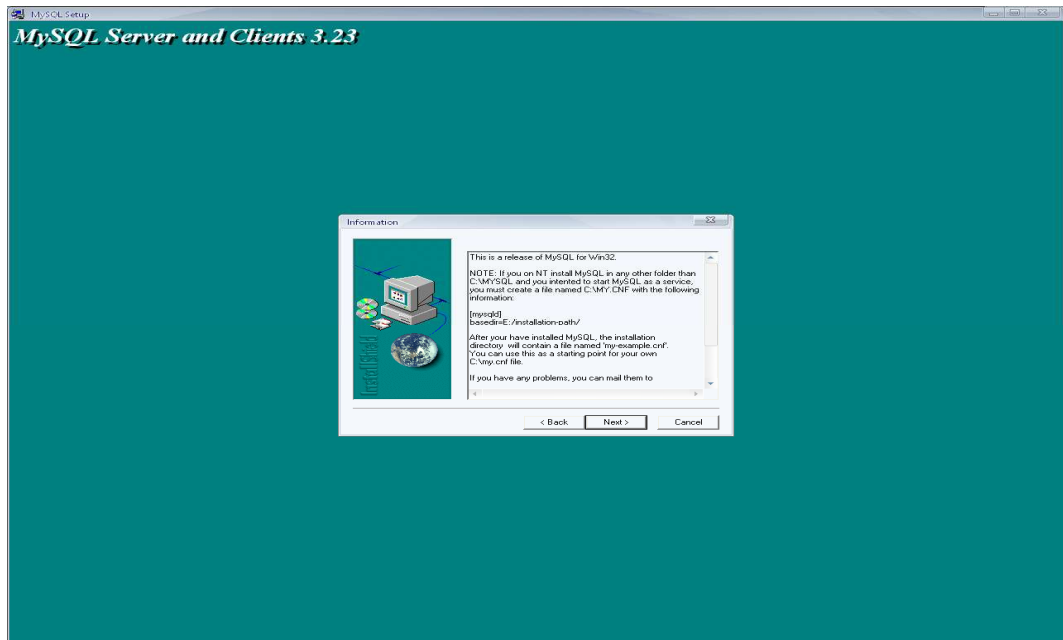


Figura 4: Mensaje para la ruta de instalación del gestor de datos

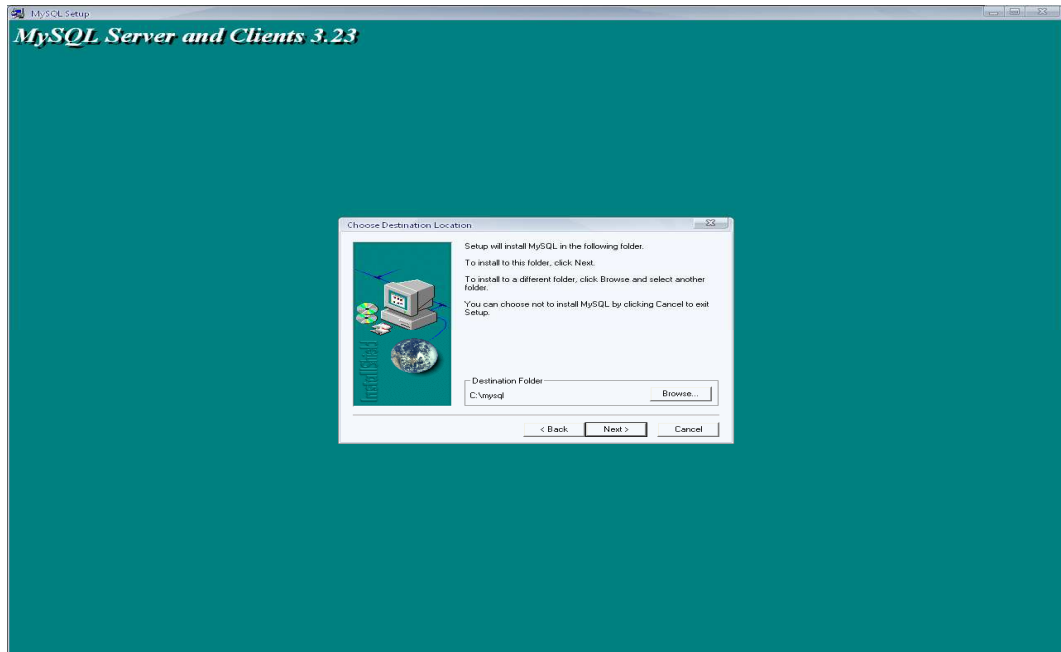


Figura 5: Mensaje para seleccionar el método de instalación del gestor de base de datos.

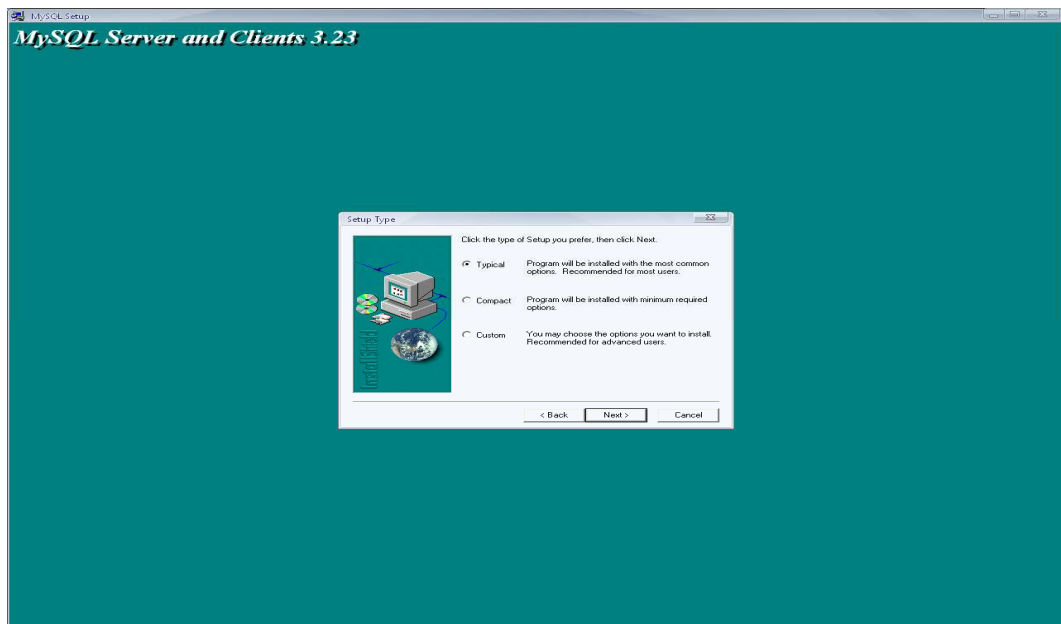
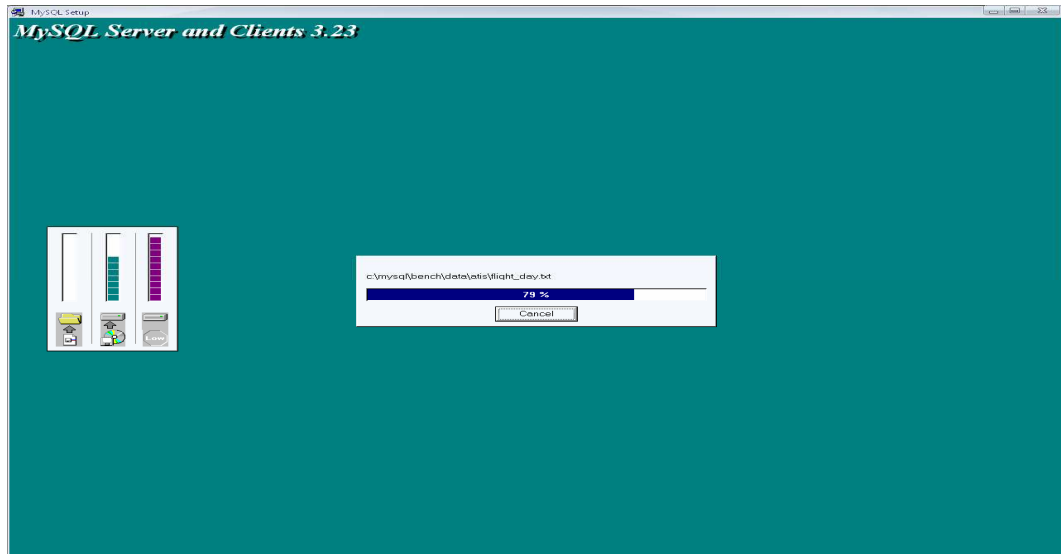


Figura 6: Proceso de instalar el gestor de base datos



2. Ingresar a C:\mysql para diligenciar nombre y contraseña del administrador inicializando la herramienta de administración WinMysqlAdmin.exe

Figura 7: Ingreso a C:\mysql para diligenciar nombre y contraseña del administrador iniciando la herramienta de administración winmysqladmin.exe

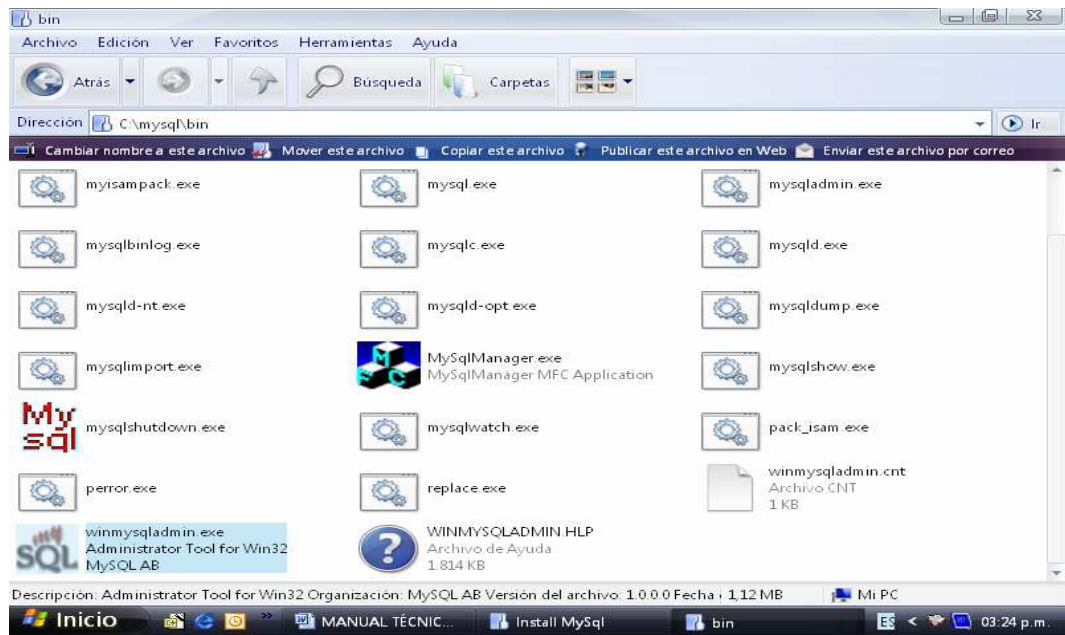


Figura 8: Diligenciar nombre y contraseña del administrador. Nombre: root, password: root



3. Almacenar la(s) BD “usb” C:\mysql\data\

4. Instalar la herramienta de desarrollo J2sdk 1.4.1 o una versión superior.

Figura 9: Instalar la herramienta de desarrollo java j2sdk 1.41 o superior.

