



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Sistema de Registro de Incidencias y Control de Inventarios (SIRICI)

TESIS

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A N :

Ángel Alberto Badillo Cinta

Sergio Andrés Becerril López

Ulises Pedro Hernández Molina

DIRECTOR DE TESIS:

M.I. Juan Carlos Roa Beiza



Ciudad Universitaria, Cd. Mx., 2017



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Los tres integrantes del presente trabajo agradecemos el apoyo del Programa de Apoyo a la Titulación (PAT) de la Facultad de Ingeniería, sin el cual hubiera sido imposible alcanzar este logro.

En particular, le extendemos un sincero y profundo agradecimiento al M.I. Juan Carlos Roa Beiza, quien nos guio durante todo el proceso con dedicación y vocación profesional. Gracias Maestro.

Ángel

A mi madre, Leticia.

Por su amor, comprensión y apoyo sin condición. Por ser un ejemplo y enseñarme que para ser el arquitecto de mi futuro primero hay que tener fuertes bases éticas y morales. Además de dejarme la mejor herencia, la dedicación y compromiso con la educación como recurso para ser un mejor ciudadano y crear una mejor nación. Este trabajo se lo dedico. Es mi manera de decirle gracias y transmitirle el orgullo que siento por ella como mujer, profesora y lo mas importante como madre.

A la memoria de mis abuelos, Guadalupe, Esperanza y Ángel.

Por sus cuidados, guía y haber estado conmigo en cada paso. Gracias por haberme inculcado valores para ser una persona recta y honorable.

A la memoria de mi tía, María de la Luz.

Por haber estado al pendiente de mi y apoyarme para llegar a lo mas altos. Gracias por esos memorables viajes a Michoacán.

A mi familia.

Por estar conmigo en todo momento y por ser mi soporte principal en cada meta y proyecto emprendido.

A mi Universidad Nacional Autónoma de México.

En especial a la Facultad de Ingeniería por darme el gran honor de ser egresado de tan valiosa institución y brindarme el conocimiento y visión para transformar a mi país. Es un gran orgullo ser Ingeniero egresado de la Universidad Nacional Autónoma de México.

A mis compañeros de tesis.

Por haber aceptado el reto de embarcarnos en este dura e interesante aventura y llegar a buen puerto. Este trabajo es la culminación de un objetivo y esfuerzos compartidos. También es el inicio de nuevos proyectos.

Quiero hacer parte de mi agradecimiento a todas las personas que han dejado una enorme huella y que con su energía, conocimiento, experiencia y sonrisa han cambiado mi visión del mundo. A todos, mi mayor gratitud.

Sergio

Agradezco primeramente a mis compañeros, Ángel y Ulises, por apoyarme a la conclusión de este trabajo y por sacrificarse para que fuera un esfuerzo de calidad.

A mi familia: mi esposa Elena, mi hija Mariana, y mi papá Sergio. Gracias por apoyarme en estos meses, como siempre lo han hecho. Este logro es tan suyo como mío.

A mi escuela, la Facultad de Ingeniería, por brindarme los conocimientos necesarios para resolver este y todos los retos profesionales a los que me he enfrentado.

A la Escuela Nacional de Lenguas, Lingüística y Traducción. Mi segunda casa. Espero que este trabajo rinda todos los resultados deseados, por un largo tiempo.

A mi mamá. No pudiste verlo, pero sin ti no lo hubiera logrado. Te llevo siempre.

Ulises

Primero, quiero dar gracias a *Dios* por permitirme cerrar el camino de la titulación, estar conmigo e iluminarme en este trayecto.

A mi esposa, Luz Maria por su apoyo incondicional, por su amor, comprensión y la mejor consejera. A Mis hijos Melissa y Ulises Uriel, ejemplo de superación e inspiración. A mi madre querida que día a día se preocupa por mí, por su cariño incondicional. Mi padre y Héctor mi hermano Q.E.P.D. quienes me recordaba que hay que terminar para obtener la satisfacción completa. Mis hermanos quienes han estado al tanto de mi bienestar.

A mi familia mencionada, que en su momento fueron el motor para seguir adelante en mis estudios, por dejarme la mejor herencia que es la educación, el mejor camino para enfrentar la vida.

A mis compañeros de Tesis, Sergio y Angel. Gracias por haber aceptado trabajar este reto conmigo, por la convicción al profesionalismo y trabajo.

A la UNAM, la Facultad de Ingeniería, y todas las instituciones relacionadas con esta Máxima Casa de Estudios en las que tuve el placer de estar, por permitirme obtener conocimientos, la oportunidad de conocer gente admirable entre profesores y alumnos.

También quisiera hacer parte de mi agradecimiento a todas las personas que al paso de mi vida han dejado una huella y ahora son parte de mí, en general a toda esa gente que con su luz llenan de vida mis días.

A todos con cariño mi mayor reconocimiento y gratitud.

Índice

1. Antecedentes	1
1.1. Introducción	3
1.2. La Escuela Nacional de Lenguas, Lingüística y Traducción.....	7
2. Análisis y planteamiento del problema	11
2.1. Análisis del problema	13
2.2. Análisis de los procesos de control de activos de hardware de la ENALLT.....	16
2.3. Análisis de los procesos de registro y seguimiento de solicitudes de soporte técnico de la ENALLT	20
2.4. Levantamiento de requerimientos.....	23
2.5. Alcance y acotamiento del problema.....	26
2.6. Planteamiento de la solución.....	29
2.7. Diseño de la arquitectura del sistema.....	37
3. Marco teórico	45
3.1. Características, ventajas y desventajas de las bases de datos relacionales y el manejador de bases de datos MySQL	47
3.2. Características, ventajas y desventajas del lenguaje python	54
3.3. Características, ventajas y desventajas del framework Django	58
3.4. Características, ventajas y desventajas del lenguaje HTML.....	61
3.5. Características, ventajas y desventajas del servidor de páginas web Apache	64
3.6. Características, ventajas y desventajas de JavaScript y CSS.....	71
4. Diseño y construcción del sistema	77
4.1. Modelado del sistema	79
4.1.1. Diagrama de contexto.....	79
4.1.2. Diagrama de casos de uso.....	81
4.1.3. Diagrama de flujo de datos	87
4.1.4. Diagrama de componentes	95
4.1.5. Diagrama de despliegue.....	100
4.1.6. Diagrama de clases para el módulo web	103
4.2. Diseño, construcción e implementación de la base de datos.....	107

4.2.1. Diagrama de entidad-relación.....	108
4.2.2. Diccionario de datos.....	111
4.2.3. Normalización	134
4.3. Diseño y construcción del frontend	135
4.3.1. Diseño de la interfaz gráfica empleando RUP y UML	139
4.4. Pruebas y mantenimiento (riesgos e imprevistos y mitigación de estos).....	144
4.5. Evaluación integral del aplicativo backend, frontend y reportes	154
Conclusiones	159
Bibliografía	165

CAPÍTULO I. ANTECEDENTES

1.1 Introducción.

La administración se encuentra vinculada al rendimiento y funcionamiento de las instituciones. Un recurso es un medio que permite conseguir la solución a algún problema. Por lo tanto, la administración de recursos consiste en el manejo adecuado de estos medios y conlleva a la capacitación profesional y al bienestar de cualquier empresa, dependencia o entidad para proporcionar servicios de forma eficiente.

La función de un almacén está ligada al concepto de gestión. Ésta es la administración y dirección de una entidad empleando una serie de procedimientos, a través de la organización de recursos, buscando satisfacer objetivos de manera eficaz. La gestión de un almacén no es únicamente salvaguardar los bienes allí depositados sino, primordialmente, actuar como un intermediario entre las entradas, salidas, altas, bajas, envíos de soporte y seguimiento de los recursos.

Los inventarios tienen el propósito de administrar recursos con el objetivo de satisfacer la continuidad de algún servicio proporcionado. Estos existen en cualquier organización, en particular, en centros educativos. El llevar a cabo un control preciso y una actualización constante es uno de los principales retos.

El almacén es un espacio de servicios destinado a la custodia, control y abastecimiento de bienes dentro de una estructura organizacional. De igual forma, los almacenes son una infraestructura imprescindible para la actividad de todo tipo de agentes (económicos, sociales, entre otros).

En este trabajo de Tesis se aborda el problema que significa administrar grandes e importantes cantidades de activos de hardware en instituciones educativas, particularmente en la Escuela Nacional de Lenguas, Lingüística y Traducción (ENALLT), con la finalidad de llevar a cabo registros de las características específicas de estos activos, así como mitigar riesgos como desubicación y pérdida de bienes, obsolescencia, uso indebido, incumplimiento en soporte, fraude y falsificación. También es importante

para las organizaciones como la ENALLT la generación de reportes para el control de sus activos de hardware, así como para el cumplimiento de entrega de información a terceros según sea el caso. En particular, las dependencias de la UNAM deben entregar periódicamente reportes a otras dependencias, como la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC).

Aunado a este problema, se requiere un control de las solicitudes de soporte que los usuarios de la ENALLT requieren, que pueden o no estar ligadas a un activo de hardware y que requieren un registro fehaciente, no solo para la generación de reportes (como el número de solicitudes atendidas) y métricas de tiempos de atención, sino también para la ubicación de activos de hardware especialmente problemáticos (con alta incidencia de atención) y para contar con una base de datos que pueda ser consultada por el equipo técnico para la mejora en la atención de solicitudes de soporte posteriores.

En una primera instancia se reseñan los actuales procesos de control de activos de hardware en la ENALLT, así como el registro, verificación y seguimiento de eventos. Además, se detallará el ciclo de vida de un activo desde su adquisición, solicitud de salida, actualización, asignación o reasignación hasta su baja definitiva.

El sistema que se desarrolla en el presente trabajo tiene dos objetivos principales:

1. Llevar un control eficiente de los activos de hardware como: CPU's, monitores, teclados, mouses, dispositivos de almacenamiento (discos duros en red), dispositivos de telecomunicaciones (switches, APs), tabletas de digitalización, cámaras digitales, bocinas, proyectores, impresoras, lápices ópticos y pantallas, como se muestra en la figura 1.1.1:



Figura 1.1.1 Activos de hardware

En ese sentido, abordaremos el problema a través de un análisis de los actuales procesos de administración de recursos, puntualizando las principales oportunidades de mejora y buscando desarrollar un sistema que eficiente estos procesos y que sea accesible por todos los usuarios que lo requieran.

2. Llevar un control de las solicitudes de soporte (*tickets*) dentro de la ENALLT, facilitando el proceso de atención por parte del personal técnico y automatizando la generación de reportes necesarios para la Escuela, tanto internos como externos, incluyendo métricas del tiempo de atención.

También se plantea la metodología para el desarrollo e implementación del sistema, mostrando los criterios que se emplean para la elección de la plataforma, lenguaje de programación, bases de datos y estándares existentes.

El sistema adopta el nombre de Sistema de Registro de Incidencias y Control de Inventarios (SIRICI).

La razón por la que elegimos este tema obedece a la necesidad de implementar una propuesta tecnológica que resuelva el problema que implica el registro de solicitudes de soporte y control de activos mediante métodos manuales. De igual modo, proporcionamos una metodología detallada acerca de la recopilación de los distintos recursos de hardware de la ENALLT, solicitudes de soporte técnico, levantamiento de requerimientos y planteamiento de soluciones.

Los sistemas de inventario tradicionales se enfocan en el control de existencias en un almacén y, por tanto, no resuelven de manera efectiva el problema de la Escuela. Es por ello que es necesario desarrollar una solución de software que permita resolver las necesidades específicas de este problema.

Este trabajo expone los módulos funcionales que conforman al sistema así como pruebas, mantenimiento y evaluación de eficiencia operativa. De la misma manera, se describen el diseño y desarrollo haciendo énfasis en los aspectos de modelado del sistema y la metodología elegida presentando los diagramas de contexto, casos de uso, flujo de datos, entidad - relación, componentes, despliegue y clases que apoyan al modelado del sistema.

La base de datos cobra una gran importancia debido a que nos proveerá de herramientas que eviten la duplicidad de registros, suplantación de identidades y fraudes, garantizando la integridad referencial y favoreciendo la normalización.

De todo el estudio se obtienen requerimientos tanto generales como particulares que son cubiertos por una solución tecnológica y que además plantea probables mejoras en el futuro que garanticen el flujo eficiente de la información.

1.2 La Escuela Nacional de Lenguas, Lingüística y Traducción.

La Escuela Nacional de Lenguas, Lingüística y Traducción tiene su origen en el Centro de Enseñanza de Lenguas Extranjeras (CELE). La dependencia fue creada el 30 de noviembre de 1966.

Desde su fundación, la labor de la ENALLT ha sido la de desarrollar la enseñanza y el aprendizaje de lenguas extranjeras; la certificación, formación y actualización de sus recursos; así como la investigación y divulgación de sus servicios a la comunidad universitaria. Actualmente se enseñan 16 lenguas internacionales y una nacional: alemán, francés, inglés, italiano, portugués, ruso, japonés, árabe, catalán, chino, coreano, griego moderno, hebreo, rumano, sueco, vasco y náhuatl. Véase figura 1.2.1.



Figura 1.2.1 Instalaciones de la Escuela Nacional de Lenguas, Lingüística y Traducción

En 1979 fue creado el Departamento de Lingüística Aplicada con seis áreas de investigación y un Seminario Permanente de Lingüística Aplicada. Ese mismo año, se aprobó la Maestría de Lingüística Aplicada y la institución comenzó a tomar parte en la Especialización en la Enseñanza de Español como lengua extranjera.

En 1980 fue inaugurada la biblioteca e iniciaron los cursos de la Coordinación de Formación de Profesores. Fue hasta 1984 cuando la biblioteca tomó el nombre de “Stephen A. Bastien” en memoria de uno de sus fundadores.

La certificación es un objetivo muy importante. En 1989 se estableció la Coordinación de Evaluación y Certificación, con la misión de certificar programas de estudio de idiomas en la universidad y para sus estudiantes. Actualmente, la institución certifica seis idiomas: alemán, chino, francés, inglés, italiano y portugués.

En 1992 entra en funcionamiento la Sala de Cómputo y en 1994 la institución amplía sus servicios con la puesta en marcha de centros y programas de extensión. Un año después abre sus puertas la Coordinación de Mediateca.

En los últimos ocho años, la Mediateca incorporó la Sala de Innovación, Docencia e Investigación (SIDI) que cuenta con tecnologías de la información y comunicación para la enseñanza de lenguas, promoviendo el desarrollo de recursos y capacitación docente.

El 24 de marzo del presente año, el Consejo Universitario aprobó “la creación de la Escuela Nacional de Lenguas, Lingüística y Traducción con la consecuente modificación de los artículos 8º del Título Segundo, y 106 fracción IV del Título Octavo del Estatuto General de la Universidad Nacional Autónoma de México”¹ a fin de responder a la creciente demanda de profesionistas en esta área de estudio. Esta transformación se fundamenta en información que arroja un estimado de 200 millones de personas que residen fuera de su país de origen, y esta movilidad se encuentra enfrascada en una gran diversidad lingüística.

La institución ha ampliado sus funciones de acuerdo a las exigencias de las nuevas tecnologías, ofreciendo:

- Formación e información especializada en línea,

¹ Acuerdo de la sesión extraordinaria del 24 de marzo de 2017 del Consejo Universitario de la Universidad Nacional Autónoma de México.

- Un Departamento de Traducción e Interpretación.
- Desarrollo de proyectos académicos en línea.

La ENALLT ha tenido diversas colaboraciones con universidades nacionales y extranjeras, fomentando el intercambio de profesores e investigaciones. Sus programas se sustentan en el estudio e investigación de la lingüística aplicada. En los últimos años, la institución ha respondido a un incremento en su matrícula con la modernización de sus sedes y dando respuesta a necesidades específicas en aprendizaje de lenguas planteadas por entidades universitarias y gubernamentales.

La ENALLT ha llevado a cabo estrategias de comunicación con el objetivo de divulgar las actividades en los diversos ámbitos de la institución. Asimismo, la Biblioteca “Stephen A. Bastien” y el Departamento de Cómputo se enriquecieron con nuevos acervos y recursos, cumpliendo con el objetivo de satisfacer con servicios de apoyo a la comunidad.

CAPÍTULO II. ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA

2.1 Análisis del problema

Como ya hemos visto, actualmente la ENALLT realiza un procedimiento de control de sus activos de hardware primordialmente a través de su Unidad Administrativa. El proceso puede resumirse a través de la figura 2.1.1.

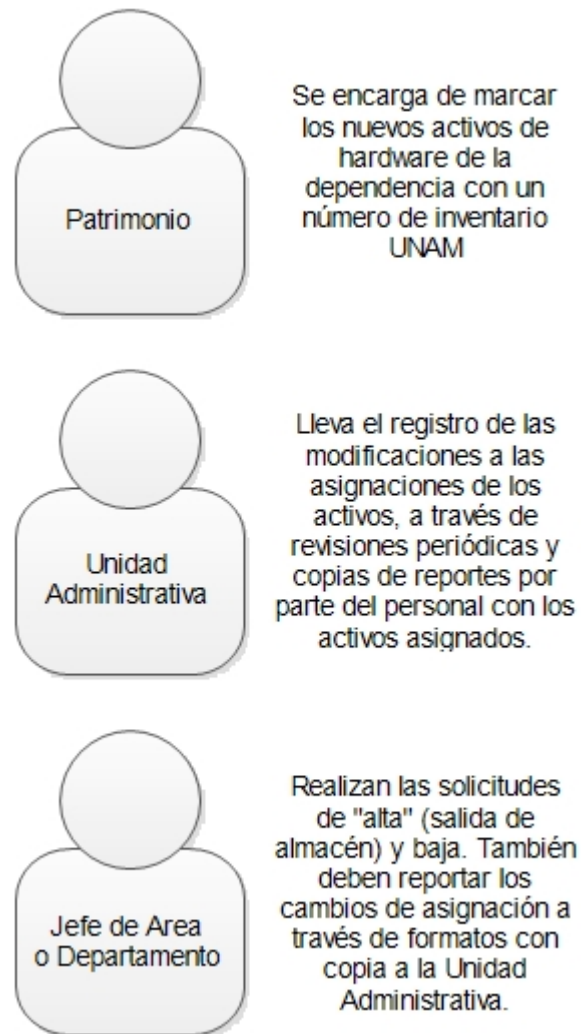


Figura 2.1.1. Actores involucrados en el proceso de control de activos de hardware de la ENALLT.

Existen múltiples oportunidades de mejora en las que podemos trabajar.

- La información recolectada acerca de los activos de hardware es muy limitada. Únicamente se registra quién tiene asignado el equipo, su número de inventario, marca, modelo y número de serie.

- Los activos de hardware pueden sufrir cambios en su asignación sin ser necesariamente levantado un reporte, debido a que los cambios de asignación se realizan a discreción del Departamento de Cómputo.
- Aún al realizar un reporte, no necesariamente se emite la copia respectiva a la Unidad Administrativa, debido a que sacar una copia y entregarla implica bajar dos pisos y luego entregarla en el primer piso de otro edificio.
- Obtener un reporte para el Departamento de Cómputo (listando, por ejemplo, todas las impresoras) es difícil, ya que la Unidad Administrativa no tiene manera de ordenar los registros, que se van agregando conforme van llegando. De manera efectiva, el número de inventario es el único orden posible para los activos.
- La Dirección General de Tecnologías de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) de la UNAM, solicita de manera anual a cada dependencia varios reportes, segregando los activos de cómputo de acuerdo a diferentes criterios (tipo de procesador, sistema operativo instalado, etc.) Actualmente no existe manera de realizar estos reportes de manera automática, y se deben generar manualmente con una revisión de cada uno de los activos de hardware.
- Buscar un activo de hardware específico tampoco es fácil, si por alguna razón su ubicación no fue actualizada, y no conocemos su número de inventario. Esta situación ya ha causado gastos innecesarios de tiempo por parte del personal del Departamento de Cómputo en varias ocasiones.

Por su parte, el proceso de registro de solicitudes de soporte técnico y la atención a las mismas se resume en anotaciones en una libreta. Aquí también existen múltiples oportunidades de mejora:

- Las solicitudes de soporte son registradas exclusivamente por una persona (la secretaria del Departamento de Cómputo), a pesar de que cualquier miembro del equipo técnico puede atender las solicitudes.

- Por carga de trabajo, usualmente no se anotan (mucho menos a detalle) las acciones realizadas por el equipo técnico del Departamento de Cómputo, ni la fecha y hora de las mismas.
- Debido a que el registro de estas solicitudes y atenciones es, en el mejor de los casos, esporádico, es imposible tener un registro fehaciente de la historia de soporte de un activo de hardware, negando la posibilidad de identificar aquellos activos que son problemáticos y, por ende, un reemplazo proactivo de los mismos.
- La jefatura del Departamento no puede determinar si las solicitudes son atendidas de manera rápida, eficiente, o incluso si los reportes son atendidos completamente. Es factible, debido a la ausencia de un registro 100% completo, que hoy en día existan solicitudes que nunca fueron atendidas. Tampoco es posible entregar métricas relativas al caso, ni realizar planeación o alguna medida de mejora, al no existir una línea base de soporte. Tampoco es posible determinar el nivel de servicio del Departamento de Cómputo.
- Cada año es necesario que el personal técnico del Departamento de Cómputo realice un reporte, indicando entre otras cosas, el número de solicitudes de soporte atendidas. Actualmente, este cálculo se realiza de manera estimada, ya que el registro de dichas atenciones no se lleva con cuidado.

