Universidad de Cienfuegos "Carlos Rafael Rodríguez"
Facultad de Informática
Carrera de Ingeniería Informática

Trabajo de diploma para optar por el título de

Ingeniería en informática

Título:

"Sistema automatizado para la gestión de tiempos de máquina"

Autor:
Raphie Ann Morgan

Tutores:
Dr. Sc.Ernesto R.Fuentes Garí
Ing.Danaysi Ruiz Bravo

Cienfuegos, Cuba

Curso 2005 – 2006

"Ano de la Revolución Energética en Cuba"

…… As an individual,
you can stand up and feel proud
of the independence and maturity
that have come from the dedication
and determination in your life….

Ben Daniels

A mi Mamá

A la memoria de mi bisabuela

A mi amigo Leon Senior

Agradecimientos

Quisiera agradecer a todas las personas que de una forma u otra han contribuido con su ayuda y sin la cual no hubiese sido posible la realización de este trabajo, en especial a:

- *A Dios porque siempre ha estado a mi lado, le pude ver a través de los personas que me han ayudado.*
- *A Dr. Ernesto Roberto Fuente Garí, tutor de esta investigación.*
- *A mi profesora y amiga Danaysi por el tiempo dedicado, por estar a mi lado en tiempos difíciles y por permitirme compartir mis preocupaciones.*
- *A los profesores de la Universidad de Cienfuegos que me trasmitieron conocimientos y valores que me permitiera formarme como una profesional.*
- *A los compañeros de grupo, por estos cinco años compartidos.*
- *A mi amiga Lissette por su paciencia, el tiempo que me dedicó.*

A todos, Muchísimas Gracias.

To my mother

To my friend, my love, my life……

*Thanks for showing me the true meaning of the word **LOVE***

You are always there for me in good and bad times. Thanks for your wisdom, knowledge and understanding that you have portraited.. You always make me feel so confident and

You always show me the light through the darkness.

Thanks for all your support.

Everything I do Mom I do it for you.

Love you

To my great grandmother

To my angel, Jane Roberts…….

Thanks for grooming the most important person in my life to

the person that she is today, my Mom. I know that you are here

sharing this special moment with us.

May you soul rest in peace

To a very special person, Leon Senior…….

Words on paper can't explain my appreciation for all that you have done for me .If it wasn't for you I would not be here today in Cuba , making this important step in my life. I hope that I've made you proud .I will always love you for what you have done for me .

Love you

Declaración de autoría

Yo, Raphie Ann Morgan declaro que soy la única autora de este trabajo realizado en la Universidad de Cienfuegos “Carlos Rafael Rodríguez” como parte de la culminación de los estudios de la especialidad de Ingeniería Informática. Autorizo a que el mismo sea utilizado por la institución para los fines que estime conveniente, ya sea parcial o totalmente y que no será presentado en ningún evento ni publicado sin la autorización de la Facultad de Informática.

_____

Firma del Autor

Raphie Ann Morgan

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura.

_____

Firma del Tutor

Dr. Sc. Ernesto R. Fuentes Garí

_____

Firma del Tutor

Ing. Danaysi Ruiz Bravo

_____

Firma ICT

_____

Vicedecana

Resumen

El desarrollo de las tecnologías de la información y las comunicaciones (TIC) ha provocado un gran impacto en la sociedad y de manera relevante en la enseñanza superior. Se inserta entonces la enseñanza de la Informática en todos los planes de estudio de las carreras universitarias, necesitándose graduar profesionales capaces de utilizar de manera efectiva los recursos informacionales que se encuentren a su alcance. La planificación de los tiempos de máquina resulta un proceso de vital importancia, atendiendo a que la correcta gestión de los mismos garantiza un buen uso de los equipos y la satisfacción de los usuarios de los laboratorios.

Teniendo en cuenta lo anterior, en la Universidad de Cienfuegos “Carlos Rafael Rodríguez” (UCf) surge la idea de un proyecto con el objetivo de automatizar la gestión de los tiempos de máquina, posibilitando este sistema un mejor control y gestión de los procesos que tienen lugar en los laboratorios docentes de las facultades de este centro de enseñanza.

Se llevaron a cabo las etapas del proceso de desarrollo del software propuesto, según lo que especifica la metodología de Proceso Unificado de Desarrollo, obteniéndose como producto final un sistema automatizado para la gestión de los tiempos de máquina.

Índice

Introducción 1

Capítulo 1. Fundamentación del tema. 5

1.1 Introducción. 5

1.2 Uso de las TIC en la educación superior. 5

1.3 La enseñanza de la informática en todas las carreras universitarias. 5

1.4 Descripción del objeto de estudio. 6

1.4.1 Servicios que brindan los laboratorios docentes 7

1.4.2 Análisis de la ejecución de los procesos que origina la situación problémica. 7

1.5 Descripción de los sistemas existentes 8

1.6 Descripción del objeto de automatización. 9

1.7 Tendencias, metodologías y tecnologías actuales 10

1.7.1 Metodologías actuales a considerar 10

1.7.2 Tecnologías actuales a considerar. 11

1.8 Conclusiones 19

Capítulo 2. Modelo del negocio. 20

2.1 Introducción 20

2.2 Descripción del modelo de negocio 20

2.3 Reglas del negocio a considerar 21

2.4 Modelación del modelo de negocio 22

2.4.1 Actores y trabajadores del negocio. 22

2.4.2 Diagrama de Caso de Uso del Negocio 23

2.4.3 Descripción del caso de uso del negocio. 24

2.4.4 Diagramas de actividades del negocio 25
2.4.5 Modelo de objetos del negocio 27
2.5 Conclusiones 27
Capítulo 3. Modelo de sistema 28
3.1 Introducción 28
3.2 Descripción del modelo del sistema 28
3.3 Modelación del sistema 28
3.3.1 Requerimientos funcionales 28
3.3.2 Requerimientos no funcionales 30
3.3.3 Actores del sistema 32
3.3.4 Casos de uso del sistema. 32
3.3.5 Paquetes y sus relaciones 33
3.3.6 Diagrama de caso de uso del sistema 34
3.3.7 Descripción de los casos de usos del sistema 37
3.4 Conclusiones. 48
Capítulo 4. Construcción del sistema. 49
4.1 Introducción 49
4.1.1 Diagrama de clases 49
4.1.2 Diagramas del modelo lógico de datos. 63
4.1.3 Diagramas del modelo físico de datos 64
4.1.4 Diagramas de implementación 65
4.2 – Principios de diseño del sistema. 66
4.2.1 Estándares en la interfaz de la aplicación. 66
4.2.2 Tratamiento de excepciones. 66

4.3 – Conclusiones 67
Conclusiones 68
Recomendaciones 69
Referencias Bibliográficas 70
Bibliografía 72
Anexo 1. Prototipos. 76
Anexo 1.1 Caso de Uso: Gestionar PC 76
Anexo 1.2 Caso de Uso: Gestionar Usuario 77
Anexo 1.3 Caso de Uso: Gestionar software 78
Anexo 1.4 Caso de Uso: Gestionar laboratorio 78
Anexo 1.5 Caso de Uso: Gestionar actividad docente 79
Anexo 1.6 Caso de Uso: Deshabilitar PC con dificultades 80
Anexo 1.7 Caso de Uso: Gestionar turno 80
Anexo 1.8 Caso de Uso: Visualizar reporte de estado de las PC 81
Anexo 1.9 Caso de Uso: Visualizar PC reportadas con problemas 81
Anexo 1. 10 Caso de Uso: Mostrar tiempos de máquina disponibles 82
Anexo 1.11 Caso de Uso: Mostrar las actividades docentes planificadas 83
Anexo 1.12 Caso de Uso: Visualizar detalles de una PC 84
Anexo 1.13 Caso de Uso: Registrar tiempo de máquina 85
Anexo 1.14 Caso de Uso: Autenticarse 86
Anexo 1.15 Caso de Uso: Reportar PC con problemas 86
Anexo 2. Bibliografía tomada de EBSCO. 87

Introducción

Hablar de las TIC es hablar de un amplio abanico de técnicas, herramientas, ámbitos de investigación y desarrollo. Grande ha sido el impacto de estas tecnologías en todos los sectores y de forma especial en la enseñanza superior, provocando la ocurrencia de cambios transcendentales en las formas de acceder al conocimiento y de ejecutar la comunicación.

Este fenómeno ha repercutido directamente en todos los procesos universitarios que son resultantes o generadores de la actividad de gestión de la información y del conocimiento. Ha incidido de manera muy significativa en la forma de enseñar y aprender las profesiones.

Se plantea que estos cambios rebasan la frontera del conocimiento que hasta la actualidad habían aportado las técnicas de computación. Por lo anteriormente expuesto en la Universidad Cubana surgió la necesidad de insertar la enseñanza de la informática en todas sus carreras, permitiendo desarrollar en los estudiantes habilidades que les posibilitara hacer un uso efectivo de los recursos y servicios que se prestan en los centros de información y documentación, recuperar datos soportados en formato electrónico, organizar el proceso de recuperación y gestión del recurso información, partiendo de la solución de un problema investigativo del ejercicio de su profesión, hacer uso de los sistemas de mensajería electrónica como herramienta de acceso a servicios de información científica y tecnológica, elaborar documentos electrónicos, etc.

La enseñanza de esta disciplina tendría como objeto de estudio los recursos informacionales y las TIC como medio de acceso a los servicios de información y a la producción de conocimiento. Por otra parte se necesitaba contar con la tecnología que permitiera dar cumplimiento a la enseñanza de la computación a todos los estudiantes universitarios, creándose los laboratorios docentes, pudiéndose afirmar que en la actualidad todas las facultades universitarias del país cuentan con al menos uno de éstos. Estos laboratorios son utilizados en actividades docentes grupales, así como en el estudio individual.

La UCf, como un centro de enseñanza superior más, no está exenta de esta situación. En el caso de esta institución, las necesidades siempre crecientes, así como el aumento progresivo de la matrícula han provocado que aún el número de estudiantes por cada computadora en los laboratorios docentes de sus facultades, sea mayor que el deseado, siendo necesario brindar vital importancia a la adecuada y equitativa planificación del uso de las computadoras por parte de los mismos. Esta distribución de los tiempos de máquina se realiza en la actualidad bajo la responsabilidad de los técnicos y a solicitud de los estudiantes. Cada estudiante debe acercarse al técnico y solicitar su tiempo de máquina, el técnico deberá chequear si la solicitud puede ser aprobada o no, es decir si la máquina está libre o no en el tiempo que está solicitando el estudiante. Esto provoca la obligada presencia del técnico y del estudiante que realiza la solicitud, además de introducir demoras en este proceso y la aparición de insatisfacciones en los usuarios de los laboratorios docentes.

Teniendo en cuenta la situación problémica anterior se define como el problema a resolver con este trabajo el siguiente:

La carencia de un sistema en la UCf que facilite la gestión de los tiempos de máquina en los laboratorios docentes de sus facultades, así como el tratamiento y manipulación de información importante en dichos laboratorios.

Teniendo en cuenta lo anterior y como solución al problema planteado, se define como **Objetivo General** de este trabajo**:**

Analizar, diseñar e implementar un sistema para la gestión de tiempos de máquina en los laboratorios docentes de la UCf.

De este objetivo general se desprenden los siguientes **Objetivos Específicos:**

- Estudiar los procesos de asignación de tiempos de máquina en cada una de las facultades de la UCf.
- Diseñar una Base de Datos que almacene los datos de los usuarios y PCs de la red.

- Diseñar e implementar una aplicación Web que facilite el pedido de tiempos de máquina, la búsqueda de información y la programación de actividades docentes.

Para dar cumplimiento a estos objetivos se definieron las siguientes actividades:

1. Entrevistar a los técnicos, administradores y usuarios de la red para conocer las actividades que se desarrollan en los laboratorios de la Facultades.
2. Valorar las facilidades que debe brindar el sistema para la gestión en los laboratorios de acuerdo a las necesidades actuales.
3. Seleccionar las tendencias y tecnologías que se van a utilizar, para dar solución al problema.
4. Analizar y diseñar la base de datos que contendrá la información necesaria para automatizar la gestión de los tiempos de máquina.
5. Implementar un módulo para los usuarios de la red que les facilite la realización de búsquedas y de pedidos de tiempos de máquina.
6. Implementar un módulo para los técnicos del laboratorio que facilite la actualización de los datos de los laboratorios y el manejo de la información referente a los usuarios y las computadoras.
7. Documentar el sistema con la información referente al análisis, diseño e implementación que se generó durante la etapa de desarrollo.

Con la realización de este trabajo se defiende la siguiente idea:

La introducción de un sistema automatizado permitirá mejorar el control y la gestión en los procesos que tienen lugar en los laboratorios de las facultades de la UCf.

El **Aporte Práctico** que realiza este trabajo, es el desarrollo de un método alternativo para el control de los tiempos de máquina, automatizando este proceso con alta fiabilidad, rapidez y comodidad en el manejo y obtención de informaciones.

Para guiar el desarrollo de este trabajo se siguieron las pautas propuestas por el Proceso de Unificado de Desarrollo (RUP) y se utilizó el Lenguaje Unificado de

Modelación (UML) para representar los artefactos que propone RUP en cada etapa del ciclo de vida del sistema propuesto.

El presente trabajo, estructurado en 4 capítulos, resume la siguiente información:

Capitulo 1. Fundamentación del tema. Análisis del objeto de estudio, tendencias y tecnologías actuales seleccionadas a emplear en el desarrollo de la propuesta y el porqué de su utilización.

Capitulo 2. Modelo del Negocio. Descripción de los procesos, actores, trabajadores y casos de uso del negocio; y diagramas de clases del modelo de objetos del negocio.

Capítulo 3. Descripción de la solución propuesta. Se definen los requisitos funcionales y no funcionales; actores y casos de uso del sistema.

Capítulo 4. Construcción de la solución propuesta. Descripción del diseño a través del diagrama de clases, el de clases persistentes y el modelo de datos. Se definen, además, los principios de diseño seguidos en la aplicación y el modelo de implementación mediante los diagramas de despliegue y componentes.

Capítulo 1. Fundamentación del tema.

1.1 Introducción.

En este capítulo se puntualizan los conceptos necesarios para comprender el dominio del problema y el objeto de estudio. Además se analiza el flujo de los principales procesos que se desarrollan en el campo de acción y las causas que originan la situación problémica. Se describe además el objeto de automatización y las tendencias, metodologías y tecnologías actuales, sobre las cuales se fundamenta la propuesta de automatización.

1.2 Uso de las TIC en la educación superior.

Como se había expresado anteriormente, entre las principales tendencias en la Educación Superior se destaca el uso de las TIC. La introducción de éstas tecnologías en el funcionamiento de una Institución de Educación Superior, están asociadas a lograr las siguientes habilidades en el profesional: [1]

- Utilizar de manera efectiva los recursos informacionales disponibles a su alcance.
- Lograr el aprendizaje de los sistemas computacionales que le faciliten el manejo de la información.
- Lograr que los estudiantes sean capaces de programar en algún lenguaje de programación visual orientado a objetos que le permita resolver problemas concretos en su área de conocimiento.
- Emplear los algoritmos y las operaciones lógicas de la computación como vía de interacción con los sistemas automatizados de gestión del conocimiento.

1.3 La enseñanza de la informática en todas las carreras universitarias.

Por la introducción masiva de las nuevas tecnologías en la sociedad cubana se hace imprescindible la utilización ordenada y masiva de las tecnologías de la información y las comunicaciones en todas las esferas de la sociedad cubana. Como paso previo e

imprescindible para el desarrollo científico técnico fue necesario introducir profundos cambios en los sistemas y estructuras educacionales.

El Ministerio de Educación Superior ha implementado una estrategia la cual incluye cambios certeros en la tecnología informática que han permitido que las universidades cuenten con laboratorios de alta tecnología. Estos laboratorios presentan características generales independientemente de la facultad o carrera a la que pertenezcan. En el caso específico de la UCf, los laboratorios docentes de todas las facultades cuentan con un plan de seguridad informática, así como con un reglamento que refleja los deberes y derechos de los usuarios de dichos laboratorios. Cada laboratorio cuenta además con un técnico y un jefe de laboratorios a los que va dirigido cualquier inquietud o problema presentado durante el uso de los mismos.

Como parte del proceso lógico de desarrollo todas las carreras han incorporado la enseñanza de la informática a sus planes de estudio, esto, sumado al creciente número de alumnos que ingresan a nuestras universidades cada año provocan que la cantidad de computadoras que se tienen en los laboratorios docentes de las facultades de la UCf resulte aún insuficiente y que el número mínimo de estudiantes por cada equipo sea superior al deseado. Resulta entonces de vital importancia administrar adecuadamente el tiempo de cada estudiante en las computadoras de los laboratorios docentes para lograr el mayor aprovechamiento de los equipos y lograr un nivel de satisfacción alto para los usuarios de estos laboratorios. Surge entonces la idea de realizar un proyecto que automatice la planificación de los tiempos de máquina en la UCf y aunque la idea es que este proyecto pueda aplicarse a todas las facultades de la institución se tomó como referencia la facultad de Informática.

1.4 Descripción del objeto de estudio.

Ha sido objeto de estudio la gestión en los laboratorios de computación de la UCf, en los cuales se desarrolla parte importante del proceso docente.

1.4.1 Servicios que brindan los laboratorios docentes

En los laboratorios docentes de la UCf hay un promedio de 10 a 15 computadoras en cada uno, conectadas a la red de computadoras de la institución (RED UCf). Dicha red tiene por objeto garantizar el intercambio de información académica.

Los usuarios de los laboratorios son los estudiantes y técnicos. En los laboratorios los usuarios desarrollan actividades docentes y de estudio independiente. Las actividades docentes pueden ser: de pregrado (curso regular diurno y curso para trabajadores) y de postgrados (diplomados, maestrías y doctorados); sólo se realizan entre las 8:00 a.m. y las 6:00 p.m. y las independientes, siempre que no estén ocupados los laboratorios con actividades docentes.

1.4.2 Análisis de la ejecución de los procesos que origina la situación problémica.

En la actualidad no se cuenta con un sistema automatizado que facilite el control de los laboratorios, permita realizar pedidos de tiempos de máquina y almacene datos generales de usuarios y computadoras.

Si los usuarios de la red desean solicitar tiempo de máquina en la semana en curso deben acercarse al técnico del laboratorio, si lo desean para la semana entrante deben esperar al viernes para realizar el pedido. El técnico consulta una tabla de Excel, en la cual están registrados los tiempos de máquina de la semana en cuestión. Si no se puede satisfacer el pedido de tiempo de máquina, es decir no hay computadoras disponibles para la fecha y hora solicitadas, el técnico negocia con el usuario para otorgar otro tiempo de máquina. Este proceso introduce demoras y por tanto es molesto para los usuarios.

