

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍAS

Título: Sistema de asignación de recursos de cómputo

Autor: JUAN SEBASTIAN MANTILLA QUINTERO

Jurado

Jurado

Director

Cartagena, Enero 12 de 2017

JUAN SEBASTIAN MANTILLA QUINTERO

Directores:

Juan Carlos Martínez Santos
Edwin Alexander Puertas del Castillo

**Universidad Tecnológica de Bolívar
Facultad de Ingenierías
Programa de Ingeniería de Sistemas
Cartagena**

Enero 12 de 2017

Sistema de asignación de recursos de cómputo

JUAN SEBASTIAN MANTILLA QUINTERO

Trabajo de grado para optar al título de

Ingeniero de Sistemas

Directores:

Juan Carlos Martínez Santos
Edwin Alexander Puertas del Castillo

**Universidad Tecnológica de Bolívar
Facultad de Ingenierías
Cartagena**

Enero 12 de 2017

Resumen

En cualquier establecimiento, empresa o plantel educativo en donde se tengan disponibles equipos de cómputo para préstamos, normalmente, hacen este préstamo con muy poco control por lo que se genera un descontento al momento de:

- Obtener un registro de los préstamos de los equipos de cómputo
- Recibir un equipo: es posible que se le asigne al usuario un equipo que no tiene los programas que necesita.
- Obtener una gestión de las reservas: es posible que muchos usuarios no puedan acceder a un equipo de cómputo por falta de planificación en la asignación del recurso.
- Querer reservar un equipo de cómputo de forma ágil.

Para la Universidad Tecnológica de Bolívar es indispensable crear una herramienta que posibilite una gestión de todas las reservas de equipos y salones de cómputo teniendo en cuenta las necesidades de los usuarios tanto día y hora de la reserva como hardware y software que vaya a necesitar de una manera ágil, óptima y segura y así mismo, permitir generar un inventario actualizado de todo el hardware y software de los equipos de cómputo ligados a la herramienta. Para crear esta herramienta es necesario obtener una base de datos del inventario de hardware y software. Teniendo este inventario, se puede discriminar los equipos de cómputo que no cumplan con las

necesidades del usuario.

Agradecimientos

Agradecemos a Jairo Enrique Serrano Castañeda por llevar un seguimiento en la implementación de la aplicación web.

Contents

1	Introducción	13
1.1	Planteamiento del problema	14
1.2	Objetivos	15
1.2.1	Objetivo general	15
1.2.2	Objetivos específicos	15
1.3	Justificación	16
2	Marco teórico	18
2.0.1	Obtención del inventario de hardware y software manualmente	18
	Desventajas	19
	Ventajas	19
2.0.2	Obtención del inventario de hardware y software automáticamente	19
	Desventajas	20
	Ventajas	20
2.1	Estado del arte	20
	Formato de administración de información (MIF)	22

Capa de servicio	22
Interface de componente (CI)	22
Interface de administración (MI)	22
2.2 Inventario de hardware y software	23
2.3 Elección de una herramienta de inventario de hardware y software . .	23
2.3.1 Solución a problemas	25
3 Propuesta	27
3.1 Requerimientos	27
3.2 Diseño general	30
3.3 Diagramas	30
3.3.1 Diagrama de casos de uso	30
3.3.2 Diagrama de clases	30
Diagrama de clases general	30
Diagrama de clases de equipos de cómputo	30
3.3.3 Diagrama de entidad-relación	30
3.4 Sistema integrado para administración de recursos de cómputo	30
3.4.1 Administrador	36
3.4.2 Organizador	36
3.4.3 Usuario normal	36
3.5 Implementación	36
3.5.1 Hardware	36

3.5.2	Algoritmos	37
	Asignación de reserva hecha por un usuario	37
	Asignación de reserva hecha por un organizador	37
	Cancelación de reserva	38
4	Resultados	39
4.1	Resultados obtenidos	39
4.2	Simulación	39
5	Conclusiones y Trabajo a futuro	49
5.1	Conclusiones	49
5.2	Trabajo a futuro	50

Lista de Figuras

1.1	Recursos de cómputo prestados vs negados vs disponibilidad total. . .	14
2.1	Comparación de herramientas de inventario.	25
3.1	Modelo de SAIRCO.	28
3.2	BPMN de servicios de SAIRCO.	29
3.3	Diagrama de componentes.	31
3.4	Diagrama de casos de uso.	32
3.5	Diagrama de clases general.	33
3.6	Diagrama de clases de equipos.	34
3.7	Diagrama de entidad-relación.	35
4.1	Login de la aplicación, dependiendo del rol redirecciona a diferentes interfaces	40
4.2	Panel del administrador	40
4.3	Estadísticas de préstamos de cada software	40
4.4	Equipos en la base de datos de SAIRCO	41
4.5	Salones en la base de datos de SAIRCO	41

4.6	Vista de actualización de un equipo	42
4.7	Formulario para agregar un nuevo administrador	42
4.8	Panel del organizador	43
4.9	Equipos con reservas disponibles en el día actual	43
4.10	Formulario para asignar una reserva de último minuto	44
4.11	Formulario para crear un nuevo organizador	44
4.12	Panel del usuario normal	45
4.13	Formulario para reservar un equipo de cómputo	45
4.14	Vista de la gestión de reservas por parte del usuario normal	46
4.15	Formulario para cancelar una reserva	46
4.16	Vista de la reserva cancelada	47
4.17	Disponibilidad vs horas libres por hora	48

Lista de Tablas

2.1	Comparación de herramienta de inventario	24
-----	--	----

Capítulo 1.

Introducción

Actualmente, para poder usar un computador de las aulas de informática de la Universidad Tecnológica de Bolívar, hay que ir personalmente a las mismas y preguntar si hay alguna sala libre que tenga computadores disponibles. Este sistema no es el más eficiente ya que si el estudiante requiere un computador con un software específico y los computadores de la sala disponible no cuentan con ese software, no les serviría en absoluto usarlo. También es posible que cuando la persona que necesita el equipo vaya hasta las instalaciones de las aulas de cómputo y no haya ningún equipo de cómputo disponible. Para obtener un inventario actualizado del software instalado en todos estos computadores, se debe hacer de forma manual. Esta tarea requiere de una cantidad exagerada de tiempo y puede haber algún error humano en el proceso. Además, es posible que se estén pagando licencias por software que los estudiantes están utilizando muy poco o que se esté pagando pocas licencias de software que se necesitan demasiado. Incluso es probable que varios estudiantes utilicen un computador demasiado potente para tareas que no necesitan tanta potencia y, si hay otro estudiante que sí necesita un computador potente, no puede usarlo porque está ocupado.

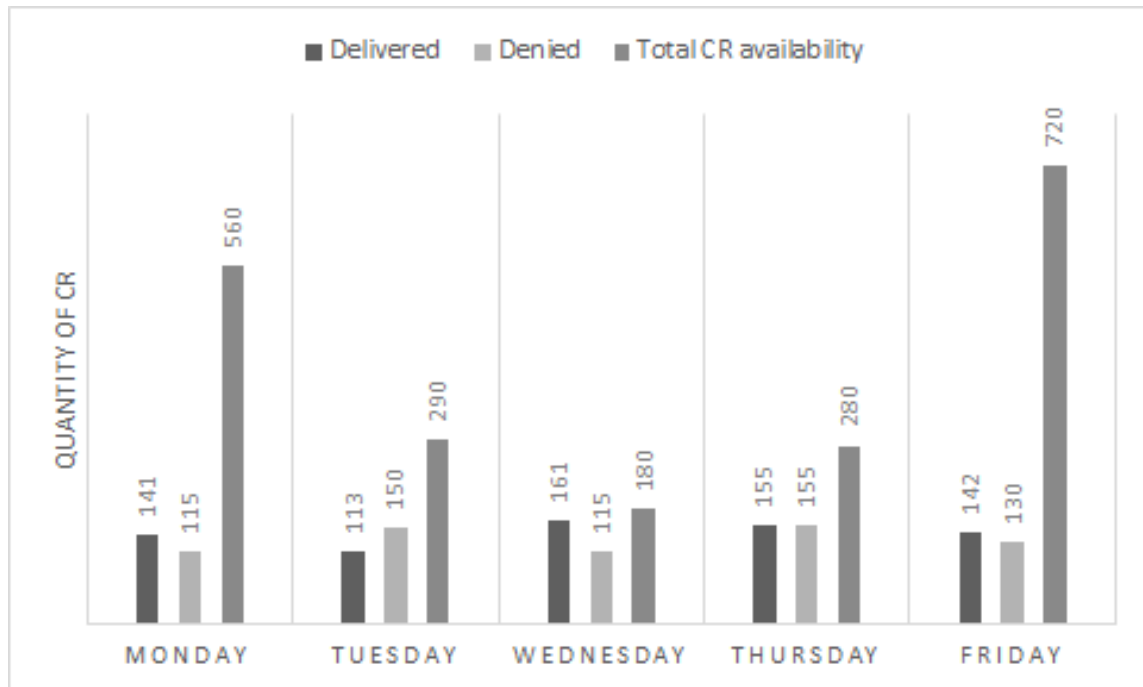


Figure 1.1: Recursos de cómputo prestados vs negados vs disponibilidad total.