El personal del Departamento de Cómputo insiste en una falta de conveniencia a la hora de realizar el registro de solicitudes y, sobre todo, de atenciones. Indican que toma mucho tiempo llenar un registro en la libreta, que no siempre está la libreta disponible, o que deben registrar mucho después del evento, confundiendo en ocasiones detalles sobre el mismo.

También falta una forma de ligar las atenciones de soporte a los activos de hardware de manera que sea posible tener un registro histórico.

Todas estas situaciones podrían mejorarse con un sistema de control que manejara la información de la manera más automatizada posible, minimizando los tiempos de entrada de datos y automatizando la generación de los reportes que la Escuela necesita.

2.2 Análisis de los procesos de control de activos de hardware de la ENALLT.

El inventario de activos en la ENALLT constituye una estadística física de los recursos existentes, que normalmente se confrontan con el registro anotado en bitácoras. Su control y actualización es uno de los principales problemas de la institución.

En esta sección se describirá el proceso de inventariado de activos de hardware que se encuentran resguardados en el Departamento de Cómputo de la ENALLT. Además, se explicará el método de adquisición de recursos mediante el esquema de compra directa a proveedor y el curso que sigue hasta el fin de su vida útil. Cabe señalar que ambos procedimientos se llevan a cabo mediante formatos que son entregados al encargado o dependencia asignada al almacén y/o compras según sea el caso.

A continuación, en la figura 2.2.1 se lleva a cabo la descripción de adquisición de bienes, así como el proceso de marcado con número de inventario y asignación al Departamento o Área correspondiente.

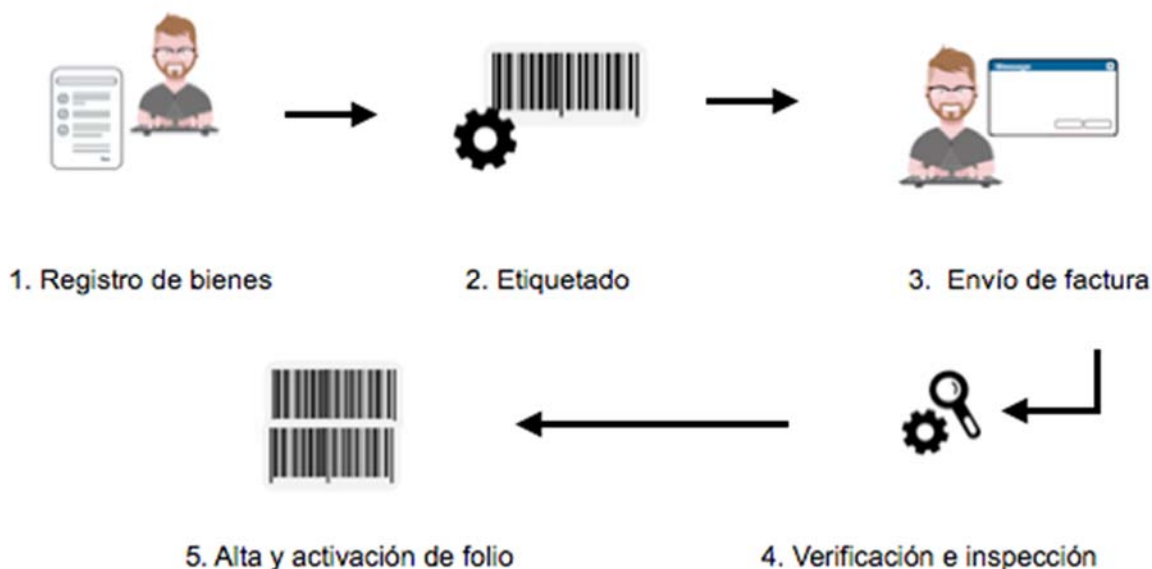


Figura 2.2.1 Proceso de adquisición de bienes por Entidades o Dependencias Universitarias.

1. Las Entidades o Dependencias Universitarias llevan a cabo el registro de los bienes en la Dirección General del Patrimonio Universitario (DGPU) durante los 10 días hábiles posteriores a la fecha que se facturó.
2. Se registran en el Módulo de Captura y se etiquetan.
3. Se envían, de manera digitalizada, la factura y la solicitud interna de compra al sitio web <http://www.patrimonio.unam.mx/sicopimx/>.
4. La Dirección General del Patrimonio Universitario revisa la documentación y programa la verificación física de los bienes para corroborar que la marca, número de serie y modelo concuerden con lo registrado.
5. Se activa el folio de Alta de Bienes Adquiridos, las Entidades o Dependencias Universitarias imprimen el “Comprobante de Registro de Bienes Autorizados por la DGPU por Trámite de Pago en la Unidad de Proceso”² con doble código de barras.

En el diagrama de proceso de Control de Activos de Hardware, que se muestra en la figura 2.2.2 se detalla el proceso de alta de los activos adquiridos.

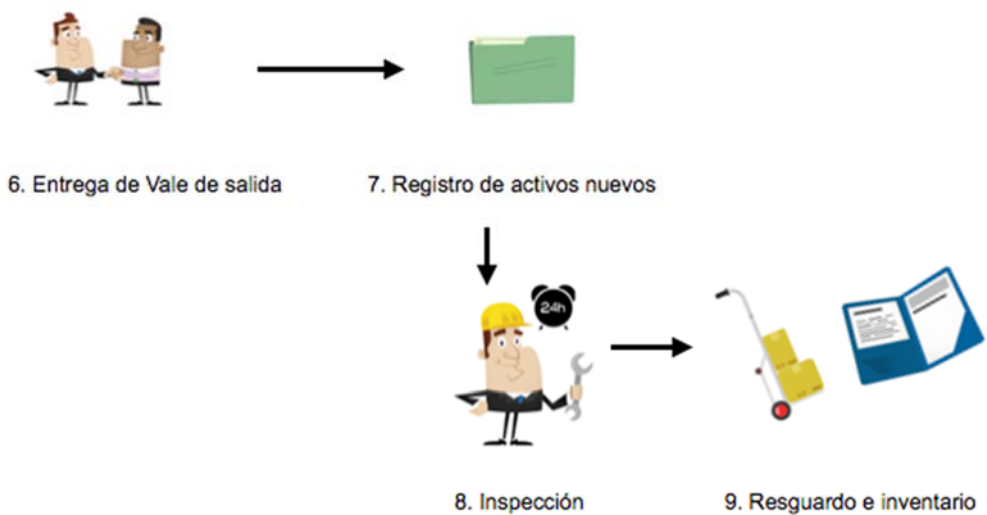


Figura 2.2.2 Proceso de Control de Activos de Hardware

² Alta de Bienes Muebles Nacionales Adquiridos bajo la modalidad de compra directa, por el folio DCI/AC Recuperado de: http://www.patrimonio.unam.mx/patrimonio/ver_Trm_pdf.php?ids=58

6. El Área o Departamento a quién se le asignó el activo envía a la Unidad Administrativa un vale de salida de almacén.
7. La Unidad Administrativa registra los activos nuevos.
8. Se lleva a cabo una verificación periódica.
9. La información es consignada en formatos de Resguardo interno de bienes de activo fijo.

Durante este proceso pueden presentarse algunos cambios. En dado caso, el Departamento de Cómputo registra las modificaciones mediante una solicitud de baja o actualización.

Además de llevar el control del modelo y marca de los diferentes activos de cómputo, la Escuela no lleva un registro de las diferentes características de hardware y software de los equipos. Asimismo, la generación de reportes a través de solicitudes (llenadas a mano o en Microsoft Excel) es una tarea que debe realizarse personalmente.

A continuación, en la figura 2.2.3 se describe el proceso de baja de un equipo a causa de fallas o desperfectos.

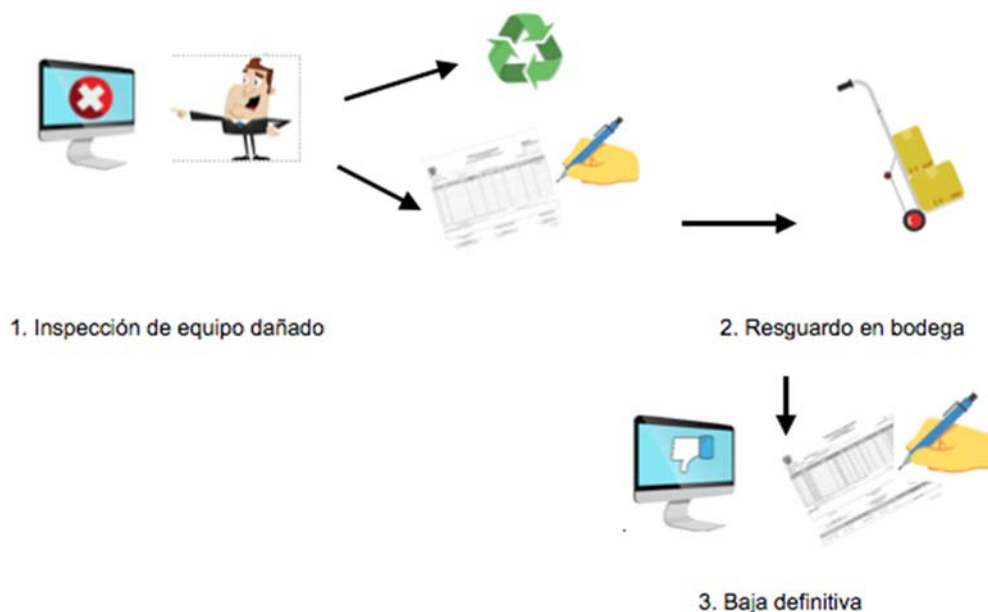


Figura 2.2.3 Proceso de baja de un activo.

10. Cuando un activo sufre alguna falla, el Departamento de Cómputo lo inspecciona. Pueden ocurrir dos situaciones: algunas piezas del equipo dañado se envían para ser reutilizadas o se realiza el cambio mediante una Solicitud de actualización.
11. Si se lleva a cabo un cambio, el equipo es retirado a una bodega en donde es evaluado para confirmar su salida definitiva.
12. El activo es marcado para su baja y retirado. A través del formato de Solicitud de baja se da fin a su vida útil.

La sala de cómputo es el espacio que alberga equipo como computadoras personales, impresoras, periféricos, programas, etc. que el Departamento de Cómputo pone a disposición de los usuarios para la realización de sus tareas y proyectos.

Los activos de hardware de la institución se encuentran clasificados en distintas categorías. Vea la figura 2.2.4.



Figura 2.2.4 Categorías de activos de hardware.

2.3 Análisis de los procesos de registro y seguimiento de solicitudes de soporte técnico de la ENALLT.

El soporte técnico forma parte de un rango de servicios por medio del cual se proporciona asistencia a los usuarios que tienen algún problema en el uso de algún equipo. Cuando el soporte es debidamente organizado, se pueden brindar varios niveles de asistencia: desde un contacto directo que solucione incidencias triviales hasta ayuda especializada.

El personal de la Sala de Cómputo brinda asesoría a los usuarios en el uso y manejo de hardware y software. El usuario puede solicitar asistencia contactando a un miembro del personal de apoyo, o a través de su jefe directo. Se consideran usuarios internos a alumnos, personal académico, estudiantes del curso de formación de profesores, de posgrado o de diplomados, becarios y prestadores de servicio social.

A continuación, se detallan los perfiles de los usuarios comunes que solicitan servicio y pueden reportar algún incidente.

Profesores de carrera.

- Son profesores o investigadores que dedican medio tiempo o tiempo completo en la realización de labores académicas.
- Realizan investigación con fines docentes.
- Contribuyen al desarrollo y actualización de la docencia en su campo.

Profesores de asignatura.

- Son académicos a quienes se les remunera en función de las horas de clase que imparten.
- Su único objetivo es dedicarse a la docencia.

Técnicos académicos.

- Tienen experiencia y aptitudes suficientes en una determinada especialidad, materia o área, para realizar tareas específicas y sistemáticas de los programas académicos y/o servicios técnicos de una dependencia.
- Tienen nombramiento de interinos, definitivos o a contrato de tiempo completo o medio tiempo.

Ayudantes de profesor.

- Auxilian a los profesores e investigadores en su labores.
- Su nombramiento se otorga por un plazo no mayor a un año, y puede renovarse siempre que así lo requieran los planes de la dependencia a la que estén adscritos.

Actualmente, el Departamento de Cómputo no cuenta con protocolos formales para solicitudes de soporte y su correspondiente seguimiento. Sin embargo, si clasifica el nivel de prioridad de una solicitud. Véase la figura 2.3.1

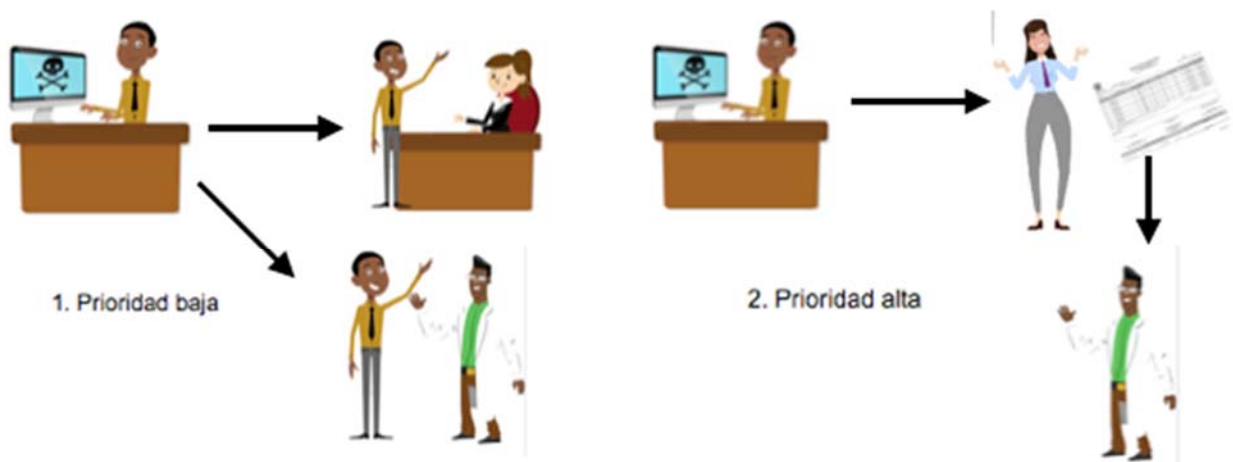


Figura 2.3.1 Niveles de prioridad en una solicitud de soporte técnico

1. Cualquier usuario puede solicitar asistencia de manera directa del Departamento de Cómputo, contactando a un integrante del personal de apoyo o a la secretaria. Este tipo de solicitud se categoriza como prioridad baja.
2. Una solicitud de alta prioridad es cuando el usuario solicita el apoyo a través de su jefe de área y este envía la solicitud al jefe del Departamento de Computo.

Toda notificación llega a un miembro del personal del Departamento de Cómputo. A continuación, en la figura 2.3.2 se describe el proceso de atención de una solicitud.

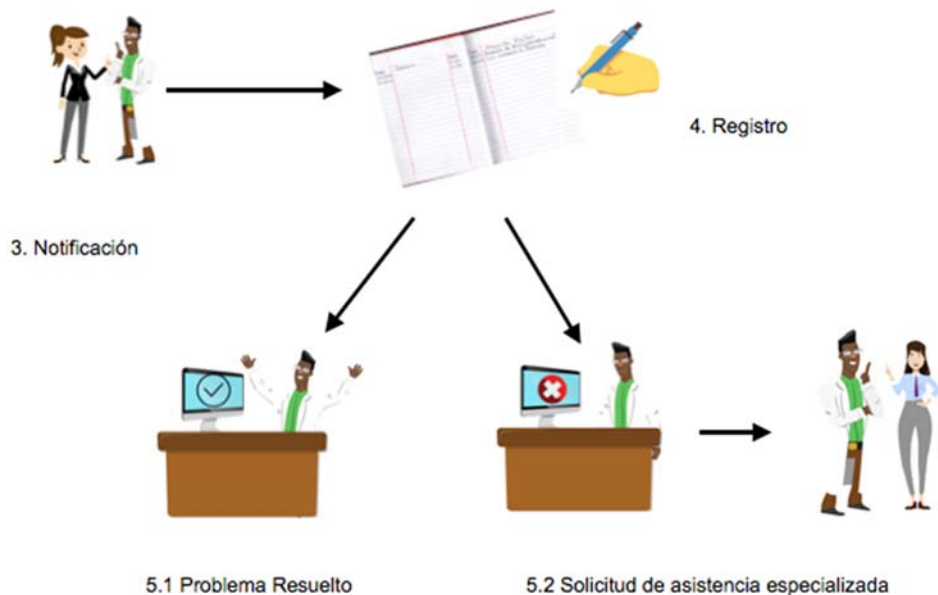


Figura 2.3.2 Proceso de Atención de Solicitud de Soporte Técnico.

3. La secretaria notifica a un miembro del personal del Departamento de Cómputo sobre una solicitud de soporte. La atención es presencial.
4. El personal del Departamento de Cómputo puede o no registrar en una bitácora el reporte. Este proceso no se considera de alta prioridad por parte del personal del Departamento, en parte porque la bitácora no siempre está disponible, y en parte porque consume mucho tiempo.
5. Pueden ocurrir dos situaciones.
 - 5.1 Que el problema se resuelva en el momento de la atención.

5.2 Se requiera de asistencia especializada, en dado caso se notifica al jefe del Departamento de Cómputo.

No se lleva un registro de los tiempos de atención, ya que la bitácora solamente registra la fecha de atención, y no la hora. Además, nunca se registra la fecha y hora en la que quedó corregido el problema. Por último, tampoco se registran todas las posibles atenciones que se pudieran haber brindado, en caso de que el problema haya sido escalado a otros miembros del Departamento de Cómputo; únicamente se registra la persona que atiende la solicitud de soporte por primera vez.

2.4 Levantamiento de requerimientos

Podemos definir requerimiento como el acto de pedir o solicitar algo que obtendremos para un fin. En este punto se presentan los diferentes requerimientos que son necesarios para la realización del sistema de **inventario y atención de solicitudes de soporte**, se obtienen las necesidades generales y particulares de las personas interesadas en el sistema, esta información ayuda para que el sistema de inventario y atención de solicitudes de soporte cumpla con todas las expectativas esperadas. Es necesario tener respaldo de las entrevistas con el encargado de proporcionar la información necesaria y basarnos en los objetivos planteados.

Requerimientos generales.

- El sistema se debe implementar con equipo propio de la institución, y software de código abierto. Esto se debe a que la Escuela no desea invertir recursos financieros en el desarrollo de este sistema.
- Se debe tener acceso por navegador web, donde el usuario pueda acceder desde cualquier parte dentro de la red de la Escuela y por medio de la VPN institucional desde fuera de las instalaciones.
- Como un acuerdo se pretende que el sistema de inventario y solicitudes de soporte funcione con una arquitectura cliente servidor. Las ventajas de esta arquitectura son:

Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor.

Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado.

Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varias computadoras independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor.

- Las bases de datos de los inventarios de equipo de cómputo y solicitudes de soporte tienen relación de tal manera que se pueda tener información en tiempo real de los equipos y su historial de incidentes y requerimientos. Un buen diseño de relación entre módulos de las bases de datos responde a las expectativas de compartir información.
- Los usuarios con acceso son empleados administrativos, de confianza, profesores, personal académico de apoyo e ingenieros de soporte. Es decir, todos los empleados de la Escuela tendrán acceso y dependiendo de sus privilegios podrán hacer uso del sistema. Esto deberá ser complementado con filtrado a través de la dirección IP del cliente.
 - Los alumnos no tendrán acceso al sistema, en caso de necesitar alguna petición la realizarán por medio de la persona indicada, profesor o empleado administrativo.
 - El personal empleado por la institución debe poder levantar solicitudes de soporte, aunque de manera inicial la Escuela solicita que solo el personal del Departamento de Cómputo reciba estos privilegios.
 - Cualquier miembro del Departamento de Cómputo puede realizar modificaciones al sistema: alta de activos de hardware, baja de los mismos, y cambios en cualquier campo de ellos.
 - Personal selecto del Departamento de Cómputo podrá también realizar tareas administrativas del sistema (administración de usuarios y generación de reportes específicos).