Los administradores tampoco cuentan con una aplicación que les facilite conocer los tiempos de máquina asignados e informaciones relacionadas con las computadoras y los usuarios. No existe un registro con los datos de las computadoras: sistema operativo, software, hardware; ni de las computadoras rotas y en reparación. La gestión de la información de los usuarios, de las computadoras y de los tiempos de máquina es muy engorrosa.

1.5 Descripción de los sistemas existentes

Después de realizar una búsqueda sobre los sistemas relacionados con el campo de acción de la gestión de los tiempos de máquina en laboratorios solo se encontró un proyecto que se está desarrollando en la Universidad Politécnica de Madrid para gestionar redes, servidores y algunos servicios que brindan los laboratorios de estudiantes. Este sistema tiene algunas similitudes al que se propone en el presente trabajo.

Breve descripción del Sistema de Administración Centralizada

El Departamento de Ingeniería de Sistemas Telemáticos (DIT), perteneciente a la Universidad Politécnica de Madrid, desarrolla un sistema de gestión de redes, servidores y servicios para sus laboratorios multipropósito, los cuales están distribuidos en diferentes dependencias de la Escuela Superior de Ingenieros Técnicos de Telecomunicaciones.

Este sistema de gestión de laboratorios ha de permitir gestionar grandes redes de ordenadores con múltiples propósitos y sistemas operativos de forma racional y sostenible, haciendo que el costo de mantenimiento disminuya con el número de computadoras gestionadas en lugar de crecer, tal y como viene siendo habitual con los sistemas de administración actuales. Como ventaja añadida, está la de compartir los recursos (equipos de comunicaciones, estaciones, espacio físico, tiempo, etc.) entre los diversos grupos de prácticas, de tal forma que todos utilicen los mismos ordenadores pero con dos o más configuraciones y/o sistemas operativos diferentes. En la actualidad, los laboratorios docentes del DIT dan servicio a más de 2000 alumnos anuales matriculados en más de 10 asignaturas diferentes con soporte práctico. La realización de ese proyecto tendrá como resultado, entre otras ventajas, las siguientes: [2]

- Mejorar sensiblemente la instalación y actualización de estaciones idénticas o muy similares.
- Facilitar a administradores sin experiencia, o incluso a los propios usuarios, la oportunidad de instalar y actualizar sus máquinas.

- Permitir identificar cualquier característica de una máquina y gestionar la configuración de diferentes sistemas operativos en la misma.
- Mejorar la productividad, debido a que se podrán reinstalar y actualizar múltiples estaciones de forma automatizada, sin requerir la atención del personal asignado, más que en tareas de supervisión avanzada.
- Aumentar la flexibilidad del entorno de trabajo, permitiendo reconfigurar laboratorios enteros (cientos de máquinas) en breves períodos de tiempo.

El sistema antes descrito se encuentra aún en etapa de desarrollo, no cumple con el total de las funcionalidades que se pretende lograr con la automatización de la gestión de los tiempos de máquina para la UCf, no ajustándose por presentar características muy diferentes los laboratorios del DIT, además se debe tener en cuenta que al culminarse dicho proyecto para la adquisición del mismo, es muy probable que se tenga que pagar algún precio por el.

1.6 Descripción del objeto de automatización

Los procesos a automatizar son los siguientes: realizar solicitud de tiempos de máquina, almacenar datos generales de los usuarios y de las computadoras, así como la actualización de la información almacenada por parte de los administradores. Existen dos tipos de usuarios: un administrador y un usuario común.

Los estudiantes pueden reservar un máximo de 3 horas de tiempo de máquina para la semana entrante, los días jueves y viernes de la semana en curso. En la semana en curso pueden reservar como máximo otras 8 horas si hay disponibilidad. Dos o más estudiantes no pueden reservar para igual máquina y tiempo. Los tiempos de máquinas son otorgados para el horario de la noche, es decir para el horario extradocente.

Además, los estudiantes pueden reportar una computadora con dificultades para que sea chequeada por el administrador y si está averiada, ponerla como inactiva.

Los usuarios pueden ver las clases programadas en los laboratorios, buscar las características de una computadora y así escoger en cual debe sentarse según sus necesidades.

1.7 Tendencias, metodologías y tecnologías actuales

En el presente epígrafe se realiza un estudio de las principales tendencias, metodologías y tecnologías actuales a tener en cuenta para el desarrollo de la propuesta.

1.7.1 Metodologías actuales a considerar

Lenguaje Unificado de Modelación (UML)

UML (por sus siglas en inglés, *Unified Modelling Language*) es el lenguaje de modelado de sistemas de software más conocido en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. El UML ofrece un estándar para escribir un "plano" del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. UML cuenta con varios tipos de modelos, los cuales muestran diferentes aspectos de las entidades representadas.

Rational Unified Process (RUP)

El Proceso Unificado de Desarrollo o RUP (Rational Unified Process), es un proceso de desarrollo de software que utilizando UML, constituye una metodología estándar muy utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Tiene las siguientes características: [3]

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores practicas en ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos

- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

RUP es un producto de Rational. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (productos tangibles del proceso) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al concluir cada ciclo, cada ciclo se divide en las siguientes fases:

- Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos
- Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos
- Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario
- Transición: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requerimientos a ser analizados.

1.7.2 Tecnologías actuales a considerar.

En este epígrafe se realiza un estudio de las tecnologías a tener en cuenta para el desarrollo de la propuesta de software.

1.7.2.1 Soluciones de Código Abierto (Open Source)

En la actualidad ha surgido una alternativa que permite la existencia de nuevos esquemas de desarrollo cultural y técnico basado en un nuevo modelo de código abierto y libre.

Código abierto en inglés ***Open Source***, es conocido también como software distribuido, término usado como reemplazo al ambiguo nombre original del software libre (*free software*), pues la expresión “software libre” puede indicar que se encuentra disponible de forma gratuita, mientras que “código abierto” significa que en vez de que el código del sistema o de cada uno de los programas sea un secreto celosamente guardado por la empresa que lo produce, éste es puesto a disposición del público, para que pueda ser modificado, mejorado o corregido.

Un programa de código abierto puede ser un programa semi-libre o uno completamente propietario, pero sobre todo puede ser un software libre pues la mayoría se distribuyen bajo la licencia pública general GNU (GPL en inglés, General Public License), que explícitamente establece en su preámbulo que "tiene por objeto garantizar la libertad de compartir y modificar el software libre". Por su parte, el software propietario, que compone la base de la mayoría del software que con más frecuencia se utiliza, se distribuye bajo condiciones muy diferentes y el código fuente normalmente se encuentra oculto y fuera del alcance del usuario. Una licencia propietaria prohíbe la modificación, copia o redistribución sin el permiso de la compañía. Se crea por un grupo relativamente pequeño de desarrolladores de una empresa concreta, trabajando a menudo bajo la presión de los plazos de entrega, por lo que pueden aparecer fallos que se traducen en problemas para los compradores y aunque estos sepan cómo arreglarlos no se les permite. [4]

El modelo de “código abierto” radica en el deseo de ayudar y ser ayudado. Una persona escribe un programa para resolver una necesidad específica, la comenta con otros y estos al ver que dicho programa también los beneficia, le ayudan a realizarlo. En el proceso, la propiedad intelectual no se vuelve una lucha de "egos", se transforma en un trabajo de todos y para todos; se generan líderes naturales con autoridad moral sobre el proyecto, pero nunca dueños. Un proceso tan diferente es difícil de asimilar. [5]

A continuación se describe brevemente las tres herramientas necesarias para el desarrollo de una aplicación Web siguiendo la línea del software libre.

1.7.2.2 Servidor Apache

A partir del análisis de los datos obtenidos del sitio de Netcraft [6] donde constantemente se realizan encuestas sobre los servidores web que más se utilizan mundialmente, se puede apreciar con claridad en la Figura 1.1 la superioridad del servidor Web Apache sobre otros reconocidos como el Internet Information Server (IIS).

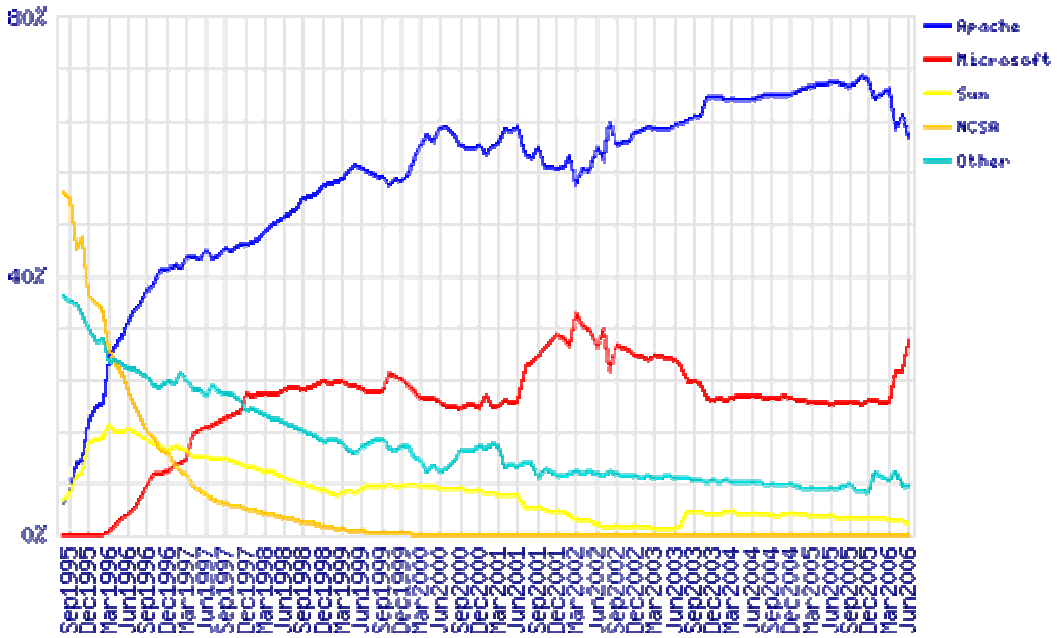


Figura 1.1. Servidores Web más utilizados mundialmente. (Septiembre 1995 – Junio 2006) [6]

Developer	May2006	Percent	June 2006	Percent	Change
Apache	52819517	64.76	52389885	61.25	-3.51
Microsoft	20764239	25.46	25415611	29.71	4.25
Sun	1917950	2.35	1311822	1.53	-0.82
Zeus	550437	0.67	531399	0.62	-0.05

De acuerdo a las estadísticas de Netcraft para el mes de Junio del 2006, Apache sigue siendo el servidor Web más utilizado en la red, acaparando el 61.25% de 52,389, 885 servidores consultados.

Apache además de constituir el servidor más utilizado, forma parte del proyecto de "código abierto", implicando esto que puede obtenerse libremente a través de Internet de forma gratuita la aplicación y su código fuente, lo que ha permitido el desarrollo de funciones del Apache por muchos programadores con carácter voluntario, dando lugar a rápidas mejoras en su funcionalidad y posibilitando la rápida subsanación de errores o fallos de seguridad.

Aunque Apache es un servidor inicialmente pensado para ser ejecutado en servidores Unix, se ha realizado un verdadero esfuerzo de migración, de modo que es un software capaz de funcionar en una gran cantidad de plataformas: aix, aux, beos, bs2000-osd, bsdi, cygwin, darwin, dgux, digitalunix, freebsd, hpux, irix, linux, macosx, macosxserver, netbsd, netware, openbsd, os2, os390, osf1, qnx, reliantunix, rhapsody, sinix, solaris, sunos, unixware y win32. Pero sin dudas, Linux es su plataforma favorita y en la que más servidores de Internet están trabajando. [7].

1.7.2.3 Servidor de Bases de Datos MySQL

MySQL es uno de los Gestores de Bases de Datos más populares desarrollados bajo la filosofía de código abierto, puede utilizarse gratuitamente y su código fuente está disponible. Es desarrollado y mantenido por la empresa MySql AB y considerado una de las aplicaciones de código abierto más exitosas, abriéndose camino gracias a su nivel competitivo y su accesibilidad. Entre los clientes de la compañía se cuentan nada menos que la **NASA, Yahoo, Suzuki, Google y la Agencia Associated Press**. [8]

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan. Poco a poco los elementos faltantes en MySQL están

siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar: [9]

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Dentro de las Ventajas que MySQL ofrece se encuentran: [9]

- Alto rendimiento.
- Elevada velocidad tanto al conectar con el servidor como al servir consultas y demás funciones.
- Excelentes utilidades de administración (backup, recuperación de errores, etc).
- Aunque se bloquee, no suele perder información ni corromper los datos.
- Excelente integración con PHP.
- No tiene límites en el tamaño de los registros.
- Significativo control de acceso, es decir, qué usuarios tienen acceso a que tablas y con qué permisos.
- MySQL se comporta mejor que otros motores de Base de Datos como SQL Server y Oracle a la hora de modificar ó añadir campos a una tabla.

- Es multi-hilo, o sea, maneja muchas conexiones al mismo tiempo, donde cada una tiene su propio hilo, de modo que ningún hilo tiene que esperar por otro, a menos que uno esté modificando una tabla que otro quiera acceder.

1.7.2.4 Lenguaje de Programación PHP

PHP (siglas que originalmente significaban Personal Home Page) fue primero escrito por Rasmus Lerdorf como un simple conjunto de scripts de Perl para guiar a los usuarios en sus páginas. PHP actualmente significa Hypertext Preprocessor.

Es uno de los lenguajes de lado servidor más extendidos en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente creciente que ha tenido una gran aceptación en la comunidad de webmasters debido sobre todo a la potencia y simplicidad que lo caracterizan.

PHP permite embeber sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas íntegramente en un lenguaje distinto al HTML. Se podría efectuar una comparación entre el lenguaje PHP y la tecnología ASP a la hora de generar páginas dinámicas estableciéndose algunas diferencias: [10]

- PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones.
- ASP, siendo una tecnología Microsoft, esta orientado hacia sistemas Windows, especialmente NT.
- Las tareas fundamentales que puede realizar directamente el lenguaje son definidas en PHP como funciones mientras que ASP invoca más frecuentemente los objetos. Por supuesto, esto no es más que una simple cuestión de forma ya que ambos lenguajes soportan igualmente ambos procedimientos. ASP realiza numerosas tareas sirviéndose de componentes (objetos) que deben ser comprados (o programados) por el servidor a determinadas empresas especializadas.

- PHP presenta una filosofía totalmente diferente y, con un espíritu más generoso, es progresivamente construido por colaboradores desinteresados que implementan nuevas funciones en nuevas versiones del lenguaje.

Elementos para tener en cuenta en un lenguaje de scripts

Las cuatro grandes características: Velocidad, estabilidad, seguridad y simplicidad. [10]

- **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.
- **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de fallas, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para estas fallas sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini.
- **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.
- Otra característica a tener en cuenta seria la conectividad. PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

Ventajas adicionales de PHP [11]

- PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP,2003 Server) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.
- La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP.
- Entre los lenguajes del tipo C incluimos al Java y Javascript, de hecho mucha de la funcionalidad del PHP se la debe al C en funciones como fread() o srtlen(), así que muchos programadores se sentirán como en casa.
- PHP es completamente expandible. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código.
- Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como modulo CGI.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran.
- Una gran variedad de módulos cuando un programador PHP necesite una interfase para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.
- Rapidez. PHP generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

- PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.

1.8 Conclusiones

Después de realizar un estudio de los procesos que tienen lugar en los laboratorio docentes y de los sistemas vinculados a estos; se evidencia la necesidad de crear una aplicación que facilite la gestión en los laboratorios de computación. Para ello, se propone un sistema basado en tecnología Web y base de datos.

Para el desarrollo del sistema propuesto se realizó un análisis de las tecnologías actuales a considerar pudiendo concluirse que: el lenguaje que será utilizado para la implementación de la propuesta es PHP y MySql como servidor de Base de Datos, teniendo en cuenta que ambas tecnologías se han convertido es estándares para el desarrollo de aplicaciones web. PHP se ha convertido en un poderoso lenguaje de programación que se ejecuta del lado del servidor y que ha sido creado específicamente para el desarrollo de aplicaciones de este tipo, y MySql como un poderoso motor de bases de datos relacionales que proporciona el repositorio perfecto para aplicaciones que requieran acceso rápido y confiable a la información. Se utilizará Apache como servidor web, resultando es el servidor más utilizado en la red, acaparando el 61.25% de un total de 52,389,885 servidores analizados. Esta selección se realizó atendiendo a las tendencias actuales del desarrollo de aplicaciones sobre software libre.

Capítulo 2. Modelo del negocio.

2.1 Introducción

En el presente capítulo se analizan los procesos que tienen lugar en el objeto de estudio utilizando la metodología de desarrollo de software RUP. Se definen además, los actores, trabajadores y casos de uso del negocio, así como sus actividades y objetos. También se hace una descripción detallada de las reglas del negocio.

2.2 Descripción del modelo de negocio

El modelo de Casos de Uso del Negocio es un modelo que describe los procesos de negocio de una empresa en términos de casos de uso y actores del negocio en correspondencia con los procesos del negocio y los clientes, respectivamente. El modelo de casos de uso del negocio presenta un sistema (en este caso, el negocio) desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios. Este modelo permite a los modeladores comprender mejor qué valor proporciona el negocio a sus actores. [12]

Este es uno de los modelos útiles previos al desarrollo de un software. Es una técnica para comprender los procesos del negocio de la organización. Está definido a través de tres artefactos: el diagrama de casos de uso del negocio, la descripción de los casos de uso del negocio y el diagrama de actividades de casos de uso del negocio.

Para desarrollar este modelo se realizaron las siguientes actividades:

1. Entrevista a los usuarios de los laboratorios.
2. Estudio de la información actual manejada en los laboratorios.
3. Definición de los casos de uso, actores y trabajadores del negocio.
4. Elaboración de los diagramas de actividad y de clases del modelo de objeto para cada caso de uso del negocio.

En los laboratorios de la Universidad tiene lugar el proceso de pedidos de tiempos de máquinas. Cuando los usuarios desean solicitar tiempo de máquina en la semana en curso deben acercarse al técnico del laboratorio. El técnico consulta una tabla de

Excel, en la cual están registrados los tiempos de máquina de la semana. Si no se puede satisfacer el pedido de tiempo de máquina porque no hay computadoras disponibles para la fecha y hora solicitadas, el técnico negocia con el usuario para otorgar otro tiempo de máquina.

Si el usuario desea solicitar tiempo de máquina para la semana entrante, debe hacerlo el viernes de la semana anterior. En ese caso el técnico realiza la misma operación explicada anteriormente. Los usuarios también pueden realizar pedidos de tiempo de máquina enviándole un correo electrónico al técnico del laboratorio.

