

# Universidad de El Salvador

FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE

DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



## TRABAJO DE GRADUACIÓN

**“SISTEMA INFORMÁTICO PARA LA UNIDAD DE HARDWARE Y SOFTWARE DEL DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA DE LA UNIVERSIDAD DE EL SALVADOR – FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE”**

**PARA OPTAR AL GRADO DE:  
INGENIERO DE SISTEMAS INFORMÁTICOS**

**PRESENTAN:**

**Amaya Centeno, Rodrigo  
Barrientos Padilla, Hugo Alejandro  
Linares Melara, Roberto Carlos**

**DOCENTE DIRECTOR:  
Ing. Carlos Stanley Linares Paula**

Santa Ana

El Salvador  
Septiembre 2009

Centro América

**UNIVERSIDAD DE EL SALVADOR**

**RECTOR**

**ING. Y MSC. RUFINO ANTONIO QUEZADA SÁNCHEZ**

**VICE-RECTOR ACADÉMICO**

**ARQ- Y MASTER MIGUEL ÁNGEL PÉREZ RAMOS**

**VICE-RECTOR ADMINISTRATIVO**

**LICDO. Y MASTER OSCAR NOE NAVARRETE**

**SECRETARIO GENERAL**

**LICDO. DOUGLAS VLADIMIR ALFARO CHAVEZ**

**FISCAL GENERAL**

**DR. RENE MADECADEL PERLA JIMÉNEZ**

**FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE**

**DECANO**

**LICDO. JORGE MAURICIO RIVERA**

**VICE-DECANO**

**LICDO. Y MASTER ELADIO EFRAÍN ZACARÍAS ORTEZ**

**SECRETARIO DE FACULTAD**

**LICDO. VICTOR HUGO MERINO QUEZADA**

**JEFE DE DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**

**ING. RAUL ERNESTO MARTÍNEZ BERMÚDEZ**

## Agradecimientos

A Dios, por amarme, acompañarme y moldearme; así como a mi mamá, y a mis abuelos, por su inagotable paciencia y cariño, desde mi infancia hasta el día de hoy.  
A ustedes cuatro, infinitas gracias.

-- Rodrigo

A Dios, por escucharme cuando más lo necesito; a mi madre y mi padre por su amor para educarme y su apoyo incondicional en todas las facetas de mi vida; a mi hermana por soportarme y brindarme ese cariño que únicamente ella me lo puede otorgar; a mis abuelas, que con sus enseñanzas me ayudaron a forjar el hombre que soy ahora. Sin Uds. esto no sería posible. Gracias.

-- Hugo

A Dios primeramente, por haberme dado la vida, la capacidad intelectual y las oportunidades de superación que permitieron que hoy haya llegado hasta donde me encuentro y la confianza de que en el futuro continuaré ganándome sus bendiciones.

A mi madre quien siempre apoyó mis sueños y aspiraciones en la vida y me inculcó valores morales haciendo de mí lo que soy ahora. Espero haberme convertido y continuar siendo en el hombre que ella esperaba de mí, enorgullecíndola con cada una de mis acciones.

-- Roberto

## ÍNDICE

### Resumen Ejecutivo

CAPÍTULO I: Generalidades del Proyecto .....	8
Introducción .....	8
Planteamiento del Problema .....	9
Justificación .....	12
Objetivos .....	14
Objetivo General .....	14
Objetivos Específicos .....	14
Alcances.....	16
Limitaciones.....	20
Metodología de Investigación .....	20
CAPÍTULO II: Resultados de la Investigación .....	23
2.1 Metodología de investigación.....	23
2.1.1 Fuentes de información primaria .....	23
2.1.1.1 Opinión de expertos.....	23
2.1.1.2 Documentación Bibliográfica .....	27
2.2 Análisis de los Resultados .....	28
CAPÍTULO III: Análisis de Requerimientos del Proyecto .....	37
Requerimientos de desarrollo .....	37
Requerimientos de Producción .....	39
Estudio de Factibilidad .....	40
CAPÍTULO IV: Diseño y Desarrollo del sistema.....	47
Modelado y Diseño de la Base de Datos .....	47

Diseño y Desarrollo de la Capa de Acceso a Datos .....	56
Diagrama de Clases .....	59
Diseño de la Capa de Negocios .....	70
Desarrollo de módulo de inventarios (JInvent) .....	74
Desarrollo de módulo de Peticiones de servicios (JRequest) .....	77
Desarrollo de módulo de Equipo multimedia (JCanon) .....	78
Desarrollo de módulo de seguridad (JHardmin) .....	80
Desarrollo del módulo de Administración de grupos de laboratorio (ManLab).....	82
Desarrollo del módulo de Manejo de Contenidos (JWiki y JProCur) .....	84
Diseño de la Interfaz de Usuario general .....	85
CAPÍTULO V: Plan de Implementación del Software .....	95
Objetivos del plan de implementación .....	95
Listado de actividades del plan de implementación.....	96
Costo de Implementación .....	96
Pruebas Unitarias.....	97
CAPITULO VI: Documentación del Software: .....	103
Descripción de pantallas .....	103
Configuración de la plataforma del sistema.....	114
Diccionario de datos.....	121
Conclusiones.....	166
Recomendaciones.....	169
Anexos.....	171
Bibliografía .....	174
Índice de gráficos, figuras y tablas .....	175

## Resumen Ejecutivo

El presente trabajo aborda el tratamiento de los problemas encontrados en una investigación hecha dentro del Departamento de Ingeniería y Arquitectura de la Facultad Multidisciplinaria de Occidente de la UES-FMO. Dicha investigación trata de los problemas que acarrea la Unidad de Hardware y Software del mencionado departamento que, entre otras cosas, lleva a cabo las solicitudes de soporte técnico y realiza los mantenimientos correctivos y preventivos de toda la Facultad. Por lo tanto, en esta tesis se podrá conocer la propuesta de solución para la problemática que la investigación arrojó: un sistema informático que ataque las falencias de la Unidad, y que la conviertan en un lugar de más eficiencia con reducción de tiempos y poder llegar a ayudar a más sectores de la comunidad universitaria.

Se puede conocer la investigación realizada, desde el punto de vista que la población beneficiada será el estudiantado de ingeniería y arquitectura, que ronda alrededor de los 1300 estudiantes, el personal docente de dicho Departamento y el personal administrativo de toda la Facultad. Se tomaron muestras acordes a los universos de las personas beneficiadas.

También conocerán la solución propuesta, llamada JHard, que busca mejorar las áreas más débiles de la Unidad con el fin que se convierta en una herramienta de consulta para todos los involucrados dentro de la Facultad. Conocerán además las herramientas y tecnologías utilizadas en la elaboración de JHard, herramientas diversas como Subversion, Tortoise, Google Code, Google Groups, Mysql Administrator y Query Browser, Netbeans, entre otras y las tecnologías como Java, Java Server Faces, ICEfaces, MySQL, AJAX y TopLink de Oracle.

## CAPÍTULO I: Generalidades del Proyecto

### Introducción

La universidad de El Salvador, se constituye una de las instituciones educativas más grandes del país; eso la convierte en un ente muy difícil de adaptarse a los cambios, mayormente, si dichos cambios son de carácter estructural. Esto es una verdad, desde el punto de vista teórico, pues las sociedades tienden a cambiar antes que las instituciones - a nivel macro: estructura y superestructura-; dicho fenómeno se reproduce a nivel micro en las instituciones así vemos en nuestro medio, instituciones anquilosadas, inadaptadas a los tiempos modernos, instituciones que parece que se quedaron atrapadas en un momento dado de la historia.

Nuestra universidad no es ajena a este fenómeno, y es común que los cambios lleguen con demora. Hemos sido testigos de diversos avances tecnológicos, pero fuera de la universidad, aunque existe un componente económico, muchos avances no dependen de esa condición, máxime dentro de una institución de educación superior como la Universidad de El Salvador, donde el recurso más importante: es el recurso humano.

La naturaleza de la estructura organizativa dentro de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador hace que los laboratorios de prácticas de cada Departamento se manejen con administraciones ajenas. En la Unidad de Hardware y Software del Departamento de Ingeniería y Arquitectura, que constituye como el objeto de estudio e implementación a lo largo de este trabajo de grado; el Laboratorio de Prácticas actualmente le da servicio a diferentes carreras o departamentos de la FMO, generando una serie de inconvenientes a alumnos, docentes y personal administrativo, que se traduce en perdida de horas clase para alumnos y docentes, los cuales se acentúan en momentos claves para el quehacer académico.



Gráfico 1.1: los encuestados son preguntados sobre cuál es el mayor problema de la inscripción de grupos de laboratorios en el Laboratorio de Cómputo

Lo anterior hace que los objetivos que se persiguen en la consecución de este proyecto: la automatización de los procesos comunes y que se fomente una mayor eficiencia dentro de la Unidad de Hardware, se hagan con la ayuda de herramientas tecnológicas para crear un sistema informático robusto y extensible.

### Planteamiento del Problema.

Como en la Unidad de Hardware, aun hay labores que no se llevan a cabo con la ayuda de un sistema informático, tales como el inventario físico, la inscripción de grupos de laboratorios y las peticiones de soporte técnico. Otros procesos como los de la inscripción y creación de grupos de laboratorios se hacen de maneras tradicionales, que son lentas y en algunos casos tediosos.

La eficiencia de la Unidad está condicionada al recurso humano existente, que adicional a los puestos de trabajo designados, se le suman estudiantes de Ingeniería de Sistemas Informáticos que realizan su servicio social. En el año hay momentos en los que el trabajo de soporte técnico y órdenes de servicio sobrepasa la capacidad de dicho recurso<sup>1</sup>. El tiempo se pierde considerablemente cuando se da prioridad equivocadamente a problemas sencillos, que los mismos usuarios finales, llámense docentes, administrativos, invitados, estudiantes, etc. podrían resolver por ellos mismos. Sin embargo la falta de conocimientos fundamentales de informática detienen a estas personas a resolver sus propios problemas y a dejar más carga a la Unidad de Hardware y Software. Buscando también fomentar la labor social, se ha tenido a bien pensar dentro de la Unidad, que sería de utilidad brindar orientación Informática a los educadores con especialidad en las ramas humanísticas.

El manejo del inventario físico de hardware es propenso a fallas, poco práctico y no es automatizado. Para controlar la salida y entrada de todo el hardware, ya sea en uso o desuso, se debe de contar con herramientas orientadas a esta labor.

Cada computadora al que se le brinda mantenimiento, ya sea correctivo o preventivo, no posee su respectivo control de mantenimiento, lista de fallas recurrentes y reportes de soluciones, es decir, un historial de la misma. Se apela a la memoria de los técnicos o los que dan el soporte técnico para recordar cuales han sido los fallos de las computadoras o hardware afectado. Se busca que se tenga un registro de las reparaciones realizadas para no perder aun más tiempo buscando síntomas y obteniendo diagnósticos inútiles una y otra vez.

La realización de un software a la medida de las necesidades de la Unidad es una solución propicia, pero debido que a la naturaleza de la misma (Unidad) y de la diversidad de tareas que se llevan a cabo sería muy difícil encontrar un software en el mercado que abarque de manera fidedigna todas las operaciones de esta. Y también hay que agregar los factores

---

<sup>1</sup> Experiencia propia de estudiantes en servicio social dentro de la Unidad de Hardware y Software y de uno de los autores del presente Trabajo de Grado

económicos de la Universidad, para la cual se le imposibilita por lo apretado del presupuesto un gasto en software para control, administración y manejo de recursos y actividades, que es lo que se plantea en este Trabajo de Grado.

**Por lo tanto, el problema encontrado dentro de la Unidad de Hardware y Software del Departamento de Ingeniería y Arquitectura es el manejo convencional de sus labores que le resta eficiencia a su quehacer diario.**

El sistema que se propone para este trabajo de grado, al que se le llamará: JHard, será el compilado de una serie de módulos, cada uno de los cuales desempeñará una de las funciones de la Unidad de Hardware y Software; por lo tanto está planteado para que sea un Sistema de Información, Control y Administración de recursos y actividades específicas.

Los módulos propuestos son los siguientes:

- Módulo de inventarios para controlar el hardware y mobiliario a cargo del laboratorio de Hardware.
- Módulo de inscripción de grupos de laboratorio, así como el mantenimiento de los mismos.
- Módulo para promocionar cursos de informática para personal docente y administrativo.
- Módulo de peticiones de servicio de mantenimiento de hardware y software dentro de la Facultad.
- Módulo de conocimiento sobre soluciones a problemas informáticos comunes y biblioteca de manuales y software libre.
- Módulo de reserva de cañones y laptops del Departamento de Ingeniería.

Otras funciones del sistema, siempre enmarcadas dentro de los módulos antes mencionados se encuentran:

- Control de asistencia a los instructores mediante el acceso al sistema al impartir una práctica/laboratorio.
- De manera similar que el literal anterior, se llevará un control de los estudiantes con sesión iniciada a cada laboratorio.
- Se propone una Arquitectura de sistema Cliente-Servidor. Con esta aproximación, el uso del sistema será indiferente al hardware, software y sistema operativo que se tengan en años venideros dentro del Laboratorio de Hardware LABCOM-1. Cabe mencionar que lo único que se necesita del lado del cliente o usuario final, es un navegador Web, instalado en cualquiera de los sistemas operativos modernos.
- Además, los módulos serán diseñados, pensando en extensibilidad y escalabilidad, sin dejar de lado la usabilidad para los usuarios finales y los administradores del mismo.

## Justificación

Las nuevas tecnologías informáticas se están aplicando en todos los rubros de la vida cotidiana. Desde la educación hasta los deportes. Desde las finanzas hasta la investigación de operaciones y la religión. Así de distantes los temas que se mencionan, así de unidos se encuentran por el uso de las tecnologías de la información. No hay una tan sola actividad que no haya contemplado adquirir entre su inventario físico una estación de trabajo con software especializado para su dinámica particular.

Se plantea que la computadora constituye un amplificador intelectual de todas las personas, una herramienta utilísima y elemental del actual mundo globalizado en donde las distancias se hacen cada vez más cortas y menos tensas. Es un hombro articulado en el

cual se pueden descansar grandes tareas y procesos largos y tediosos para el desarrollo humano simple y que ayudan a una tan sola entidad: La obtención de información.

Pero se dirá: "La información es de todos". Es claro, la información está al alcance de cualquiera. Es de todos y para todos. Lo que es innegable es la calidad de la misma. Esto genera un punto de inflexión entre un trabajo común y una actividad sumamente productiva. En resumidas cuentas es lo que se quiere lograr con este sistema para la Unidad de Hardware y Software: aumentar la eficiencia de las actividades en la que nos enfocamos en este trabajo.

En la era de las comunicaciones; una dependencia donde se enseña a manejar tecnologías y ponerlas al servicio de la sociedad debe contar con una poderosa herramienta diseñada a la medida de este cuerpo para que mecanicen sus procesos principales y tediosos, haciendo ganar a toda la Facultad, que se ve beneficiada por la labor que se realiza en la Unidad de Hardware.

Es por eso que se hace de suma necesidad esta eficaz y potente herramienta compuesta por módulos consumibles que aumentará la eficiencia de la Unidad de Hardware y por ende de toda la Facultad, servirá como ejemplo para las nuevas generaciones de estudiantes de ingeniería de sistemas y también cumplirá una función social al poner al servicio de todo el personal de la UES-FMO guías y pequeños tutoriales para el manejo de herramientas y de errores de informática comunes. Además de ir a la vanguardia y proyectar una mejor imagen brindando servicios eficientes de información, de forma interna a la facultad.

## Objetivos

### Objetivo General

- Diseñar, desarrollar, e implementar un sistema informático, que automatice los procesos de las principales funciones de la Unidad de Hardware y Software del Departamento de Ingeniería y Arquitectura, las cuales son: soporte técnico a toda la Facultad, inventario de hardware, mantenimiento del centro de cómputo, inscripción de grupos de laboratorios y cursos de formación en informática para personal docente y administrativo, buscando que esta sea vuelva más eficiente.

### Objetivos Específicos

- Crear un módulo que controle de una manera más eficaz y exacta todo el inventario físico de hardware y de mobiliario que maneja el Laboratorio de Hardware.
- Desarrollar un módulo que realice una inscripción de grupos de laboratorio de manera más moderna, permitiendo una eficiente administración del cupo de éstos.
- Diseñar un sistema modular, flexible y seguro, orientado a un ambiente web, cuya área de acción será la Intranet de la Facultad, para aprovechar este recurso ampliamente disponible en la Facultad.

- Legar una herramienta bien diseñada y funcional que fomente el uso de las soluciones informáticas y la computación general para docentes y personal administrativo de la Universidad.
- Promocionar, mediante el sistema que se desarrollará, la imagen de la Unidad de Hardware y su quehacer general, las labores educativas, de mantenimiento y social que desarrolla, además de los cursos de informática que se pueden impartir en ella.

## Alcances

El sistema JHard, está compuesto por siete subsistemas o módulos robustos a los que tienen acceso los usuarios de todo tipo, sin embargo está formado por otra gran cantidad de subcomponentes que añaden características adicionales, necesarias para el funcionamiento interno del mismo. A continuación, una lista de los 7 módulos principales con los que tendrá interacción los usuarios finales, con las necesidades que se pretenden solventar:

1. **JInvent:** Manejo de Inventario (no consumible) del Laboratorio de Hardware.
2. **JRequest:** Solicitud de servicio al Laboratorio de Hardware, para mantenimiento y reparaciones en la FMO.
3. **JWiki:** Módulo de conocimiento, con soluciones a problemas comunes, y vínculos a utilerías o descarga de software.
4. **JProCur:** Módulo de Manejo de Contenidos (CMS) para la promoción de cursos del Laboratorio de Hardware
5. **ManLab:** Módulo de para la gestión de inscripción de laboratorios prácticos en el Laboratorio de Hardware.
6. **JCanon:** Módulo para la gestión de reserva de cañones y/o laptops.

7. **JHardmin:** Módulo de mantenimiento y seguridad para JHard, con el que se podrá administrar los perfiles/roles de los usuarios y a los mismos.

- JHard también generará los siguientes reportes:

Reportes básicos:

- Asistencia de alumnos a laboratorios
  - Asistencia de instructores a sus laboratorios
  - Reportes de inventario de piezas y accesorios sin equipos asignados
  - Órdenes de Servicio por PC
  - Listado de piezas y accesorios sin existencia asignada
  - Reserva de Cañones (Horarios) con filtros: por docente, fecha.
  - Alumnos por materia que están inscritos en grupos de laboratorio
- 
- La seguridad del sistema estará dada mediante el uso de perfiles predefinidos asignables a los usuarios sobre los módulos y las acciones sobre estos. Los perfiles de usuario que con los que se cuentan:
    - Súper Administrador
    - Administrador de Sistema
    - Administrativo (Secretaria)

- Docente
- Editores de Contenidos
- Estudiante
- Instructor

Y cualquiera que acceda desde la Intranet de la Facultad, será tratado como Público General.

- Se seguirán los lineamientos que el encargado de la Sala de Servidores plantee, para seguir al máximo los requerimientos que este especifique y procurar su futura integración en los servidores de la facultad. Sin embargo, el sistema no se instalará en los servidores de la Facultad, sino que en una computadora dedicada a este en el mismo Laboratorio de Hardware. El sistema será instalado y se utilizarán pruebas locales para asegurar su adecuado funcionamiento, pero el sistema no será probado en un escenario real (inicio de ciclo).
- Por motivos de seguridad institucional JHard no se integrará con ADACAD, y será implementada bajo el criterio y responsabilidad de los administradores finales o el encargado de los servidores de la facultad.
- Para notificaciones, JHard se limitará a usar el mismo sistema de promoción de cursos. JHard no usara notificaciones por el correo institucional (hay un problema con un disco duro de los servidores del correo institucional, y a esto añadimos la demanda extra sobre el ancho de banda de salida, si se permitieran notificaciones a correos no institucionales), pero se implementarán notificaciones mediante suscripción a Atom/RSS Feeds.

- JHard estará disponible únicamente para la Intranet de la Facultad. La configuración de DNS e integración con los servidores de la Facultad Multidisciplinaria de Occidente, se dejara a discreción de los administradores finales o del encargado de la Sala de Servidores.
- El Hardware en el que se ejecutara JHard estará sujeto a lo que sea capaz de proporcionar el Departamento de Ingeniería o la Unidad de Hardware y Software y según los requerimientos operativos mencionados en el Capítulo 3 de este trabajo. Así mismo el mantenimiento del mismo Hardware estará sujeto a los repuestos de los que disponga la Unidad de Hardware y Software y que el Departamento de Ingeniería proporcione para este fin.
- No se realizaran capacitaciones del uso de JHard, porque este estará diseñado de tal forma que su uso sea sencillo y amigable. Tampoco se entregara manual de usuario, como reemplazo de este se realizaran videos demostrativos de las principales operaciones realizables en cada modulo integrado al Sitio.
- Se colocara únicamente contenido de ejemplo en el Sistema.
- Se limitará el tamaño de archivos almacenados en JHard para que no colapse en un futuro.
- Se entregará documentación de código fuente, y scripts necesarios para el correcto funcionamiento del sistema.

## Limitaciones

- Estrictas peticiones que hace el administrador de los servidores de la UES-FMO que impide una implementación más sencilla en éstos.
- Se carece del hardware adecuado o apropiado para la implementación. Hardware improvisado será el que albergará al sistema.
- Existe un problema con los discos duros del correo institucional, y por motivos del uso de ancho de banda, no se implementaran notificaciones por correo, pero sí se puede implementar un sistema de suscripción por RSS/Atom feeds.

## Metodología de Investigación

El proyecto se ejecutará haciendo uso de los lineamientos establecidos por la **investigación de tipo operativa**, ya que además de diagnosticar la situación problemática se hará una propuesta concreta para solucionar dicha situación. El motivo por el cual se realiza el presente proyecto es para solucionar los problemas existentes en la Unidad de Hardware y Software de la Universidad de El Salvador – Facultad Multidisciplinaria de Occidente y darles de esta forma una solución óptima.

Según el análisis realizado en la institución se tomó la decisión de resolver los problemas utilizando las siguientes fases:

### **FASE 1: RECOPILACIÓN DE INFORMACIÓN PERTINENTE**

Primero se investigará la situación actual para identificar los problemas existentes, tomando en cuenta todas las variables que intervienen en ellos.

Luego se procederá a la recopilación de información que nos brinden personas cercanas a la Unidad de Hardware y Software que se pretende beneficiar, por medio de entrevistas, encuestas, etc. con el fin que provean de una teoría que sustente el diseño del software propuesto.

### **FASE 2: DISEÑO DEL SOFTWARE A PROPONER**

Luego de contar con suficientes bases teóricas y de seleccionar el lenguaje de programación adecuado, así como el sistema gestor de base de datos, se procederá a diseñar el software planteado. Este deberá reunir ciertas características necesarias para solucionar la problemática encontrada.

### **FASE 3: DESARROLLO DEL SOFTWARE PROPUESTO**

En esta fase se realizará todo el desarrollo y la programación de cada uno de los módulos de JHard.

### **TÉCNICAS DE INVESTIGACIÓN**

Entre las técnicas a utilizar para la recolección de datos, se pueden mencionar las siguientes:

**Entrevista:** Con los encargados y responsables de la Unidad de Hardware y Software, el Lic. José Antonio Madrid y el Ing. Luis Alonso Barrera. Con esto se conocerán las falencias de la Unidad y las cuales se pueden suplir con la inserción de un sistema informático en la unidad

**Revisión Documental:** Este consiste en consultas a libros, tesis y cualquier otro material que aporte información útil para el tema a desarrollar.

**Consultas de la Web:** Se visitarán diferentes sitios en Internet, que puedan proporcionar información importante para el desarrollo del proyecto.

**Encuestas:** por medio de estas se tendrá un panorama real de lo que el público en general (estudiantes, docentes, administrativos, etc.) conoce y opina acerca de la Unidad de Hardware y Software.

## CAPÍTULO II: Resultados de la Investigación

### 2.1 Metodología de investigación.

Para viabilizar el presente trabajo, se optó por consultar cuatro niveles de fuentes de información primaria:

- 1- Opinión especializada de expertos.
- 2- Encuesta a estudiantes.
- 3- Encuesta a docentes.
- 4- Encuesta a Personal Administrativo.

#### 2.1.1 Fuentes de información primaria

##### 2.1.1.1 Opinión de expertos.

Para la obtención de la información conducente a apoyar la implementación de este proyecto, consideramos, muy importante la opinión de dos jefes relacionados con la dependencia objeto de estudio- convirtiéndose esta fuente en piedra angular del presente trabajo-. Se hicieron tres preguntas concretas, con las que se esperaba una respuesta de la misma índole:

**1- Haga una radiografía de la actualidad de la Unidad de Hardware tomando en cuenta parámetros como manejo de inventario, manejo de Grupos de Laboratorios en el LABCOM-1, soporte técnico y de los cursos informáticos:**

**2- ¿Cómo espera que modifique la actualidad de la Unidad con el sistema informático?**

**3- ¿Cree que es posible educar en computación e informática al personal administrativo y docente con la ayuda de una herramienta en línea dispuesta para ello?**

Con la segunda fuente de información, en primer lugar se definió el tamaño de la muestra.- Esta se obtiene por medio de la siguiente ecuación:

n = tamaño de la muestra para poblaciones finitas:

$$n = Z_{\alpha}^2 \frac{N \cdot p \cdot q}{i^2(N-1) + Z_{\alpha}^2 \cdot p \cdot q}$$

<b>Tamaño muestral</b>	z 1,96 (a=0,05) 2,58 (a=0,01)
47	1,96
<b>p (frecuencia esperada del parámetro)</b>	p (frecuencia esperada del parámetro)
0,85	0,85
<b>i (error que se prevé cometer)</b>	i (error que se prevé cometer)
0,1	0,1
<b>Población</b>	Población
1000	1000

Tabla 2.1: Fórmula y tamaño muestral que se utilizó para realizar el estudio de campo

A los estudiantes se les formularon las preguntas siguientes:

- 1. ¿Qué problemas identifica al momento de inscribir grupos de laboratorios al inicio de ciclo?**
- 2. ¿Cuántas horas clase cree que pueden ahorrarse si se solventaran las problemáticas anteriormente seleccionadas?**
- 3. ¿Cree que se enriquecería su proceso de aprendizaje con la posibilidad que pueda consultar una base de datos de conocimientos acerca de manejo de Hardware y Software?**
- 4. ¿Estaría interesado en participar en cursos de manejo y soporte técnico de hardware y software dentro de la Unidad de Hardware y Software?**

Tabla 2.2: Preguntas de las encuestas a estudiantes

Como el lector ya habrá notado, las preguntas son bastante concretas, por la misma naturaleza del problema. Fueron cuarenta y siete (47) las boletas que se pasaron, sin más condición, que la que los encuestados, fueran estudiantes regulares. Una tercera fuente de información fueron docentes, quienes deberían tener la condición de docentes inmersos en la problemática objeto de estudio, a estos docentes se formularon las siguientes interrogantes:

- 1. ¿Cómo forma los grupos de laboratorios de prácticas?**
- 2. ¿Cuántas horas clase dedica a formar grupos de laboratorios?**
- 3. ¿Reserva Ud. equipo audiovisual (cañones y laptops) en el Depto. de Ingeniería?**
- 3.1 Si su respuesta es sí, de los siguientes, ¿cuál es el mayor problema al realizar esta actividad?**
- 4. ¿Cree que se le facilitaría la reserva de material audiovisual con un sistema web diseñado para ello?**

Tabla 2.3: Preguntas de encuestas a docentes

En última instancia se recabo información en el sector administrativo de la Facultad; también ellos serán usuarios finales al igual que los estudiantes, del laboratorio hardware:

**1- De los siguientes, ¿cuáles son los problemas informáticos más frecuentes?**

- a) Virus
- b) Correo electrónico
- c) Instalación de nuevos programas
- d) Microsoft Office y otro software
- e) Configuración de impresoras
- f) Manejo de impresoras
- g) Conexión a Internet
- h) Otros:

**2- ¿Cree Ud. que sería capaz de resolver los problemas informáticos más comunes en los que se ve involucrado, apoyándose con un sistema tutor en línea?**

**3- ¿Estaría dispuesto a someterse a capacitaciones en el área de la informática como Soporte de Hardware?**

**4- ¿Cree que con la implementación de un sistema informático en línea (web) se podrían solucionar de manera más eficaz los problemas más comunes con respecto a equipo informático?**

Tabla 2.4: Preguntas a personal administrativo

Consideramos que con estas fuentes y la manera en que está diseñada la recolección de información, tenemos insumos suficientes y un panorama completo para hacer un diagnóstico y presentar una solución integral, al problema en cuestión.

### **2.1.1.2 Documentación Bibliográfica.**

Umberto Eco, "Cómo se Hace un Tesis – Técnicas y Procedimientos de estudio, investigación y escritura", Roma, Italia.

Raúl Rojas Soriano, "Investigación Social", San Salvador, El Salvador.

Calderón Peraza, Ernesto Alexander. "Desarrollo de un sistema informático para la actualización del Centro de Desarrollo Profesional Docente del Ministerio de Educación de El Salvador, Departamento de Ingeniería y Arquitectura, Universidad de El Salvador". Santa Ana, El Salvador, 2005.

## 2.2 Análisis de los Resultados

Cuando se inicia la exploración de un problema, ya en la práctica ésta se ve enriquecida por la diversidad que de ella emana, en este caso dicha diversidad se manifiesta en distintas líneas de acción que en muchos casos nacen en la praxis; líneas de acción que son distintas a las que el estudio había predeterminado, en este caso suponíamos que la perdida de horas clase – como lo establece los gráficos 2.1 y 2.2 - podría constituirse en un problema importante, lo cual es cierto dados los resultados, sin embargo es un problema que podría solventarse con una carga académica, ya con el ciclo en marcha, pero lo que no suponía el estudio que era tan importante, por ejemplo la falta de horarios para estudiantes trabajadores y este si es un problema igual o mas serio que la perdida de dos horas clase promedio en el inicio de ciclo, por cuanto involucra otro tipo de variables de carácter socioeconómico, para estudiantes trabajadores, en donde como Universidad de El Salvador, estamos obligados a dar respuesta. Es fácil inferir entonces, aunque no se haya estudiado a profundidad, que:

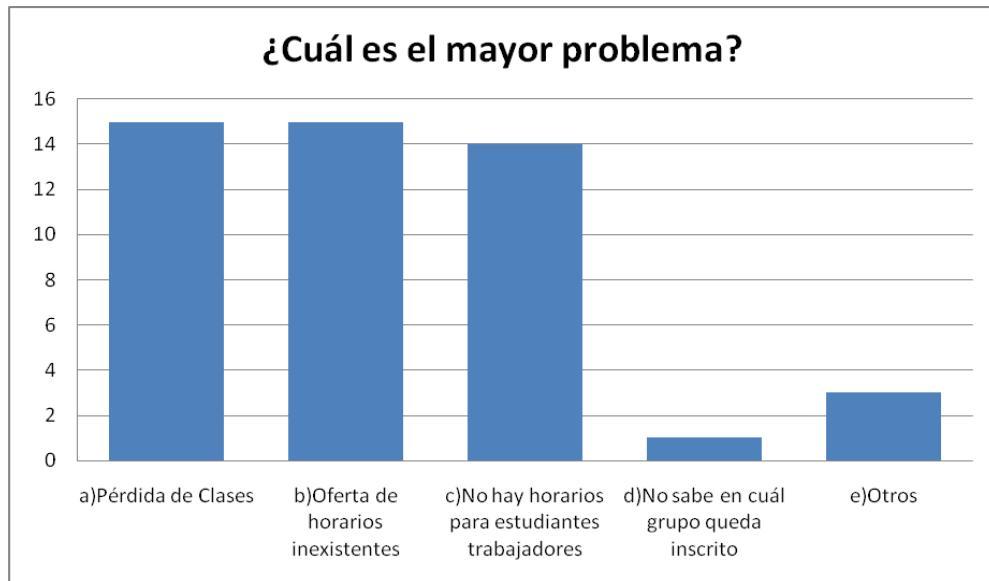


Gráfico 2.1: Cuál es el mayor problema que acarrea inscribir grupos de práctica en el laboratorio de cómputo de Ingeniería y Arquitectura

Probablemente, muchos estudiantes atrasarían su formación, fueran menos productivos en sus respectivos trabajos o podrían perder oportunidades de empleo, ocasionando

probablemente deserción al no salir con metas personales que hayan trazado de antemano, por ejemplo hacer en seis, siete o más años una carrera de cinco. La evidencia del grafico 2.2 es elocuente al respecto:

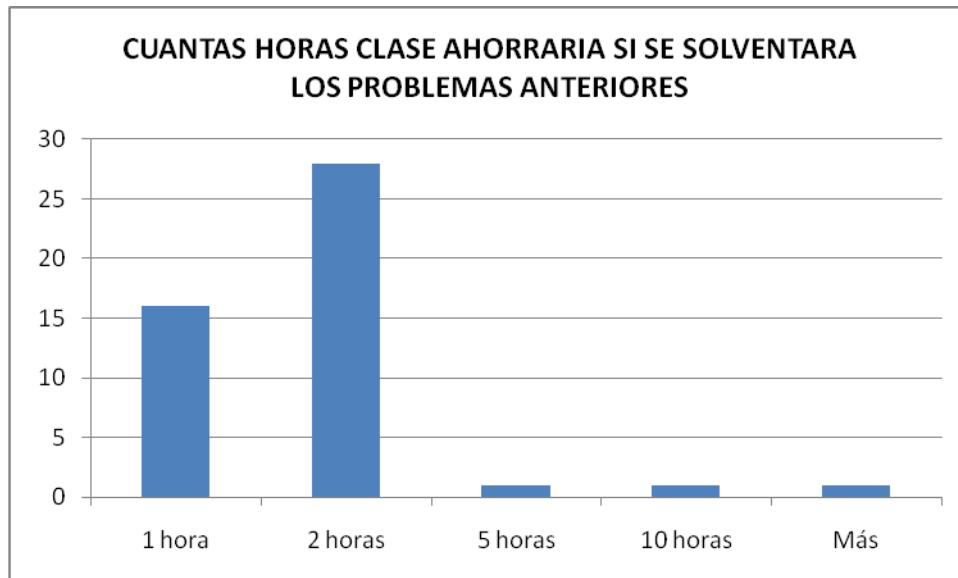


Gráfico 2.2: Cuántas horas clase podrían aprovechar los estudiantes si se solventaran los problemas anteriormente descritos

Los graficos 2.3 y 2.4, demuestran que tan interesados estan los estudiantes en participar en actividades extracurriculares, pero a la vez ligada a la actividad academica



Gráfico 2.3: Se cuestiona si una Wiki de conocimientos informáticos será de utilidad para aumentar el aprendizaje



Gráfico 2.4: Se les pregunta a los estudiantes si estarían interesados en participar en cursos de mantenimiento y reparación de hardware, que es una de las principales actividades que pretende impulsar la Unidad.

## GRÁFICOS DE ENCUESTAS DE DOCENTES

Para los docentes es un proceso muy delicado la inscripción de sus estudiantes en los grupos de prácticas. Muchos de ellos dedican bastantes horas clases y mucho tiempo fuera de las mismas en tratar de organizarlos. El gráfico 2.5 nos indica las horas clase que se dedican a esta actividad:

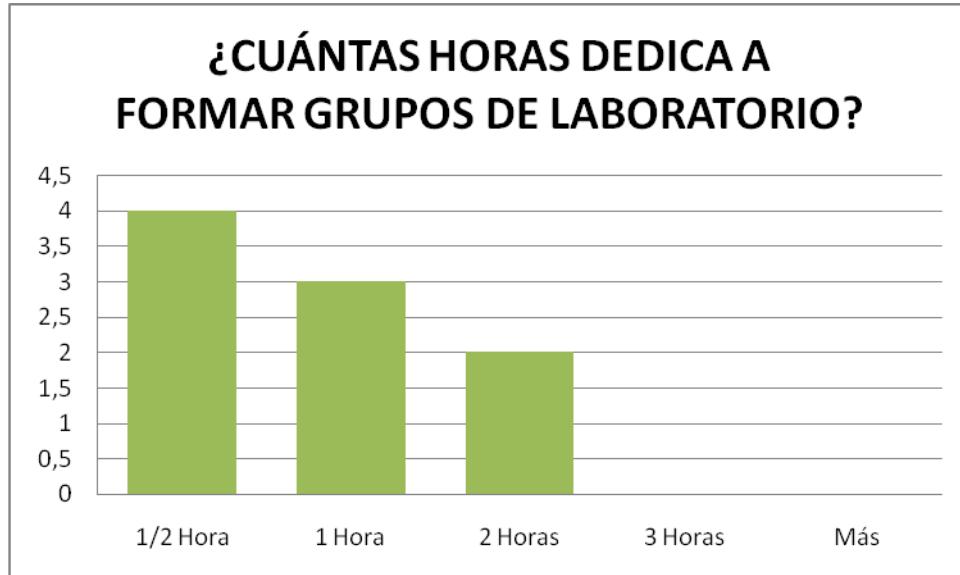


Gráfico 2.5: Cuánto tiempo dedica en sus clases por ciclo a formar grupos de práctica

Es más que obvio el resultado de esta pregunta: los docentes no pueden perder el tiempo que se debería dedicar a elaborar, preparar e impartir de clases.

La organización y sistematización de este proceso solventaría esta problemática de raíz, al no verse nunca más involucrados en este engoroso procedimiento, que no sea para más que solicitar los grupos de prácticas necesarios para el total de sus alumnos.