- El servidor central se ubica dentro de las instalaciones del edificio, específicamente en el cuarto de servidores.
- El sistema deberá ser capaz de tener los límites necesarios para la navegación para los usuarios de acuerdo con su perfil de usuario.
- El sistema debe poder generar reportes con la información contenida en las bases de datos de inventario y solicitudes de soporte. Los reportes solicitados son los siguientes:
 - Una lista de cada tipo de activo de hardware (CPUs, monitores, etcétera).
 - Una lista de los equipos (CPUs, laptops, y equipos todo en uno) con información específica: ubicación del activo, usuario asignado, sistema operativo instalado, dirección IP homolada asignada (en su caso), marca y modelo.
 - Una lista de los equipos que existen en cada área física de la Escuela.
 - Una lista de las solicitudes de soporte que cada técnico de soporte tiene asignadas y no ha atendido.
 - Una lista de todas las solicitudes que ese usuario ya ha atendido.
 - Una lista de solicitudes de soporte recientes por área, usuario atendido, y activo de hardware.
 - Una lista de todas las solicitudes de soporte de la Escuela.
- Los campos deben contar con soluciones de autocompletado para agilizar el proceso de llenado de los formularios.
- Cada solicitud de soporte deberá contar con el registro de la fecha en que fue levantada, así como la fecha de cada atención. Se deberá llevar un registro de cada atención, en caso de que la solicitud de soporte requiera escalamiento a otro personal de soporte.
- El sistema debe validar que los activos de hardware cuenten con un número de inventario UNAM único.
- El sistema debe reflejar la interacción entre los diferentes activos de hardware. Se necesita saber, por ejemplo, que un monitor está conectado a un CPU específico. Esto se extiende a conexiones con los equipos de energía (e.g. reguladores), pero

no a los equipos de red, aunque sí se solicita saber la dirección IP homologada asignada si así es el caso

- El sistema debe contar con soporte para introducir la información de compra de los activos (proveedor, factura) aunque esta información no será agregada en este momento.

2.5 Alcance y acotamiento del problema.

Debido a la necesidad de llevar el control de inventario de los activos de cómputo (hardware) de la Escuela Nacional de Lenguas, Lingüística y traducción (ENALLT) surge la necesidad de mencionar los requisitos y características del sistema a implementar. Los tipos de activos de hardware que la Escuela desea administrar son los siguientes:

- Monitores
- Teclados
- Mouses
- CPU (Unidad de procesamiento central)
- Impresoras
- Discos Duros externos
- Discos duros en red
- Routers
- Switches
- Respaldos de energía
- Tablet digitalizadoras
- Cámaras digitales
- Proyector
- Bocinas

Un ejemplo de los cuales se puede apreciar en la figura 1.5.1.



Figura 2.5.1. Ejemplo ilustrativo de activos de hardware que se administrarán.

La siguiente información es considerada por la ENALLT como necesaria para cualquier tipo de activo de hardware:

- Tipo de equipo.
- Número de serie del equipo.
- Número de inventario de la UNAM.
- Nombre del usuario que fue asignado para usar el equipo.
- Ubicación del equipo.
- Se encuentra activo o inactivo.

Con los mismos recursos y compartiendo los datos del sistema de inventarios se trabajará en el módulo de registro de solicitudes de soporte. De esta forma podemos saber de una mejor manera la relación del usuario con los equipos de cómputo reportado y dar seguimiento hasta el cierre del mismo. La referencia principal es el identificador

(ID) que es el número de solicitud de soporte (conocido también como *ticket*). Cada ticket es guardado y funcionará como referencia para otros casos.

Otros campos de información que han sido solicitados por la ENALLT son la fecha de registro de cada evento, tanto de inicio de solicitud como de cierre, así como notas generales sobre la solución del incidente.

De manera general el sistema debe identificar a un usuario por el número de inventario o número de serie de su equipo, así como por su nombre; para tareas de control, también debe poder generar métricas de tiempos de atención, número de incidencias por equipo, entre otras.

Específicamente, el sistema:

- Da diferentes privilegios de acceso a la información de acuerdo a las necesidades de los usuarios, técnicos y colaboradores de la ENALLT dependiendo su cargo.
- Trabaja con 2 módulos que interactuarán. Estos son los módulos de inventario y reporte de incidencias.
- Utiliza recursos propios de la institución y software de código abierto.
- Es compartido en la intranet propia del instituto.
- Genera métricas del número de solicitudes de soporte por activo de hardware o usuario atendido, así como por técnico de soporte; también genera métricas de los activos de hardware con base en sus características.
- Permite levantar solicitudes con el fin de proporcionar soporte técnico.

- Tiene un sistema confiable en las búsquedas y reportes requeridos del inventario y casos generados.

2.6 Planteamiento de la solución.

Después de tener los requerimientos y con base en análisis expuesto, necesitamos proponer la solución a todas las áreas y personal del instituto involucrada en el proyecto. Primero nos vemos en la necesidad de escoger el software necesario para trabajar y desarrollar el sistema de inventario y ticketing.

Considerando las características de nuestro servidor Linux, y revisando la tabla 2.6.1. **Apache 2.4** será nuestro servidor, las condiciones de rendimiento son similares a los otros productos.

Servidor Web	Apache	nginx	ISS
Software Libre	Sí	Sí	No
Licencia/costo	No	No	No
Conexiones permanentes	Sí	Sí	Sí
Soporte Virtual	Sí	Sí	Sí
Escala a Servidor SMP	Sí	Sí	Sí

Tabla 2.6.1. Comparativa de servidores web

Se evaluaron 3 lenguajes de programación para web: PHP, Ruby y Python. De acuerdo a la siguiente tabla 2.6.2, las 2 características que pesan en nuestra decisión se refieren al software libre y la complejidad de aprendizaje, Python 2.7.9 será el lenguaje de programación con el que trabajaremos. Las otras características son similares en los otros tipos de software, también funcionales para poder trabajar. La facilidad que tiene Python en la lectura de código y la documentación en caso de necesitarla son otros puntos a su favor.

Lenguaje programación	Phyton	PHP	Ruby
Complejidad de aprendizaje	Sencillo	Medio	Medio
Software libre	Sí	Sí	Sí
Velocidad de ejecución	Veloz	Veloz	Velocidad media
Documentación a usuario	Muy bueno	Muy bueno	Bueno
Uso de paginas dinámicas	Sí	Sí	Sí

Tabla 2.6.2. Comparativa de lenguajes de procesamiento del lado del servidor

Para apoyar al software web en el manejo rápido buscamos un framework de aplicaciones que nos permita el desarrollo de sitios web dinámicos, web services (servicios web) y aplicaciones web. El propósito de este tipo de framework es permitir construir aplicaciones web y centrarse en los aspectos interesantes, mejorando la típica tarea repetitiva asociada con patrones comunes de desarrollo web. La mayoría de los frameworks de aplicaciones web proporcionan los tipos de funcionalidad básica común, tales como sistemas de templates (plantillas), manejo de sesiones de usuario, interfaces comunes con el disco o el almacenamiento en base de datos de contenido *cache*, y persistencia de datos. Normalmente, los frameworks de aplicación web además promueven la reutilización y conectividad de los componentes, así como la reutilización de código, y la implementación de bibliotecas para el acceso a base de datos.

Dentro de los manejadores de Bases de Datos relacionales la elegida fue **MySQL 5.5.54**, quien tiene las propiedades suficientes para poder manejar el intercambio de información. La realidad es que no necesitamos un manejador tan robusto como Oracle ni pagar por él.

FRAMEWORK	CARACTERÍSTICAS
Angular js	Un framework basado en <i>JavaScript</i>
React	Liberado por Facebook, en <i>JavaScript</i> , permite desarrollar aplicaciones móviles para IOS y Android
Ionic	Para móviles, usando HTML, Js, Sass y Angular
Meteor	En <i>JavaScript</i> , para web y móviles
Ruby on Rail	Framework MVC basado en Ruby, orientado al desarrollo de aplicaciones web
Django	Framework Python que promueve el desarrollo rápido y el diseño limpio, software libre.
CakePHP	Framework MVC para PHP de desarrollo rápido
Zend	Framework para PHP 5, simple, claro y open-source
Catalyst	Framework para aplicaciones web MVC elegante

Tabla 2.6.3. Comparativa de frameworks para desarrollo web

Base de datos	MySQL	Postgre SQL	ORACLE
Costo-Mantenimiento	Disponible sin costo	Sin costo	Muy costoso
Licenciamiento	Sin costo	Sin costo	Muy costoso
Sistema de gestión	Relacional	Objeto-relación	Objeto-reacción
Multithread	Sí	Sí	Sí
Software libre	Si	Sí	No
Manejo de grandes volúmenes de información	Sí, pero limitado	Si, pero limitado	Sí, muy alto
Velocidad transaccional	Velocidad limitada	Relativamente lenta	Muy eficiente
Escalabilidad	Sí, muy alta	Sí	Sí
Subconsultas	Sí, limitada	Sí	Sí
Seguridad	Menor en la versión libre	Limitada	Robusta
Recuperación de desastres	Limitada	Sí	Sí
Configuración e instalación	Menos compleja	Menos compleja	Compleja

Tabla 2.6.4. Comparativa de motores de bases de datos (DBMS)

Hardware.

El instituto cuenta ya con infraestructura funcionando donde se instalaría el software para la construcción del sistema. El servidor físico es marca Dell PowerEdge R820 con las siguientes características:

- 2 procesadores Intel Xeon E5-4610V2.
- 64GB en RAM.
- 6TB en Disco Duro (RAID 1).
- Fuente redundante de poder.
- 8 tarjetas de red.
- Sistema operativo VMware ESXi 6.0

Este servidor físico tiene 30 servidores virtuales. Uno de estos servidores se usará para el proyecto de inventario y ticketing. El servidor virtual tiene lo siguiente.

384MB RAM

25GB en Disco Duro.

Sistema operativo Debian/Linux 8.

El diseño funcional se muestra en la figura 2.6.5, donde aparecen los componentes principales y su interacción.

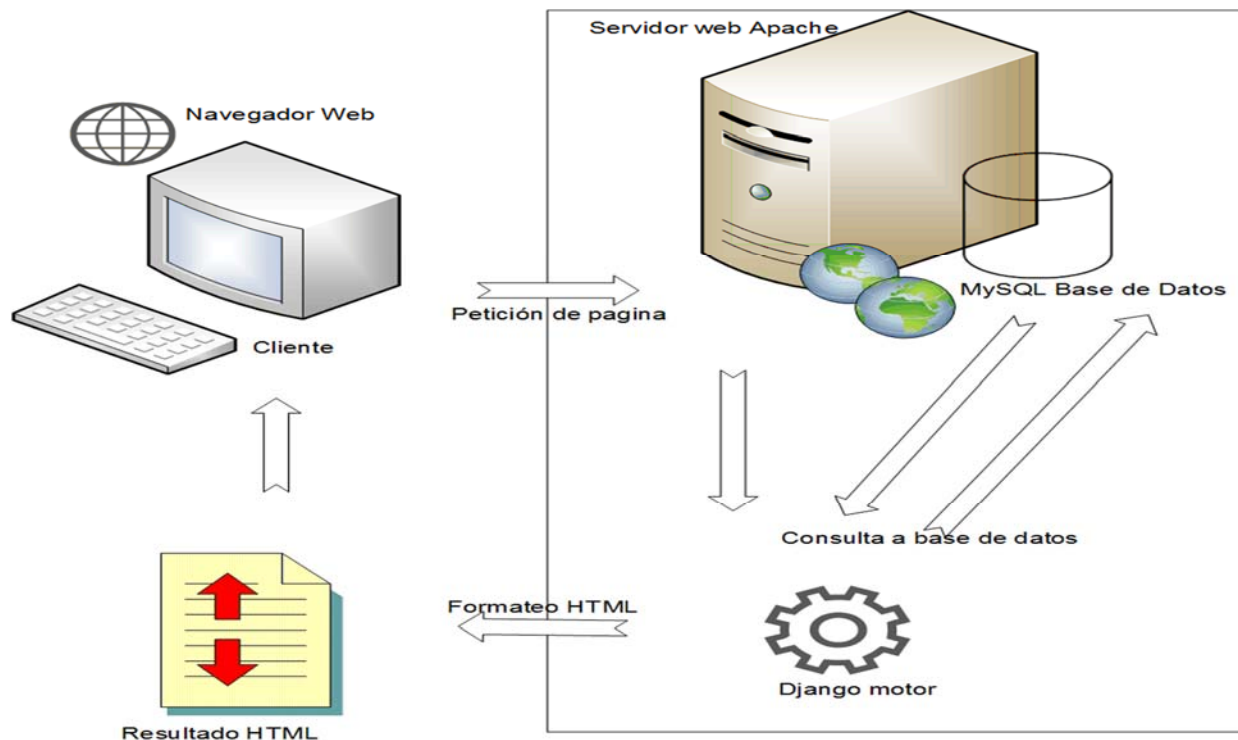


Figura 2.6.5. Diseño funcional.

Proponemos el desarrollo del sistema utilizando una arquitectura de tres capas:

Interfaz de usuario. El objetivo de una interfaz de usuario es que este pueda comunicar información a través de ella hacia algún tipo de dispositivo o sistema. Una vez se logre esta comunicación el siguiente objetivo es el que dicha comunicación se desarrolle de la forma más fácil y cómoda posible para las características del usuario que utiliza el servicio.

Bases de Datos. O también llamadas Bancos de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su uso posterior. En la actualidad, y debido al desarrollo tecnológico, la mayoría de las bases de datos están en formato digital que ofrece un amplio rango de soluciones al problema de almacenar datos.

Lógica de negocio es la parte de un sistema que se encarga de codificar las reglas del proceso en cuestión, que determinan cómo la información puede ser creada, mostrada y cambiada.

Los siguientes módulos deberán estar presentes en las bases de datos de inventarios:

- Proveedor.
Nombre, Número de etiqueta UNAM.
- Modelo
Nombre
- Tipo de procesador
Nombre
- Tipos de sistema operativo
Nombre
- Clase de dispositivo de almacenamiento
Nombre.
- Tipo de impresora
Nombre, marca.
- Tipo de scanner
Nombre, marca.
- Ubicación
Nombre
- Tipos de usuario
Nombre
- Fecha de expiración de garantía.
Nombre
- Nombre del proveedor
Nombre.
- Sistema operativo
Nombre
- Marcas de regulador
Nombre, modelo, marca.
- Marcas de equipo

Nombre, modelo.

- Marcas de dispositivo de almacenamiento
Factura, ubicación, modelo, serial. Usuario.
- Marcas de dispositivo de telecomunicaciones
Factura, ubicación, modelo, serial. Usuario.
- Marcas de monitor
Factura, ubicación, modelo, serial. Usuario.
- Marcas de teclado
Factura, ubicación, modelo, serial. Usuario.
- Marcas de mouse.
Factura, ubicación, modelo, serial. Usuario.
- Marcas de tableta digital.
Factura, ubicación, modelo, serial. Usuario.
- Marcas de cámara.
Factura, ubicación, modelo, serial. Usuario.
- Marcas de bocinas
Factura, ubicación, modelo, serial. Usuario.
- Marcas de scanner.
Factura, ubicación, modelo, serial. Usuario.
- Marcas de proyector.
Factura, ubicación, modelo, serial. Usuario.
- Marcas de impresora
Factura, ubicación, modelo, serial.
- Procesador.
Nombre
- Tipos de memoria RAM
Nombre
- Antivirus
Nombre.
- Utiliza licencia del Instituto
Nombre

- Paquetes de software de oficina.
Nombre
- Tamaña disco duro.
Nombre
- Marca Discos duros.
Factura, ubicación, modelo, serial.
- IP del dispositivo.
Inventario, tipo, ubicación, usuario, nombre.

Para la interfaz de solicitudes de soporte, esto son los módulos que se tendrán que determinar:

- Número de ticket.
Número
- Personal de soporte asignado a los casos.
Nombre
- Seguimiento.
Nombre
- Número de etiqueta del instituto.
Nombre
- Escalación.
Nombre.

2.7 Diseño de la arquitectura del sistema

En esta sección utilizaremos diagramas para describir la arquitectura del sistema de manera general.

Como todo sistema implementado sobre Django, la arquitectura general del sistema se puede apreciar en la figura 2.7.1:

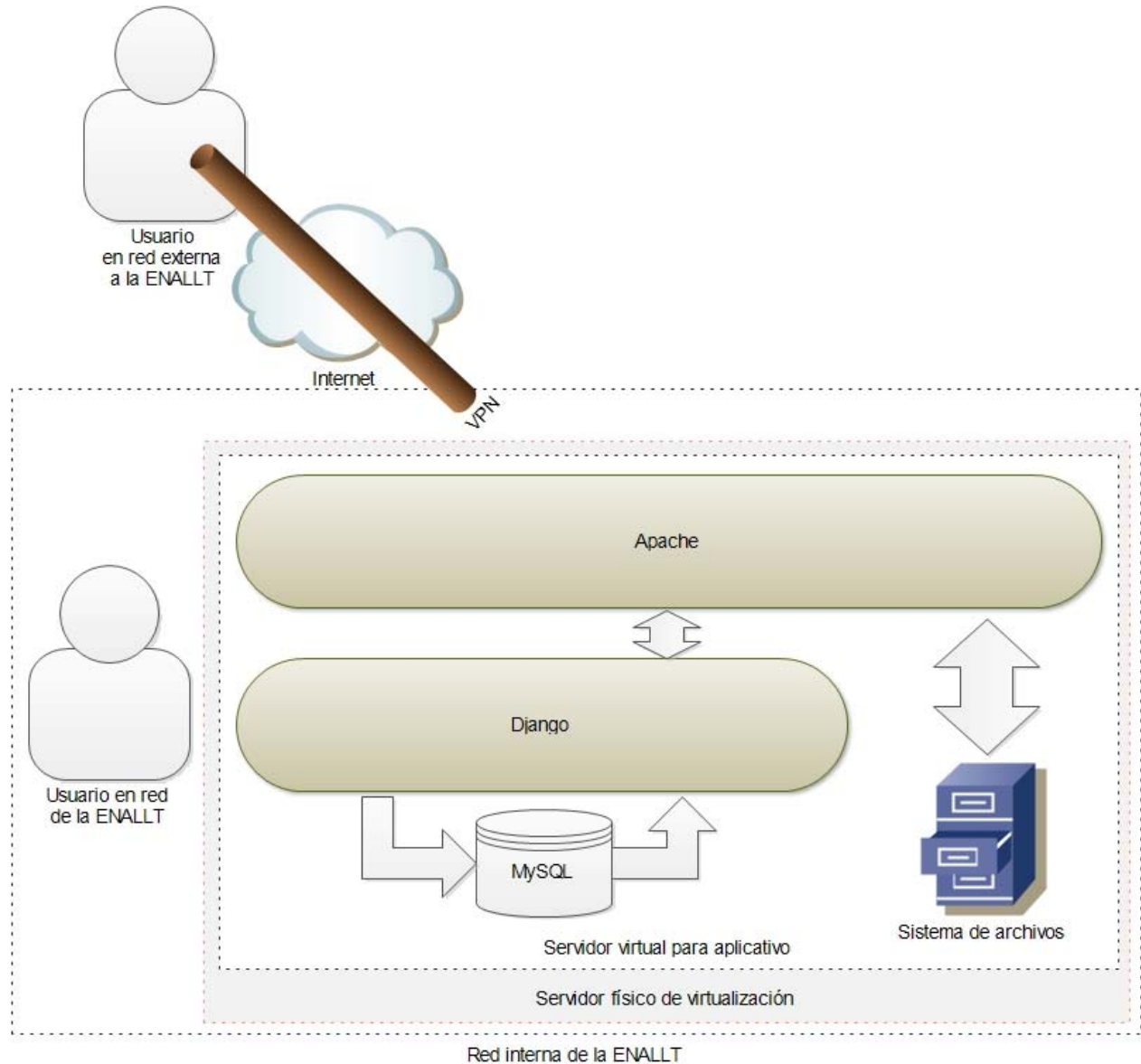


Figura 2.7.1. Arquitectura de alto nivel del sistema.