En el caso de la Facultad de Informática, el técnico de laboratorio actualiza todos los lunes los tiempos de máquina asignados para la semana en curso y los publica en su página Web personal.

2.3 Reglas del negocio a considerar

En los laboratorios de la Universidad los usuarios pueden desarrollar actividades docentes programadas e independientes. Las actividades docentes pueden ser: de pregrado (del curso regular diurno y del curso para trabajadores) y de postgrados (diplomados, maestrías y doctorados).

Las actividades docentes de pregrado son planificadas semestralmente por el Departamento de Planificación de la UCf. Es decir, al inicio de cada semestre se pueden registrar la fecha y hora de todas las actividades a desarrollarse durante ese período.

Las actividades docentes de postgrado se realizan en horario de clases, de 8:00 a.m. a 6:00 p.m., según la disponibilidad de los laboratorios, en la fecha y hora que no hay planificadas actividades docentes de pregrado.

La planificación de ambos tipos de actividades puede sufrir cambios, estos cambios deben ser conveniados con los técnicos de laboratorios.

Los usuarios pueden realizar actividades independientes en los laboratorios fuera del horario de clases y dentro del horario, si no hay actividades docentes planificadas.

En el horario de clases, las máquinas son ocupadas por los usuarios para realizar actividades independientes según el orden de llegada.

Los pedidos de tiempos de máquinas se realizan para el horario extradocente (de 8:00 p.m. a 12:00 m). Durante los días jueves y viernes los usuarios pueden reservar un máximo de 3 turnos para la semana entrante. En la semana en curso pueden reservar como máximo otras 8 turnos, si hay disponibilidad.

Dos o más usuarios no pueden reservar en la misma máquina y hora, ni un mismo usuario puede tener otorgado tiempo de máquina en dos o más computadoras a la vez.

Pasados veinte minutos de comenzar cada tiempo de máquina, los usuarios pierden el derecho y el mismo puede ser ocupado por otros según el orden de llegada.

2.4 Modelación del modelo de negocio

En el presente epígrafe se realiza una descripción detallada del modelo de negocio a través de la definición de: los actores y trabajadores, el diagrama de casos de usos, la descripción de cada caso de uso y sus diagramas de actividades y de clases del modelo de objeto.

2.4.1 Actores y trabajadores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [13]

Los actores del negocio se listan a continuación:

Actores del Negocio	**Justificación**
Estudiante	Solicita tiempo de máquina.

Tabla 2.1: Actores del negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol. [13]

Se definen como trabajadores del negocio a:

Trabajadores del Negocio	**Justificación**
Técnico de Laboratorio	Gestiona las solicitudes de tiempos de máquinas.
Tabla de Excel	Registra las asignaciones de tiempos de máquinas.

Tabla 2.2: Trabajadores del negocio

2.4.2 Diagrama de Caso de Uso del Negocio

Diagrama de casos de uso del negocio

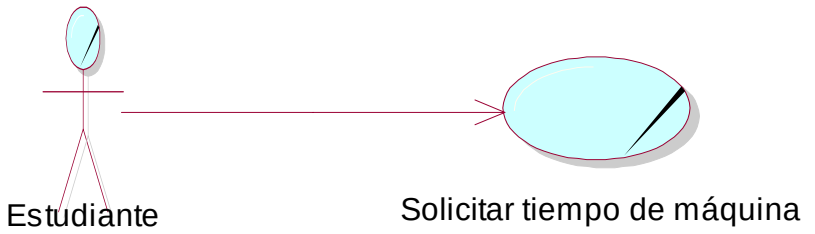


Figura 2.1: Diagrama de caso de uso del negocio

2.4.3 Descripción del caso de uso del negocio.

En la tabla 2. 3 se muestra en formato la descripción del caso de uso del negocio "Solicitar tiempo de máquina"

Caso de Uso del Negocio	**Solicitar tiempo de máquina**
Actores	Estudiante
Propósito	Asignar tiempo de máquina a un estudiante.
Resumen: El caso de uso se inicia cuando el estudiante solicita tiempo de máquina. El técnico de laboratorio chequea si hay disponibilidad de máquinas. En ese caso registra el tiempo de máquina. El caso de uso finaliza cuando el estudiante recibe la respuesta de a su solicitud.	
Acción del Actor	**Repuesta del proceso del Negocio**
1. El estudiante realiza la solicitud de tiempo de máquina.	

	2. El técnico de laboratorio recibe la solicitud. 3. El técnico de laboratorio le solicita a la tabla de Excel los registros de tiempos de máquinas. 4. La tabla de Excel brinda el registro de los tiempos de máquina. 5. El técnico de laboratorio chequea si hay tiempo de máquina disponible para la PC solicitada. De lo contrario se pasa al paso 7. 6. Registra el tiempo de máquina. Se pasa al paso 9. 7. Si no hay tiempo de máquina para la PC solicitada se ubica en otra PC disponible. Se pasa al paso 6. 8. Si no hay PC disponible se rechaza la solicitud.
9. El estudiante recibe la respuesta a su solicitud.	

Tabla 2.3: Descripción del caso de uso del negocio “Solicitar tiempo de máquina”

2.4.4 Diagramas de actividades del negocio

Los Diagramas de Actividades muestran una vista dinámica de un sistema. Muestran el flujo de una actividad a otra.

La especificación de UML, versión 1.3, describe los Diagramas de Actividades como un mecanismo para capturar los procesos del negocio, las acciones que se procesan y los flujos de ejecución de los casos de uso. En la figura 2.2 se muestra el diagrama de actividades para el caso de uso del negocio “Solicitar tiempo de máquina”.

Diagrama de actividades del caso de uso “Solicitar tiempo de máquina”

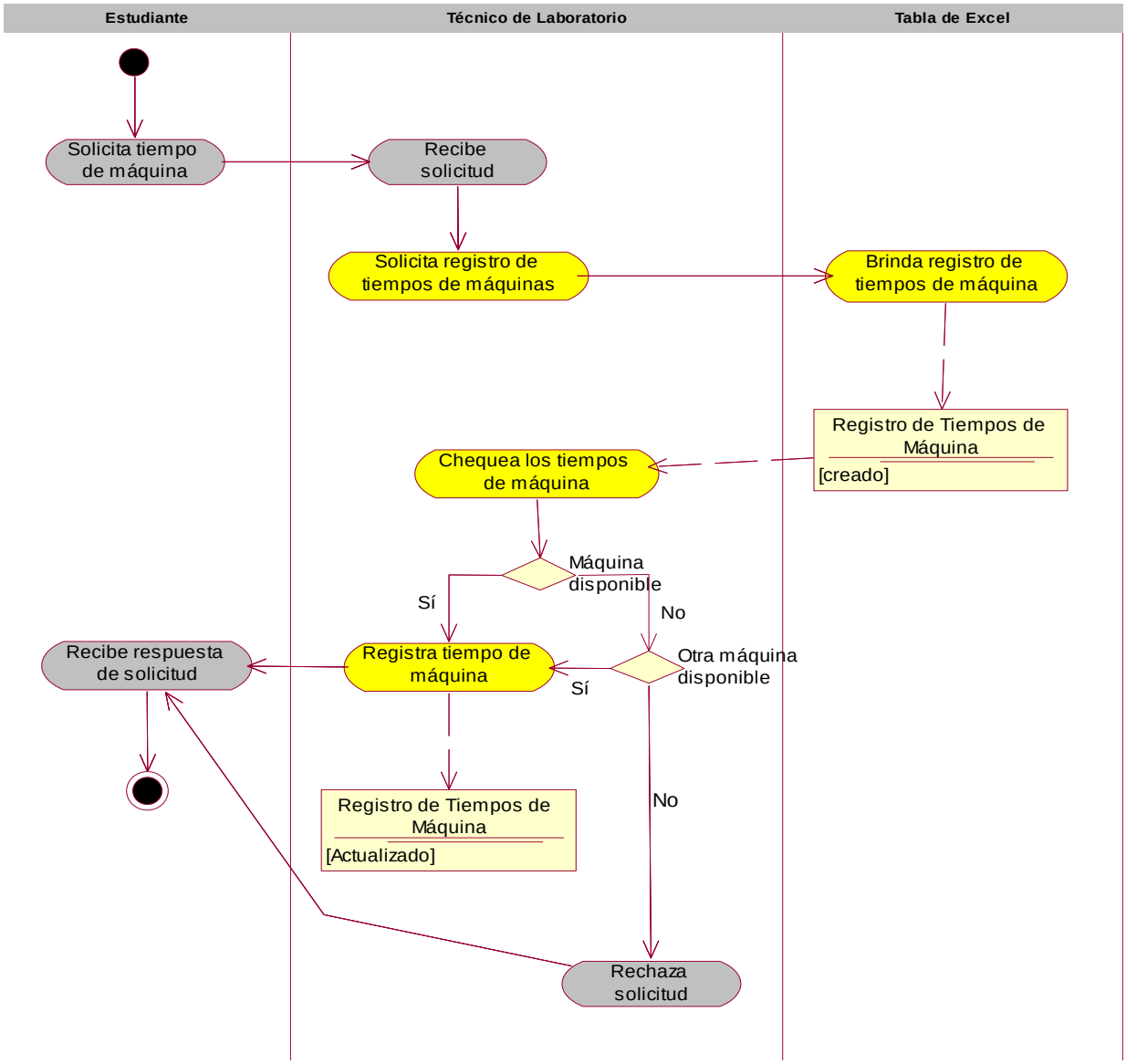


Figura 2.2: Diagrama de actividades del caso de uso “Solicitar tiempo de máquina”

2.4.5 Modelo de objetos del negocio

El modelo de objetos del negocio es un modelo que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio.

A continuación se muestra en la figura 2.3 el modelo de objetos del negocio.

Modelo de Objetos del Caso de Uso "Solicitar tiempo de máquina"

Figura 2.3: Diagrama de clases del modelo de objetos del caso de uso "Solicitar tiempo de máquina"

2.5 Conclusiones

En este capítulo fueron descritos los procesos que se ejecutan en los laboratorios de las facultades, específicamente en la Facultad de Informática; siendo identificados, además, los roles y entidades u objetos del negocio, así como su relación en esos procesos. Esta descripción fue realizada mediante el modelo del negocio, para lo cual se elaboraron los modelos de casos de uso y objetos del negocio. La modelación del negocio propició una mejor comprensión del problema que el sistema tiene que resolver y así, a partir de él, derivar los requerimientos para la aplicación.

Capítulo 3. Modelo de sistema

3.1 Introducción

En el presente capítulo se describe la solución propuesta utilizando algunos de los artefactos que propone la Metodología RUP. Los artefactos referidos son: los requerimientos funcionales y no funcionales, el diagrama de casos de uso del sistema y la descripción de cada uno, así como una descripción general del sistema que se propone.

3.2 Descripción del modelo del sistema

El sistema propuesto, dirigido a controlar la actividad que tiene lugar en los laboratorios docentes de la Universidad de Cienfuegos, se implementará en dos módulos o subsistemas principales: Gestión y Administración.

Módulo de Gestión: dirigido a gestionar, como su nombre lo indica, las actividades docentes y los tiempos de máquinas en los laboratorios.

Módulo de Administración: destinado a registrar y controlar toda la información necesaria para el funcionamiento del otro módulo.

3.3 Modelación del sistema

3.3.1 Requerimientos funcionales

Los requerimientos funcionales son una descripción de las necesidades o deseos de un producto. La meta primaria de la fase de requerimientos es identificar y documentar lo que en realidad se necesita, en una forma que claramente se lo comunique al cliente y a los miembros del equipo de desarrollo. [14].

El sistema propuesto debe tener las siguientes funcionalidades:

1. Agregar computadora a un laboratorio.
2. Eliminar computadora de un laboratorio.
3. Modificar datos de una computadora.
4. Agregar usuario.

5. Eliminar usuario.
6. Modificar datos de un usuario.
7. Agregar nuevo software.
8. Eliminar software.
9. Visualizar toda la información de una computadora determinada.
10. Agregar laboratorio.
11. Modificar datos de un laboratorio.
12. Eliminar laboratorio.
13. Mostrar las actividades docentes planificadas.
14. Mostrar los tiempos de máquinas disponibles.
15. Realizar autenticación.

    15.1 Brindar el identificador y la contraseña y acceder al sistema si ambos son válidos.

16. Agregar actividad docente.
17. Eliminar actividad docente.
18. Modificar actividad docente.
19. Cambiar el estado de las computadoras que presenten dificultades.
20. Registrar tiempo de máquina.
21. Insertar un nuevo turno de laboratorio.
22. Modificar un turno existente.
23. Eliminar un turno de laboratorio.
24. Emitir reporte con estado de las computadoras.

    24. 1 El estado de las computadoras puede ser Activo o Inactivo.

25. Reportar PC con problemas.
26. Mostrar las computadoras reportadas con problemas.

3.3.2 Requerimientos no funcionales

Los requisitos no funcionales especifican cualidades, propiedades del sistema; como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, etc. Requisito que especifica restricciones físicas sobre un requisito funcional. [15]

Para la propuesta de sistema se plantean los siguientes requisitos no funcionales:

Apariencia o interfaz externa

Se desarrollará una interfaz WEB para los usuarios, la cual debe ser amigable, comprensible y permitirá una navegación fácil y rápida. El contenido se mostrará de forma clara y concreta.

Se evitarán las imágenes y animaciones grandes, con el fin de agilizar la navegación en el sitio.

Usabilidad

El sistema puede ser utilizado por los estudiantes, profesores y administradores. Los usuarios del sistema no necesitan tener amplios conocimientos en el manejo de la computadora y en el ambiente Web. La interfaz Web tiene dos módulos, uno para cada tipo de usuario. Cada módulo tiene opciones de acceso a la base de datos de acuerdo al tipo de usuario que sea y los privilegios que tenga.

Rendimiento

Se concibe un sistema diseñado sobre la arquitectura cliente/servidor, de manera que se cuenta con varios terminales en la red. Se requiere de una capacidad de procesamiento alta. Los tiempos de respuesta deben ser cortos.

La información deberá estar disponible las 24 horas del día, aunque una caída momentánea del sistema no provocará el paro de las tareas que realizan los usuarios que están conectados; sin embargo, el sistema deberá recuperarse en un corto período de tiempo ante una falla.

Soporte

La instalación del sistema será responsabilidad del administrador de cada laboratorio y la carga inicial y el mantenimiento, de los administradores de la redes de la facultades.

El sistema brindará la posibilidad de futuras mejoras y nuevas opciones que se deseen incorporar.

Legales

El sistema se desarrollará con el gestor de bases de datos MySQL y el lenguaje de programación PHP, ambos son tecnologías “open source”. Por tanto, para la comercialización del producto no es necesaria la adquisición de licencias.

Confiabilidad

Es importante que el sistema presente un mecanismo de respuesta rápida ante fallos y que en caso de ocurrencia se minimicen las pérdidas de información.

La aplicación debe ser capaz de almacenar información consistente y de prever mecanismos para la recuperación y control de datos.

Software

La aplicación necesita para su ejecución MySQL como sistema gestor de base de datos y Apache como servidor Web. En las terminales clientes solo se necesitará un navegador Web.

Hardware

Se requiere disponer de una computadora como servidor de base de datos con los requerimientos de hardware que necesita MySQL. Además se requiere una computadora como servidor Web Apache. Las terminales clientes sólo requerirán una computadora conectada a la red.

Seguridad

El sistema debe garantizar seguridad en el manejo de los datos almacenados y enviados a través de la red. Para esto se debe tener en cuenta:

Establecer niveles de privilegio para los diferentes tipos de usuarios.

Se garantizará que cada usuario tenga disponible solamente las opciones asociadas a su nivel de privilegio.

Cifrado de los datos almacenados en la base de datos, utilizando para ellos la función MD5.

Estas medidas de seguridad no impedirán que los usuarios accedan de manera rápida y operativa al sistema.

3.3.3 Actores del sistema

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interaccionan con estos Casos de Uso. Los actores representan terceros fuera del sistema que colaboran con el mismo. Una vez que se han identificado los autores del sistema, tenemos identificado el entorno externo del sistema. [16]

En la tabla 3.1 se muestran los actores del sistema.

Actor	Descripción
Usuario	Encargado de registrar la solicitud de tiempo de máquina, puede además visualizar los tiempos de máquina así como las actividades docentes programadas, debe autenticarse en el sistema. Requisitos funcionales asociados: 9, 13,14, 15, 20, 26.
Administrador	Realiza las mismas accione que un Usuario y además puede actualizar la información concerniente a los usuarios, las computadoras, los laboratorios y las actividades docentes. Requisitos funcionales asociados:1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 16, 17, 18, 19.

Tabla 3.1: Actores del sistema

3.3.4 Casos de uso del sistema.

Cada forma en que los actores usan el sistema se representa con un Caso de Uso. Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un Caso de

Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. [17]

Los casos de uso definidos para el sistema se listan a continuación:

1. Gestionar PC.
2. Gestionar usuario.
3. Gestionar software.
4. Gestionar laboratorio.
5. Gestionar actividades docentes.
6. Deshabilitar PC con dificultades.
7. Gestionar turno.
8. Visualizar reporte de estado de las PC.
9. Visualizar PC reportadas con problemas.
10. Mostrar tiempos de máquina disponibles.
11. Mostrar las actividades docentes planificadas.
12. Visualizar detalles de una PC.
13. Registrar tiempo de máquina.
14. Autenticarse.
15. Reportar PC con problemas.

3.3.5 Paquetes y sus relaciones

Los paquetes constituyen un mecanismo que posibilitan una mejor comprensión del sistema organizando sus elementos. Dividen el modelo en otros más pequeños que colaboran entre sí y se crean atendiendo a diferentes criterios, en este caso, el criterio seleccionado fue agrupar los casos de uso requeridos para dar soporte a cada actor del sistema.

Para modelar el sistema propuesto se crearon dos paquetes: Administración y Gestión. En la figura 3.1 se muestra la relación de dependencia entre ambos paquetes.

Diagrama de casos de uso por paquetes

Figura 3.1: Diagrama de casos de uso por paquetes

3.3.6 Diagrama de caso de uso del sistema

Un diagrama de casos de uso explica gráficamente un conjunto de casos de uso del sistema, los actores y la relación entre éstos y los casos de uso. Tiene por objeto establecer una clase de diagrama contextual que nos permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan. [18]

En las figuras 3.2 y 3.3 se muestran los diagramas de casos de uso correspondientes a los paquetes de Administración y Gestión respectivamente.

Paquete *Administración.* Diagrama de casos de uso.