Todos los docentes reservan equipo audiovisual<sup>2</sup> y todos conocen el proceso para realizarlo. El gráfico 2.6 es claro al respecto:



Gráfico 2.6: Pregunta si el docente utiliza equipo multimedia para sus clases expositivas

Este procedimiento es sumamente áspero y poco amigable, además del bajo control que se hace del equipo, que no pasa de unas cuantas hojas con algunas firmas y la memoria de la persona a cargo del equipo. La falta de control es la causa más achacada por los docentes del Departamento de Ingeniería para que haya un cambio en el actual proceso de reserva de equipo audiovisual. El gráfico 2.7 nos muestra dicha causa:

<sup>2</sup> Entiéndase por **equipo audiovisual** todo tipo de cañones y laptops para clases expositivas

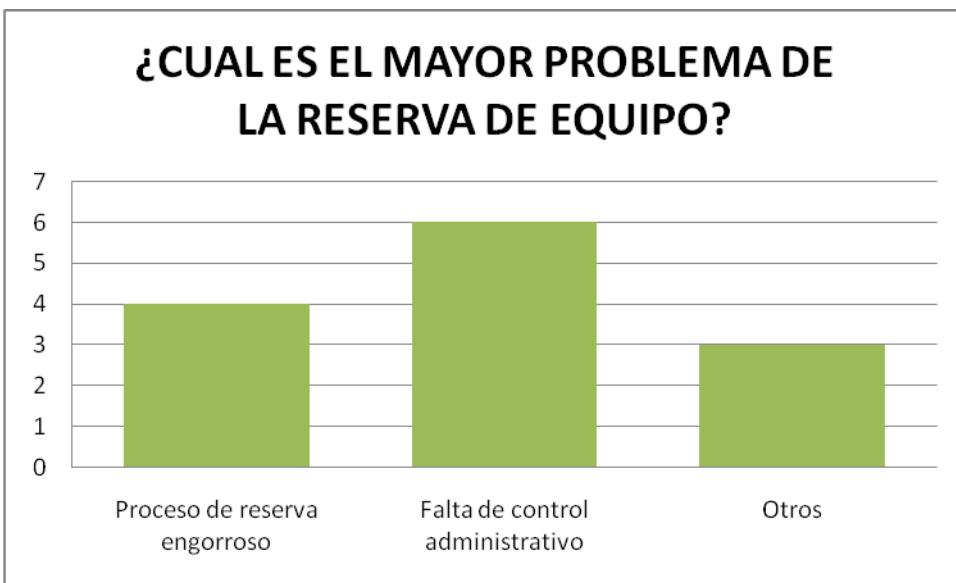


Gráfico 2.7: Cuál es el mayor problema al momento de reservar equipo multimedia en el Departamento de Ingeniería y Arquitectura

Y como en Ingeniería y Arquitectura, las personas están siempre dispuestas al cambio que sea bueno en el ambiente universitario, arrolladoramente, todos los docentes opinan que con la implementación de un sistema web la reserva de equipo sería mucho más sencilla y se tendría un mejor control y un mejor servicio.



Gráfico 2.8: Se le cuestiona a los docentes si con un sistema web se podría mejorar el sistema de reserva de equipo multimedia

## GRÁFICOS DE ENCUESTAS DE ADMINISTRATIVOS

El personal administrativo de la UES-FMOcc es más delicado con respecto a los cambios. Por ser personas en la mayoría poco acostumbradas a la informática y la computación, temen que los cambios sacudan los cimientos de lo que dan por sentado. Sin embargo, y a pesar de esta premisa, se tuvieron respuestas favorables ante la implementación de un sistema tutor en línea para solventar problemas informáticos comunes. Los mismos que ellos dieron a conocer. En la gráfica 2.9 los señalan, siendo los virus informáticos, los ganadores de este apartado, siendo mencionado por cada uno de los encuestados.

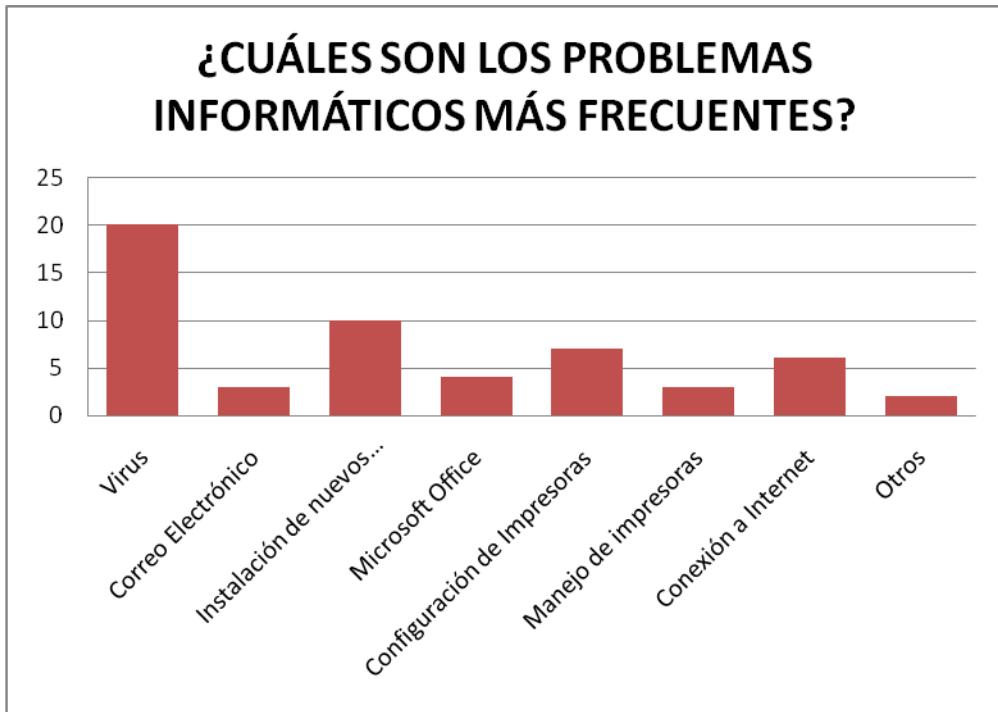


Gráfico 2.9: Cuál es el mayor problema informático que tiene el personal administrativo de la UES-FMO

El sistema tutor en línea vendría a ser una tabla de salvación para los empleados administrativos y todo aquel que recibe soporte técnico por parte de la Unidad de Hardware y Software, al mismo tiempo que aligera la carga de trabajo de dicha dependencia, delegando a los usuarios mismos la resolución de sus problemas más

sencillos, en un círculo de aprendizaje continuo. Todos los encuestados, tal y como lo refleja la gráfica 2.10, creen que el sistema tutor les brindará mucha ayuda. Con algunas capacitaciones extra, el diámetro del círculo de aprendizaje crecería aun más, por lo que las ganancias fueran tremadamente mayores para el administrativo. El gráfico 2.11 refleja el apoyo de los empleados ante tales medidas.

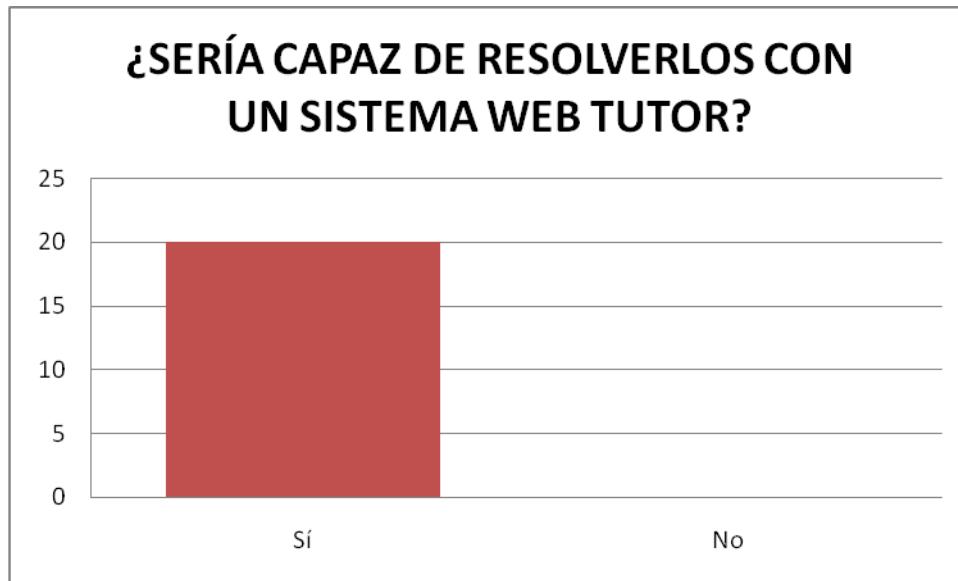


Gráfico 2.10: Podrá resolver los problemas antes mencionados con la ayuda web en línea

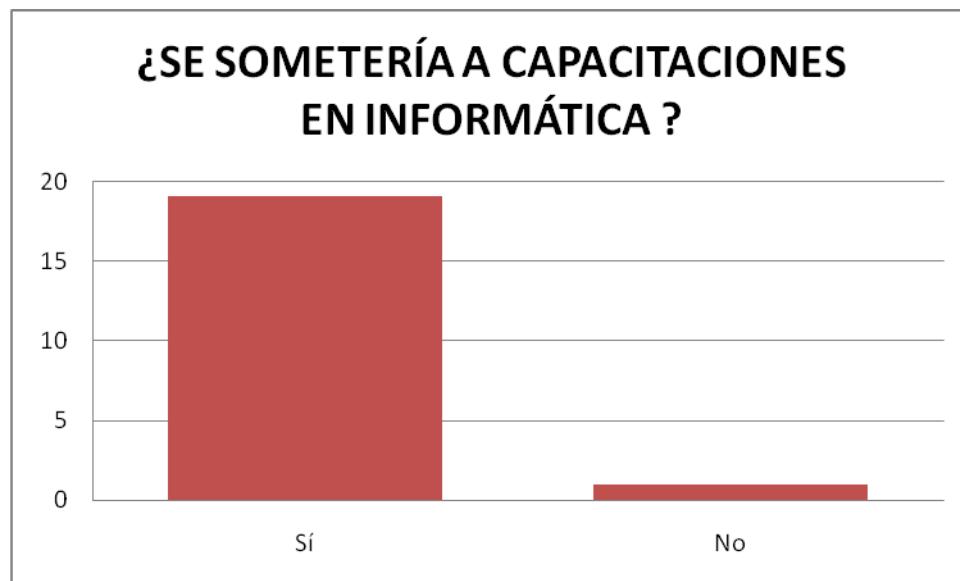


Gráfico 2.11: Está dispuesto a someterse a capacitaciones en informática propias para personal administrativo

El apoyo hacia este trabajo de grado es absoluto, por lo tanto queda debidamente justificado, teniendo el apoyo de todos las personas directamente beneficiadas con resultados arrolladores. La gráfica 2.12 nos muestra que el sistema informático propuesto como trabajo de grado será de gran utilidad y que resolverá muchos problemas actuales en el ámbito informático de la UES-FMOcc.

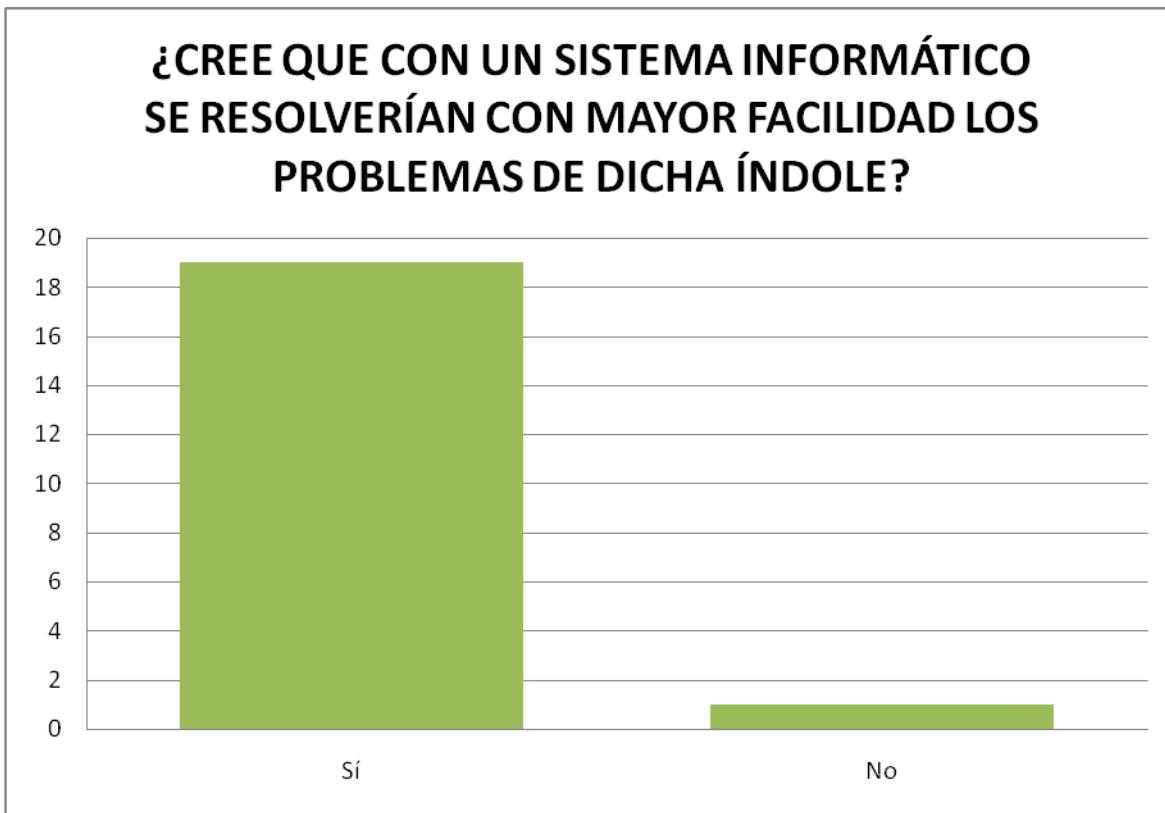


Gráfico 2.12: Los encuestados responden si con un sistema informático se resolvieran más fácilmente problemas informáticos en la UES-FMO

## CAPÍTULO III: Análisis de Requerimientos del Proyecto

### Requerimientos de desarrollo

#### Hardware, Software y Tecnologías de Red

No.	Hardware
1	Desktops Windstorm Core 2.33ghz para desarrolladores (2)
2	Desktop Dell Vostro 220 Mini Tower para Project Manager (1)
3	UPS MINUTE MAN Pro 320 (5)
Software	
4	Apache Tomcat 6.0.18
5	Netbeans 6.5.1
6	MySQL Server 5.1
7	MySQL Administrator
8	MySQL Query Browser
9	MySQL Workbench
10	Java Development Kit
12	Subversion 1.5
13	Sistema Operativo Ubuntu 9.04
14	IceFaces
15	DHTMLXScheduler
Tecnologías de Red	
16	Conexión a Internet

Tabla 3.1: Requerimientos de Desarrollo de JHard

### Requerimientos de Recursos Humanos

ROL	RESPONSABLE	REQUERIMIENTOS
1 <b>Project Manager</b>	<ul style="list-style-type: none"> <li>- Administrar personal del proyecto</li> <li>- Administración de Recursos y tareas</li> <li>- Velar por el cumplimiento del cronograma de actividades y WBS</li> </ul>	<ul style="list-style-type: none"> <li>- Ingeniero de Sistemas Informáticos</li> <li>- Especializado en administración de Proyectos</li> <li>- Experiencia en el área de Software Development</li> </ul>
2 <b>Arquitecto de Sistemas</b>	<ul style="list-style-type: none"> <li>- Partitionar el sistema en módulos flexibles y extensibles.</li> <li>- Establecer los requisitos del sistema</li> <li>- Llevar a cabo análisis costo-beneficio</li> <li>- Análisis de requerimientos</li> </ul>	<ul style="list-style-type: none"> <li>- Ingeniero de Sistemas Informáticos o Técnico en Ingeniería de Sistemas Informáticos.</li> <li>- Especializado en el área de diseño y arquitectura de sistemas.</li> </ul>
3 <b>Analista programador</b>	<ul style="list-style-type: none"> <li>- Diseño de módulos</li> <li>- Diagramar la aplicación con UML</li> </ul>	<ul style="list-style-type: none"> <li>- Ingeniero de Sistemas Informáticos o Técnico en Ingeniería de Sistemas Informáticos.</li> <li>- Especializado en el análisis de sistemas</li> </ul>
4 <b>Programador</b>	<ul style="list-style-type: none"> <li>- Programar módulos del sistema</li> </ul>	<ul style="list-style-type: none"> <li>- Ingeniero de Sistemas Informáticos o Técnico en Ingeniería de Sistemas Informáticos.</li> <li>- Especializado en programación con Java y JSP</li> </ul>
5 <b>Ingeniero de pruebas</b>	<ul style="list-style-type: none"> <li>- Revisar posibles fallos del sistema.</li> <li>- Anunciar sobre los vacíos en seguridad del sistema</li> </ul>	<ul style="list-style-type: none"> <li>- Ingeniero de Sistemas Informáticos o Técnico en Ingeniería de Sistemas Informáticos.</li> <li>- Especializado en el área de</li> </ul>

	-Reportar todas las fallas	programación.
--	----------------------------	---------------

Tabla 3.2: Tabla resumen de Requerimientos de RR.HH. para el desarrollo de JHard

## Requerimientos de Producción

### Hardware, Software y Tecnologías de Red

No.	Hardware
1	Servidor de Aplicaciones DELL PowerEdge 840
2	UPS MINUTE MAN Pro 320
<b>Software</b>	
4	Mozilla Firefox 3.0.10
5	MySQL Server 5.1
6	SSH
7	Java Development Kit 1.6
8	Sistema Operativo Debian Lenny
<b>Tecnologías de Red</b>	
9	Conexión a Internet

Tabla 3.3: Tabla resumen de Requerimientos de producción de JHard

### Requerimientos de Recursos Humanos

No.	ROL	RESPONSABLE	REQUERIMIENTOS
1	<b>Súper Administrador</b>	<ul style="list-style-type: none"> <li>- Realiza el control administrativo de JHard</li> <li>- Asignar roles de usuario</li> <li>- Crear cuentas de usuario</li> <li>- Borrar usuarios</li> <li>- Revisar contenido</li> </ul>	<ul style="list-style-type: none"> <li>- Ingeniero de Sistemas Informáticos</li> <li>- Especializado en administración de sistemas informáticos</li> <li>- Experiencia en el área de Software</li> </ul>

2	<b>Administrador de Módulo</b>	<ul style="list-style-type: none"> <li>- Administración precisa de un módulo en particular.</li> <li>- Velar por el cumplimiento de los requisitos de funcionalidad del módulo a cargo.</li> </ul>	<ul style="list-style-type: none"> <li>- Ingeniero de Sistemas Informáticos, Técnico en Ingeniería de Sistemas Informáticos o estudiante de Ingeniería de Sistemas Informáticos con más del 80% de su carrera cursada.</li> </ul>
3	<b>Editor de Contenido</b>	<ul style="list-style-type: none"> <li>- Ingreso de contenido a los diversos módulos de JHard que lo requieran</li> <li>- Revisar el contenido de otros editores y personas particulares.</li> </ul>	<ul style="list-style-type: none"> <li>- Ingeniero de Sistemas Informáticos, Técnico en Ingeniería de Sistemas Informáticos o estudiante de Ingeniería de Sistemas Informáticos con más del 60% de su carrera cursada.</li> <li>- Excelente redacción y ortografía.</li> </ul>

Tabla 3.4: Tabla resumen de Requerimientos de RR.HH. para JHard en producción

## Estudio de Factibilidad

### Factibilidad técnica

Para que nuestro proyecto sea técnicamente factible, es necesario identificar las áreas del conocimiento, herramientas, maquinaria, materiales, etc. necesarios en sus procesos de desarrollo e implementación. Tales requerimientos fueron ya listados en el apartado anterior, dentro de los que incluyen las siguientes categorías:

#### 1. Hardware

Tanto las computadoras Windstorm Core<sup>3</sup> como las Dell Vostro<sup>4</sup> pueden obtenerse en el país ahorrando gastos de envío. Las primeras a través de la tienda Tecnoservice<sup>5</sup> o

<sup>3</sup> Computadora cotizada para los programadores de JHard. Ver Anexo 2 de Cuadros de Cotizaciones para computadoras para programadores y Project Managers

<sup>4</sup> Computadora cotizada para el Project Manager de JHard. Ver Anexo 2 de Cuadros de Cotizaciones para computadoras para programadores y Project Managers

<sup>5</sup> Sitio web de Tecnoservice: <http://bit.ly/techserv>

cualquier otra tienda de ventas de hardware al mayoreo y detalle y las segundas directamente en la tienda Dell de El Salvador<sup>6</sup>. El UPS Minuteman Pro también es posible adquirirlo en la tienda Tecnoservice o en caso de no encontrar existencias, este es fácilmente sustituible por uno de la misma calidad y precio similar. El hardware requerido para producción, Dell PowerEdge, tambien se encuentra disponible en la tienda Dell de El Salvador.

- **Software**

Cada una de las herramientas de Software listadas en los requerimientos es libre y gratuita permitiendo su obtención mediante la descarga de los mismos desde sus respectivos sitios web<sup>7</sup>. Además de ello, se puede comprobar que los requerimientos mínimos de hardware para la instalación de estas herramientas se encuentran más que cubiertos por el software propuesto en los requerimientos<sup>8</sup>.

En dichos requerimientos se puede notar que una computadora con un mínimo de 1 GB de memoria RAM, 1.6 Ghz de procesador y 5 GB de espacio en disco duro es suficiente para ejecutar dichas herramientas de software. Comparando estas capacidades con las que poseen los equipos propuestos en los requerimientos de hardware nos damos cuenta que definitivamente los equipos a utilizar soportan la instalación de las herramientas de software requeridas por lo que es técnicamente factible el uso del software propuesto.

- **Tecnologías de red**

Las tecnologías de red requeridas para el proyecto se limitan a la capacidad de acceso a Internet, tanto de la adquisición del servicio por parte de algún ISP<sup>9</sup> y el hardware

---

<sup>6</sup> Sitio web de Dell El Salvador: <http://bit.ly/dellsy>

<sup>7</sup> Descargar Apache Tomcat: <http://bit.ly/dwldtomcat>; descargar Productos de MySql: <http://bit.ly/dwldmysql>; descargar Java: <http://bit.ly/dwldjava>

<sup>8</sup> Requerimientos de hardware de Ubuntu: <http://bit.ly/ubuntureqs>; requerimientos de hardware de Java 5: <http://bit.ly/javareqs>; requerimientos de hardware de MySql: <http://bit.ly/mysqlreqs>

<sup>9</sup> ISP: Proveedor de Servicios de Internet (Internet Service Provider)

necesario para su conexión (tarjetas de red, cable UTP, modem/switch/router) los cuales generalmente vienen incluidos en el plan al adquirir el servicio. Según el SVNet<sup>10</sup>, entre los ISPs disponibles en El Salvador se encuentran Telefónica, Telemovil, Claro, entre otros.

Las velocidades de navegación ofrecidas van desde los 56kbps (vía módem) hasta 2Mbps lo cual es más que suficiente para las tareas que se llevan a cabo en este proyecto de desarrollo tales como la descarga del software requerido, la búsqueda de documentación, el control de versiones del software y la comunicación entre los miembros del equipo. Esto demuestra que es técnicamente factible adquirir y hacer uso de las tecnologías de red requeridas.

- **Recurso humano**

Los estudiantes encargados de desarrollar este trabajo de grado poseen la suficiente experiencia en el desarrollo de aplicaciones usando el lenguaje Java orientado al web ya que durante las últimas asignaturas cursadas en la facultad se vieron en la tarea de hacer uso de dichas tecnologías para llevar a cabo proyectos de software implicados en ellas.

Además de esto, cada uno de ellos se encuentra laborando en empresas u organizaciones donde constantemente aplican estas habilidades adquiridas durante el periodo universitario en proyectos de la vida real con lo cual van adquiriendo nuevos conocimientos y experiencia a través del tiempo, aplicando conceptos como control de versiones, pruebas unitarias, control de calidad, administración de tiempos de las tareas, administración de requerimientos, etc. Esto los califica como completamente capacitados para llevar a cabo este proyecto de tesis con grandes probabilidades de éxito.

Además de ello, los que cumplirán el rol de usuarios finales y administradores del sistema son también profesionales graduados de la carrera de ingeniería en sistemas o carreras

---

<sup>10</sup> ISPs en El Salvador, según SVNet: <http://bit.ly/svnetisp>

afines además de poseer suficiente experiencia en el área de sistemas, la cual les permitirá hacer un buen uso del sistema aplicando sus propios conocimientos a la hora de comprender su funcionamiento.

Todo esto demuestra que es técnicamente factible llevar a cabo este proyecto con el recurso humano propuesto.

### **Factibilidad Operacional**

Se refiere al hecho de que si trabajará o no el sistema si este se llega a desarrollar. Entre las preguntas clave a contestar en este apartado se encuentran:

1. ¿Existe apoyo suficiente para el proyecto por parte de la administración?, ¿Y por parte de los usuarios?
2. Los métodos que actualmente se usan en la empresa, ¿son aceptados por los usuarios?
3. ¿Los usuarios han participado en la planeación y desarrollo del proyecto?, ¿Cómo lo han hecho?
4. ¿El sistema propuesto causará perjuicios?
5. ¿Producirá resultados pobres en algún área?
6. ¿Se perderá control en algún área específica?
7. ¿Se perderá la facilidad de acceso a la información?
8. ¿La productividad de los empleados será menor después de instalado el sistema?
9. ¿Los clientes se verán afectados por la implantación?

Para dar respuesta a cada una de las preguntas anteriormente planteadas se propuso realizar una serie de encuestas y entrevistas con todas las personas que tendrán la oportunidad de interactuar con el sistema una vez terminado. Entre estas personas se encuentran los docentes de la facultad, estudiantes activos y los administradores del sistema como tal.

En la sección denominada *Metodología de la Investigación* del capítulo dos de esta tesis se detalla todo el proceso que se llevo a cabo para obtener información de las personas que estarán involucradas con el sistema. Al estudiar la sección que contiene el análisis de los resultados es posible darse cuenta que los involucrados perciben la necesidad de la existencia de un sistema que administre las tareas relacionadas con la unidad de Hardware y Software de la facultad.

Tomando en cuenta los resultados mencionados en el párrafo anterior se puede concluir que los usuarios finales y otros facilitadores requeridos están en la disposición total de apoyar el desarrollo del proyecto en cuestión, lo cual garantiza en gran manera el éxito del mismo.

Los resultados también indican que la pregunta 2 de esta lista posee también una respuesta afirmativa, lo cual indica que los usuarios aceptan la metodología de trabajo actual ya que no existe una mejor, aunque también en los resultados dichos usuarios expresan su deseo y apoyo al desarrollo de un sistema que agilice los procesos actuales. La tercera pregunta es confirmada por las constantes reuniones que existen entre los desarrolladores del proyecto y los usuarios finales, mismos que se encargan de administrar la Unidad de Hardware y Software.

Además, dichos resultados también demuestran que las preguntas de la 4 a la 9 poseen una respuesta negativa al hacer notar que todos los cambios que se harán en los procesos actuales de la unidad de Hardware y Software serán orientados a una mejora en la calidad del servicio brindado por la misma.

### **Factibilidad Financiera y Económica**

Un sistema puede ser factible desde el punto de vista técnico y operacional, pero sino es factible económicamente para la organización no puede ser implantado. Para ello, se debe hacer una estimación del costo total del proyecto y compararlo con los beneficios

esperados de tal manera que se pueda demostrar que la inversión será retribuida en un futuro.

### Costos del Proyecto

<b>Hardware</b>		<b>\$2,184.5</b>
<b>2</b>	Desktop Windstorm Core 2.33ghz (2 x \$388.00)	\$776.00
<b>1</b>	Desktop Dell Vostro 220 Mini Tower	\$379.00
<b>3</b>	UPS MINUTE MAN Pro 320 (3 x \$43.50)	\$130.50
<b>1</b>	Servidor de Aplicaciones DELL PowerEdge 840	\$899.00
<b>Software</b>		<b>\$0.00</b>
<b>3</b>	Apache Tomcat 6.0.x	\$0.00
<b>3</b>	Netbeans 6.5.1	\$0.00
<b>3</b>	Mozilla Firefox 3.0.10	\$0.00
<b>3</b>	Firebug 1.4.0	\$0.00
<b>3</b>	MySQL Server 5.1, Administrador, Query Browser y Workbench	\$0.00
<b>3</b>	Java Development Kit 1.6	\$0.00
<b>3</b>	IceFaces 1.8.0	\$0.00
<b>3</b>	Subversion 1.5	\$0.00
<b>3</b>	Ubuntu 9.04	\$0.00
<b>1</b>	Debian Lenny	\$0.00
<b>Tecnologías de Red</b>		<b>\$603.75</b>
<b>3</b>	Conexión a Internet 512 Kbps (\$28.75 x 7 meses x 3)	\$603.75
<b>TOTAL</b>		<b>\$2,788.25</b>

Tabla 3.5: Costo de JHard

### Factibilidad Legal

Para la realización del proyecto no se incurre en ninguna infracción de tipo legal, ya que el software utilizado se encuentra en su totalidad bajo licencias GPL o derivadas. Estas nos permiten el libre uso, modificación y distribución del software bajo esta licencia tanto para la fase de desarrollo como para la implementación de este proyecto.

A continuación se listan las herramientas de software requeridas para desarrollar e implementar este proyecto, junto con su respectiva licencia de uso:

- Apache Tomcat: **Apache Public License**
- Netbeans y sus plugins: **Netbeans Public License**
- Mysql Server, Administrator, Query Analyzer y Workbench: **General Public License (GPL) para la Community Edition**
- Java Development Kit y Runtime Machine: **Common Development and Distribution License (CDDL) en conjunto con la General Public License (GPL)**
- DHTMLX Scheduler **Open Source –GPL License**

## CAPÍTULO IV: Diseño y Desarrollo del sistema

### Modelado y Diseño de la Base de Datos

La base de datos de JHard cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Se trata del actual paradigma en los modelos de base de datos.

El gestor que alberga la base de datos de JHard es MySQL 5.1

¿Por qué MySQL?

- Goza de completa compatibilidad con Java (SUN es el fabricante de ambos)
- Posee completa integración con Netbeans MySQL,
- Total compatibilidad con JPA y persistencia,
- Soporta procesos almacenados, triggers,
- Licencia GPL

Las necesidades encontradas en la investigación de campo que atañen a todas las personas involucradas son variadas. Todas igual de importantes. Esto influye en todo el diseño de la aplicación, desde el diseño de su base de datos, el diseño de las reglas de negocio, así como el diseño de su interfaz y de las opciones a ofrecer al usuario final. Se decidió dividir o *modularizar* el sistema en las áreas funcionales que lo conformarán. Los módulos propuestos según las necesidades encontradas son:

1. **JInvent:** Manejo de inventario del laboratorio de hardware (No consumibles)
2. **JRequest:** Solicitud de servicio al laboratorio
3. **JWiki/JProCur:** Modulo colaborativo de conocimiento, con soluciones a problemas

comunes, etc.

4. **ManLab:** Modulo de para la gestión de inscripción de laboratorios prácticos en el Laboratorio de Hardware.
5. **JCanon:** Modulo para la gestión de reserva de cañones y/o laptops.
6. **JHardmin:** Interfaz administrativa con la que se podrá administrar los perfiles, roles y autorizaciones de los usuarios.

El diagrama Entidad-Relación (ER) se ordenó a fin que cada módulo tuviera un grupo de tablas o entidades afines para las labores que realizan. Sin embargo, cada módulo no es un sistema aislado, por lo tanto hay relaciones entre ellos. El diagrama ER de JHard es el siguiente:

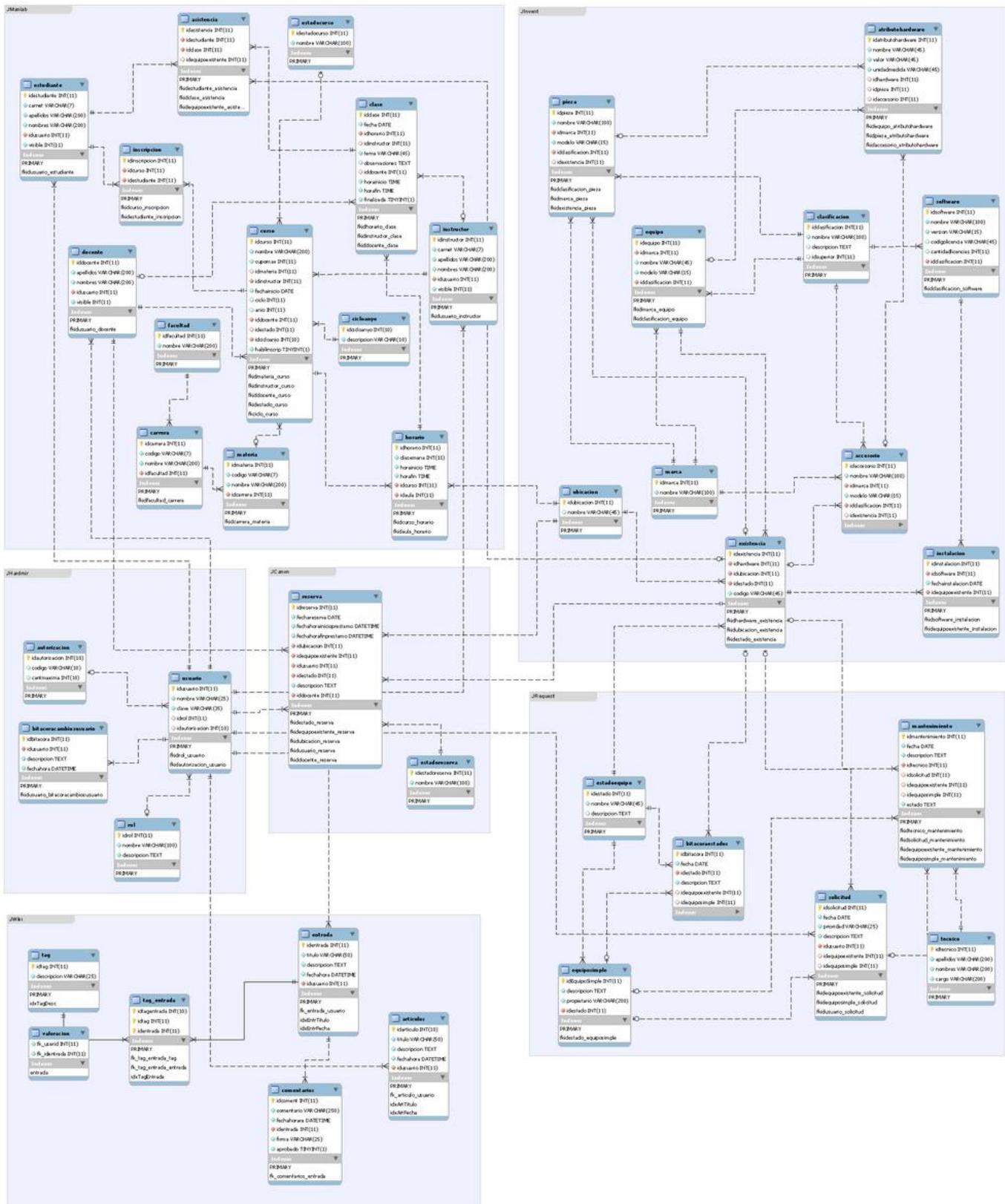


Figura 4.1: Diagrama Entidad-Relación de JHard

El ER representa las relaciones entre las tablas de la base de datos. Cada tabla o entidad tiene sus propios atributos o campos. Del lado de la aplicación, la persistencia *mapea* las tablas de la base de datos y crea las entidades, las cuales son objetos instanciables dentro de nuestro sistema. Ahora se mostrará el ER segmentado por módulos, y las relaciones con otras *entities*:

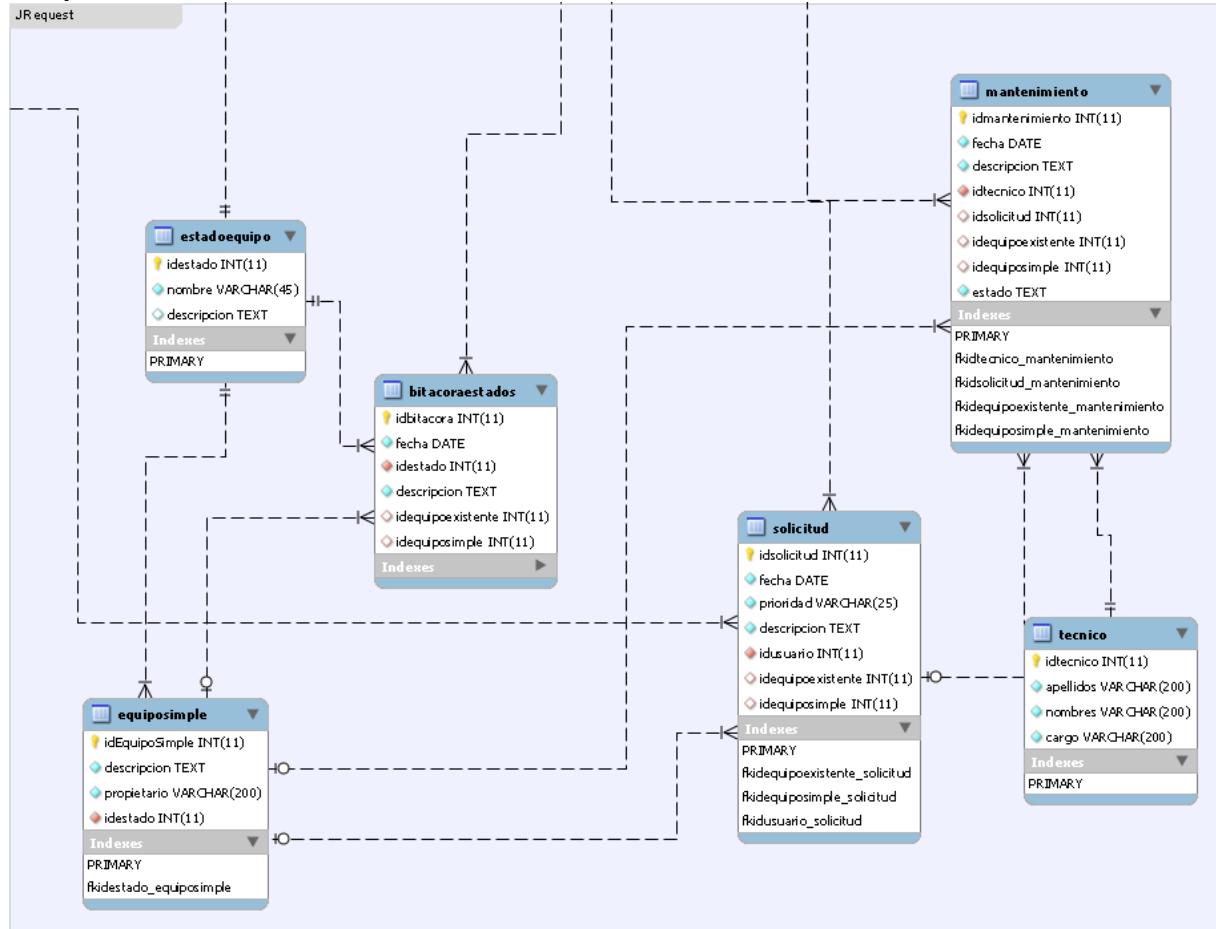
**JRequest**

Figura 4.2: Diagrama ER de JRequest

JRequest está fuertemente relacionado con los siguientes módulos:

- Un **Estadoequipo** puede tener muchas Existencia de **JInvent**. La relación es **Estadoequipo/Existencia** a través del Index **fkidestadoequipo\_existencia**.