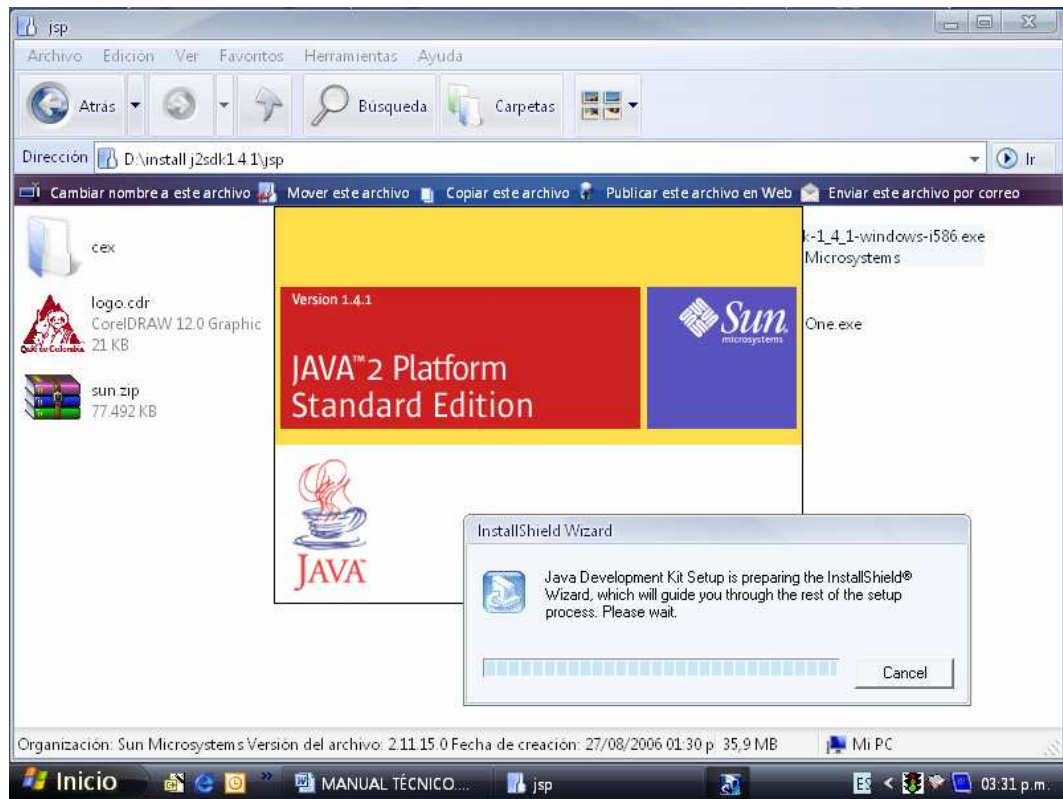


Figura 11: Mensaje de bienvenida para la instalación de java j2dk 1.4.1

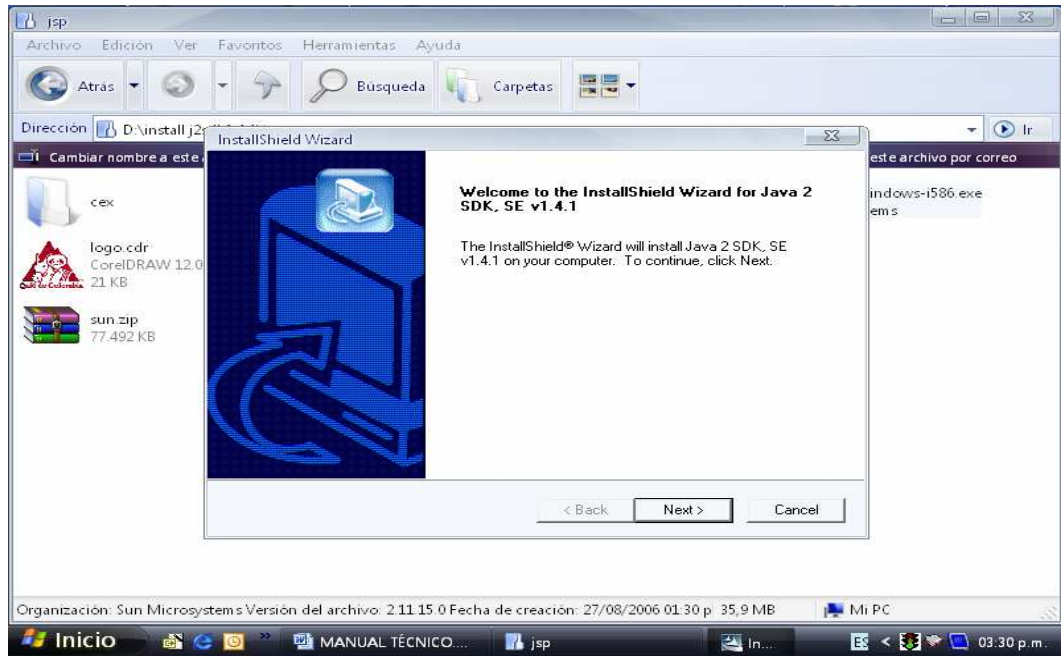


Figura 12: Mensaje inicial de las condiciones legales para instalar la herramienta java j2sdk 1.4.1

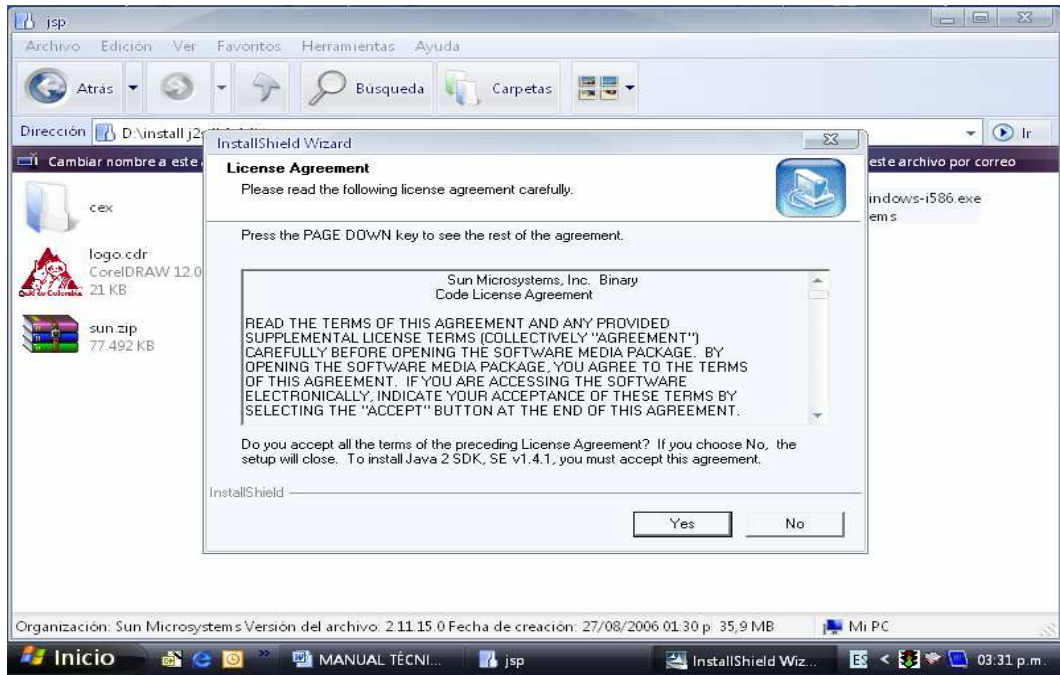


Figura 13: Mensaje para la ruta de instalación de java j2sdk 1.4.1

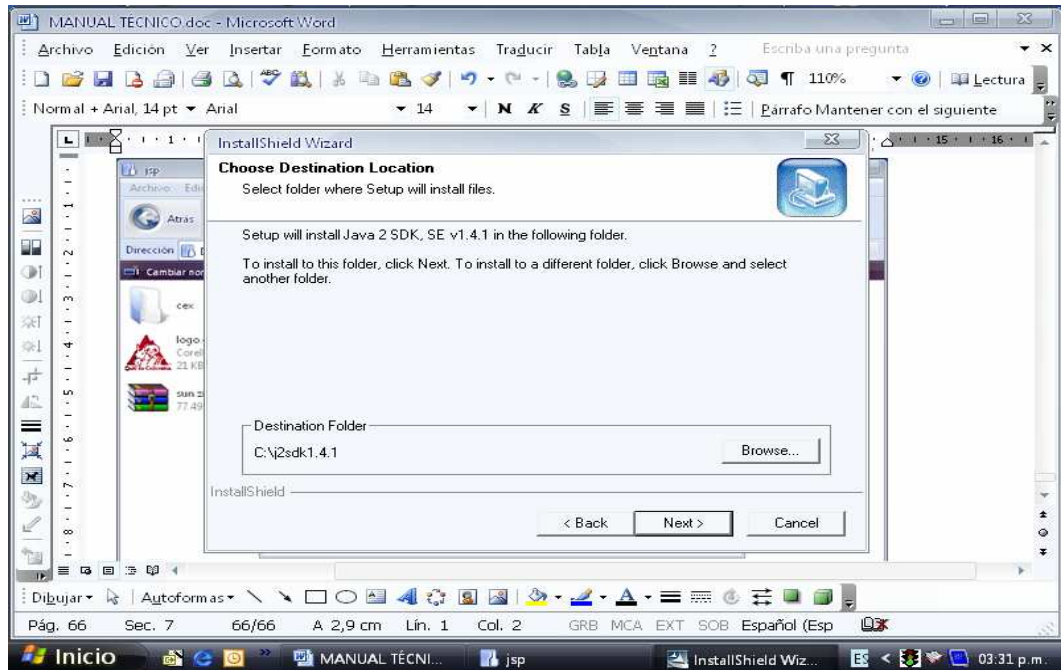


Figura 14: Mensaje para seleccionar el método de instalación de la herramienta j2sk 1.4.1

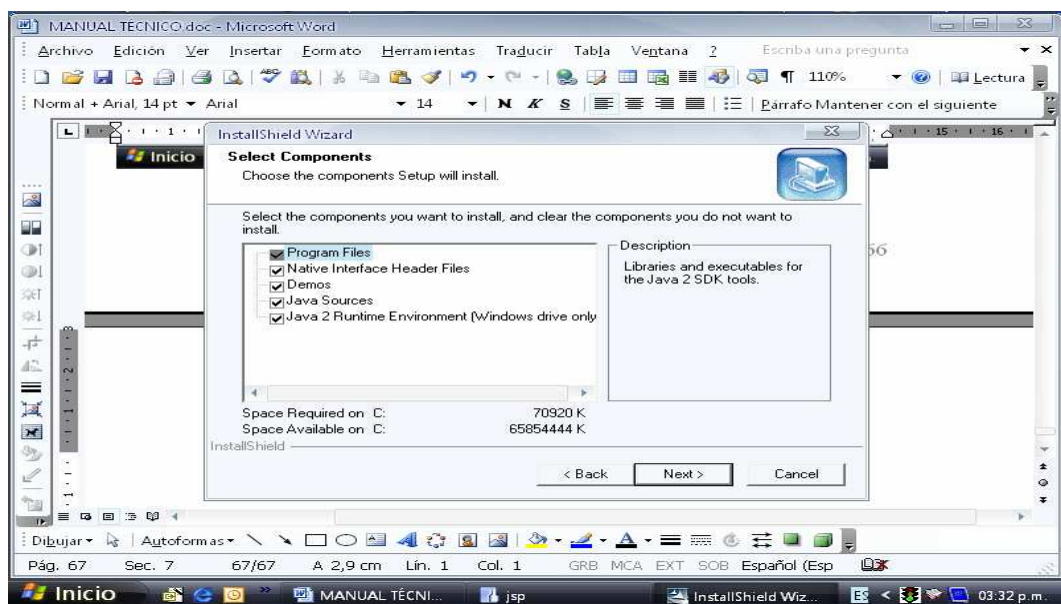


Figura 15: Mensaje para la selección del navegador Web de la instalación de la herramienta j2sdk