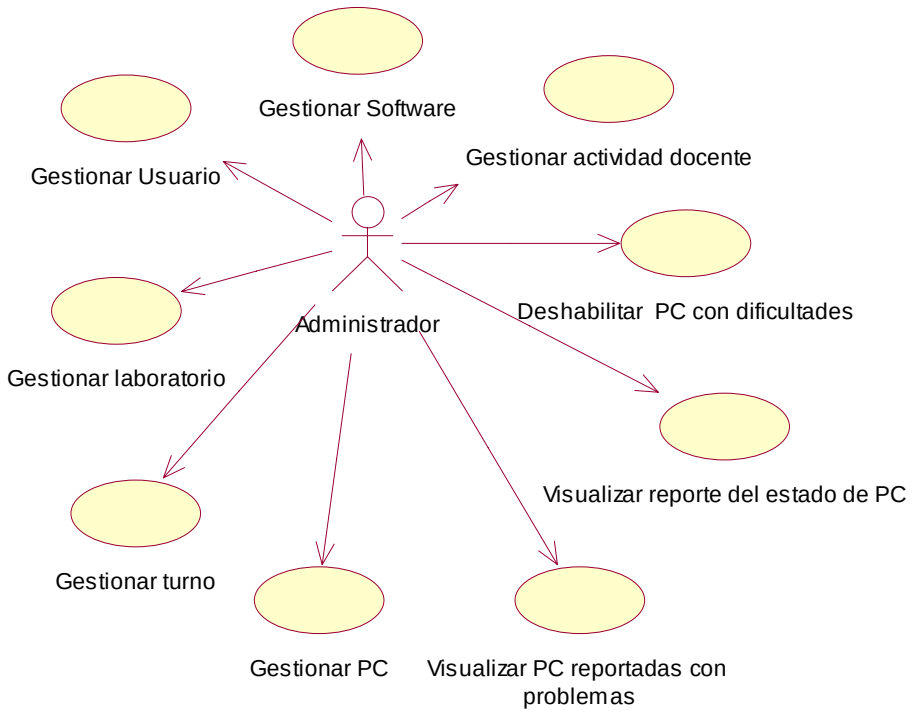


Figura 3.2: Paquete Administración. Diagrama de casos de uso del sistema.

Paquete *Gestión.* Diagrama de casos de uso.

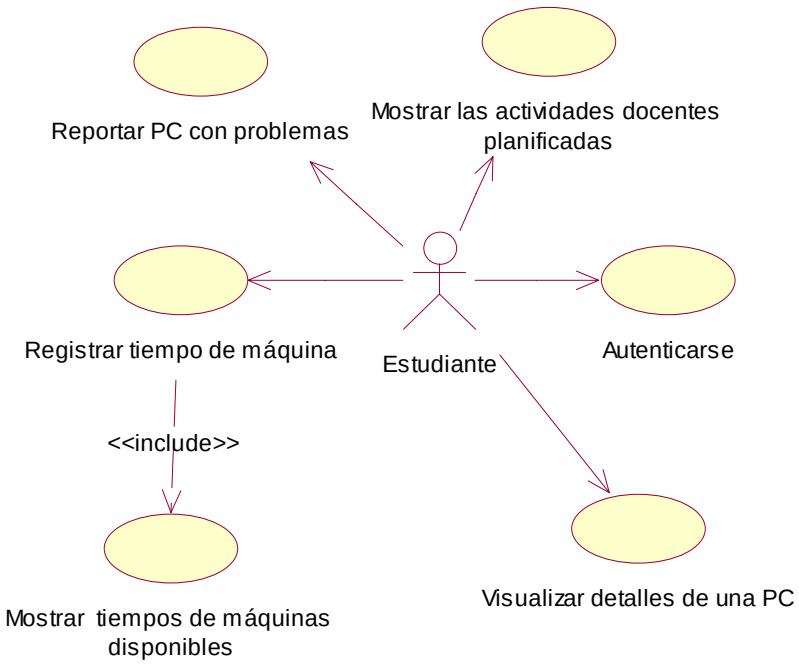


Figura 3.3: Paquete Gestión. Diagrama de casos de uso del sistema.

3.3.7 Descripción de los casos de usos del sistema

Paquete de Administración

CU-1	**“Gestionar PC”**
Actores	Administrador
Propósito	Permitir insertar y eliminar una PC de un laboratorio, así como modificar sus datos.
Resumen	El caso de uso se inicia cuando el Administrador desea insertar una computadora, eliminar una computadora o modificar los datos de una computadora de un laboratorio dado. En caso de insertar, el Administrador debe proporcionar los datos de la computadora. Si desea eliminar o modificar el Administrador debe indicar la computadora. En todos los casos debe especificarse el laboratorio. El caso de uso finaliza cuando se inserta, se elimina o se modifican los datos de una computadora en un laboratorio.
Referencias	R1, R2, R3
Precondiciones	-
Poscondiciones	Se actualizan las computadoras en el laboratorio seleccionado. • Si acción: insertar, se inserta una nueva computadora. • Si acción: modificar, se modifican los datos de una computadora. • Si acción: eliminar, se elimina computadora.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.1**

CU-2	**“Gestionar usuario”**
Actores	Administrador
Propósito	Permitir insertar y eliminar un usuario, así como modificar sus datos.
Resumen	El caso de uso se inicia cuando el Administrador desea insertar un usuario, eliminar un usuario o modificar los datos del mismo. En caso de insertar, el Administrador debe proporcionar los datos del usuario. Si desea eliminar o modificar el Administrador debe indicar el usuario. El caso de uso finaliza cuando se inserta, se elimina o se modifican los datos de un usuario.
Referencias	R4, R5, R6
Precondiciones	-
Poscondiciones	Se actualizan los usuarios • Si acción: insertar, se inserta un nuevo usuario. • Si acción: modificar, se modifican los datos de un usuario. • Si acción: eliminar, se elimina un usuario.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.2**

CU-3	**“Gestionar software”**
Actores	Administrador
Propósito	Permite agregar o eliminar un software.
Resumen	El caso de uso se inicia cuando el Administrador desea agregar o eliminar un software. En caso de insertar el Administrador debe proporcionar los datos necesarios. El caso de uso finaliza cuando se actualizan los softwares.
Referencias	R7, R8
Precondiciones	-
Poscondiciones	Se actualizan los software • Si acción: insertar, se inserta un nuevo software. • Si acción: eliminar, se elimina el software seleccionado.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.3**

CU-4	**“Gestionar laboratorios”**
Actores	Administrador
Propósito	Agregar un nuevo laboratorio, eliminar o modificar los datos de uno ya existente.
Resumen	El caso de uso se inicia cuando el Administrador desea agregar un laboratorio, o modificar los datos de un laboratorio existente. En caso de agregar un nuevo laboratorio el Administrador debe introducir los datos del mismo. El caso de uso finaliza cuando se actualizan los laboratorios.
Referencias	R10, R11, R12
Precondiciones	-

Poscondiciones	Se actualizan los laboratorios. • Si acción: insertar, se inserta un nuevo laboratorio. • Si acción: eliminar, se elimina un laboratorio existente. • Si acción: modificar, se modifican los datos del laboratorio.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.4**

CU-5	**“Gestionar actividades docentes”**
Actores	Administrador
Propósito	Permitir agregar, eliminar o modificar una actividad docente en un laboratorio.
Resumen	El caso de uso se inicia cuando el Administrador desea agregar, eliminar o modificar una actividad docente. En caso de insertar una nueva actividad, el Administrador debe proporcionar los datos necesarios. En caso de eliminar debe seleccionar la actividad que desea eliminar. Si se desea modificar debe determinarse la actividad a modificar. El caso de uso finaliza cuando quedan actualizadas las actividades docentes.
Referencias	R16, R 17, R18
Precondiciones	.
Poscondiciones	Se actualizan las actividades docentes en los laboratorios. • Si acción: insertar, se inserta una nueva actividad docente. • Si acción: eliminar, se elimina la actividad docente seleccionada. • Si acción: modificar, se modifican los datos de una actividad

	docente determinada.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.5**

CU-6	**“Deshabilitar PC con dificultades”**
Actores	Administrador
Propósito	Permitir reportar las computadoras con problemas técnicos.
Resumen	El caso de uso se inicia cuando el Administrador desea modificar el estado de una computadora por presentar la misma algún desperfecto técnico. El caso de uso finaliza cuando queda notificada la computadora
Referencias	R19
Precondiciones	
Poscondiciones	Se actualiza el estado de una computadora.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.6**

CU-7	**“Gestionar turnos”**
Actores	Administrador
Propósito	Permitir agregar, eliminar o modificar un turno.

Resumen	El caso de uso se inicia cuando el Administrador desea agregar, eliminar o modificar un turno. En caso de insertar un nuevo turno, el Administrador debe proporcionar los datos necesarios. En caso de eliminar y modificare debe seleccionar el turno que desee eliminar o modificar. El caso de uso finaliza cuando quedan actualizados los turnos de laboratorio.
Referencias	R21, R 22, R23
Precondiciones	.
Poscondiciones	Se actualizan los turnos. • Si acción: insertar, se inserta un nuevo turno. • Si acción: eliminar, se elimina un turno. • Si acción: modificar, se modifican los datos de un turno.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.7**

CU-8	**“Visualizar reporte del estado de las PC”**
Actores	Administrador
Propósito	Permitir mostrar un reporte del estado de las computadoras de todos los laboratorios.
Resumen	El caso de uso se inicia cuando el Administrador desea obtener un reporte del estado de todas las computadoras de los laboratorios, mostrándose el nombre de la máquina y si está activa o inactiva. El caso de uso finaliza cuando se muestra el reporte.
Referencias	R24
Precondiciones	.
Poscondiciones	Se crea el reporte.

Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.8**

CU-9	**"Visualizar PC reportadas con problemas"**
Actores	Administrador
Propósito	Permitir mostrar en un reporte las computadoras que han sido reportadas con problemas.
Resumen	El caso de uso se inicia cuando el Administrador desea visualizar las computadoras que han sido reportadas con problemas. En dicho reporte se mostrará la fecha en que fue reportada y el motivo por el cual fue reportada. El caso de uso finaliza cuando se visualiza el reporte.
Referencias	R25
Precondiciones	.
Poscondiciones	Se muestra el reporte.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.9**

Paquete Gestión.

CU-10	**"Mostrar los tiempos de máquinas disponibles"**
Actores	Usuario
Propósito	Permitir visualizar los tiempos de máquinas disponibles en cada computadora.
Resumen	El caso de uso se inicia cuando se registra un nuevo tiempo de máquina y el sistema muestra los tiempos disponibles en cada máquina. El caso de uso finaliza cuando son mostrados los tiempos de máquina disponibles.

Referencias	R14
Precondiciones	Se registra un nuevo tiempo de máquina
Poscondiciones	Se muestra un reporte con los tiempos de máquina disponibles
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.10**

CU-11	**“Mostrar las actividades docentes planificadas”**
Actores	Usuario
Propósito	Permitir visualizar las actividades docentes planificadas en los laboratorios.
Resumen	El caso de uso se inicia cuando un Usuario desea conocer las actividades docentes que se van a realizar en los laboratorios. Se muestran en forma de tabla las actividades en cada laboratorio, son su fecha y hora correspondiente. El caso de uso finaliza cuando se muestra la información solicitada.
Referencias	R13
Precondiciones	
Poscondiciones	-
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.11**

CU-12	**“Visualizar detalles de PC”**
Actores	Usuario
Propósito	Permitir mostrar los datos de una computadora determinada.
Resumen	El caso de uso se inicia cuando el Usuario desea conocer los datos de una computadora. El Usuario debe seleccionar la computadora de la que desea conocer los datos.
Referencias	R9
Precondiciones	
Poscondiciones	Se muestra los datos de la computadora seleccionada.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.12**

CU-13	**“Registrar tiempo de máquina”**
Actores	Usuario
Propósito	Permitir registrar un tiempo de máquina.
Resumen	El caso de uso se inicia cuando un Usuario desea solicitar tiempo de máquina. El sistema muestra los tiempos de máquinas disponibles en la semana en curso. A partir del jueves, muestra los de la siguiente semana. El Usuario, brinda los datos necesarios para hacer la solicitud, si son correctos se registra, de lo contrario se muestra un mensaje de error. El caso de uso finaliza cuando se registra el tiempo de máquina solicitado.
Referencias	R20 **Casos de uso asociados** Mostrar los tiempos de máquinas disponibles (Include)
Precondiciones	Deben existir tiempos de máquina disponibles.
Poscondiciones	Se registra un nuevo tiempo de máquina.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.13**

CU-14	**“Autenticarse”**
Actores	Usuario
Propósito	Autenticarse en el sistema.
Resumen	El caso de uso se inicia cuando un Usuario desea entrar al sistema, para ello es necesario autenticarse. El Usuario debe brindar su identificador y contraseña. Luego el sistema verifica que sus datos sean correctos, en ese caso, entra al sistema y puede ver las opciones según su tipo de usuario, de lo contrario, se muestra un mensaje de error. El caso de uso finaliza cuando el Usuario se autentica.
Referencias	R15
Precondiciones	
Poscondiciones	Se muestran las opciones disponibles al usuario que entrado al sistema.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.14**

CU-15	**“Reportar PC con problemas”**
Actores	Usuario
Propósito	Permitir reportar una PC con problemas.
Resumen	El caso de uso se inicia cuando un Usuario desea reportar una computadora con problemas. El Usuario debe indicar la computadora que desea reportar, así como el motivo por el cual la reporta. El caso de uso finaliza cuando el Usuario reporta la computadora.

Referencias	R25
Precondiciones	
Poscondiciones	Se reporta una computadora con problemas.
Requerimientos Especiales	-
Prototipo	**Ver Anexo 1.15**

3.4 Conclusiones.

Buscando establecer un común entendimiento entre los usuarios y el desarrollador fueron definidos, en este capítulo, los requerimientos funcionalidades y no funcionales del sistema, e identificados los casos de uso y actores, representando sus relaciones en el diagrama de casos de uso. Los casos de uso fueron descritos en formato de alto nivel, mostrándose los prototipos correspondientes a cada uno de ellos.

Capítulo 4. Construcción del sistema.

4.1 Introducción

En este capítulo se describe la construcción de la solución propuesta mostrándose los diagramas de clases, el modelo lógico y físico de los datos y el diagrama de implementación, brindando estos artefactos una descripción de la fase de implementación del sistema.

4.1.1 Diagrama de clases.

Un diagrama de clases presenta las clases del sistema con sus relaciones (estructurales y de herencia). En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones entre las páginas, donde cada página lógica puede ser representada como una clase.

Los diagramas de clases Web, fueron realizados, a partir de los casos de uso definidos en el epígrafe 3.3.4. A continuación se muestran los mismos.