- Una Existencia de JInvent puede tener muchos Estadoequipo. La relación es Existencia/Bitacoraestados a través del Index fkidequipoexistente\_bitacoraestados.
- Una Existencia de JInvent puede tener muchos Mantenimiento. La relación es Existencia/Mantenimiento a través del Index fkidequipoexistente\_mantenimiento.
- Una Existencia de JInvent puede tener muchas Solicitud. La relación es Existencia/Solicitud a través del Index fkidequipoexistente\_solicitud.
- Un Usuario de JHardmin puede tener muchas Solicitud. La relación es Usuario/Solicitud a través del Index fkidusuario\_solicitud.

### JWiki y JProCur

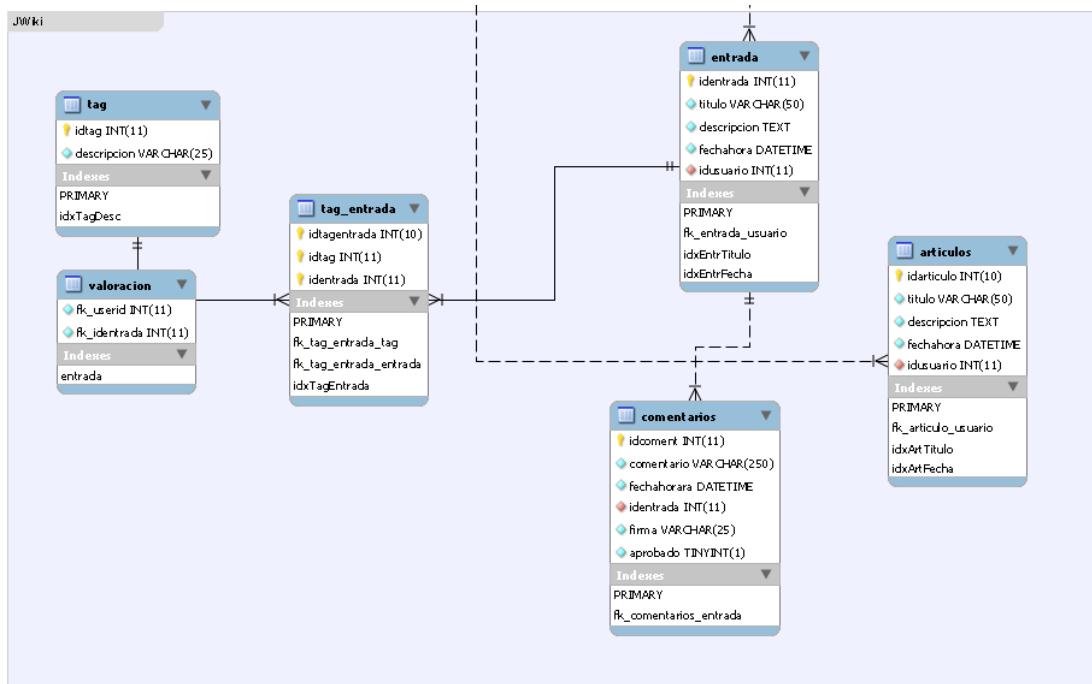


Figura 4.3: Diagrama ER de JWiki/JProCur

**La relación que mantiene es únicamente con JHardmin y es de la siguiente manera:**

- Un Usuario de JHardmin puede tener muchas Entrada. La relación es Usuario/Entrada a través del Index fkidusuario\_entraida.

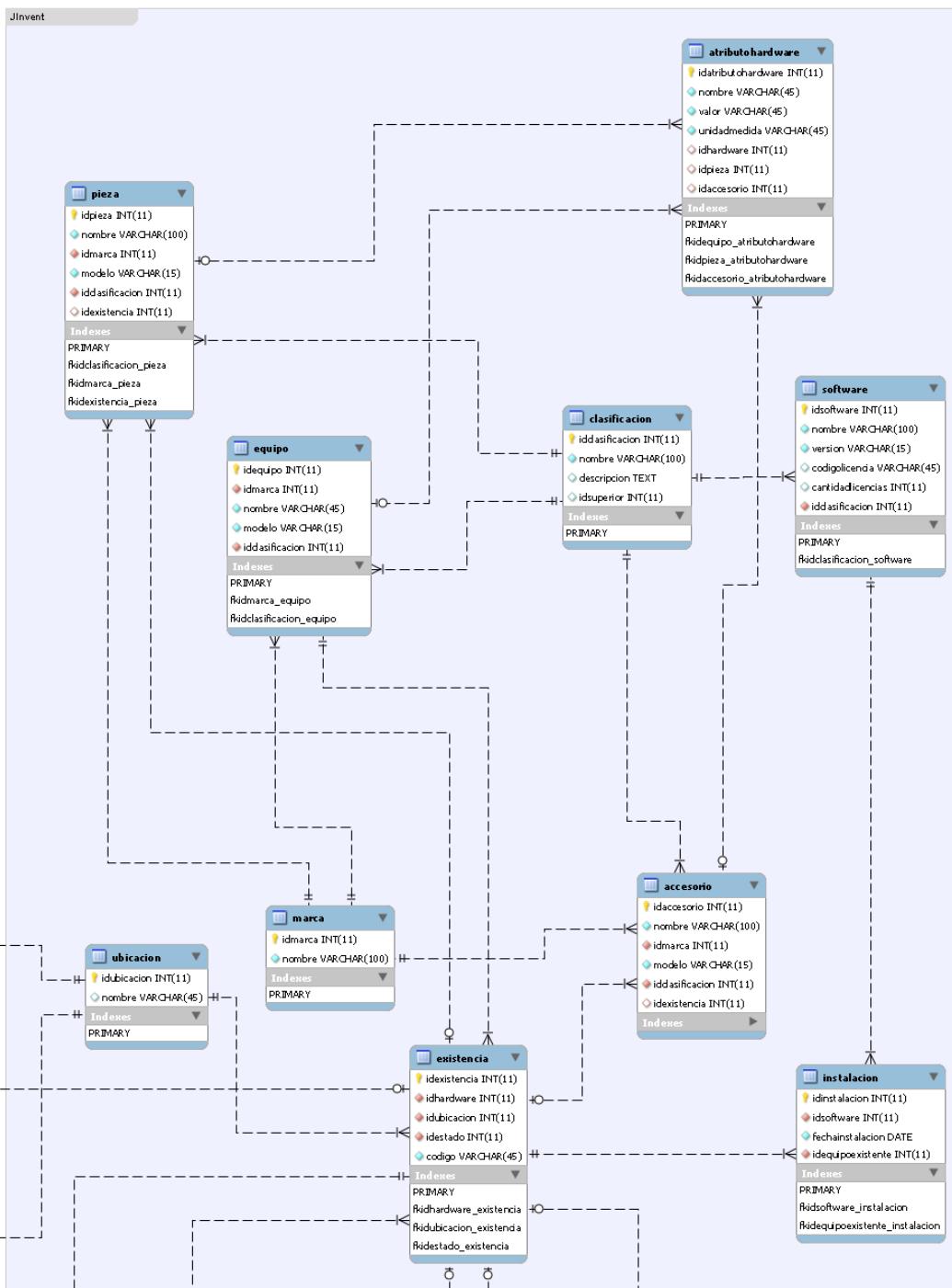
**JInvent**

Figura 4.4: Diagrama ER de JInvent

Además de las relaciones antes mencionadas entre JRequest y JInvent, éste último posee las siguientes relaciones con el resto de módulos:

- Un Existencia puede tener muchas Asistencia de ManLab. La relación es Existencia/Asistencia a través del Index fkidequipoexistente\_asistencia.

- Una Ubicación puede tener muchos Horario de ManLab. La relación es Ubicación/Horario a través del Index fkidubicacion\_horario.
- Una Existencia puede tener muchas Reserva de JCanon. La relación es Existencia/Reserva a través del Index fkidequipoexistente\_reserva.

### ManLab

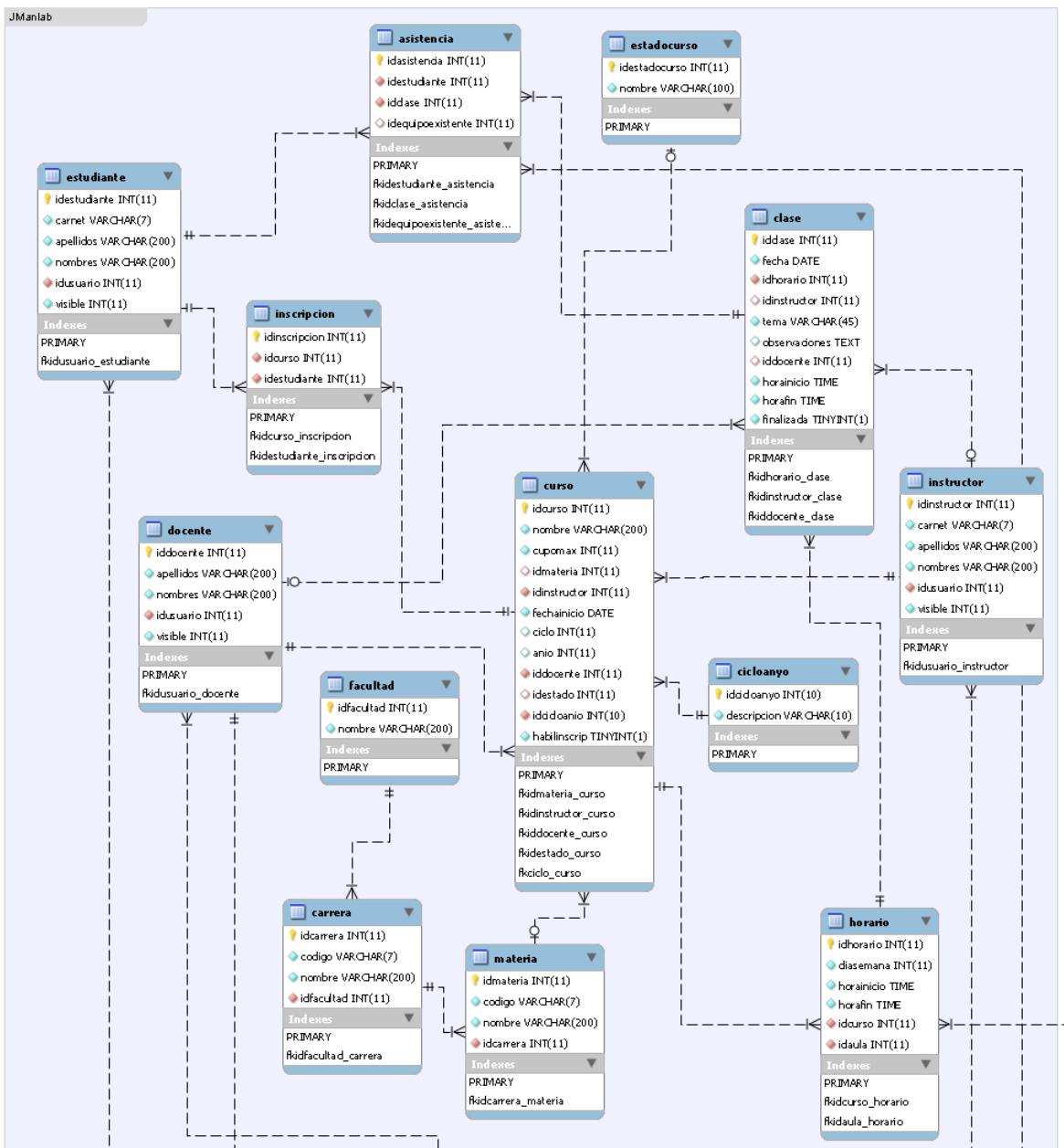


Figura 4.5: Diagrama ER de ManLab

El módulo de ManLab posee muchas relaciones con el módulo JHardmin. Son las siguientes:

- Un Usuario de JHardmin puede tener un Docente. La relación es Usuario/Docente a través del Index fkidusuario\_docente.
- Un Usuario de JHardmin puede tener un Estudiante. La relación es Usuario/Estudiante a través del Index fkidusuario\_estudiante.
- Un Usuario de JHardmin puede tener un Instructor. La relación es Usuario/Instructor a través del Index fkidusuario\_instructor.

### JCanon

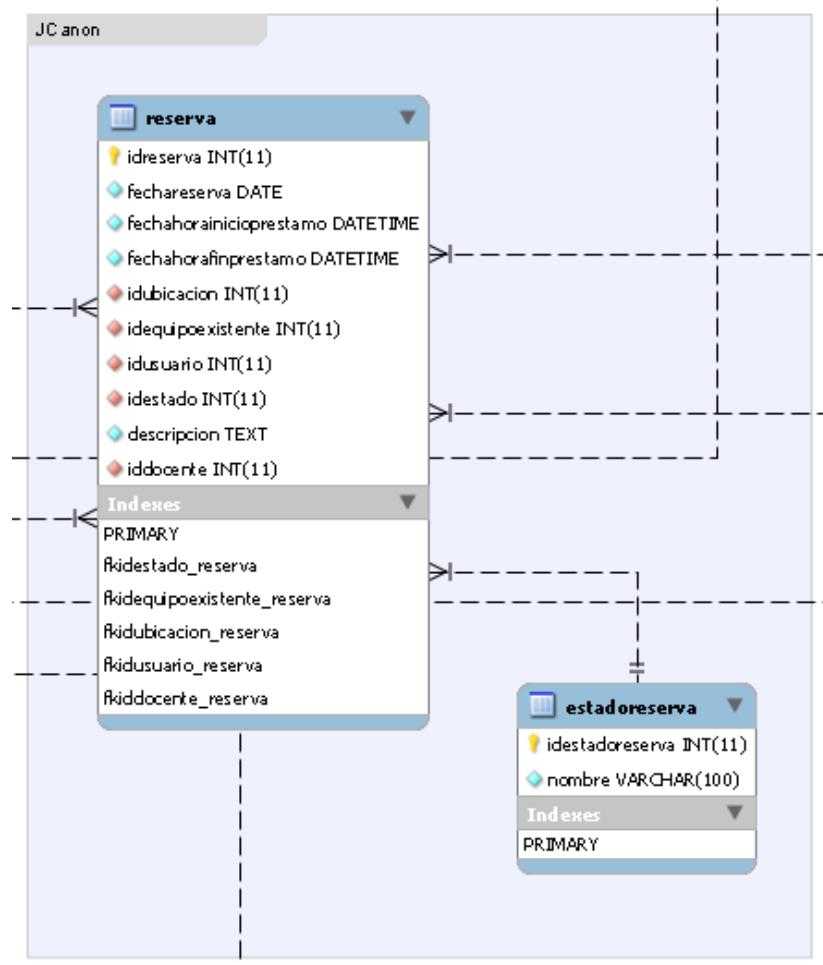


Figura 4.6: Diagrama ER de JCanon

JCanon tiene las siguientes relaciones con módulos de JHard:

- Una Reserva puede tener muchos Usuario de JHardmin. La relación es Reserva/Usuario a través del Index fkidusuario\_reserva.
- Una Reserva puede tener muchos Docente de ManLab. La relación es Reserva/Docente a través del Index fkiddocente\_reserva.

JHardmin

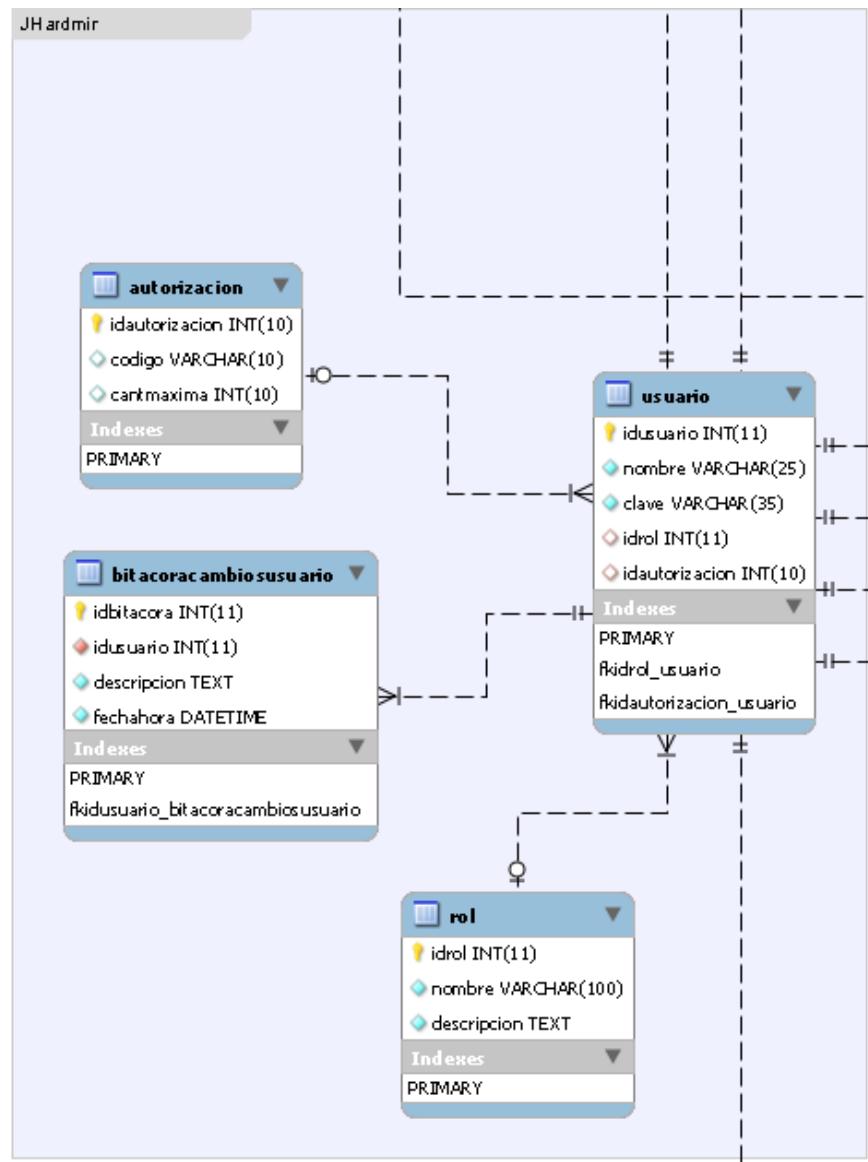


Figura 4.7: Diagrama ER de JHardmin

JHardmin, por su carácter administrativo, se relaciona con cada una de los módulos restantes, ya que es el encargado, entre otras cosas, de manejar las sesiones de cada usuario registrado a nivel de aplicación.

## Diseño y Desarrollo de la Capa de Acceso a Datos

La capa de Acceso a datos de JHard es manejada con la Java Persistence API, más conocida por sus siglas como *JPA*.

JPA es la API<sup>11</sup> de persistencia desarrollada para la plataforma Java EE<sup>12</sup> e incluida en el estándar EJB3<sup>13</sup>. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJO's).

Proporciona un **estándar** para gestionar datos relacionales en aplicaciones Java SE<sup>14</sup> o Java EE, de forma que además se **simplifique el desarrollo** de la persistencia de datos.

Aunque ha sido definida como parte de la especificación EJB 3.0 (Java EE 5), que supone una simplificación sobre versiones anteriores, ya no requiere de un contenedor EJB ni un servidor de aplicaciones Java EE.

Es una API de persistencia de POJO's (Plain Old Java Object). Es decir, objetos simples que no heredan ni implementan otras clases (como los EJBs).

En su definición, ha combinado ideas y conceptos de los principales frameworks de persistencia, como Hibernate<sup>15</sup>, Toplink y JDO<sup>16</sup>, y de las versiones anteriores de EJB. Todos estos cuentan actualmente con una implementación JPA.

---

<sup>11</sup> Una **API** o **interfaz de programación de aplicaciones** (en inglés Application Programming Interface) es el conjunto de funciones, métodos y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

<sup>12</sup> **Java Platform, Enterprise Edition** o **Java EE** es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

<sup>13</sup> **EJB3** es un reacondicionamiento y una simplificación a la especificación de EJB. Las metas de EJB 3.0 son simplificar el desarrollo, facilitar el desarrollo conducido por prueba, y centrarse más en escribir Plain Old Java Objects (POJO's) en lugar de complejas API's EJB.

<sup>14</sup> **Java Standard Edition** o **Java SE** es una colección de API del lenguaje de programación Java útiles para muchos programas de la Plataforma Java. La Plataforma Java 2, Enterprise Edition incluye todas las clases en el Java SE, además de algunas de las cuales son útiles para programas que se ejecutan en servidores sobre estaciones de trabajo.

El mapeo objeto-relacional (es decir, la relación entre entidades Java y tablas de la base de datos, *queries* con nombre, etc) se realiza mediante anotaciones en las propias clases de entidad. No se requieren ficheros descriptores XML. También pueden definirse transacciones como anotaciones JPA.

Existió una alta motivación para crear la API persistencia de Java. Muchos programadores enterprise de Java han estado utilizando los objetos persistentes ligeros proporcionados por frameworks open-source u objetos de acceso a datos instanciados a partir de *entity beans*, ya que los Entities de Enterprise Beans eran considerados demasiado pesados y complicados, y podrían ser utilizados solamente en los servidores de aplicaciones de Java EE. Muchas de las características de los frameworks de persistencia *Third-Party* fueron incorporadas en la API de persistencia de Java, y los proyectos como Hibernate y la versión open-source TopLink Essentials del fabricante Oracle son ahora implementaciones de la API de persistencia de Java.

La implementación de JPA que JHard utiliza es TopLink Essentials<sup>17</sup> de Oracle. Ésta ofrece una solución probada de Java para todas las necesidades de persistencia basadas en alto rendimiento y capacidad de conversión a escala y flexibilidad en arquitectura y diseño. Está probado que Oracle con TopLink trae mayor agilidad, una mejor toma de decisión, y coste y riesgo reducidos a diversos ámbitos de *IT* en las empresas actuales.

---

<sup>15</sup> **Hibernate** es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Es software libre, distribuido bajo los términos de la licencia GNU LGPL.

<sup>16</sup> **Los objetos de datos de Java** (JDO en inglés) son una especificación de la persistencia de objetos Java. Una de sus características es la transparencia de los servicios persistentes al modelo del dominio. Los objetos persistentes de JDO son clases de programación ordinarias del lenguaje de Java; no existe ningún requisito para que ellas ejecuten ciertos interfaces o extiendan de clases especiales.

<sup>17</sup> **TopLink Essentials** es la puesta en práctica de la referencia de la persistencia API (JPA) de EJB 3.0 Java y de la Community Edition de TopLink de Oracle. Essentials es una versión limitada del producto propietario. Proporciona un marco de gran alcance y flexible para almacenar los objetos de Java en una base de datos relacional o para convertir los objetos de Java a los documentos de XML.

Una de las características que hacen de TopLink una implementación de JPA robusta es el mapping<sup>18</sup> que realiza de las tablas de la base de datos a por medio de un enlace Objeto XML, el cual convierte la entidad en un objeto instanciable a través de una estructura XML, muy utilizada en estos días en el mundo del desarrollo en IT.

### ¿Por qué apostar por JPA?

- Genera automáticamente la capa de Acceso a Datos<sup>19</sup> de la aplicación y se encarga de mantenerla.
- Genera automáticamente las clases (entities) que representan objetos del la aplicación, que a su vez son las mismas tablas dentro de la base de datos.
- Es mucho más fácil realizar varios CRUD<sup>20</sup> para objetos definidos en la aplicación
- Nos centramos únicamente en la capa del Negocio y la Interfaz de usuario o GUI.

### ¿Por qué TopLink?

- El prestigio de Oracle respalda a dicha implementación
- El *mapeo Objeto-XML* de las entidades lo hace más efectivo y ágil en comparación de otras implementaciones que utilizan DOM, SAX, o StAX, que son implementaciones de mapeos de objetos hacia XML.
- Licencia GPL para desarrolladores

### ¿Qué son las Entities?

Se ha hablado mucho de la implementación de las entities (entidades en español) como parte fundamental de la API de persistencia.

<sup>18</sup> **Data Mapping** es el proceso de crear “mapas” de elementos de datos entre dos modelos de datos distintos. El trazado de los datos se utiliza en primer lugar para una gran variedad de tareas de la integración de datos.

<sup>19</sup> **La Capa de Acceso a Datos** forma parte de la **programación por capas**. Es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. La capa de datos únicamente provee la información desde una base de datos hacia el negocio de dicha aplicación.

<sup>20</sup> **CRUD** es el acrónimo de Crear, Obtener, Actualizar y Borrar (Create, Read, Update y Delete en inglés). Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un sistema de software.

Una entidad de persistencia es una clase Java ligera que representa típicamente una tabla en una base de datos relacional. Los casos de la entidad corresponden a las filas individuales en la tabla. Las entidades tienen típicamente relaciones con otras entidades, y estas relaciones se expresan con objetos/relaciones de metadatos. El objeto/relación de metadatos se puede especificar directamente en el archivo de la clase de la entidad usando anotaciones, o en un archivo separado del descriptor de XML distribuido con el uso.

JPA, además de innovar el acceso a datos de una aplicación, innova en otros sentidos. Uno de ellos es que posee su propio lenguaje para construir consultas estructuradas, conocido como Java Persistence Query Language (Lenguaje de Consultas de la Persistencia de Java) conocido ampliamente en el mundo de la persistencia por sus siglas en inglés, JPQL .

Se utiliza para hacer consultas contra las entidades almacenadas en una base de datos relacional. Las consultas se asemejan a la sintaxis SQL tradicional, pero trabajan en las entidades que han sido mapeadas directamente de la base de datos relacional. A pesar que se asemeja al SQL convencional, posee ciertas diferencias en cuanto a algunas palabras reservadas y operaciones que permite. Las consultas nombradas o *Named Queries* se declaran dentro de la entidad propietaria de éstas.

## Diagrama de Clases

El diagrama de clases es proporcionado por el JPA y la persistencia de entidades de la base de datos convertidas a clases Java. Tómese a consideración que el detalle de cada uno de los atributos y métodos de cada clase irá minuciosamente explicada en el Diccionario de Datos del Capítulo VI de este documento. El diagrama es el siguiente:

<b>Docente</b>
<p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer iddocente private String apellidos private String nombres private int visible</pre>
<p><i>Operations</i></p> <pre>public Docente( ) public Docente( Integer iddocente ) public Docente( Integer iddocente, String apellidos, String nombres, int visible ) public Integer getIddocente( ) public void setIddocente( Integer iddocente ) public String getApellidos( ) public void setApellidos( String apellidos ) public String getNombres( ) public void setNombres( String nombres ) public int isVisible( ) public void setVisible( int visible ) public Curso[0..*] getCursoCollection( ) public void setCursoCollection( Curso cursoCollection[0..*] ) public Clase[0..*] getClaseCollection( ) public void setClaseCollection( Clase claseCollection[0..*] ) public Usuario getIdusuario( ) public void setIdusuario( Usuario idusuario ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>

<b>Estadoequipo</b>
<p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idestado private String nombre private String descripcion</pre> <p><i>Operations</i></p> <pre>public Estadoequipo( ) public Estadoequipo( Integer idestado ) public Estadoequipo( Integer idestado, String nombre ) public Integer getIdestado( ) public void setIdestado( Integer idestado ) public String getNombre( ) public void setNombre( String nombre ) public String getDescripcion( ) public void setDescripcion( String descripcion ) public Equiposimple[0..*] getEquiposimpleCollection( ) public void setEquiposimpleCollection( Equiposimple equiposimpleCollection[0..*] ) public Existencia[0..*] getExistenciaCollection( ) public void setExistenciaCollection( Existencia existenciaCollection[0..*] ) public Bitacoraestados[0..*] getBitacoraestadosCollection( ) public void setBitacoraestadosCollection( Bitacoraestados bitacoraestadosCollection[0..*] ) public int hashCode( ) public boolean equals( Object object ) public String toString( ) public String getImagenEstado( )</pre>

<h3><b>Estadocurso</b></h3> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idestadocurso private String nombre</pre> <p><i>Operations</i></p> <pre>public Estadocurso( ) public Estadocurso( Integer idestadocurso ) public Estadocurso( Integer idestadocurso, String nombre ) public Integer getidestadocurso( ) public void setidestadocurso( Integer idestadocurso ) public String getNombre( ) public void setNombre( String nombre ) public Curso[0..*] getCursoCollection( ) public void setCursoCollection( Curso cursoCollection[0..*] ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>	<h3><b>Equipo</b></h3> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idequipo private String nombre private String modelo</pre> <p><i>Operations</i></p> <pre>public Equipo( ) public Equipo( Integer idequipo ) public Equipo( Integer idequipo, String nombre, String modelo ) public Integer getIdequipo( ) public void setIdequipo( Integer idequipo ) public String getNombre( ) public void setNombre( String nombre ) public String getModelo( ) public void setModelo( String modelo ) public Existencia[0..*] getExistenciaCollection( ) public void setExistenciaCollection( Existencia existenciaCollection[0..*] ) public Marca getIdmarca( ) public void setIdmarca( Marca idmarca ) public Clasificacion getIdclasificacion( ) public void setIdclasificacion( Clasificacion idclasificacion ) public Atributohardware[0..*] getAtributohardwareCollection( ) public void setAtributohardwareCollection( Atributohardware atributohardwareCollection[0..*] ) public int hashCode( ) public boolean equals( Object object ) public String toString( ) public int getExistenciaSize( )</pre>	<h3><b>Instalacion</b></h3> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idinstalacion private Date fechainstalacion</pre> <p><i>Operations</i></p> <pre>public Instalacion( ) public Instalacion( Integer idinstalacion ) public Instalacion( Integer idinstalacion, Date fechainstalacion ) public Integer getIdinstalacion( ) public void setIdinstalacion( Integer idinstalacion ) public Date getFechainstalacion( ) public void setFechainstalacion( Date fechainstalacion ) public Software getIdsoftware( ) public void setIdsoftware( Software idsoftware ) public Existencia getidequipoexistente( ) public void setidequipoexistente( Existencia idequipoexistente ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>
---	---	---

<p><b>Materia</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idmateria private String codigo private String nombre</pre> <p><i>Operations</i></p> <pre>public Materia( ) public Materia(Integer idmateria) public Materia(Integer idmateria, String codigo, String nombre) public Integer getIdmateria( ) public void setIdmateria(Integer idmateria) public String getCodigo( ) public void setCodigo(String codigo) public String getNombre( ) public void setNombre(String nombre) public Carrera getIdcarrera( ) public void setIdcarrera(Carrera idcarrera) public Curso[0..*] getCursoCollection( ) public void setCursoCollection(Curso cursoCollection[0..*]) public int hashCode( ) public boolean equals(Object object) public String toString( )</pre>	<p><b>Comentarios</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idcomentario private String comentario private Date fechahorara private String firma private boolean aprobado</pre> <p><i>Operations</i></p> <pre>public Comentarios( ) public Comentarios(Integer idcomentario) public Integer getIdcomentario( ) public void setIdcomentario(Integer idcomentario) public String getComentario( ) public void setComentario(String comentario) public Date getFechahorara( ) public String getFecha( ) public void setFechahorara(Date fechahorara) public String getFirma( ) public void setFirma(String firma) public boolean getAprobado( ) public void setAprobado(boolean aprobado) public Entrada getIdentrada( ) public void setIdentrada(Entrada identrada) public int hashCode( ) public boolean equals(Object object) public String toString( )</pre>	<p><b>Instructor</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idinstructor private String carnet private String apellidos private String nombres private int visible</pre> <p><i>Operations</i></p> <pre>public Instructor( ) public Instructor(Integer idinstructor) public Integer getIdinstructor( ) public void setIdinstructor(Integer idinstructor) public String getCarnet( ) public void setCarnet(String carnet) public String getApellidos( ) public void setApellidos(String apellidos) public String getNombres( ) public void setNombres(String nombres) public int getVisible( ) public void setVisible(int visible) public Usuario getIdusuario( ) public void setIdusuario(Usuario idusuario) public Curso[0..*] getCursoCollection( ) public void setCursoCollection(Curso cursoCollection[0..*]) public Clase[0..*] getClaseCollection( ) public void setClaseCollection(Clase claseCollection[0..*]) public boolean equals(Object object) public String toString( )</pre>
---	--	--

<p><b>EquipoSimple</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idEquipoSimple private String descripcion private String propietario</pre> <p><i>Operations</i></p> <pre>public EquipoSimple( ) public EquipoSimple( Integer idEquipoSimple ) public EquipoSimple( Integer idEquipoSimple, String descripcion, String propietario ) public Integer getIdEquipoSimple( ) public void setIdEquipoSimple( Integer idEquipoSimple ) public String getDescripcion( ) public void setDescripcion( String descripcion ) public String getPropietario( ) public void setPropietario( String propietario ) public Mantenimiento[0..*] getMantenimientoCollection( ) public void setMantenimientoCollection( Mantenimiento mantenimientoCollection[0..*] ) public EstadoEquipo getEstado( ) public void setEstado( EstadoEquipo iestado ) public Solicitud[0..*] getSolicitudCollection( ) public void setSolicitudCollection( Solicitud solicitudCollection[0..*] ) public BitacoraEstados[0..*] getBitacoraEstadosCollection( ) public void setBitacoraEstadosCollection( BitacoraEstados bitacoraEstadosCollection[0..*] ) public int hashCode( ) public boolean equals( Object object ) public String toString( ) public String getDescUltimaSolicitud( ) public String getNombreTecnicoActualMnto( )</pre>	<p><b>Ubicacion</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idubicacion private String nombre</pre> <p><i>Operations</i></p> <pre>public Ubicacion( ) public Ubicacion( Integer idubicacion ) public Integer getIdubicacion( ) public void setIdubicacion( Integer idubicacion ) public String getNombre( ) public void setNombre( String nombre ) public Horario[0..*] getHorarioCollection( ) public void setHorarioCollection( Horario horarioCollection[0..*] ) public Existencia[0..*] getExistenciaCollection( ) public void setExistenciaCollection( Existencia existenciaCollection[0..*] ) public Reserva[0..*] getReservaCollection( ) public void setReservaCollection( Reserva reservaCollection[0..*] ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>	<p><b>Configuracion</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private String clave private String valor</pre> <p><i>Operations</i></p> <pre>public Configuracion( ) public Configuracion( String clave ) public Configuracion( String clave, String valor ) public String getClave( ) public void setClave( String clave ) public String getValor( ) public void setValor( String valor ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>
---	---	--