La arquitectura del servidor virtual es similar a cualquier servidor **LAMP** (**L**inux/**A**pache/**M**ySQL/**p**hp) en concepto: un servidor web (Apache, en este caso)

sobre un sistema operativo Linux, utilizando un motor de base de datos MySQL, y con un lenguaje de scripting procesando peticiones del lado del servidor (en este caso, python toma el lugar de php, a través del framework Django). Este esquema permite establecer una infraestructura de cliente-servidor a través de HTTP de manera rápida y eficiente, aprovechando la confiabilidad y estabilidad de estas plataformas. El estándar HTTP se describe en la figura 2.7.2:

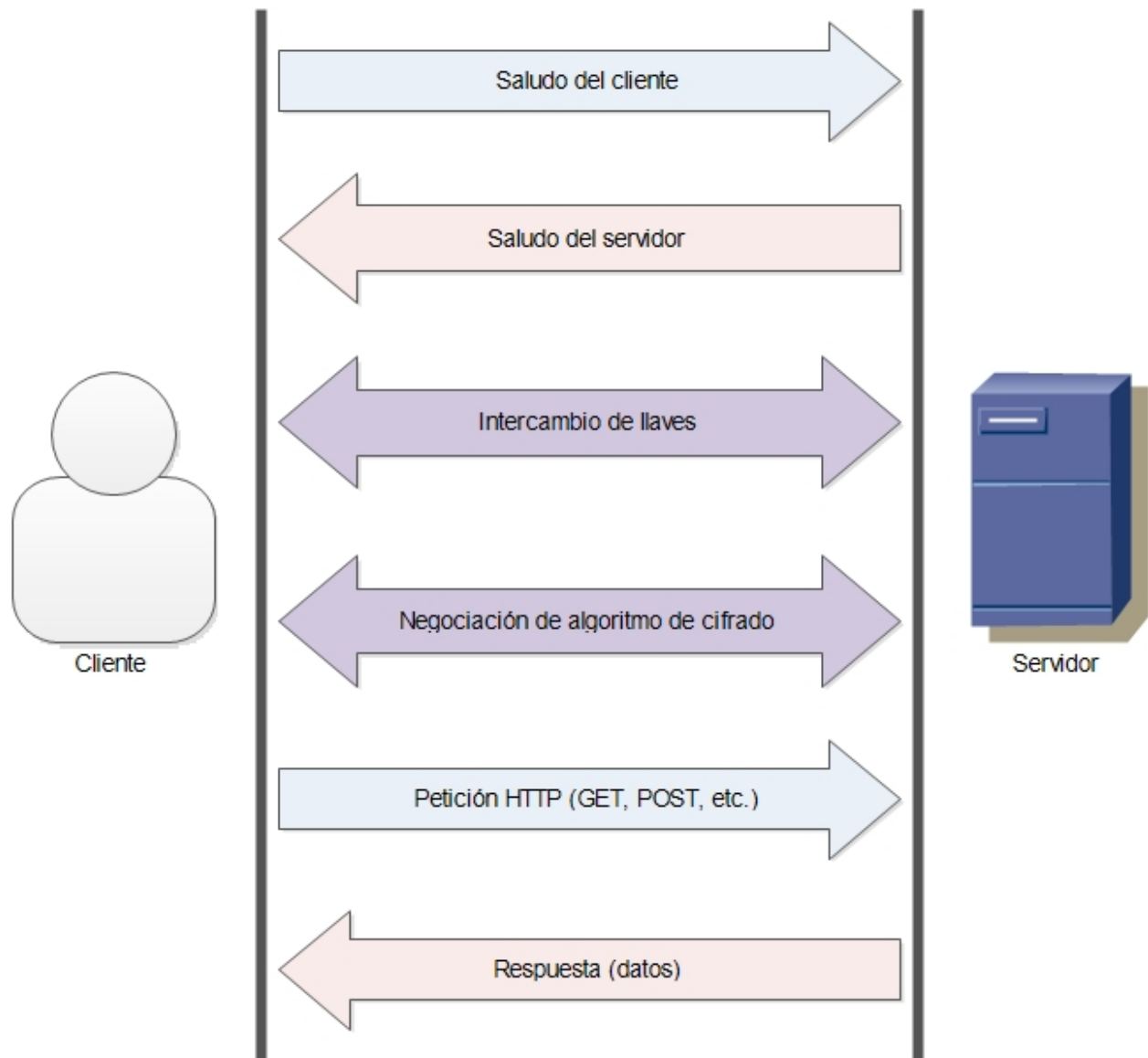


Figura 2.7.2. Arquitectura cliente-servidor general de HTTP.

Después de un **three way handshake (saludo de tres pasos³)** que en este caso abreviamos como saludo de cliente y saludo de servidor, se realiza un posible proceso de intercambio de llaves y negociación de algoritmo de cifrado, en caso de tener habilitada la comunicación a través de SSL (**Secure Socket Layer, Capa de Conexión Segura**). Este proceso es completamente opcional, aunque altamente recomendado, en la comunicación de http⁴. Para propósitos de este documento, dejaremos de lado la descripción detallada del funcionamiento de HTTPS (HTTP sobre SSL).

En cualquier caso, una vez terminado estos procesos, el cliente realiza una petición de tipo HTTP. Por ejemplo, si queremos la página de inicio de un sitio web, emitiríamos una petición del siguiente tipo:

```
GET / HTTP/1.0
User-Agent: Navegador/1.0
```

Como respuesta a estas peticiones, el servidor nos entregaría una respuesta apropiada. La manera en la que se determina esta respuesta depende de varios factores: si el recurso solicitado existe y el servidor http tiene privilegios de acceso al mismo (dentro del contexto del sistema operativo del servidor), si el cliente tiene privilegios de acceso al mismo (a través de políticas definidas en la configuración del sitio o servidor web), etc. Asumiendo que estos factores se cumplen, recibiríamos los datos que componen al recurso solicitado. Por ejemplo:

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354

<html>
<body>
<h1>Contenido web</h1>
(contenido adicional)
.
.
.
</body>
</html>
```

³ El three way handshake es la manera en que se inicia cualquier comunicación sobre TCP, lo que incluye al estándar de HTTP. De <https://tools.ietf.org/html/rfc793>, recolectado el 16 de Septiembre de 2017.

⁴ <https://developers.google.com/web/fundamentals/security/encrypt-in-transit/why-https>. Recolectado el 16 de Septiembre de 2017

En el caso de requerir un recurso procesado por un lenguaje de scripting (por ejemplo, php), el recurso no es juzgado únicamente en el contexto del sistema de archivos del servidor, sino que además, una vez ubicado y determinados los privilegios de manera exitosa, el archivo es procesado por el lenguaje en cuestión y la respuesta que entrega el servidor web es precisamente aquella que es entregada a él por el procesamiento del script.

En el caso de Django, el procesamiento no depende únicamente del script que atenderá la petición, sino también de otros factores dentro de la configuración de Django. En efecto, Django actúa como un servidor web *dentro* de otro servidor web. Un esquema del funcionamiento de Django se presenta en la figura 2.7.3:

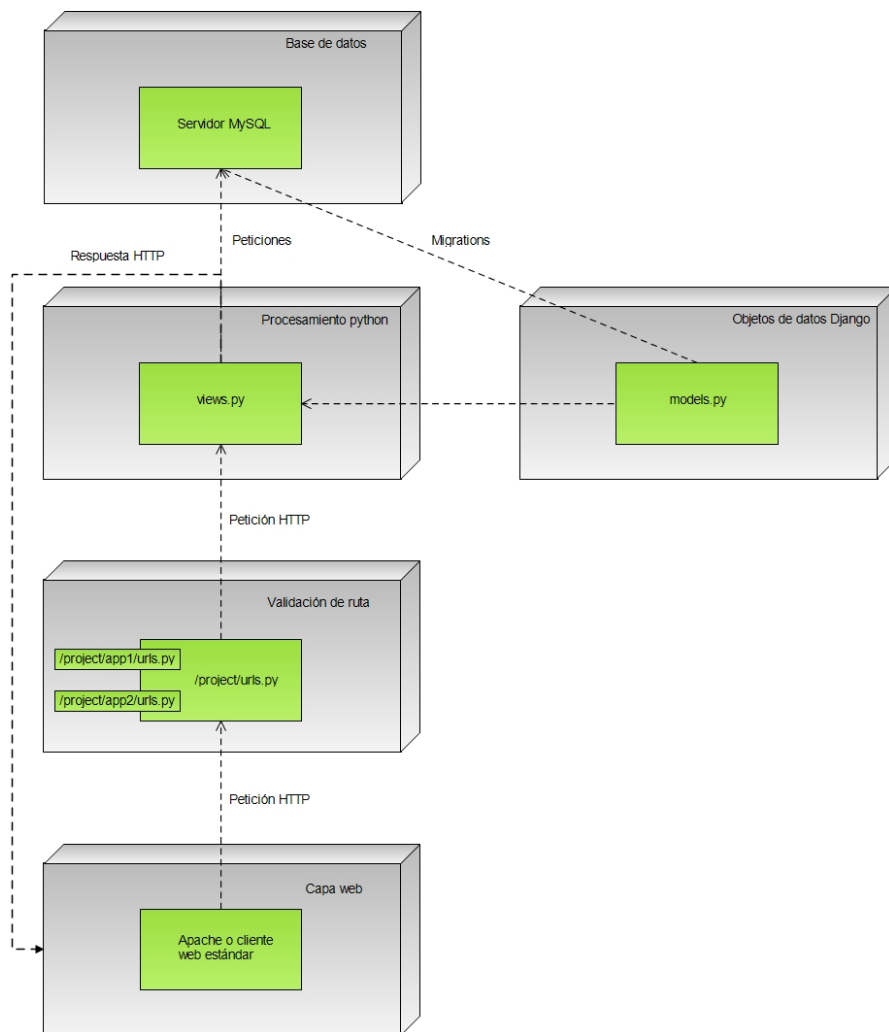


Figura 2.7.3. Arquitectura del framework Django.

Una plataforma de Django se compone de una o más aplicaciones (abreviadas como *apps*) dentro de un proyecto. Todas las apps dentro del mismo proyecto comparten la misma arquitectura, archivos de control, etc.

Django recibe la petición HTTP de manera íntegra, sin modificaciones⁵, para realizar su procesamiento. Esta petición incluye, entre otras cosas, el recurso que se está solicitando, y el primer paso de Django es validar si la ruta solicitada por este recurso está dentro de sus rutas permitidas. Esto se hace procesando el archivo `urls.py`. Existe un archivo “maestro”, correspondiente al proyecto, y pueden existir archivos `urls.py` adicionales, para cada app (o en la medida que el desarrollador considere conveniente). Como en la mayoría de las configuraciones de Django, el nombre de `urls.py` es simplemente una convención y no se requiere conservar este nombre para el correcto funcionamiento de la plataforma, si se mantiene una consistencia interna en las referencias.

Si la ruta es válida, el mismo archivo `urls.py` indica la rutina de Python que deberá hacerse cargo del procesamiento. Esta rutina se encuentra en alguno de los archivos `views.py`. En el contexto de Django, una **vista** es aquella rutina que nos permite procesar una URL.

Estas rutinas suelen hacer uso de uno o más **objetos de datos** definidos en Django, a través del archivo `models.py`. Este archivo permite la definición de la base de datos a través de clases de Python, y estas clases a las que hacemos referencia cuando procesamos información de la base de datos en alguna vista. El archivo de `models.py` será discutido a mayor detalle en la sección 4.2, *Diseño, construcción e implementación de la base de datos*. Cuando un archivo `views.py` hace alguna solicitud a la base de datos, lo hace a través de métodos propios de Django que permiten que el conocimiento del motor de base de datos sea innecesario, creando abstracciones sobre el lenguaje Python.

⁵ Dependiendo de la configuración de instalación de Django, es posible que se realicen modificaciones a ciertos metadatos dentro de la petición. Por ejemplo, si Django se configura a través de gunicorn, veremos diferentes valores para la cabecera `Remote_ADDR`. Ver <https://serverfault.com/questions/847141/how-can-i-get-the-real-remote-ip-at-gunicorn-from-nginx>, recuperado el 16 de Septiembre de 2017.

Finalmente, el archivo `views.py` debe producir una respuesta de tipo HTTP para entregar a la capa web. Esta respuesta puede variar, al igual que en un script de otro lenguaje, con base en condiciones programáticas.

La arquitectura de virtualización permite minimizar el uso de recursos de hardware dedicados a este sistema, sin sacrificar su segregación de otros recursos críticos, además de brindar ventajas inherentes a la plataforma, como respaldos periódicos, redundancia de recursos, etc. Sin embargo, esta plataforma ya está implementada y no se cubrirá a detalle en este documento.

De igual manera, la Escuela ya cuenta con una solución de VPN para proveer a su personal con acceso a su red interna desde ubicaciones externas a la misma, conservando sus estándares de seguridad. Al estar ya implementada y no representar algún cambio para el desarrollo de esta solución, no será cubierta a detalle en este documento.

La infraestructura de red de la Escuela también se encuentra ya implementada, y se menciona en la figura 2.7.4 exclusivamente para propósitos informativos.

El servidor físico donde se alojará el servidor virtual se encuentra en el site principal de la Escuela, y utiliza nodos de red directamente del Catalyst, brindándole la mejor conectividad posible. Además, cuenta con redundancia y balanceo de carga de tráfico de red a través de ocho interfaces, optimizando el desempeño de cualquier aplicación basada en red.

Recordando que Django permite tener diferentes apps dentro de un proyecto, se propone una solución donde se conjuguen ambas propuestas (para control de activos de hardware y para control de solicitudes de soporte) como apps separadas dentro de un único proyecto. Además de permitir una administración más eficiente, podrían compartir objetos de datos entre ellos, algo que se proyecta como un requerimiento considerando que las solicitudes de soporte pueden llegar a involucrar activos de hardware.

Una breve descripción de la arquitectura del sistema a nivel de software se puede apreciar en la figura 2.7.5.

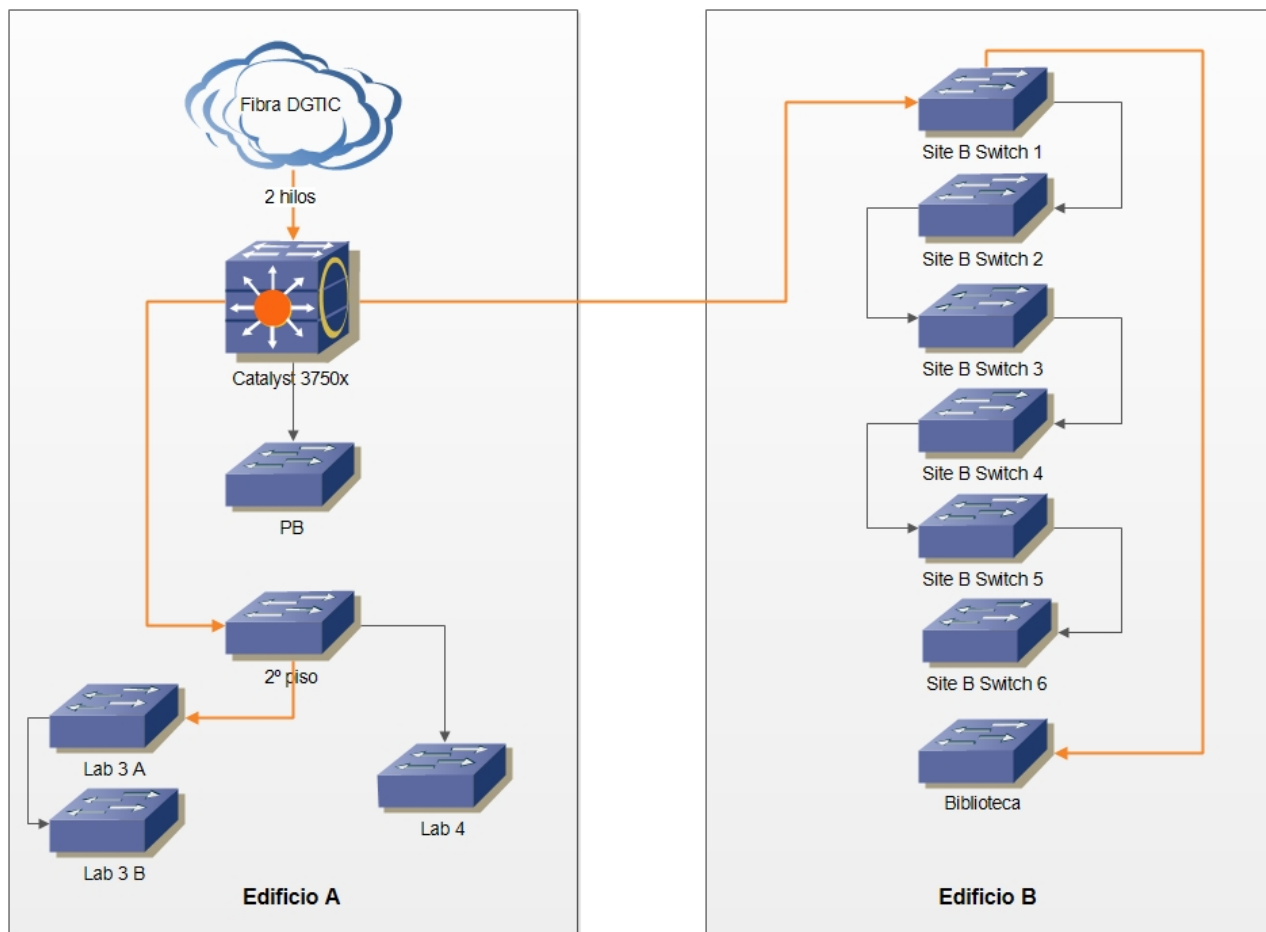


Figura 2.7.4. Infraestructura de red de la ENALLT.

Django crea de manera automática un módulo de administrador, que puede ser habilitado a través de una línea de configuración. Este módulo provee interfaces prediseñadas, que permiten realizar funcionalidades básicas (alta, baja, cambios y listados) así como la administración de los usuarios del sistema. Aunque este panel no es suficiente para la mayoría de los proyectos de Django, es conveniente para la administración de algunas tareas básicas, y se sugiere que se habilite en cualquier caso. Como medida adicional de seguridad, podemos restringir el acceso a este panel administrativo a través de directivas *Require* de Apache 2.4, basado en IP o a través de autenticación, según sean las necesidades de la Escuela.

El módulo de usuario consistirá en la programación de las vistas que la Escuela considere necesarias. De manera preliminar, se sugiere contar con las dos apps (control de activos de hardware, abreviada *inventario*, y control de solicitudes de soporte, abreviada *ticket*).

En el caso de inventario, se prevén vistas de alta de activos de hardware, de control de cambios a los mismos (incluyendo el caso de baja de los activos), y de reportes según sea la necesidad expresada por la Escuela. En el caso de ticket, se prevén vistas para el registro de nuevas solicitudes, para el seguimiento de solicitudes existentes, y para el escalamiento o cierre de las solicitudes que pudieran requerirlo.

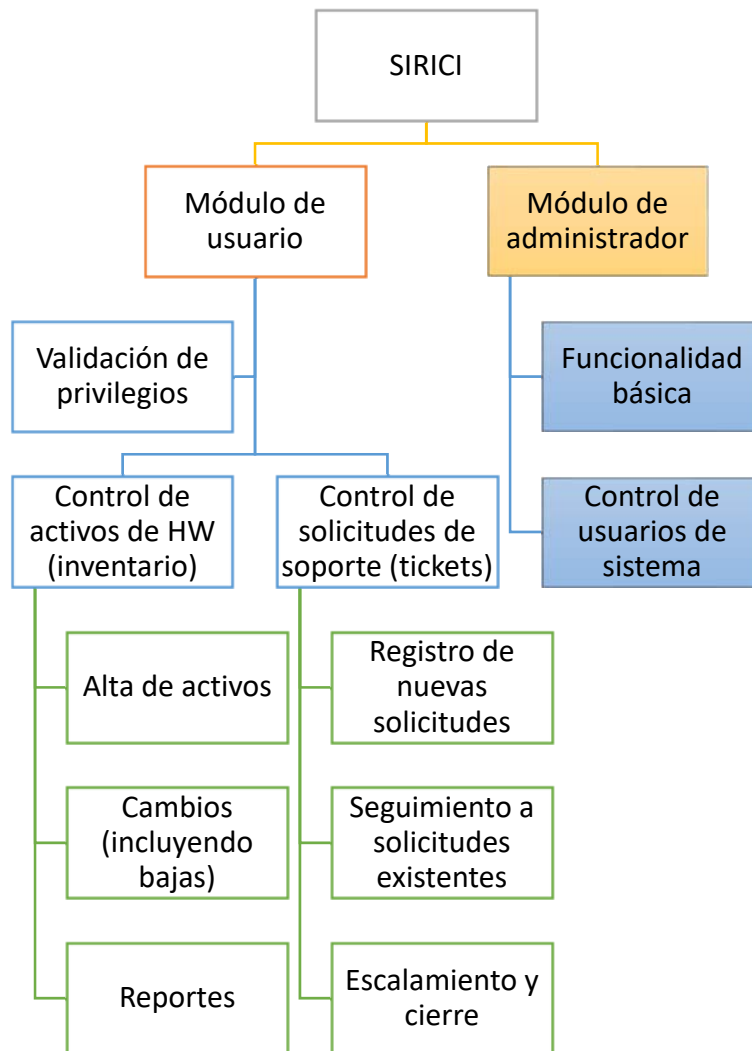


Figura 2.7.5. Arquitectura de alto nivel de la solución de software.

CAPÍTULO III. MARCO TEÓRICO

3.1 Características, ventajas y desventajas de las bases de datos relacionales y el manejador de base de datos MySQL.

Una base de datos relacional surge de la necesidad de tener un modelo de base de datos menos complejo. Éstas se componen de varias tablas o relaciones. No puede tener dos tablas con el mismo nombre, ni registro. Cada tabla es a su vez un conjunto de registros (filas y columnas). La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas). Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos. Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

De manera sintetizada una base de datos relacional es una colección de datos organizados en tablas de renglones y columnas, relacionadas por campos que son llave.

Para tener un modelo relacional en una base de datos se necesita que en la fase de diseño se tomen en cuenta las características que definen a un modelo de datos relacional. Las características relevantes que se deben tomar en cuenta son:

- Una base de datos relacional se compone de varias tablas o relaciones.
- Independencia física y lógica de los datos.
- No pueden existir dos o más tablas con el mismo nombre o registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y foráneas (ajenas).
- La información debe ser consistente, es decir, no debe haber incongruencias en los datos que se almacenan.
- No debe haber redundancia en la información. Se debe de eliminar al máximo la existencia de duplicidad de los datos.
- Cada campo de un registro tiene uno y sólo un valor.

- Las claves primarias son un identificador único.
- Las claves primarias son la clave principal de un registro en una tabla y éstas deben cumplir con la integridad de datos.
- Las claves foráneas se encuentran en la tabla hija, contienen el mismo valor que la clave primaria del registro padre y por medio de éstas se hacen las relaciones.
- Hay que asegurar que sólo los usuarios que estén autorizados para acceder a los datos puedan hacerlo en cualquier momento. Que la información sea accesible en cualquier momento.
- Los datos deben ser íntegros. Es decir los datos tendrán que ser consistentes dentro de la base, además los valores que posean los datos, deben ser válidos de acuerdo a las dependencias funcionales entre tablas y de acuerdo a las políticas de negocio. La integridad de la base se puede lograr por las siguientes afirmaciones:
 - El establecimiento de llaves primarias.
 - El mantenimiento de una redundancia mínima y controlada.
 - La validación de dependencias entre las tablas relacionadas.
 - La concepción de reglas de validación durante la inserción y edición de datos.

Ventajas del modelo de base de datos relacional.

- Los datos son compartidos por diferentes usuarios y aplicaciones.
- Los usuarios finales como las aplicaciones no necesitan conocer los detalles de las estructuras de almacenamiento.
- Podemos acceder a los datos por medio de llaves.
- Es más comprensible y aplicable que otros modelos de Bases de Datos.
- Podemos agregar posteriormente datos al sistema sin tener que reescribir lo que se tiene.

- Los datos y procedimientos pueden ser transportables conceptualmente a través de diferentes sistemas de Gestión de Bases de Datos.
- Es muy eficiente ya que no tendremos duplicación de datos y tampoco tendremos grandes cantidades de “celdas vacías”. De esta manera, no se tienen que almacenar más datos de los necesarios, ni gastar recursos al revisar las áreas vacías en nuestras tablas.
- Al eliminar un registro se eliminan todos los registros dependientemente relacionados.
- Es simple al momento de la planeación y la esquematización de la base de datos.
- Fácil uso para el usuario final.
- Hay un control sobre la redundancia de datos.
- La creación de esquemas entidad-relación es relativamente corta.
- Mejora la concurrencia al poder dar acceso a varios usuarios al mismo tiempo.
- Soporte más flexible para tipos de datos complejos.
- Integridad de datos mediante las restricciones o reglas que se establezcan.
- La consulta de información se hace más sencilla y rápidamente.
- El modelo de base de datos relacional está creciendo, sus estándares son conocidos y utilizados por los fabricantes de bases de datos y desarrolladores.

Desventajas del modelo de base de datos relacional.

- Los bloques de texto no se manipulan de manera sencilla. Hay que tener conocimiento de esto para no caer en la complicación.
- La creación y mantenimiento del sistema implica un costo en software y mantenimiento, dependiendo del tamaño es su complejidad.
- Los sistemas complejos de bases de datos relacionales pueden conducir a que estas bases de datos se conviertan en información aislada, donde la información no puede ser compartida fácilmente entre sistemas.
- Existen deficiencias con datos gráficos, multimedia y sistemas de información geográfica.

- El acceso a los datos puede llegar a ser lento, dependiendo del tamaño de la información.
- La seguridad de la información también puede llegar a ser un problema, ya que al ser un sistema concurrente, la información se compromete y de no tener reglas estrictas y específicas para todas las posibles eventualidades se puede llegar a tener problemas de seguridad.

Manejador de base de datos MySQL.

MySQL apareció en la década de los 90s. No se sabe el origen del nombre MySQL de donde proviene, se dice que en su biblioteca han llevado el prefijo “my” en los diez últimos años, por otra parte, la hija de uno de los desarrolladores se llama My. Así que no está claramente definido.

El logo de MySQL se aprecia en la figura 3.1.1.



Figura 3.1.1. Logo de MySQL

El sistema de gestión de base de datos relacional MySQL fue creado en 1996 y sigue siendo actualizado en los últimos años por MYSQL AB, una compañía sueca. Inicialmente fue concebido porque se necesitaba una base de datos relacional capaz de manejar grandes cantidades de datos en equipos de bajo costo. El nombre del delfín de MySQL es "Sakila".

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo

incorporados tanto por desarrolladores internos, como por desarrolladores de software libre, en las últimas versiones.

MySQL es propiedad de una empresa privada, quien posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento. También de la venta de licencias privadas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía remota.

MySQL es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando y optimizando. Por eso es una de las bases de datos más usadas en Internet.

En las últimas versiones se pueden destacar las siguientes características principales:

- Flexibilidad en el sistema de contraseñas y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- Soporta gran cantidad de tipos de datos para las columnas.
- MySQL soporta UNICODE, permitiendo el desarrollo de aplicaciones sin importar la plataforma o el lenguaje del país donde se genere. También permite el manejo interno de mensajes de error.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Almacena de manera dinámica con comandos SQL. Esto permite que MySQL tenga un mejor desempeño que otros manejadores de bases de datos. MySQL posee comandos más recientes, lo cual permite un acceso inmediato a posteriores solicitudes que impliquen datos procesados con anterioridad.
- Manejo transaccional seguro. InnoDB es un tipo de tabla que soporta transacciones y manejo transaccional seguro, lo que permite decidir la confirmación o el rechazo de un conjunto de operaciones efectuadas con anterioridad en la base de datos. Asimismo soporta operaciones de revisión del estado, las cuales son utilizadas para transferir una base de datos a un determinado estado después de una falla importante del sistema.

- Los clientes pueden conectarse al servidor MySQL mediante TCP/IP o sockets de UNIX utilizando aplicaciones hechas con C, C++, Java, Perl o PHP. Es importante mencionar que MySQL puede conectarse directamente a otros manejadores de bases de datos utilizando ODBC.
- MySQL puede manejar bases de datos con 60 millones de registros y más de 70,000 tablas. Soporta 64 índices por tabla y cada índice puede estar compuesto de 15 columnas, con una longitud máxima de 1,000 bytes. Los índices pueden utilizar prefijos de columnas de tipo CHAR, VARCHAR, BLOB o TEXT.
- Utiliza privilegios y contraseñas, esto hace que MySQL sea un gestor de bases de datos relacionales, flexible y seguros, todas las claves viajan encriptadas a través de la Web. Mediante las direcciones IPs del cliente, MySQL puede restringir y administrar el acceso a los recursos del servidor.
- MySQL manipula el multiprocesamiento distribuido en uno o más CPU's.
- Para los tipos de datos que puede haber en un campo, se pueden agrupar en tres grupos: fecha y hora, numéricos y cadenas (caracteres).