Paquete *Administración.* Caso de Uso: *Gestionar PC*

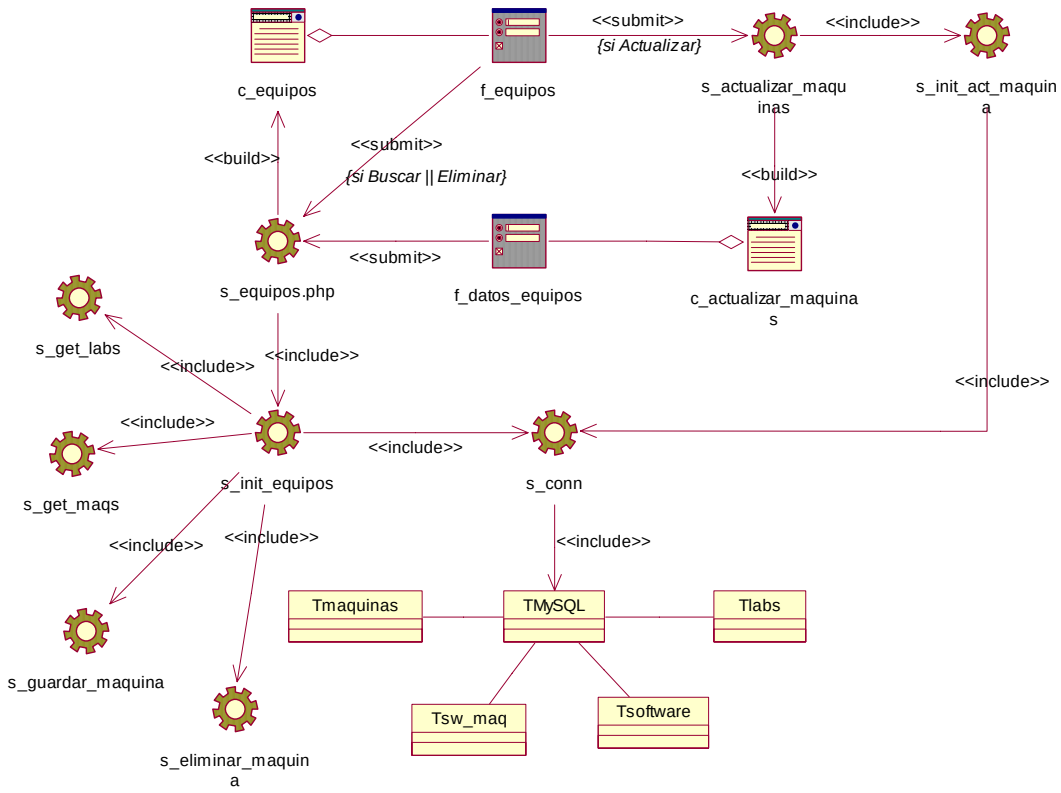


Figura 4.1 Diagrama de clases web. Caso de uso: Gestionar PC

Paquete *Administración.* Caso de Uso: *Gestionar usuario*

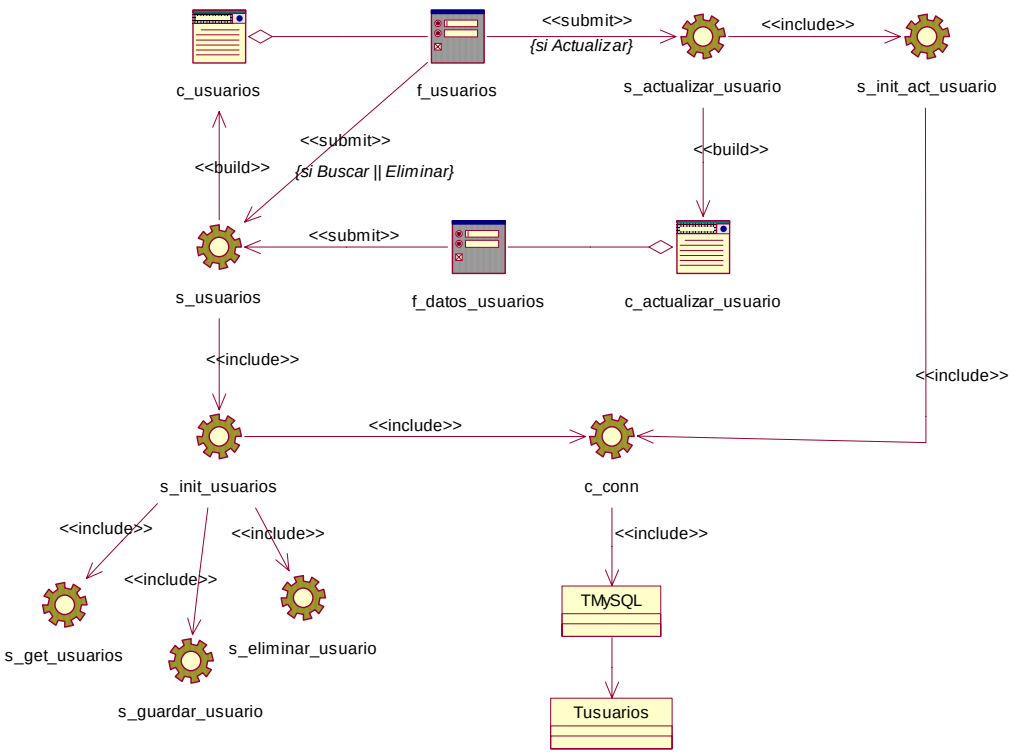


Figura 4.2 Diagrama de clases web. Caso de uso: Gestionar usuario

Paquete *Administración.* Caso de Uso: *Gestionar software*

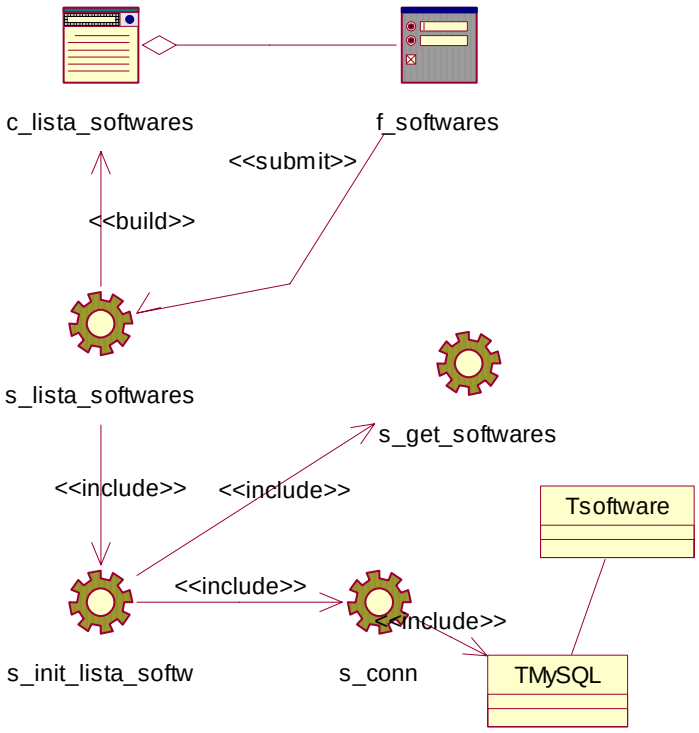


Figura 4.3 Diagrama de clases web. Caso de uso: Gestionar software.

Paquete *Administración.* Caso de Uso: *Gestionar laboratorio*

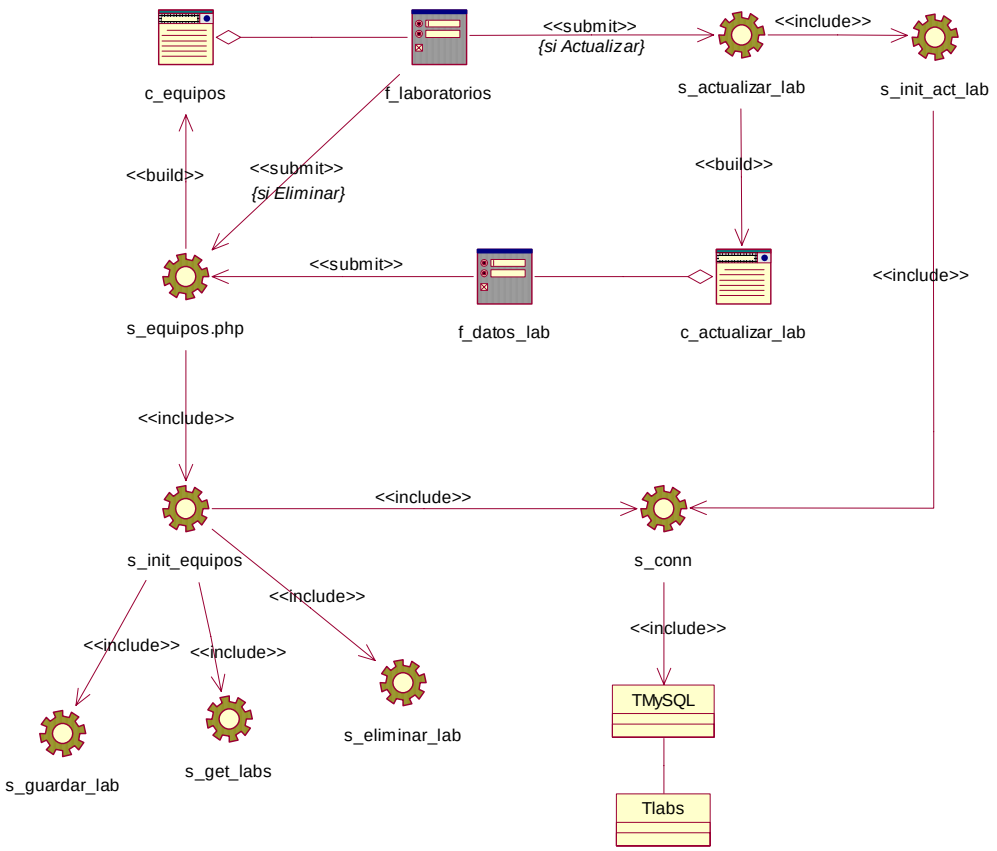


Figura 4.4 Diagrama de clases web. Caso de uso: Gestionar laboratorio

Paquete *Administración.* Caso de Uso: *Gestionar actividades docentes*

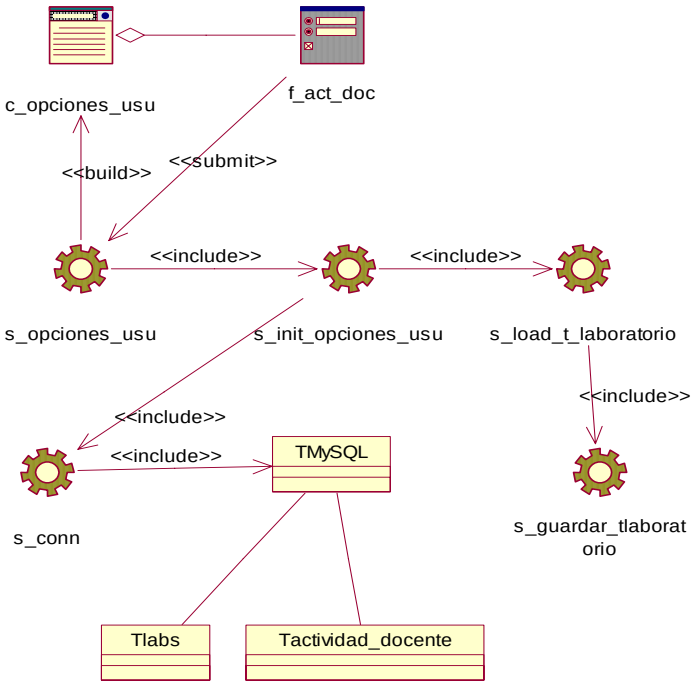


Figura 4.5 Diagrama de clases web. Caso de uso: Gestionar actividades docentes

Paquete *Administración.* Caso de Uso: *Gestionar turno*

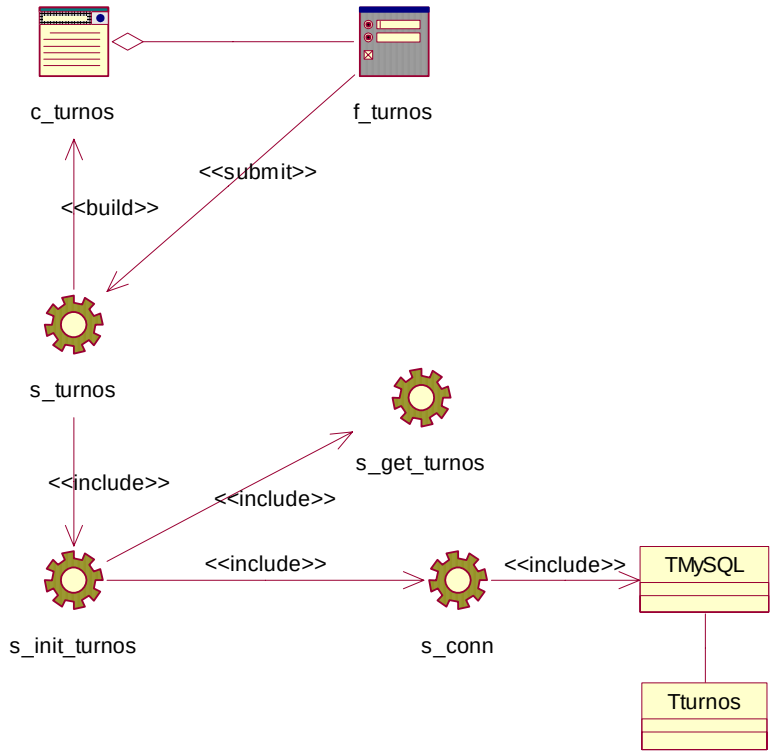


Figura 4.6 Diagrama de clases web. Caso de uso: Gestionar turno

Paquete *Administración.* Caso de Uso: *Deshabilitar PC con dificultades*

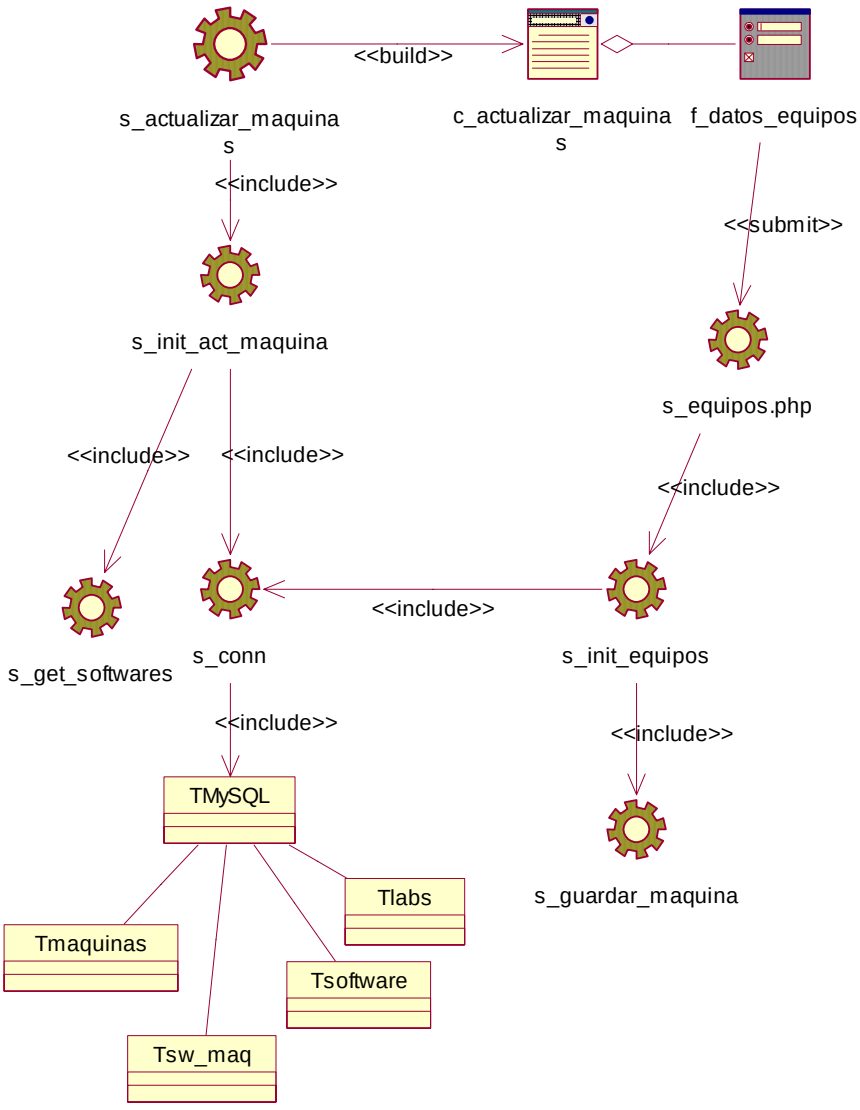


Figura 4.7 Diagrama de clases web. Caso de uso: Deshabilitar PC con dificultades

Paquete *Administración.* Caso de Uso: *Visualizar reporte de estado de las PC*

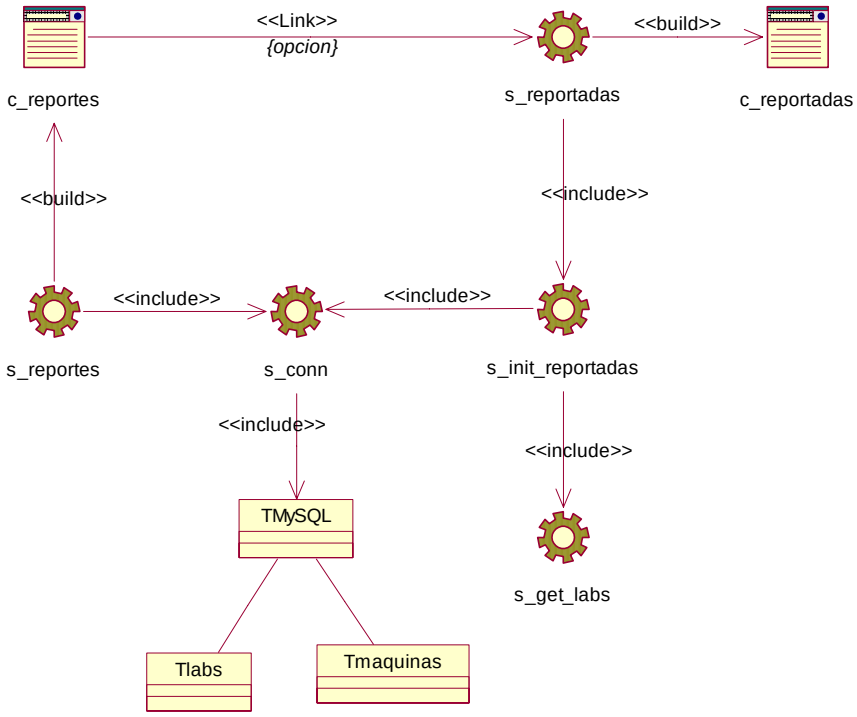


Figura 4.8 Diagrama de clases web. Caso de uso: Visualizar reporte de estado de las PC

Paquete *Administración.* Caso de Uso: *Visualizar PC reportadas con problemas*

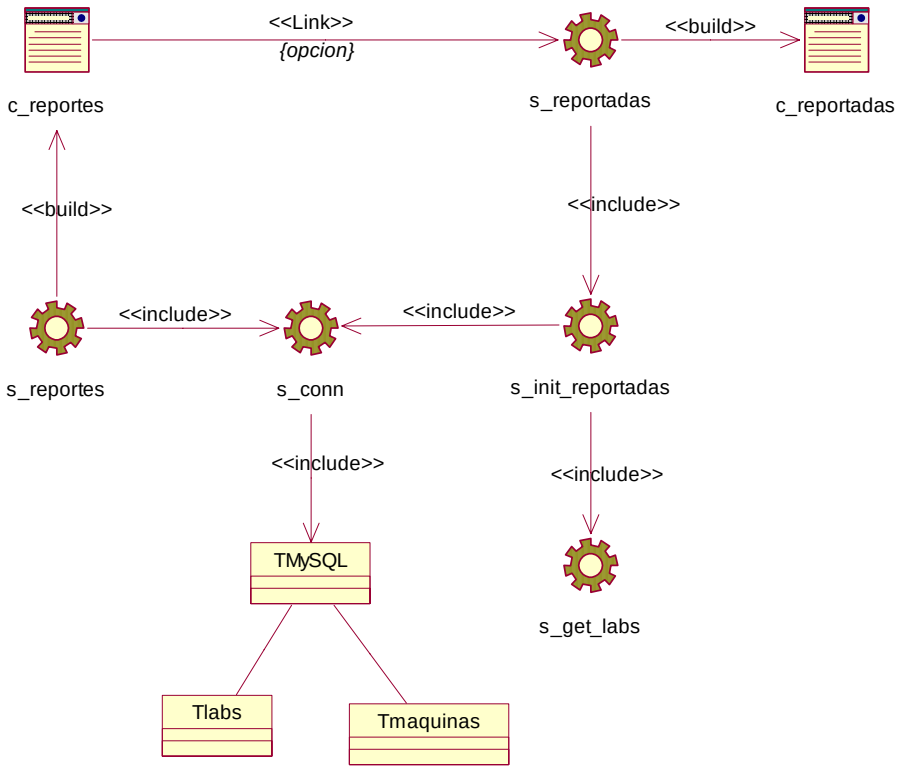


Figura 4.9 Diagrama de clases web. Caso de uso: Visualizar PC reportadas con problemas

Paquete *Gestión.* Caso de Uso: *Mostrar tiempos de máquina disponibles*

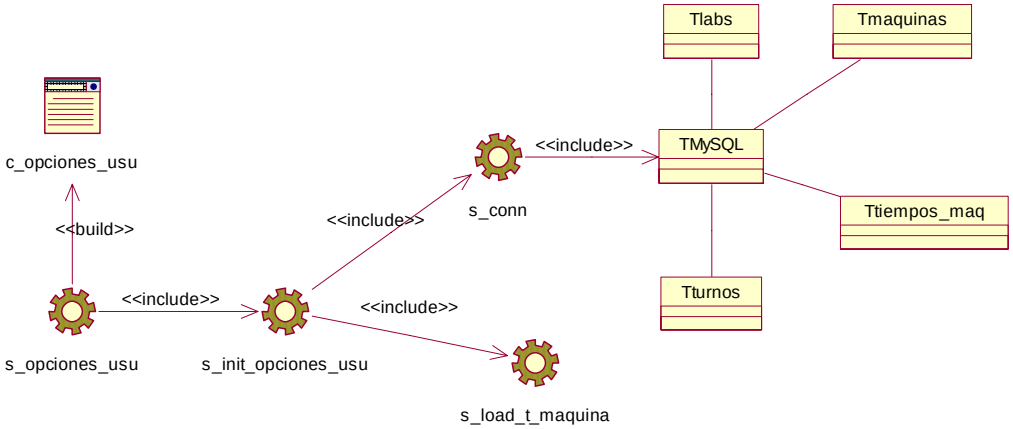


Figura 4.10 Diagrama de clases web. Caso de uso: Mostrar tiempos de máquina disponibles