<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">■ Asistencia</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;"> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idasistencia</pre> <p><i>Operations</i></p> <pre>public Asistencia( ) public Asistencia( Integer idasistencia ) public Integer getidasistencia( ) public void setidasistencia( Integer idasistencia ) public Clase getidclase( ) public void setidclase( Clase idclase ) public Existencia getidequipoexistente( ) public void setidequipoexistente( Existencia idequipoexistente ) public Estudiante getidestudiante( ) public void setidestudiante( Estudiante idestudiante ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre> </td></tr> </tbody> </table>	■ Asistencia	<p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idasistencia</pre> <p><i>Operations</i></p> <pre>public Asistencia( ) public Asistencia( Integer idasistencia ) public Integer getidasistencia( ) public void setidasistencia( Integer idasistencia ) public Clase getidclase( ) public void setidclase( Clase idclase ) public Existencia getidequipoexistente( ) public void setidequipoexistente( Existencia idequipoexistente ) public Estudiante getidestudiante( ) public void setidestudiante( Estudiante idestudiante ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">■ Reserva</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;"> <p><i>Attributes</i></p> <pre>private Integer idreserva private Date fechareserva private Date fechahorainicioprestamo private Date fechahorafinprestamo private String descripcion</pre> <p><i>Operations</i></p> <pre>public Reserva( ) public Reserva( Integer idreserva ) public Integer getidreserva( ) public void setidreserva( Integer idreserva ) public Date getFechareserva( ) public void setFechareserva( Date fechareserva ) public Date getFechahorainicioprestamo( ) public void setFechahorainicioprestamo( Date fechahorainicioprestamo ) public Date getFechahorafinprestamo( ) public void setFechahorafinprestamo( Date fechahorafinprestamo ) public String getDescripcion( ) public void setDescripcion( String descripcion ) public Estadoreserva getdestado( ) public void setdestado( Estadoreserva idestado ) public Existencia getidequipoexistente( ) public void setidequipoexistente( Existencia idequipoexistente ) public void setidubicacion( Ubicacion idubicacion ) public Usuario getidusuario( ) public void setidusuario( Usuario idusuario ) public Docente getiddocente( ) public void setiddocente( Docente iddocente ) public boolean equals( Object object ) public String toString( ) public String getfechaReserva( ) public String getthoraInicio( ) public String getthoraFinal( )</pre> </td></tr> </tbody> </table>	■ Reserva	<p><i>Attributes</i></p> <pre>private Integer idreserva private Date fechareserva private Date fechahorainicioprestamo private Date fechahorafinprestamo private String descripcion</pre> <p><i>Operations</i></p> <pre>public Reserva( ) public Reserva( Integer idreserva ) public Integer getidreserva( ) public void setidreserva( Integer idreserva ) public Date getFechareserva( ) public void setFechareserva( Date fechareserva ) public Date getFechahorainicioprestamo( ) public void setFechahorainicioprestamo( Date fechahorainicioprestamo ) public Date getFechahorafinprestamo( ) public void setFechahorafinprestamo( Date fechahorafinprestamo ) public String getDescripcion( ) public void setDescripcion( String descripcion ) public Estadoreserva getdestado( ) public void setdestado( Estadoreserva idestado ) public Existencia getidequipoexistente( ) public void setidequipoexistente( Existencia idequipoexistente ) public void setidubicacion( Ubicacion idubicacion ) public Usuario getidusuario( ) public void setidusuario( Usuario idusuario ) public Docente getiddocente( ) public void setiddocente( Docente iddocente ) public boolean equals( Object object ) public String toString( ) public String getfechaReserva( ) public String getthoraInicio( ) public String getthoraFinal( )</pre>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">■ Cicloanyo</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;"> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idcicloanyo private String descripcion</pre> <p><i>Operations</i></p> <pre>public Cicloanyo( ) public Cicloanyo( Integer idcicloanyo ) public Cicloanyo( Integer idcicloanyo, String descripcion ) public Integer getidcicloanyo( ) public void setidcicloanyo( Integer idcicloanyo ) public String getDescripcion( ) public void setDescripcion( String descripcion ) public Curso[0..*] getCursoCollection( ) public void setCursoCollection( Curso cursoCollection[0..*] ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre> </td></tr> </tbody> </table>	■ Cicloanyo	<p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idcicloanyo private String descripcion</pre> <p><i>Operations</i></p> <pre>public Cicloanyo( ) public Cicloanyo( Integer idcicloanyo ) public Cicloanyo( Integer idcicloanyo, String descripcion ) public Integer getidcicloanyo( ) public void setidcicloanyo( Integer idcicloanyo ) public String getDescripcion( ) public void setDescripcion( String descripcion ) public Curso[0..*] getCursoCollection( ) public void setCursoCollection( Curso cursoCollection[0..*] ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>
■ Asistencia								
<p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idasistencia</pre> <p><i>Operations</i></p> <pre>public Asistencia( ) public Asistencia( Integer idasistencia ) public Integer getidasistencia( ) public void setidasistencia( Integer idasistencia ) public Clase getidclase( ) public void setidclase( Clase idclase ) public Existencia getidequipoexistente( ) public void setidequipoexistente( Existencia idequipoexistente ) public Estudiante getidestudiante( ) public void setidestudiante( Estudiante idestudiante ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>								
■ Reserva								
<p><i>Attributes</i></p> <pre>private Integer idreserva private Date fechareserva private Date fechahorainicioprestamo private Date fechahorafinprestamo private String descripcion</pre> <p><i>Operations</i></p> <pre>public Reserva( ) public Reserva( Integer idreserva ) public Integer getidreserva( ) public void setidreserva( Integer idreserva ) public Date getFechareserva( ) public void setFechareserva( Date fechareserva ) public Date getFechahorainicioprestamo( ) public void setFechahorainicioprestamo( Date fechahorainicioprestamo ) public Date getFechahorafinprestamo( ) public void setFechahorafinprestamo( Date fechahorafinprestamo ) public String getDescripcion( ) public void setDescripcion( String descripcion ) public Estadoreserva getdestado( ) public void setdestado( Estadoreserva idestado ) public Existencia getidequipoexistente( ) public void setidequipoexistente( Existencia idequipoexistente ) public void setidubicacion( Ubicacion idubicacion ) public Usuario getidusuario( ) public void setidusuario( Usuario idusuario ) public Docente getiddocente( ) public void setiddocente( Docente iddocente ) public boolean equals( Object object ) public String toString( ) public String getfechaReserva( ) public String getthoraInicio( ) public String getthoraFinal( )</pre>								
■ Cicloanyo								
<p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idcicloanyo private String descripcion</pre> <p><i>Operations</i></p> <pre>public Cicloanyo( ) public Cicloanyo( Integer idcicloanyo ) public Cicloanyo( Integer idcicloanyo, String descripcion ) public Integer getidcicloanyo( ) public void setidcicloanyo( Integer idcicloanyo ) public String getDescripcion( ) public void setDescripcion( String descripcion ) public Curso[0..*] getCursoCollection( ) public void setCursoCollection( Curso cursoCollection[0..*] ) public int hashCode( ) public boolean equals( Object object ) public String toString( )</pre>								

<p><b>Entrada</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer identrada private String titulo private String descripcion private Date fechahora</pre> <p><i>Operations</i></p> <pre>public Entrada() public Entrada(Integer identrada) public Integer getIdentrada() public void setIdentrada(Integer identrada) public String getTitulo() public void setTitulo(String titulo) public String getDescripcion() public void setDescripcion(String descripcion) public Date getFechahora() public void setFechahora(Date fechahora) public Comentarios[0..*] getComentariosCollection() public Comentarios[0..*] getComentariosList() public Usuario getIdUsuario() public void setIdUsuario(Usuario idUsuario) public int hashCode() public boolean equals(Object object) public String toString() public String getFecha()</pre>	<p><b>Existencia</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idexistencia private String codigo</pre> <p><i>Operations</i></p> <pre>public Existencia() public Existencia(Integer idexistencia) public Existencia(Integer idexistencia, String codigo) public Integer getIdexistencia() public void setIdexistencia(Integer idexistencia) public String getCodigo() public void setCodigo(String codigo) public Equipo getIdHardware() public void setIdHardware(Equipo idHardware) public Ubicacion getIdUbicacion() public void setIdUbicacion(Ubicacion idUbicacion) public EstadoEquipo getIdEstado() public void setIdEstado(EstadoEquipo idEstado) public Solicitud[0..*] getSolicitudCollection() public Asistencia[0..*] getAsistenciaCollection() public Reserva[0..*] getReservaCollection() public void setReservaCollection(Reserva reservaCollection[0..*]) public Accesorio[0..*] getAccesorioCollection() public Pieza[0..*] getPiezaCollection() public void setPiezaCollection(Pieza piezaCollection[0..*]) public int hashCode() public boolean equals(Object object) public String toString() public String getImagenEstado() public String getNombreTecnicoActualMnto()</pre>	<p><b>Accesorio</b></p> <p><i>Attributes</i></p> <pre>private long serialVersionUID = 1L private Integer idaccesorio private String nombre private String modelo</pre> <p><i>Operations</i></p> <pre>public Accesorio() public Accesorio(Integer idaccesorio) public Accesorio(Integer idaccesorio, String nombre, String modelo) public Integer getIdaccesorio() public void setIdaccesorio(Integer idaccesorio) public String getNombre() public void setNombre(String nombre) public String getModelo() public void setModelo(String modelo) public Marca getIdMarca() public void setIdMarca(Marca idMarca) public Clasificacion getIdClasificacion() public void setIdClasificacion(Clasificacion idClasificacion) public Existencia getIdExistencia() public void setIdExistencia(Existencia idExistencia) public AtributoHardware[0..*] getAtributoHardwareCollection() public int hashCode() public boolean equals(Object object) public String toString()</pre>
---	--	--

<b>Usuario</b>	
<i>Attributes</i>	
private long serialVersionUID = 1L	
private Integer idusuario	
private String nombre	
private String clave	
<i>Operations</i>	
public Usuario( )	
public Usuario( Integer idusuario )	
public Usuario( Integer idusuario, String nombre, String clave )	
public Integer getIdusuario( )	
public void setIdusuario( Integer idusuario )	
public String getNombre( )	
public void setNombre( String nombre )	
public String getClave( )	
public void setClave( String clave )	
public Instructor[0..*] getInstructorCollection( )	
public Instructor[0..*] getInstructorList( )	
public Rol getIdrol( )	
public void setIdrol( Rol idrol )	
public Autorizacion getIdautorizacion( )	
public void setIdautorizacion( Autorizacion idautorizacion )	
public Estudiante[0..*] getEstudianteList( )	
public Docente[0..*] getDocenteList( )	
public int hashCode( )	
public boolean equals( Object object )	
public String toString( )	

<b>Software</b>	
<i>Attributes</i>	
private Integer idsoftware	
private String nombre	
private String version	
private String codigolicencia	
private Integer cantidadlicencias	
<i>Operations</i>	
public Software( )	
public Software( Integer idsoftware )	
public Software( Integer idsoftware, String nombre, String version )	
public Integer getIdsoftware( )	
public void setIdsoftware( Integer idsoftware )	
public String getNombre( )	
public void setNombre( String nombre )	
public String getVersion( )	
public void setVersion( String version )	
public String getCodigolicencia( )	
public void setCodigolicencia( String codigolicencia )	
public Integer getCantidadlicencias( )	
public void setCantidadlicencias( Integer cantidadlicencias )	
public Clasificacion getIdclasificacion( )	
public void setIdclasificacion( Clasificacion idclasificacion )	
public int hashCode( )	
public boolean equals( Object object )	
public String toString( )	

<b>Articulos</b>	
<i>Attributes</i>	
private Integer idarticulo	
private String titulo	
private String descripcion	
private Date fechahora	
<i>Operations</i>	
public Articulos( )	
public Articulos( Integer idarticulo )	
public Integer getIdarticulo( )	
public void setIdarticulo( Integer idarticulo )	
public String getTitulo( )	
public void setTitulo( String titulo )	
public String getDescripcion( )	
public String getDescripcionCorta( )	
public void setDescripcion( String descripcion )	
public Date getFechahora( )	
public void setFechahora( Date fechahora )	
public Usuario getIdusuario( )	
public void setIdusuario( Usuario idusuario )	
public String getFecha( )	
public int hashCode( )	
public boolean equals( Object object )	
public String toString( )	

<b>Clasificacion</b>
<i>Attributes</i>
private long serialVersionUID = 1L
private Integer idclasificacion
private String nombre
private String descripcion
private Integer idsuperior
<i>Operations</i>
public Clasificacion( )
public Clasificacion( Integer idclasificacion )
public Clasificacion( Integer idclasificacion, String nombre )
public Integer getIdclasificacion( )
public void setIdclasificacion( Integer idclasificacion )
public String getNombre( )
public void setNombre( String nombre )
public String getDescripcion( )
public void setDescripcion( String descripcion )
public Integer getIdsuperior( )
public void setIdsuperior( Integer idsuperior )
public Software[0..*] getSoftwareCollection( )
public Equipo[0..*] getEquipoCollection( )
public Pieza[0..*] getPiezaCollection( )
public int hashCode( )
public boolean equals( Object object )
public String toString( )

<b>Tecnico</b>
<i>Attributes</i>
private long serialVersionUID = 1L
private Integer idtecnico
private String apellidos
private String nombres
private String cargo
<i>Operations</i>
public Tecnico( )
public Tecnico( Integer idtecnico )
public Integer getIdtecnico( )
public void setIdtecnico( Integer idtecnico )
public String getApellidos( )
public void setApellidos( String apellidos )
public String getNombres( )
public void setNombres( String nombres )
public String getCargo( )
public void setCargo( String cargo )
public int hashCode( )
public boolean equals( Object object )
public String toString( )

<b>Inscripcion</b>
<i>Attributes</i>
private long serialVersionUID = 1L
private Integer idinscripcion
<i>Operations</i>
public Inscripcion( )
public Inscripcion( Integer idinscripcion )
public Integer getIdinscripcion( )
public void setIdinscripcion( Integer idinscripcion )
public Curso getIdcurso( )
public void setIdcurso( Curso idcurso )
public Estudiante getIdestudiante( )
public void setIdestudiante( Estudiante idestudiante )
public int hashCode( )
public boolean equals( Object object )
public String toString( )

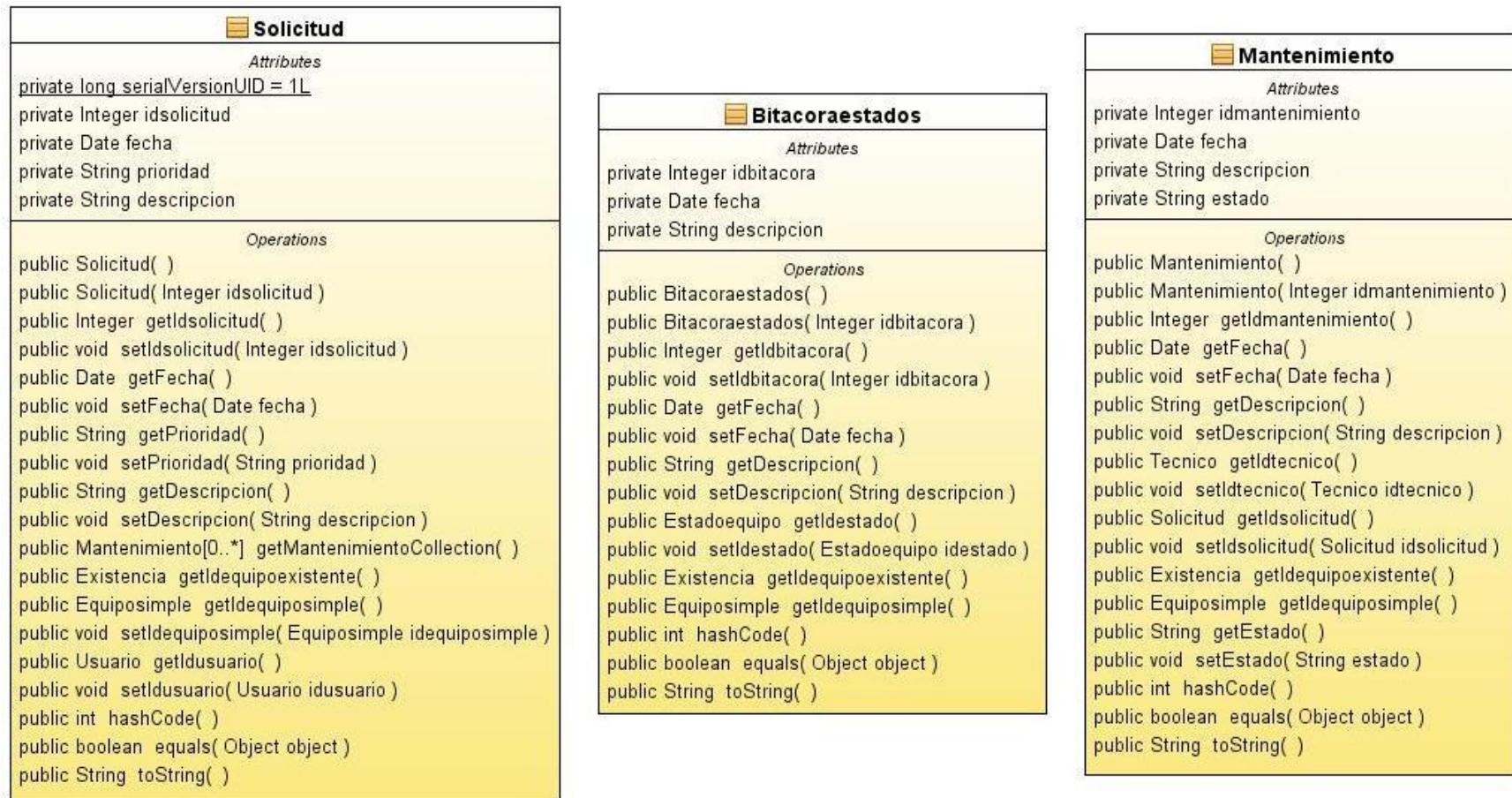


Figura 4.8: Diagrama de Clases de JHard

Tabla Resumen de Clases JHard

<b>Clases de JHard</b>
Accesorio
Administrador
Asistencia
Atributohardware
Autorizacion
Bitacoracambiosusuario
Bitacoraestados
Carrera
Cicloanyio
Clase
Clasificacion
Comentarios
Curso
Docente
Entrada
Equipo
Equiposimple
Estadocurso
Estadoequipo
Estadoreserva
Estudiante
Existencia
Facultad
Horario
Inscripcion
Instalacion
Instructor
Mantenimiento
Marca
Materia
Pieza
Reserva
Rol
Software
Solicitud
Tag
TagEntrada
Tecnico
Ubicación
Usuario

Tabla 4.1: Tabla de Clases de JHard

## Diseño de la Capa de Negocios

**JavaServer Faces** (JSF por sus siglas en inglés) es una tecnología para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. **JSF** usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL<sup>21</sup>.

JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos librerías de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

Basándonos en la estructura de JSF, la capa de negocios está escrita en *Managed Beans*, de *sesión* (session) o de *p petición* (request).

JHard está basado en el modelo Vista-Controlador (MVC)<sup>22</sup> que nos permite separar la lógica de control (qué cosas hay que hacer pero no cómo), la lógica del modelo (cuál es la

---

<sup>21</sup> XUL (acrónimo de *XML-based User-interface Language*, lenguaje basado en XML para la interfaz de usuario) es la aplicación de XML a la descripción de la interfaz de usuario en el navegador Mozilla.

<sup>22</sup> El **Modelo Vista Controlador (MVC)** es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, tal y como JHard, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

información a mostrar y manipular) y la lógica de presentación (cómo interactuar con el usuario).

Utilizando este tipo de patrón es posible conseguir más calidad, un mantenimiento más fácil, etc. La arquitectura del patrón MVC, adaptado al contexto de una aplicación J2EE se muestra en la siguiente figura:

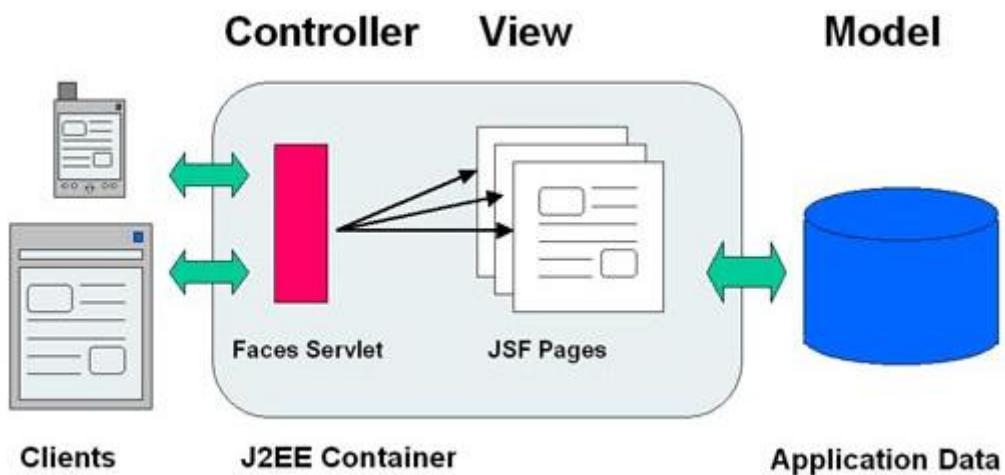


Figura 4.9: Arquitectura MVC

Además, este modelo de arquitectura presenta otras importantes ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- Hay una API muy bien definida; cualquiera que use la API, podrá reemplazar el modelo, la vista o el controlador, sin demasiada dificultad.
- La conexión entre el modelo y sus vistas (ya que puede haber varias) es dinámica: se produce en tiempo de ejecución, no en tiempo de compilación.

Java Server Faces (JSF por sus siglas en inglés) se encarga de conectar la vista y el modelo. Una etiqueta JSP (propia de la vista) puede ligarse a un atributo de un *bean* propio del modelo. De esta manera, JSF establece el vínculo de enlace entre vista y modelo.

Al margen de todo esto, una de las cosas más importantes que permite el uso de este patrón consiste en normalizar y estandarizar el desarrollo de Software, ya que dejamos una vista diseñada y la capa de negocios queda escrita en *JavaBeans*.

Un apartado importante en el diseño JHard como aplicación web es la separación de la presentación y la lógica de negocio. JSF usa *beans* para lograr esta separación. Las páginas JSF se refieren a las propiedades del *bean*, y la lógica del programa está contenida en el código de implementación del *bean*. Los beans son fundamentales para programar JSF. Además, cada una de las acciones ejecutadas por los eventos de las páginas están manejadas por métodos del bean.

Los **JavaBeans** son un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java.

Se usan para encapsular varios objetos en un único objeto (conocido también en el ámbito del desarrollo como “la vaina”), para hacer uso de un sólo objeto en lugar de varios más simples.

La especificación oficial de JavaBeans de Sun Microsystems los define como *“componentes de software reutilizables que se puedan manipular visualmente en una herramienta de construcción”*<sup>23</sup>.

Sun da otras convenciones con respecto a los JavaBeans, entre las cuales están:

- Debe tener un constructor sin argumentos.
- Sus propiedades deben ser accesibles mediante métodos get y set que siguen una convención de nomenclatura estándar.
- Debe ser serializable.

---

<sup>23</sup> <http://java.sun.com/javase/technologies/desktop/javabeans/index.jsp>

A primera vista, un *bean* parece ser similar a cualquier otro objeto. Sin embargo, los *beans* se manejan de una forma más concreta. Cualquier objeto se crea y se manipula dentro de un programa Java llamando a los constructores e invocando a los métodos. Sin embargo, los *beans* pueden ser configurados y manipulados sin programar, a través de entornos de trabajo (*frameworks*) o entornos de desarrollo integrados (*IDE-Integrated Development Environment*), que los utilizan mediante técnicas de introspección.

En el contexto de JavaServer Faces, los *beans* no se utilizan para nada relacionado con la interfaz de usuario: los *beans* se utilizan cuando se necesita conectar las clases Java con páginas web o archivos de configuración.

Cada uno de los módulos de JHard posee su propio Bean con operaciones de CRUD y otras básicas de negocio para el funcionamiento óptimo de cada módulo. Los beans (denominados por los desarrolladores de JHard como BeanBase) son los siguientes:

Módulo	BeanBase
EntityManager	BeanBase.java
JHardmin	BeanBaseJHardmin.java
JRequest	BeanBaseJRequest.java
JInvent	BeanBaseJInvent.java
ManLab	BeanBaseManLab.java
JCanon	BeanBaseJCanon.java
JWiki	BeanBaseJWiki.java

Tabla 4.2: Tabla de beans de negocios de JHard

Cada uno de los BeanBase de los módulos, heredan de BeanBase.java, que es el que brinda los métodos necesarios para obtener el *EntityManager* de la persistencia.

Entre las atribuciones que tienen los métodos de estas clases están:

- Registrar nuevas instancias de objetos de cada módulo en la persistencia e ingresarlas a la base de datos.

- Actualizar instancias de objetos de cada módulo en la persistencia y actualizarlas a la base de datos.
- Obtener instancias o arreglos de instancias de objetos de cada módulo de la persistencia para poder manipularlas en el contexto de la aplicación.
- Eliminar instancias de objetos de cada módulo en la persistencia y borrarlas de la base de datos.
- Obtener objetos a partir de registros de la base de datos mediante claves primarias.
- Algunas reglas de negocio básicas.

## Desarrollo de módulo de inventarios (JInvent)

JInvent es el módulo de JHard encargado de administrar el inventario de la Unidad de Hardware y software de la UES-FMO en todo lo que son bienes no consumibles, es decir que dicho módulo solamente está capacitado para administrar activos fijos.

La Unidad de Hardware y Software, entre muchas otras, tiene dos grandes funciones: atender las peticiones de soporte técnico de toda la Facultad y administrar el Laboratorio de Cómputo del Departamento de Ingeniería y Arquitectura.

Gracias a ésta última función es que la Unidad tiene a su cargo mucho activo fijo, compuesto en su gran mayoría por hardware, herramientas y mobiliario. Por mencionar algunos:

- Computadoras completas para prácticas
- Impresoras
- Muebles
- UPS
- Scanners
- Discos Duros

- Memorias RAM
- Laptops
- Etc.

Es por ello que dicho módulo está entre los 3 más importantes de JHard (junto con JRequest y ManLab) por las atribuciones de la Unidad dentro de la Universidad.

Hablando estrictamente del negocio<sup>24</sup> de esta sección, básicamente se trata de un CRUD de equipos, piezas, accesorios, hardware, software y demás elementos que forman parte del inventario físico real de la Unidad.

Como se pudo apreciar en el diagrama ER para JInvent (ver figura 4.4) el modelo que se siguió para abstraer al equipo que se tiene en inventario y el número de existencias es el siguiente:

- Cada registro o tupla en la tabla equipo representa un tipo de equipo hardware que posean en la Unidad, ya sean computadoras desktop, laptops, cañones, scanners, impresoras, etc.
- Cada equipo tiene un nombre de modelo propio
- Cada equipo está asociado a una marca dentro de dicha tabla
- Cada equipo tiene una clasificación.
- Cada registro en la tabla clasificación representa un tipo de hardware que puede albergar la Unidad. La clasificación se compone de una serie de tuplas, todas relacionadas jerárquicamente dependiendo todas de una clasificación padre. Las relaciones entre clasificaciones fijas (que no se podrán eliminar):
  - General
    - Hardware
      - Equipos
        - Desktops

---

<sup>24</sup> El negocio es lo que respecta a la capa de negocios en un modelo de programación en un modelo de aplicación de tres capas

- Laptops
- Accesorios
  - Impresoras
  - Proyectores
- Piezas
- Herramientas de mantenimiento
- Dispositivos de red
- Software
  - Sistemas operativos
  - Utilerías
  - Herramientas didácticas (Netbeans, Autocad, etc.)
- Se podrán agregar nuevas categorías en cualquiera de los niveles anteriormente descritos, pero las mencionadas no se podrán eliminar.
- Cada equipo puede tener  $n$  existencias. Cada registro de la tabla existencia es cada uno de los equipos físicos representados en la tabla Equipo. Un equipo puede tener muchas existencias. Por ejemplo, si la Universidad adquirió para el Departamento de Ingeniería 20 computadoras **Dell Vostro 220 Mini Tower**, habrá un registro en la tabla equipo que haga mención a dicha computadora, y 20 registros en la tabla existencia, con relación con la tupla en la tabla equipo que sea para las **Dell Vostro 220 Mini Tower**. Todas las tuplas de existencia llevarán su respectivo código que las diferencie.
- Una existencia también puede tener asociadas piezas y accesorios. Relaciones con las respectivas tablas en la base de datos
- Una existencia también tiene relación con las instalaciones de software que se le realizan.
- Una existencia posee además una ubicación física dentro del campus, lo cual implica una relación a un registro de la tabla Ubicaciones.

Por lo tanto, el Managed Bean para JInvent tiene las siguientes reglas de negocio:

- Agregar, eliminar y actualizar equipos
- Agregar, eliminar y actualizar piezas
- Agregar, eliminar y actualizar accesorios
- Agregar, eliminar y actualizar existencias
- Agregar, eliminar y actualizar clasificaciones que sean agregadas por el usuario, no las definidas por JHard
- Agregar, eliminar y actualizar software e instalaciones del mismo
- Mostrar listas de equipos, existencias, piezas, accesorios y software
- Obtener el estado de una existencia específica
- Obtener las ubicaciones de toda la Facultad
- Reportar una existencia como fallida

## Desarrollo de módulo de Peticiones de servicios (JRequest)

JRequest desempeña una labor fundamental dentro de las funciones de JHard: además que es el módulo que será más utilizado por los usuarios administrativos y docentes, cumple una de las funciones más importantes de la Unidad de Hardware y Software: recibir y paliar todas las solicitudes de soporte técnico para el equipo informático de la Facultad Multidisciplinaria de Occidente.

Las reglas de negocio de JRequest:

- Un usuario registrado previamente en el sistema puede emitir una solicitud para soporte técnico ya sea para hardware o software.
- Una solicitud se realiza hacia un Equipo simple. Un equipo simple es una computadora de la Facultad Multidisciplinaria de Occidente que no forma parte del inventario real de la Unidad de Hardware y Software, pero que requiere de mantenimiento, ya sea preventivo o correctivo.
- Un equipo simple puede tener muchas solicitudes.

- Las solicitudes se encolan, y solo el administrador de la Unidad puede observarlas. Él, con su criterio, decide cuáles son las que se atenderán primero y les asigna técnicos para que las solventen. Una solicitud desencadena un mantenimiento. Un mantenimiento va amarrado a un equipo simple y una solicitud. Un equipo simple puede tener muchos mantenimientos.
- Cuando los técnicos que laboran de turno en la Unidad resuelven el problema, entonces el administrador cambia el estado del mantenimiento a “Finalizado”. Cuando hace esto escribe lo que se le realizó a la computadora para solventar el impasse. Con esto se genera una Bitácora de estados. Un equipo simple tiene muchas bitácoras.
- Las bitácoras son totalmente modificables. No pueden ser eliminadas por razones de seguridad y para guardar siempre el registro de cada máquina.
- A cada equipo simple, se le puede actualizar el estado del equipo. Un equipo posee una clave para un estado. Las reglas de negocio estipulan que existen cuatro posibles estados para un equipo:
  - Excelentes condiciones
  - En reparación
  - Dañado
  - Irreparable
- Se realizan actividades de mantenimientos (crear y actualizar únicamente) a equipos simples, solicitudes, y técnicos.

## Desarrollo de módulo de Equipo multimedia (JCanon)

La Unidad de Hardware y Software tiene a su cargo diverso equipo multimedia utilizado, en la mayor parte del tiempo, en las clases expositivas del Departamento de Ingeniería. Los resultados de las encuestas arrojaron que el procedimiento para la reserva de equipo multimedia (laptops y proyectores) es poco eficiente y deberían de ser automatizados y más amigables. La respuesta de JHard ante esta problemática es JCanon: el módulo para el

manejo y administración de la reserva de equipo multimedia del Departamento de Ingeniería y Arquitectura y que implementa una interfaz novedosa para la visualización de la jornalización del préstamo del equipo.

JCanon es el módulo con el menor número de tablas, por lo tanto las labores de mantenimiento son menores a las del resto de módulos, sin embargo es muy rico en reglas de negocio.

- Únicamente un usuario registrado podrá realizar una reserva de equipo.
- Únicamente el administrador de JHard podrá ingresar a la página administrativa de reservas de equipo y a otras opciones, como agregar equipo multimedia, posponer reservas y ver reportes.
- Cualquiera que sea el rol del usuario está habilitado para pedir una reserva.
- Se hacen reservas de laptops y proyectores por separado, aunque en el mismo formulario se puedan reservar ambos para un mismo evento (una clase expositiva o una presentación por ejemplo).
- La disponibilidad de equipo para una determinada fecha y hora está ligado estrictamente al total de equipo en calidad de reservable que no está reservado a en la hora y el día deseados. Esto se realiza con una consulta JPQL así:

```
@NamedQueries({
```

```
//Contar equipo con calidad de reservables
```

```
@NamedQuery (name = "Existencia.contarEquipos", query = "SELECT COUNT(e) FROM Existencia e LEFT JOIN e.idhardware eq WHERE eq.idclasificacion.idclasificacion=:idclasificacion"),
```

```
//Comprobar las reservas que comienzan a la misma hora
```

```

@NamedQuery (name = "Reserva.findMismaHora", query = "SELECT COUNT(r) FROM
Reserva r WHERE r.fechahorainicioprestamo = :fechahorainicioprestamo"),
})

```

- Para el administrador de JHard:
  - Se puede agregar equipo multimedia (cañones y proyectores)
  - Ver Reportes
  - Posponer reservas
  - Cambiar el estado de la reserva: las reservas pueden tener 2 posibles estados:
    - **Pendiente:** Aun no ha llegado la fecha y hora para que se retire el equipo
    - **En Uso:** El período de reserva se está llevando a cabo.
    - **Despachada:** El período de reserva ha culminado y el equipo ha sido entregado satisfactoriamente.
  - Eliminar reservas.

## Desarrollo de módulo de seguridad (JHardmin)

El módulo de JHardmin es de carácter administrativo y de seguridad. Es el encargado de manejar las sesiones de los usuarios loggeados al sistema, así como de realizar tareas administrativas a los usuarios de JHard.

- Maneja el login de los usuarios que intentan acceder a JHard. Lo realiza con el siguiente método del LoginManager:

```
/*
```

\* Agrega el usuario a la lista de usuarios logueados, si este existe en la base de datos y si no estaba logueado antes

```
*/
```

```

public synchronized int Login(String userName, String userPwd, String url){
    if(!this.isLoggedIn(userName)){
        System.out.println("Encriptando... ");
        if(this.existsInBD(userName, userPwd)){
            Usuario usr = new BeanBaseJHardmin().getUsuario(userName, encrypt(userPwd));
            this.loggedUsers.put(usr.getIdusuario(), new LoggedUser(usr.getIdusuario(),
            userName, usr.getIdrol(), url));
            return usr.getIdusuario();
        }
    }
    else{
        return this.getUser(userName).getUid();
    }
    return -1;
}

```

- Encripta claves mediante MD5
- Utiliza Managed Beans de sesión para el manejo de usuarios loggeados a JHard en cada máquina. Se declara de esta manera en el faces-config.xml:

```

<managed-bean>
    <managed-bean-name>JHardminInstance</managed-bean-name>
    <managed-bean-class>edu.ues.jhard.beans.BeanBaseJHardmin</managed-bean-
    class>
    <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

```

- Por cada usuario que logra loggearse se crea una instancia de LoggedUser.
- Solo el administrador de JHard puede acceder a las páginas jsp de JHardmin
- El administrador tiene designado:
  - Agregar nuevos usuarios

- Actualizar usuarios
- Eliminar usuarios
- Cambiar la contraseña de acceso de los usuarios existentes
- Asignar roles a los usuarios existentes.
- Crear códigos de autorización para que usuarios anónimos puedan registrarse en el sistema con el rol Estudiante

## Desarrollo del módulo de Administración de grupos de laboratorio (ManLab)

ManLab es un poderoso módulo que administrará los grupos de prácticas para las carreras del Departamento de Ingeniería y Arquitectura de la UES-FMO.

Básicamente está diseñado en 5 grandes áreas, distribuidos en una serie de páginas JSP, cada una validada de acuerdo a jerarquía de roles y usuarios:

- Control de Asistencia
  - ManLab permite controlar la asistencia a todas las clases prácticas que se lleven a cabo dentro del Laboratorio de Hardware.
  - El instructor o un administrador está autorizado para iniciar o habilitar un curso.
  - El mismo instructor o administrador debe finalizar el curso
  - Cuando el curso se habilita, los estudiantes inscritos al mismo y que han asistido a dicha clase, marcan su asistencia.
  - Se generan reportes de asistencia por clases.
  - Se generan reportes de asistencia por el total de laboratorios impartidos por un instructor
- Gestión de Clases
  - La autorización de las clases para un respectivo grupo de laboratorio las realiza esta sección.

- Es estricto con los horarios para activar y desactivar grupos, sin embargo la rigurosidad de este módulo así lo exige.
- Control de Horarios
  - La creación de la manta de horarios por ciclo se hace en este módulo.
  - Se pueden agregar cursos a distintos horarios.
  - Hay completa libertad en la asignación de los horarios a los cursos, así como del número de horarios asignados para un curso.
  - Este módulo es puramente administrativo, por lo tanto el encargado de la Unidad de Hardware y Software tiene completa libertad sobre el mismo.
  - Se muestra también en un popUp aparte la manta de horarios para el ciclo actual. Muestra también la semana actual en la que se desarrolla el ciclo.
  - Las labores de mantenimientos de horarios también se realizan en este módulo. El administrador tiene completa libertad sobre la misma
- Inscripciones
  - Este módulo permite acceder a los estudiantes hacia los distintos cursos a impartirse en determinado ciclo.
  - El administrador habilita cursos para que los estudiantes autorizados puedan ingresar a ellos.
  - El límite de personas en curso lo coloca el administrador
  - Mientras no llega al máximo asignado, los estudiantes se siguen inscribiendo.
  - El administrador da mantenimiento a dicha área.
- Mantenimientos
  - Ciertos ítems necesitan mantenimientos, es decir, labores de agregado, eliminado y actualizado.
  - Objetos como materia, curso, inscripciones, etc. necesitan dichos mantenimientos, y esta área se encarga de realizarlos
  - Área completamente administrativa.

## Desarrollo del módulo de Manejo de Contenidos (JWiki y JProCur)

Estos módulos son una manera innovadora de dar a conocer información necesaria desde el ámbito administrativo/organizativo de la Facultad, así como en conocimientos meramente informáticos.

- Están realizados 100% en Java e ICEfaces.
- Interfaces amigables para el usuario
- Sistema de búsquedas de artículos y entradas sensible a las necesidades del usuario
- Las entradas son para JProCur, que es un CMS puro, y los artículos son para JWiki, que es una Wiki de conocimientos informáticos.
- Administradores y editores de contenido pueden entrar a las partes administrativas de estos módulos.
- Cada usuario con privilegios de escritura tiene entradas y/o artículos asociados, según sea el módulo al que llegue.
- La interfaz para crear nuevas entradas y/o artículos es sumamente amigable al usuario, soporta texto enriquecido y guarda grandes cantidades de texto dentro de la base de datos.
- La navegabilidad entre estos módulos es sencilla.
- La integración entre JRequest y JWiki es completa, ya que desde el módulo de peticiones de servicio se pueden consultar los artículos publicados en JWiki sin ningún problema.
- Cualquier usuario invitado puede accesar a los artículos de JWiki y las entradas de JProCur.

## Diseño de la Interfaz de Usuario general

La interfaz visual de JHard está diseñada en un 95% con el framework ICEfaces, sustituto “natural” del proyecto Woodstock de Sun.

ICEfaces Es un framework de código abierto para construir aplicaciones web con AJAX tipo RIA (Rich Internet Application).

Permite al programador incluir una serie de Ajax-tags en sus JSP o XHTML de tal manera que el código Ajax es generado por el propio framework automáticamente.



Figura 4.9: Logo de ICEfaces

ICEFaces aisla completamente al desarrollador de AJAX. No hacen falta etiquetas especiales: se ponen los controles en la pantalla e ICEFaces se encarga de enviar sólo la información necesaria entre cliente y servidor.

Es decir, ya no se envían los formularios a la antigua usanza, en un POST de HTTP, sino que sólo se envían los cambios que ha hecho el usuario del cliente al servidor, y los cambios en la pantalla del servidor al cliente.

Además, con la inclusión de la librería Scriptaculous en ICEFaces, se dispone de arrastrar+soltar y de efectos (fundidos, parpadeos, apariciones) para los controles.

### ***Características.***

ICEfaces es considerado un framework que integra funcionalidad AJAX y permite a los desarrolladores Java EE crear aplicaciones RIA (Rich Internet Applications) de una manera sencilla.

Las aplicaciones desarrolladas en ICEfaces no necesitan plugins de navegador o applets para ser vistas.

Estas aplicaciones están basadas en JavaServer Faces (JSF), así que permite el desarrollo de aplicaciones Java EE con la posibilidad de utilizar de forma fácil desarrollos basados en JavaScript.