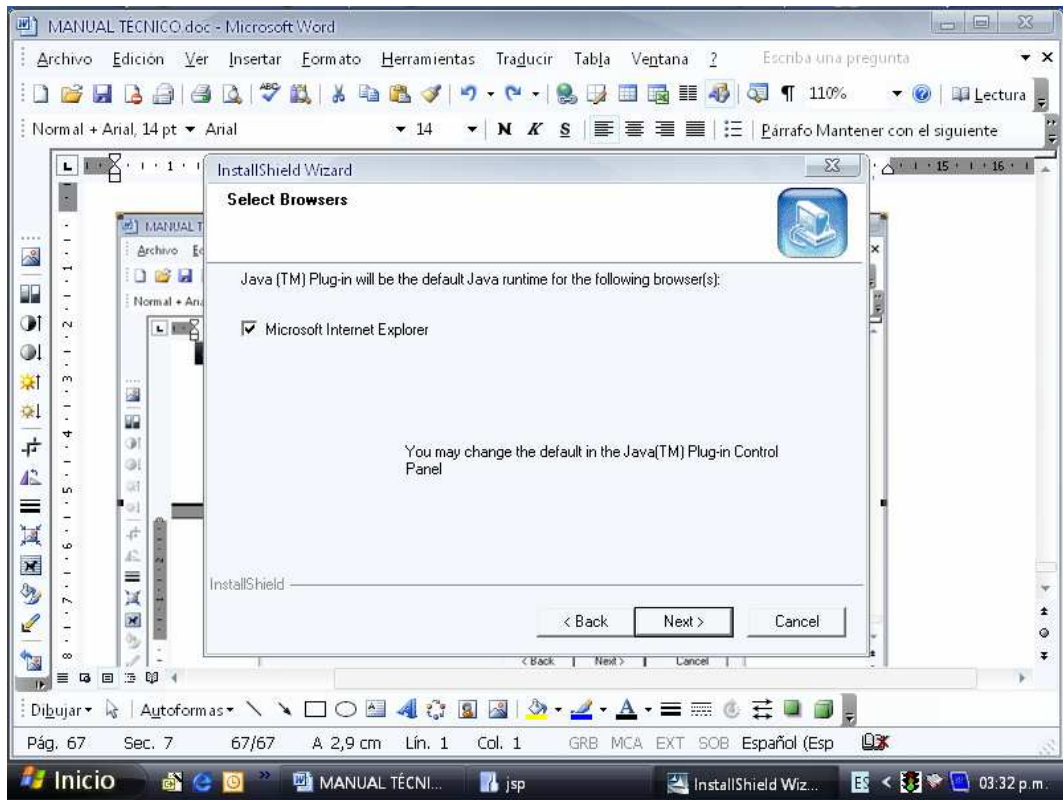


Figura 16: Proceso de instalación de la herramienta j2sdk

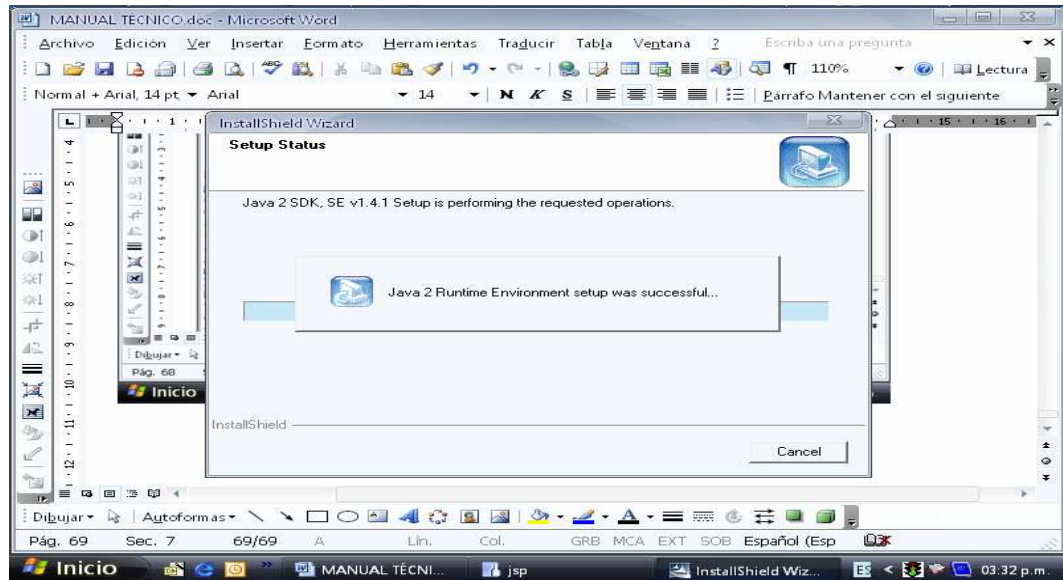
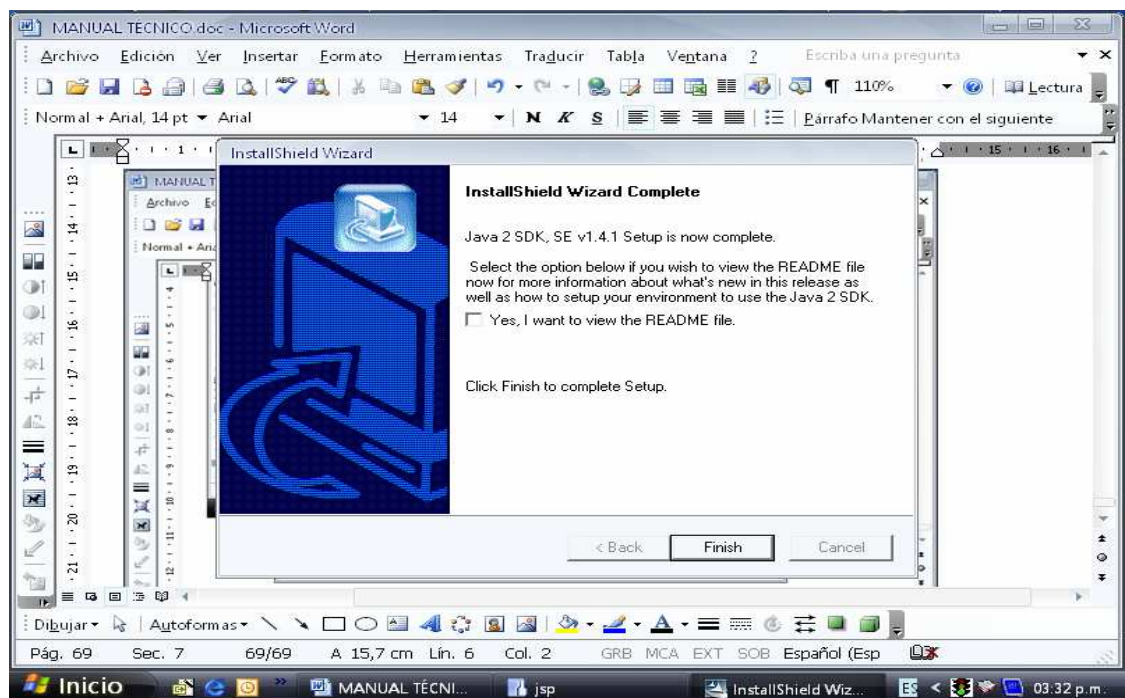
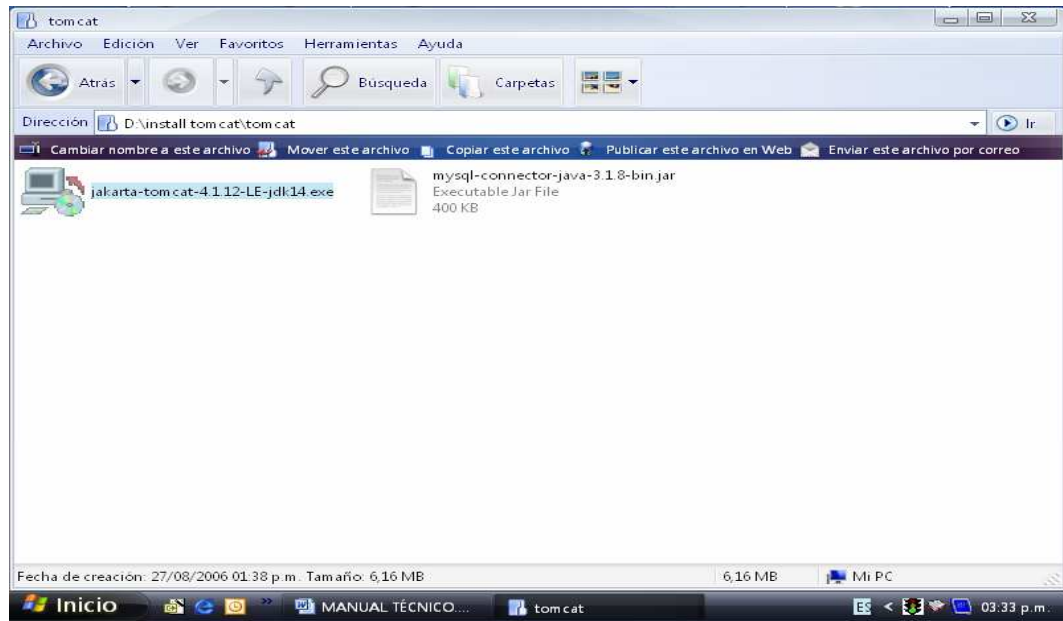


Figura 17: Mensaje de finalización de instalación de la herramienta



5. Instalar el servidor Web Apache Tomcat 4.x.

Figura 18: Instalar el servidor Web apache tomcat 4.1.12



6. Verificar que reconozca la carpeta de la herramienta de desarrollo de Java J2sdk 1.4.1.

Figura 19: Mensaje de verificación del reconocimiento de la carpeta donde esta instalado la herramienta j2sdk 1.4.1

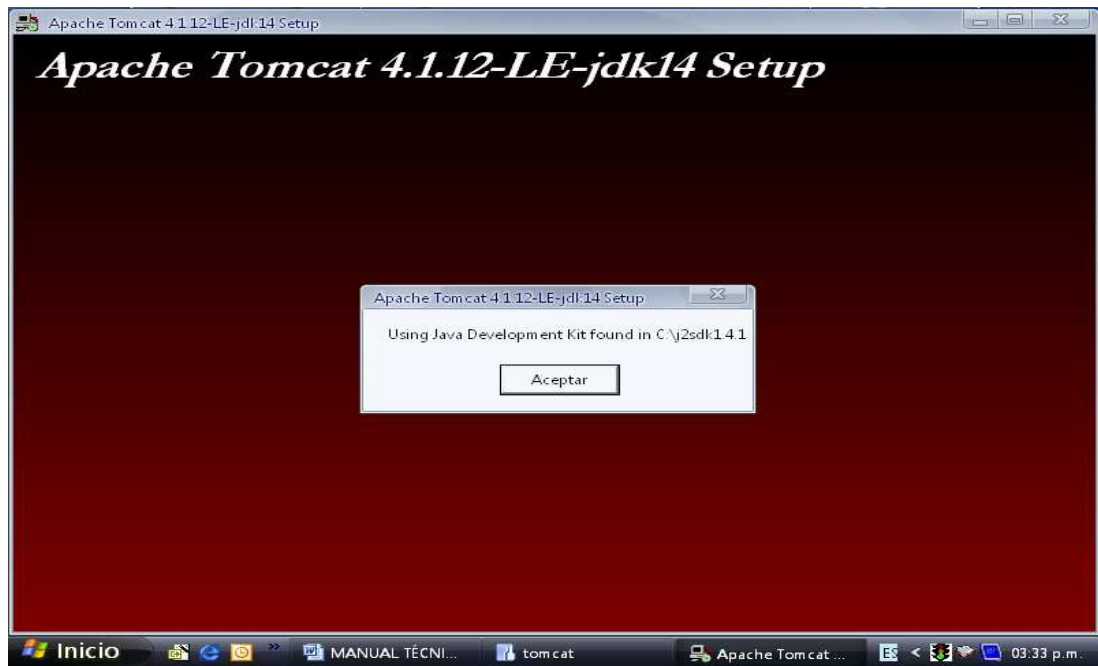


Figura 20: Mensaje inicial de la condiciones legales para la instalación del servidor Web apache tomcat 4.1.12

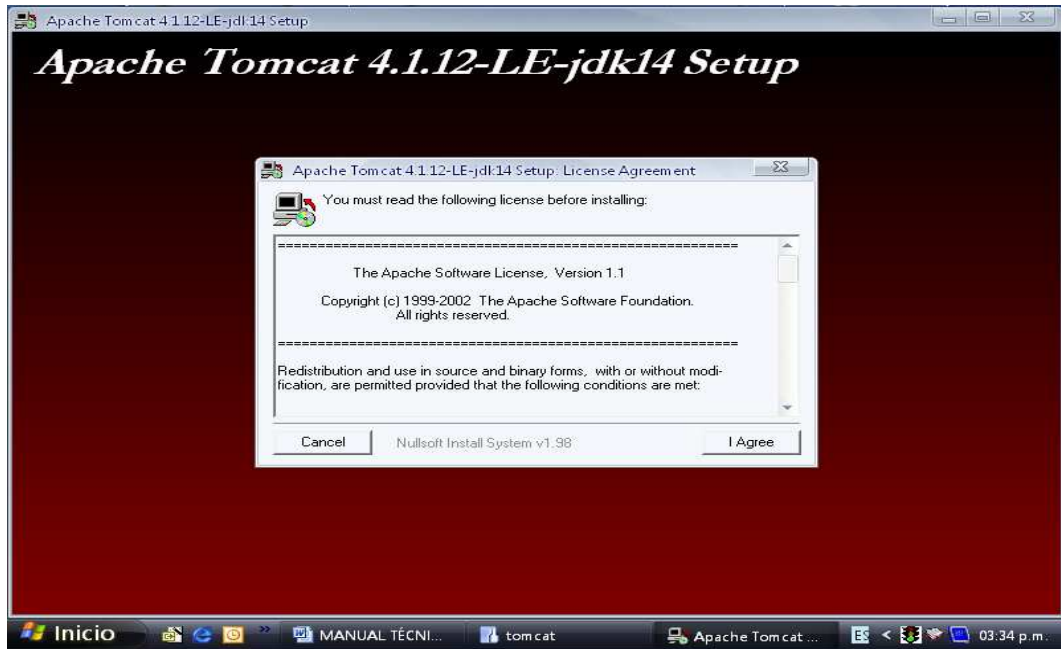


Figura 21: Mensaje para seleccionar el tipo de instalación del servidor Web apache tomcat

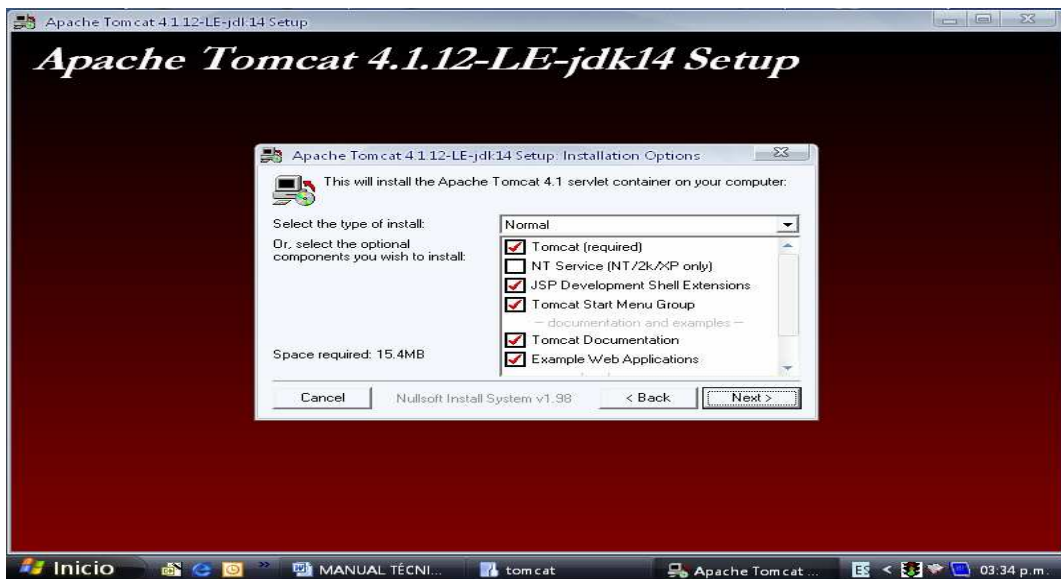


Figura 22: Mensaje para la ruta de instalación del servidor Web apache tomcat 4.1.12