1.1 Planteamiento del problema

Presentamos un caso de estudio de los préstamos de equipos de cómputo realizados en la última semana del semestre 2016-01 de la Universidad Tecnológica de Bolívar (UTB) que demuestra porqué es necesario un sistema de asignación de recursos de cómputo diferente al existente actualmente. Escogimos esta semana ya que es la que representa la mayor demanda de equipos de cómputo durante todo el semestre 2016-01. En la Fig.1.1 podemos ver la suma de todos los equipos prestados, negados y la disponibilidad total de los equipos de cómputo diariamente. Aunque el número de equipos de cómputo prestados es mayor que los que se niegan (excepto el jueves) es claro que el total de equipos de cómputo disponibles de la UTB es mucho mayor

que incluso la suma de los equipos de cómputo prestados y negados. La UTB tiene 140 equipos de cómputo que se usan entre las 7:00 y 20:00, lo que proporciona 1820 recursos de cómputo para clases normales y préstamos. Así, se evidencia que la UTB puede satisfacer completamente la demanda de préstamos de los estudiantes. Esto sucede porque la UTB carece de una forma eficiente de asignar todos los equipos de cómputo que tiene disponible para préstamos.

1.2 Objetivos

1.2.1 Objetivo general

Obtener una aplicación web que permita gestionar la asignación de los recursos de cómputo de la sala de informática A1-402 a los estudiantes de la UTB dependiendo de las necesidades de hardware y software de los mismos.

1.2.2 Objetivos específicos

- Obtener un estado del arte de los diferentes sistemas de gestión de cómputo y de las herramientas más eficientes para hacer inventario de software y hardware.
- Identificar o crear una herramienta que permita extraer la información necesaria de hardware y software de los equipos de cómputo.
- Desarrollar una interface que arroje un “best match” usando como base las necesidades del estudiante con relación a la base de datos generada de hardware y software de los computadores y enlazarlo a una aplicación o script entendible

para la página web.

1.3 Justificación

Una empresa, plantel educativo o cualquier entidad que ofrezca el servicio de préstamos de cualquier tipo (incluso un puesto de alquiler de bicicletas en la calle) que no tenga una herramienta que permita gestionar las reservas de manera óptima, rápida y segura y mantener un inventario de los artículos disponibles para préstamo, está destinada al fracaso. Sin una herramienta que haga esto, se genera un desorden en el recaudo monetario, el registro de las reservas; no se sabe a quien se le asignó un artículo, no hay una planificación de los mantenimientos, no hay un inventario de los artículos que están disponibles, cuáles no y cuándo estarán disponibles.

Por otra parte, estamos en una época en donde casi todo se maneja por medio de internet; si alguien quiere conseguir un sitio para alquilar computadores, celulares por días, bicicletas o cualquier otro producto, lo busca por internet. Por lo tanto, es evidente la necesidad de tener un portal web que posibilite las reservas de artículos y que los usuarios puedan encontrarlo de manera fácil y rápida.

En un ámbito más relacionado con lo que queremos presentar en este proyecto (reservas de equipos de cómputo) si a una empresa o plantel educativo que no cuenta con un software que gestione los préstamos se le daña un equipo de cómputo, no hay forma de saber quién fue la persona que lo dañó. Además no cuenta con un inventario de hardware y software crucial para saber detalladamente qué tipo de

productos tiene y, por lo tanto, no sabría qué puede ofrecer, no podría agendar mantenimientos y que no se podría impedir la reserva de los equipos de cómputo durante estos mantenimientos, así mismo es posible que muchos usuarios no puedan acceder a un equipo de cómputo por falta de planificación en la asignación de los recursos o que accedan a un equipo de cómputo que no tiene el software que necesita para realizar sus tareas lo que sería completamente obsoleta la reserva para el.

Además, como se mencionó anteriormente (Fig. 1.1), la UTB cuenta con los recursos necesarios para cubrir totalmente la demanda de equipos de cómputo de sus estudiantes. Por estos motivos es indispensable tener una herramienta web de gestión de asignación de recursos de cómputo ágil, rápida y segura; es mucho más fácil y cómodo para cualquier persona reservar un equipo o salón de cómputo desde su ordenador o dispositivo móvil en su casa o cualquier sitio que se encuentre que tener que ir hasta el sitio donde están físicamente los equipos y preguntar si hay algún equipo disponible contando con el riesgo de que no haya ninguno disponible o que hayan pero no con el software que el necesita.

Capítulo 2.

Marco teórico

Para poder habilitar al usuario elegir qué software y/o hardware quiere utilizar, es necesario conocer qué hardware y software tiene cada equipo, por lo que nos permitiremos idear o planear dos formas diferentes de obtener un inventario de hardware y software en una empresa y exponer las ventajas y desventajas de cada una.

2.0.1 Obtención del inventario de hardware y software manualmente

- Un empleado de IT guarda todo el hardware y software de cada equipo de cómputo.
- Almacena este inventario en un servidor.
- Cada vez que detecte un cambio de hardware o software al momento de hacer el inventario, guardarlo en el servidor.

Desventajas

:

- Demasiado tiempo para su ejecución.
- Posibles errores humanos.

Ventajas

:

- No requiere tiempo en investigación.

2.0.2 Obtención del inventario de hardware y software automáticamente

- Crear o encontrar un agente que esté programado para extraer la información necesaria de cada computador cada vez que sea necesario.
- El agente se instala en cada equipo de cómputo al que se le quiera extraer la información relevante para el inventario de hardware y software.
- El agente envía la información del inventario a un servidor.
- El servidor almacena de forma organizada la información.

Desventajas

:

- Requiere tiempo en investigación.

Ventajas

:

- Automatización de la extracción del inventario.
- Rápida ejecución.
- Programable (puede actualizar el inventario cuantas veces sea necesario de forma automática).

Es evidente que las ventajas de obtener el inventario de forma automática superan a las de obtenerlo manualmente. Ahora, teniendo esta serie de actividades necesarias para obtener un inventario de un conjunto de computadores automáticamente, entraremos a exponer qué herramientas o proyectos lo han hecho o, en su defecto, las más parecidas en la sección "estado del arte".

2.1 Estado del arte

Hay varias herramientas como OCS inventory NG [8] u opsi [10], que permiten obtener el inventario de los computadores y almacenarlo usando las siguientes actividades:

- Instalación de un servidor con una base de datos.

- Instalación de un agente en cada uno de los equipos a ser inventariados.
- El agente recopila los datos de hardware y/o software necesarios utilizando el framework Desktop Management Interface (DMI) que viene en la mayoría de los computadores fabricados entre los últimos 18 y 20 [13] años y crea un reporte o inventario para luego mandarlo al servidor.
- El servidor lo recibe y almacena.
- Una aplicación web muestra organizadamente toda la información del inventario.

El DMI es una especificación de Desktop Management Task Force (DMTF [7]) que establece un framework estándar para administrar computadores en red [6]. Como se evidencia en las actividades mencionadas, se obtiene el inventario y se almacena en una base de datos.

DMI provee un espacio común para soporte técnico, administradores de tecnologías de información y usuarios comunes que permite acceder a información de cualquier aspecto de un computador - incluyendo el tipo de procesador, fecha de instalación, impresoras conectadas y cualquier otro periférico, fuentes de energía e historia de mantenimiento. Provee un formato compartido para describir productos.

DMI no está atado a un hardware específico, sistema operativo o protocolos de administración. Es fácil de adaptar a protocolos de administración existentes como "Simple Network Management Protocol" (SNMP), y se puede usar en computadores

que no están conectados a la red. DMI está compuesto de cuatro componentes:

Formato de administración de información (MIF)

Es un archivo de texto que contiene la información del hardware y software en un computador. Los fabricantes pueden crear su propio MIF específico a un componente.