En torno a AJAX han surgido varios frameworks (Prototype, DWR, GWT) que, si bien aportaban facilidad de uso, no acababan de convencer a la comunidad de programadores. Algunos porque sólo eran clientes Javascript, otros porque, si bien integraban la parte de servidor con la de cliente, no eran realmente frameworks, sino librerías de comunicación. Además, no estaba claro cómo juntarlos con la arquitectura JEE.

Con la llegada de JSF, se empezó a vislumbrar posibilidades de integración. Si JSF permitía al desarrollador aislar de la arquitectura web y ver sus aplicaciones como algo parecido a una aplicación de escritorio, debería entonces ser sencillo utilizar AJAX para hacer estos controles más funcionales. Y así fue, empezaron a aparecer AJAX4JSF, ICEFaces, Tobago, etc.

Sin embargo, de estas propuestas, ICEFaces fue una de las más acogidas ya que aisla completamente al desarrollador de AJAX. No hacen falta etiquetas especiales: se ponen los controles en la pantalla e ICEFaces se encarga de enviar entre cliente y servidor sólo la información necesaria.

Se presenta una figura con la arquitectura de una aplicación en JSF integrada con ICEFaces:

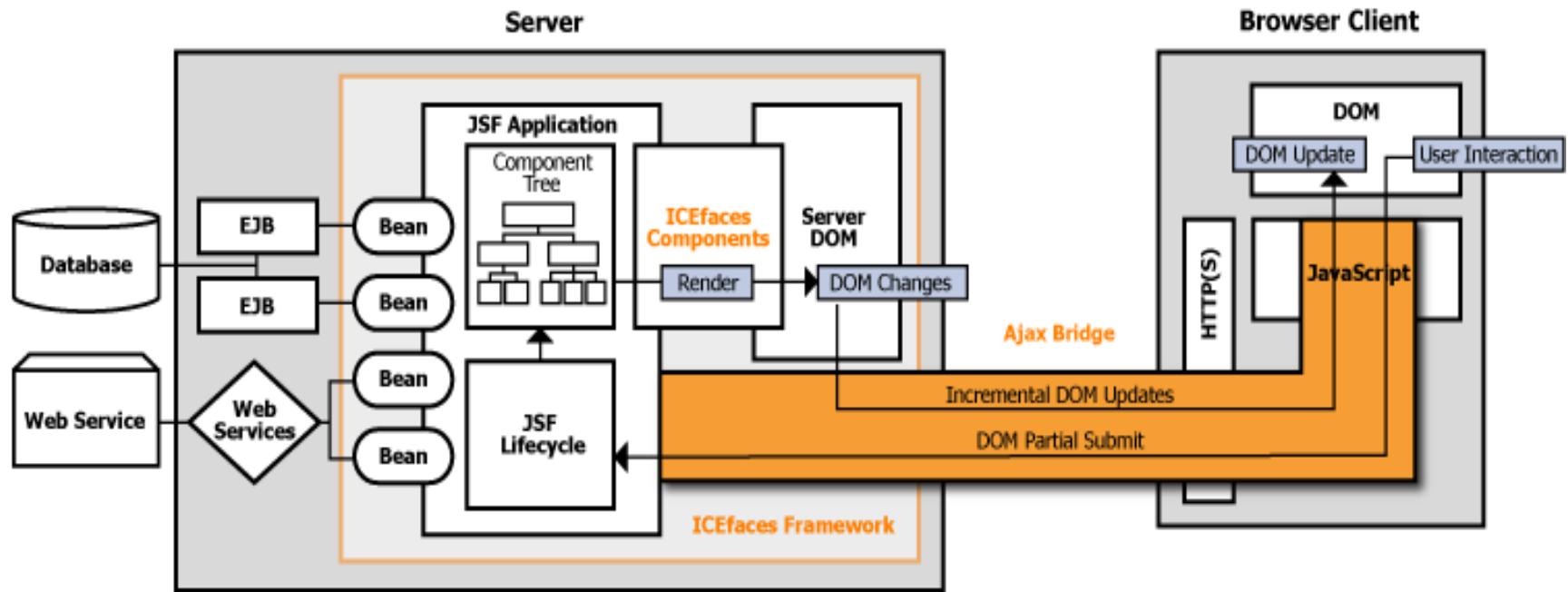


Figura 4.10: Arquitectura general de ICEfaces

Los principales elementos de la arquitectura ICEfaces incluyen:

- Persistent Faces Servlet: Las URLs con extensión ".iface" son mapeadas por el servlet 'Persistent Faces Servlet'. Cuando se realiza una petición de la página inicial en la aplicación, este servlet se hace responsable de la ejecución del ciclo de vida JSF para petición asociada.
- Blocking Servlet: Se encarga de la gestión de todas las peticiones de bloqueo y no-bloqueo después de las primeras páginas.
- D2D ViewHandler: Se encarga de establecer el Direct-to-DOM, incluyendo la inicialización de la 'DOM Respuesta Writer'. El ViewHandler también invoca al Parser para analizar el árbol de componentes JSF en la página inicial.
- Parseador D2D: Responsable del montaje de un componente de documentos JSP. El Parser ejecuta la etiqueta de JSP de procesamiento del ciclo de vida con el fin de crear el árbol, pero lo hace sólo una vez para cada página. La compilación del estándar JSP y el proceso de análisis no es compatible con ICEfaces.
- DOM Response Writer: Se encarga de la escritura en el DOM. También inicia la serialización DOM para la primera prestación, y desbloquea el DOM Updater para actualizaciones incrementales.
- DOM Serializer: Responsable de la serialización del DOM de la página inicial.
- DOM Updater: Se encarga de conjuntar las de las 'DOM mutations' en una única actualización DOM.

- Component Suite: Ofrece un conjunto de componentes 'rich JSF' con influencia AJAX y características del puente, proporcionando los elementos básicos para aplicaciones ICEfaces.
- Client-side AJAX Bridge: Responsable de la actualización DOM en curso generada por la solicitud y la respuesta del proceso. También es el encargado de centrar la gestión y de presentar el proceso.

Jhard cumple los requisitos para poder utilizar ICEfaces.

<b>JHard</b>	
<b>IDE</b>	Netbeans 6.5.1
<b>Servidor de aplicaciones</b>	Apache Tomcat 6.0.18
<b>Framework Java EE</b>	Java Server Faces 1.2
<b>Gestor de base de datos</b>	MySQL 5.1
<b>Framework Ajax</b>	ICEfaces 1.8.0

Tabla 4.3: Tabla resumen de tecnologías de JHard

### Ajax-Push

JHard con ICEfaces 1.8.0 utiliza la tecnología Ajax-Push.

Ajax Push se refiere a un patrón de diseño de Ajax que utiliza conexiones duraderas del HTTP para permitir que la comunicación se mantenga en un estado de baja latencia entre un web server y un cliente. Es una capacidad revolucionaria en el marco de ICEfaces que permite el empuje asincrónico, en Internet de los cambios de la presentación al navegador del cliente basado en acontecimientos del lado del servidor. Básicamente es una manera de enviar datos del cliente al servidor y un mecanismo para “empujar” datos del servidor

de nuevo al navegador del cliente. Esta comunicación del servidor-cliente toma una de dos formas posibles:

Este mecanismo se define en figura 4.11:

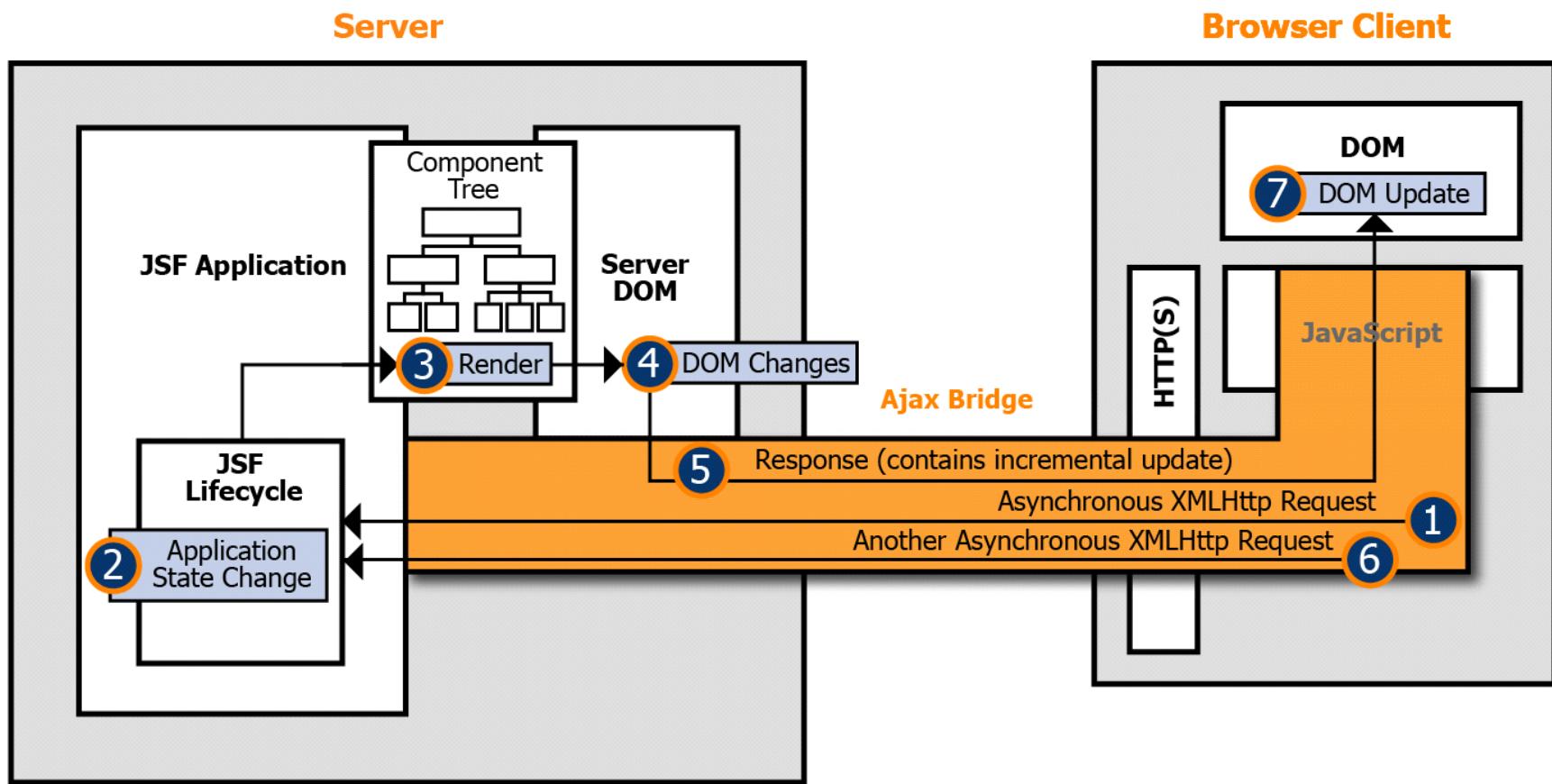


Figura 4.11: Arquitectura de Ajax Push de ICEfaces

De acuerdo con este mecanismo básico ICEfaces ofrece las características siguientes.

- \* Conexión *Heartbeating* (Latidos de corazón) y supervisión: Para mantener conexiones potencialmente duraderas, ICEfaces incluye un mecanismo configurable de latidos de corazón (*Heartbeating*) que supervise activamente la salud y las condiciones de la conexión, y proporciona alarmas cuando se detectan los problemas de conexión.
- \* Distribución de la conexión del navegador: Superando el límite de conexiones de navegadores asociados a la aplicación, ICEfaces facilita la distribución de la conexión del *empuje* entre las vistas múltiples de un mismo navegador, cosa de la cual se aprovecha muy bien JHard. Esto significa que las viñetas o las ventanas múltiples de una misma aplicación trabajarán de la misma manera.
- \* Renderización optimizada: ICEfaces utiliza el ciclo de vida de JSF para forzar la representación de los clientes que requieren una actualización.

#### El modelo de programación

Mientras que el mecanismo subyacente para el Ajax Push es intrincado, el modelo de programación proporcionado en ICEfaces es simple. Los elementos claves del mecanismo de Ajax Push de ICEfaces se describen así:

- \* Render Manager: Es un managed bean de aplicación encargado de la representación que despacha peticiones. Maneja el ciclo vital de JSF. El Render Manager maneja además el registro y la población de los Render Groups.
- \* Render Groups: Es un objeto que se registra con el Render Manager, y se utiliza para organizar los Render Groups (clientes) que recibirán las mismas actualizaciones de Ajax Push.

\* Renderizable: Cualquier bean de petición o de sesión que ejecuta el interfaz de ICEfaces *Renderizable*, puede ser agregado a un Render Group, y participa en los eventos de Ajax Push.

Ajax Push describe la puesta en práctica de cualquiera de estos modelos, incluyendo una combinación de ambos. El patrón del diseño también se conoce como Ajax Reverse, Ajax Full-Duplex y Ajax Streaming.

Cualquier aplicación web donde los usuarios están observando que los datos se refrescan dinámicamente, es porque Ajax Push se encarga de traer esos datos vivos con las actualizaciones instantáneamente. JHard se beneficia de dicha capacidad.

---

### Controles ICEfaces que se utilizan en JHard

---

#### Controles Personales

- **CommandButton**
- **CommandLink**
- **InputText**
- **SelectOneMenu**
- **DataTable**
- **DataPaginator**
- **OutputLabel**
- **SelectDate**
- **SelectOneListBox**

#### Paneles de Diseño

- **PanelCollapsible**
- **PanelDivider**

- 
- **PanelGroup**
  - **PanelPopUp**
  - **PanelGrid**
- 

Tabla 4.4: Controles ICEfaces que implementa JHard

## CAPÍTULO V: Plan de Implementación del Software

### Objetivos del plan de implementación

#### Objetivo General

- Entregar una propuesta de implementación sumamente sencilla y totalmente adaptable a las condiciones reales de la Unidad de Hardware y Software del Departamento de Ingeniería de la UES-FMO.

#### Objetivos Específicos

- Obtener datos precisos del costo de implementación para que cualquier administración ajena a la Unidad de Hardware y Software pueda tener conocimientos del costo de JHard funcionando.
- Crear una lista de procesos que sirvan como instrucciones para que cualquier persona encargada de la Unidad haga que el sistema JHard sea instalado exitosamente sin problemas.
- Solucionar todos los errores que pueda acarrear el desarrollo de JHard al ambiente de producción y que no se puedan percibir en el ambiente de desarrollo.

## Listado de actividades del plan de implementación

Las actividades para la implementación de JHard son las siguientes:

<b>Actividades de la implementación de JHard</b>	
•	<b>Configuración del servidor</b>
○	<b>Instalación y configuración JDK 1.6</b>
•	<b>Configuración de software (cambiar el orden)</b>
•	<b>Instalación de Apache Tomcat 6.0.18</b>
•	<b>Creación de usuarios y demás configuración para Tomcat</b>
•	<b>Instalación de MySQL 5.1</b>
•	<b>Restauración de script SQL de la base de datos de JHard</b>
•	<b>Creación del usuario para la base de datos de JHard</b>
•	<b>Iniciar servidor Apache Tomcat</b>
•	<b>Desplegar la aplicación dentro del servidor</b>

Tabla 5.1: Actividades para implementación de JHard

## Costo de Implementación

Rubro			Costo
<b>Hardware</b>	<b>1</b>	<b>Servidor de Aplicaciones DELL PowerEdge 840</b>	<b>\$899.00</b>
	<b>1</b>	<b>UPS MINUTE MAN Pro 320</b>	<b>\$43.50</b>
<b>Software</b>	<b>1</b>	<b>Apache Tomcat 6.0.18</b>	<b>\$0.00</b>
	<b>1</b>	<b>Java Development Kit 1.6</b>	<b>\$0.00</b>
	<b>1</b>	<b>ICEfaces</b>	<b>\$0.00</b>
	<b>1</b>	<b>MySQL Server 5.1</b>	<b>\$0.00</b>
	<b>1</b>	<b>MySQL Administrator</b>	<b>\$0.00</b>
	<b>1</b>	<b>iReports 3.5</b>	<b>\$0.00</b>

<b>Recurso Humano</b>	<b>3</b>	<b>Ingenieros de Sistemas Informáticos (\$300.00 c/u)</b>	<b>\$900.00</b>
<b>Total</b>			<b>\$1,842.5</b>

Tabla 5.2: Costo de implementación de JHard

## Pruebas Unitarias.

La metodología del ciclo de corrección y pruebas en el sistema, se implementó con Unit Testing, precisamente con la librería de java: JUnit Testing.

Una prueba unitaria es una forma de probar el correcto funcionamiento de una pieza de código. Esto sirve para asegurar que esa pieza de código (ese método) funcione correctamente.

La idea es escribir casos de prueba, que en su salida, producen una respuesta esperada. Cada caso de prueba es independiente.

### Características

Para que una prueba unitaria sea válida se deben cumplir los siguientes requisitos:

- **Automatizable:** no se requiere una intervención manual. Esto es especialmente útil para integración continua.
- **Completas:** deben cubrir la mayor cantidad de código.
- **Repetibles o Reutilizables:** no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. También es útil para *integración continua*.
- **Independientes:** la ejecución de una prueba no debe afectar a la ejecución de otra.

- **Profesionales:** las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.

Aunque estos requisitos no tienen que ser cumplidos al pie de la letra, se recomienda seguirlos o de lo contrario las pruebas pierden parte de su función.

### Ventajas

El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el trozo de código debe satisfacer. Estas pruebas aisladas proporcionan cinco ventajas básicas:

- **Fomentan el cambio:** Las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura (práctica conocida como refactorización), puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.
- **Simplifica la integración:** Puesto que permiten llegar a la fase de integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.
- **Documenta el código:** Las propias pruebas son documentación del código puesto que ahí se puede ver cómo utilizarlo.
- **Los errores están más acotados y son más fáciles de localizar:** dado que tenemos pruebas unitarias que pueden desenmascararlos.

JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

JUnit es también un medio de controlar las pruebas de regresión, necesarias cuando una parte del código ha sido modificado y se desea ver que el nuevo código cumple con los requerimientos anteriores y que no se ha alterado su funcionalidad después de la nueva modificación.

JHard implementa JUnit completamente. Todos se encuentran en el paquete de tests del proyecto, que es un espejo del sistema de paquetes del código fuente.

Algunas pruebas unitarias de JHard son las siguientes:

```
public class BeanBaseJWikiTest {  
  
    public BeanBaseJWikiTest() {  
    }  
  
    /**  
     * Obtenemos una entrada de la BD  
     */  
    @Test  
    public void testGetEntrada() {  
        System.out.println("getEntrada");  
        BeanBaseJWiki instance = new BeanBaseJWiki();  
        Entrada result = instance.getEntrada(1);  
        assertNotNull(result);  
        System.out.println("Exito en la prueba!");  
    }  
  
    /**  
     * Obtejemos N entradas de la BD  
     */  
    @Test  
    public void testNEntradas() {  
        int n = 2;  
        BeanBaseJWiki instance = new BeanBaseJWiki();  
        List<Entrada> resultado = instance.getUltimasNEntradas(n);  
        if(resultado.size()==n)  
            assertNotNull(resultado.get(0)); //por lo menos, que la primera NO sea null  
        else  
            fail("resultado menor que el esperado (" +n+ ")");  
    }  
}
```

```
@Test
public void testgetAllEntradas() {
    BeanBaseJWiki instance = new BeanBaseJWiki();
    List<Entrada> resultado = instance.getAllEntradas();
    assertNotNull(resultado);
}

/**
 * Obtenemos los comentarios asociados a una entrada
 */
@Test
public void testComentariosDeEntrada() {
    int identrada = 1;
    BeanBaseJWiki instance = new BeanBaseJWiki();
    Entrada resultado = instance.getEntrada(identrada);
    Collection<Comentarios> comentarios = resultado.getComentariosCollection();
    System.out.println("Comentarios asociados con la Entrada: " + resultado.getTitulo());
    for (Comentarios c : comentarios) {
        System.out.println(c.getComentario());
    }
    assertTrue(comentarios.size()>0);
}

/**
 * Obtiene las etiquetas asociadas a una entrada
 */
@Test
public void testEtiquetasDeEntrada() {
    int identrada = 1;
    BeanBaseJWiki instance = new BeanBaseJWiki();
    Entrada entrada = instance.getEntrada(identrada);
    Collection<Tag> resultado= instance.getEtiquetas(entrada);
    System.out.println("Tags asociados con la Entrada: " + entrada.getTitulo());
    for (Tag t : resultado) {
        System.out.println(t.getDescripcion());
    }
    assertNotNull(resultado);
}

/**
 * Busqueda por titulo de entrada
 */
@Test
public void testSearchEntradaPorTitulo() {
    String criteria = "ulo 4";
    BeanBaseJWiki instance = new BeanBaseJWiki();
    List<Entrada> e = (List<Entrada>)instance.searchEntradaPorTitulo(criteria);
    assertNotNull(e);
}
```

```
/*
 * Agregamos una Entrada a la BD
 */
@Test
public void testRegistrarEntrada() {
    BeanBaseJHardmin hardmin = new BeanBaseJHardmin();
    Usuario usuario = hardmin.getUsuario(1);
    Entrada entrada = new Entrada(9999, "La entrada 9999", "nain, nine, nueve, 9, iiiiiii, etc etc etc", new Date(2009, 9, 9), usuario);
    BeanBaseJWiki instance = new BeanBaseJWiki();
    if(!instance.createEntrada(entrada)) fail("fallo en REGISTRAR ENTRADA");
}

/*
 * Agregamos un comentario a la BD
 */
@Test
public void testRegistrarComentario() {
    Comentarios comentario = new Comentarios(9999, "Oh!!! este es el comentario para la entrada 9999!!!", new Date(2009, 9, 9), "TEST", true);
    BeanBaseJWiki instance = new BeanBaseJWiki();
    if(!instance.createComentario(9999, comentario)) fail("fallo en REGISTRAR COMENTARIO");
}

/*
 * @author HugoL
 */
public class BeanBaseJRequestTest {

    @Test
    public void testGetSolicitudByPrioridad() {
        System.out.println("getSolicitudByPrioridad");
        String prioridad = "Alta";
        BeanBaseJRequest instance = new BeanBaseJRequest();
        Solicitud result = instance.getSolicitudByPrioridad(prioridad);
        assertNotNull(result);
        System.out.println("EXISTE LA PRIORIDAD ALTA!!!");
    }
}
```

```
@Test  
public void testGetSolicitudesByPrioridad() {  
    System.out.println("getSolicitudesByPrioridad");  
    BeanBaseJRequest instance = new BeanBaseJRequest();  
    String prioridad="Media";  
    Solicitud[] result = instance.getSolicitudesByPrioridad(prioridad);  
    assertNotNull(result);  
    System.out.println(result);  
    System.out.println("OBTENGO TODAS LAS SOLICITUDES DE UNA PRIORIDAD ");  
}  
}
```

## CAPITULO VI: Documentación del Software:

### Descripción de pantallas y Manual de Programador

Las principales pantallas de JHard son las siguientes:

#### Página Principal

The screenshot shows the main interface of the JHard software. At the top, there is a navigation bar with links: Index, Mantenimientos, Laboratorios, Inventario, Cursos, Wiki, and Catálogos. Below the navigation bar, the title "Laboratorio de Hardware" is displayed, along with the subtitle "Universidad de El Salvador, Facultad Multidisciplinaria de Occidente". On the left side, there is a sidebar titled "Sesión" which shows "Bienvenido usuario hugol" and a "Logout" button. Another sidebar titled "Tareas Comunes" lists various administrative tasks such as "Cambiar clave de acceso", "Administración de Usuarios y Permisos", "Ver Reportes de JHard", "Administración de Solicitudes de JRequest", "Tareas Administrativas de JRequest", "Emitir Solicitud de Soporte Técnico", "Administración de Reserva de Equipo", and "Multimedia". The main content area is titled "Página Principal" and contains three sections: "Wiki" (with links to "Últimos 5 artículos del wiki", "¿Qué es un Wiki?", "¿Qué es JHard?", and "¿Cómo saco el dispositivo USB del ordenador?"), "Existencias en estado fallido" (listing items like "codigo de barras - Cañon Epson Epson", "codigo de barras - Cañon Epson Epson", "codigo de barras - Laptop Toshiba Satellite", and "codigo de barras - Cañon Epson ProView"), and "Equipos simples [total: 4]" (listing items like "Computadora Carmencita - Carmencita: Virus", "Compu Medicina - Carmencita.", "Computadora Arq Centeno - Carmencita: No sirve el monitor", and "Big Boss - Daddy Yankee"). At the bottom of the page, there is a footer with the text: "Universidad de El Salvador Facultad Multidisciplinaria de Occidente", "Todos los Derechos (C) Reservados - Teléfonos: (503)2449-0349, Fax:(503)2449-0352 Apdo. 1908", and "Créditos - Página Principal de la UES FMOcS".

### Descripción:

- En la parte principal de la página presenta un *dashboard*<sup>25</sup> para obtener información detallada y consolidada más importante de cada módulo.
- Muestra los últimos artículos de la Wiki
- Las existencias y equipos de la Facultad con estado fallido
- Los equipos que se encuentran actualmente en mantenimiento
- Las reservas de equipo multimedia realizadas para el día actual
- En la sección de “Tareas Comunes” se muestran links hacia tareas u opciones de todos los módulos de JHard, mostradas, claro está, de acuerdo a la jerarquía de usuario, siendo el administrador el que más privilegios goza.

### Página de JHardmin

The screenshot shows the JHardmin application interface. At the top, there is a navigation bar with links: Index, Mantenimientos, Laboratorios, Inventario, Cursos, Wiki, and Canones. On the right side of the header is a logo for "Universidad de El Salvador, Facultad Multidisciplinaria de Occidente".

The main content area has several sections:

- Sesión:** Shows "Bienvenido usuario hugol" and a "Logout" button.
- Tareas Comunes:** A sidebar with links: Cambiar clave de acceso, Administrar Solicitudes de JRequest, Tareas Administrativas de JRequest, and Emitir Solicitud de Soporte Técnico.
- Administración de Usuarios:**
  - Mi Cuenta:** Displays account details: Usuario: hugol, Rol: Administrador, Carnet: BP04004, Nombres: Hugo Alejandro, Apellidos: Barrientos Padilla. It also includes fields for changing the password: Clave anterior, Nueva clave, Repita la nueva clave, and a "Cambiar clave" button.
  - Usuarios:** A table showing a list of users. The columns are Nombre, Rol, and Autorización. The table lists 13 users, including "hugol" (Administrador) and "Karilita" (Administrativo). Each row has "Editar" and "Eliminar" buttons.
  - A pagination bar at the bottom of the users section shows pages 1 through 5.
  - A "Agregar usuario" button is located at the bottom of the users section.

<sup>25</sup> Del inglés “tablero de instrumentos”. Dicho vocablo se utiliza en varias aplicaciones web, sobre todo las de Google como Google Analytics o Blogger, en donde se le muestran al usuario los “instrumentos” que más suele utilizar para que los tenga a un alcance más rápido

## Descripción

- Es la página administrativa de JHard.
- Permite cambiar la clave de acceso para cualquier usuario
- Muestra los datos de la cuenta que se encuentra logeada
- Muestra la lista de usuarios registrados al sistema, y permite editarlos o suprimirlos, si el usuario que ha ingresado al sistema es de tipo Administrador
- Muestra las autorizaciones que se hacen a estudiantes, si el usuario que ha ingresado al sistema es de tipo Administrador.

## Administración de solicitudes, mantenimientos y bitácoras de JRequest

The screenshot displays the JRequest administrative interface. At the top, a navigation bar includes links for Principal, Mantenimientos, Laboratorios, Inventario, Cursos, Wiki, and Cañones. The main content area has a header "Laboratorio de Hardware" and a sub-header "Universidad de El Salvador, Facultad Multidisciplinaria de Occidente". On the left, a sidebar titled "Sesión" shows "Bienvenido usuario hugol" and a "Logout" button. Below it, a list of "Tareas Comunes" includes: Cambiar clave de acceso, Ver Reportes de JHard, Administrar Solicitudes de JRequest, Tareas Administrativas de JRequest, Emitir Solicitud de Soporte, and Técnico. A central panel titled "Mantenimiento de Hardware y Software" contains the sub-header "Administración de Solicitudes de Mantenimiento UES-FMOcc" and a section titled "Seleccione una solicitud de la lista de la derecha y asigne la a Mantenimiento". This section lists several computer hardware items with status messages. To the right, a form is displayed with fields for "Persona encargada" (set to "PERSONA"), "Nombre de la solicitud" (set to "NOMBRE"), "Prioridad" (set to "Alta"), "Técnico a asignar" (set to "Gabriel Aguirre"), and two buttons: "Proceder a Mantenimiento" and "Realizar una Solicitud". At the bottom, a footer bar includes links for Mantenimientos, Bitácoras, and Solicitudes, along with copyright information for the University of El Salvador.

## Descripción

- Esta página es de uso exclusivo para usuarios de tipo Administrador

- Permite manejar las solicitudes de mantenimiento de la facultad, así como las internas para el centro de cómputo de Ingeniería y Arquitectura. Las recibe y el administrador decide cuáles son las que pasan a mantenimiento.
- Maneja además el equipo que se encuentra en mantenimiento. Cuando ya ha finalizado se crea una bitácora y se saca el equipo de mantenimiento
- También se pueden observar y modificar todas las bitácoras de cada equipo. Se escoge el equipo y muestras sus respectivas bitácoras de estados y sus cambios.

### Búsqueda de soluciones de la Wiki desde JRequest

The screenshot shows the JHard system interface. At the top, there is a navigation bar with links: Principal, Mantenimientos, Laboratorios, Inventario, Cursos, Wiki, and Cañones. Below the navigation bar, the title "Laboratorio de Hardware" is displayed, along with the logo of the "Universidad de El Salvador, Facultad Multidisciplinaria de Occidente". On the left side, there is a sidebar titled "Sesión" which shows "Bienvenido usuario hugol" and a "Logout" button. Another sidebar titled "Tareas Comunes" lists items such as "Cambiar clave de acceso", "Ver Reportes de JHard", "Administrar Solicitudes de JRequest", "Tareas Administrativas de JRequest", "Emitir Solicitud de Soporte Técnico", and "Técnico". The main content area is titled "Mantenimiento de Hardware y Software" and contains the message: "Este es el sitio si requiere Soporte Técnico ya sea en Hardware o Software en la UES-FMOcc". Below this, there is a search section titled "Busquedas JWiki" with a search input field containing "extraer USB" and a "Buscar" button. A preview of the search results is shown, with the first result being "¿Cómo saco el dispositivo USB del ordenador?". At the bottom of the page, there is a footer with the text: "Universidad de El Salvador Facultad Multidisciplinaria de Occidente", "Todos los Derechos (C) Reservados. Teléfono: (503)2449-0249, Fax: (503)2449-0352 Apdo. 1908", "Créditos - Página Principal de la UES FMOcc".

### Descripción

- Cualquier usuario que ingresa a JHard puede buscar respuestas o soluciones a problemas informáticos, a través de la búsqueda de JRequest. El usuario ingresa y coloca las palabras clave de búsqueda en el cuadro de texto disponible. Luego, con presionar el botón “Buscar” saldrá una lista con los temas relacionados a la búsqueda inmediatamente.

- Si no encuentra nada que solucione su problema, para usuarios registrados hay un apartado para enviar la solicitud de soporte técnico a los administradores de JHard

### **Envío de solicitud de soporte técnico de JRequest**

The screenshot shows a web application for managing hardware and software requests. At the top, there's a navigation bar with links: Principal, Mantenimientos, Laboratorios, Inventario, Cursos, Wiki, and Cañones. Below the navigation is a header with the text "Laboratorio de Hardware" and "Universidad de El Salvador, Facultad Multidisciplinaria de Occidente". A logo of a person is also present.

The main content area has two main sections:

- Sesión:** Shows a welcome message "Bienvenido usuario hugol" and a "Logout" button.
- Solicitud de Mantenimiento de Hardware y Software:** This section contains fields for "Ingresar una nueva solicitud", "Usuario que realiza la petición" (set to "hugol"), "Descripción" (an empty text area), "Nombre Equipo" (an empty text area), a "Agregar Equipo" button, and an "Enviar" button.

At the bottom of the page, there's a footer with the text: "Universidad de El Salvador Facultad Multidisciplinaria de Occidente", "Todos los Derechos (C) Reservados - Teléfonos:(503)2449-0349, Fax:(503)2449-0352 Apdo. 1908", and "Créditos - Página Principal de la UES FMOCC".

#### **Descripción:**

- Un sencillo formulario para el envío de solicitudes de soporte técnico. Se muestra el nombre del usuario que enviará la solicitud, la descripción del problema que lo aqueja y que escoja el equipo afectado. De no estar registrado, puede agregarlo por él mismo.

**Página de JInvent**

The screenshot displays the JInvent software interface, specifically the 'Administración del Inventario' (Inventory Management) module. The interface is divided into several sections:

- Sesión:** Shows 'Bienvenido usuario hugol' and a 'Logout' button.
- Tareas Comunes:** A list of common tasks including 'Cambiar clave de acceso', 'Administrar Solicitudes de JRequest', 'Tareas Administrativas de JRequest', 'Emitir Solicitud de Soporte Técnico', and 'Búsquedas'.
- Administración del Inventario:** This is the main workspace. It features a tree view on the left under 'Elementos del inventario en la clasificación General' (General Inventory Elements) showing categories like 'General (0)', 'Hardware (0)', 'Equipos (0)', etc. To the right, there are four tabs: 'Existencias de Equipos (Total: 0)', 'Definiciones de Equipos (Total: 0)', 'Software (Total: 0)', and 'Accesorios (Total: 0)'. Below these tabs are tables for 'Piezas (Total: 0)' and 'Marcas (Total: 13)'. At the bottom, there are sections for 'Ubicaciones (Total: 2)' and navigation buttons.
- Footer:** Contains the university's name, copyright information, and a link to the main page of the FMOsc.

**Descripción:**

- Esta página contiene toda la funcionalidad del módulo de JInvent para manejar inventarios de activos fijos para la Unida de Hardware y Software.
- Se visualizan los equipos por clasificaciones y jerarquías en el lado izquierdo de la panel dividido
- Se muestra el detalle de los equipos y existencias al lado derecho del panel divivido

- Para cada clasificación hay links para agregar nuevos equipos
- Muestra también las marcas registradas al inventario. Permite agregar, modificar y eliminar existentes.
- Muestra las ubicaciones agregadas de la Facultad. También permite el mantenimiento completo
- Módulo estrictamente para usuarios de tipo Administrador

### JCanon: Reserva de equipo multimedia

**Laboratorio de Hardware**  
Universidad de El Salvador, Facultad Multidisciplinaria de Occidente

**Sesión**

Bienvenido usuario hugol  
[Logout](#)

**Tareas Comunes**

- » Cambiar clave de acceso
- » Ver Reportes de Jhard
- » Administrar Reserva de Equipo
- » Multimedia
- » Ver calendario de reservas

**Reserva de Equipo Multimedia**

6 Jul 2009 – 13 Jul 2009

Lun, Julio 6	Mar, Julio 7	Me, Julio 8	Jue, Julio 9	Vie, Julio 10	Sab, Julio 11	Dom, Julio 12
07:00						
08:00						
09:00						
10:00	Presto de Cañón Epson	Presto de Cañón Epson	Presto de Cañón Epson			
11:00	Presto de Cañón Epson	Presto de Cañón Epson	Presto de Cañón Epson			
12:00						
13:00				Prestamo de Laptop Sony VGN		
14:00						
15:00						
16:00						

**Solicitante**  
hugol

**Equipo a reservar**

Cañones	Laptops
<input type="checkbox"/> Reservar Cañon	<input type="checkbox"/> Reservar Laptop

**Seleccione una fecha**

4 agosto > 4 2009 <
lun mar mié jue vie sáb dom
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31

**Hora Inicio**  
6:45

**Hora Finalización**

**Docente Responsable**  
Carlos Stanley Linares Paula

**Crear Reserva**

**Administración**

Universidad de El Salvador Facultad Multidisciplinaria de Occidente  
Todos los Derechos (C) Reservados - Teléfonos:(503)2449-0349, Fax:(503)2449-0352 Apdo. 1908  
Créditos - Página Principal de la UES FMOC

## Descripción

- Permite visualizar todas las reservas de equipo multimedia con su novedoso esquema de calendario (Scheduler)
- Permite agregar una nueva reserva de equipo multimedia, escogiendo el tipo de equipo y la existencia a reservar.
- Tiene links además para la página administrativa de JCanon, con permisos de visibilidad únicamente para usuarios de tipo Administrador.
- Permite, además, visualizar en calendario en una ventana aparte.

## JWiki y JProCur: Páginas para contenido

The screenshot displays the JHard system's user interface. At the top, a navigation bar includes links for Principal, Mantenimientos, Laboratorios, Inventario, Cursos, Wiki, and Cañones. The main header reads "Laboratorio de Hardware" with the subtitle "Universidad de El Salvador, Facultad Multidisciplinaria de Occidente".

**Sesión:** Bienvenido usuario hugol | Logout

**Tareas Comunes:**

- ✗ Cambiar clave de acceso
- ✗ Ver Reportes de Jhard
- ✗ Administrar Wiki
- ✗ Administrar Cursos

**Búsqueda:** Búsqueda | Buscar

**Wiki del Laboratorio:**

**¿Cómo saco el dispositivo USB del ordenador?** (Wed Aug 05 14:05:25 CST 2009)

Como para todo hay formas y formas de hacerlo: una correcta con la podremos estar seguros (*nunca un 100%*) de no perder la información almacenada por el camino y la otra "resolutiva" (*generalmente no suele ocurrir nada, aunque el riesgo es mucho mayor*) sabemos que está mal hecho pero muchas veces por rapidez o por desconocimiento de las demás opciones terminamos echando mano de ésta.

Métodos para extraer correctamente un dispositivo USB

- Quizás la más rápida y la más extendida: clic izquierdo sobre el icono de extracción segura > seleccionar el dispositivo a extraer y volver a hacer clic.
- Muy similar a la anterior pero con más opciones: clic derecho sobre el icono de extracción segura > quitar hardware con seguridad > en el nuevo panel seleccionar el dispositivo a extraer > hacer clic en detener. Observar que marcando la opción *mostrar componentes de dispositivos* se despliega una lista con el nombre y letra de unidad correspondiente.
- Forma rápida y efectiva: desde MI PC > clic derecho sobre el dispositivo a extraer > expulsar.