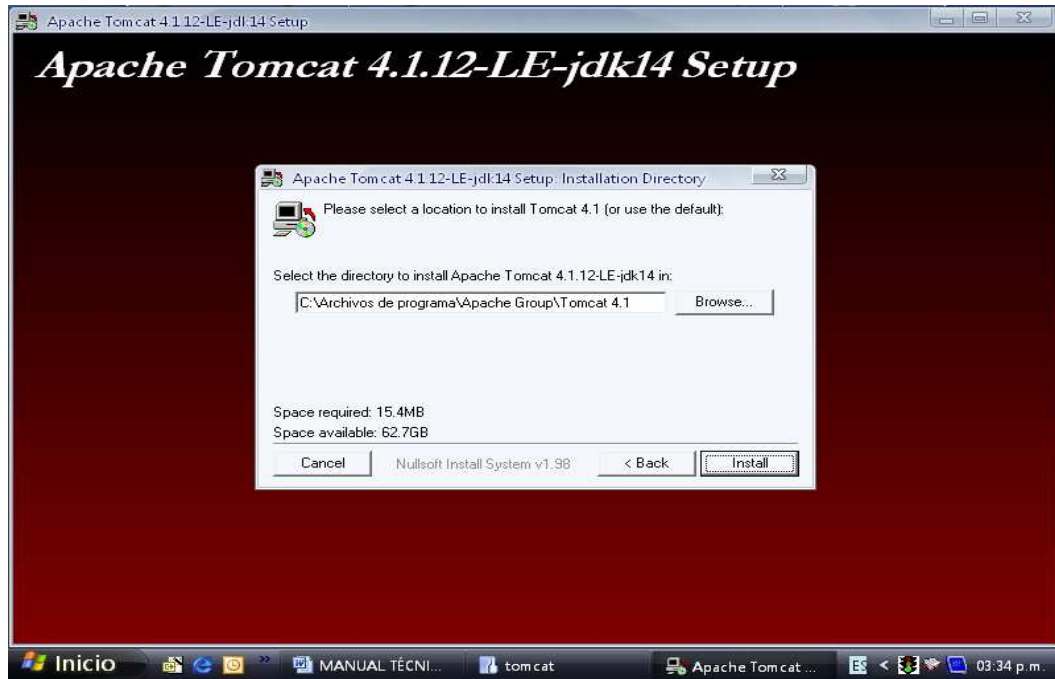
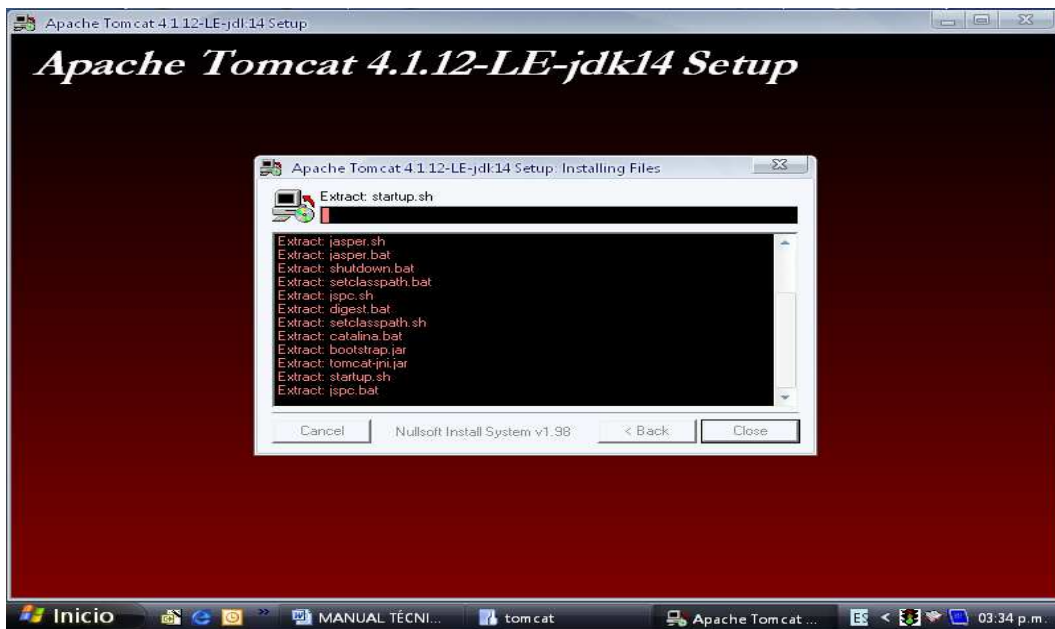


Figura 23: Proceso de instalación del servidor web.



7. Ingresar el nombre del usuario y la contraseña del administrador (Name: root, Password: root).

Figura 24: Ingresar el nombre del usuario y la contraseña del administrador para la instalación del servidor web

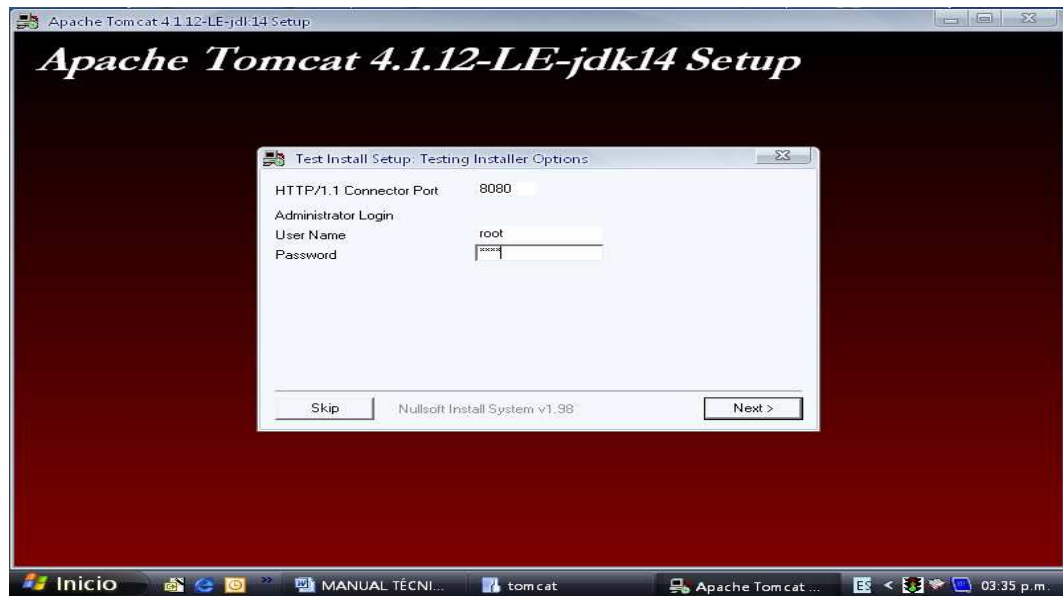
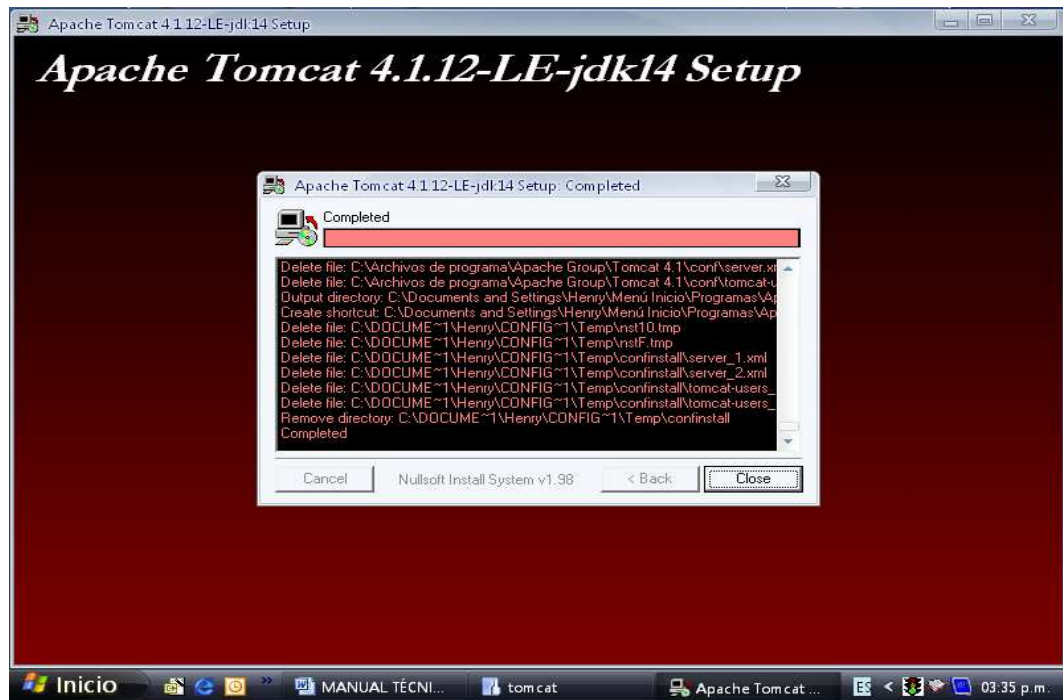


Figura 25: Proceso de instalación del servidor web



8. Ejecutar o subir El servidor Web Apache Tomcat desde su ventana de grupo.

Figura 26: Ejecución del servidor Web desde su ventana de grupo

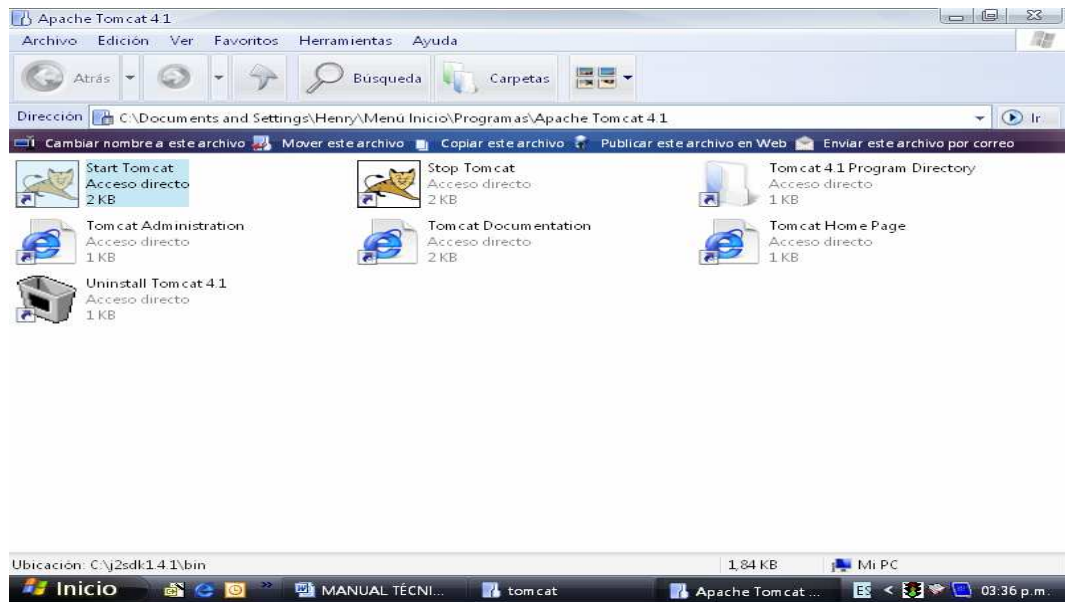
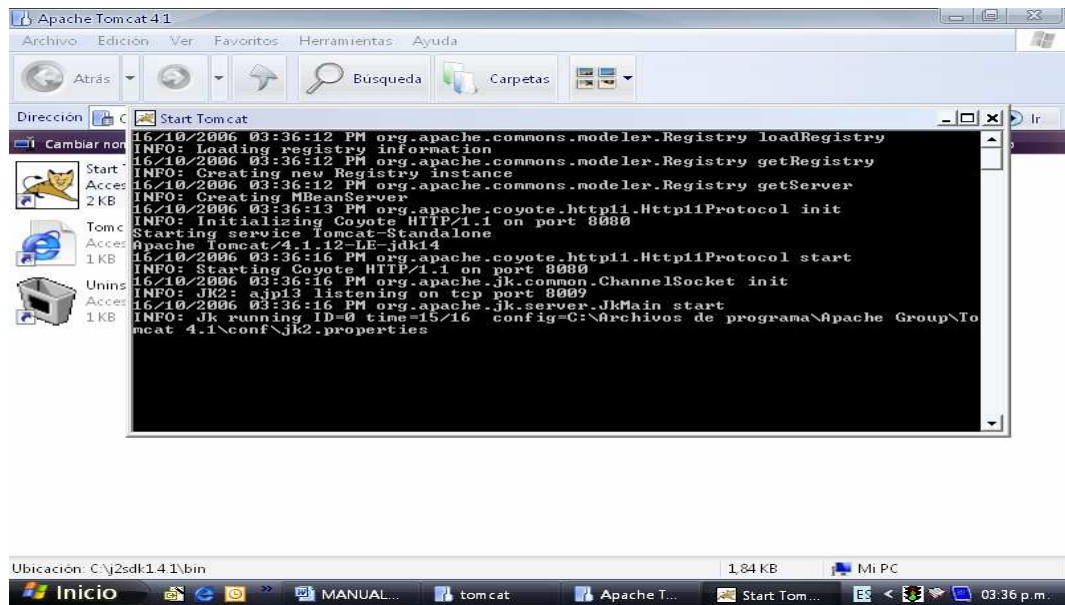


Figura 27: Ejecución del servidor Web apache tomcat



9. Almacenar en la carpeta C:\j2sdk1.4.x\jre\lib\ext\ el controlador de Mysql para java Mysql-conector-java-3.1.x-bin.jar, el cual contiene los drivers de conectividad a bases de datos para Java (JDBC).