Capa de servicio

Es un add-on al sistema operativo que conecta el MFI y la interface de componente y permite software de administración para acceder a los archivos MIF. La capa de servicio también provee una interface común llamada "local agent", que se usa para administrar componentes individuales.

Interface de componente (CI)

Es una interfaz de programa de la aplicación (API) que envía información al archivo MIF via la capa de servicio. Los comandos incluyen *GET*, *SET*, y *EVENT*.

Interface de administración (MI)

Es la interface de administración del software a la capa de servicio. Sus comandos son *GET*, *SET*, y *LIST*. Los drives de CI, MI y la capa de servicio están disponibles en internet. el administradon de clientes de Intel LANDesk (LDC) [9] está basado en DMI.

Además de las herramientas mencionadas anteriormente, hay proyectos conocidos como "Asset Management" como la de cherwell software [2], Aranda Software [1] o spiceworks [5] que usan este inventario y le agregan un "dinamismo" que consiste,

entre otras cosas, en identificar cuáles aplicaciones no se han usado, cuáles sí y con qué frecuencia. Actualmente no existe una herramienta que permita obtener el inventario de hardware y software y usarlo para la reserva de equipos de cómputo de forma organizada y centralizada.

2.2 Inventario de hardware y software

Para poder definir qué equipo se le va a asignar a cada reserva teniendo en cuenta el hardware y software de la misma, es necesario obtener de alguna manera el hardware y software de cada equipo disponible para las reservas. Para esto, existen varias formas de obtener un inventario de hardware y software pero nos centraremos en 3 particulares: 2 que son open source y una de forma manual.

2.3 Elección de una herramienta de inventario de hardware y software

Para definir cuál herramienta de inventario de hardware y software usar, comparamos tres herramientas: OCS Inventory NG, OPSI, y como la recolección del inventario se puede hacer manualmente, también se consideró.

Para probar cada una de las funcionalidades, se instaló cada herramienta en un servidor local (con CentOS 6, Intel core (Haswell) 3.59 GHz (2 procesadores), y 2GB de memoria RAM), y su agente en dos computadores diferentes para poder obtener una comparación. Los resultados se muestran en la tabla 2.1. Esta es la forma en la que se calificó cada categoría:

Table 2.1: Comparación de herramienta de inventario

	OCS Inventory NG	OPSI	Manual
Inventario de hardware	1	1	1
Inventario de software	1	1	1
Multi-plataforma (agente)	1	1	0
Nivel de dificultad de obtención del inventario	1	0	0
Base de datos	1	1	0
Automático	1	1	0

- *Inventario de hardware.* Capacidad de generar un inventario de hardware. 1 si lo provee, 0 si no.
- *Inventario de software.* Capacidad de generar un inventario de software. 1 si lo provee, 0 si no.
- *Multi-plataforma.* Disponibilidad de obtener el inventario de Windows, Mac y linux. si lo provee, 0 si no.
- *Nivel de dificultad de obtención del inventario.* Si es de fácil instalación el servidor y el agente 1, 0 si no.
- *Base de datos.* Si usa un motor de base de datos 1 , 0 si no.
- *Automático.* Si se puede obtener el inventario de forma automática 1, 0 si no.

Con esta comparación, se obtuvo la gráfica 2.1. Se puede evidenciar que la mejor opción como herramienta para obtener un inventario de hardware y software es OCS Inventory NG ya que provee todo lo que se necesita para este proyecto.

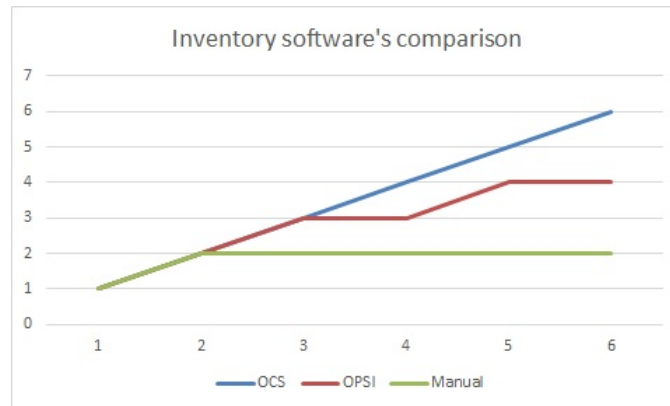


Figure 2.1: Comparación de herramientas de inventario.

2.3.1 Solución a problemas

Como se mencionó anteriormente, OCS Inventory NG es la herramienta que se escogió para recolectar el inventario de hardware y software pero hubo varios problemas con que se tuvieron que solucionar. Primero, se tuvo que instalar X Apache, MySQL, PHP, Perl server (XAMPP) para poder usar el software de OCS Inventory NG como servidor [12]. Esto se hizo en un computador con CentOS 6, Intel core (Haswell) 3.59 GHz (dos procesadores), y 2GB de memoria RAM.

Del inventario obtenido por OCS Inventory NG, se obtuvo todo el hardware y software. Pero, aún así debido a que es una herramienta open-source, cualquiera puede descargar e instalar su agente e inyectar el inventario de cualquier equipo a la base de datos del equipo que se está usando como servidor. Esto es un inconveniente crítico ya que se podría llenar la base de datos con datos completamente irrelevantes para el sistema por lo que se tuvo que implementar un componente en nuestro sistema que permite inyectar inventario de sólo los equipos autorizados.

Ahora se tiene el inventario de sólo los equipos autorizados, pero el inventario de software que provee OCS Inventory NG guarda registros de software que el usuario no reservará cuando esté eligiendo el software que va a usar. Software como drivers de video, de sonido, actualizaciones de windows, etc. Cada equipo con windows genera más de 100 registros de este tipo de software. Se quiere que el usuario seleccione el software que va a usar en el equipo, se decidió mostrar una lista de software seleccionado que se sabe que van a usar (esto depende del establecimiento en donde se vaya tener SAIRCO funcionando) y actualizar esta lista de software junto con los usuarios periódicamente

Con estos problemas resueltos, se pudo empezar el proceso de implementación de la aplicación.

Capítulo 3. Propuesta

3.1 Requerimientos

Se necesita un software donde el administrador pueda administrar y obtener información de los recursos de cómputo como hardware y software incluyendo equipos que tengan los sistemas operativos Windows, Mac o linux.

Se necesita un software donde un usuario normal pueda reservar un equipo de cómputo dependiendo del hardware y software que vaya a utilizar y cancelar la misma.

Como se muestra en la Fig. 3.1, la forma apropiada de hacer esto es instalar el agente de OCS en cada computador que esté disponible para préstamos, conectarlos a través de un servidor y desarrollar un software que use el inventario que generan los agentes que permita al usuario reservar un equipo de cómputo indicando qué hardware y software va a utilizar.

Para el diseño de este software, se decidió usar dos metodologías de desarrollo de software: el modelo cascada [16] para la documentación, y SCRUM [17] para el desarrollo del software. Las ventajas de usar estos dos métodos son que se pudo determinar detalladamente todos los requerimientos funcionales, actores y objetos

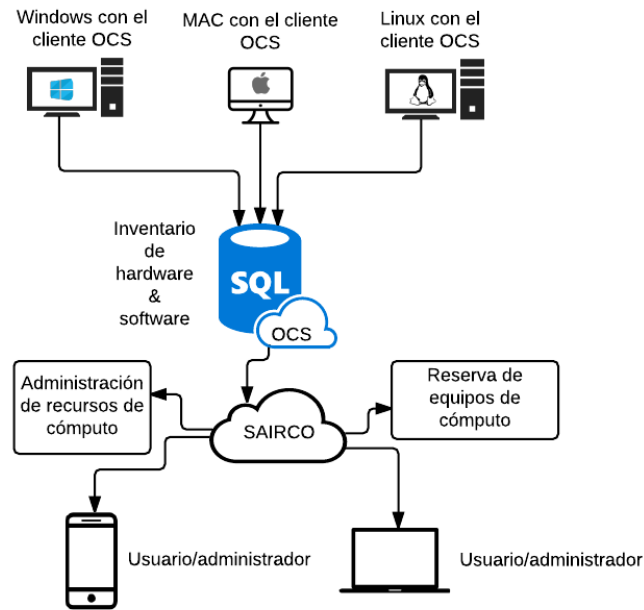


Figure 3.1: Modelo de SAIRCO.

fácilmente con reuniones semanales y asignando tareas a cada rol involucrado.