**¿Qué es JHard?** (Sun Jul 26 09:18:43 CST 2009)

El sistema JHard, es una solución informática creada para el Laboratorio de Hardware y Software de la Universidad de El Salvador - Facultad Multidisciplinaria de Occidente.

Este sistema es compuesto por seis módulos robustos a los que tienen acceso los usuarios de todo tipo, sin embargo está compuesto de otra gran cantidad de subcomponentes que añaden características adicionales, necesarias para el funcionamiento interno del mismo. A continuación, se listan los seis módulos principales con los que tendrá interacción los usuarios finales, y las necesidades que cada uno pretenden solventar:

1. JInvent: Manejo de Inventario del Laboratorio de Hardware (No consumibles).
2. JRquest: Solicitud de servicio al laboratorio.
3. JWiki: Módulo colaborativo de conocimiento, con soluciones a problemas comunes.
4. ProCur: Módulo de Manejo de Contenidos (CMS) para la promoción de cursos del Laboratorio de Hardware.
5. ManLab: Módulo de para la gestión de inscripción de laboratorios prácticos en el Laboratorio de Hardware.
6. JCanon: Módulo para la gestión de reserva de cañones y/o laptops.

El sistema JHard tendrá además, una interfaz administrativa (JHardmin) con la que se podrá dar mantenimiento al sitio, y administrar los perfiles/roles de los usuarios.

**Descripción:**

- Esta página muestra el contenido tanto para el módulo de JWiki como para JProCur.
- Muestra artículos para JWiki, como entradas para JProCur
- Las muestran ordenadas por fecha, mostrando en primer lugar la escrita más recientemente.

**JWiki y JProCur: Páginas para administración**

The screenshot shows the JWiki administration interface. At the top, there is a navigation bar with tabs: Principal, Mantenimientos, Laboratorios, Inventario, Cursos, Wiki, and Catálogos. The 'Wiki' tab is currently selected. Below the navigation bar, the page title is 'Laboratorio de Hardware' and it includes the subtitle 'Universidad de El Salvador, Facultad Multidisciplinaria de Occidente'. On the left side, there is a sidebar titled 'Sesión' which says 'Bienvenido usuario hugol' and has a 'Logout' button. Another sidebar titled 'Tareas Comunes' lists several items: 'Cambiar clave de acceso', 'Ver Reportes de JHard', 'Ir a Wiki', and 'Ir a Promoción de Cursos'. The main content area is titled 'Administración Wiki' and contains a 'Administrador Artículos' section. It features a 'Título:' input field, a large rich text editor with various formatting tools (bold, italic, underline, etc.), and buttons for 'Administrar Artículos' and 'Agregar Artículo'. At the bottom of the main content area, it says 'Escrito por: hugol' and 'Wed Aug 05 14:05:25 CST 2009'. There are 'Guardar' and 'Cancelar' buttons at the very bottom. A footer at the bottom of the page contains the text: 'Universidad de El Salvador Facultad Multidisciplinaria de Occidente', 'Todos los Derechos Reservados - Teléfonos: (503)2449-0249, Fax: (503)2449-0352 Apdo. 1908', and 'Créditos - Página Principal de la UES FMOOC'.

## Descripción

- Estas páginas tienen interfaz para escribir nuevas entradas y artículos.
- Muestran las entradas o artículos del usuario registrado al sistema.
- Permite editar o eliminar dichas entradas o artículos.

## Página administrativa

The screenshot shows a web-based administrative interface for a hardware laboratory. At the top, there's a navigation bar with links: Principal, Mantenimientos, Laboratorios, Inventario, Cursos, Wiki, and Cañones. Below the navigation is the title "Laboratorio de Hardware" and the subtitle "Universidad de El Salvador, Facultad Multidisciplinaria de Occidente".

**Sesión:** Bienvenido usuario hugol (Logout)

**Tareas Comunes:**

- Ver calendario de Horarios
- Ir a página principal de
- Inscripción de Horarios
- Cambiar clave de acceso

**Administración de Horarios:**

Reservas de Equipo Multimedia

Grupo de Práctica	Día	Hora Inicio	Hora Fin	Modificar Horario	Eliminar
Programación II - Grupo 1	Lunes	10:05 AM	10:55 AM	Modificar Horario	Eliminar
Programación II - Grupo 1	Lunes	10:55 AM	11:45 AM	Modificar Horario	Eliminar
Programación II - Grupo 1	Martes	7:35 AM	8:25 AM	Modificar Horario	Eliminar
Programación II - Grupo 1	Martes	10:05 AM	10:55 AM	Modificar Horario	Eliminar
Programación II - Grupo 2	Martes	10:55 AM	11:45 AM	Modificar Horario	Eliminar
Programación II - Grupo 1	Miércoles	7:35 AM	9:15 AM	Modificar Horario	Eliminar
Programación II - Grupo 1	Miércoles	7:35 AM	7:00 PM	Modificar Horario	Eliminar
Programación II - Grupo 1	Miércoles	10:00 AM	10:55 AM	Modificar Horario	Eliminar
Programación II - Grupo 2	Jueves	10:55 AM	11:45 AM	Modificar Horario	Eliminar
Programación II - Grupo 1	Viernes	10:00 AM	10:55 AM	Modificar Horario	Eliminar

Navigation icons: < > << >>

At the bottom, there's a footer with the text: Universidad de El Salvador Facultad Multidisciplinaria de Occidente, Todos los Derechos (C) Reservados - Teléfonos: (503)2449-0349, Fax: (503)2449-0352 Apdo. 1908, Créditos - Página Principal de la UES FMOsc.

## Descripción

- Muchas páginas de carácter administrativo en JHard tienen la apariencia de la figura anterior.
- Muestra un listado de ítems (en este caso horarios de grupos de laboratorio) con la información básica asociada en cada una de las columnas de la tabla.
- En la parte de la derecha de la tabla, muestra links para modificar cada uno de los ítems y para eliminarlo.
- Las “tareas comunes” son propias para cada módulo en cuestión, donde se encuentre dicha página administrativa.

**ManLab**

The screenshot shows the ManLab system interface. At the top, there is a navigation bar with links: Principal, Mantenimientos, Laboratorios (highlighted in orange), Inventario, Cursos, Wiki, and Catálogos. Below the navigation bar, the title "Laboratorio de Hardware" is displayed, along with the text "Universidad de El Salvador, Facultad Multidisciplinaria de Occidente". On the left, there is a sidebar titled "Sesión" with a "Logout" button. Another sidebar titled "Tareas Comunes" lists "Cambiar clave de acceso" and "Ver Reportes de JHard". The main content area is titled "Actividades del Laboratorio" and contains the message "Estas son las actividades que su usuario puede realizar en el Laboratorio...". It features three icons: "Inscribirse a un Curso" (with a pen icon), "Asistencia Clase/Materia" (with a green checkmark icon), and "Horarios del Laboratorio" (with a calendar icon). At the bottom of the page, there is a footer with the text "Universidad de El Salvador Facultad Multidisciplinaria de Occidente", "Todos los Derechos (C) Reservados - Teléfonos: (503)2449-0349, Fax: (503)2449-0352 Apdo. 1908", and "Créditos - Página Principal de la UES PMOCC".

**Descripción:**

- Las actividades de ManLab se muestran según sean los privilegios del usuario. Para la figura de arriba se muestran las opciones a las que tiene acceso un estudiante, que serían: “Inscribirse a un curso”, “Asistencia Clase/Materia” y “Horarios del Laboratorio”.
- Cada uno de éstos tiene una funcionalidad muy sencilla, con unos cuantos clics el usuario hace las actividades relacionadas para cada uno.

## Configuración de la plataforma del sistema

### Instalación y Configuración de Tomcat 6.0.20 en Debian/Ubuntu

La siguiente sección supone una instalación de Tomcat desde cero, en una distribución de GNU/Linux basada en Debian (para este caso, se empleo ubuntu-jeos). Se usara un proceso manual, debido a que el paquete nativo presenta muchos problemas a la hora de instalarlo.

#### Instalación de Java 6

Antes de instalar Tomcat, es necesario asegurarnos que se tiene Java Runtime (JRE) instalado. Para verificar la instalacion del JRE se puede utilizar el siguiente comando:

```
usuario@ubuntu:/$ dpkg -l | grep sun-java
```

Que produce una salida similar a esta:

```
ii  sun-java6-bin   6-14-0ubuntu1.8.04  Sun Java(TM) Runtime Environment  
(JRE) 6  
ii  sun-java6-jre   6-14-0ubuntu1.8.04          Sun Java(TM) Runtime  
Environment
```

Si el comando no produce resultados, podemos instalar Java con el siguiente comando:

```
root@ubuntu:/# apt-get install sun-java6-jdk
```

Se mostrara el cuadro para aceptar la licencia de distribución de Java:

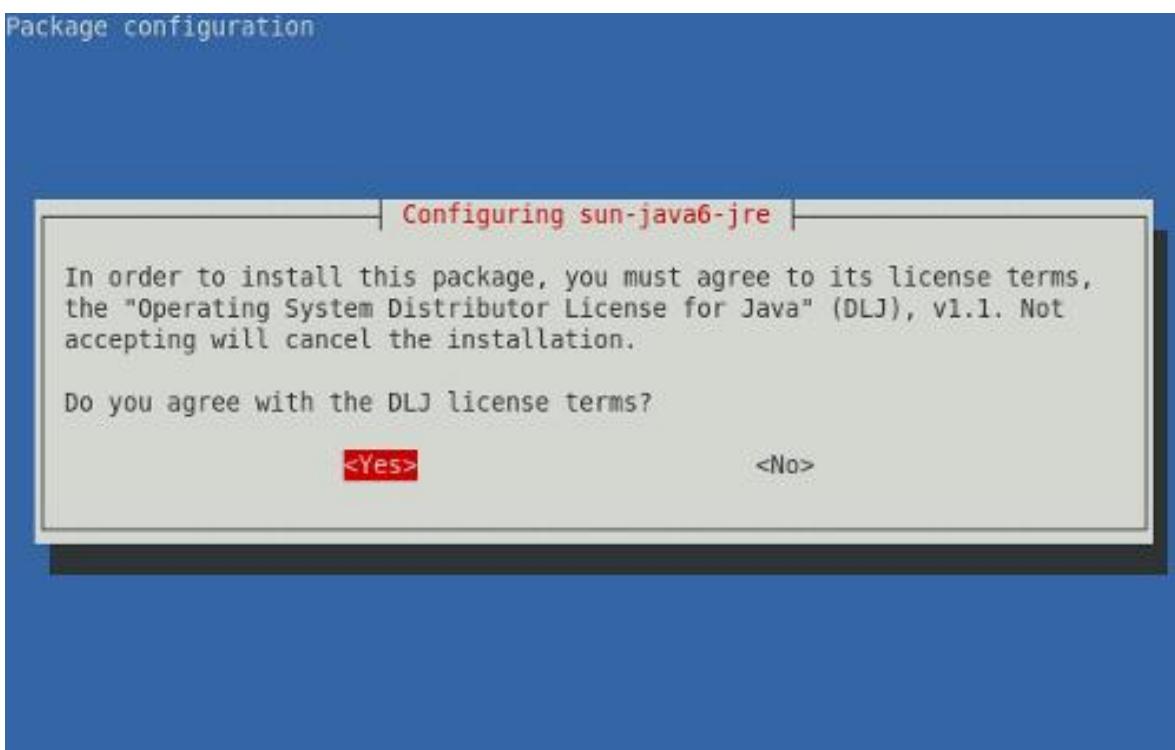


Figura 6.1: Configuración de Java JRE 6 en Ubuntu

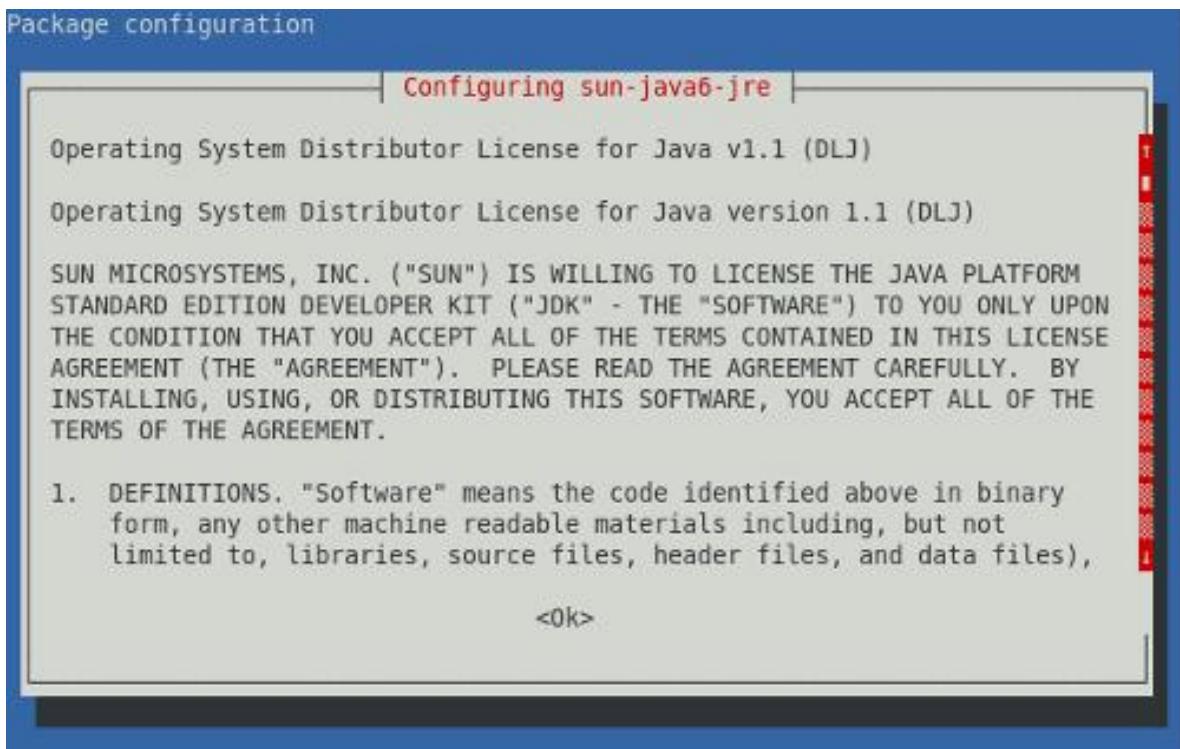


Figura 6.2: Configuración de Java JRE 6 en Ubuntu

## Obteniendo Tomcat 6

Ahora es necesario descargar y descomprimir Tomcat del sitio de Apache Software Foundation (<http://tomcat.apache.org>). Al momento de redactar este documento, se utilizo la versión 6.0.20 de Tomcat.

```
root@ubuntu:~# wget http://apache.osuosl.org/tomcat/tomcat-6/v6.0.20/bin/apache-tomcat-6.0.20.tar.gz
```

Después de obtener el paquete, procedemos a descomprimirlo:

```
root@ubuntu:~# tar xf apache-tomcat-6.0.20.tar.gz
```

Y ahora movemos la carpeta extraída (apache-tomcat-6.0.20) a “/usr/local/tomcat”:

```
root@ubuntu:~# mv apache-tomcat-6.0.20 /usr/local/tomcat
```

Tomcat requiere la variable “JAVA\_HOME” en el “Environment”, y añadir la ubicación de la ruta de los binarios a la variable “PATH” (también en el “Environment”).

Esto se logra de dos formas, primero editando el archivo “.bashrc” del usuario “root”:

```
root@ubuntu:~# nano .bashrc
```

Añadiendo al final, las siguientes líneas:

```
#para el javapath
export JAVA_HOME=/usr/lib/jvm/java-6-sun
export PATH=$JAVA_HOME/bin:$PATH
```

*Nota Importante: la ruta usada en JAVA\_HOME puede variar entre diferentes versiones de Ubuntu/Debian, así como entre distribuciones Linux. Verifique la ruta especificada.*

Para que este cambio tenga efecto, es necesario que el usuario root, termine la sesión e inicie una nueva. La segunda forma, se describe en la siguiente sección.

## Inicio Automático

Para hacer que Tomcat inicie automáticamente cuando inicia el servidor, es necesario crear un script, que llamaremos “Tomcat” en la carpeta “/etc/init.d”:

```
root@ubuntu:~# nano /etc/init.d/tomcat
```

Ahora copiamos en ese archivo, el siguiente script:

```
# Tomcat inicio automatico
# description: inicia automaticamente a Tomcat
# processname: tomcat
# pidfile: /var/run/tomcat.pid
export JAVA_HOME=/usr/lib/jvm/java-6-sun
export PATH=$JAVA_HOME/bin:$PATH
case $1 in
start)
    sh /usr/local/tomcat/bin/startup.sh
;;
stop)
    sh /usr/local/tomcat/bin/shutdown.sh
;;
restart)
    sh /usr/local/tomcat/bin/shutdown.sh
    sh /usr/local/tomcat/bin/startup.sh
;;
esac
exit 0
```

Ahora hay que conceder los permisos de ejecución al script, con el comando chmod:

```
root@ubuntu:/# chmod 755 etc/init.d/tomcat
```

Y el último paso es crear un vinculo simbólico (acceso directo) entre el script de Tomcat y la carpeta del nivel de ejecución en el que queremos que inicie Tomcat.

```
root@ubuntu:/# ln -s etc/init.d/tomcat etc/rc1.d/K99tomcat
root@ubuntu:/# ln -s etc/init.d/tomcat etc/rc2.d/S99tomcat
```

Ahora Tomcat está completamente instalado y funcional.

### Desplegando la aplicación

Para desplegar la aplicación, detenemos primero el servidor Tomcat con el comando:

```
root@ubuntu:/# /etc/init.d/tomcat stop
```

Renombramos la carpeta “\$CATALINA\_HOME/webapps/ROOT”, crearemos una carpeta nueva para el proyecto, y descomprimiremos el contenido del archivo WAR (jhard.war) en esta:

```
root@ubuntu:/usr/local/tomcat/webapps/# mv ROOT ROOT.old
root@ubuntu:/usr/local/tomcat/webapps/# mkdir jhard
root@ubuntu:/usr/local/tomcat/webapps/# cd JHard
root@ubuntu:/usr/local/tomcat/webapps/jhard/# unzip jhard.war
root@ubuntu:/usr/local/tomcat/webapps/jhard/# rm jhard.war
```

Finalmente, podemos iniciamos Tomcat:

```
root@ubuntu:/# /etc/init.d/tomcat start
```

*Nota Importante: Segundo los requerimientos mínimos de funcionamiento de la aplicación web, es necesario especificar la cantidad de memoria RAM que el proceso de TOMCAT puede reservar para funcionar. Esto se logra, modificando el script **catalina.sh**, agregándole a la variable CATALINA\_OPTS las opciones –Xms256M y –Xmx512M, así:*

**CATALINA\_OPTS =”–Xms256M –Xmx512M”**

### Instalación y configuración de MySQL

Para instalar MySQL 5.0, solo necesitamos ejecutar el siguiente comando:

```
root@ubuntu:/# apt-get install mysql-server-5.0
```

Al instalar MySQL, se solicita una contraseña para el usuario “root” de MySQL:

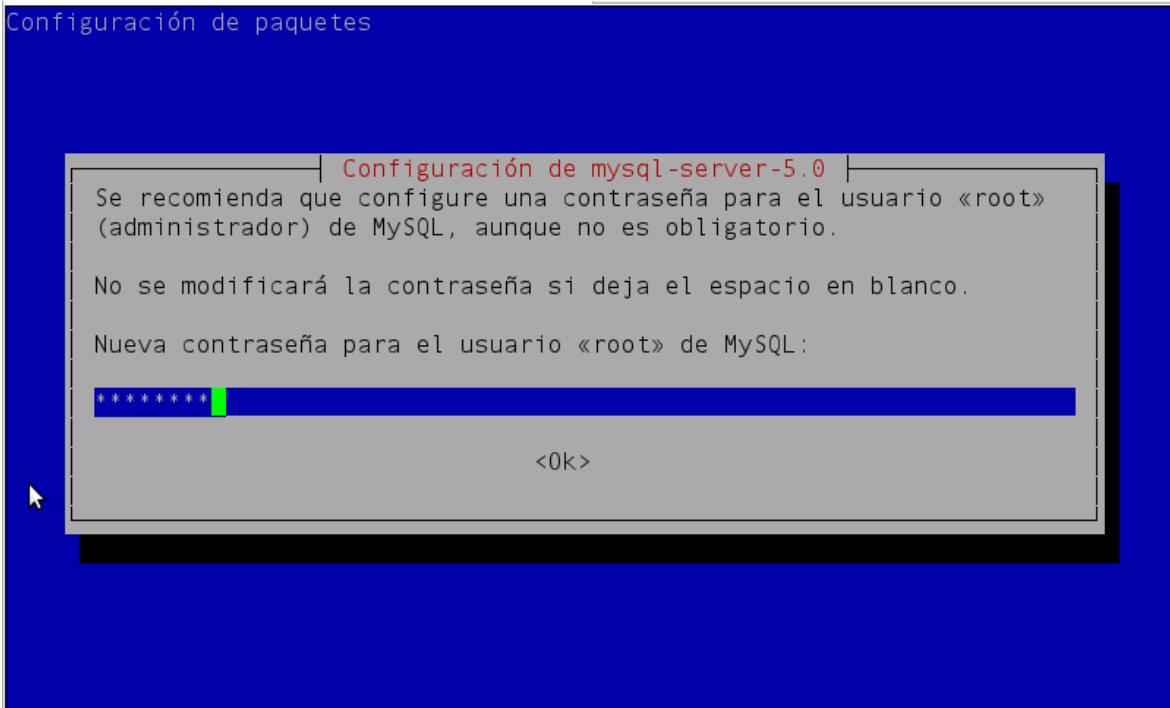


Figura 6.3: Configuración MySQL 5.1 en Ubuntu

Después de instalar MySQL, vamos a restaurar la base de datos de jhard, que se encuentra en el archivo jhard.sql:

```
root@ubuntu:/# mysql -uroot -pCoNtRaSeÑa < jhard.sql
```

*Nota Importante: La mínima cantidad de datos, necesaria para que JHard sea funcional, se encuentran en el script SQL **jhard.sql**.*

#### Creación de usuario y asignación de permisos:

Después de la restauración de la base de datos, vamos a crear un usuario para jhard y asignar los permisos al mismo:

```
mysql> CREATE USER 'jharduser'@'localhost' IDENTIFIED BY 'jhardpwd';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'jharduser'@'localhost' WITH GRANT OPTION;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CREATE USER 'jharduser'@'%' IDENTIFIED BY 'jhardpwd';
Query OK, 0 rows affected (0.01 sec)
mysql> GRANT ALL PRIVILEGES ON *.* TO 'jharduser'@'%' WITH GRANT OPTION;
```

Con esto, MySQL está instalado, la base de datos restaurada.

#### **Firewall para JHard:**

Si bien la configuración avanzada de un Firewall, va más alla del alcance de este documento, podemos hacer un par de recomendaciones que el administrador del servidor en el que se encuentre JHard, debe tomar en cuenta.

Suponiendo una herramienta de consola y una configuración de texto plano, se recomienda utilizar **Shorewall** (<http://www.shorewall.net/>) como herramienta para configurar un Firewall.

Sino, también se puede utilizar **ipkungfu** (<http://freshmeat.net/projects/ipkungfu/>), que provee un Firewall preconfigurado (“Out of the Box”) que le permitirá configurar fácilmente las reglas del mismo, en archivos de configuración sencillos y bien documentados.

Recuerde NO bloquear los puertos de Tomcat (8080 por defecto), y MySQL (3306).

## Diccionario de datos

### 1. Información de la Base de Datos

<b>Descripción</b>	Base de datos para JHard. UES-FMOCC
<b>Nombre</b>	jhard
<b>Character Set</b>	Latin1
<b>SQL para crear la DB</b>	CREATE DATABASE IF NOT EXISTS `jhard`; USE `jhard`;

### 2. Tablas

#### 2.1 accesorio

Nombre del Parámetro	Valor del Parámetro
Tipo de Tabla	INNODB
Formato de la Fila	DEFAULT
Temporal	NO
Checksum de Fila	NO

Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idaccesorio	int(11)	SI	SI	SI			Id correlativo unico de cada accesorio
nombre	varchar(100)	NO	SI	NO			Nombre del accesorio
idmarca	int(11)	NO	SI	NO			Referencia a la marca del accesorio
modelo	varchar(15)	NO	SI	NO			Modelo del accesorio
idclasificacion	int(11)	NO	SI	NO			Referencia a la clasificacion en la que se encuentra este accesorio
idexistencia	int(11)	NO	NO	NO			

Index	Columnas
PRIMARY	idaccesorio
fkidmarca_accesorio	idmarca
fkidclasificacion_accesorio	idclasificacion
fkidexistencia_accesorio	idexistencia

**Table Create SQL**

```

CREATE TABLE `accesorio` (
  `idaccesorio` int(11) NOT NULL AUTO_INCREMENT COMMENT 'Id correlativo unico de cada accesorio',
  `nombre` varchar(100) NOT NULL COMMENT 'Nombre del accesorio',
  `idmarca` int(11) NOT NULL COMMENT 'Referencia a la marca del accesorio',
  `modelo` varchar(15) NOT NULL COMMENT 'Modelo del accesorio',
  `idclasificacion` int(11) NOT NULL COMMENT 'Referencia a la clasificacion en la que se encuentra este accesorio',
  `idexistencia` int(11),
  PRIMARY KEY(`idaccesorio`),

```

```

INDEX `fkidmarca_accesorio`(`idmarca`),
INDEX `fkidclasificacion_accesorio`(`idclasificacion`),
INDEX `fkidexistencia_accesorio`(`idexistencia`),
CONSTRAINT `fkidclasificacion_accesorio` FOREIGN KEY
(`idclasificacion`)
    REFERENCES `clasificacion`(`idclasificacion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkidexistencia_accesorio` FOREIGN KEY
(`idexistencia`)
    REFERENCES `existencia`(`idexistencia`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkidmarca_accesorio` FOREIGN KEY (`idmarca`)
    REFERENCES `marca`(`idmarca`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idclasificacion`) REFER
`jhard/clasificacion`(`idclasifi`;

```

## 2.2 administrador

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idadministrador	int(11)	SI	SI	NO			Id correlativo unico de cada administrador
clave	varchar(45)	NO	SI	NO			Clave del administrador
idusuario	int(11)	NO	SI	NO			referencia al usuario relacionado con este administrador
Index	Columnas						
PRIMARY		idadministrador					
fkidusuario_administrador		idusuario					

**Table Create SQL**

```

CREATE TABLE `administrador` (
  `idadministrador` int(11) NOT NULL COMENTARIO 'Id correlativo
unico de cada administrador',
  `clave` varchar(45) NOT NULL COMENTARIO 'Clave del
administrador',
  `idusuario` int(11) NOT NULL COMENTARIO 'referencia al
usuario relacionado con este administrador',
  LLAVE PRIMARIA(`idadministrador`),
  INDEX `fkidusuario_administrador`(`idusuario`),
  CONSTRAINT `fkidusuario_administrador` FOREIGN KEY
(`idusuario`)
    REFERENCES `usuario`(`idusuario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)

```

```
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idusuario`) REFER
`jhard/usuario`(`idusuario`) ON DELETE';
```

### 2.3 adquisicion

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Flags	Valor por Defecto	Comentario
idadadquisicion	int(11)	SI	SI	SI			Id correlativo unico de la adquisicion
fecha	date	NO	SI	NO			Fecha en la que se adquirio el equipo o software
precio	double	NO	SI	NO			Precio de compra del equipo o software (dejar a cero si fue una donacion)
descripcion	text	NO	NO	NO			Detalles de la adquisicion
proveedor	varchar(100)	NO	NO	NO			Nombre del proveedor o tienda donde se compro el equipo o software (en caso de haber sido comprado)
Index	Columnas						
PRIMARY	idadadquisicion						
<b>Table Create SQL</b>	<pre>CREATE TABLE `adquisicion` (   `idadadquisicion` int(11) NOT NULL AUTO_INCREMENT COMENTARIO   'Id correlativo unico de la adquisicion',   `fecha` date NOT NULL COMENTARIO 'Fecha en la que se adquirio el equipo o software',   `precio` double NOT NULL COMENTARIO 'Precio de compra del equipo o software (dejar a cero si fue una donacion)',   `descripcion` text COMENTARIO 'Detalles de la adquisicion',   `proveedor` varchar(100) COMENTARIO 'Nombre del proveedor o tienda donde se compro el equipo o software (en caso de haber sido comprado)',   LLAVE PRIMARIA(`idadadquisicion`) ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB';</pre>						

### 2.4 articulos

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Flags	Valor por Defecto	Comentario
idarticulo	int(10)	SI	SI	SI	UNSIGNED		

D				
titulo	varchar(50)	NO	SI	NO
descripcion	text	NO	SI	NO
fechahora	datetime	NO	SI	NO
idusuario	int(11)	NO	SI	NO
Index	Columnas			
PRIMARY	idarticulo			
fk_articulo_usuario	idusuario			
idxArtTitulo	titulo			
idxArtFecha	fechahora			
<b>Table Create SQL</b>	<pre> CREATE TABLE `articulos` (   `idarticulo` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,   `titulo` varchar(50) NOT NULL,   `descripcion` text NOT NULL,   `fechahora` datetime NOT NULL,   `idusuario` int(11) NOT NULL,   PRIMARY KEY(`idarticulo`),   INDEX `fk_articulo_usuario`(`idusuario`),   INDEX `idxArtTitulo`(`titulo`),   INDEX `idxArtFecha`(`fechahora`),   CONSTRAINT `fkarticulosusuario` FOREIGN KEY (`idusuario`)     REFERENCES `usuario`(`idusuario`)     ON DELETE NO ACTION     ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'Articulos de jwiki; InnoDB free: 5120 kB; (`idusuario`) REFER `jhard/usuario`(`i`; </pre>			

## 2.5 asistencia

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Fla gs	Valor por Defecto	Comentario
idasistencia	int(11)	SI	SI	SI			Id correlativo unico para cada asistencia
idestudiente	int(11)	NO	SI	NO			Referencia al estudiante que asistio al curso
idclase	int(11)	NO	SI	NO			Referencia a la clase a la cual pertenece esta asistencia
idequipoexistente	int(11)	NO	NO	NO			Referencia al equipo de hardware que se utilizo en dicha asistencia a la clase
Index	Columnas						
PRIMARY	idasistencia						

fkidestudiante_asistencia	idestudiante
fkidclase_asistencia	idclase
fkidequipoexistente_asistencia	idequipoexistente

**Table Create SQL**

```

CREATE TABLE `asistencia` (
    `idasistencia` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico para cada asistencia',
    `idestudiante` int(11) NOT NULL COMENTARIO 'Referencia al
estudiante que asistio al curso',
    `idclase` int(11) NOT NULL COMENTARIO 'Referencia a la clase
a la cual pertenece esta asistencia',
    `idequipoexistente` int(11) COMENTARIO 'Referencia al equipo
de hardware que se utilizo en dicha asistencia a la clase',
    LLAVE PRIMARIA(`idasistencia`),
    INDEX `fkidestudiante_asistencia`(`idestudiante`),
    INDEX `fkidclase_asistencia`(`idclase`),
    INDEX `fkidequipoexistente_asistencia`(`idequipoexistente`),
    CONSTRAINT `fkidclase_asistencia` FOREIGN KEY (`idclase`)
        REFERENCES `clase`(`idclase`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fkidequipoexistente_asistencia` FOREIGN KEY
(`idequipoexistente`)
        REFERENCES `existencia`(`idexistencia`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fkidestudiante_asistencia` FOREIGN KEY
(`idestudiante`)
        REFERENCES `estudiante`(`idestudiante`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idclase`) REFER
`jhard/clase`(`idclase`) ON DELETE NO AC';

```

## 2.6 atributohardware

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Flag	Valor por Defecto	Comentario
idatributohardware	int(11)	SI	SI	SI			Id correlativo unico del atributo de hardware
nombre	varchar (45)	NO	SI	NO			Nombre del atributo
valor	varchar (45)	NO	SI	NO			Valor del atributo
unidadmedida	varchar (45)	NO	SI	NO			Unidad de medida del atributo
idhardware	int(11)	NO	NO	NO			Referencia al elemento de hardware (equipo,

e	pieza o accesorio) al que pertenece el atributo			
idpieza	int(11)	NO	NO	NO
idaccesorio	int(11)	NO	NO	NO
<b>Index</b>				
PRIMARY	idatributohardware			
fkidequipo_atributohardware	idhardware			
fkidpieza_atributohardware	idpieza			
fkidaccesorio_atributohardware	idaccesorio			

<b>Table Create SQL</b>	<code>CREATE TABLE `atributohardware` (</code>
	`idatributohardware` int(11) NOT NULL AUTO_INCREMENT
	COMENTARIO 'Id correlativo unico del atributo de hardware',
	`nombre` varchar(45) NOT NULL COMENTARIO 'Nombre del atributo',
	`valor` varchar(45) NOT NULL COMENTARIO 'Valor del atributo',
	`unidadmedida` varchar(45) NOT NULL COMENTARIO 'Unidad de medida del atributo',
	`idhardware` int(11) COMENTARIO 'Referencia al elemento de hardware (equipo, pieza o accesorio) al que pertenece el atributo',
	`idpieza` int(11),
	`idaccesorio` int(11),
	LLAVE PRIMARIA(`idatributohardware`),
	INDEX `fkidequipo_atributohardware`(`idhardware`),
	INDEX `fkidpieza_atributohardware`(`idpieza`),
	INDEX `fkidaccesorio_atributohardware`(`idaccesorio`),
	CONSTRAINT `fkidaccesorio_atributohardware` FOREIGN KEY (`idaccesorio`)
	REFERENCES `accesorio`(`idaccesorio`)
	ON DELETE NO ACTION
	ON UPDATE NO ACTION,
	CONSTRAINT `fkidequipo_atributohardware` FOREIGN KEY (`idhardware`)
	REFERENCES `equipo`(`idequipo`)
	ON DELETE NO ACTION
	ON UPDATE NO ACTION,
	CONSTRAINT `fkidpieza_atributohardware` FOREIGN KEY (`idpieza`)
	REFERENCES `pieza`(`idpieza`)
	ON DELETE NO ACTION
	ON UPDATE NO ACTION
	)
	ENGINE=INNODB
	COMENTARIO = 'InnoDB free: 5120 kB; (`idaccesorio`) REFER `jhard/accesorio`(`idaccesorio`) ON ';

## 2.7 autorizacion

Nombre del Parámetro	Valor del Parámetro
Tipo de Tabla	INNODB
Formato de la Fila	DEFAULT
Temporal	NO
Checksum de Fila	NO

Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idautorizacion	int(10)	SI	SI	SI	UNSIGNED		
codigo	varchar(10)	NO	NO	NO			
cantmaxima	int(10)	NO	NO	NO	UNSIGNED		
Index	Columnas						
PRIMARY	idautorizacion						
<b>Table Create SQL</b>		<pre>CREATE TABLE `autorizacion` (   `idautorizacion` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,   `codigo` varchar(10),   `cantmaxima` int(10) UNSIGNED,   LLAVE PRIMARIA(`idautorizacion`) ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB';</pre>					
<b>2.8 bitacoracambiosusuario</b>							
Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idbitacora	int(11)	SI	SI	SI			Id correlativo unico de cada bitacora
idusuario	int(11)	NO	SI	NO			Referencia al usuario que realizo el cambio
descripcion	text	NO	SI	NO			Descripcion del cambio que realizo el usuario
fechahora	datetime	NO	SI	NO			Fecha y hora a la que el usuario realizo el cambio
Index	Columnas						
PRIMARY	idbitacora						
fkidusuario_bitacoracambiosusuario	idusuario						
<b>Table Create SQL</b>		<pre>CREATE TABLE `bitacoracambiosusuario` (   `idbitacora` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada bitacora',   `idusuario` int(11) NOT NULL COMENTARIO 'Referencia al usuario que realizo el cambio',   `descripcion` text NOT NULL COMENTARIO 'Descripcion del cambio que realizo el usuario',   `fechahora` datetime NOT NULL COMENTARIO 'Fecha y hora a la que el usuario realizo el cambio',</pre>					

```

    LLAVE PRIMARIA(`idbitacora`),
    INDEX `fkidusuario_bitacoracambiosusuario`(`idusuario`),
    CONSTRAINT `fkidusuario_bitacoracambiosusuario` FOREIGN KEY
    (`idusuario`)
        REFERENCES `usuario`(`idusuario`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idusuario`) REFER
`jhard/usuario`(`idusuario`) ON DELETE';

```

## 2.9 bitacoraestados

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idbitacora	int(11)	SI	SI	SI			Id correlativo unico de cada bitacora
fecha	date	NO	SI	NO			Fecha en la que ocurrio el cambio de estado
ideestado	int(11)	NO	SI	NO			Referencia al estado al cual cambio el equipo
descripcion	text	NO	SI	NO			Descripcion del cambio realizado
idequipoexistente	int(11)	NO	NO	NO			Referencia al equipo que sufrio el cambio de estado
idequiposimple	int(11)	NO	NO	NO			
Index	Columnas						
PRIMARY	idbitacora						
fkidestado_bitacoraestados	ideestado						
fkidequipoexistente_bitacoraestados	idequipoexistente						
fkidequiposimple_bitacoraestados	idequiposimple						

**Table Create SQL**

```

CREATE TABLE `bitacoraestados` (
    `idbitacora` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico de cada bitacora',
    `fecha` date NOT NULL COMENTARIO 'Fecha en la que ocurrio el
cambio de estado',
    `ideestado` int(11) NOT NULL COMENTARIO 'Referencia al estado
al cual cambio el equipo',
    `descripcion` text NOT NULL COMENTARIO 'Descripcion del
cambio realizado',
    `idequipoexistente` int(11) COMENTARIO 'Referencia al equipo
que sufrio el cambio de estado',
    `idequiposimple` int(11),
    LLAVE PRIMARIA(`idbitacora`),
    INDEX `fkidestado_bitacoraestados`(`ideestado`),