Ventajas

MySQL se desarrolló para manejar bases de datos muy grandes en forma más rápida que otro tipo de bases de datos y trabaja en sistemas de producción de alta demanda. Esta característica ofrece un rico y muy actualizado conjunto de funciones, así también ofrece interfaces con muchos tipos ODBC para exportar a otras aplicaciones. Algunas de las ventajas son:

- La rapidez del código de MySQL fue escrito teniendo como meta principal un excelente desempeño, tablas en memoria que son usadas como temporales.
- Posee un sistema de privilegios y contraseñas que es muy flexible y seguro y que permite verificación basada en el host. Las contraseñas son seguras. Todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

- Los clientes pueden conectarse con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP 2003). los clientes pueden usar **named pipes** (conexión de software temporal entre dos programas o comandos) para la conexión. En sistemas Unix, los clientes se conectan usando ficheros socket Unix.
- En MySQL los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción **shared-memory** (memoria compartida). Los clientes se pueden conectar a través de memoria compartida usando la opción `protocol=memory`.
- La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible.
- Escrito en C y en C++.
- Probado con un amplio rango de compiladores diferentes.
- APIs (Interfaz de Programación de Aplicaciones) disponibles para C.
- Las funciones SQL están implementadas usando una biblioteca altamente optimizada y son tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- El diseño Multicapa de MySQL Server, con módulos independientes. El código Berkeley DB es muy estable, pero todavía se está trabajando para mejorarlo con el interfaz del motor de almacenamiento transaccional BDB en MySQL Server.

Desventajas

MySQL surgió por la necesidad de un grupo de personas de un manejador de bases de datos rápido. MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características y aunque algunas de ellas ya se han incluido en las versiones más recientes no se puede asegurar la estabilidad y buen funcionamiento de éstas. A continuación, algunas desventajas:

- Los triggers (disparadores). Aunque han mejorado, reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
- Los procedimientos almacenados fueron implementados hasta la versión 5, la cual aún presenta algunos problemas de estabilidad.
- A pesar que admite la declaración de claves foráneas en la creación de tablas, internamente no las procesa de forma diferente al resto de los campos índices.
- Los índices agilizan la recuperación pero hace más lentas las operaciones de insertar, borrar y las actualizaciones de valores en las columnas indexadas porque la escritura no afecta sólo a la fila de datos, sino a menudo también a los índices.

3.2 Características, ventajas y desventajas del lenguaje Python.



Figura 3.2.1 Logo Python

Python es un lenguaje de programación que emplea una sintaxis que favorece un código legible y posee una licencia de código abierto llamada **Python Software Foundation License**⁶ (**Licencia de la Fundación del Software Python**). Véase figura 3.2.1. No es una licencia de copyleft en términos de derechos de autor y promoción del libre uso y distribución del software. Sin embargo, permite modificaciones del código fuente, así como la creación de trabajos derivados sin requerir que ni las modificaciones ni los desarrollos derivados tengan que ser de código abierto.

⁶ Python Software Foundation License es una organización sin fines de lucro creada el 6 de marzo de 2001 dedicada al lenguaje de programación Python.

Python es un lenguaje orientado a objetos, multiplataforma y multiparadigma, es decir, que soporta diversas propuestas tecnológicas adoptadas por comunidades de programadores y desarrolladores cuyo objetivo es tratar de resolver problemas específicos. De igual forma, se fundamenta en la programación imperativa y funcional.

La programación imperativa se describe en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican a la computadora como llevar a cabo una tarea, de la misma manera que el modo imperativo en los lenguajes naturales humanos le dice que hacer al interlocutor.

La programación funcional está basada en el uso de funciones matemáticas como el cálculo lambda, un sistema formal diseñado en 1930 con el fin de investigar la definición de función, la noción de aplicación de funciones y la recursión. El cálculo lambda consiste en una regla de transformación (sustitución de variables) y un esquema simple para definir funciones.

Python fue creado por Guido van Rossum a principio de los años 90, su nombre está inspirado en el grupo cómico británico "Monty Python". Su sintaxis es limpia. Así mismo, se trata de un lenguaje interpretado o de script con tipado fuerte y dinámico.

Un lenguaje interpretado o de script es aquel que se ejecuta empleando un programa intermedio llamado intérprete. A diferencia de los lenguajes compilados, no compila el código a lenguaje máquina que pueda comprender y ejecutar directamente. De esta manera, hace que el lenguaje sea más flexible y portable.

En Python, al igual que en Java, el código fuente se traduce a un pseudo código máquina intermedio la primera vez que se ejecuta generando archivos .py que son los que se ejecutarán en ocasiones sucesivas.

Python tiene una gran biblioteca estándar, base de su filosofía **batteries included** (**baterías incluidas**), combinada con la habilidad de disponer de lenguajes de bajo nivel (C, C++) con los cuales es capaz de interactuar con otras bibliotecas. De igual forma, posee diversas implementaciones como:

- CPython, es la implementación oficial.
- IronPython para .NET
- Jython, hecho en Java.
- ActivePython es una implementación con extensiones para servidores y aplicaciones de misión crítica.
- Pippy fue realizada para la primera generación de dispositivos móviles.

Características.

- Es un lenguaje simple. No hace falta tener conocimientos avanzados de programación para poder hacer un programa.
- Permite varios estilos de programación.
- Es fuertemente tipado. No permite tratar a una variable como si fuera de un tipo distinto al que es originalmente, es necesario convertir de forma explícita dicha variable al nuevo tipo,
- Es multiplataforma. El intérprete de Python esta disponible en varias plataformas (UNIX, Solaris, Windows, Mac OS X, etc.) por lo que no requiere de bibliotecas específicas.
- Emplea enlaces dinámicos de métodos, es decir, enlaza un método y un nombre de variable durante la ejecución del programa.
- Puede incluirse en aplicaciones que requieran alguna interfaz programable.
- Emplea palabras en donde otros lenguajes usan símbolos.
- Utiliza un conteo de referencias para contabilizar las veces que un determinado recurso está siendo referido permitiendo, cuando no exista referencia a ese bloque, pueda ser liberado.

Ventajas.

- Su sintaxis es simple, clara y sencilla.
- Se pueden escribir nuevos módulos en otros lenguajes.
- Su intérprete estándar tiene un modo interactivo en el cual las expresiones pueden ser introducidas una a una con la oportunidad de ver su evaluación de manera inmediata.
- Se pueden probar porciones de código antes de integrarlo como parte de un sistema.
- Es fácil de leer.
- Acepta varios estilos para programar.
- Simplifica la gestión de memoria. Toda variable contiene una referencia a su valor. A ello se suma un sistema de recolección automática de basura que evita punteros colgantes y fugas de memoria.
- Tiene una enorme biblioteca estándar.
- Facilita la detección de errores mediante excepciones.
- Es ideal para aplicaciones cliente y desarrollos web.

Desventajas.

- No es adecuado para programación de bajo nivel.
- Si las referencias forman un ciclo, los objetos involucrados no se liberarán.
- Los programas interpretados son más lentos que los compilados.
- No es ideal para aplicaciones donde el rendimiento sea crítico.

3.3 Características, ventajas y desventajas del framework Django.



Figura 3.3.1 Logo Django.

Django es un framework de desarrollo web de código abierto escrito en Python que emplea el modelo vista – controlador. Véase figura 3.3.1 Dicho modelo es una arquitectura que propone la construcción de tres componentes: modelo, vista y controlador. Por un lado, define componentes para la representación de la información y por otro para la interacción del usuario. Esta arquitectura se basa en la idea de reutilización de código que busca facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Los frameworks ofrecen gran funcionalidad debido a los patrones de diseño con los que son construidos, siendo su característica principal su alta cohesión y bajo acoplamiento. Para acceder a esa funcionalidad se desarrollan objetos que vinculan las necesidades del sistema con la funcionalidad que éste presta.

Algunas características notables de un framework son:

- En un framework, a diferencia de las bibliotecas, “el flujo de control no es dictado por el programa que llama sino por el mismo”⁷.
- Tiene un comportamiento predeterminado por defecto útil e identificable.
- Extensibilidad. Los usuarios pueden ampliar sus características sin modificar su código.

⁷ Riehle, Dirk. *Framework Design: A Role Modeling Approach*, Swiss Federal Institute of Technology.

El modelo vista – controlador (MVC) es una arquitectura que separa los datos y la lógica de una aplicación de la interfaz del usuario y el módulo encargado de gestionar los eventos. Para ello, propone la construcción de tres componentes: modelo, vista y controlador. Por un lado define componentes para la representación de la información, y por otro para la interacción con el usuario. Esta arquitectura de software es la base de la reutilización de código y la separación de conceptos, características fundamentales de frameworks como Django que buscan facilitar la tarea de desarrollo de aplicaciones.

El MVC ha sido ampliamente adaptado como arquitectura para implementar aplicaciones web en los principales lenguajes de programación. Los primeros frameworks MVC para desarrollo web plantean un enfoque en donde el cliente manda la petición de cualquier hipervínculo o formulario al controlador y recibe de la vista una página completa y actualizada; el modelo y el controlador se encuentran alojados en el servidor. Actualmente, existen frameworks que permiten que ciertos componentes MVC se ejecuten parcial o totalmente del lado del cliente. Véase figura 3.3.2.

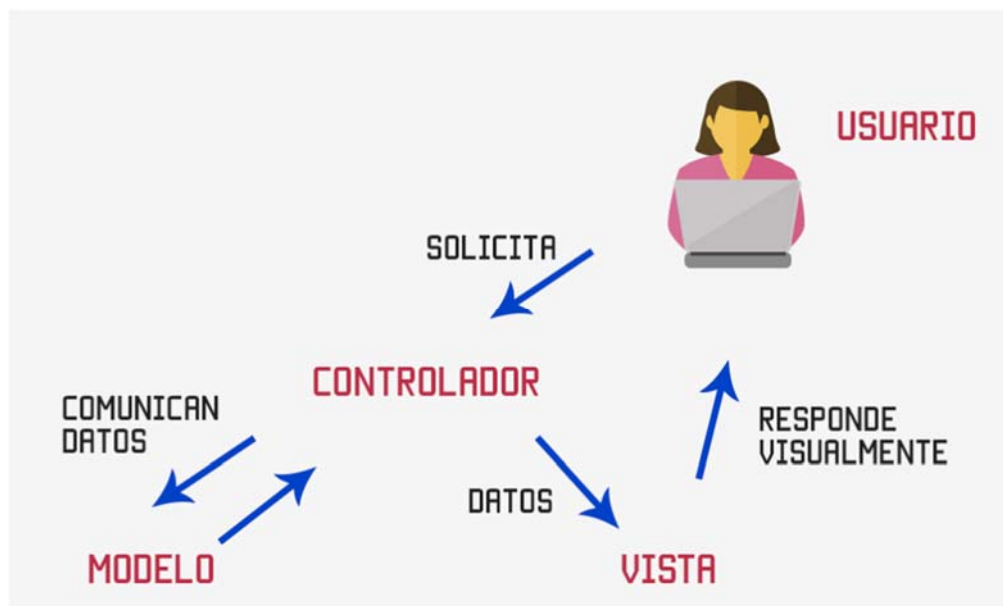


Fig. 3.3.2 Modelo Vista Controlador

Django debe su nombre al guitarrista de jazz Django Reinhardt. En un principio, fue desarrollado para gestionar un sistema de servicio de noticias. Sus orígenes en la administración de páginas de noticias ha facilitado el desarrollo de sitios web orientados

a contenidos. Por ejemplo, en lugar de escribir controladores y vistas para la administración de una página, Django tiene incorporada una aplicación para administrar los contenidos. Dicha aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas.

Características.

- Mapeo objeto – relacional. Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional.
- Es una API⁸ de base de datos robusta.
- Es un sistema incorporado de vistas que ahorra el escribir la lógica de ciertas tareas.
- Es un sistema extensible de plantillas basado en etiquetas.
- Su distribución principal incluye componentes que proporcionan cacheo, normalización de URLs y soporte de sesiones.
- Documentación accesible mediante una aplicación administrativa.
- Es una plataforma habitual que brinda múltiples herramientas.

Ventajas.

- Facilita el desarrollo de sitios web complejos.
- Pone énfasis en el re-uso, conectividad y extensibilidad de componentes.
- Incluye un servidor liviano para realizar pruebas.
 - No requiere de otras bibliotecas de Python para tener una funcionalidad básica.
- Es un framework web de alto nivel.
- Tiene una filosofía de no crear bloques de código iguales, fomentando la reutilización del mismo.

⁸ Application Programming Interface (Interfaz de programación de aplicaciones). Conjunto de subrutinas, funciones y procedimientos de una biblioteca para ser usado por otro software como una capa de abstracción.

- Tiene por defecto un sistema de administración activo que no requiere de algún tipo de configuración.
- ORM (Object Relational Mappers) es un modelo de programación que consiste en transformar las tablas de una base de datos en entidades que simplifiquen el acceso a los datos por parte del programador.

Desventajas.

- Su servidor integrado de pruebas solo permite un usuario a la vez.
- Presenta problemas de rendimiento.
- El esquema de URLs puede ser pesado al obligar a cada petición a recorrer todo el esquema de URLs.
- Algunos usuarios no lo consideran ideal para aplicaciones en tiempo real.

3.4 Características, ventajas y desventajas del lenguaje HTML.

El lenguaje HTML es un estándar, compuesto por recomendaciones publicadas por un consorcio internacional conocido como el "World Wide Web Consortium" (**W3C**).

Las especificaciones oficiales de HTML describen las "instrucciones" del lenguaje, es decir, cómo las interpretan los programas informáticos. Esto permite visualizar páginas web independientemente del sistema operativo o arquitectura del equipo del usuario.

A pesar que las especificaciones son detalladas, hay un margen para la interpretación por parte del navegador web y esta es la razón por la cual la misma página puede aparecer de manera diferente en un navegador u otro.

También algunos editores de software adicionan instrucciones HTML exclusivas que no existen en las especificaciones de W3C. Por este motivo, las páginas Web que contienen dichas instrucciones pueden ser vistas en un navegador y ser completa o parcialmente ilegibles en otros. Se recomienda, que las páginas web sigan las recomendaciones de W3C, de tal manera lleguen al dominio público de manera más amplia. Vea logo en la figura 3.4.1.



Figura 3.4.1 Logo HTML.

Características.

HyperText Markup Language (Lenguaje de Marcas de Hipertexto) son las siglas de HTML. Predomina para la construcción de páginas. Uno de los elementos fundamentales del lenguaje HTML es su naturaleza al hipertexto, se desarrolla en documentos digitalizados como Wordpad de Windows, Gedite de linux o cualquier otro editor que admita texto sin formato.

La conectividad es otra característica esencial de HTML hasta el punto que suele afirmar que su hipertextualidad está relacionada a su conectividad. Los enlaces no se realizan únicamente a través de los elementos textuales, sino que ahora la conexión entre documentos y dentro de ellos se puede realizar mediante conexiones o enlaces a través de imágenes y hacia otros componentes, aplicaciones y medios no textuales, a pesar de esto el texto sigue siendo una forma esencial del hipertexto.

Los editores WYSIWYG (What You See Is What You Get) tienen como característica ver el resultado de lo que se está editando en tiempo actual, conforme se va desarrollando el documento. Con esta forma tenemos una manera más simple de estar editando, ya que estos programas también tienen la opción de trabajar con la vista preliminar, con su propia sección HTML que va generando todo el código conforme se va trabajando. Podemos mencionar algunos ejemplos de editores WYSIWIG como son Macromedia Dreamweaver o Microsoft FrontPage.

HTML también se usa para describir la estructura y el contenido en forma de texto, y complementar el texto con objetos o imágenes. Se escribe en "etiquetas", abierto y cerrado por corchetes angulares (<,>). HTML puede describir la apariencia de un documento y puede incluir un script como JavaScript y otros, para manejar el comportamiento de navegadores Web y otros procesos de HTML. Al editar todo en HTML y si en algún momento se olvida algún código o etiqueta, de manera sencilla se usa el editor visual o WYSIWYG y proseguimos ahí la edición, o inversamente.