Además, se decidió usar los patrones de modelo de procesos incremental para proyectos de campo rápido (ICMEGP [14]). Las ventajas de usar estos patrones son que se pudo determinar en una forma detallada todos los requerimientos funcionales, actores y objetos de una manera fácil con reuniones semanales y asignación de tareas a cada rol involucrado en el proceso. Además, ayudó al equipo de desarrollo a producir un proceso de decisión viable tempranamente en el proceso de ciclo de vida con un tiempo rápido para el mercado y alta satisfacción del cliente.

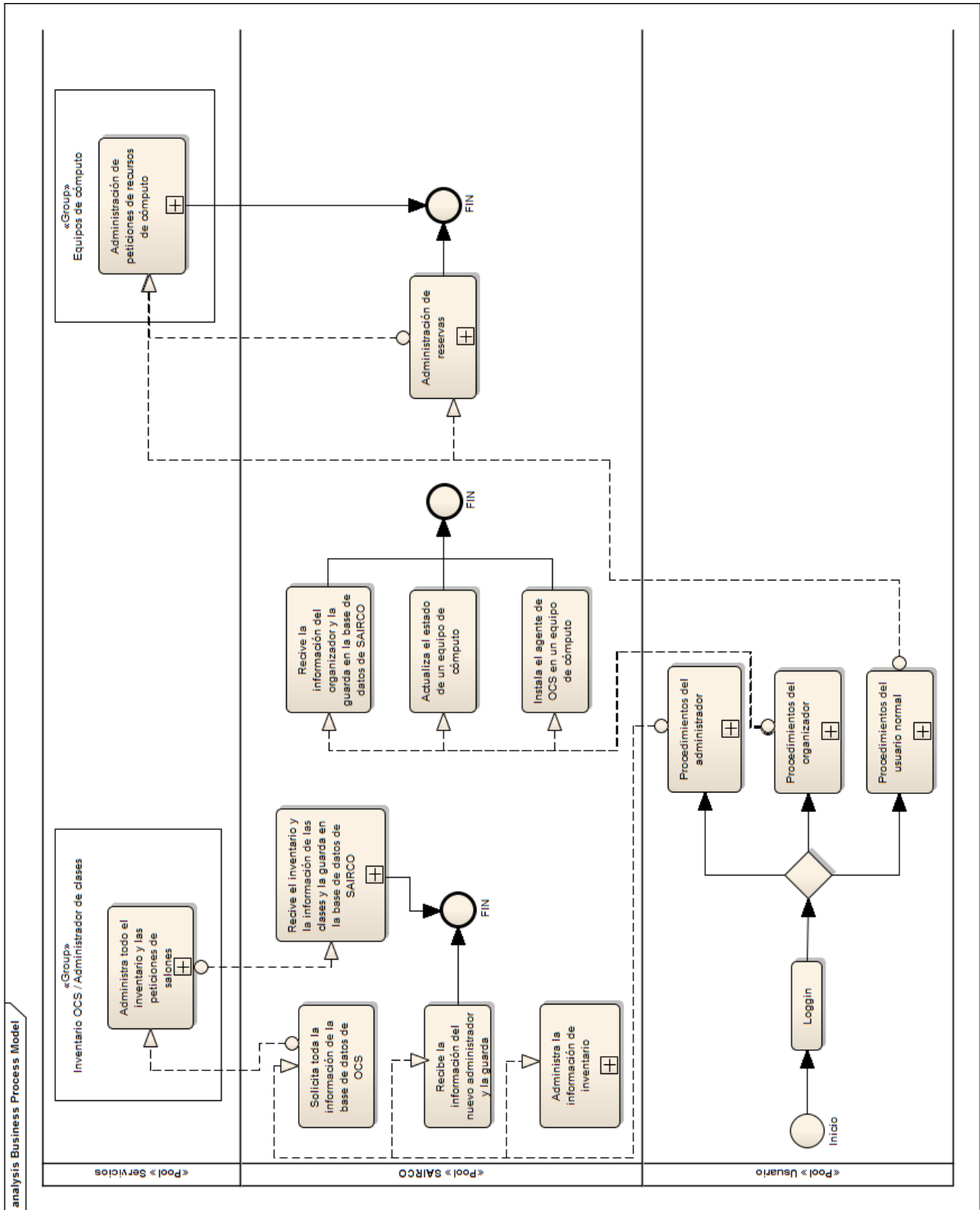


Figure 3.2: BPMN de servicios de SAIRCO.

3.2 Diseño general

Las figs. 3.1 y 3.2 muestran un diseño general de cómo funciona SAIRCO y permiten evidenciar la manera en que interactúan todos sus componentes y la fig. 3.3 muestra de manera detallada la interacción entre ellos.

3.3 Diagramas

Obteniendo los diagramas generales de bpmn y componentes, se puede seguir con los diagramas de casos de uso, clases y entidad-relación necesarios para poder iniciar con la implementación de la aplicación.

3.3.1 Diagrama de casos de uso

3.3.2 Diagrama de clases

Diagrama de clases general

Diagrama de clases de equipos de cómputo

3.3.3 Diagrama de entidad-relación

3.4 Sistema integrado para administración de recursos de cómputo

La Fig.3.2 expone cómo funciona SAIRCO [4]. Se mostrará cómo funciona por medio de tres diferentes actores: administrador, organizador y usuario normal. Cada uno tiene diferentes procesos:

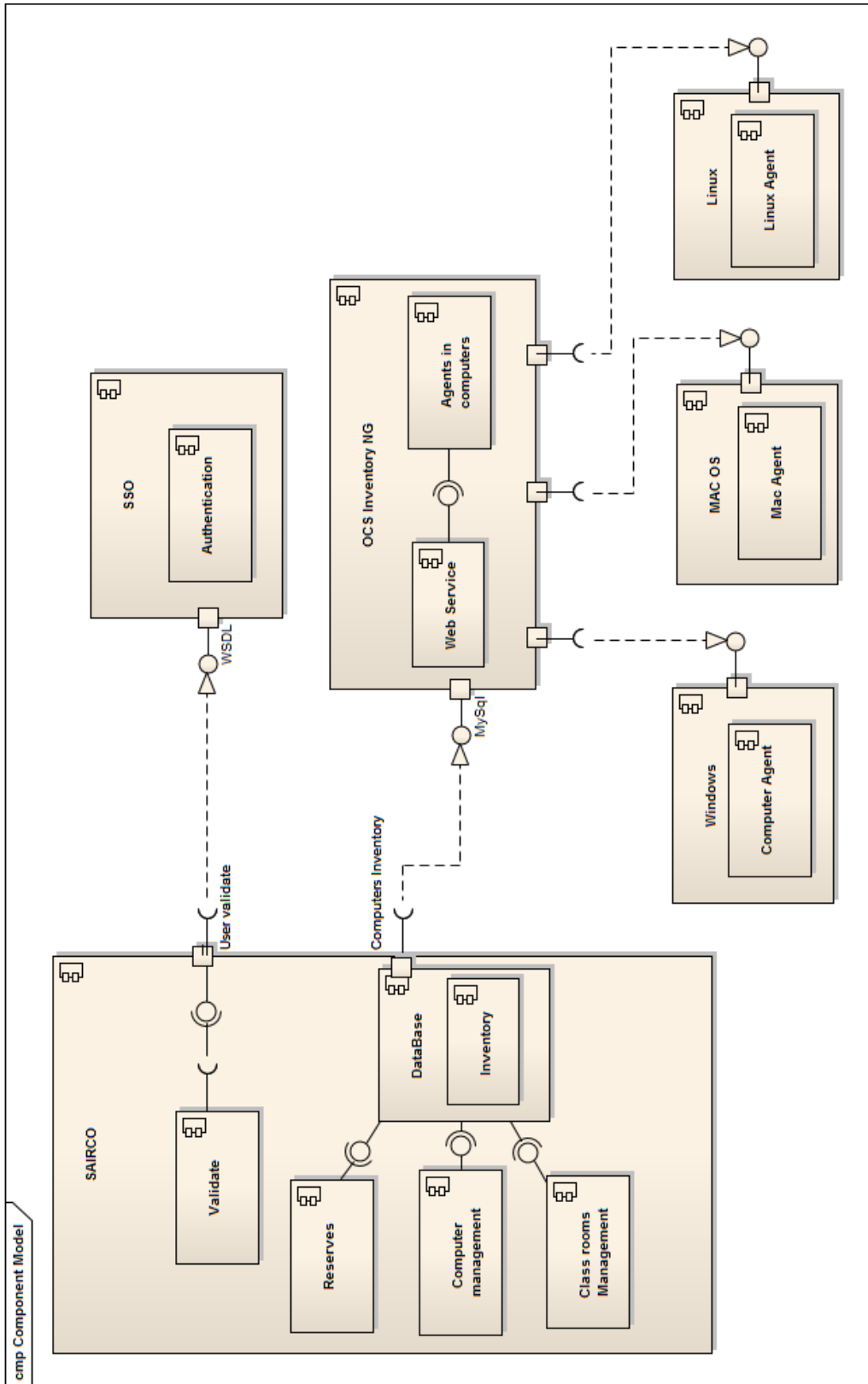


Figure 3.3: Diagrama de componentes.