Figura 28: Almacenar en la carpeta C:\J2SDK1.4.X\JRE\LIB\EXT\ el controlador de mysql para java MYSQL-CONECTOR-JAVA-3.1.X-BIN.JAR

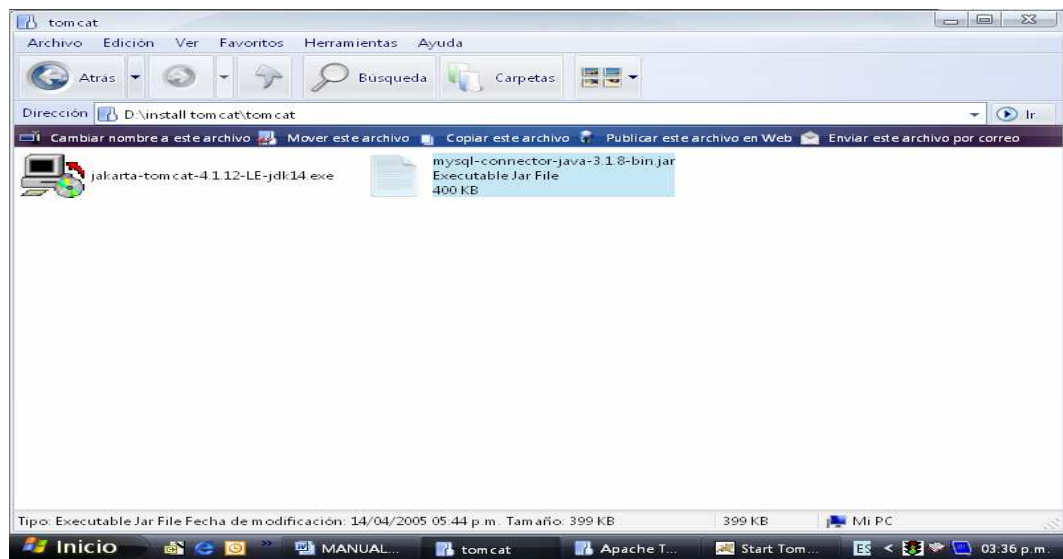
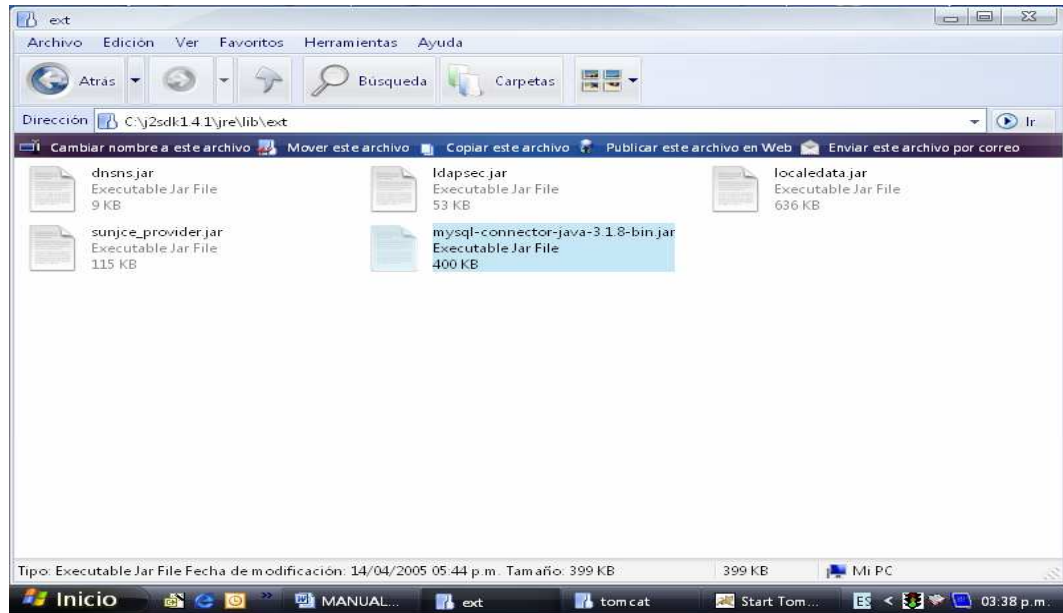


Figura 29: Archivo almacenado en la carpeta de la herramienta j2sdk



10. Los archivos fuentes de la aplicación deberá ir almacenados en la siguiente ruta:

C:\Archivos de programa \ apache Group\ Tomcat 4.1\ webapps\ examples.

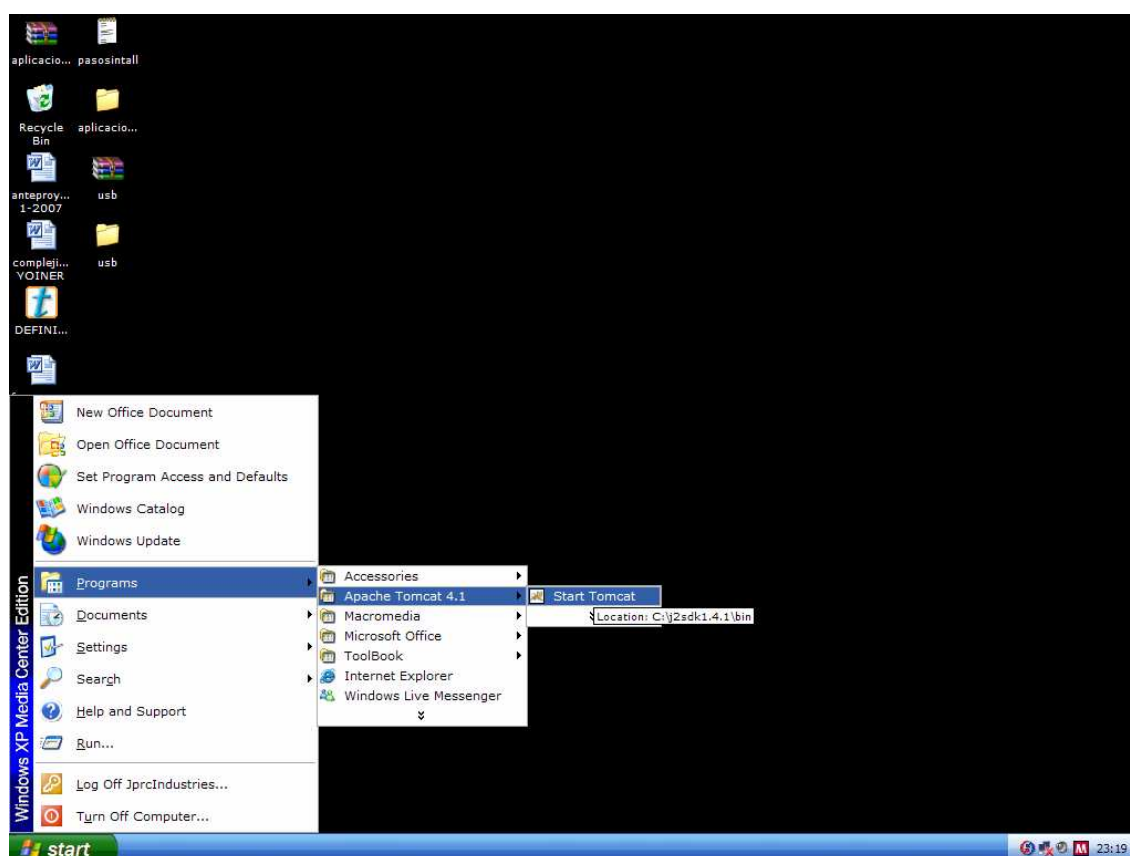
11. Abrir el navegador web (internet Explorer) y escribir en la barra de direcciones <http://localhost:8080/examples>

ANEXO B

MANUAL DE USUARIO

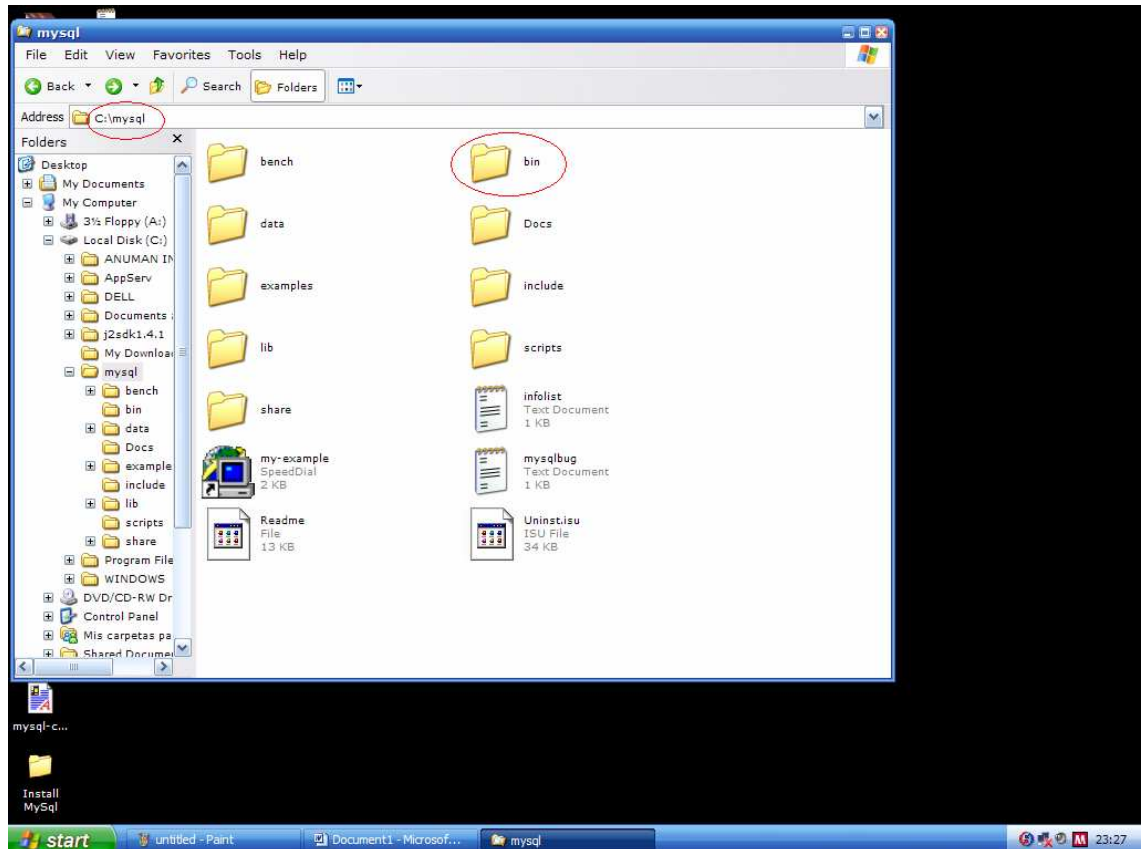
1. Para empezar a utilizar el software después de realizar la instalación, el primer paso es subir el servicio de Apache Tomcat.

Figura 1. Iniciar Apache Tomcat



2. Después de subir el servicio, debe iniciar MySQL, para esto debe entrar en mi PC, buscar la carpeta mysql que está alojada en la raíz C: Después dentro de la carpeta mysql debe buscar la carpeta bin.

Figura 2. Iniciar MSQl



3. después de ubicar la carpeta bin debe buscar el ejecutable "winmysqladmin"