HTML es un lenguaje sencillo para comprender y utilizado para la presentación de la información, aunque no incluye facilidades para algún tipo de interacción con el usuario. Es un lenguaje cuyo propósito es la generación de sitios informativos y estáticos.

Resumiendo de manera simple las características.

- HTML se utiliza para crear páginas web.
- Es un lenguaje basado en etiquetas.
- Las etiquetas de HTML están rodeadas por un soporte angular.
- Puede utilizar amplios rangos de colores, objetos y diseños.

Ventajas

- Fácil de aprender y usar.
- Ampliamente utilizado.
- Es libre, y no requiere la adquisición de ningún software en particular.
- Compatible con casi todos los navegadores, si no con todos los navegadores.
- Cualquier navegador admite HTML, aunque los estándares soportados varían.
- Ampliamente utilizado, gracias a la World Wide Web.
- Muy parecido a la sintaxis XML, utilizada cada vez más para el almacenamiento de datos.
- Fácil de codificar aun para programadores principiantes.
- La navegación de las páginas busca ser más eficiente a través de uso de cachés.

Desventajas

- Sólo puede crear páginas estáticas y planas con HTML puro.
- Necesidad de escribir gran cantidad de código para hacer una simple página web.
- La estructuración de documentos HTML en algunos casos es difícil de entender.
- Estar consciente que hay etiquetas obsoletas, hay que asegurarse de no usarlas.
- Las etiquetas obsoletas generalmente aparecen porque otro lenguaje que funciona con HTML ha reemplazado el trabajo original de la etiqueta. Esto puede significar que se necesita aprender ese otro lenguaje.
- En algunos casos puede ser que el botón de atrás del navegador no se comporte como deseamos.
- Para un sitio completamente accesible a personas con capacidades diferentes, es necesario tener en cuenta etiquetas específicas, que muchas veces no son utilizadas ni bien documentadas.
- Los scripts se pueden complicar bastante cuando se tienen que comunicar varios marcos entre sí.

3.5 Características, ventajas y desventajas del servidor de páginas web Apache.

Web Apache o Apache es uno de los servidores Web más utilizados en el mundo. Empezó sus funciones en 1995. El NCSA (National Center for Super Computing Applications) trabajó y creó el servidor Web que llegaría, en algún momento, a ser el más utilizado. NCSA abandona el proyecto y los mismos usuarios de Apache crearon un foro para poder compartir información y parches sobre el servidor. Este foro evolucionó, eventualmente, hasta convertirse en el Apache Group. El servidor Apache fue creado, entonces, utilizando el código fuente del servidor de NCSA. La primera versión del servidor apareció en abril de 1995. Apache se considera un servidor flexible y sencillo que se ejecuta en diferentes plataformas como Linux, UNIX, Windows 95/98/NT/XP/2000/2003, etc. En 1998 IBM llegó a un acuerdo que permitió que Apache web funcionara en plataformas Windows convirtiéndose en una alternativa al servidor IIS (Internet Information Server).

La instalación y puesta a punto depende del sistema operativo. Todas las distribuciones de Linux son compatibles con el servidor Apache, en muchos casos contando con un paquete de instalación dentro de sus repositorios; en estos casos solamente hay que seleccionar la opción para instalar el servidor y éste quedara instalado y funcionando. Apache a partir de la versión 2 utiliza para Windows un paquete completo de instalación en forma de ejecutable (.msi). El paquete de instalación puede descargarse desde:

- <http://www.apache.org>

Para Windows XP o superior es necesario instalar el Service Pack 1. Si se instala sobre Windows 98 o superiores es necesario instalar Windows Socket 2 update que se descarga del sitio de Microsoft (esto no es necesario si se tiene una versión más actual (98, Me, NT, 2000, XP, 7 etc.). Vea logo en la figura 3.5.1.



Figura 3.5.1 Logo Apache

Características

Es configurable y extensible mediante módulos. Existen una gran cantidad de módulos disponibles para Apache y permite la creación de nuevos módulos. Éstos pueden ir desde el soporte de lenguaje de programación del lado del servidor hasta esquemas de autenticación.

Algunas interfaces de lenguaje de Apache comunes incluyen a Python, Perl, Tcl y PHP. Los módulos de autenticación con mayor popularidad incluyen mod_access, mod_auth, mod_digest y mod_auth_digest, el sucesor de mod_digest. Un ejemplo de otras características de Apache incluyen Secure Sockets Layer y Transport Layer Security (mod_ssl), un módulo proxy (mod_proxy), un módulo de reescritura de URL (mod_rewrite), archivos de registro personalizados (mod_log_config) y soporte de filtrado (mod_include y mod_ext_filter).

En Apache el alojamiento virtual permite que una instalación sirva a muchos sitios web diferentes. Como ejemplo, una máquina con una instalación de Apache podría servir simultáneamente www.example.com y www.example.org.

Los métodos populares de compresión en Apache incluyen el módulo de extensión externo, `mod_gzip`, este fue implementado para ayudar con la reducción del tamaño (peso) de las páginas Web servidas a través de HTTP. ModSecurity es un motor de detección y prevención de intrusiones de código abierto para aplicaciones web. Los registros de Apache se pueden analizar a través de un navegador Web mediante scripts gratuitos, como AWStats / W3Perl o Visitantes.

Apache ofrece mensajes de error configurables, bases de datos de autenticación basadas en **DBMS** (*Database Management System*, motores de bases de datos) y negociación de contenido. También es compatible con varias interfaces gráficas de usuario (**GUI**, *Graphical User Interface*).

Es compatible con la autenticación de contraseña y autenticación de certificados digitales. Debido a que el código fuente está disponible gratuitamente, cualquiera puede adaptar el servidor para necesidades específicas y hay una gran biblioteca pública de complementos de Apache.

Para casos de implementar una única arquitectura, Apache proporciona una variedad de Módulos Multiprocesamiento (MPM) que permiten que Apache se ejecute en un proceso híbrido (proceso e hilo) o híbrido de eventos, para satisfacer mejor las demandas de cada infraestructura en particular. Esto implica que la elección del MPM correcto y la configuración correcta es importante. El diseño de Apache es reducir la latencia y aumentar el rendimiento, en relación con el simple manejo de más solicitudes, garantizando así un procesamiento consistente y fiable de las solicitudes dentro de periodos razonables.

Para la entrega de páginas estáticas, Apache versión 2.2 se consideró significativamente más lento. Para solucionar este problema, los desarrolladores de Apache crearon el motor MPM, que mezcla el uso de varios procesos y varios subprocesos por proceso en un bucle asíncrono basado en eventos. Esta arquitectura y la forma en que se

implementó en la serie Apache 2.4, proporciona un rendimiento equivalente o ligeramente mejor que los servidores web basados en eventos.

El proyecto Apache es un esfuerzo de desarrollo de software colaborativo dirigido a crear una implementación de código fuente robusta, comercial, rica en funciones y libremente disponible de un servidor HTTP web. Este proyecto forma parte de la Apache Software Foundation. Además, cientos de usuarios han aportado ideas, código y documentación al proyecto. Apache 2.4 dejó de soportar BeOS, TPF y plataformas más antiguas.

El código del Servidor HTTP Apache fue actualizado a la Licencia Apache 2.0 (de la licencia 1.1 anterior) en enero de 2004, y Apache HTTP Server 1.3.31 y 2.0.49 fueron las primeras versiones con la nueva licencia.

Al proyecto de OpenBSD no le gustó el cambio y continuó el uso de las versiones anteriores a Apache 2.0, que junto a Apache 1.3.x para sus propósitos. Inicialmente lo reemplazaron con Nginx, y poco después hicieron su propio reemplazo, OpenBSD Httpd, basado en el proyecto relayd.

Características de las versiones.

Versión 1.1: La Licencia Apache 1.1 fue aprobada por la ASF en 2000: El cambio principal de la licencia 1.0 está en la 'cláusula de publicidad' (sección 3 de la licencia 1.0); los productos derivados ya no están obligados a incluir la atribución en sus materiales publicitarios, sino sólo en su documentación.

Versión 2.0: La ASF adoptó la Licencia Apache 2.0 en enero de 2004. Los objetivos establecidos de la licencia incluían hacer que la licencia fuera más fácil para los proyectos que no son ASF, mejorando la compatibilidad con el software basado en GPL, permitiendo que la licencia fuera incluida por referencia de la lista de cada archivo, aclarar la licencia sobre las contribuciones y la exigencia de una licencia de patente sobre las contribuciones que necesariamente infringen patentes de un contribuyente propio.

Actualmente se trabaja con versiones 2.2.X, pero muchos servidores en producción utilizan la versión más extendida 1.3.33. La última versión de Apache es la 2.4.27 (Julio 2017). Se recomienda utilizar la versión 2.2 ya que incluye nuevas funcionalidades.

En la figura 3.5.2 se aprecia un diagrama de la arquitectura básica donde se incluye un servidor apache.

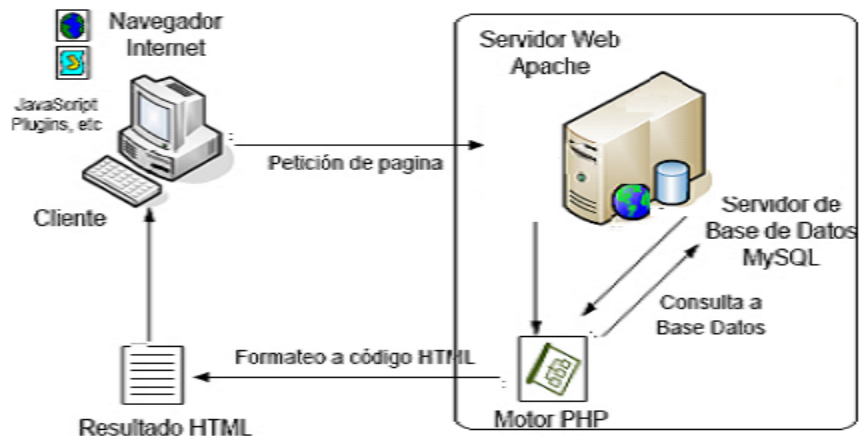


Figura 3.5.2. Arquitectura de un servidor con Apache.

Ventajas

Entre sus ventajas podemos destacar las siguientes:

- Estándar

Cumple con el estándar HTTP/1.1 que amplía su funcionalidad, su soporte es amplio para este estándar.

- Precio.

Apache Web Server es un software de código abierto, lo que significa que está disponible para que cualquiera pueda descargarlo sin costo alguno. Su código fuente puede ser modificado para satisfacer sus necesidades individuales. Esto le da a Apache una ventaja significativa sobre casi todos sus competidores sin sacrificar ninguna característica. El estado de código abierto también elimina la licencia recurrente y los cargos de soporte necesarios para continuar ejecutando otros programas de servidor Web.

- Modular.

Modular, consta de un núcleo y un conjunto de módulos que aportan funcionalidad adicional al servidor Web como por ejemplo comunicaciones de vías seguras, autenticación contra LDAP, soporte para páginas dinámicas en Perl, PHP, Python, etc. Esto permite personalizar el servidor dependiendo los requerimientos del usuario.

- Características.

Apache es un poderoso programa de servidor Web con características que se comparan con sus competidores de alto precio. El software incluye un panel de control de administración, mensajes de error personalizables y esquemas de autenticación. El módulo de alojamiento virtual le permite ejecutar varios sitios web desde el mismo servidor. Además de éstas y otras características estándar incluidas en el paquete de instalación, como el Servicio de nombres de dominio, Protocolo simple de transferencia de correo (SMTP) y Protocolo de transferencia de archivos (FTP), existen numerosos complementos de terceros que puede instalar para personalizar su servidor Web. La vibrante comunidad de desarrolladores de Apache también produce actualizaciones frecuentes para que siempre tenga acceso a las últimas características y parches de seguridad. Tiene fácil administración ya que se cuenta con interfaces gráficas, así como un conjunto de herramientas desarrolladas con este propósito.

- Compatibilidad.

Apache es compatible con numerosas configuraciones de hardware y sistemas operativos. Funciona en Linux, Windows NT, MacOS, Unix y muchos otros sistemas. Cada instalación se puede ajustar para adaptarse a las capacidades técnicas de su hardware. Apache incluye soporte para lenguajes de programación como PHP, Perl y Python, entre muchos otros, junto con cifrado SSL y TLS para sitios web que requieren seguridad elevada. También puede configurarse de manera sencilla ya que cuenta con archivos de configuración específicamente preconfigurados para cada sistema operativo.

- Soporte técnico

Los recursos de soporte técnico para Apache están disponibles en varios sitios web de todo el mundo. Esto permite a los propietarios de servidores acceder a artículos de referencia. Esto le da a Apache una gran ventaja sobre los programas que sólo tienen un sitio web de la compañía como una fuente de apoyo. Cuando se encuentra un nuevo error, la comunidad de usuarios de código abierto normalmente crea un parche para solucionarlo y publica la solución de forma gratuita en foros y sitios web de redes sociales.

- Eficiencia.

Muestra ventajas en pruebas de rendimiento respecto a otros servidores Web. Consume menos recursos que otros servidores Web, dado que su código se encuentra en constante optimización para su desempeño.

Desventajas

Para mayor referencia las desventajas de Apache se vinculan a la comparación con otras marcas en el mercado como IIS y Nginx etc.

- Seguridad.

Buena pero no excelente. Desde que Apache fue desarrollado para un sistema operativo que no es Microsoft, y la mayoría de los programas maliciosos han aprovechado vulnerabilidades en Windows. Usar Apache en Windows no tiene el máximo de confiabilidad.

- Centralización de información.

Si se encuentra con un problema de configuración, no hay una oficina central a la que recurrir para obtener soporte técnico, sólo los foros web llenos de personas que tuvieron el mismo problema y esperamos documentar sus soluciones.

- Manejo de variables.

Debido a que HTTP es un protocolo *stateless*, no existe forma de manejar variables entre sesiones de manera nativa. Esto significa que dependemos de los lenguajes incrustados sobre Apache para el manejo de información entre diferentes peticiones http.

- Rendimiento.

Aunque Apache puede ser configurado para mejorar su rendimiento, nunca puede igualar el rendimiento de la aplicación NGINX. El principal problema de Apache es la forma en que maneja las solicitudes concurrentes. Apache puede servir un gran número de peticiones por segundo, pero a medida que aumenta el número de solicitudes, el rendimiento de Apache comienza a disminuir. Apache no proporciona contenido estático y solo contacta con el procesador externo cuando es necesario.

3.6 Características, ventajas y desventajas de JavaScript y CSS.

A pesar de que es posible crear un sistema web 100% funcional utilizando únicamente HTML y procesamiento del lado del servidor, la vasta mayoría⁹ de los sitios modernos dependen de JavaScript (abreviado en ocasiones como JS) para proveer una experiencia de usuario mucho más dinámica – desde algo tan sencillo como una caja de diálogo que muestra algún mensaje, hasta peticiones asíncronas o procesamientos de cifrado.

JavaScript es una tecnología “vieja” para los estándares de internet, publicado originalmente con Netscape Navigator en el año de 1995¹⁰. Fue un proyecto en conjunto con Sun Microsystems^{id.}, y originalmente era solo una de las múltiples opciones de lenguajes de **scripting (interpretado)** para navegadores de internet – compitiendo directamente con otros lenguajes.

⁹ 94.7%. Recolectado el 5 de Septiembre de 2017 del sitio web <https://w3techs.com/technologies/details/cp-javascript/all/all>.

¹⁰ <https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html>. Recolectado el 5 de Septiembre de 2017.

En 1997 se publicó el estándar ECMAScript, del cual JavaScript se convirtió en la implementación mejor conocida. Con el paso del tiempo esta estandarización significó la victoria para JavaScript sobre sus competidores, ya que los desarrolladores prefirieron crear código multiplataforma, en vez de comprometerse con un pequeño sector del mercado (por ejemplo, navegadores Internet Explorer de Microsoft, al usar VBScript).

Hoy en día, JavaScript cuenta con una variedad de bibliotecas que incrementan tremendamente su potencial, al facilitar la programación de funcionalidades complejas y minimizar la repetición de código comúnmente utilizado. Entre las más conocidas se encuentran JQuery, React y Angular.

Un punto crucial en la utilidad de JavaScript y, por ende, su importancia en los sitios y sistemas web modernos, es el uso de llamadas asíncronas. Esto se logra a través de AJAX, siglas para **Asynchronous JavaScript And XML (JavaScript y XML asíncrono)**. La figura 3.6.1 brinda un breve esquema de su funcionamiento, el cual explicaremos a continuación.