Paquete *Gestión.* Caso de Uso: *Mostrar actividades docentes planificadas*

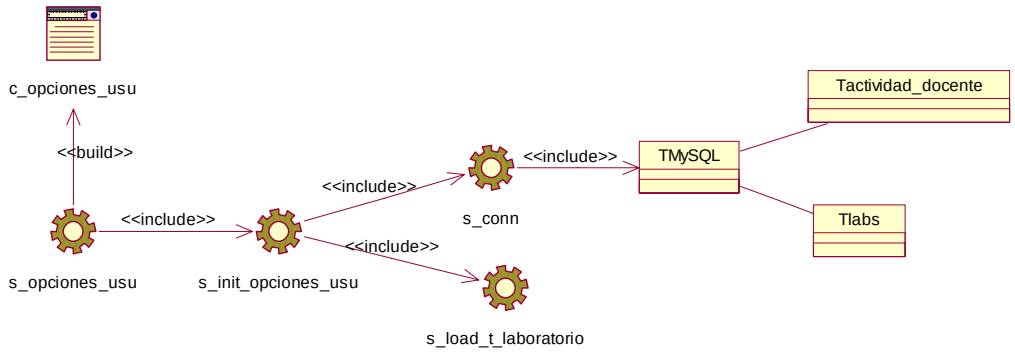


Figura 4.11 Diagrama de clases web. Caso de uso: Mostrar actividades docentes planificadas

Paquete *Gestión.* Caso de Uso: *Visualizar detalles de PC*

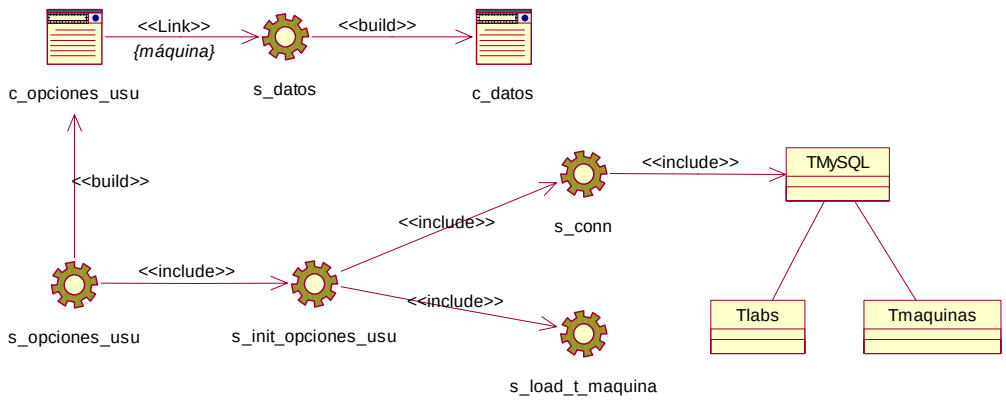


Figura 4.12 Diagrama de clases web. Caso de uso: Visualizar detalles de PC

Paquete *Gestión.* Caso de Uso: *Registrar tiempo de máquina*

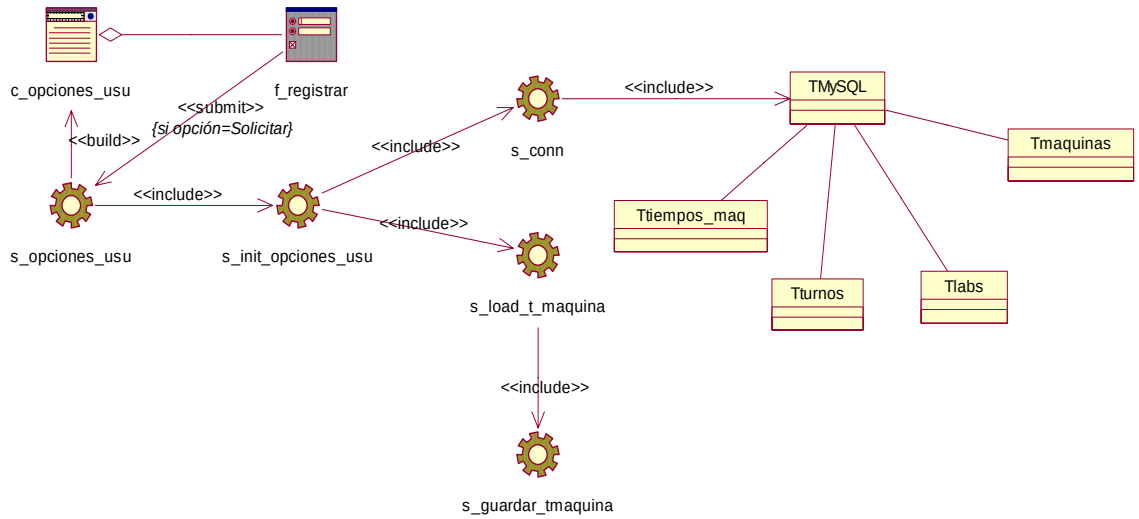


Figura 4.13 Diagrama de clases web. Caso de uso: Registrar tiempo de máquina

Paquete *Gestión.* Caso de Uso: *Autenticarse*

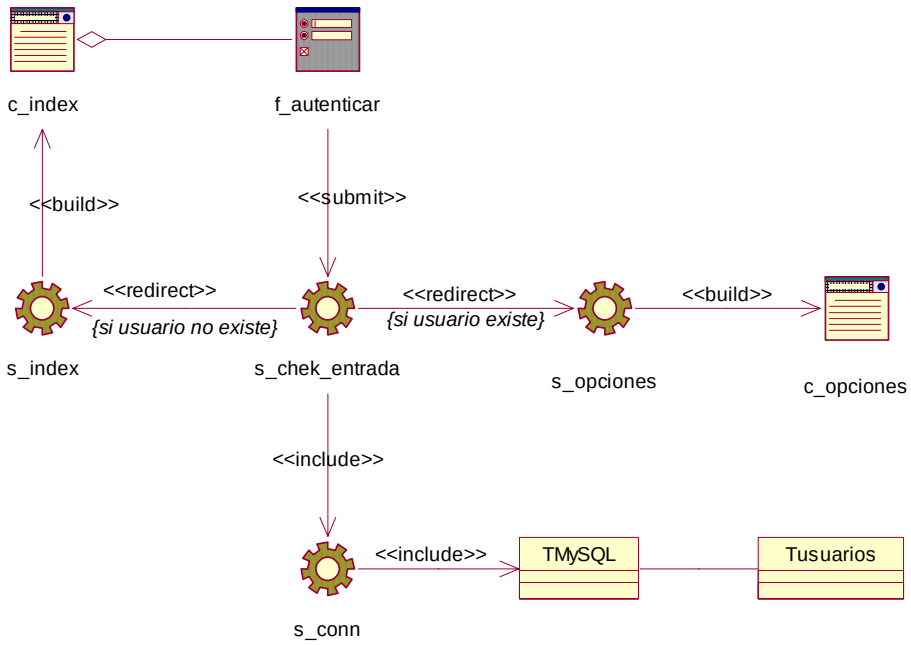


Figura 4.14 Diagrama de clases web. Caso de uso: Autenticarse

Paquete *Gestión.* Caso de Uso: *Reportar PC con problemas*

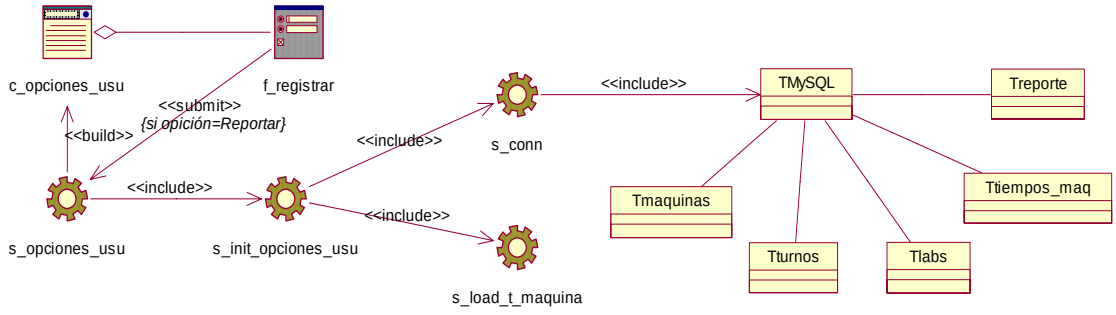


Figura 4.15 Diagrama de clases web. Caso de uso: Reportar PC con problemas

4.1.2 Diagramas del modelo lógico de datos.

El modelo lógico de datos es usado para explorar el diseño conceptual y los detalles de la arquitectura de una base de datos. Muestra las entidades lógicas, sus atributos y las relaciones entre ellas. En la Figura 4.16 se muestra el modelo lógico del sistema.

Modelo lógico de datos

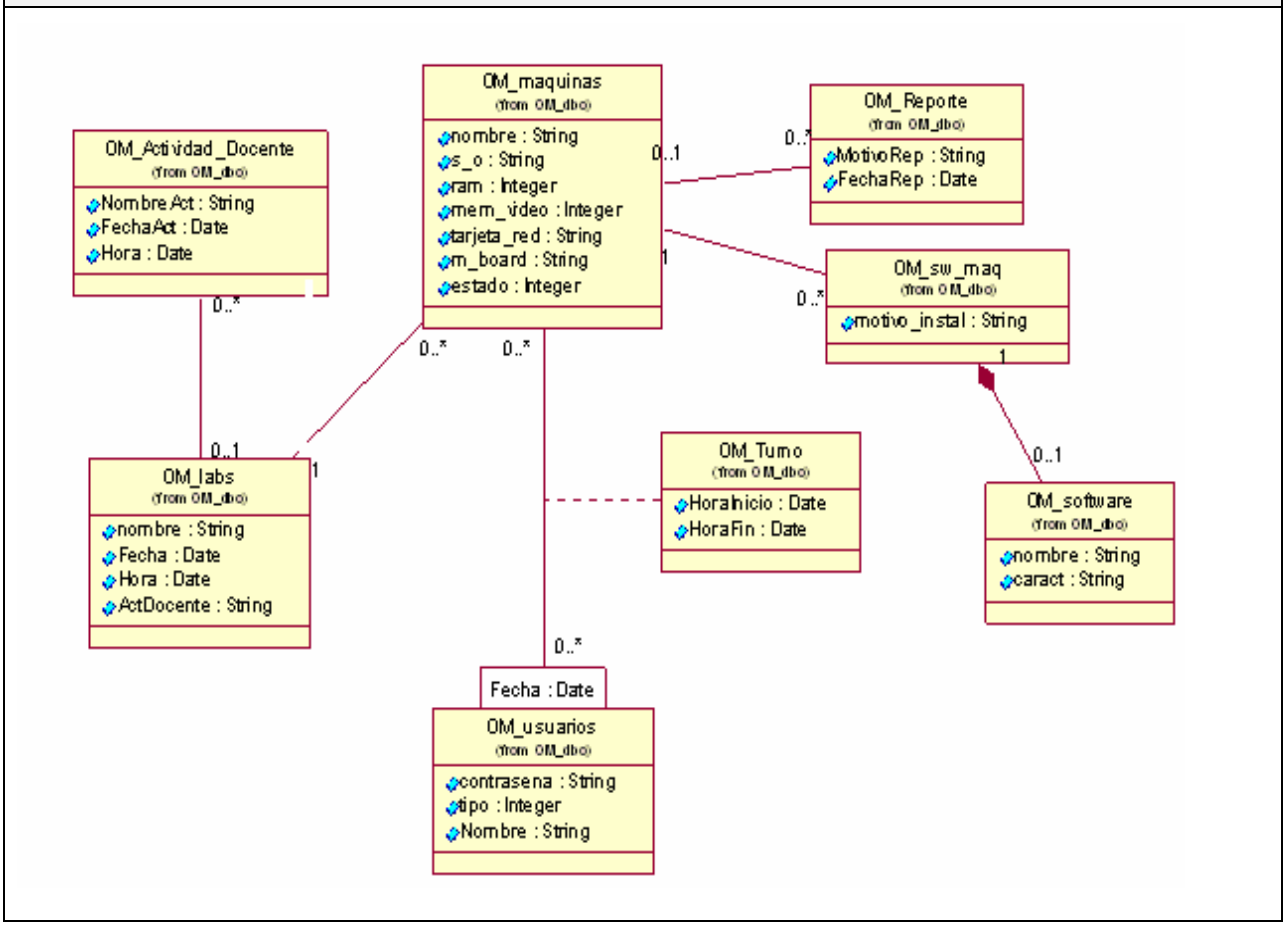


Figura 4.16 Modelo lógico

4.1.3 Diagramas del modelo físico de datos

El modelo físico de datos es el diseño del esquema interno de la base de datos, mostrando las tablas y las relaciones entre estas. En la figura 4.17 se muestra el modelo físico de los datos.

Modelo físico de datos

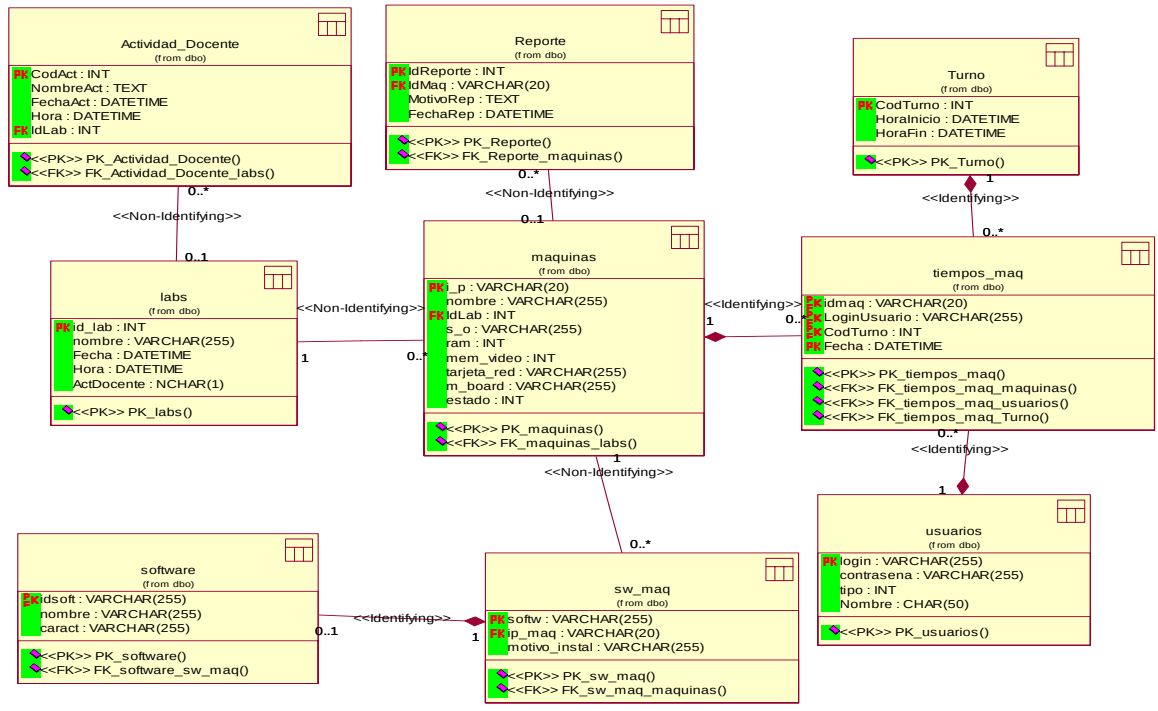


Figura 4.17 Modelo físico.

4.1.4 Diagramas de implementación

El modelo de implementación describe como los elementos del modelo de diseño, como las clases, se implementan en términos de componentes. Describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y como dependen los componentes unos de otros. [19]

Un diagrama de implementación es un diagrama que uniendo los modelos de componentes y de despliegue muestra en una sola vista los nodos y componentes. Aunque se pueden construir por separado estos diagramas, en ellos no queda clara la relación directa que puede darse entre componentes ubicados en nodos diferentes.

Diagrama de Implementación

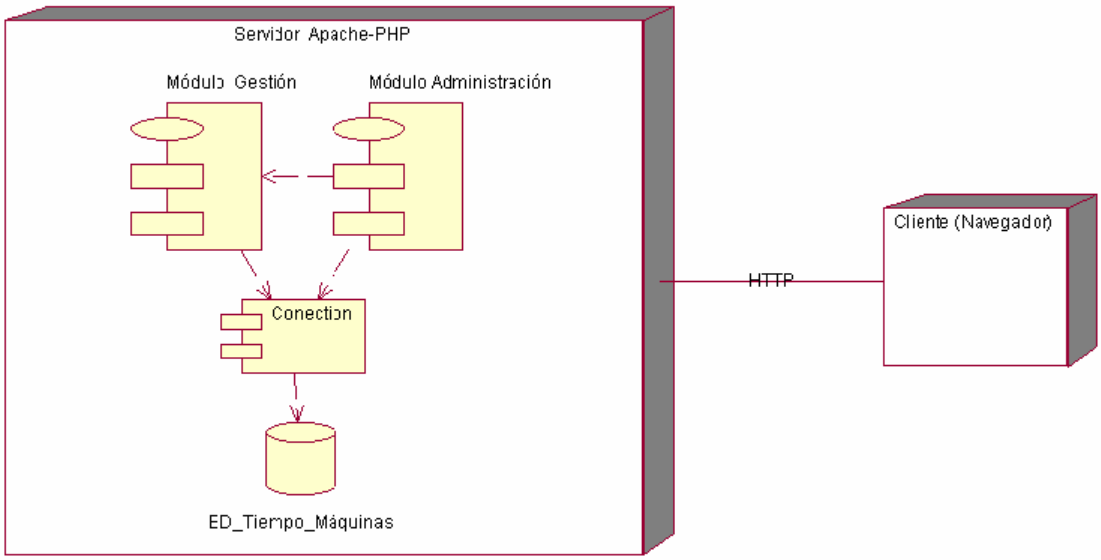


Figura 4.18 Diagrama de implementación.

4.2 – Principios de diseño del sistema.

En este epígrafe se listan los principios que se tuvieron en cuenta en el diseño del sistema propuesto.