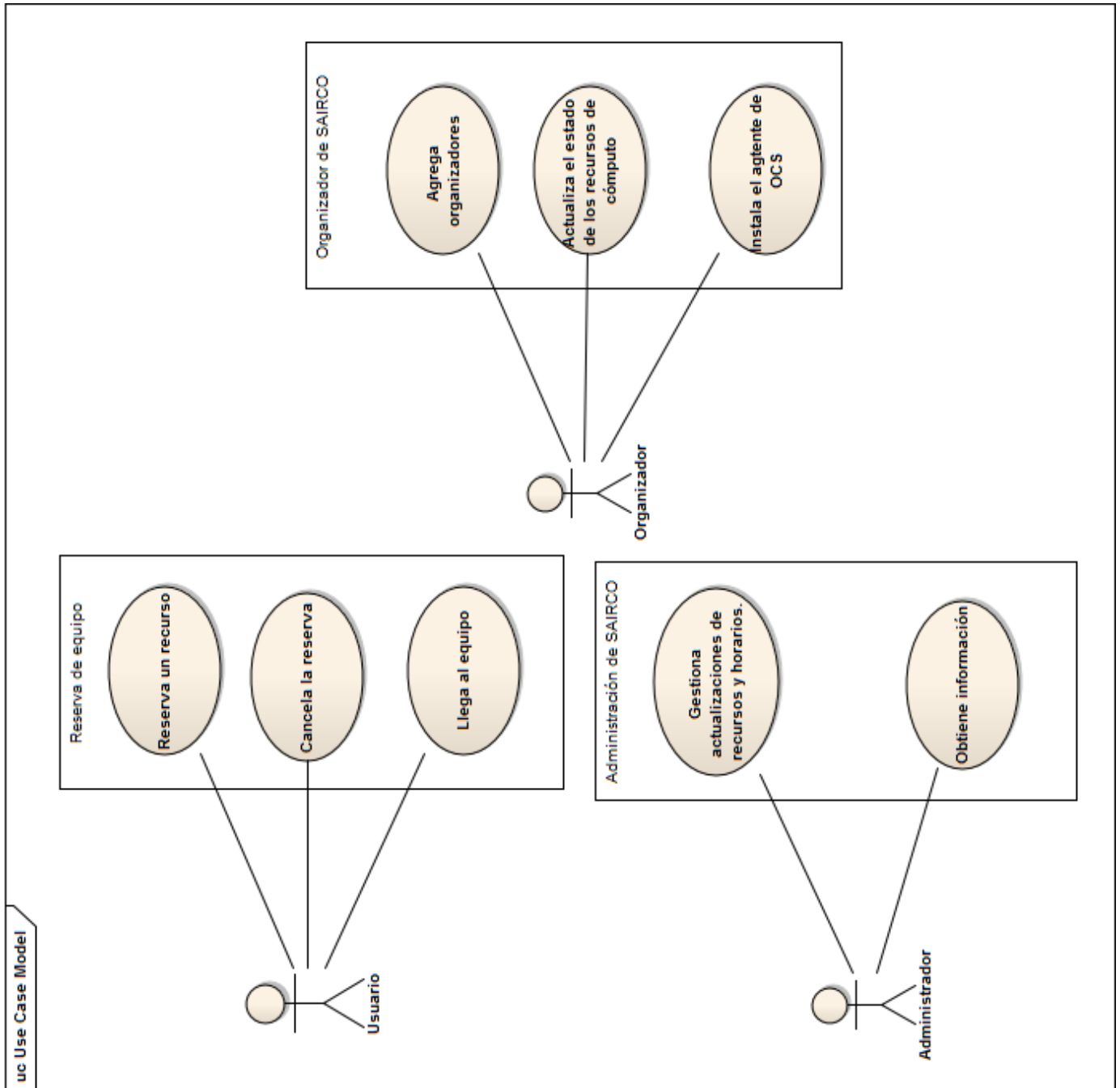


Figure 3.4: Diagrama de casos de uso.

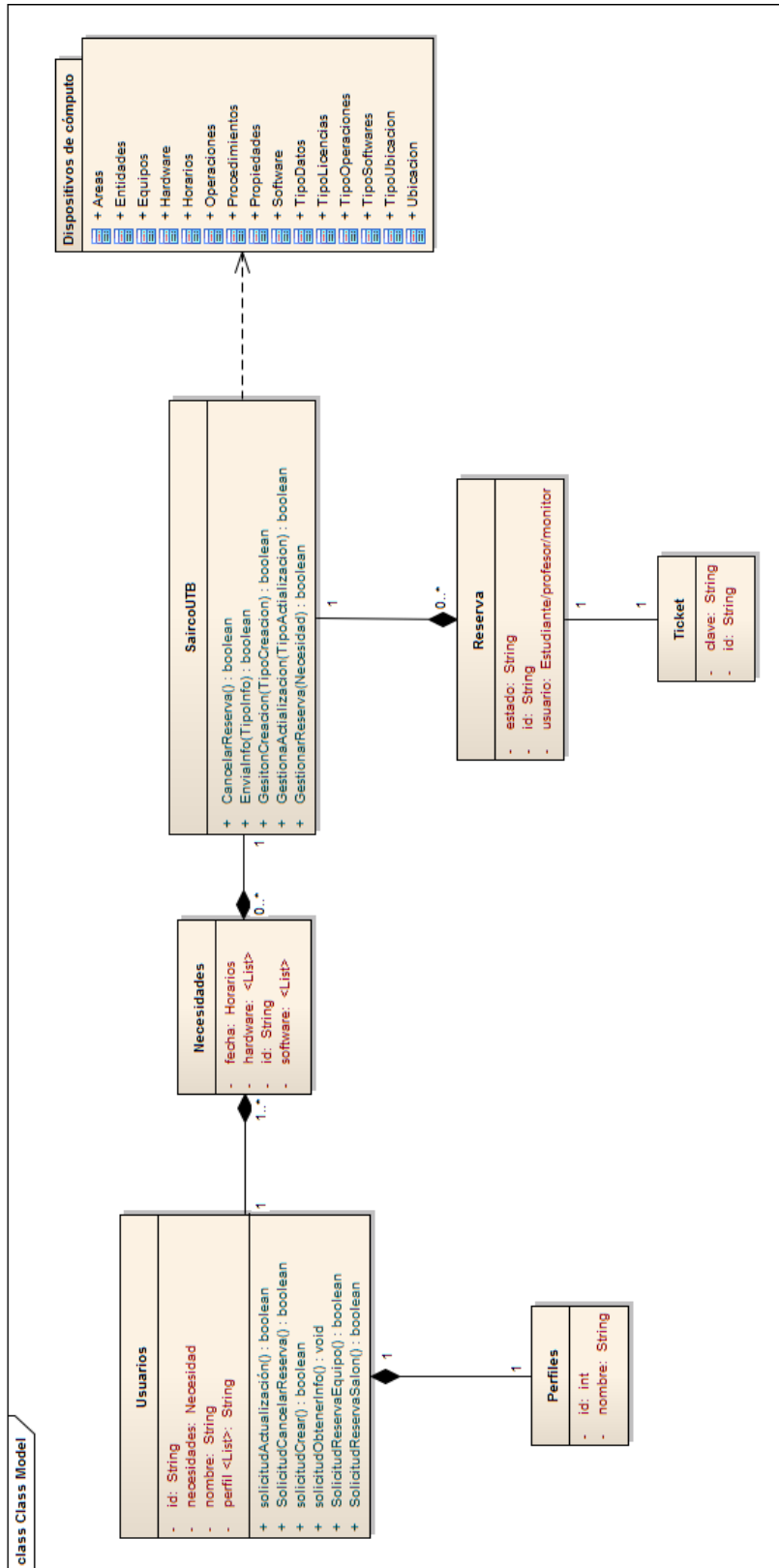


Figure 3.5: Diagrama de clases general.