```

```

INDEX
`fkidequipoexistente_bitacoraestados`(`idequipoexistente`),
INDEX `fkidequiposimple_bitacoraestados`(`idequiposimple`),
CONSTRAINT `fkidequipoexistente_bitacoraestados` FOREIGN KEY
(`idequipoexistente`)
    REFERENCES `existencia`(`idexistencia`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkidequiposimple_bitacoraestados` FOREIGN KEY
(`idequiposimple`)
    REFERENCES `equiposimple`(`idEquipoSimple`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkidestado_bitacoraestados` FOREIGN KEY
(`idestado`)
    REFERENCES `estadoequipo`(`idestado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idequipoexistente`) REFER
`jhard/existencia`(`idexistenc';

```

## 2.10 carrera

Nombre del Parámetro	Valor del Parámetro
Tipo de Tabla	INNODB
Formato de la Fila	DEFAULT
Temporal	NO
Checksum de Fila	NO

Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idcarrera	int(11)	SI	SI	SI			Id correlativo unico de cada carrera
codigo	varchar(7)	NO	SI	NO			Codigo de la carrera, distintivo en el sistema adacad
nombre	varchar(200)	NO	SI	NO			Nombre de la carrera
idfacultad	int(11)	NO	SI	NO			Referencia a la facultad a la cual pertenece esta carrera

Index	Columnas
PRIMARY	idcarrera
fkidfacultad_carrera	idfacultad

Table Create SQL	CREATE TABLE `carrera` ( `idcarrera` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada carrera', `codigo` varchar(7) NOT NULL COMENTARIO 'Codigo de la carrera, distintivo en el sistema adacad', `nombre` varchar(200) NOT NULL COMENTARIO 'Nombre de la carrera', `idfacultad` int(11) NOT NULL COMENTARIO 'Referencia a la facultad a la cual pertenece esta carrera', LLAVE PRIMARIA(`idcarrera`), INDEX `fkidfacultad_carrera`(`idfacultad`),

```

CONSTRAINT `fkidfacultad_carrera` FOREIGN KEY (`idfacultad`)
    REFERENCES `facultad`(`idfacultad`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idfacultad`) REFER
`jhard/facultad`(`idfacultad`) ON DEL';

```

## 2.11 cicloanyo

Nombre del Parámetro		Valor del Parámetro								
Tipo de Tabla		INNODB								
Formato de la Fila		DEFAULT								
Temporal		NO								
Checksum de Fila		NO								
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Flags	Valor por Defecto	Comentario			
idcicloanyo	int(10)	SI	SI	NO	UNSIGNED	ED	Id del ciclo año			
descripcion	varchar(10)	NO	SI	NO	Descripcion del ciclo y año ( I - 2009 )					
Index	Columnas									
PRIMARY	idcicloanyo									
Trigger Name	Trigger Time	Trigger Event								
<b>Table Create SQL</b>		<pre> CREATE TABLE `cicloanyo` (     `idcicloanyo` int(10) UNSIGNED NOT NULL COMENTARIO 'Id del     ciclo año',     `descripcion` varchar(10) NOT NULL COMENTARIO 'Descripcion     del ciclo y año ( I - 2009 )',     LLAVE PRIMARIA(`idcicloanyo`) ) ENGINE=INNODB COMENTARIO = 'Tabla con lista de ciclo y año; InnoDB free: 5120 kB'; </pre>								

## 2.12 clase

Nombre del Parámetro		Valor del Parámetro					
Tipo de Tabla		INNODB					
Formato de la Fila		DEFAULT					
Temporal		NO					
Checksum de Fila		NO					
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Flags	Valor por Defecto	Comentario
idclase	int(11)	SI	SI	SI	Id correlativo unico para cada clase		
fecha	date	NO	SI	NO	Fecha en la que se llevo a cabo esta clase		

idhorario	int(11)	NO	SI	NO	Referencia al horario en el que se recibio esta clase
idinstructor	int(11)	NO	NO	NO	Referencia al instructor encargado de dar esta clase (en caso que haya sido un instructor)
tema	varchar(45)	NO	SI	NO	Tema visto en esta clase
observaciones	text	NO	NO	NO	Observaciones obtenidas segun el resultado general de la clase
iddocente	int(11)	NO	NO	NO	Referencia al docente encargado de dar esta clase (en caso que haya sido un docente)
horainicio	time	NO	SI	NO	Hora de inicio de la clase (hora real)
horafin	time	NO	SI	NO	Hora "marcada" como fin de la clase
finalizada	tinyint(1)	NO	SI	NO	

Index	Columnas
PRIMARY	idclase
fkidhorario_clase	idhorario
fkidinstructor_clase	idinstructor
fkiddocente_clase	iddocente

**Table Create SQL**

```

CREATE TABLE `clase` (
  `idclase` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico para cada clase',
  `fecha` date NOT NULL COMENTARIO 'Fecha en la que se llevo a cabo esta clase',
  `idhorario` int(11) NOT NULL COMENTARIO 'Referencia al horario en el que se recibio esta clase',
  `idinstructor` int(11) COMENTARIO 'Referencia al instructor encargado de dar esta clase (en caso que haya sido un instructor)',
  `tema` varchar(45) NOT NULL COMENTARIO 'Tema visto en esta clase',
  `observaciones` text COMENTARIO 'Observaciones obtenidas segun el resultado general de la clase',
  `iddocente` int(11) COMENTARIO 'Referencia al docente encargado de dar esta clase (en caso que haya sido un docente)',
  `horainicio` time NOT NULL COMENTARIO 'Hora de inicio de la clase (hora real)',
  `horafin` time NOT NULL COMENTARIO 'Hora \"marcada\" como fin de la clase',
  `finalizada` tinyint(1) NOT NULL,
  LLAVE PRIMARIA(`idclase`),
  INDEX `fkidhorario_clase`(`idhorario`),
  INDEX `fkidinstructor_clase`(`idinstructor`),
  INDEX `fkiddocente_clase`(`iddocente`),
  CONSTRAINT `fkiddocente_clase` FOREIGN KEY (`iddocente`)
    REFERENCES `docente`(`iddocente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fkidhorario_clase` FOREIGN KEY (`idhorario`)
    REFERENCES `horario`(`idhorario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,

```

```

CONSTRAINT `fkidinstructor_clase` FOREIGN KEY
(`idinstructor`)
  REFERENCES `instructor`(`idinstructor`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`iddocente`) REFER
`jhard/docente`(`iddocente`) ON DELETE';

```

## 2.13 clasificacion

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Fl ag	Valor por Defecto	Comentario
idclasificacion	int(11)	SI	SI	SI			Id correlativo unico de cada clasificacion
nombre	varchar(100)	NO	SI	NO			Nombre de la clasificacion
descripcion	text	NO	NO	NO			Descripcion de la clasificacion
idsuperior	int(11)	NO	NO	NO			Referencia a la clasificacion padre. Si este campo es nulo, indica que esta es una clasificacion raiz
Index	Columnas						
PRIMARY	idclasificacion						
<b>Table Create SQL</b>	<pre> CREATE TABLE `clasificacion` (   `idclasificacion` int(11) NOT NULL AUTO_INCREMENT COMENTARIO   'Id correlativo unico de cada clasificacion',   `nombre` varchar(100) NOT NULL COMENTARIO 'Nombre de la   clasificacion',   `descripcion` text COMENTARIO 'Descripcion de la   clasificacion',   `idsuperior` int(11) COMENTARIO 'Referencia a la   clasificacion padre. Si este campo es nulo, indica que esta es   una clasificacion raiz',   LLAVE PRIMARIA(`idclasificacion`) ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB'; </pre>						

## 2.14 comentarios

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Fl ag	Valor por Defecto	Comentario
comentario	text	NO	SI	SI			Comentario de la clase
Index	Columnas						
PRIMARY	comentario						

Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIn c	Flags	Valor por Defecto	Comentario
idcoment	int(11)	SI	SI	SI	UNSIGNED		
comentario	varchar(250)	NO	SI	NO			
fechahorara	datetime	NO	SI	NO			
identrada	int(11)	NO	SI	NO	UNSIGNED		
firma	varchar(25)	NO	SI	NO			
aprobado	tinyint(1)	NO	SI	NO	UNSIGNED		
Index	Columnas						
PRIMARY	idcoment						
fk_comentarios_entrada	identrada						
<b>Table Create SQL</b>	<pre>CREATE TABLE `comentarios` (   `idcoment` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,   `comentario` varchar(250) NOT NULL,   `fechahorara` datetime NOT NULL,   `identrada` int(11) UNSIGNED NOT NULL,   `firma` varchar(25) NOT NULL,   `aprobado` tinyint(1) UNSIGNED NOT NULL,   PRIMARY KEY(`idcoment`),   INDEX `fk_comentarios_entrada`(`identrada`),   CONSTRAINT `fk_comentarios_entrada` FOREIGN KEY (`identrada`)     REFERENCES `entrada`(`identrada`)     ON DELETE NO ACTION     ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB; (`identrada`) REFER `jhard/entrada`(`identrada`) ON DELETE';</pre>						

## 2.15 curso

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Aut oIn c	Flag s	Valor por Defecto	Comentario
idcurso	int(11)	SI	SI	SI			Id correlativo unico de cada curso
nombre	varchar(200)	NO	SI	NO			Nombre del curso (por si este difiere del nombre de la materia o por si no esta relacionado con una materia especifica)
cupoma x	int(11)	NO	SI	NO			Cantidad maxima de alumnos que pueden inscribirse a este curso
idmateri	int(11)	NO	NO	NO			Referencia a la materia relacionada con este curso (en

a						caso que este relacionado con alguna)
idinstructo	int(11)	NO	SI	NO		Referencia al instructor asignado a impartir este curso
fechainicio	date	NO	SI	NO		Fecha de inicio del curso
ciclo	int(11)	NO	NO	NO		Ciclo en el que se imparte este curso (1=ciclo impar, 2=ciclo par)
anio	int(11)	NO	NO	NO		anio en el que se imparte este curso
iddocente	int(11)	NO	SI	NO		Referencia al docente encargado de impartir este curso
ideestado	int(11)	NO	NO	NO		
idcicloanio	int(10)	NO	SI	NO	UNSIGNED	fk al ciclo año al que pertenece el curso
habilinscrip	tinyint(1)	NO	SI	NO		esta habilitado el curso para inscripcion de alumnos?

Index	Columnas
PRIMARY	idcurso
fkidmateria_curso	idmateria
fkidinstructor_curso	idinstructor
fkiddocente_curso	iddocente
fkidestado_curso	ideestado
fkciclo_curso	idcicloanio

**Table Create SQL**

```

CREATE TABLE `curso` (
    `idcurso` int(11) NOT NULL AUTO_INCREMENT COMMENT 'Id correlativo unico de cada curso',
    `nombre` varchar(200) NOT NULL COMMENT 'Nombre del curso (por si este difiere del nombre de la materia o por si no esta relacionado con una materia especifica)',
    `cupomax` int(11) NOT NULL COMMENT 'Cantidad maxima de alumnos que pueden inscribirse a este curso',
    `idmateria` int(11) COMMENT 'Referencia a la materia relacionada con este curso (en caso que este relacionado con alguna)',
    `idinstructor` int(11) NOT NULL COMMENT 'Referencia al instructor asignado a impartir este curso',
    `fechainicio` date NOT NULL COMMENT 'Fecha de inicio del curso',
    `ciclo` int(11) COMMENT 'Ciclo en el que se imparte este curso (1=ciclo impar, 2=ciclo par)',
    `anio` int(11) COMMENT 'anio en el que se imparte este curso',
    `iddocente` int(11) NOT NULL COMMENT 'Referencia al docente encargado de impartir este curso',
    `ideestado` int(11),
    `idcicloanio` int(10) UNSIGNED NOT NULL COMMENT 'fk al ciclo año al que pertenece el curso',
    `habilinscrip` tinyint(1) NOT NULL COMMENT 'esta habilitado el curso para inscripcion de alumnos?',
    PRIMARY KEY(`idcurso`),
    INDEX `fkidmateria_curso`(`idmateria`),

```

```

INDEX `fkidinstructor_curso`(`idinstructor`),
INDEX `fkiddocente_curso`(`iddocente`),
INDEX `fkidestado_curso`(`ideestado`),
INDEX `fkcciclo_curso`(`idcicloanio`),
CONSTRAINT `fkcciclo_curso` FOREIGN KEY (`idcicloanio`)
    REFERENCES `cicloanyo`(`idcicloanyo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkiddocente_curso` FOREIGN KEY (`iddocente`)
    REFERENCES `docente`(`iddocente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkidestado_curso` FOREIGN KEY (`ideestado`)
    REFERENCES `estadocurso`(`idestadocurso`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkidinstructor_curso` FOREIGN KEY
(`idinstructor`)
    REFERENCES `instructor`(`idinstructor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkidmateria_curso` FOREIGN KEY (`idmateria`)
    REFERENCES `materia`(`idmateria`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idcicloanio`) REFER
`jhard/cicloanyo`(`idcicloanyo`) ON ';

```

## 2.16 docente

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre	Tipo	Llave Primaria	Not Null	Auto Increment	Fl	Valor por Defecto	Comentario
iddocente	int(11)	SI	SI	SI			Id correlativo unico para cada docente
apellidos	varchar(200)	NO	SI	NO			Apellidos del docente
nombres	varchar(200)	NO	SI	NO			Nombres del docente
idusuario	int(11)	NO	SI	NO			Referencia al usuario con el que el docente ingresa al sistema
visible	int(11)	NO	SI	NO			Es 1 por defecto. Indica que el usuario esta activo. Al momento de borrar usuarios del sistema, nada mas cambiar este estado a 0.
Index	Columnas						
PRIMARY	iddocente						

fkidusuario_docente	idusuario
---------------------	-----------

**Table Create SQL**

```

CREATE TABLE `docente` (
    `iddocente` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico para cada docente',
    `apellidos` varchar(200) NOT NULL COMENTARIO 'Apellidos del
docente',
    `nombres` varchar(200) NOT NULL COMENTARIO 'Nombres del
docente',
    `idusuario` int(11) NOT NULL COMENTARIO 'Referencia al
usuario con el que el docente ingresa al sistema',
    `visible` int(11) NOT NULL COMENTARIO 'Es 1 por defecto.
Indica que el usuario esta activo. Al momento de borrar
usuarios del sistema, nada mas cambiar este estado a 0.',
    LLAVE PRIMARIA(`iddocente`),
    INDEX `fkidusuario_docente`(`idusuario`),
    CONSTRAINT `fkidusuario_docente` FOREIGN KEY (`idusuario`)
        REFERENCES `usuario`(`idusuario`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idusuario`) REFER
`jhard/usuario`(`idusuario`) ON DELETE';

```

## 2.17 entrada

Nombre del Parámetro	Valor del Parámetro
----------------------	---------------------

Tipo de Tabla	INNODB
Formato de la Fila	DEFAULT
Temporal	NO
Checksum de Fila	NO

Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIn c	Flags	Valor por Defecto	Comentari o
identrada	int(11)	SI	SI	SI	UNSIGNED		
titulo	varchar(50)	NO	SI	NO			
descripcion	text	NO	SI	NO			
fechahora	datetime	NO	SI	NO			
idusuario	int(11)	NO	SI	NO			

Index	Columnas
PRIMARY	identrada
fk_entrada_usuario	idusuario
idxEntrTitulo	titulo
idxEntrFecha	fechahora

**Table Create SQL**

```

CREATE TABLE `entrada` (
    `identrada` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    `titulo` varchar(50) NOT NULL,
    `descripcion` text NOT NULL,
    `fechahora` datetime NOT NULL,
    `idusuario` int(11) NOT NULL,
    LLAVE PRIMARIA(`identrada`),

```

```
INDEX `fk_entrada_usuario`(`idusuario`),
INDEX `idxEntrTitulo`(`titulo`),
INDEX `idxEntrFecha`(`fechahora`),
CONSTRAINT `fk_entrada_usuario` FOREIGN KEY (`idusuario`)
    REFERENCES `usuario`(`idusuario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idusuario`) REFER
`jhard/usuario`(`idusuario`) ON DELETE';
```

## 2.18 equipo

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idequipo	int(11)	SI	SI	SI			Id correlativo unico de cada equipo
idmarca	int(11)	NO	SI	NO			Referencia a la marca que posee este equipo
nombre	varchar(45)	NO	SI	NO			Nombre del equipo
modelo	varchar(15)	NO	SI	NO			Modelo al cual pertenece el equipo
idclasificacion	int(11)	NO	SI	NO			Referencia a la clasificacion a la cual esta relacionado este equipo

<b>Index</b>	<b>Columnas</b>
PRIMARY	idequipo
fkidmarca_equipo	idmarca
fkidclasificacion_equipo	idclasificacion

```
Table Create SQL CREATE TABLE `equipo` (
    `idequipo` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico de cada equipo',
    `idmarca` int(11) NOT NULL COMENTARIO 'Referencia a la marca
que posee este equipo',
    `nombre` varchar(45) NOT NULL COMENTARIO 'Nombre del equipo',
    `modelo` varchar(15) NOT NULL COMENTARIO 'Modelo al cual
pertenece el equipo',
    `idclasificacion` int(11) NOT NULL COMENTARIO 'Referencia a
la clasificacion a la cual esta relacionado este equipo',
    LLAVE PRIMARIA(`idequipo`),
    INDEX `fkidmarca_equipo`(`idmarca`),
    INDEX `fkidclasificacion_equipo`(`idclasificacion`),
    CONSTRAINT `fkidclasificacion_equipo` FOREIGN KEY
(`idclasificacion`)
        REFERENCES `clasificacion`(`idclasificacion`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fkidmarca_equipo` FOREIGN KEY (`idmarca`)
```

```

        REFERENCES `marca`(`idmarca`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
    )
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idclasificacion`) REFER
`jhard/clasificacion`(`idclasifi`;

```

## 2.19 equiposimple

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre de Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idEquipoSimple	int(11)	SI	SI				Id correlativo unico de cada equipo simple
descripcion	text	NO	SI	NO			Descripcion del equipo simple
propietario	varchar(200)	NO	SI	NO			Nombre del propietario del equipo simple
ideestado	int(11)	NO	SI	NO			Referencia al estado en el que se encuentra el equipo simple
Index	Columnas						
PRIMARY		idEquipoSimple					
fkideestado_equiposimple		ideestado					
<b>Table Create SQL</b>	<pre> CREATE TABLE `equiposimple` (     `idEquipoSimple` int(11) NOT NULL AUTO_INCREMENT COMENTARIO     'Id correlativo unico de cada equipo simple',     `descripcion` text NOT NULL COMENTARIO 'Descripcion del equipo simple',     `propietario` varchar(200) NOT NULL COMENTARIO 'Nombre del propietario del equipo simple',     `ideestado` int(11) NOT NULL COMENTARIO 'Referencia al estado en el que se encuentra el equipo simple',     LLAVE PRIMARIA(`idEquipoSimple`),     INDEX `fkideestado_equiposimple`(`ideestado`),     CONSTRAINT `fkideestado_equiposimple` FOREIGN KEY (`ideestado`)         REFERENCES `estadoequipo`(`ideestado`)         ON DELETE NO ACTION         ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB; (`ideestado`) REFER `jhard/estadoequipo`(`ideestado`) ON DEL'; </pre>						

## 2.20 estadocurso

Nombre del Parámetro	Valor del Parámetro
Tipo de Tabla	INNODB
Formato de la Fila	DEFAULT

Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idestadocurso	int(11)	SI	SI	SI			Id correlativo unico para cada estado del curso

Index	Columnas
PRIMARY	idestadocurso

**Table Create SQL**

```

CREATE TABLE `estadocurso` (
  `idestadocurso` int(11) NOT NULL AUTO_INCREMENT COMENTARIO
  'Id correlativo unico para cada estado del curso',
  `nombre` varchar(100) NOT NULL COMENTARIO 'Nombre del estado del curso',
  LLAVE PRIMARIA(`idestadocurso`)
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB';

```

## 2.21 estadoequipo

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
ideestado	int(11)	SI	SI	SI			Id correlativo unico de cada estado
nombre	varchar(45)	NO	SI	NO			Nombre del estado
descripcion	text	NO	NO	NO			Descripcion del estado
Index	Columnas						
PRIMARY	ideestado						

**Table Create SQL**

```

CREATE TABLE `estadoequipo` (
  `ideestado` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico de cada estado',
  `nombre` varchar(45) NOT NULL COMENTARIO 'Nombre del estado',
  `descripcion` text COMENTARIO 'Descripcion del estado',
  LLAVE PRIMARIA(`ideestado`)
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB';

```

## 2.22 estadoreserva

Nombre del Parámetro	Valor del Parámetro
Tipo de Tabla	INNODB

Formato de la Fila	DEFAULT				
Temporal	NO				
Checksum de Fila	NO				
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Flags
idestadorestable	int(11)	SI	SI	SI	
nombre	varchar(100)	NO	SI	NO	
<b>Index</b>		<b>Columnas</b>			
PRIMARY		idestadorestable			
<b>Table Create SQL</b>		<pre>CREATE TABLE `estadorestable` (     `idestadorestable` int(11) NOT NULL AUTO_INCREMENT COMMENT 'Id correlativo unico de cada reserva',     `nombre` varchar(100) NOT NULL COMMENT 'Nombre del estado de la reserva',     PRIMARY KEY(`idestadorestable`) ) ENGINE=INNODB COMMENT = 'InnoDB free: 5120 kB';</pre>			

## 2.23 estudiante

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre de Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIncr	Flags	Valor por Defecto	Comentario
idestudiante	int(11)	SI	SI	SI			Id correlativo unico para cada estudiante
carnet	varchar(7)	NO	SI	NO			Carnet del estudiante, representativo y distintivo en el registro de la facultad
apellidos	varchar(20)	NO	SI	NO			Apellidos del estudiante
nombres	varchar(20)	NO	SI	NO			Nombres del estudiante
idusuario	int(11)	NO	SI	NO			Referencia con la que el usuario ingresa al sistema
visible	int(1)	NO	SI	NO	1		Indica el estado de este estudiante. Es 1 por defecto. Indica que el usuario esta activo. Al momento de borrar usuarios del sistema, nada mas cambiar este estado a 0.
<b>Index</b>							
<b>Columnas</b>							

PRIMARY	idestudiante
fkidusuario_estudiante	idusuario
<b>Table Create SQL</b>	<pre> CREATE TABLE `estudiante` (   `idestudiante` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico para cada estudiante',   `carnet` varchar(7) NOT NULL COMENTARIO 'Carnet del estudiante, representativo y distintivo en el registro de la facultad',   `apellidos` varchar(200) NOT NULL COMENTARIO 'Apellidos del estudiante',   `nombres` varchar(200) NOT NULL COMENTARIO 'Nombres del estudiante',   `idusuario` int(11) NOT NULL COMENTARIO 'Referencia con la que el usuario ingresa al sistema',   `visible` int(11) NOT NULL DEFAULT '1' COMENTARIO 'Indica el estado de este estudiante. Es 1 por defecto. Indica que el usuario esta activo. Al momento de borrar usuarios del sistema, nada mas cambiar este estado a 0.',   LLAVE PRIMARIA(`idestudiante`),   INDEX `fkidusuario_estudiante`(`idusuario`),   CONSTRAINT `fkidusuario_estudiante` FOREIGN KEY (`idusuario`)     REFERENCES `usuario`(`idusuario`)     ON DELETE NO ACTION     ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB; (`idusuario`) REFER `jhard/usuario`(`idusuario`) ON DELETE'; </pre>

## 2.24 existencia

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idexistencia	int(11)	SI	SI	SI			Id correlativo unico para cada existencia
idhardware	int(11)	NO	SI	NO			Referencia al hardware al cual pertenece esta existencia
idubicacion	int(11)	NO	SI	NO			Referencia a la ubicacion donde se encuentra localizada esta existencia
idestado	int(11)	NO	SI	NO			Referencia al estado en el que se encuentra esta existencia
codigo	varchar(45)	NO	SI	NO			Codigo con el cual clasificar esta existencia en el inventario
Index	Columnas						
PRIMARY	idexistencia						
fkidhardware_existencia	idhardware						
fkidubicacion_existencia	idubicacion						

fkidestado_existencia	ideestado
<b>Table Create SQL</b>	
<pre> CREATE TABLE `existencia` (     `idexistencia` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico para cada existencia',     `idhardware` int(11) NOT NULL COMENTARIO 'Referencia al hardware al cual pertenece esta existencia',     `idubicacion` int(11) NOT NULL COMENTARIO 'Referencia a la ubicacion donde se encuentra localizada esta existencia',     `ideestado` int(11) NOT NULL COMENTARIO 'Referencia al estado en el que se encuentra esta existencia',     `codigo` varchar(45) NOT NULL COMENTARIO 'Codigo con el cual clasificar esta existencia en el inventario',     LLAVE PRIMARIA(`idexistencia`),     INDEX `fkidhardware_existencia`(`idhardware`),     INDEX `fkidubicacion_existencia`(`idubicacion`),     INDEX `fkidestado_existencia`(`ideestado`),     CONSTRAINT `fkidestado_existencia` FOREIGN KEY (`ideestado`)         REFERENCES `estadoequipo`(`ideestado`)         ON DELETE NO ACTION         ON UPDATE NO ACTION,     CONSTRAINT `fkidhardware_existencia` FOREIGN KEY (`idhardware`)         REFERENCES `equipo`(`idequipo`)         ON DELETE NO ACTION         ON UPDATE NO ACTION,     CONSTRAINT `fkidubicacion_existencia` FOREIGN KEY (`idubicacion`)         REFERENCES `ubicacion`(`idubicacion`)         ON DELETE NO ACTION         ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB; (`ideestado`) REFER `jhard/estadoequipo`(`ideestado`) ON DEL'; </pre>	

## 2.25 facultad

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoI nc	Flag s	Valor por Defecto	Comentario
idfacultad	int(11)	SI	SI	SI			Id correlativo unico de cada facultad
nombre	varchar(200 )	NO	SI	NO			Nombre de la facultad
Index	Columnas						
PRIMARY	idfacultad						
<b>Table Create SQL</b>		<pre> CREATE TABLE `facultad` (     `idfacultad` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada facultad',     `nombre` varchar(200) NOT NULL COMENTARIO 'Nombre de la facultad', </pre>					

```

        LLAVE PRIMARIA(`idfacultad`)
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB';

```

## 2.26 horario

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idhorario	int(11)	SI	SI	SI			Id correlativo unico para cada horario
diaseman a	int(11)	NO	SI	NO			Dia de la semana que se brinda el curso (1= lunes, 7= domingo)
horainicio	time	NO	SI	NO			Hora a la que da inicio el curso
horafin	time	NO	SI	NO			Hora a la que finaliza el curso
idcurso	int(11)	NO	SI	NO			Referencia al curso relacionado con este horario
idaula	int(11)	NO	SI	NO			Aula en la que se imparte este curso en este horario

Index	Columns
PRIMARY	idhorario
fkidcurso_horario	idcurso
fkidaula_horario	idaula

**Table Create SQL**

```

CREATE TABLE `horario` (
  `idhorario` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico para cada horario',
  `diasemana` int(11) NOT NULL COMENTARIO 'Dia de la semana que
se brinda el curso (1= lunes, 7= domingo)',
  `horainicio` time NOT NULL COMENTARIO 'Hora a la que da
inicio el curso',
  `horafin` time NOT NULL COMENTARIO 'Hora a la que finaliza el
curso',
  `idcurso` int(11) NOT NULL COMENTARIO 'Referencia al curso
relacionado con este horario',
  `idaula` int(11) NOT NULL COMENTARIO 'Aula en la que se
imparte este curso en este horario',
  LLAVE PRIMARIA(`idhorario`),
  INDEX `fkidcurso_horario`(`idcurso`),
  INDEX `fkidaula_horario`(`idaula`),
  CONSTRAINT `fkidaula_horario` FOREIGN KEY (`idaula`)
    REFERENCES `ubicacion`(`idubicacion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fkidcurso_horario` FOREIGN KEY (`idcurso`)
    REFERENCES `curso`(`idcurso`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)

```

```
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idaula`) REFER
`jhard/ubicacion`(`idubicacion`) ON DELETE';
```

## 2.27 inscripcion

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idinscripcion	int(11)	SI	SI	SI			Id correlativo unico para cada inscripcion
idcurso	int(11)	NO	SI	NO			Referencia al curso al cual se inscribio el estudiante
idestudiente	int(11)	NO	SI	NO			Referencia al estudiante inscrito en este curso
Index	Columnas						
PRIMARY		idinscripcion					
fkidcurso_inscripcion		idcurso					
fkidestudiente_inscripcion		idestudiente					
<b>Table Create SQL</b>	<pre>CREATE TABLE `inscripcion` (   `idinscripcion` int(11) NOT NULL AUTO_INCREMENT COMENTARIO   'Id correlativo unico para cada inscripcion',   `idcurso` int(11) NOT NULL COMENTARIO 'Referencia al curso al   cual se inscribio el estudiante',   `idestudiente` int(11) NOT NULL COMENTARIO 'Referencia al   estudiante inscrito en este curso',   LLAVE PRIMARIA(`idinscripcion`),   INDEX `fkidcurso_inscripcion`(`idcurso`),   INDEX `fkidestudiente_inscripcion`(`idestudiente`),   CONSTRAINT `fkidcurso_inscripcion` FOREIGN KEY (`idcurso`)     REFERENCES `curso`(`idcurso`)     ON DELETE NO ACTION     ON UPDATE NO ACTION,   CONSTRAINT `fkidestudiente_inscripcion` FOREIGN KEY   (`idestudiente`)     REFERENCES `estudiante`(`idestudiente`)     ON DELETE NO ACTION     ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB; (`idcurso`) REFER `jhard/curso`(`idcurso`) ON DELETE NO AC';</pre>						

## 2.28 instalacion

Nombre del Parámetro	Valor del Parámetro
Tipo de Tabla	INNODB
Formato de la Fila	DEFAULT
Temporal	NO

Checksum de Fila		NO								
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoI nc	Fla gs	Valor por Defecto	Comentario			
idinstalacion	int(11)	SI	SI	SI			Id correlativo unico de cada instalacion			
idsoftware	int(11)	NO	SI	NO			Referencia al software instalado			
fechainstalacion	date	NO	SI	NO			Fecha en la que se realizo la instalacion			
idequipoexistente	int(11)	NO	SI	NO			Referencia al equipo donde se instalo el software			
Index		Columnas								
PRIMARY		idinstalacion								
fkidsoftware_instalacion		idsoftware								
fkidequipoexistente_instalacion		idequipoexistente								
<b>Table Create SQL</b>	<pre> CREATE TABLE `instalacion` (   `idinstalacion` int(11) NOT NULL AUTO_INCREMENT COMENTARIO   'Id correlativo unico de cada instalacion',   `idsoftware` int(11) NOT NULL COMENTARIO 'Referencia al software instalado',   `fechainstalacion` date NOT NULL COMENTARIO 'Fecha en la que se realizo la instalacion',   `idequipoexistente` int(11) NOT NULL COMENTARIO 'Referencia al equipo donde se instalo el software',   LLAVE PRIMARIA(`idinstalacion`),   INDEX `fkidsoftware_instalacion`(`idsoftware`),   INDEX `fkidequipoexistente_instalacion`(`idequipoexistente`),   CONSTRAINT `fkidequipoexistente_instalacion` FOREIGN KEY (`idequipoexistente`)   REFERENCES `existencia`(`idexistencia`)   ON DELETE NO ACTION   ON UPDATE NO ACTION,   CONSTRAINT `fkidsoftware_instalacion` FOREIGN KEY (`idsoftware`)   REFERENCES `software`(`idsoftware`)   ON DELETE NO ACTION   ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB; (`idequipoexistente`) REFER `jhard/existencia`(`idexistenc'; </pre>									

## 2.29 instructor

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombr e de Columna	Tipo de Dato	Llave Primaria	Not Null	Au toI nc	Fl a g s	Valor por Defecto	Comentario

idinstrutor	int(11)	SI	SI	SI		Id correlativo unico para cada instructor
carnet	varchar(7)	NO	SI	NO		Carnet con el cual se encuentra registrado en adacad
apellidos	varchar(200)	NO	SI	NO		Apellidos del instructor
nombres	varchar(200)	NO	SI	NO		Nombres del instructor
idusuario	int(11)	NO	SI	NO		Referencia al usuario con el que instructor accede al sistema
visible	int(11)	NO	SI	NO	1	Estado del instructor. Por defecto es 1, lo cual significa que esta en uso. Al momento de borrar un instructor, nada mas se cambia este campo a 0.

Index	Columnas
PRIMARY	idinstrutor
fkidusuario_instructor	idusuario

**Table Create SQL**

```

CREATE TABLE `instructor` (
    `idinstrutor` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico para cada instructor',
    `carnet` varchar(7) NOT NULL COMENTARIO 'Carnet con el cual
se encuentra registrado en adacad',
    `apellidos` varchar(200) NOT NULL COMENTARIO 'Apellidos del
instructor',
    `nombres` varchar(200) NOT NULL COMENTARIO 'Nombres del
instructor',
    `idusuario` int(11) NOT NULL COMENTARIO 'Referencia al
usuario con el que instructor accede al sistema',
    `visible` int(11) NOT NULL DEFAULT '1' COMENTARIO 'Estado del
instructor. Por defecto es 1, lo cual significa que esta en
uso. Al momento de borrar un instructor, nada mas se cambia
este campo a 0.',
    LLAVE PRIMARIA(`idinstrutor`),
    INDEX `fkidusuario_instructor`(`idusuario`),
    CONSTRAINT `fkidusuario_instructor` FOREIGN KEY (`idusuario`)
        REFERENCES `usuario`(`idusuario`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idusuario`) REFER
`jhard/usuario`(`idusuario`) ON DELETE';

```

## 2.30 mantenimiento

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Column	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Fla gs	Valor por Defecto	Comentario

a						
idmantenimiento	int(11)	SI	SI	SI		Id correlativo unico de cada mantenimiento
fecha	date	NO	SI	NO		Fecha en la que se efectuo el mantenimiento
descripcion	text	NO	SI	NO		Descripcion del mantenimiento
idtecnico	int(11)	NO	SI	NO		Referencia al tecnico que efectuo el mantenimiento
idsolicitud	int(11)	NO	NO	NO		Referencia a la solicitud de mantenimiento realizada, en caso de existir una
idequipoexistente	int(11)	NO	NO	NO		Referencia al equipo al cual se efectuo el mantenimiento
idequiposimple	int(11)	NO	NO	NO		
estado	text	NO	SI	NO		

Index	Columnas
PRIMARY	idmantenimiento
fkidtecnico_mantenimiento	idtecnico
fkidsolicitud_mantenimiento	idsolicitud
fkidequipoexistente_mantenimiento	idequipoexistente
fkidequiposimple_mantenimiento	idequiposimple

**Table Create SQL**

```

CREATE TABLE `mantenimiento` (
  `idmantenimiento` int(11) NOT NULL AUTO_INCREMENT COMENTARIO
  'Id correlativo unico de cada mantenimiento',
  `fecha` date NOT NULL COMENTARIO 'Fecha en la que se efectuo
  el mantenimiento',
  `descripcion` text NOT NULL COMENTARIO 'Descripcion del
  mantenimiento',
  `idtecnico` int(11) NOT NULL COMENTARIO 'Referencia al
  tecnico que efectuo el mantenimiento',
  `idsolicitud` int(11) COMENTARIO 'Referencia a la solicitud
  de mantenimiento realizada, en caso de existir una',
  `idequipoexistente` int(11) COMENTARIO 'Referencia al equipo
  al cual se efectuo el mantenimiento',
  `idequiposimple` int(11),
  `estado` text NOT NULL,
  LLAVE PRIMARIA(`idmantenimiento`),
  INDEX `fkidtecnico_mantenimiento`(`idtecnico`),
  INDEX `fkidsolicitud_mantenimiento`(`idsolicitud`),
  INDEX
  `fkidequipoexistente_mantenimiento`(`idequipoexistente`),
  INDEX `fkidequiposimple_mantenimiento`(`idequiposimple`),
  CONSTRAINT `fkidequipoexistente_mantenimiento` FOREIGN KEY
  (`idequipoexistente`)
    REFERENCES `existencia`(`idexistencia`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fkidequiposimple_mantenimiento` FOREIGN KEY
  (`idequiposimple`)
    REFERENCES `equiposimple`(`idEquipoSimple`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
)

```

```

CONSTRAINT `fkidsolicitud_mantenimiento` FOREIGN KEY
(`idsolicitud`)
    REFERENCES `solicitud`(`idsolicitud`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fkidtecnico_mantenimiento` FOREIGN KEY
(`idtecnico`)
    REFERENCES `tecnico`(`idtecnico`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idequipoexistente`) REFER
`jhard/existencia`(`idexistenc`;

```

### 2.31 marca

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idmarca	int(11)	SI	SI	SI			Id correlativo unico de cada marca
nombre	varchar(100 )	NO	SI	NO			Nombre de la marca
Index	Columnas						
PRIMARY	idmarca						
<b>Table Create SQL</b>	<pre> CREATE TABLE `marca` ( `idmarca` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada marca', `nombre` varchar(100) NOT NULL COMENTARIO 'Nombre de la marca', LLAVE PRIMARIA(`idmarca`) ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB'; </pre>						

### 2.32 materia

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Fla gs	Valor por Defecto	Comentario
idmateria	int(11)	SI	SI	SI			Id correlativo unico de cada materia
codigo	varchar(7)	NO	SI	NO			Codigo de la materia, con el cual se identifica en adacad

nombre	varchar(200)	NO	SI	NO	Nombre de la materia
idcarrera	int(11)	NO	SI	NO	Referencia a la carrera a la cual pertenece esta materia
Index	<b>Columnas</b>				
PRIMARY	idmateria				
fkidcarrera_materia	idcarrera				
<b>Table Create SQL</b>		<pre>CREATE TABLE `materia` (     `idmateria` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada materia',     `codigo` varchar(7) NOT NULL COMENTARIO 'Codigo de la materia, con el cual se identifica en adacad',     `nombre` varchar(200) NOT NULL COMENTARIO 'Nombre de la materia',     `idcarrera` int(11) NOT NULL COMENTARIO 'Referencia a la carrera a la cual pertenece esta materia',     LLAVE PRIMARIA(`idmateria`),     INDEX `fkidcarrera_materia`(`idcarrera`),     CONSTRAINT `fkidcarrera_materia` FOREIGN KEY (`idcarrera`)         REFERENCES `carrera`(`idcarrera`)         ON DELETE NO ACTION         ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB; (`idcarrera`) REFER `jhard/carrera`(`idcarrera`) ON DELETE';</pre>			