4.2.1 Estándares en la interfaz de la aplicación.

La interfaz diseñada para ambos módulos del sistema está estrechamente vinculada y presenta las siguientes características:

El diseño será conservador, el tipo de letra utilizada es Times New Roman de estilo regular y tamaño variado atendiendo a la cantidad de información que se muestre.

La carga visual es adecuada y el lenguaje de las opciones que han sido utilizadas es de fácil comprensión para el usuario.

Facilidad para su navegabilidad y uso.

La entrada de información por parte de los usuarios se realiza a través de los formularios.

Las interacciones se basan en selecciones de tipo menú y en acciones físicas sobre elementos de código visual botones, imágenes y mensajes.

Las operaciones que se realizan al acceder a la información almacenada en la base de datos y ficheros son rápidas e incrementales con efectos inmediatos.

4.2.2 Tratamiento de excepciones.

El diseño de la interfaz del sistema está dirigido a evitar errores. El sistema debe detectar los problemas en el proceso de autentificación por parte de algún usuario, manteniendo las restricciones de introducción de información errónea y mostrando en todo momento las opciones e información que debe, teniendo en cuenta el tipo de usuario.

4.3 – Conclusiones

En este capítulo se muestran los diagramas de clases web y el diseño lógico y físico de los datos. Se definieron además los principios de diseño que rigieron el desarrollo de la aplicación.

Conclusiones

Con el desarrollo de este trabajo se arribaron a las siguientes conclusiones:

- Como resultado de este trabajo se logró el diseño de un sistema para la solicitud y asignación de tiempos de máquina en los laboratorios docentes de la UCf.
- Se realizaron entrevistas a los técnicos de laboratorios, recopilándose detalles de las actividades que se desarrollan en los laboratorios de las facultades, definiéndose los procesos claves a automatizar.
- Se realizó un estudio de las principales tendencias, tecnologías y metodologías, así como los requisitos funcionales y no funcionales; lo cual fue punto de partida para lograr un buen diseño e implementación de la solución propuesta.
- Se desarrolló e implementó un sistema que automatizó la solicitud y asignación de los tiempos de máquina en los laboratorios docentes, el mismo permite un mejor control y gestión de los procesos que tienen lugar en dichos laboratorios.
- Se cumplió el objetivo general del trabajo lográndose la conclusión de la mayor parte de los flujos de trabajo especificados en la metodología RUP.

Recomendaciones

A pesar de que como resultado final de este trabajo se haya obtenido un producto que implementa un conjunto de prestaciones importantes para el control y gestión de los procesos que tienen lugar en los laboratorios docentes, se recomienda incorporar al software resultante las siguientes prestaciones:

1. Lograr la integración con el sistema de directorios activos (Active Directory) de manera que permita automatizar el proceso de permitir o denegar el uso de las computadoras en los laboratorios en función de las solicitudes de tiempos de máquina de los usuarios.
2. Que a partir del cumplimiento de la primera recomendación se puedan implementar las funciones de registro o log que permitan generar reportes que contengan la información referente al uso y aprovechamiento del sistema por parte de los usuarios finales.

Referencias Bibliográficas

[1] Guardado. Manuel La Informatización, una disciplina necesaria necesaria en la educación superior cubana. Tomado de: www.mes.edu.cu/PedagogiaUniversitaria/1998/2/Index.html, 22 de abril de 2006.

[2] Wallid. Omar. Implementación de un sistema de gestión para laboratorios docentes. Tomado de: http://oasis.dit.upm.es/~omar/pfc, 25 de abril de 2006.

[3] Jacobson, I.; "El Proceso Unificado de Desarrollo de software"/I. Jacobson, G. Booch, J. Rumbaugh,. _ _ [s,l]: Addison-Wesley. 2000. _ _ p 25.

[4] Jordán Enríquez, Odalys. "Plataforma Web para el Desarrollo de Revistas Virtuales"./ Odalys Jordán Enríquez. Trabajo de Diploma para optar por el título de Ingeniero Informático. Instituto Superior Politécnico José Antonio Echeverría. Tutor: Dr Sc. Julio Martínez Ciudad de la Habana, junio 2003. p45

[5] Manifiesto del Software Libre. Tomado de: http://manifiesto.cofradia.org/ , 8 de marzo de 2006.

[6]Estadística de uso de servidores. Tomado de: http://news.netcraft.com/archives/2005/06/index.html, 10 de abril de 2006.

[7] Software Linux: Programas y Aplicaciones de Linux. Tomado de http://linux.bankhacker.com/software/Apache/, 10 de junio de 2006.

[8] MySQL toma nuevo impulso y ya es una seria amenaza para Oracle, IBM y Microsoft. Tomado de: http://www.noticiasdot.com/publicaciones/2005/0405/1404/noticias/noticias , 15 de mayo de 2006.

[9]Servidores de Base de Datos. Tomado de: http://www.soportelinux.com.pe/menu/soluciones.php?solucion_id=2, 10 de mayo de 2006.

[10] Programación en php. Tomado de : http://www.desarrolloweb.com/manuales/12, 1ro de junio de 2006.

[11] Tomas, Alex. ¿Pór elegir PHP? Tomado de http://www.noticias.com , 1ro de junio de 2006.

[12] Jacobson, I.; "El Proceso Unificado de Desarrollo de software"/I. Jacobson, G. Booch, J. Rumbaugh,. _ _ [s,l]: Addison-Wesley. 2000. _ _ p 116.

[13] Idem p. 118

[14] Larman, Craig. *UML y Patrones: Introducción al análisis y diseño orientado a objetos*./ Craig Larman._ _ [s,l]: Editorial Prentice Hall, 1999. _ _ p41

[15] Jacobson, I.; "El Proceso Unificado de Desarrollo de software"/I. Jacobson, G. Booch, J. Rumbaugh,. _ _ [s,l]: Addison-Wesley. 2000. _ _ p 432.

[16] Idem. p 128

[17] Idem. p 129

[18] Larman, Craig. *UML y Patrones: Introducción al análisis y diseño orientado a objetos*./ Craig Larman._ _ [s,l]: Editorial Prentice Hall, 1999. _ _ p55

[19] Jacobson, I.; "El Proceso Unificado de Desarrollo de software"/I. Jacobson, G. Booch, J. Rumbaugh,. _ _ [s,l]: Addison-Wesley. 2000. _ _ p 257

Bibliografía

Brea, Orlando Fabián, *"Introducción a los Web Services en PHP"*. Tomado de : http://www.microsoft.com/spanish/msdn/comunidad/MTJ_3227.asp , Junio 2006.

Bruegge, Bernd, *"Ingeniería de software orientado a objetos"*./ Bernd Bruegge. _ _ [s.l]:[s.n.], 2002. 420p

Connolly, Thomas. Strachan, Anne. *Database Systems - A Practical Approach to Design, Implementation, and Management*./ Tomas Connolly, Carolyn Begg, Anne Strachan. _ _ [s,l] Addison-Wesley; [s,a] _ _370p

Díaz Antón Maria Gabriela, *"Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática"* 2004. / Maria Angélica Pérez Tomado de: http://www.academia-interactiva.com/ise.pdf, 1ro de junio de 2006

Enciclopedia Wikipedia. Wikimedia Foundation, Inc. Tomado de: http://es.wikipedia.org/wiki/Servicios_Web , 4 de abril de 2006

Estadística de uso de servidores. Tomado de: http://news.netcraft.com/archives/2005/06/index.html , 10 de abril de 2006

Ferrá, Grau Xavier *"Desarollo orientado a objetos con UML"* Universidad Politécnica de Madrid, 2004. Tomado de: http:// www.clikear.com/manuales/uml/introduccion.asp, 10 de abril de 2006

Guardado. Manuel La Informatización, una disciplina necesaria necesaria en la educación superior cubana. Tomado de: www.mes.edu.cu/PedagogiaUniversitaria/1998/2/Index.html, 22 de abril de 2006.

Jacobson, I.; "El Proceso Unificado de Desarrollo de software"/I. Jacobson, G. Booch, J. Rumbaugh,. _ _ [s,l]: Addison-Wesley. 2000. _ _ 435p.

Conallen, Jim. *Building Web Applications with UML*, Object Technology/ Jim Conallen [s,l]Addison Wesley Longman, 1999. _ _139p

Larman, Craig. *UML y Patrones: Introducción al análisis y diseño orientado a objetos/* Craig Larman. _ _ Editorial Prentice Hall, 1999. _ _503 p

Letelier Torres Patricio "*Desarrollo de Software Orientado a Objeto usando UML"* Mayo 2004, Departamento Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España. Tomado de: http://www.creangel.com/uml/intro.php , Abril, 2006.

Manifiesto del Software Libre. Tomado de: http://manifiesto.cofradia.org/ (8/03/2006)

Montero Ayala, Ramón *"XML, iniciación y referencia"/*Ramón Montero Ayala. _ _ Editorial McGraw-Hill, 2001. _ _327 p

MySQL toma nuevo impulso y ya es una seria amenaza para Oracle, IBM y Microsoft. Tomado de:

http://www.noticiasdot.com/publicaciones/2005/0405/1404/noticias/noticias_14040, 15 de abril de 2006

Pressman, Roger. Ingeniería de Software, un enfoque práctico./ Roger Pressman. _ _E.U: Mc Graw-Hill, 1998_ _ 450p.

Programación en php. Tomado de : http://www.desarrolloweb.com/manuales/12, 1ro de junio de 2006

Rodríguez, Luis [Manual] **"***Conoce a PEAR: El Repositorio de Aplicaciones y Extensiones de PHP",* Febrero, 2005. Tomado de :http://www.programacion.com/blogs/60_php_land/archive/335_conoce_a_pear_el_repositorio_de_aplicaciones_y_extensiones_de_php.html, 10 de abril de 2006

Rumbaugh, J.; Jacobson, I. y Booch, G.; "El Lenguaje Unificado de Modelado. Manual de referencia./ I. Jacobson, G. Booch, J. Rumbaugh,. _ _ [s,l]. Addison-Wesley. 2000. _ _ 326p

Servidores de Base de Datos. Tomado de

http://www.soportelinux.com.pe/menu/soluciones.php?solucion_id=2 , 10 de mayo de 2006

Software Linux: Programas y Aplicaciones de Linux. Tomado de http://linux.bankhacker.com/software/Apache/ , 10 de mayo de 2006

Tedeschi, Nicolás. *"Web Services, un ejemplo práctico"* / Nicolás Tedeschi. _ _ .

Tomado de:

http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3227.asp, 10 de abril de 2006

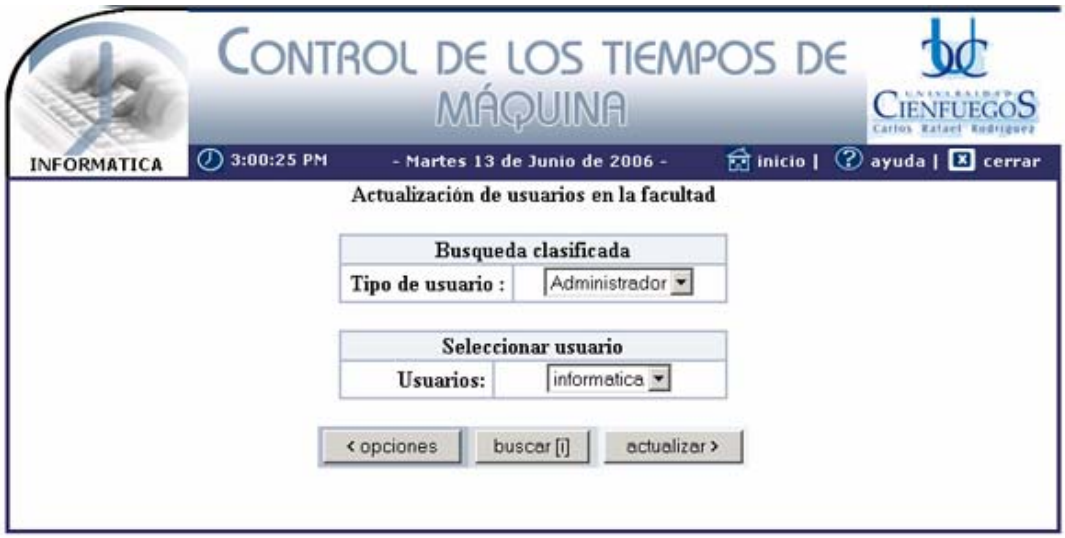
Tomas, Alex. ¿Pór elegir PHP? Tomado de http://www.noticias.com , 2 de junio de 2006

Wallid. Omar. *Implementación de un sistema de gestión para laboratorios docentes*. Tomado de http://oasis.dit.upm.es/~omar/pfc, 25 de abril de 2006.

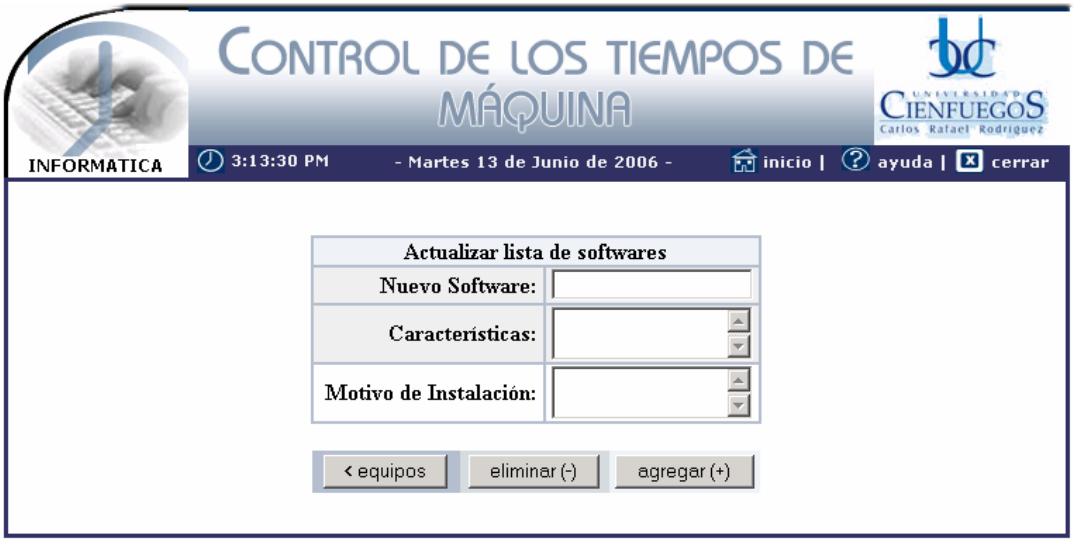
Anexo 1. Prototipos.

Anexo 1.1 Caso de Uso: Gestionar PC

Anexo 1.2 Caso de Uso: Gestionar Usuario



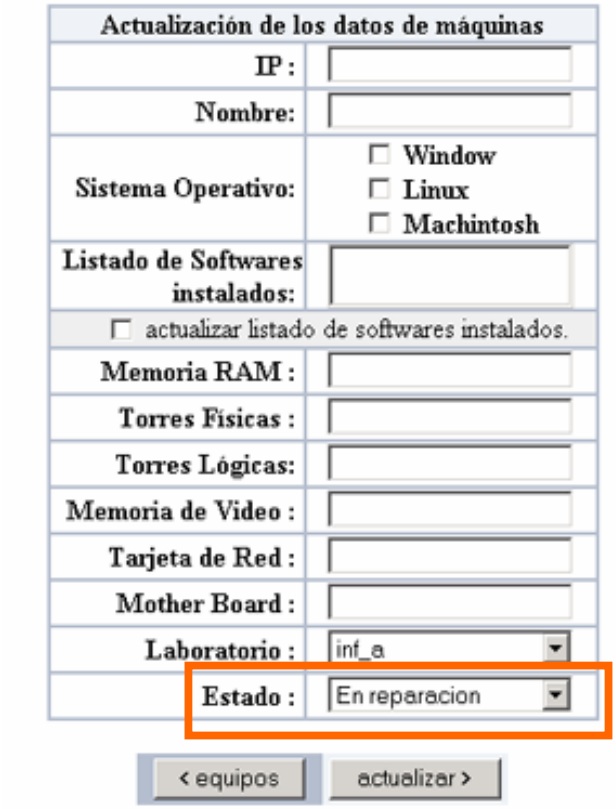
Anexo 1.3 Caso de Uso: Gestionar software



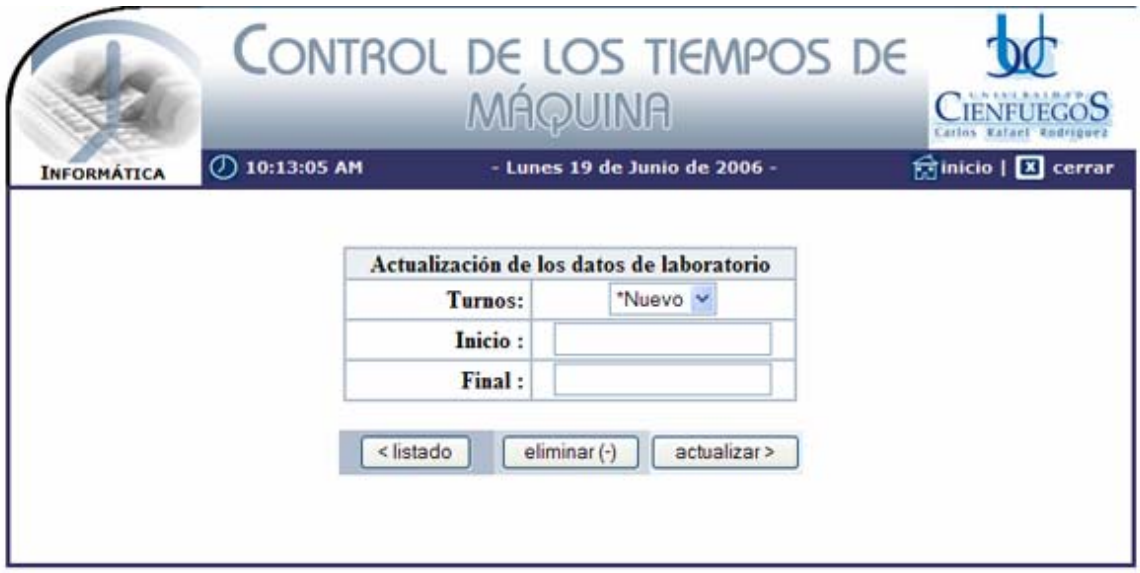
Anexo 1.4 Caso de Uso: Gestionar laboratorio