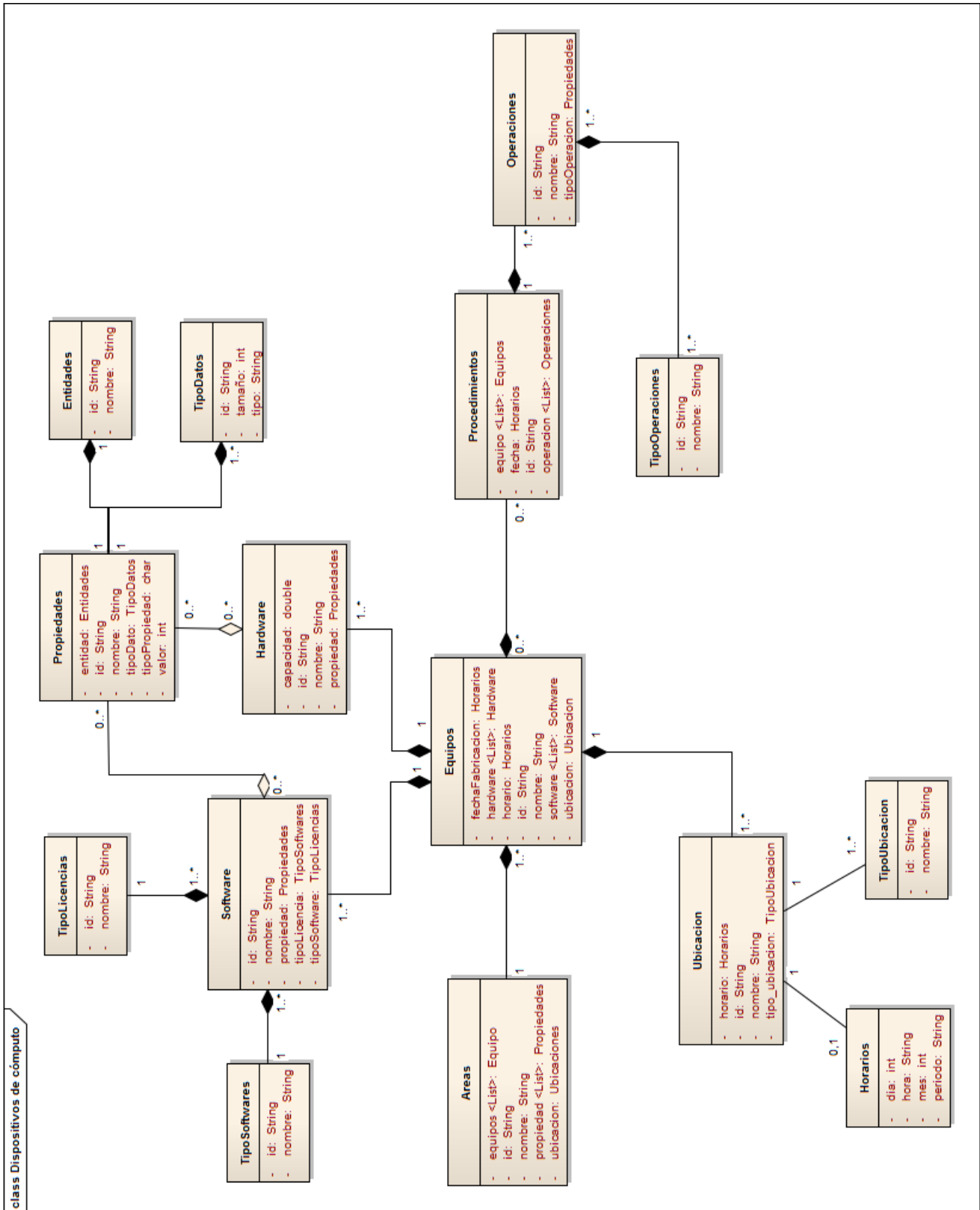


Figure 3.6: Diagrama de clases de equipos.

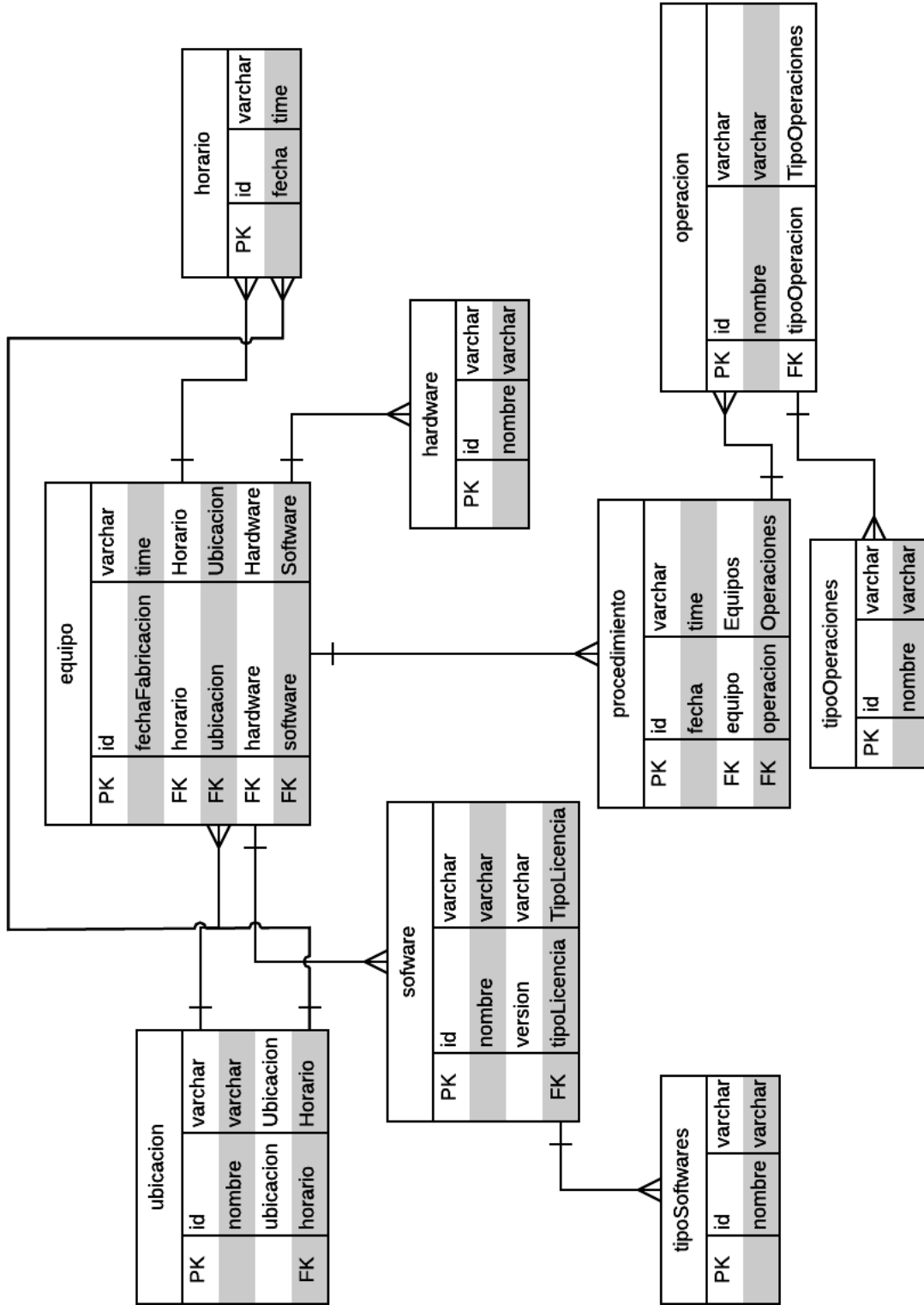


Figure 3.7: Diagrama de entidad-relacion.

3.4.1 Administrador

Un administrador puede asociar el inventario de OCS con el de la base de datos de SAIRCO. Además, puede administrar la información de cualquier recurso de cómputo como nombre y ubicación. Incluso puede visualizar cuántas veces se ha reservado un determinado software o un equipo en específico.