### 2.33 pieza

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idpieza	int(11)	SI	SI	SI			Id correlativo unico de cada pieza
nombre	varchar(100)	NO	SI	NO			Nombre de la pieza
idmarca	int(11)	NO	SI	NO			Referencia a la marca de la pieza
modelo	varchar(15)	NO	SI	NO			Modelo de la pieza
idclasificacion	int(11)	NO	SI	NO			Referencia a la clasificacion en la que se encuentra la pieza
idexistencia	int(11)	NO	NO	NO			
Index	<b>Columnas</b>						
PRIMARY	idpieza						

fkidclasificacion_pieza	idclasificacion
fkidmarca_pieza	idmarca
fkidexistencia_pieza	idexistencia

**Table Create SQL**

```

CREATE TABLE `pieza` (
    `idpieza` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada pieza',
    `nombre` varchar(100) NOT NULL COMENTARIO 'Nombre de la pieza',
    `idmarca` int(11) NOT NULL COMENTARIO 'Referencia a la marca de la pieza',
    `modelo` varchar(15) NOT NULL COMENTARIO 'Modelo de la pieza',
    `idclasificacion` int(11) NOT NULL COMENTARIO 'Referencia a la clasificacion en la que se encuentra la pieza',
    `idexistencia` int(11),
    LLAVE PRIMARIA(`idpieza`),
    INDEX `fkidclasificacion_pieza`(`idclasificacion`),
    INDEX `fkidmarca_pieza`(`idmarca`),
    INDEX `fkidexistencia_pieza`(`idexistencia`),
    CONSTRAINT `fkidclasificacion_pieza` FOREIGN KEY
(`idclasificacion`)
    REFERENCES `clasificacion`(`idclasificacion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT `fkidexistencia_pieza` FOREIGN KEY
(`idexistencia`)
    REFERENCES `existencia`(`idexistencia`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT `fkidmarca_pieza` FOREIGN KEY (`idmarca`)
    REFERENCES `marca`(`idmarca`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idclasificacion`) REFER
`jhard/clasificacion`(`idclasifi`';

```

## 2.34 reserva

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	Auto Inc	Flags	Valor por Defecto	Comentario
idreserva	int(11)	SI	SI	SI			Id correlativo unico de cada reserva
fechareserva	date	NO	SI	NO			fecha en la que se reservo el equipo
fechahorainicio	datetime	NO	SI	NO			Fecha y hora inicial a la que se utilizara el equipo
fechahorafinprestamo	datetime	NO	SI	NO			Fecha y hora final a la que se utilizara el equipo

idubicacion	int(11)	NO	SI	NO	Referencia al aula donde se utilizara el equipo
idequipoexistente	int(11)	NO	SI	NO	Referencia al equipo que se utilizara
idusuario	int(11)	NO	SI	NO	Referencia al usuario que registro la reserva
ideestado	int(11)	NO	SI	NO	Referencia al estado en el que se encuentra esta reserva
descripcion	text	NO	SI	NO	Descripcion y justificacion de la reserva
iddocente	int(11)	NO	SI	NO	
<b>Index</b>		<b>Columnas</b>			
PRIMARY		idreserva			
fkideestado_reserva		ideestado			
fkidequipoexistente_reserva		idequipoexistente			
fkidubicacion_reserva		idubicacion			
fkidusuario_reserva		idusuario			
fkiddocente_reserva		iddocente			

**Table Create SQL**

```

CREATE TABLE `reserva` (
  `idreserva` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada reserva',
  `fechareserva` date NOT NULL COMENTARIO 'fecha en la que se reservo el equipo',
  `fechahorainicioprestamo` datetime NOT NULL COMENTARIO 'Fecha y hora inicial a la que se utilizara el equipo',
  `fechahorafinprestamo` datetime NOT NULL COMENTARIO 'Fecha y hora final a la que se utilizara el equipo',
  `idubicacion` int(11) NOT NULL COMENTARIO 'Referencia al aula donde se utilizara el equipo',
  `idequipoexistente` int(11) NOT NULL COMENTARIO 'Referencia al equipo que se utilizara',
  `idusuario` int(11) NOT NULL COMENTARIO 'Referencia al usuario que registro la reserva',
  `ideestado` int(11) NOT NULL COMENTARIO 'Referencia al estado en el que se encuentra esta reserva',
  `descripcion` text NOT NULL COMENTARIO 'Descripcion y justificacion de la reserva',
  `iddocente` int(11) NOT NULL,
  LLAVE PRIMARIA(`idreserva`),
  INDEX `fkideestado_reserva`(`ideestado`),
  INDEX `fkidequipoexistente_reserva`(`idequipoexistente`),
  INDEX `fkidubicacion_reserva`(`idubicacion`),
  INDEX `fkidusuario_reserva`(`idusuario`),
  INDEX `fkiddocente_reserva`(`iddocente`),
  CONSTRAINT `fkiddocente_reserva` FOREIGN KEY (`iddocente`)
    REFERENCES `docente`(`iddocente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fkidequipoexistente_reserva` FOREIGN KEY
(`idequipoexistente`)
    REFERENCES `existencia`(`idexistencia`)
    ON DELETE NO ACTION
)

```

```

        ON UPDATE NO ACTION,
        CONSTRAINT `fkidestado_reserva` FOREIGN KEY (`idestado`)
            REFERENCES `estadoreserva`(`idestadoreserva`)
            ON DELETE NO ACTION
            ON UPDATE NO ACTION,
        CONSTRAINT `fkidubicacion_reserva` FOREIGN KEY
(`idubicacion`)
            REFERENCES `ubicacion`(`idubicacion`)
            ON DELETE NO ACTION
            ON UPDATE NO ACTION,
        CONSTRAINT `fkidusuario_reserva` FOREIGN KEY (`idusuario`)
            REFERENCES `usuario`(`idusuario`)
            ON DELETE NO ACTION
            ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`iddocente`) REFER
`jhard/docente`(`iddocente`) ON DELETE';

```

### 2.35 rol

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoI nc	Flag s	Valor por Defecto	Comentario
idrol	int(11)	SI	SI	SI			Id correlativo unico para cada rol
nombre	varchar(100)	NO	SI	NO			Nombre del rol
descripcion	text	NO	SI	NO			Descripcion del rol
Index	Columnas						
PRIMARY	idrol						

**Table Create SQL**

```

CREATE TABLE `rol` (
`idrol` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico para cada rol',
`nombre` varchar(100) NOT NULL COMENTARIO 'Nombre del rol',
`descripcion` text NOT NULL COMENTARIO 'Descripcion del rol',
LLAVE PRIMARIA(`idrol`)
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB';

```

### 2.36 software

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre	Tipo de	Llave	Not	Auto	Fla	Valor por	Comentario

Columna	Datos	Primaria	Null	Inc	gs	Defecto
idsoftware	int(11)	SI	SI	SI		Id correlativo unico de cada software
nombre	varchar(100)	NO	SI	NO		Nombre del software
version	varchar(15)	NO	SI	NO		Version del software
codigolicencia	varchar(45)	NO	NO	NO		codigo de la licencia del software (en caso de poseer alguno)
cantidadlicencias	int(11)	NO	NO	NO		Cantidad de licencias disponibles para instalar (en caso de poseer licencias)
idclasificacion	int(11)	NO	SI	NO		Referencia a la clasificacion que posee este software
Index	Columnas					
PRIMARY	idsoftware					
fkidclasificacion_software	idclasificacion					

Table Create SQL	CREATE TABLE `software` ( `idsoftware` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada software', `nombre` varchar(100) NOT NULL COMENTARIO 'Nombre del software', `version` varchar(15) NOT NULL COMENTARIO 'Version del software', `codigolicencia` varchar(45) COMENTARIO 'codigo de la licencia del software (en caso de poseer alguno)', `cantidadlicencias` int(11) COMENTARIO 'Cantidad de licencias disponibles para instalar (en caso de poseer licencias)', `idclasificacion` int(11) NOT NULL COMENTARIO 'Referencia a la clasificacion que posee este software', LLAVE PRIMARIA(`idsoftware`), INDEX `fkidclasificacion_software`(`idclasificacion`), CONSTRAINT `fkidclasificacion_software` FOREIGN KEY(`idclasificacion`) REFERENCES `clasificacion`(`idclasificacion`) ON DELETE NO ACTION ON UPDATE NO ACTION ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB; (`idclasificacion`) REFER `jhard/clasificacion`(`idclasi';
------------------	--

## 2.37 solicitud

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre	Tipo	Llave	Not	Aut	Fl	Valor	Comentario
Columna	de	Primari	Null	oIn	ag	por	
a	Datos	ia	c	s	Defecto		

idsolicitud	int(11)	SI	SI	SI	Id correlativo unico de cada solicitud
fecha	date	NO	SI	NO	Fecha en la que se registro la solicitud
prioridad	varchar(25)	NO	SI	NO	Tipo de prioridad en la cual se clasifican las solicitudes de mantenimiento. Sus posibles valores son: Alta, Media y Baja
descripcion	text	NO	SI	NO	Descripcion de la solicitud
idusuario	int(11)	NO	SI	NO	Usuario que registro la solicitud
idequipoexistente	int(11)	NO	NO	NO	Equipo al cual se desea efectuar un mantenimiento
idequiposimple	int(11)	NO	NO	NO	

Index	Columnas
PRIMARY	idsolicitud
fkidequipoexistente_solicitud	idequipoexistente
fkidequiposimple_solicitud	idequiposimple
fkidusuario_solicitud	idusuario

**Table Create SQL**

```

CREATE TABLE `solicitud` (
    `idsolicitud` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico de cada solicitud',
    `fecha` date NOT NULL COMENTARIO 'Fecha en la que se registro
la solicitud',
    `prioridad` varchar(25) NOT NULL COMENTARIO 'Tipo de
prioridad en la cual se clasifican las solicitudes de
mantenimiento. Sus posibles valores son: Alta, Media y Baja',
    `descripcion` text NOT NULL COMENTARIO 'Descripcion de la
solicitud',
    `idusuario` int(11) NOT NULL COMENTARIO 'Usuario que registro
la solicitud',
    `idequipoexistente` int(11) COMENTARIO 'Equipo al cual se
desea efectuar un mantenimiento',
    `idequiposimple` int(11),
    LLAVE PRIMARIA(`idsolicitud`),
    INDEX `fkidequipoexistente_solicitud`(`idequipoexistente`),
    INDEX `fkidequiposimple_solicitud`(`idequiposimple`),
    INDEX `fkidusuario_solicitud`(`idusuario`),
    CONSTRAINT `fkidequipoexistente_solicitud` FOREIGN KEY
(`idequipoexistente`)
        REFERENCES `existencia`(`idexistencia`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fkidequiposimple_solicitud` FOREIGN KEY
(`idequiposimple`)
        REFERENCES `equiposimple`(`idEquipoSimple`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fkidusuario_solicitud` FOREIGN KEY (`idusuario`)
        REFERENCES `usuario`(`idusuario`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idequipoexistente`) REFER

```

```
`jhard/existencia`(`idexistenc`;
```

### 2.38 tag

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIn c	Flags	Valor por Defecto	Comentario
idtag	int(11)	SI	SI	SI	UNSIGNED		
descripcion	varchar(25)	NO	SI	NO			
Index	Columnas						
PRIMARY	idtag						
idxTagDesc	descripcion						
<b>Table Create SQL</b>		<pre>CREATE TABLE `tag` (   `idtag` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,   `descripcion` varchar(25) NOT NULL,   LLAVE PRIMARIA(`idtag`),   INDEX `idxTagDesc`(`descripcion`) ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB';</pre>					

### 2.39 tag\_entrada

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoIn c	Flags	Valor por Defecto	Comentario
idtagentrada	int(10)	SI	SI	SI	UNSIGNED		
idtag	int(11)	SI	SI	NO	UNSIGNED		
identrada	int(11)	SI	SI	NO	UNSIGNED		
Index	Columnas						
PRIMARY	idtagentrada,idtag,identrada						
fk_tag_entrada_tag	idtag						
fk_tag_entrada_entrad a	identrada						

idxTagEntrada	idtag,identrada
<b>Table Create SQL</b>	

```

CREATE TABLE `tag_entrada` (
  `idtagentrada` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `idtag` int(11) UNSIGNED NOT NULL,
  `identrada` int(11) UNSIGNED NOT NULL,
  LLAVE PRIMARIA(`idtagentrada`, `idtag`, `identrada`),
  INDEX `fk_tag_entrada_tag`(`idtag`),
  INDEX `fk_tag_entrada_entrada`(`identrada`),
  INDEX `idxTagEntrada`(`idtag`, `identrada`),
  CONSTRAINT `fk_tag_entrada_entrada` FOREIGN KEY (`identrada`)
    REFERENCES `entrada`(`identrada`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_tag_entrada_tag` FOREIGN KEY (`idtag`)
    REFERENCES `tag`(`idtag`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`identrada`) REFER
`jhard/entrada`(`identrada`) ON UPDATE';

```

## 2.40 tecnico

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idtecnico	int(11)	SI	SI	SI			Id correlativo unico para cada tecnico
apellidos	varchar(200 )	NO	SI	NO			Apellidos del tecnico
nombres	varchar(200 )	NO	SI	NO			Nombres del tecnico
cargo	varchar(200 )	NO	SI	NO			Cargo que desempenia el tecnico
Index	Columnas						
PRIMARY	idtecnico						

<b>Table Create SQL</b>	
<pre> CREATE TABLE `tecnico` (   `idtecnico` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico para cada tecnico',   `apellidos` varchar(200) NOT NULL COMENTARIO 'Apellidos del tecnico',   `nombres` varchar(200) NOT NULL COMENTARIO 'Nombres del tecnico',   `cargo` varchar(200) NOT NULL COMENTARIO 'Cargo que desempenia el tecnico',   LLAVE PRIMARIA(`idtecnico`) ) ENGINE=INNODB </pre>	

```
COMENTARIO = 'InnoDB free: 5120 kB';
```

## 2.41 ubicacion

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idubicacion	int(11)	SI	SI	SI			Id correlativo unico de cada ubicacion
nombre	varchar(45)	NO	NO	NO			Nombre de la ubicacion
Index	Columnas						
PRIMARY	idubicacion						
<b>Table Create SQL</b>		<pre>CREATE TABLE `ubicacion` (     `idubicacion` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id correlativo unico de cada ubicacion',     `nombre` varchar(45) COMENTARIO 'Nombre de la ubicacion',     LLAVE PRIMARIA(`idubicacion`) ) ENGINE=INNODB COMENTARIO = 'InnoDB free: 5120 kB';</pre>					

## 2.42 usuario

Nombre del Parámetro	Valor del Parámetro						
Tipo de Tabla	INNODB						
Formato de la Fila	DEFAULT						
Temporal	NO						
Checksum de Fila	NO						
Nombre de Columna	Tipo de Datos	Llave Primaria	Not Null	AutoInc	Flags	Valor por Defecto	Comentario
idusuario	int(11)	SI	SI	SI			Id correlativo unico para cada usuario
nombre	varchar(25)	NO	SI	NO			Nombre del usuario
clave	varchar(35)	NO	SI	NO			Clave de acceso del usuario
idrol	int(11)	NO	NO	NO			Referencia al rol que juega este usuario dentro del sistema, el cual define los modulos y acciones a las que tiene acceso
idautorizacion	int(10)	NO	NO	NO	UNSIGNED		

Index	Columns
PRIMARY	idusuario
fkidrol_usuario	idrol
fkidautorizacion_usuario	idautorizacion

**Table Create SQL**

```

CREATE TABLE `usuario` (
    `idusuario` int(11) NOT NULL AUTO_INCREMENT COMENTARIO 'Id
correlativo unico para cada usuario',
    `nombre` varchar(25) NOT NULL COMENTARIO 'Nombre del
usuario',
    `clave` varchar(35) NOT NULL COMENTARIO 'Clave de acceso del
usuario',
    `idrol` int(11) COMENTARIO 'Referencia al rol que juega este
usuario dentro del sistema, el cual define los modulos y
acciones a las que tiene acceso',
    `idautorizacion` int(10) UNSIGNED,
    LLAVE PRIMARIA(`idusuario`),
    INDEX `fkidrol_usuario`(`idrol`),
    INDEX `fkidautorizacion_usuario`(`idautorizacion`),
    CONSTRAINT `fkidautorizacion_usuario` FOREIGN KEY
(`idautorizacion`)
    REFERENCES `autorizacion`(`idautorizacion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT `fkidrol_usuario` FOREIGN KEY (`idrol`)
    REFERENCES `rol`(`idrol`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE=INNODB
COMENTARIO = 'InnoDB free: 5120 kB; (`idautorizacion`) REFER
`jhard/autorizacion`(`idautorizac';

```

## 5. Referencias

### 5.1 fkidclasificacion\_accesorio

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
clasificacion	accesorio	NO ACTION	NO ACTION	idclasificacion=idclasificacion

### 5.2 fkidexistencia\_accesorio

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
existencia	accesorio	NO ACTION	NO ACTION	idexistencia=idexistencia

### 5.3 fkidmarca\_accesorio

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
marca	accesorio	NO ACTION	NO ACTION	idmarca=idmarca

### 5.4 fkidusuario\_administrador

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	administrador	NO ACTION	NO ACTION	idusuario=idusu

ario

**5.5 fkarticulousuario**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	articulos	NO ACTION	NO ACTION	idusuario=idusuario

**5.6 fkidclase\_asistencia**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
clase	asistencia	NO ACTION	NO ACTION	idclase=idclase

**5.7 fkidequipoexistente\_asistencia**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
existencia	asistencia	NO ACTION	NO ACTION	ideexistencia=idequipoe xistente

**5.8 fkidestudiante\_asistencia**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
estudiante	asistencia	NO ACTION	NO ACTION	ideestudiante=idestu diante

**5.9 fkidaccesorio\_atributohardware**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
accesorio	atributohardware	NO ACTION	NO ACTION	idaccesorio=idacce sorio

**5.10 fkidequipo\_atributohardware**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
equipo	atributohardware	NO ACTION	NO ACTION	idequipo=idhard ware

**5.11 fkidpieza\_atributohardware**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
pieza	atributohardware	NO ACTION	NO ACTION	idpieza=idpieza

**5.12 fkidusuario\_bitacoracambiosusuario**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	bitacoracambiosusuario	NO ACTION	NO ACTION	idusuario=idusuario

**5.13 fkidequipoexistente\_bitacoraestados**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
existencia	bitacoraesta	NO ACTION	NO ACTION	ideexistencia=idequipoe

Relaciones existentes				
<b>5.14 fkidequiposimple_bitacoraestados</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
equiposimple	bitacoraestados	NO ACTION	NO ACTION	idequipoSimple=idequiposimple
<b>5.15 fkidestado_bitacoraestados</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
estadoequipo	bitacoraestados	NO ACTION	NO ACTION	idestado=idestado
<b>5.16 fkidfacultad_carrera</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
facultad	carrera	NO ACTION	NO ACTION	idfacultad=idfacultad
<b>5.17 fkiddocente_clase</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
docente	clase	NO ACTION	NO ACTION	iddocente=iddocente
<b>5.18 fkidhorario_clase</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
horario	clase	NO ACTION	NO ACTION	idhorario=idhorario
<b>5.19 fkidinstructor_clase</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
instructor	clase	NO ACTION	NO ACTION	idinstructor=idinstructor
<b>5.20 fk_comentarios_entrada</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
entrada	comentarios	NO ACTION	NO ACTION	identrada=identrada
<b>5.21 fkciclo_curso</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
cicloanyo	curso	NO ACTION	NO ACTION	idcicloanyo=idcicloanyo
<b>5.22 fkiddocente_curso</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
docente	curso	NO ACTION	NO ACTION	iddocente=iddocente

nte

**5.23 fkidestado\_curso**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
estadocurso	curso	NO ACTION	NO ACTION	idestdadocurso=ides tado

**5.24 fkidinstructor\_curso**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
instructor	curso	NO ACTION	NO ACTION	idinstructor=idinstr uctor

**5.25 fkidmateria\_curso**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
materia	curso	NO ACTION	NO ACTION	idmateria=idmat eria

**5.26 fkidusuario\_docente**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	docente	NO ACTION	NO ACTION	idusuario=idusu ario

**5.27 fk\_entrada\_usuario**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	entrada	NO ACTION	NO ACTION	idusuario=idusu ario

**5.28 fkidclasificacion\_equipo**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
clasificació n	equipo	NO ACTION	NO ACTION	idclasificacion=idclasifi cacion

**5.29 fkidmarca\_equipo**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
marca	equipo	NO ACTION	NO ACTION	idmarca=idmar ca

**5.30 fkidestado\_equiposimple**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
estadoequip o	equiposimple	NO ACTION	NO ACTION	idestado=idesta do

**5.31 fkidusuario\_estudiante**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	estudiante	NO ACTION	NO ACTION	idusuario=idusu

ario

**5.32 fkideestado\_existencia**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
estadoequipo	existencia	NO ACTION	NO ACTION	ideestado=idesta do

**5.33 fkidhardware\_existencia**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
equipo	existencia	NO ACTION	NO ACTION	idequipo=idhardw are

**5.34 fkidubicacion\_existencia**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
ubicacion	existencia	NO ACTION	NO ACTION	idubicacion=idubic acion

**5.35 fkidaula\_horario**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
ubicacion	horario	NO ACTION	NO ACTION	idubicacion=idala ula

**5.36 fkidcurso\_horario**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
curso	horario	NO ACTION	NO ACTION	idcurso=idcur so

**5.37 fkidcurso\_inscripcion**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
curso	inscripcion	NO ACTION	NO ACTION	idcurso=idcur so

**5.38 fkidestudiante\_inscripcion**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
estudiante	inscripcion	NO ACTION	NO ACTION	idestudiante=idestu diante

**5.39 fkidequipoexistente\_instalacion**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
existencia	instalacion	NO ACTION	NO ACTION	idexistencia=idequipoe xistente

**5.40 fkidsoftware\_instalacion**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
software	instalacion	NO ACTION	NO ACTION	idsoftware=idsoft

ware

**5.41 fkidusuario\_instructor**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	instructor	NO ACTION	NO ACTION	idusuario=idusuario

**5.42 fkidequipoexistente\_mantenimiento**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
existencia	mantenimiento	NO ACTION	NO ACTION	idexistencia=idequipoexistente

**5.43 fkidequiposimple\_mantenimiento**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
equiposimple	mantenimiento	NO ACTION	NO ACTION	idEquipoSimple=idequiposimple

**5.44 fkidsolicitud\_mantenimiento**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
solicitud	mantenimiento	NO ACTION	NO ACTION	idsolicitud=idsolicitud

**5.45 fkidtecnico\_mantenimiento**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
tecnico	mantenimiento	NO ACTION	NO ACTION	idtecnico=idtecnico

**5.46 fkidcarrera\_materia**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
carrera	materia	NO ACTION	NO ACTION	idcarrera=idcarrera

**5.47 fkidclasificacion\_pieza**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
clasificacion	pieza	NO ACTION	NO ACTION	idclasificacion=idclasificacion

**5.48 fkidexistencia\_pieza**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
existencia	pieza	NO ACTION	NO ACTION	idexistencia=idexistencia

**5.49 fkidmarca\_pieza**

Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
marca	pieza	NO ACTION	NO ACTION	idmarca=idmarca

ca				
<b>5.50 fkiddocente_reserva</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
docente	reserva	NO ACTION	NO ACTION	iddocente=iddocente
<b>5.51 fkidequipoexistente_reserva</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
existencia	reserva	NO ACTION	NO ACTION	idexistencia=idequipoexistente
<b>5.52 fkidestado_reserva</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
estadorese rva	reserva	NO ACTION	NO ACTION	idestadoreserva=ide stado
<b>5.53 fkidubicacion_reserva</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
ubicacion	reserva	NO ACTION	NO ACTION	idubicacion=idubic acion
<b>5.54 fkidusuario_reserva</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	reserva	NO ACTION	NO ACTION	idusuario=idusu ario
<b>5.55 fkidclasificacion_software</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
clasificació n	software	NO ACTION	NO ACTION	idclasificacion=idclasifi cacion
<b>5.56 fkidequipoexistente_solicitud</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
existencia	solicitud	NO ACTION	NO ACTION	idexistencia=idequipoexistente
<b>5.57 fkidequiposimple_solicitud</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
equiposim ple	solicitud	NO ACTION	NO ACTION	idEquipoSimple=idequip osimple
<b>5.58 fkidusuario_solicitud</b>				
Tabla Fuente	Tabla Objetivo	Acción de Borrado	Acción de Actualizado	Link
usuario	solicitud	NO ACTION	NO ACTION	idusuario=idusu

ario

**5.59 fk\_tag\_entrada\_entrada**

<b>Tabla Fuente</b>	<b>Tabla Objetivo</b>	<b>Acción de Borrado</b>	<b>Acción de Actualizado</b>	<b>Link</b>
entrada	tag_entrada	CASCADE	CASCADE	identrada=identrada

**5.60 fk\_tag\_entrada\_tag**

<b>Tabla Fuente</b>	<b>Tabla Objetivo</b>	<b>Acción de Borrado</b>	<b>Acción de Actualizado</b>	<b>Link</b>
tag	tag_entrada	NO ACTION	NO ACTION	idtag=idtag

**5.61 fkidautorizacion\_usuario**

<b>Tabla Fuente</b>	<b>Tabla Objetivo</b>	<b>Acción de Borrado</b>	<b>Acción de Actualizado</b>	<b>Link</b>
autorizacion	usuario	NO ACTION	NO ACTION	idautorizacion=idautorizacion

**5.62 fkidrol\_usuario**

<b>Tabla Fuente</b>	<b>Tabla Objetivo</b>	<b>Acción de Borrado</b>	<b>Acción de Actualizado</b>	<b>Link</b>
rol	usuario	NO ACTION	NO ACTION	idrol=idrol

## Conclusiones

Durante el desarrollo de nuestro trabajo de grado tuvimos la oportunidad de poner en práctica muchas de las enseñanzas adquiridas en las cátedras cursadas a lo largo de la carrera pudiendo demostrar su utilidad en muchos casos del diario hacer informático.

Además de lo aprendido en clases, al llevar a cabo un proyecto de desarrollo de tal magnitud nos permitió poner a prueba los conocimientos antes mencionados en un sistema que se asemeja aun más al producto final que un profesional genera. Todo lo mencionado aunado a las experiencias adquiridas participante de este trabajo de grado, que una vez egresados, se dispusieron a continuar con el ciclo de aprendizaje de forma autodidacta en el mundo laboral. Cada una de las técnicas y tecnologías de desarrollo de software empleadas han sido de gran ayuda y un factor clave en el éxito de este trabajo.

### Prácticas:

Gracias a la experiencia de adquirida en varios proyectos de gran magnitud asignados en las diversas materias de la Universidad, nos vimos en la necesidad de investigar sobre herramientas adicionales para mejorar el sistema de control de versiones de código fuente que se generan en el ajetreo de los trabajos en equipo. Debido a la inminente ventaja de utilizar software específicamente diseñado para estos propósitos, es que empleamos en nuestros proyectos herramientas como CVS, SVN, Mercurial ó GIT, que además de no tener costo (como muchas herramientas de Software Libre) y ser multiplataforma, esta demostrada su fiabilidad y eficiencia para controlar y unificar el desarrollo de software en equipo.

El uso de esta herramienta se volvió una práctica necesaria, así como lo es el uso de un IDE, o un framework de trabajo.

SVN, que fue la herramienta elegida para el proyecto, permitió que cada uno de los participantes de este trabajo de grado, pudiera diseñar, programar y depurar el proyecto en copias locales, y actualizar cada uno de los cambios individuales, unificando el trabajo, manteniendo así un repositorio central en la nube, y las últimas actualizaciones del trabajo en grupo.

Debido a que el repositorio central de código del proyecto, está alojado en “la nube” (Google Code), se tuvo la libertad y facilidad de trabajar en máquinas individuales lo que se adaptó a las necesidades de movilidad requeridas por los estudiantes.

Otra práctica que fue implementada y con resultados muy satisfactorios en este proyecto, fue el uso de Pruebas Unitarias (Unit Testing) para mantener un estricto control de la funcionalidad de las reglas de negocio del sistema, antes (y después) de elaborar la interfaz del mismo.

### **Frameworks y Estandares de la Industria**

El uso del JDK 1.6 como plataforma de desarrollo para el proyecto fué una de las decisiones más acertadas, junto con Backing Beans, estándares de java, pojos, servlets,

- Java como plataforma robusta de desarrollo.

- Web, IceFaces, framework, modelo mvc, persistencia.

### **Software de Terceros Multiplataforma**

- Mysql y Apache, software libre, aspectos multiplataforma (Vista, Windows XP y en GNU/Linux)

## Recomendaciones

- Tener en cuenta la población real de estudiantes de Ingeniería y Arquitectura y la población que asiste a recibir prácticas al centro de cómputo de Ingeniería y Arquitectura para obtener una muestra representativa
- Al trabajar con ICEfaces y con reportería exportable a formato PDF, se debe tener en cuenta que no son compatibles entre sí, para lo cual hay que utilizar la librería iText, que es una librería gratuita para Java que permite generar cualquier impresión en PDF, incluyendo el JasperPrint de Jasper Reports.
- Documentar cada una de los Backing Beans para que la programación sea más sencilla
- El redireccionamiento entre páginas JSP es más sencilla y con mejor funcionalidad con un bean de redirecciones, en lugar de realizarlo con la ayuda del archivo de configuración faces-config-xml, aunque se sacrifique un poco de rendimiento.
- Contar con un ambiente (servidor y herramientas) para alojar proyectos internos de la facultad, con el objetivo de facilitar desarrollo, integración y pruebas de calidad de los mismos.
- Contar con un ambiente de producción para alojar aplicaciones java finalizadas, seguras y disponibles para el uso interno de la facultad.
- Hacer uso del código fuente de este trabajo de grado como ejemplo práctico y funcional en las asignaturas relacionadas con informática para hacer

demostraciones de las tecnologías aquí utilizadas y además, que estos tengan la libertad de proponer mejoras continuas para el mismo.

- Fomentar el uso de lenguajes de programación y frameworks de desarrollo utilizados ampliamente en ambientes de empresas informáticas en el país.
- Dotar a todos los Backings Beans de métodos que devuelvan listas, ya que la lectura y el acceso a ellas es mucho más veloz que con arreglos, arreglos bidimensionales o enumeraciones, entre otros.

## Anexos

### Anexo 1: Cotizaciones de servidores para alojar JHard

#### DELL PowerEdge 840



Características:	Precio: \$899.00
PowerEdge 840	
Dual Core Intel®	
Pentium®E2220, 2.4GHz,	
1MB Cache, 800MHz FSB	
1GB, 1x1GB, 800Mhz RAM,	
Single Rank, DDR2	
Disco duro de 160 GB, SATA,	
de 3.5 pulgadas, con	
velocidad de 7,200 RPM	

#### HP ProLiant ML110 G5 Server series



Características:	Precio: \$699.00
Procesador Intel® Dual Xeon®	
Memoria 1 x 1GB RAM	
Disco Duro SATA 160 GB	

#### HP ProLiant DL120 G5 Server series



Características:	Precio: \$899.00
Processor Type Intel® Dual Xeon®	
Memoria 2GB RAM DDR2	

---

Discos duros SATA/SAS NHP

---

**Anexo 2: Cotizaciones de computadoras para desarrollo y para Project Managers de  
JHard**

---

**CPU WINDSTORM CORE 2.33GHz**



Características: Precio: \$388.00

PROCESADOR QUAD CORE  
2.33GHz/1066

Disco 160GB SATA, BIOSTAR  
DG31, MEMORIA RAM 1GB  
DDR2, Teclado, Mouse y  
Bocinas, Case Neova.

**CPU FAST RUNNING PENTIUM DUAL 2.5GHz**



Características: Precio: \$253.00

PROCESADOR PENTIUM  
DUAL 2.5GHz/800

Disco 160GB SATA  
FOXCONN G31MV-K,  
MEMORIA RAM 1GB DDR2,  
Teclado, Mouse y Bocinas,  
Case NEOVA.

---

**CPU CREATIVE DEVELOPER QUAD 2.6**



Características:  
PROCESADOR QUAD CORE  
2.6 GHz/1333  
  
Disco 160GB SATA Biostar  
GF7050, MEMORIA RAM  
1GB DDR2, Teclado, Mouse  
y Bocinas, Case NEOVA.

Precio: \$480.00

**Dell Vostro 220 Mini Tower**



Características:  
Intel® Celeron®, Intel®  
Pentium® Dual Core e Intel®  
Core™ 2 Duo  
Sistema Operativo Windows  
Vista Original  
Sin monitor  
Grabadora de CD/DVD de  
16x (DVD+/-RW) con  
capacidad de grabación de  
doble capa

Precio: \$479.00

## Bibliografía

- Umberto Eco, “Cómo se Hace un Tesis – Técnicas y Procedimientos de estudio, investigación y escritura”, Roma, Italia.
- Raúl Rojas Soriano, “Investigación Social”, San Salvador, El Salvador.
- Calderón Peraza, Ernesto Alexander. “Desarrollo de un sistema informático para la actualización del Centro de Desarrollo Profesional Docente del Ministerio de Educación de El Salvador, Departamento de Ingeniería y Arquitectura, Universidad de El Salvador”. Santa Ana, El Salvador, 2005.

## Índice de gráficos, figuras y tablas

### Índice de gráficos

Nombre	Descripción	Página
<b>Gráfico 1.1</b>	Los encuestados son preguntados sobre cuál es el mayor problema de la inscripción de grupos de laboratorios en el Laboratorio de Cómputo	9
<b>Gráfico 2.1</b>	Cuál es el mayor problema que acarrea inscribir grupos de práctica en el laboratorio de cómputo de Ingeniería y Arquitectura	29
<b>Gráfico 2.2</b>	Cuántas horas clase podrían aprovechar los estudiantes si se solventaran los problemas anteriormente descritos	30
<b>Gráfico 2.3</b>	Se cuestiona si una Wiki de conocimientos informáticos será de utilidad para aumentar el aprendizaje	31
<b>Gráfico 2.4</b>	Se les pregunta a los estudiantes si estarían interesados en participar en cursos de mantenimiento y reparación de hardware, que es una de las principales actividades que pretende impulsar la Unidad.	31
<b>Gráfico 2.5</b>	Cuánto tiempo dedican los docentes en sus clases por ciclo a formar grupos de práctica	32
<b>Gráfico 2.6</b>	Pregunta si el docente utiliza equipo multimedia para sus clases expositivas	33
<b>Gráfico 2.7</b>	Cuál es el mayor problema al momento de reservar equipo multimedia en el Departamento de Ingeniería y Arquitectura	34
<b>Gráfico 2.8</b>	Se le cuestiona a los docentes si con un sistema web se podría mejorar el sistema de reserva de equipo multimedia	34
<b>Gráfico 2.9</b>	Cuál es el mayor problema informático que tiene el personal administrativo de la UES-FMO	35
<b>Gráfico 2.10</b>	Podrá resolver los problemas antes mencionados con la ayuda web en línea	35
<b>Gráfico 2.11</b>	Está dispuesto a someterse a capacitaciones en informática propias para personal administrativo	36
<b>Gráfico 2.12</b>	Los encuestados responden si con un sistema informático se resolvieran más fácilmente problemas informáticos en la UES-FMO	37

## Índice de Tablas

Nombre	Descripción	Página(s)
<b>Tabla 2.1</b>	Fórmula y tamaño muestral que se utilizó para realizar el estudio de campo	25
<b>Tabla 2.2</b>	Preguntas de las encuestas a estudiantes	26
<b>Tabla 2.3</b>	Preguntas de encuestas a docentes	27
<b>Tabla 2.4</b>	Preguntas a personal administrativo	27-28
<b>Tabla 3.1</b>	Requerimientos de Desarrollo de JHard	38
<b>Tabla 3.2</b>	Tabla resumen de Requerimientos de RR.HH. para el desarrollo de JHard	39
<b>Tabla 3.3</b>	Tabla resumen de Requerimientos de producción de JHard	40
<b>Tabla 3.4</b>	Tabla resumen de Requerimientos de RR.HH. para JHard en producción	40-41
<b>Tabla 3.5</b>	Costo de JHard	47
<b>Tabla 4.1</b>	Tabla de Clases de JHard	69
<b>Tabla 4.2</b>	Tabla de beans de negocios de JHard	73
<b>Tabla 4.3</b>	Tabla resumen de tecnologías de JHard	89
<b>Tabla 4.4</b>	Controles ICEfaces que implementa JHard	93-94
<b>Tabla 5.1</b>	Actividades para implementación de JHard	96
<b>Tabla 5.2</b>	Costo de implementación de JHard	97-98

## Índice de Figuras

<b>Nombre</b>	<b>Descripción</b>	<b>Página(s)</b>
<b>Figura 4.1</b>	Diagrama Entidad-Relación de JHard	49
<b>Figura 4.2</b>	Diagrama ER de JRequest	50
<b>Figura 4.3</b>	Diagrama ER de JWiki/JProCur	51
<b>Figura 4.4</b>	Diagrama ER de JInvent	53
<b>Figura 4.5</b>	Diagrama ER de ManLab	53
<b>Figura 4.6</b>	Diagrama ER de JCanon	54
<b>Figura 4.7</b>	Diagrama ER de JHardmin	55
<b>Figura 4.8</b>	Diagrama de Clases de JHard	60-68
<b>Figura 4.9</b>	Arquitectura MVC	71
<b>Figura 4.9</b>	Logo de ICEfaces	85
<b>Figura 4.10</b>	Arquitectura general de ICEfaces	87
<b>Figura 4.11</b>	Arquitectura de Ajax Push de ICEfaces	91
<b>Figura 6.1</b>	Configuración de Java JRE 6 en Ubuntu	115
<b>Figura 6.2</b>	Configuración de Java JRE 6 en Ubuntu	115
<b>Figura 6.3</b>	Configuración MySQL 5.1 en Ubuntu	119