Figura 3. Ejecutar winmysqladmin

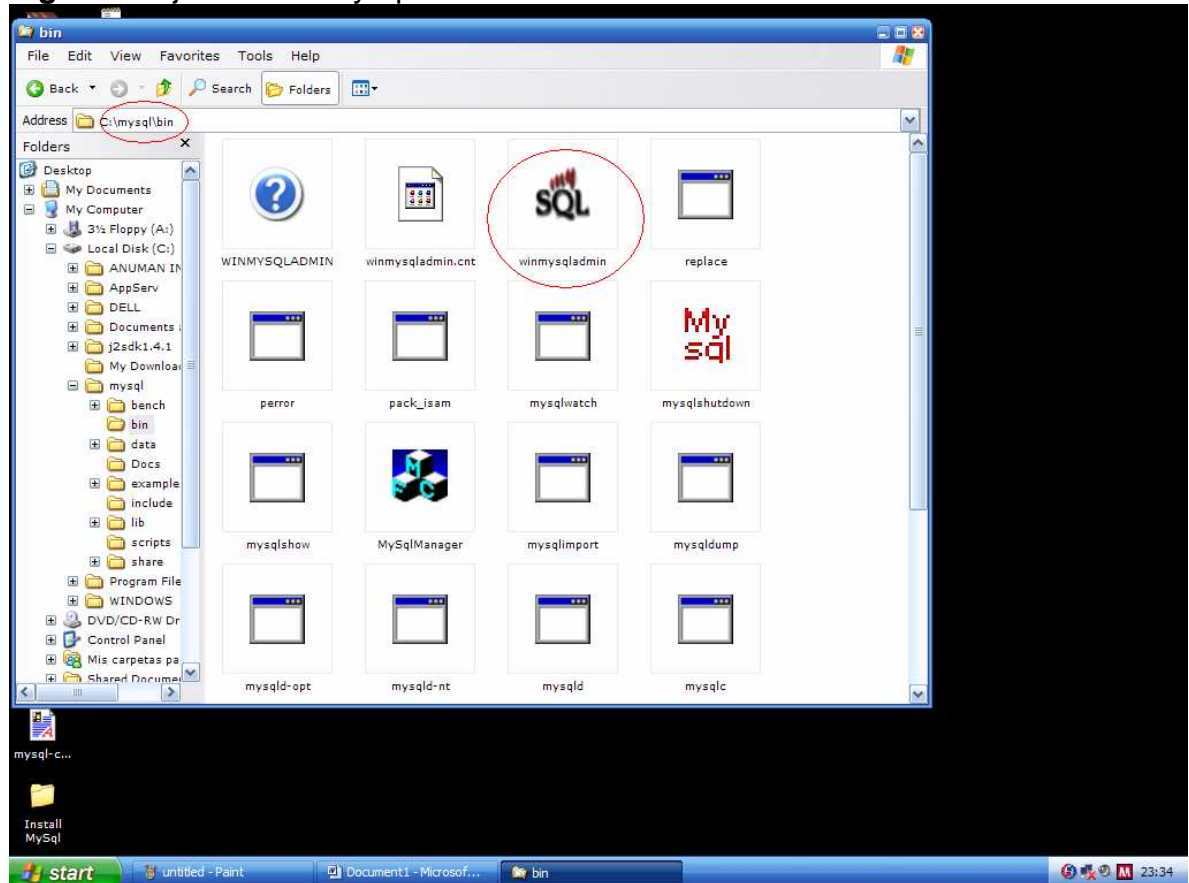
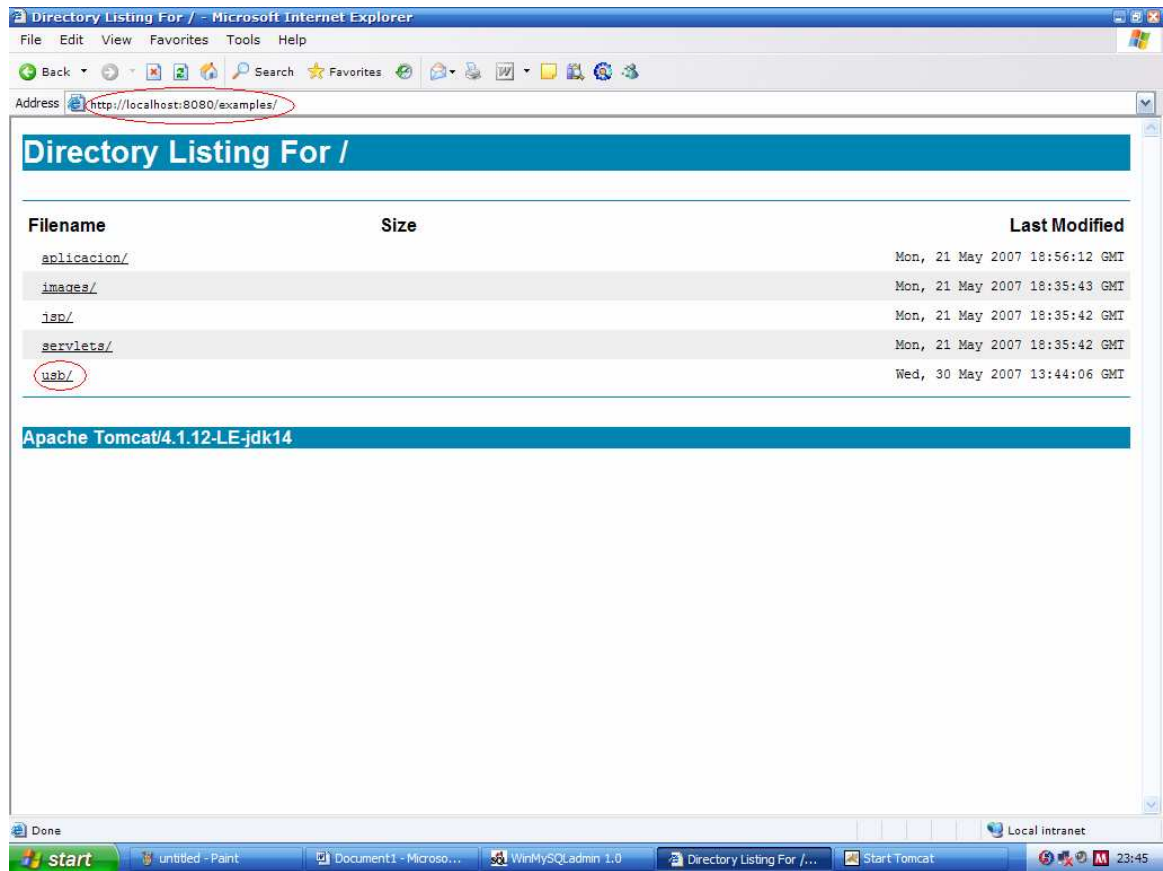


Figura 4. Servicio de Base da Datos arriba



4. Ahora puede iniciar la aplicación, para esto debe entrar al browser y poner la siguiente dirección `http://localhost:8080/examples` y escoger la aplicación USB

Figura 5. Iniciar aplicación



5. Para iniciar la sesión de trabajo debe ingresar su código USB y escoger un grupo de trabajo

Figura 6. Inicio de Sesión



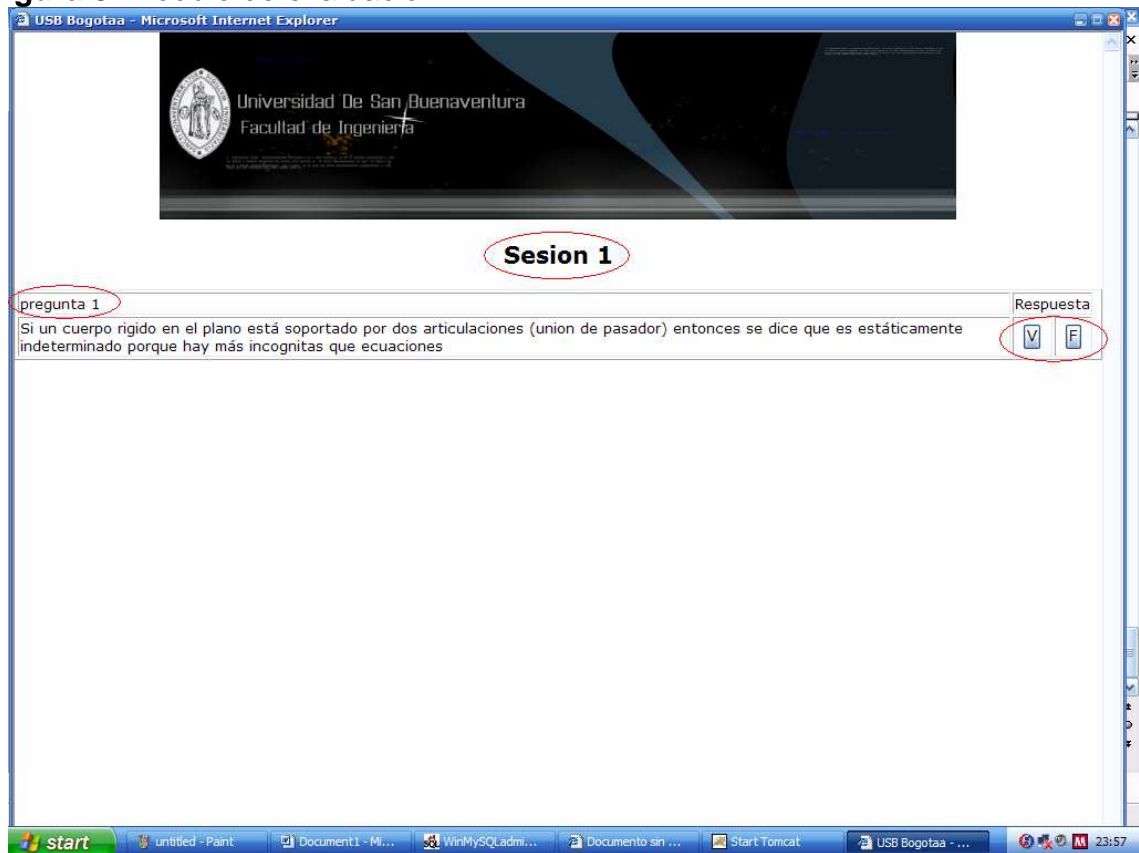
- Después de la iniciar la sesión puede empezar a navegar libremente por todos los temas, cada botón lo llevará al contenido sobre el tema. Cuando lo desee puede entrar al módulo de evolución haciendo clic en el botón de abajo

Figura 7. Navegación y entrar al módulo de evaluación



7. En el módulo de evaluación se le indicará en qué sesión se encuentra, el número de pregunta que está respondiendo, y al lado derecho las opciones de respuesta, cuando escoja una automáticamente pasará a la siguiente inferencia. Al terminar de responder las 4 inferencias volverá a pantalla de Log-in

Figura 8. Módulo de evaluación



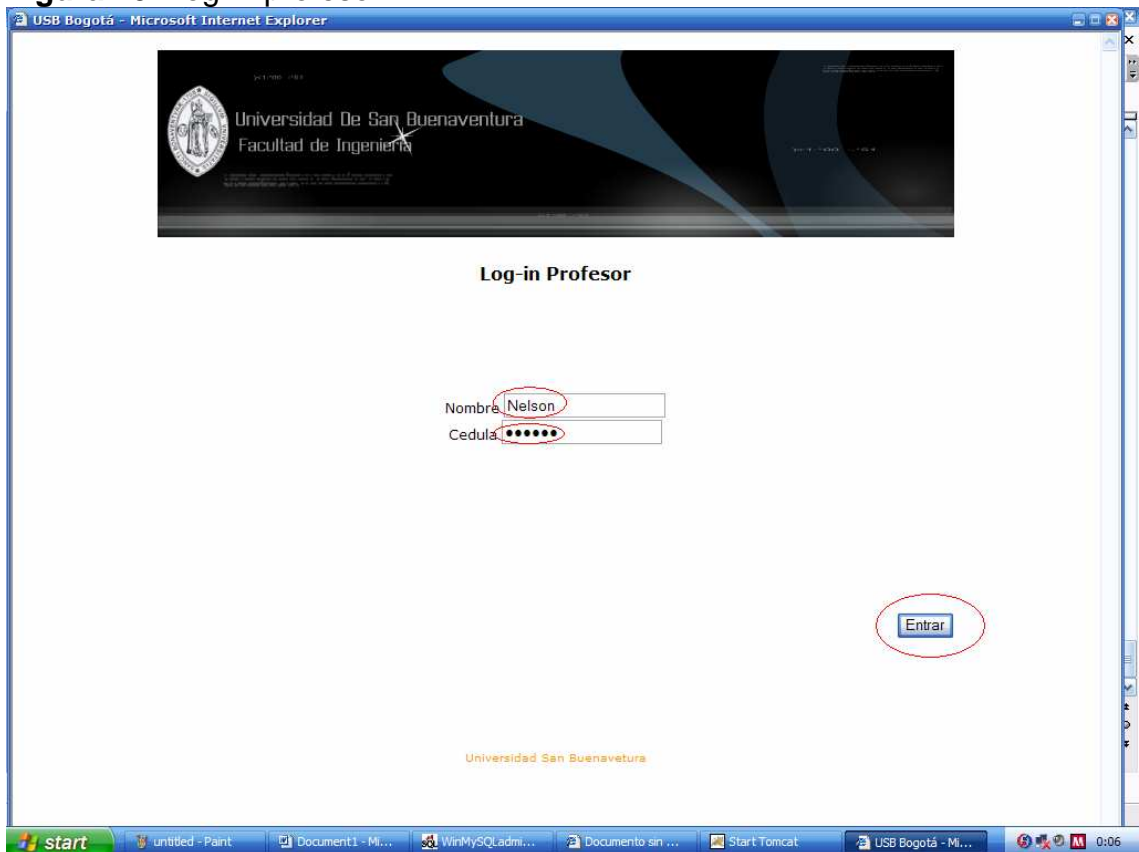
8. Para ir al módulo de profesor debe hacer click donde está el link del profesor en la pantalla de Log-in.

Figura 9. Entrar Módulo Profesor



9. Para poder ingresar al módulo de informes debe hacer el Log-in con el nombre y número de cédula en este caso se debe usar Nombre: "Nelson", Cedula: "79521448"

Figura 10. Log-in profesor



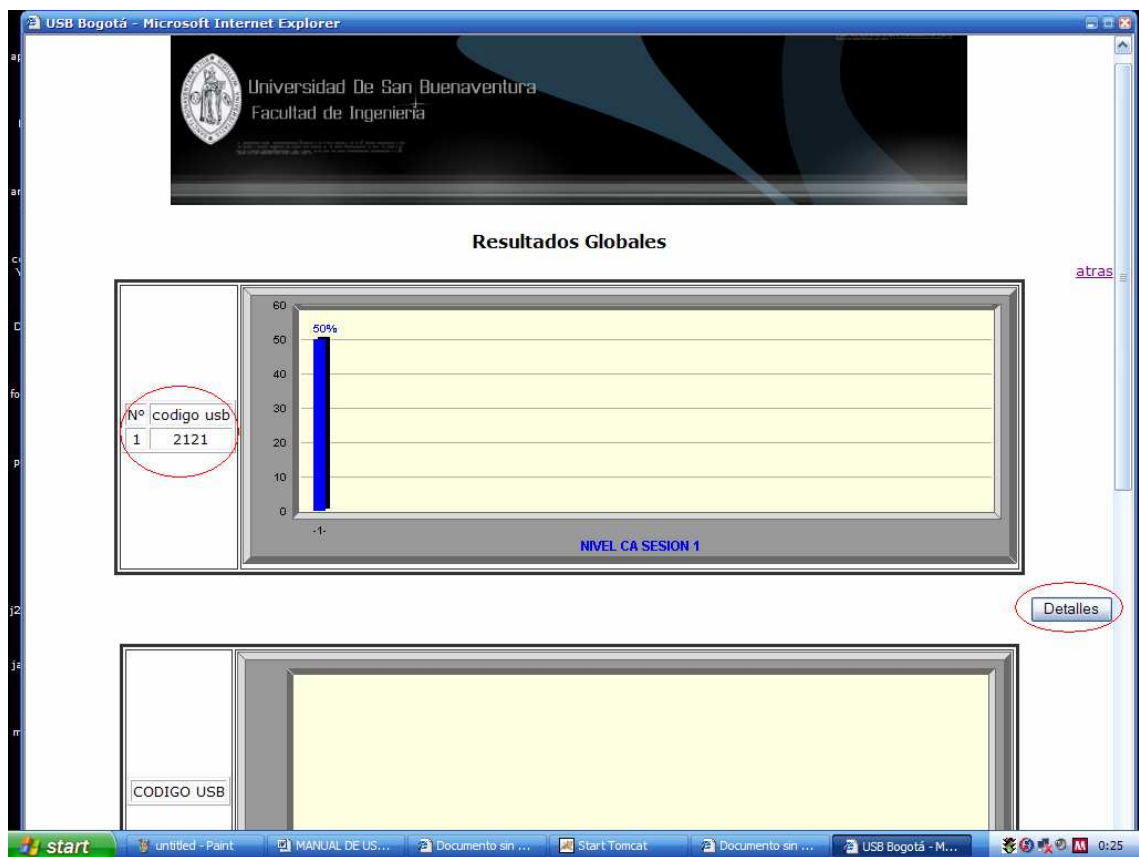
- Después de realizar el log-in puede escoger los diferentes informes que genera el prototipo, también encontrará el botón para ver el informe de la navegación que han tenido los usuarios en el contenido del aplicativo. El siguiente botón es para formatear la Base de Datos una vez que halla terminado de realizar todas las sesiones de trabajo con los usuarios y hacer una nueva muestra con otro grupo (semestre)

Figura 11. Menú de informes



- Después de escoger un informe el prototipo genera un informe con una gráfica sobre los resultados globales de los estudiantes, asociando su código al lado izquierdo de la tabla. El botón de detalles desplegará un informe detallado donde encontrará el comportamiento individual de cada estudiante. en este informe encontrará una gráfica y habilitado el botón de detalles para las tres sesiones de trabajo que debe cumplir cada grupo.