3.4.2 Organizador

El organizador puede asignar una reserva de último minuto de algún usuario que no pudo hacer la reserva por la aplicación SAIRCO. También puede crear nuevos organizadores para ayudarlo con su trabajo. El organizador está a cargo de descargar e instalar el agente de OCS en todos los computadores que se necesite.

3.4.3 Usuario normal

Un usuario normal puede reservar un equipo dependiendo de sus necesidades de hardware y software. Cuando la reserva se haya hecho satisfactoriamente, puede cancelarla si así lo desea, habilitando de nuevo el equipo para el espacio en el que lo había reservado.

3.5 Implementación

3.5.1 Hardware

Se implementó este software usando dos sistemas operativos diferentes: windows server 2012 para la aplicación web y CentOS 6 para la base de datos del inventario.

Ambos en la nube y cada uno con dos procesadores Intel core (Haswell) 3.59 GHz y 2GB of RAM, Apache, MySQL, PHP (WAMP para windows y LAMP para centOS). Se separaron debido a que la versión de php que necesita cada uno es diferente. Además, brinda mayor seguridad ya que el servidor de OCS sólo es visible dentro de la red de la Universidad Tecnológica de Bolívar. La aplicación se desarrolló usando el framework Laravel 5.2 [3] ya que provee una arquitectura de modelo-vista-controlador (MVC) [15] fácil de usar e implementar. Se usó una plantilla de bootstrap llamada "simple side bar" encontrada en [11] que permite mostrar u ocultar el menú de la aplicación web.

3.5.2 Algoritmos

Se usaron los siguientes algoritmos para la implementación del software.

Asignación de reserva hecha por un usuario

Se obtienen los equipos disponibles para reservas en el rango de fechas que ingresó el usuario, de estos se buscan cuáles son los equipos que cumplen con las necesidades del usuario en cuanto a hardware y software y se le asigna al primero que se encuentra. Esto ocasiona que se actualice el estado de reserva al mismo.

Asignación de reserva hecha por un organizador

El organizador selecciona el equipo que satisfaga las necesidades del usuario en la hora en la que lo necesita y a ese se le actualiza el estado de reserva.

Cancelación de reserva

Cuando un usuario decide cancelar una reserva, el mismo selecciona la reserva que quiera cancelar y se actualiza el estado de reserva del equipo ligado a la reserva.

Capítulo 4. Resultados

Luego de haber implementado el software, se obtuvieron los siguientes resultados.

4.1 Resultados obtenidos

Se mostrarán pantallazos de la aplicación desarrollada dividiéndolas por rol, desde la Fig. 4.1 hasta la Fig. 4.16.

4.2 Simulación

Para validar el sistema, se decidió simular el peor caso en donde todos los estudiantes reservan un equipo de cómputo en sus espacios de tiempo libre o "huecos". Los resultados se muestran en la Fig. 4.17. Esta gráfica muestra la diferencia entre la disponibilidad de los recursos de cómputo y los estudiantes de la carrera de ingeniería de sistemas de la Universidad Tecnológica de Bolívar que tienen horas libres en una determinada hora. Se escogió este programa ya que es el programa que más representa el mayor número de reservas de equipos de cómputo. Acá, se puede ver que hay horas críticas en algunos días en donde la universidad no puede satisfacer la demanda de reservas (entre 8:00 y 13:00). Esto se representa con valores negativos. En total, se

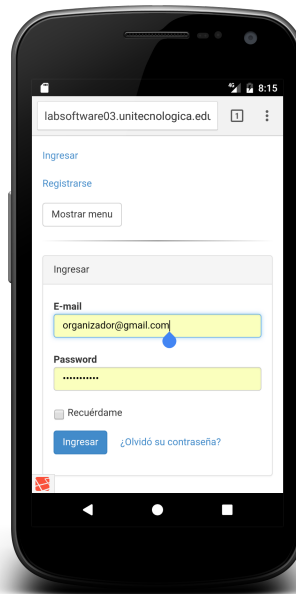


Figure 4.1: Login de la aplicación, dependiendo del rol redirecciona a diferentes interfaces