Anexo 1.5 Caso de Uso: Gestionar actividad docente

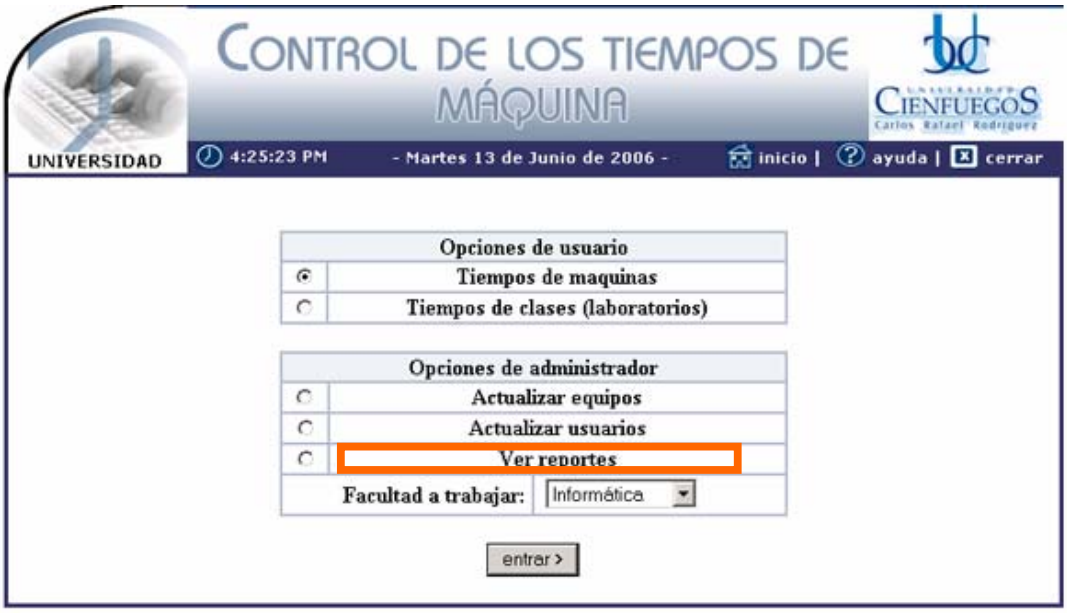
Anexo 1.6 Caso de Uso: Deshabilitar PC con dificultades



Anexo 1.7 Caso de Uso: Gestionar turno

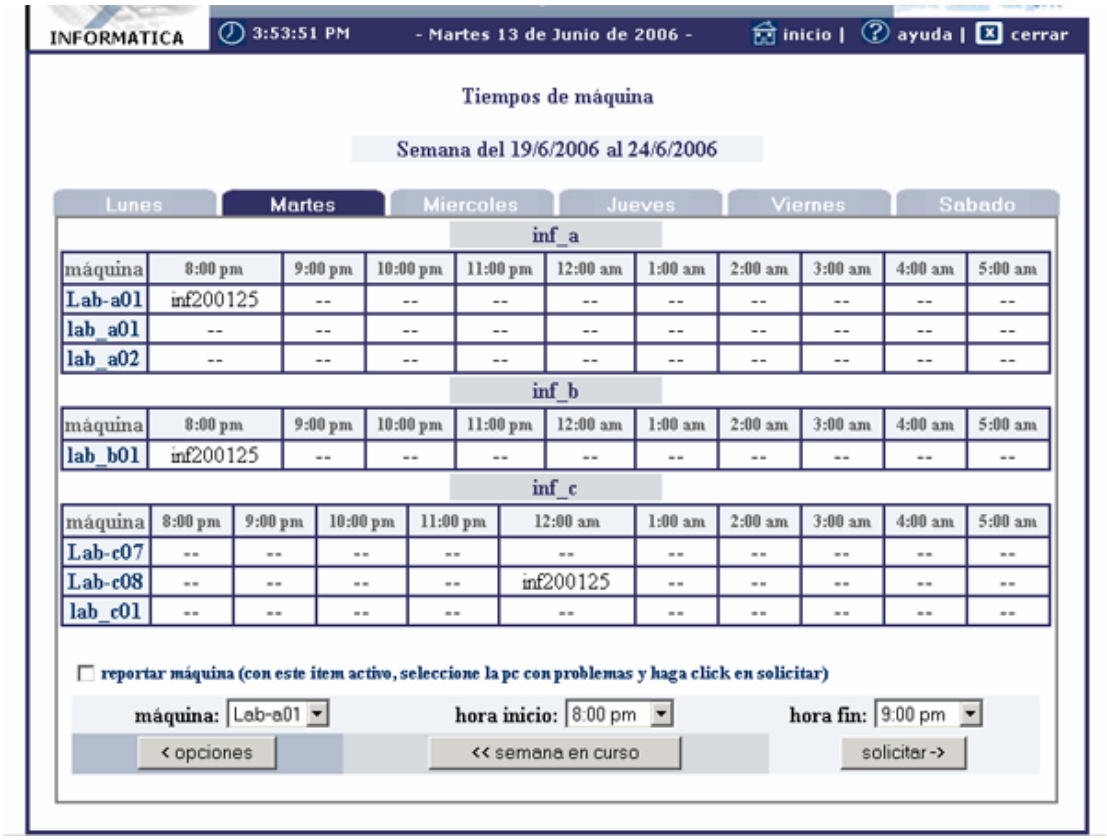


Anexo 1.8 Caso de Uso: Visualizar reporte de estado de las PC



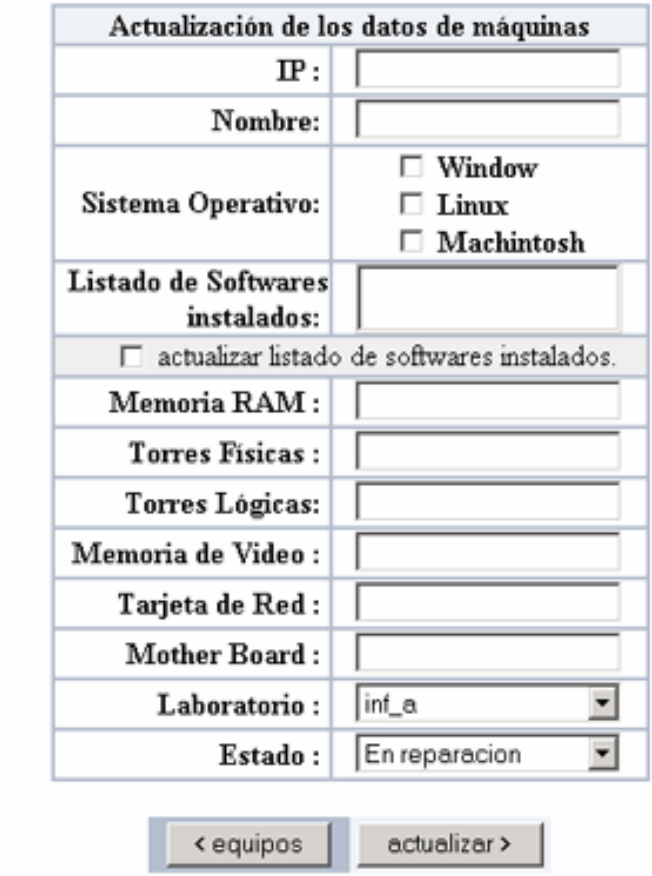
Anexo 1.9 Caso de Uso: Visualizar PC reportadas con problemas

Anexo 1. 10 Caso de Uso: Mostrar tiempos de máquina disponibles

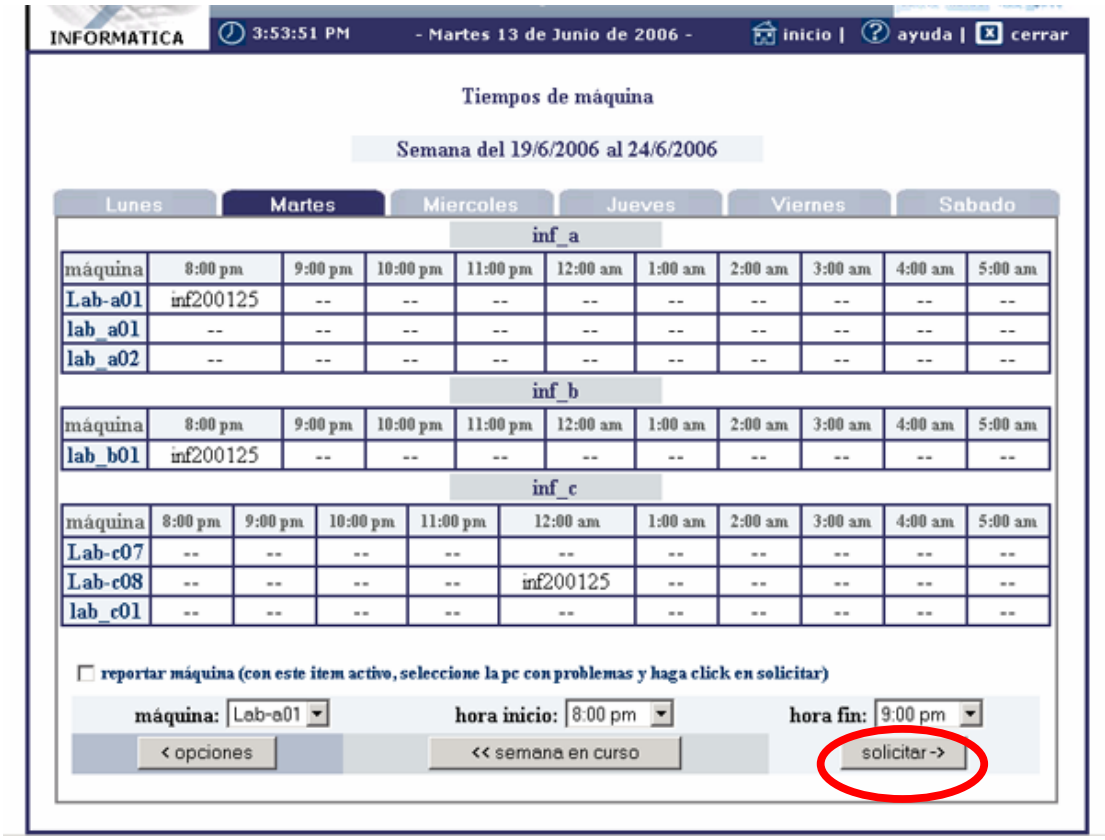


Anexo 1.11 Caso de Uso: Mostrar las actividades docentes planificadas

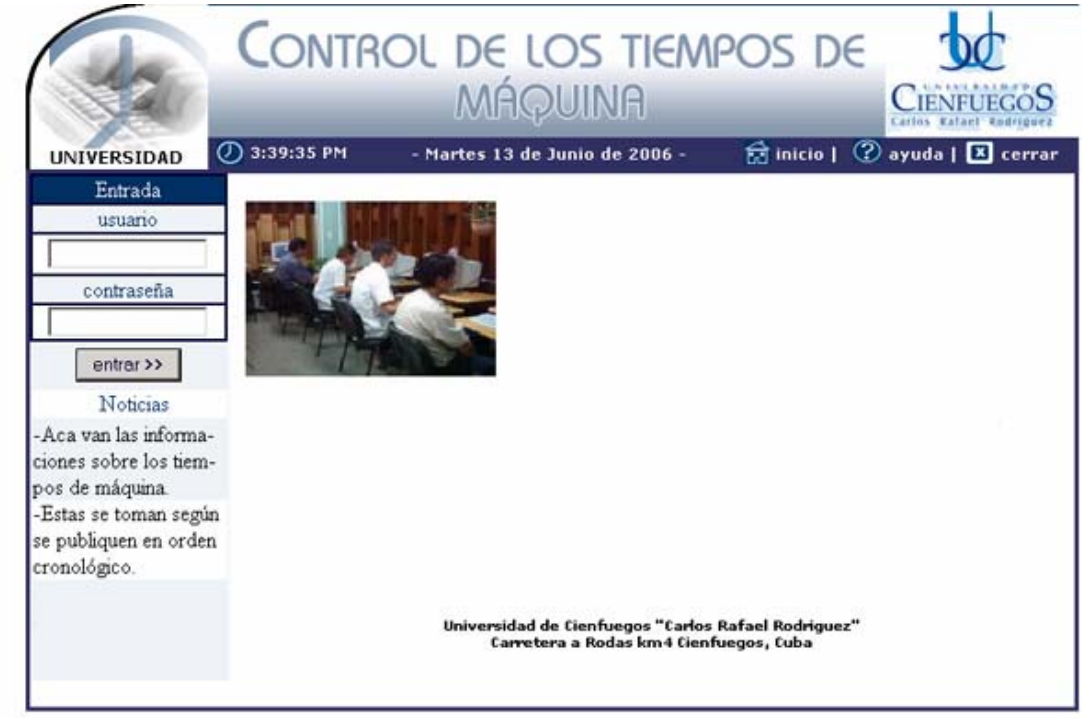
Anexo 1.12 Caso de Uso: Visualizar detalles de una PC



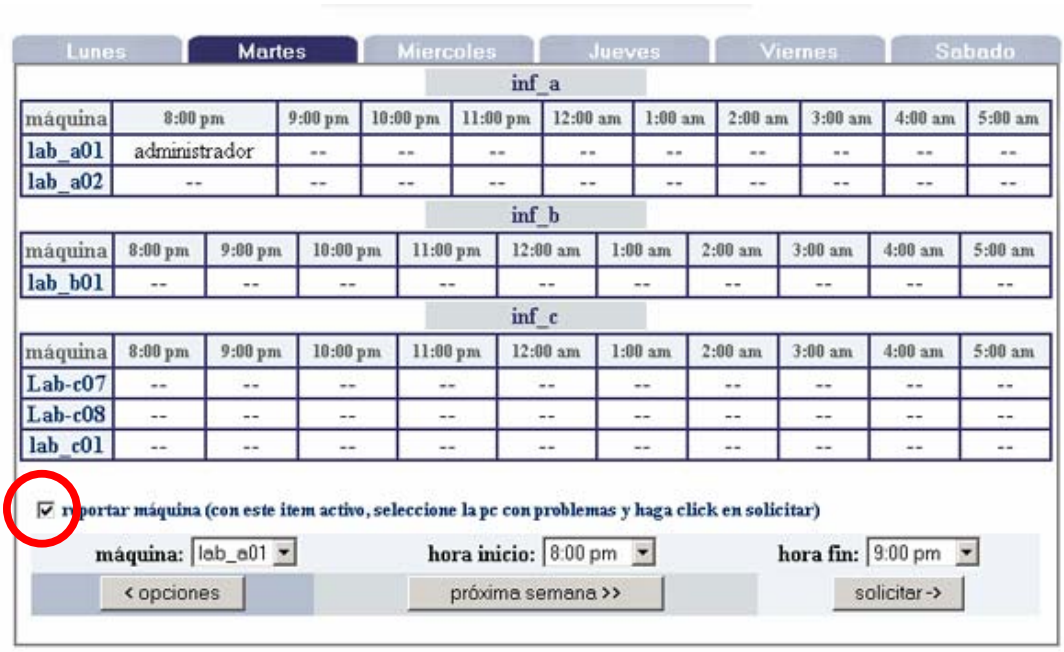
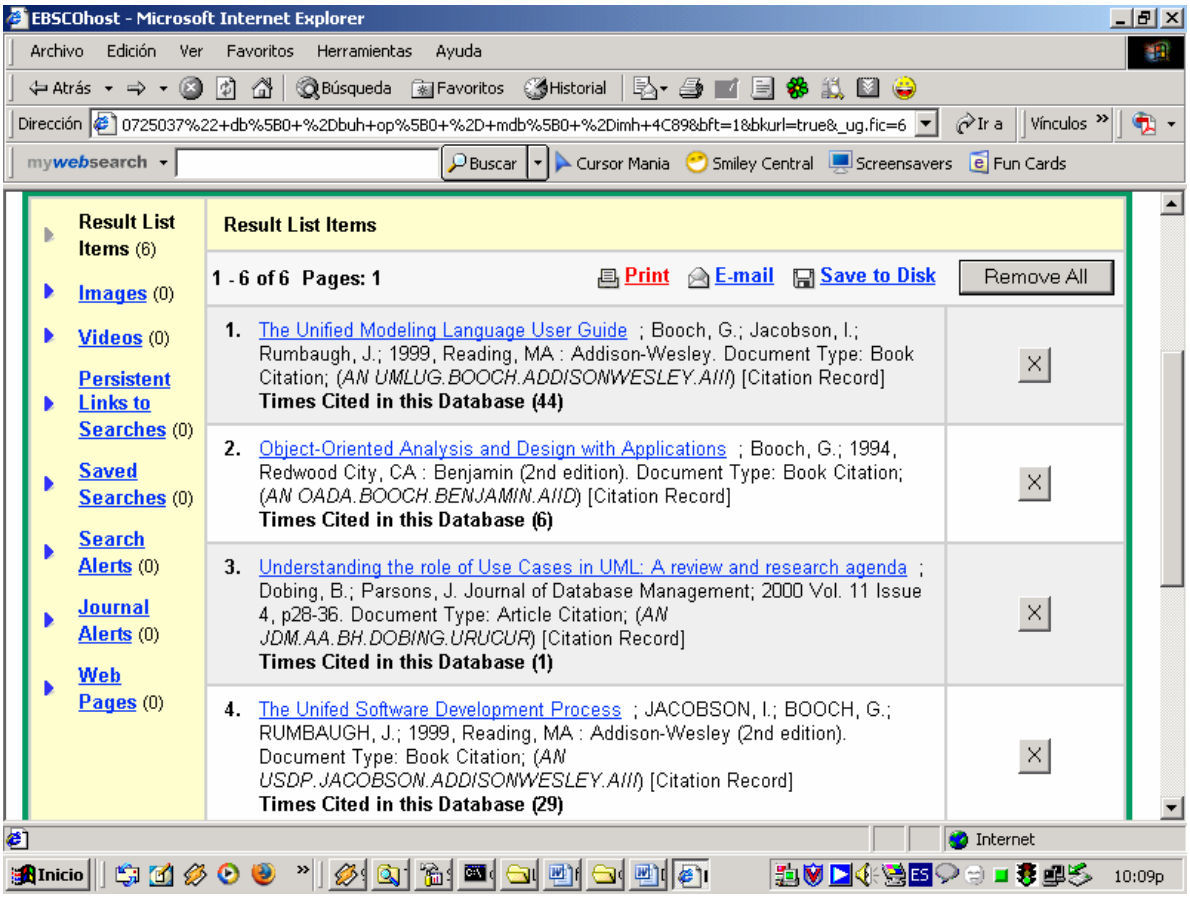
Anexo 1.13 Caso de Uso: Registrar tiempo de máquina



Anexo 1.14 Caso de Uso: Autenticarse



Anexo 1.15 Caso de Uso: Reportar PC con problemas



Anexo 2. Bibliografía tomada de EBSCO.