Figura 12. Informe resultados Globales



12. Después de escoger la opción de informe detallado podrá encontrar el título donde encontrará asociado el código del estudiante, la sesión que trabajó y que grupo esta trabajando. Enseguida encontrará una tabla que contiene la inferencia que se le presentó con la respuesta que dio comparada con la correcta más el tiempo que tomó para responder. Por ultimo la calificación del estudiante

Figura 13. Informe detallado

USB Bogotaa - Microsoft Internet Explorer

Universidad De San Buenaventura
Facultad de Ingeniería

1CA

[atras](#)

Usuario: 2121 en Sesion 1 de Nivel CA

PREGUNTAS	RESPUESTA USUARIO	RESPUESTA CORRECTA	TIEMPO
Si el soporte es un apoyo fijo (empotramiento) en 2D entonces las reacciones son una fuerza horizontal y una fuerza vertical y un torque perpendicular al plano	V	V	00:00:01
Si el apoyo es una articulacion en 2D entonces la reaccion es una fuerza con linea de accion desconocida	F	V	00:00:00
Si el apoyo es una articulacion en 2D entonces la reaccion es una fuerza con linea de accion desconocida	V	V	00:00:01
Si la union es mediante una articulacion en 2D entonces las reacciones son dos fuerzas, una horizontal y una vertical; o una fuerza con linea de accion desconocida	F	V	00:00:02

PROMEDIO 50.0%

start | unitled - Paint | MANUAL DE US... | Documento sin ... | Start Tomcat | Documento sin ... | USB Bogotaa - ... | 0:31

ANEXO C

CÓDIGO FUENTE DE LA CONSTRUCCIÓN DEL APPLLET

```
//programa en java que dibuja un cuadro estadistico en barras
```

```
import java.awt.Graphics;//librerias grafica
import java.awt.Color;
import java.awt.Font;//librerias de formatos
import java.awt.FontMetrics;
import java.io.*;//libreria standar
import java.lang.*;
import java.net.URL;
```

```
public class Barchart2 extends java.applet.Applet
{
    static final int VERTICAL = 0;
    static final int HORIZONTAL = 1;
    static final int SOLID = 0;
    static final int STRIPED = 1;

    int orientation;
    String title;
    Font titleFont;
    FontMetrics titleFontMetrics;
    Font smallFont;
    FontMetrics smallFontMetrics;
    int columns;
    int values[];//
    Object colors[];
    Object labels[];
    int styles[];
    int maxLabelWidth = 0;
    int barWidth;
    int barSpacing = 9;
    int max = 0;//
    boolean printval = true;
    boolean showavg = true;
    Color purple=new Color(51,31,101);
    Color beige=new Color(156,0,0);
    Color darkGreen=new Color(0,100,0);
    Color vinot=new Color(156,0,0);

    public synchronized void init()//inicializacion de variables
```

```

{
    String rs;
    titleFont = new java.awt.Font("Helvetica", Font.BOLD, 12);
    titleFontMetrics = getFontMetrics(titleFont);
    smallFont = new java.awt.Font("Helvetica", Font.BOLD, 10);
    smallFontMetrics = getFontMetrics(smallFont); // obtiene los
parametros del applet
    title = getParameter("title");

    if (title == null)
    {
        title = "Chart";
    }

    rs = getParameter("columns");
    if (rs == null)
    {
        columns = 5;
    }
    else
    {
        columns = Integer.parseInt(rs);
    }
    rs = getParameter("orientation");
    if (rs == null)
    {
        orientation = VERTICAL;
    }
    else if (rs.toLowerCase().equals("vertical"))
    {
        orientation = VERTICAL;
    }
    else if (rs.toLowerCase().equals("horizontal"))
    {
        orientation = HORIZONTAL;
    }
    else
    {
        orientation = VERTICAL;
    }
    rs = getParameter("printval");

```

```

if (rs == null)
{
    printval = true;
}
if (rs.toLowerCase().equals("no"))
{
    printval = false;
}
} // procesa los parametros y los asigna las variables

values = new int[columns];
colors = new Color[columns];
labels = new String[columns];
styles = new int[columns];
float sum=0.0f;
float avg;

for (int i=0; i < columns; i++)
{ // Captura el valor de la column
    rs = getParameter("C" + (i+1));
    if (rs != null)
    {
        // Invoca el metodo de next() en getDoubles class
        // para asignar la entrada de variables tipo dobles

        getDoubles getD = new getDoubles(rs);
        double dnum = 0.0d;

        try
        {
            dnum = getD.next();
        }
        catch(IOException e)
        {
            dnum=0.0d;
        }
        values[i] = (int) dnum;
        System.out.println("value= " + values[i]);
    }
    if (values[i] > max)
    {
        max = values[i];
    }
}

```

```

    }
    sum += values[i];
    avg=sum/((float)(i+1));
    rs = getParameter("C" + (i+1) + "_label");
    labels[i] = (rs == null) ? "" : rs;
    maxLabelWidth =
Math.max(smallFontMetrics.stringWidth((String)
(labels[i])), maxLabelWidth);
    // Parse the bar style
    rs = getParameter("C" + (i+1) + "_style");

    if (rs == null || rs.toLowerCase().equals("solid"))

    {
        styles[i] = SOLID;
    }
    else if (rs.toLowerCase().equals("striped"))
    {
        styles[i] = STRIPED;
    }
    else
    {
        styles[i] = SOLID;
    }
}
// captura el parametro de color de la columna
// por defecto azul

rs = getParameter("C" + (i+1) + "_color");

if (rs != null)
{

    if (rs.equals("red"))
    {
        colors[i] = Color.red;
    }
    else if (rs.equals("green"))
    {
        colors[i] = Color.green;
    }
    else if (rs.equals("darkGreen"))
    {

```

```
        colors[i] = darkGreen;
    }
    else if (rs.equals("blue"))
    {
        colors[i] = Color.blue;
    }
    else if (rs.equals("pink"))
    {
        colors[i] = Color.pink;
    }
    else if (rs.equals("beige"))
    {
        colors[i] = beige;
    }
    else if (rs.equals("magenta"))
    {
        colors[i] = Color.magenta;
    }
    else if (rs.equals("cyan"))
    {
        colors[i] = Color.cyan;
    }
    else if (rs.equals("white"))
    {
        colors[i] = Color.white;
    }
    else if (rs.equals("yellow"))
    {
        colors[i] = Color.yellow;
    }
    else if (rs.equals("gray"))
    {
        colors[i] = Color.gray;
    }
    else if (rs.equals("darkGray"))
    {
        colors[i] = Color.darkGray;
    }
    else if (rs.equals("purple"))
    {
        colors[i] = purple;
    }
}
```

```

        else if (rs.equals("black"))
        {
            colors[i] = Color.black;
        }
        else if (rs.equals("vinot"))
        {
            //colors[i] = Color.vinot;
        }
        else
        {
            colors[i] = Color.blue;
        }
    }
    else
    {
        colors[i] = Color.blue;
    }
}
}

} //configura los tamaños segun la orientacion del grafico
switch (orientation)
{

    case VERTICAL:
    default:
        barWidth = maxLabelWidth - 1;
        resize(Math.max(columns * (barWidth + barSpacing),
titleFontMetrics.stringWidth(title)) + titleFont.getSize() + 5, (int)(max) + (2 *
titleFont.getSize()) + 5 + titleFont.getSize());

        break;

    case HORIZONTAL:
        barWidth = smallFont.getSize();
        resize(Math.max(((int)max) + titleFontMetrics.stringWidth(" " +
(int)max), titleFontMetrics.stringWidth(title)) + maxLabelWidth + 5, (columns *
(barWidth + barSpacing)) + titleFont.getSize() + 10);

        break;
    }
} // ***** Fin del metodo init() *****

//***** Metodo paint() *****

```



```

public synchronized void paint(Graphics g)
{
    int i, j;
    int cx, cy,cxs,cys;
    char l[] = new char[1];
    int upshift=11;
    float fscale;    // parametro de autoescala
    float hfscale;

    // configura la auto escala

    int xoff = 45;        // x coordenada
    int yoff = 15;        // y coordenada
    int rightmarg = 15; // margen derecho
    int hdeltay=12;
    int delxh=10;

    int zeroy=size().height - 6 - (2 * titleFont.getSize()) -upshift;
    fscale= 1.25f*max / ((float)zeroy-(float)yoff);
    hfscale=1.2f*max / ((float)size().width-(float)xoff -(float)rightmarg);
    // Interval for vertical tic marks in vertical mode
    int interval = max/5;
    int now;
    int ticnow;
    int deltax=6;
    // espacio entre barras

    Color bgcolor=getColorFromRGB("bgcolor");
    if (bgcolor == null)
    {
        bgcolor = Color.lightGray;
        showStatus("NumberFormatException; defaulting to gray
background");
        System.out.println("NumberFormatException; default togray
background");
    }
    Color insetcolor=getColorFromRGB("insetcolor");
    if (insetcolor == null)

```

```

        {
            insetcolor = bgcolor;
            showStatus("NumberFormatException; defaulting to first
background");
            System.out.println("NumberFormatException; default to 1st
background");
        }

```

```

fill3DRect(g, bgcolor, 0, 0, size().width, size().height, 7, true);

```

```

Color colortic=insetcolor.darker();

```

```

    switch(orientation)
    {
        case VERTICAL:

            fill3DRect(g, bgcolor, xoff, yoff, size().width-rightmarg-
xoff,zeroy-yoff+0, 6, false);

            //dibuja el fondo del applet

            g.setColor(insetcolor);
            g.fillRect(xoff+6, yoff+6,size().width-rightmarg-xoff-12,zeroy-
yoff-12);

```

```

            g.setColor(colortic);
            g.setFont(smallFont);

            for (int k=0; k<7; k++)
            {
                now = (int)k*interval;
                ticnow = size().height - (int)(now/fscale) - 7 - (2 *
titleFont.getSize()) - upshift -deltay;

```

```

                if(ticnow >= 0)
                {
                    if(k != 0)
                    {
                        g.setColor(colortic);

```

```

                                g.drawLine(xoff+8,ticnow,size().width-
rightmarg-9,ticnow);
                                }
                                g.setColor(Color.black);
                                g.drawString(""+now, xoff -
smallFontMetrics.stringWidth(""+now)-5,ticnow + smallFont.getSize()/2-1);
                                }
                                }
                                break;

                                case HORIZONTAL:
                                break;
                                }

                                // dibuja el titulo en la parte inferior del applet

Color nowcolor=(Color)colors[0];
g.setColor(nowcolor);
i = titleFontMetrics.stringWidth(title);
g.setFont(titleFont);

switch(orientation)
{
    case VERTICAL:
        g.drawString(title, xoff+Math.max((size().width - i-xoff-10-
rightmarg)/2, 0),size().height - titleFontMetrics.getDescent() -upshift +2);
        break;
    case HORIZONTAL:
        g.drawString(title, Math.max((size().width - i)/2, 0),size().height -
titleFontMetrics.getDescent() -upshift +2);
        break;
}

//dibujar barras

for (i=0; i < columns; i++)
{
    switch (orientation)
    {
        case VERTICAL:
        default:

```

```

        cx = (Math.max((barWidth + barSpacing),
maxLabelWidth) * i)+ barSpacing + xoff + 10;//calcula coordenadas para los labels
de las graficas

```

```

        cy = size().height - (int)(values[i]/fscale) - 1- (2 *
titleFont.getSize()) -deltay;

```

```

        // dibuja el label de la barra
g.setFont(smallFont);
g.setColor(Color.black);
g.drawString((String)labels[i], cx, size().height -
smallFont.getSize()- smallFontMetrics.getDescent()-3 -upshift);
        // dibuja la sombra de la barra

```

```

if (colors[i] == Color.black)
{
    g.setColor(Color.gray);
}

```

```

switch(styles[i])
{
    case SOLID:
        g.setColor(Color.black);
        break;
    case STRIPED:

```

```

        g.setColor(Color.gray);
        break;
}

```

```

(int)(values[i]/fscale));
g.fillRect(cx +4, cy -8 -upshift, barWidth,

```

```

// dibuja la barra

```

```

g.setColor((Color)(colors[i]));

```

```

switch (styles[i])
{

```

```

    case SOLID:
    default:

```

```

        if(values[i] == 0.0f)
        {
            g.drawLine(cx, cy-7-upshift,
cx+barWidth+2,cy-7-upshift);//coordenadas y espesor
        }

        g.fillRect(cx, cy-6 - upshift,
barWidth,(int)(values[i]/fscale));

        break;

        case STRIPED:

        if(values[i] == 0.0f)
        {
            g.drawLine(cx, cy-7-upshift,
cx+barWidth+2,cy-7-upshift);
        }
        {
            int steps = (int)( values[i] /fscale) / 2;
            int ys;
            int cyy=cy-6-upshift;
            for (j=0; j < steps+1; j++)
            {
                ys = cyy + (2 * j);
                g.drawLine(cx, ys, cx + barWidth,
ys);
            }
        }
        break;
    }

    // imprime el valor de la barra si el parametro es yes
    if( printval == true )
    {
        g.setFont(smallFont);
        g.drawString(" " + values[i]+" %" , cx, cy -upshift-
smallFontMetrics.getDescent()-9);

        g.setFont(titleFont);
    }
}

```

```

        break;
        case HORIZONTAL:
            g.setFont(smallFont);

            cy = ((barWidth + barSpacing) * i) + barSpacing +
hdeltay;

            cx = maxLabelWidth + 1;
            cx += Math.max((size().width - (maxLabelWidth + 1 +
smallFontMetrics.stringWidth("" + (int)max) + ( (int)( max/hfscale)  ))) / 2, 0);

            g.setColor(Color.black);
            g.drawString((String)labels[i], cx - maxLabelWidth - 1 -
delxh,cy + smallFontMetrics.getAscent());

            if (colors[i] == Color.black)
            {
                g.setColor(Color.gray);
            }

            switch(styles[i])
            {
                case SOLID:
                    g.setColor(Color.black);
                    break;
                case STRIPED:
                    // Color nowcolor=(Color)colors[i];
                    // Color lighter=nowcolor.darker();
                    g.setColor(Color.gray);
                    break;
            }

            g.fillRect(cx + 7 -delxh,cy + 4,+(int)(values[i]
/hfscale)+3,barWidth);

            g.setColor((Color)(colors[i]));
            switch (styles[i])
            {

                case SOLID:

```

```

        default:
            g.fillRect(cx+4 -delxh, cy, (int)(values[i]/hfscale)+4,
barWidth);
            break;
            // or the striped style case
            case STRIPED:
            {
                int steps = (int)(values[i] / hfscale) / 2;
                int ys;
                int cxx=cx+4-delxh;
                for (j=0; j < steps+2; j++)
                {
                    ys = cxx + (2 * j);
                    g.drawLine(ys, cy, ys, cy +
barWidth);
                }
            }
            break;
        }// Print the value if applet parameter prinval is yes
        if(printval == true)
        {
            g.drawString(" " + values[i],cx -delxh+
(int)( values[i] / hfscale) + 12,cy + smallFontMetrics.getAscent());
        }
        break;
    }
}

// ***** fin del metodo paint() *****
// ***** Method fill3DRect *****
//// metodo para generar rectangulos 3D
//
// usa metodos en la libreria java.awt.Graphics y java.awt.Color//
// fill3DRect(g, x, y, w, h, thick, raised)//metodo general donde
//
// g = variable graficos
// x = variable para la coordenada desde la esquina superior derecha
// y = variable para la coordenada desde la esquina superior izquierda
// w = variable para el ancho de la barra en pixeles

```

```

// h = variable para el alto de la barra en pixeles
// thick = variable para el espesor del borde
// raised = variable booleana
// 3D perspectiva
//
// *****

```

```

public void fill3DRect(Graphics g, Color c, int x, int y, int w, int h, int thick,
boolean raised)
{

```

```

    Color brighter = c.brighter();
    Color darker = c.darker();
    g.setColor(c);
    g.fillRect(x+thick, y+thick, w-(2*thick), h-(2*thick));
    g.setColor(raised ? brighter : darker);
    g.fillRect(x,y,w,thick);
    g.fillRect(x,y,thick,h);
    g.setColor(raised ? darker : brighter);
    int X[] = new int[4];
    int Y[] = new int[4];

```

```

    X[0] = x;
    Y[0] = y+h;
    X[1] = x+thick;
    Y[1] = y+h-thick;
    X[2] = x+w;
    Y[2] = y+h-thick;
    X[3] = x+w; Y[3] = y+h;
    g.fillPolygon(X,Y,4);
    X[0] = x+w-thick;
    Y[0] = y+thick;
    X[1] = x+w; Y[1] = y;
    X[2] = x+w; Y[2] = y+h;
    X[3] = x+w-thick; Y[3] = y+h;
    g.fillPolygon(X,Y,4);
    g.setColor(darker);
    g.drawRect(x,y,w,h);
    g.drawRect(x+thick, y+thick, w-2*thick, h-2*thick);
    g.setColor(raised ? darker : brighter);
    g.drawLine(x,y,x+thick, y+thick);

```



```

        g.setColor(raised ? brighter : darker);
        g.drawLine(x+w, y+h, x+w-thick, y+h-thick);
        g.setColor(c);
    }

    // ***** fin metodo fill3DRect *****
    // ***** Metodo getColorFromRGB *****
    // metodo que convierte un parametro hex RGB en color
    // retorna nulo en excepciones
    // *****

protected Color getColorFromRGB(String rgb)
{
    String value = this.getParameter(rgb);
    int hexcolor;

    try
    {
        hexcolor=Integer.parseInt(value,16);
    }
    catch(NumberFormatException e)
    {
        return null;
    }
    return new Color(hexcolor);
}
// ***** fin del metodo getColorFromRGB *****
}

```

ANEXO D

DICCIONARIO DE DATOS CONSULTAS Y CÓDIGO JSP

Pagina login.jsp

Construcción del combo dinámico

consulta ="select *from Grupo order by cod_grupo asc;";//sentencia que lista los nombres de //grupo(AA,BB,CA...)

```
<select name="grupo">//inicia combo en html
```

```
<%//abre script jsp
```

```
try
```

```
{
```

```
tabla=instruccion.executeQuery(consulta);//se ejecuta la sentencia"select" y se almacena en la //variable tabla
```

```
while(tabla.next())//se recorren los resultados
```

```
{
```

```
    out.println("<option>" +tabla.getString(1)+"</option>");//se imprimen las opciones del //combo en jsp con la instrucción out.println()
```

```
}
```

```
}
```

```
catch(SQLException e)//manejo de excepciones, si la base de datos no existe o si la sentencia //sql tiene error de sintaxis
```

```
{out.println(e);}
```

```
    %>//cierra script jsp
```

```
<%@ page import="java.io.*,java.util.*,java.net.*,java.sql.*" %><%//Invocando librerias
```

```
%>
```

```
<html>
```

```
<head>
```

```
<title>Estatica</title>
```

```
<style type="text/css">
```

```
<!--
```

```
.Estilo24 {
```

```
    font-weight: bold;
```

```
    font-family: Tahoma;
```

```
}
```

```
.Estilo54 {font-weight: bold; font-size: 12px;}
```

```
.Estilo55 {font-family: Tahoma}
```

```
.Estilo56 {font-family: Tahoma; font-size: 12px; font-weight: bold; }
```

```
.Estilo57 {font-size: 12px}
```

```
-->
```

```

</style>
</head>
<body>
<p align="center">
  <%
    Connection canal = null;//canal de conexion a la Base de Datos
    ResultSet tabla= null;//Variable que almacena resultados de consultas
realizadas
    Statement instruccion=null;
    String consulta = "";//guardar consultas cadena de caracteres
    String usb="";//almacena el codigo del Estudiante
    String grupo="";//almacena el grupo elegido
    String cod_grupo="";
    int contador=0;//centinelas
    int contador1=0,contador2=0,contador3=0;
    String tiempo="";//variable que me va a capturar el tiempo como cadena de
caracteres
    int sesion=0;//variable que me va ha capturar la sesion del estudiante

    String tema="";

    String strcon = "jdbc:mysql://localhost/usb?user=root&password=root";//ruta
de la base de datos

try
{//inicializar conexion a la base de datos
Class.forName("com.mysql.jdbc.Driver").newInstance();
canal=DriverManager.getConnection(strcon);
instruccion =
canal.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCU
R_UPDATABLE);
}
catch(java.lang.ClassNotFoundException e)
{
}
catch(SQLException e)
{
};
usb= request.getParameter("usb");//asignacion del codigo usb del estudiante a la
variable usb
grupo= request.getParameter("grupo");//asignacion del nombre del grupo que
eligió el estudiante a la variable grupo

```

```

tema=request.getParameter("tema");//asignacion del tema que va a navegar el
usuario
tiempo=request.getParameter("tiempo");//variable que me almacena el tiempo de
navegacion en las paginas
if(request.getParameter("salir")!=null)//evalua el evento salir que proviene de
alguna pagina del tema
{
    consulta="insert into registro_navegacion
(cod_estudiante,cod_tema,tiempo_nav) values
("+\""+usb+"\", "+\""+tema+"\", "+\""+tiempo+"\"+");";//sentencia sql que me
registra dentro de la tabla registro_navegacion la pagina y el tiempo de visita por el
usuario
try
{
int n= instruccion.executeUpdate(consulta);//ejecucion de la sentencia sql
} //fin try no usar ; al final de dos o mas catches
catch(SQLException e)
{
};
}
if(request.getParameter("entrar")!= null || request.getParameter("usb")!= null ||
request.getParameter("grupo") != null ||contador1>0)//evalua el ingreso por primera
vez al //mapa de navegacion
{
consulta="select *from estudiante where cod_estudiante like
"+\""+usb+"\"+\"+\"";//esta sentencia me busca el registro del estudiante por medio
de su codigo usb si es la primera vez que ingresa o ya esta registrado
try
{
tabla = instruccion.executeQuery(consulta);//se ejecuta la sentencia
// ciclo de lectura del resultset
while(tabla.next()) //recoger matriz de resultados
{
contador++;//variable que hace la funcion de centinela (validacion)
}; // fin while
} //fin try no usar ; al final de dos o mas catches
catch(SQLException e)
{out.println(e);
};
if(contador==0)//si contador es igual a cero quiere decir que el usuario no esta
registrado
{

```

```

consulta="insert into estudiante values("+\""+usb+\""+");";//esta sentencia me
ingresa el registro del nuevo estudiante dentro de la tabla estudiante
try
{
int n= instruccion.executeUpdate(consulta);//instruccion ejecutar la sentencia sql
de insertar //registros a la bd
} //fin try no usar ; al final de dos o mas catches
catch(SQLException e)
{
};
consulta="insert into
estado_estudiante(cod_estudiante,sesion_actual,Numero_pregunta,cod_grupo)
values("+\""+usb+\"\", "+" 1,0, "+\""+grupo+\""+");";//esta sentencia me registra en
la tabla estado_estudiante el ingreso del nuevo estudiante
try
{
int n= instruccion.executeUpdate(consulta);//instruccion para insertar registros a la
bd
} //fin try no usar ; al final de dos o mas catches
catch(SQLException e)
{out.println(e);
};
contador++;
}
else//si contador es mayor que 0 quiere decir que el usuario ya existe
{
//si ya existe debe validar el ingreso, si puede o no ingresar al nivel que el
estudiante eligio
consulta="select estudiante.*,estado_estudiante.* from
estudiante,estado_estudiante where
estudiante.cod_estudiante=estado_estudiante.cod_estudiante and
estudiante.cod_estudiante like"+\""+usb+\""+\" and
estado_estudiante.sesion_actual<4 and
estado_estudiante.cod_grupo !="+\""+grupo+\""+";";//sentencia que busca los
registros del usuario para saber en que nivel esta y en que sesion
try
{
tabla = instruccion.executeQuery(consulta);//instruccion para consultar registros de
la bd
// ciclo de lectura del resultset
while(tabla.next()) //recoger matriz de resultados
{

```

```

contador3++;//variable que funciona como centinela
sesion=tabla.getInt(4);//asignacion a la variable sesion la columna 4 de la matriz
de resultados(resultados de la consulta)
grupo=tabla.getString(6);///asignacion a la variable grupo la columna 6 de la
matriz de resultados(resultados de la consulta)
}; // fin while
} //fin try no usar ; al final de dos o mas catches
catch(SQLException e)
{out.println(e);
};
if(contador3>0)
{
contador=0;//si en contador3 es mayor a 0 quiere decir que el usuario ya esta en
otro nivel y no lo ha completado
}
else//si contador3 es igual a cero quiere decir que el usuario ya esta registrado
{
//verificar que no tenga otro grupo sin terminar
consulta="select estudiante.*,estado_estudiante.* from
estudiante,estado_estudiante where
estudiante.cod_estudiante=estado_estudiante.cod_estudiante and
estudiante.cod_estudiante like"+"\""+usb+"\""+ " and estado_estudiante.cod_grupo
like"+"\""+grupo+"\""+";";
try
{
tabla = instruccion.executeQuery(consulta);//instruccion para consultar registros de
la bd
// ciclo de lectura del resultset
while(tabla.next()) //recoger matriz de resultados
{
contador2++;
}; // fin while
} //fin try no usar ; al final de dos o mas catches
catch(SQLException e)
{out.println(e);
};
if(contador2==0)
{//hace un nuevo registro del estudiante en la tabla estado_estudiante, porque ya
ha //terminado un grupo completamente
consulta="insert into
estado_estudiante(cod_estudiante,sesion_actual,Numero_pregunta,cod_grupo)

```

```
values("+\""+usb+"\","+\"1,0,\""+\""+grupo+"\")+";">//esta sentencia me incializa el
estado del nuevo estudiante
try
{
int n= instruccion.executeUpdate(consulta);//instruccion para insertar registros a la
bd
} //fin try no usar ; al final de dos o mas catches
catch(SQLException e)
{
};contador++;
}
}
}
if(contador>0)
{//si el contador es mayor que 0 se construye el mapa de navegacion en html
```