h

Figure 4.2: Panel del administrador



Figure 4.3: Estadísticas de préstamos de cada software

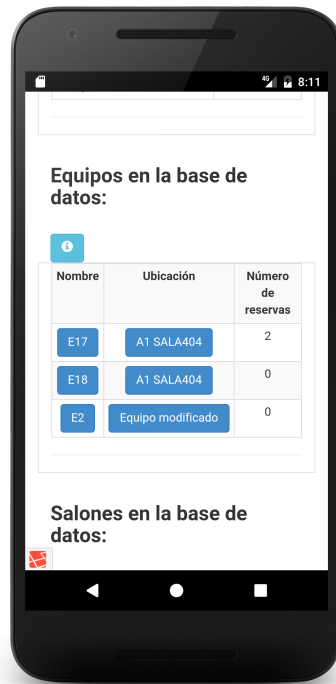


Figure 4.4: Equipos en la base de datos de SAIRCO

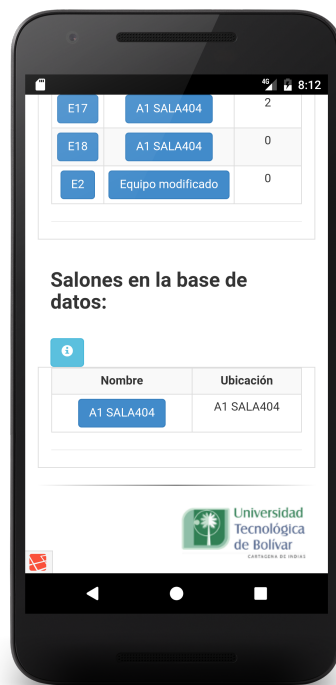


Figure 4.5: Salones en la base de datos de SAIRCO



Figure 4.6: Vista de actualización de un equipo

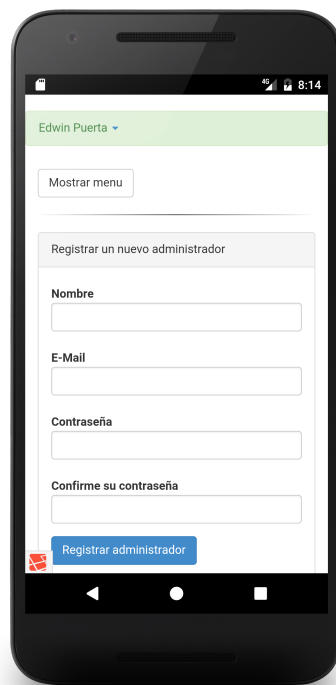


Figure 4.7: Formulario para agregar un nuevo administrador

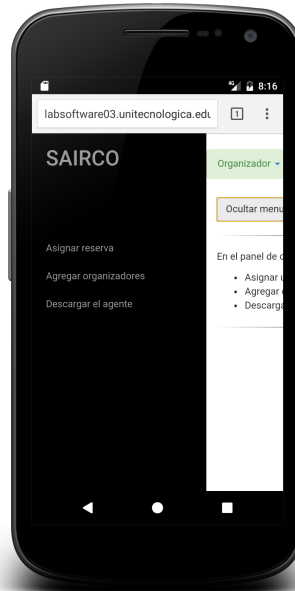


Figure 4.8: Panel del organizador



Figure 4.9: Equipos con reservas disponibles en el día actual



Figure 4.10: Formulario para asignar una reserva de último minuto

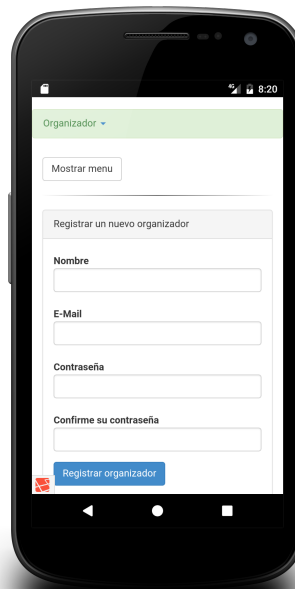


Figure 4.11: Formulario para crear un nuevo organizador



Figure 4.12: Panel del usuario normal

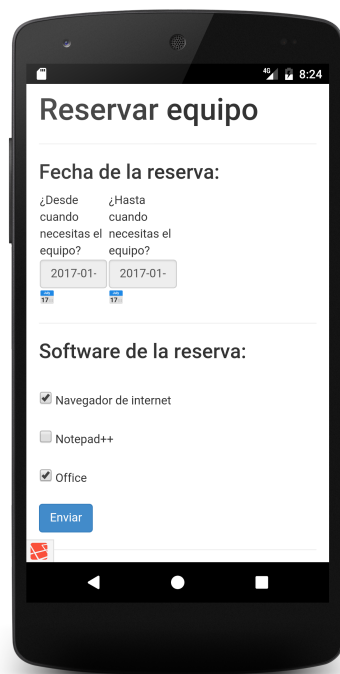


Figure 4.13: Formulario para reservar un equipo de cómputo

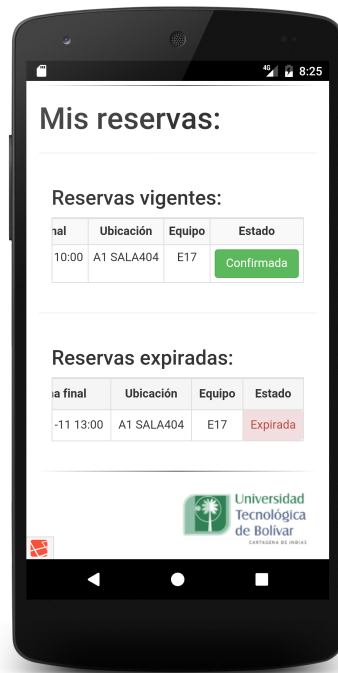


Figure 4.14: Vista de la gestión de reservas por parte del usuario normal



Figure 4.15: Formulario para cancelar una reserva

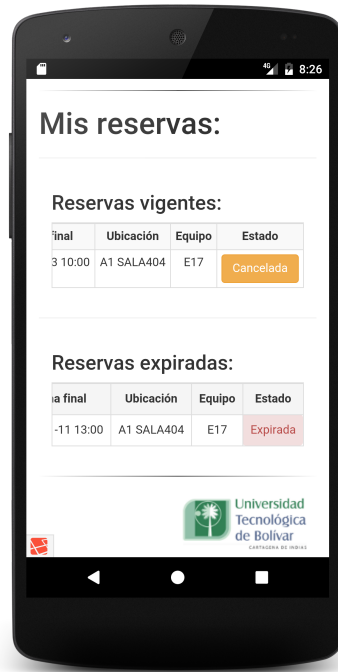


Figure 4.16: Vista de la reserva cancelada

negarían 184 solicitudes que representa el 9% de la demanda total. Esto es mucho mejor que el estado actual, es decir, una negación del 48% de las solicitudes como se muestra en la Fig. 1.1.

Aún así, si se permite al sistema sugerir y reubicar espacios de tiempo antes y después de clases, incluso venir al campus en un día en el que el estudiante no tenga clases, el sistema podría satisfacer todas las solicitudes de reserva de los estudiantes. En la Fig. 4.17, se muestra el total de recursos de cómputo disponibles por día luego de satisfacer la demanda de los estudiantes. Esto evidencia la ventaja de usar SAIRCO.

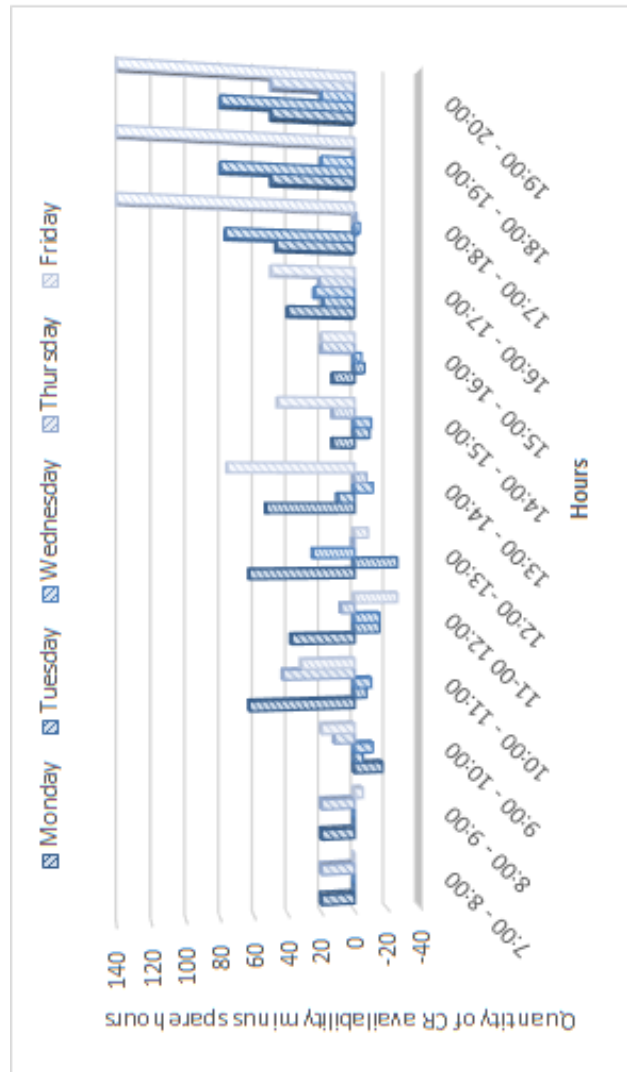


Figure 4.17: Disponibilidad vs horas libres por hora

Capítulo 5.

Conclusiones y Trabajo a futuro

5.1 Conclusiones

Con la Fig. 1.1, se muestra que el sistema de préstamos de equipos de cómputo manejado actualmente por la Universidad Tecnológica de Bolívar se está desperdiando la capacidad de préstamos de los mismos. Esto evidencia que se necesita un sistema diferente que optimice el actual.

Con el estado del arte se pudo identificar las herramientas que podían ayudar con el inventario de hardware y software y además, se encontró que esta herramienta de asignación de recursos de cómputo dependiendo del hardware y software que necesite el cliente es innovadora.

La propuesta demuestra un gran impacto positivo en el sistema actual de asignación de recursos de cómputo, según se explicó en la simulación previa.

Conociendo el número de reservas de cada software, se puede identificar si se necesitan más licencias de un determinado software o menos.

5.2 Trabajo a futuro

Como trabajo futuro se propone mejorar el algoritmo de asignación ya que actualmente sólo está asignando el primero que encuentra disponible que cumpla las necesidades del cliente.

Se podría aplicar inteligencia artificial pidiendo una calificación de cada recurso de cómputo prestado. Esto permitiría mejorar aún más la satisfacción del cliente al momento de usar los recursos de cómputo.

Bibliography

- [1] About aranda. <http://arandasoft.com/aranda-inventory/>. Accessed: 2015-10-05.
- [2] About cherwell. <https://www.cherwell.com/about/about-cherwell/>. Accessed: 2015-10-05.
- [3] About laravel. <https://laravel.com/docs/5.2/>. Accessed: 2016-11-02.
- [4] About sairco. labsoftware03.unitecologica.edu.co. Accessed: 2017-01-01.
- [5] About spicework. <http://www.spiceworks.com/app/>. Accessed: 2015-10-05.
- [6] Desktop management interfaces. <http://foldoc.org/Desktop+Management+Interface>. Accessed: 2015-10-05.
- [7] Desktop management task force. "software standard groups definition, version 2.0". <http://www.dmtf.org/tech/apps.html>. Accessed: 2015-10-05.
- [8] Global diagram of ocs inventory ng. <http://www.ocsinventory-ng.org/en/about/global-diagram.html>. Accessed: 2015-10-15.
- [9] Intel's landesk client manager. <http://foldoc.org/Intel>. Accessed: 2015-10-05.
- [10] Opsi characteristics. <http://www.opsi.org/en/features>. Accessed: 2015-10-05.
- [11] The simple sidebar template. <http://startbootstrap.com/template-overviews/simple-sidebar/>. Accessed: 2016-11-02.
- [12] Xampp apache+ mysql+ php+ perl documentation. <https://www.apachefriends.org/es/index.html>. Accessed: 2015-10-05.
- [13] P. Ferril. Keeping up with desktop details, network world magazine. pages 37–41, 2003.
- [14] S. Koolmanojwong and B. Boehm. The incremental commitment model process patterns for rapid-fielding projects. In *International Conference on Software Process*, pages 150–162. Springer, 2010.

- [15] G. E. Krasner, S. T. Pope, et al. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, 1(3):26–49, 1988.
- [16] J. Martin. *Rapid application development*. Macmillan Publishing Co., Inc., 1991.
- [17] K. Schwaber and M. Beedle. *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River, 2002.