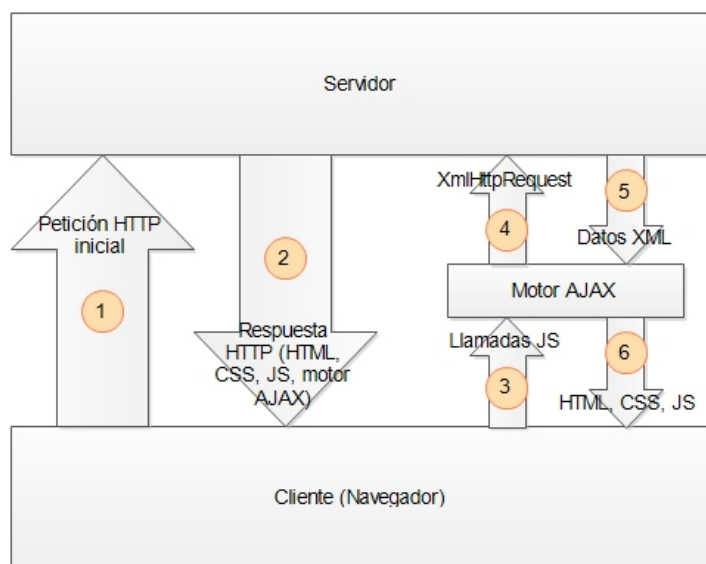


Figura 3.6.1. Esquema del funcionamiento de AJAX.

Como en cualquier comunicación HTTP, el cliente (a través del navegador, en este caso) realiza una petición HTTP y recibe una respuesta, la cual puede contener código HTML, CSS (del cual hablaremos más adelante), código JavaScript, y también, el motor de AJAX (que es solamente código JavaScript, aunque lo trataremos como un ente

separado para propósitos de esta explicación). Esto está cubierto por los números (1) y (2) de la figura 3.6.1.

Una vez que el cliente recibe el motor AJAX, este permite realizar peticiones *sin realizar una nueva petición HTTP global*. Como explicaremos, esto permite que el usuario interactúe con el sitio o sistema sin percibir una recarga de página.

La forma en que funciona es la siguiente: algún evento (un click del usuario, por ejemplo) dispara una llamada de JavaScript al motor de AJAX (número 3). Este motor procesa la llamada y genera una petición de AJAX, específicamente llamada XMLHttpRequest¹¹.

Esta petición se envía al servidor (número 4) y es procesada por el mismo de acuerdo a la programación que se tenga en ese lado de la comunicación. Diferentes servidores procesan estas llamadas de diferentes maneras.

Sin importar los pormenores de ese procesamiento, el servidor regresará una respuesta (llamada, apropiadamente, XMLHttpRequestResponse). Usualmente, esta respuesta será información de texto codificada en XML (de ahí el X de AJAX). Esto se aprecia en el número 5.

Por último, el motor de AJAX traduce esta información a elementos operables por el usuario: objetos de HTML, código de JavaScript, o incluso código de CSS. Esto se aprecia en el número 6 de la figura 3.6.1.

De esta manera, podemos generar nuevos elementos, modificar elementos existentes, e incluso eliminarlos, sin recargar toda la página, generando una experiencia mucho más dinámica para el usuario, al menos en apariencia¹².

Ventajas de JavaScript

- Permite procesamiento del lado del cliente, decrementando la carga de ancho de banda y procesamiento en el servidor.

¹¹ Un XMLHttpRequest normalmente está compuesta de texto, pero puede incluir diferentes tipos de datos – incluyendo información binaria, lo que permite que AJAX sea viable también para operaciones que involucran archivos. Lo mismo aplica para el objeto de respuesta, XMLHttpRequestResponse.

¹² De acuerdo a la escala de interactividad notada por Novak et al., Measuring the customer experience in online environments: a structural modeling approach. Mark Sci, 19 (1) (2000), pp. 22-42.

- Es un estándar compatible con todos los navegadores sin necesidad de instalar complementos adicionales, maximizando el alcance de cualquier solución programada.
- Además de ser compatible, está habilitado por la mayoría de los navegadores de manera predeterminada, a diferencia de otras tecnologías como Java o Flash.
- Cuenta con bibliotecas que facilitan el desarrollo de código complejo, y que incluyen funcionalidad como un motor de AJAX.
- Es un lenguaje relativamente sencillo de aprender, al ser débilmente tipeado, utiliza una sintaxis similar a php o C, y cuenta con una consola de desarrollo para la identificación de errores en cualquier navegador.
- Es un lenguaje basado en eventos, ajustándose perfectamente al tipo de interacciones que un usuario tiene con un sitio o sistema web.
- Tiene todas las habilidades de un lenguaje procedural, como verificación de condiciones, ciclos, y capacidades de ejecución basadas en casos.

Desventajas de JavaScript

- Al ejecutar el código en el lado del cliente, es imposible evitar que este realice modificaciones en su comportamiento. Esto tiene repercusiones de operación (es imposible garantizar que la funcionalidad será 100% como fue programada) así como de seguridad. Por ello, es importante que ninguna rutina de validación crítica se realice a través de JavaScript.
- Para el cliente es también posible sufrir ataques a través de JavaScript, si el sitio que aloja el código ha sido vulnerado, y el código reemplazado. Al momento de escribir este capítulo, 60% del malware fue encapsulado en código de JavaScript¹³.
- Actualmente JavaScript no es paralelizable.

¹³ <https://blog.barkly.com/phishing-statistics-2017>. Recolectado el 6 de Septiembre de 2017

Por su parte, CSS (**Cascading Style Sheets, Hojas de Estilo en Cascada**) es un lenguaje que permite “describir la presentación de documentos HTML o XML”¹⁴. Esto significa lo siguiente:

- Una hoja de estilo nos permite definir esta presentación, o *estilo*, para los elementos definidos en una página web – elementos *p*, *body*, *h1*, etc.

Por su parte, el término “en cascada” se refiere a la habilidad de aplicar múltiples estilos (definidos en diferentes archivos css, o en el mismo código) de manera simultánea y sin conflicto a un mismo elemento.

Ventajas de CSS

- Al utilizar especificaciones separadas para contenido (HTML) y estilo (CSS) podemos definir estilos diferentes para diferentes medios de entrega final. Por ejemplo, al desplegar un sitio o sistema web en un equipo de escritorio contra un equipo portátil, o al utilizar un sistema de navegación asistida.
- Podemos separar las labores de programación y diseño a diferentes equipos de trabajo, acortando tiempos de desarrollo y permitiendo la utilización de personal altamente capacitado en cada área – en lugar de depender del mismo personal para ambas tareas
- También ahorra tiempos al permitir aplicar el mismo estilo a todos los objetos del mismo tipo, evitando repetir código.
- Lo anterior no limita la individualización de estilos para diferentes recursos del mismo tipo. Es posible aplicar diferentes estilos a dos elementos *p*, por ejemplo, con base en su ubicación en la jerarquía del documento.
- Al aplicar los estilos en cascada, podemos afinar aún más las diferencias entre elementos, sin causar conflictos con otros objetos ni requerir cambios en los archivos de estilo general.

¹⁴ <https://developer.mozilla.org/es/docs/Web/CSS>. Recolectado el 6 de Septiembre de 2017

Desventajas de CSS

- La principal desventaja de CSS es la compatibilidad. Aunque CSS es un estándar muy difundido, el soporte en diferentes navegadores no es el mismo. Esto se ejemplifica en la figura 3.6.2.
















































<u>perspective</u>	 10	 16	 36	 9	 23
<u>perspective-origin</u>	 10	 16	 36	 9	 23
<u>resize</u>		 5	 4	 4	 15
<u>rest</u>					
<u>rest-after</u>					
<u>rest-before</u>					
<u>ruby-align</u>	 38				
<u>tab-size</u>		 4	 21	 6.1	 15
<u>text-align-last</u>	 5.5	 12	 35		
<u>text-combine-upright</u>	 11	 48	 48	 34	
<u>text-decoration-color</u>		 6			
<u>text-decoration-line</u>		 6			
<u>text-decoration-style</u>		 6			
<u>text-justify</u>	 5.5				
<u>text-orientation</u>			 48		 34
<u>text-overflow</u>	 6	 7	 4	 3.1	 11
<u>text-shadow</u>	 10	 3.5	 4	 4	 9.6
<u>text-underline-position</u>					
<u>transform</u>	 10	 16	 36	 9	 23

Figura 3.6.2. Estatus de soporte de características de CSS3 en diferentes navegadores. Recolectado de https://www.w3schools.com/cssref/css3_browsersupport.asp el 9 de Septiembre de 2017.

CAPÍTULO IV. DISEÑO Y CONSTRUCCIÓN DEL SISTEMA

4.1 Modelado del sistema.

Un modelo es una representación simplificada de un sistema elaborada para comprender, predecir y controlar el comportamiento de dicho sistema.

El modelado de sistemas es una técnica para tratar con la complejidad inherente de los mismos. El uso de modelos ayuda a "visualizar" el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente. Las herramientas de modelado y las de Ingeniería de software pueden ayudar a verificar la corrección del modelo.

4.1.1 Diagrama de contexto.

Un diagrama de contexto de un sistema representa todas las entidades externas que podrían interactuar con un sistema. Dicho diagrama retrata el sistema en el centro, sin detalles de su estructura interna, rodeado por todos los sistemas, ambientes y actividades con las que interactúa. El objetivo del diagrama es enfocar la atención en los factores externos y eventos que deberán considerarse en el desarrollo de un set completo de requerimientos y restricciones del sistema.

Los diagramas de contexto del sistema son usados tempranamente en un proyecto para obtener un acuerdo del alcance de la investigación. Son típicamente incluidos en documentos de requerimiento. Estos diagramas deben poder ser leídos por todas las partes interesadas, por lo que deben ser escritos en lenguaje sencillo, para la comprensión de todas las partes.

El diagrama de contexto del sistema de inventario y ticketing para la ENALLT se muestra en la figura 4.1.1.1.

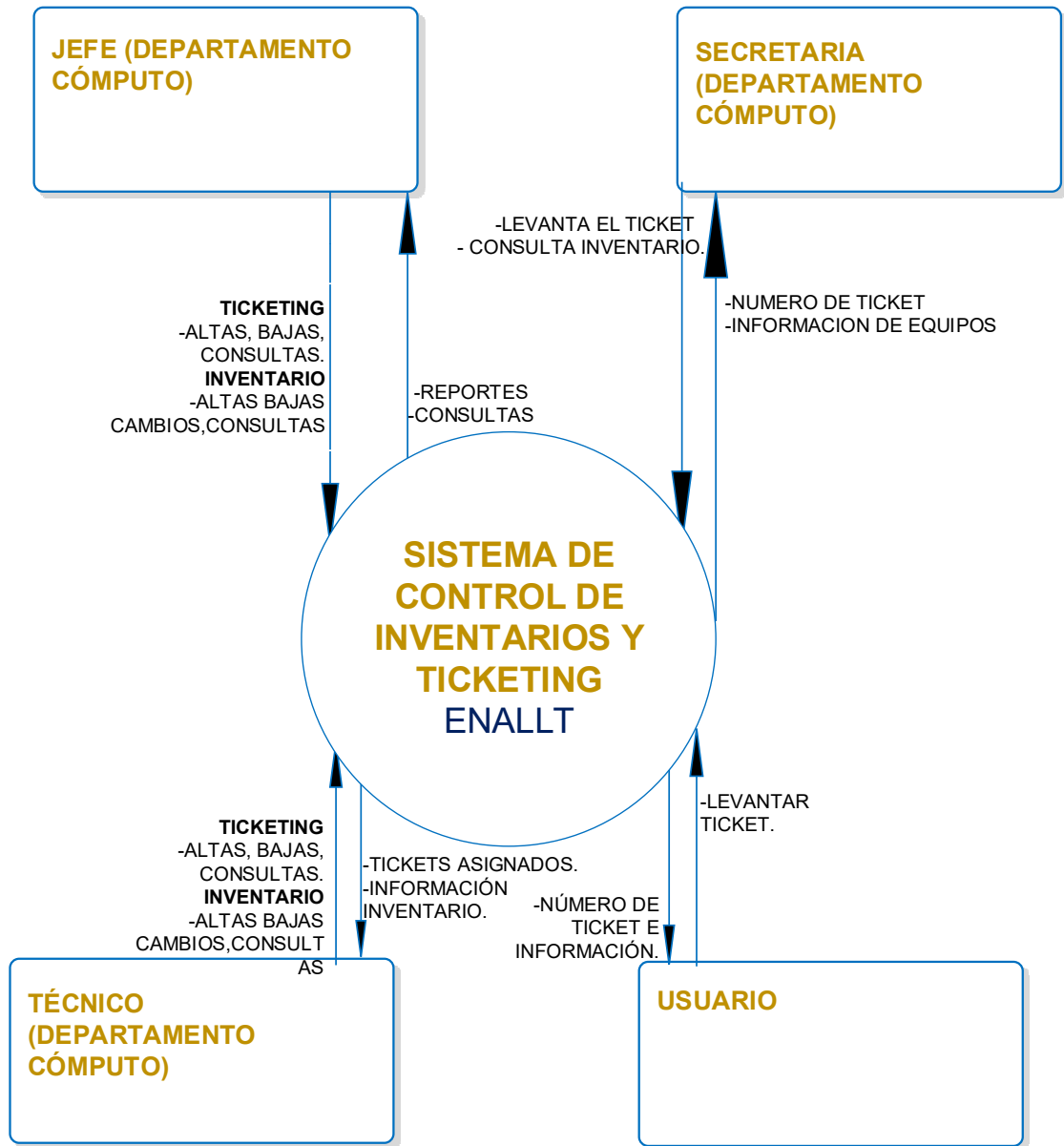


Figura 4.1.1.1. Diagrama de contexto

4.1.2 Diagrama de casos de uso.

Un **diagrama de caso de uso** describe las actividades que deben realizarse para llevar a cabo un proceso. Este representa las funciones que proporciona un sistema que son de valor para sus usuarios. Los casos de uso se muestran como parte del sistema que está siendo modelado

El **diagrama de casos de uso** también se refiere a un diagrama de comportamiento UML (Unified Modeling Language) mejorado. Define una notación gráfica para representar un sistema a ser implementado.

Los casos de uso se representan por dos componentes, uno o varios actores llamado usuario (figura de persona) y uno o varios casos de uso en forma de ovalo.

En resumen, un diagrama de **casos de uso** muestra la relación entre los actores y los casos de uso del sistema. En la figura 4.1.2.1 se muestra el diagrama de Casos de Usos de manera general para el sistema de inventario y solicitudes de soporte. Donde los actores son:

- Jefe del departamento de cómputo
- Técnico del departamento de cómputo
- Secretaria del departamento de cómputo
- Usuario.

Y los casos de uso:

- Inventario.
- Ticketing.
- Personal.
- Reportes.

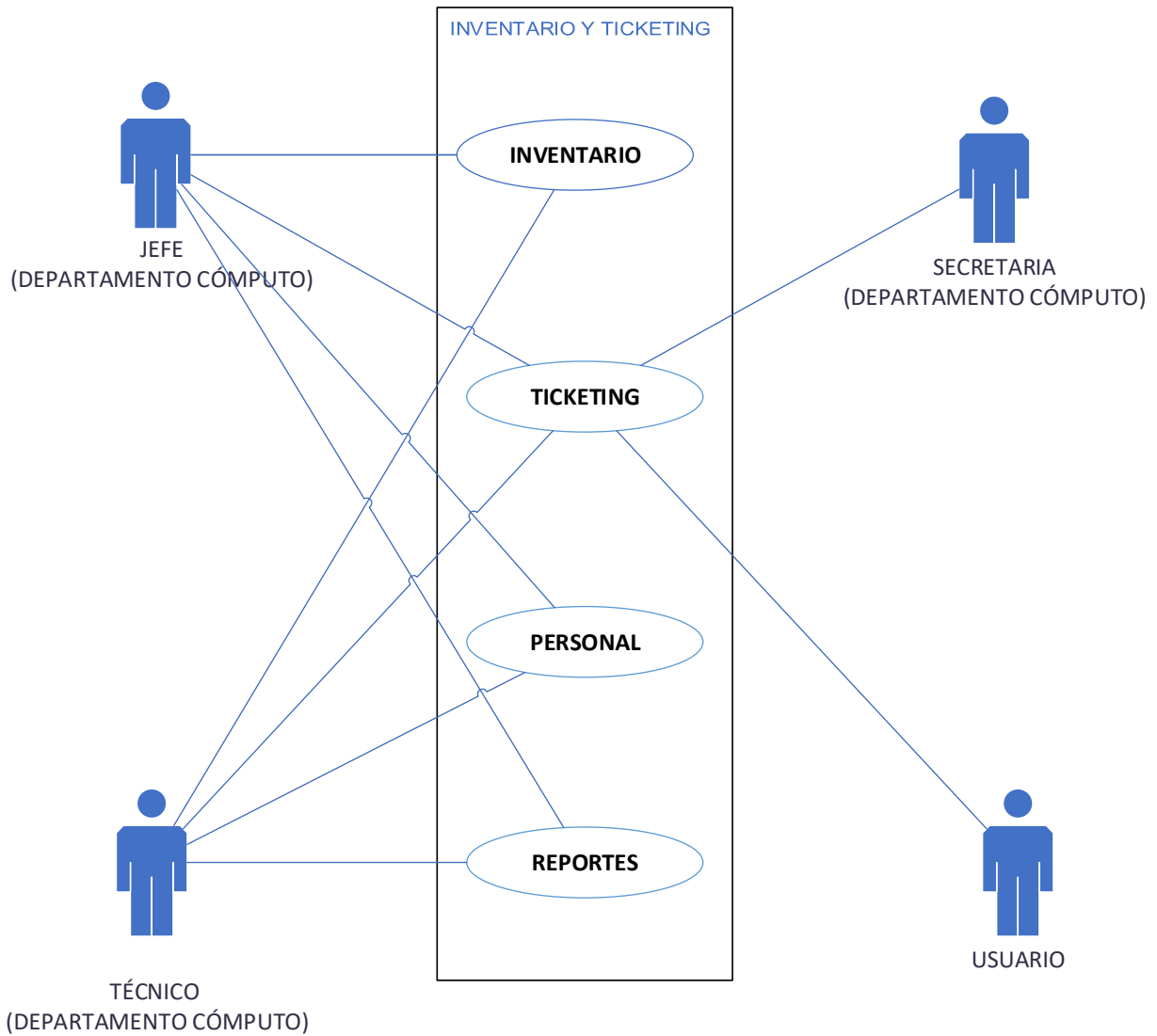


Figura 4.1.2.1 Representación general

El módulo de inventario tiene por objeto el manejo de todo el equipo de cómputo inventariado y estará conformado por las siguientes opciones. Alta de equipo, vista de equipo, cambio de IP y relación de IP. Los actores son el jefe del departamento de cómputo y el técnico del departamento de cómputo. Ver figura 4.1.2.2.

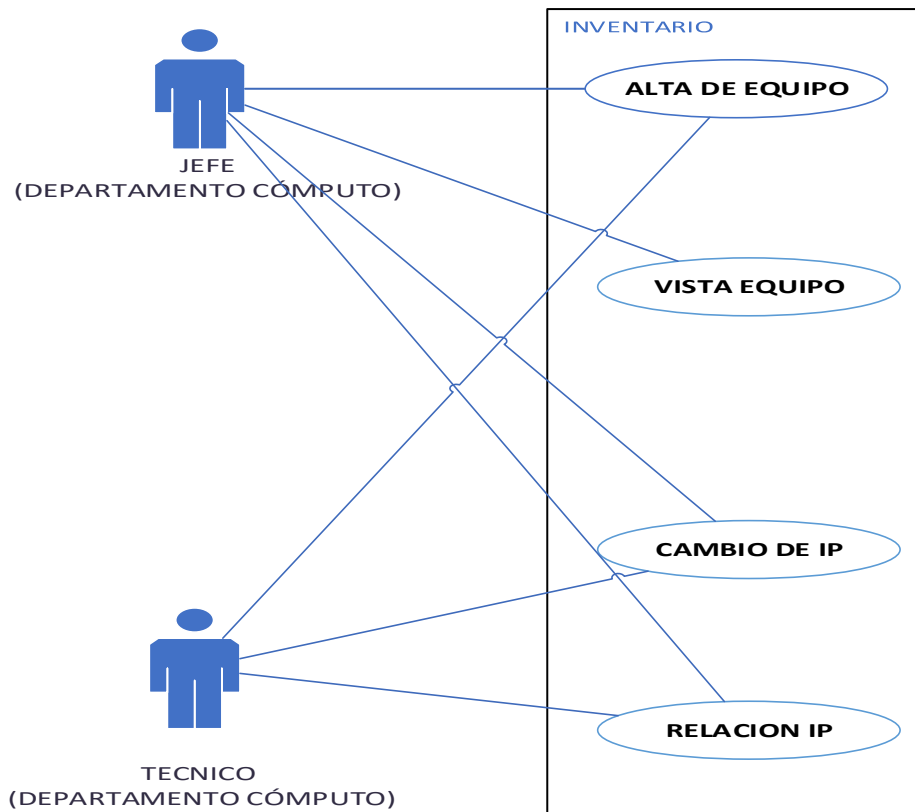


Figura 4.1.2.2. Inventario

En la tabla 4.1.2.3 representamos el cuadro de secuencias de inventario para el diagrama inventario.

Caso de uso:		Inventario-Alta de equipo		
Actores:		Jefe Departamento de Computo, técnico Departamento de Computo		
Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Ingreso al sistema con usuario y contraseña	2	Acceso a la URL inventario.	
3	Escoge la opción alta de equipo	4	Llena la plantilla con los datos del nuevo equipo	
5	Una vez terminado se salva.			

Tabla 4.1.2.3. Cuadro de secuencias de inventario para diagrama inventario

El módulo de ticketing tiene por objeto el manejo de tickets que se abren para reportar un incidente o requerimiento por parte del equipo de cómputo. Ver figura 4.1.2.4.

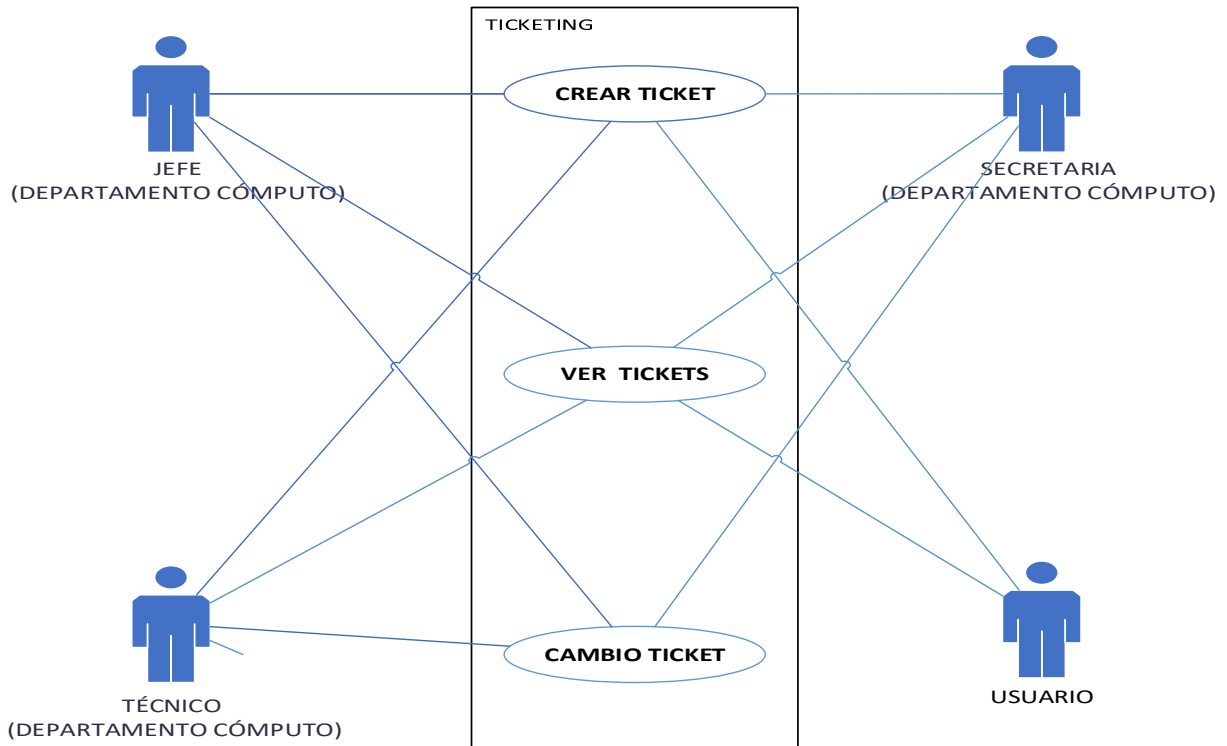


Figura 4.1.2.4. Ticketing

En la figura 4.1.2.5 representamos el cuadro de secuencias de Inventario para el caso de uso de ticket.

Caso de uso:		Ticketing-Crear Ticket		
Actores:		Jefe Departamento de Computo, Secretaria Departamento de Computo		
Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Ingreso al sistema proporcionando usuario y contraseña	2	Escoger la URL de crear ticket.	
3	Llenar la caratula con los datos necesarios	4	Salvar los datos registrados	
5				

Tabla 4.1.2.5. Cuadro de secuencias de inventario para caso de uso de ticket

El módulo de personal tiene por objeto apoyar a los módulos de inventario y ticketing en el manejo de información y su relación con el personal involucrado con el sistema. Ver figura 4.1.2.6.

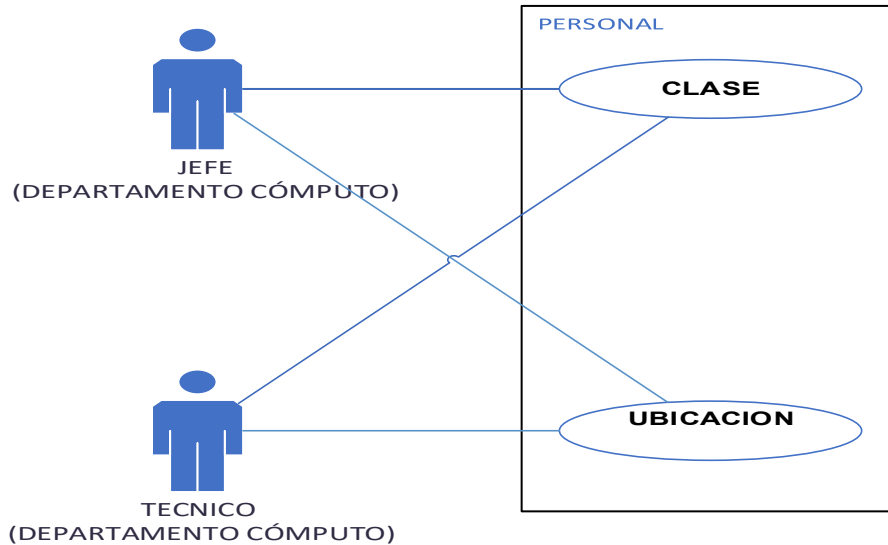


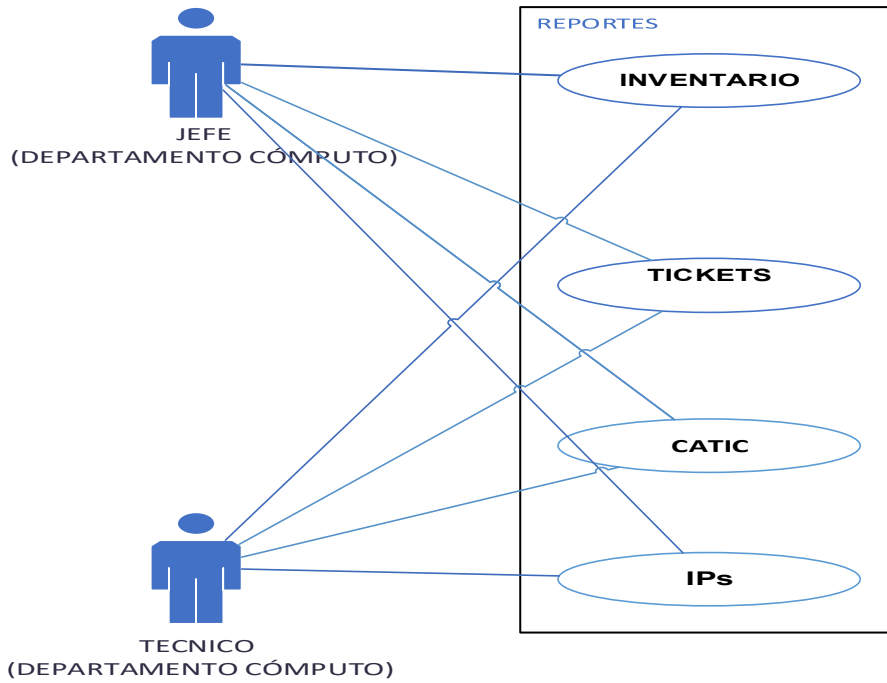
Figura 4.1.2.6. Personal

En la figura 4.1.2.7 representamos el cuadro de secuencias de personal para el caso de uso de personal.

Caso de uso:		Personal-clase		
Actores:		Jefe Departamento de Computo, técnico Departamento de Computo		
Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Ingreso al sistema proporcionando usuario y contraseña	2	Escojo la URL correcta que me lleve a la clase de personal.	
3	Reviso el personal de acuerdo a la búsqueda que necesite	4		

Figura 4.1.2.7. Cuadro de secuencias de personal

El módulo de reportes maneja los reportes generados y requeridos por el personal interesado. Generalmente quien genera los reportes es el jefe del departamento de cómputo. Ver figura 4.1.2.8.



CATIC. Consejo Asesor en Tecnología de Información y Comunicación UNAM

Figura 4.1.2.8 Reportes

En la figura 4.1.2.9 representamos el cuadro de secuencias de reportes para el caso de uso de reporte-inventario.

Caso de uso:		Reporte-inventario		
Actores:		Jefe Departamento de Computo, técnico Departamento de Computo		
Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Ingreso al sistema con usuario y contraseña	2	Acceso a la URL reportes.	
3	Mandamos el reporte a pantalla para ser guardado o imprimir			

Figura 4.1.2.9. Cuadro de secuencias de reportes

4.1.3 Diagrama de flujo de datos.

El diagrama de flujo de datos (DFD) se emplea para representar de forma gráfica los procesos que envuelven a un sistema. Los flujos de datos se simbolizan paso por paso, al igual que los procesos con sus correspondientes operacionales.

Estos diagramas permiten la comprensión de cada módulo. Por su forma gráfica, ahorran varias páginas de texto. Además, permiten identificar áreas de posibles mejoras. Se les llama diagramas de flujo debido a que utilizan flechas para indicar la secuencia de una operación.

La figura 4.1.3.1 muestra el DFD general del sistema. Más adelante se mostrarán los diferentes DFD de los módulos del sistema SIRICI.

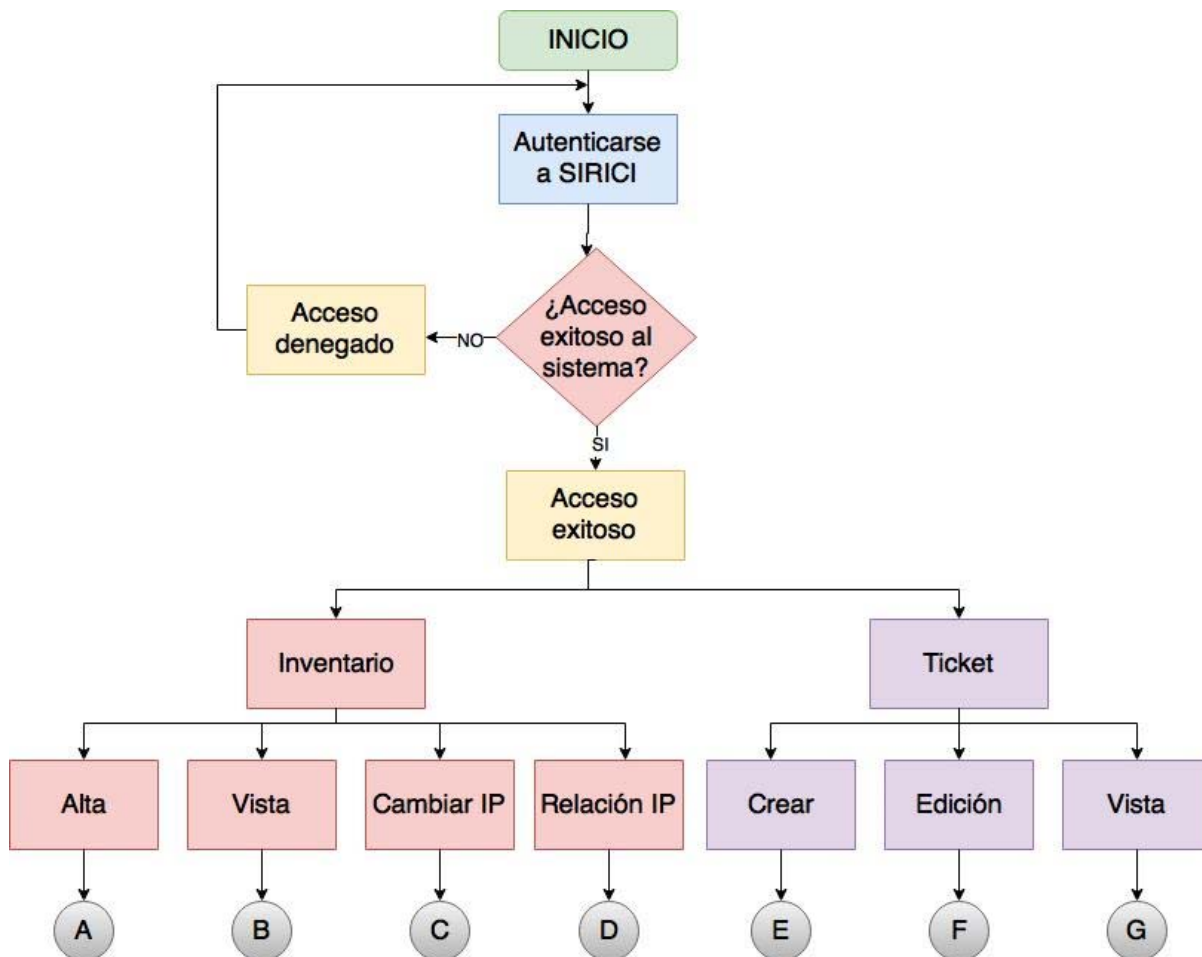


Figura 4.1.3.1 DFD general

La figura 4.1.3.2 muestra el DFD del módulo de Inventario y su proceso de alta de activos. Este módulo es esencial debido a su contribución con el cumplimiento de uno de los principales propósitos del sistema, crear un padrón de los activos de hardware.

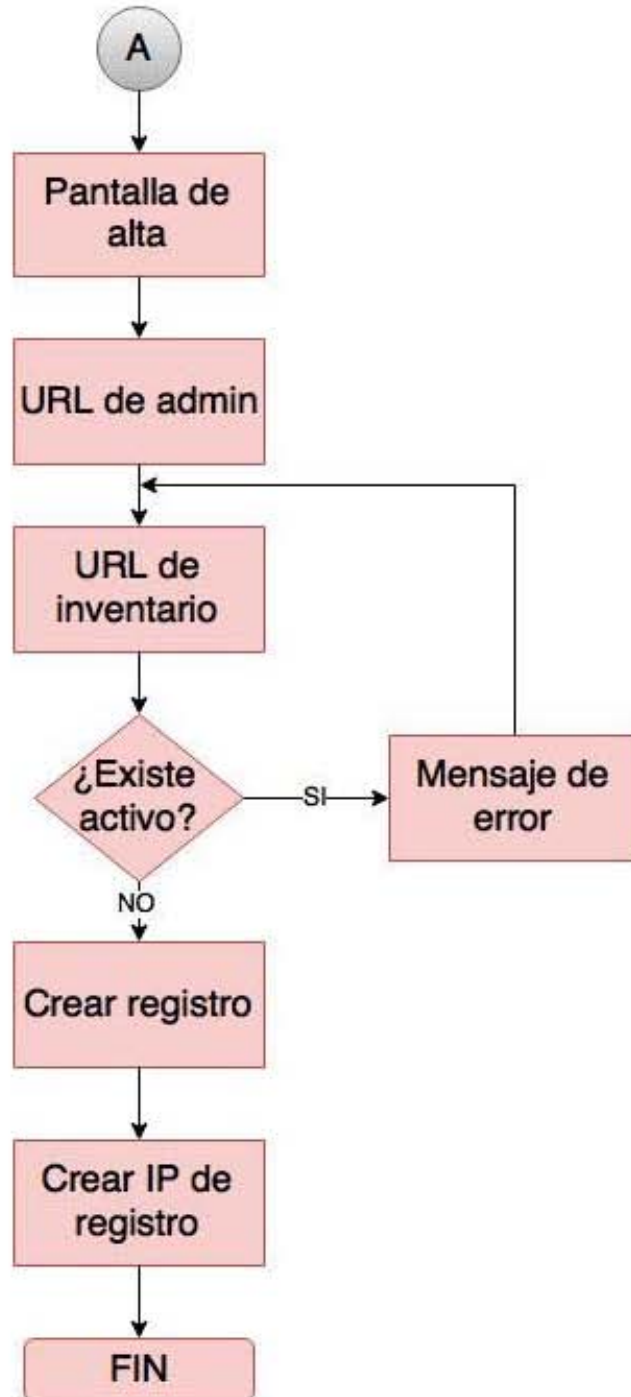


Figura 4.1.3.2 DFD A Alta (Inventario)

La figura 4.1.3.3 muestra el DFD del módulo de vista. Este módulo proporciona el acceso a consultas a jefes y unidades administrativas. El funcionamiento de este proceso es importante para la elaboración de reportes mediante la información de un listado de URLs. Por ejemplo, el censo CATIC el cual es un desglose de todos los equipos de cómputo almacenados.

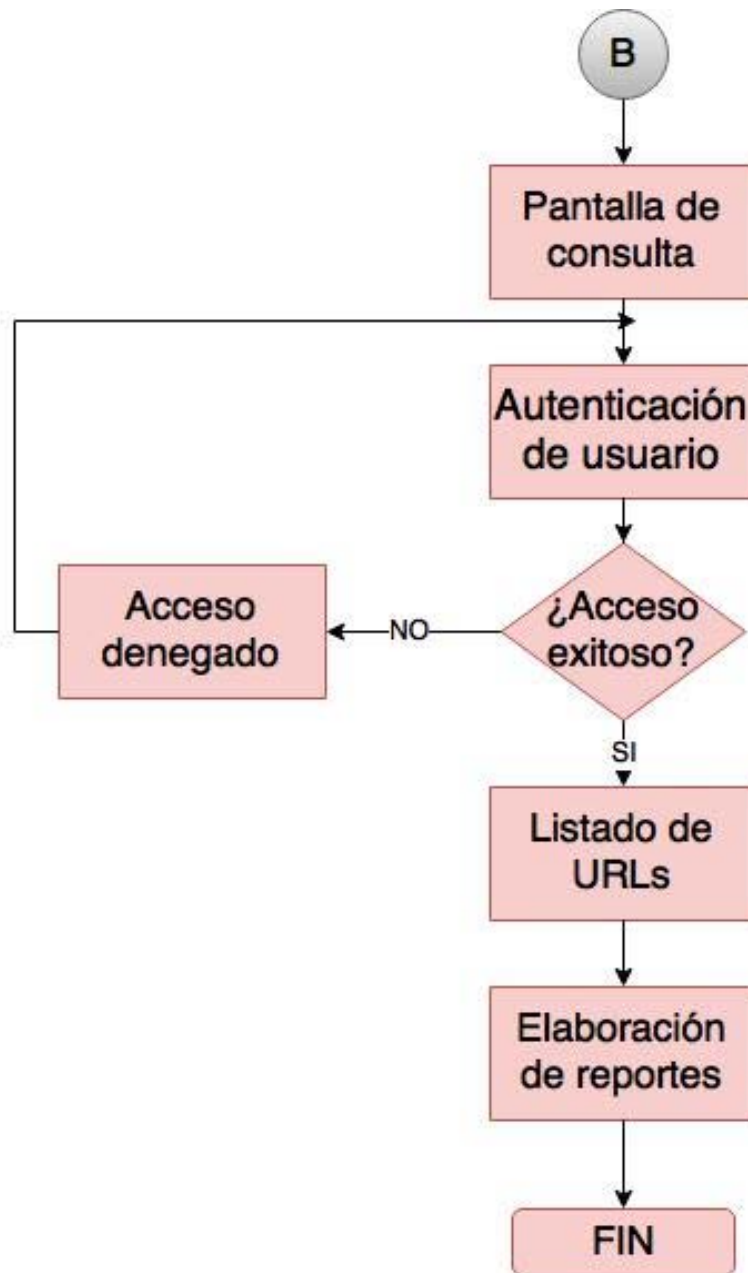


Figura 4.1.3.3 DFD B Vista (consulta) activos.

En la figura 4.1.3.4 se describe el DFD del módulo de cambio de IP. En este se da una interacción con el servidor al consumir un web service. Al servidor se le envían datos como el ID y/o categoría del activo que cuentan una IP implícita guardada en la base de datos. Además de la nueva IP que se desea asignar. Estos datos los procesa el servidor mediante una validación de identificación y disponibilidad.

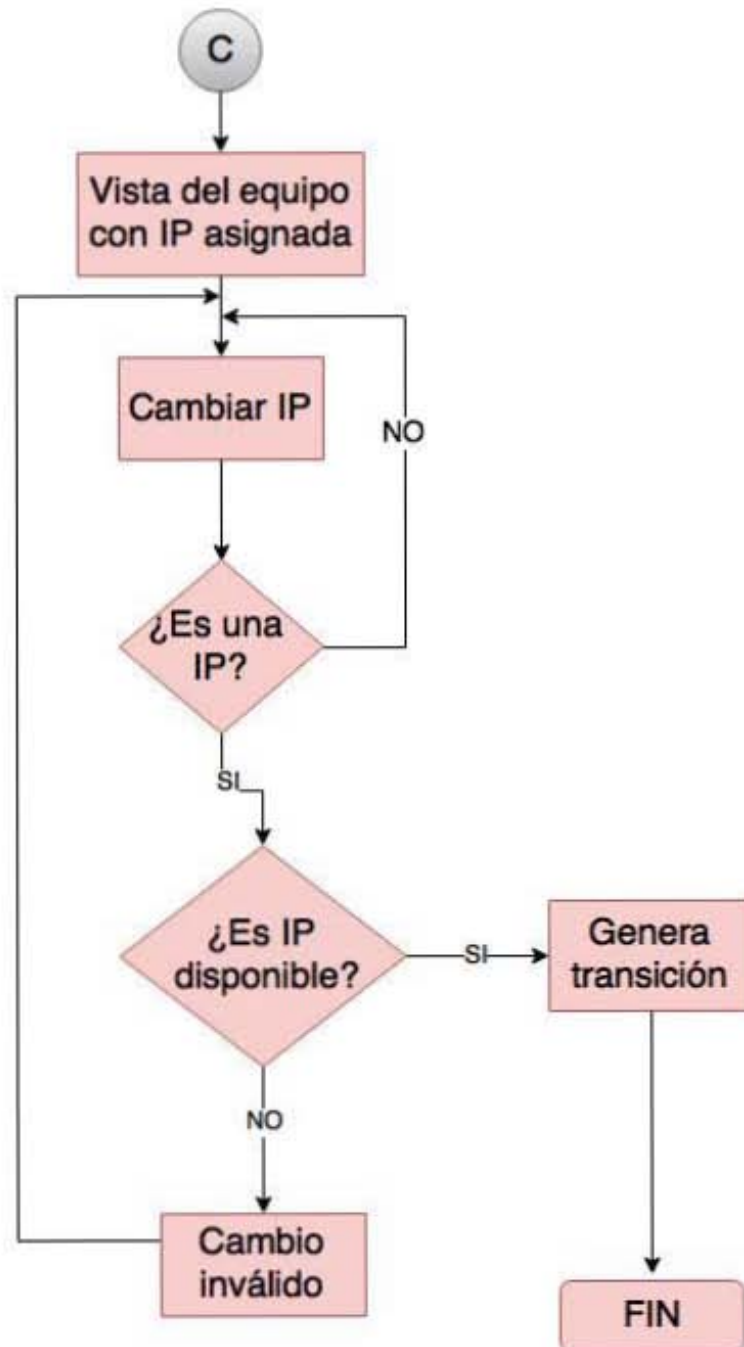


Figura 4.1.3.4 DFD C Cambio de IP

El módulo de relación IP, véase figura 4.1.3.5 consiste en la elaboración de un reporte con datos como la IP asignada a un activo, el número de inventario, activos en uso, usuarios que están utilizando un activo en específico y el tipo de usuario.



Figura 4.1.3.5 DFD D Relación IP

Un ticket es una herramienta de consulta que se realiza desde el área del cliente. Las consultas quedan registradas automáticamente y se pueden visualizar. De acuerdo a los

protocolos del Departamento de Cómputo, la secretaria es la encargada de crear tickets y el personal del departamento de atenderlos. El módulo de tickets administra y mantiene una lista de incidencias conforme sean requeridas. Véase figura 4.1.3.6.

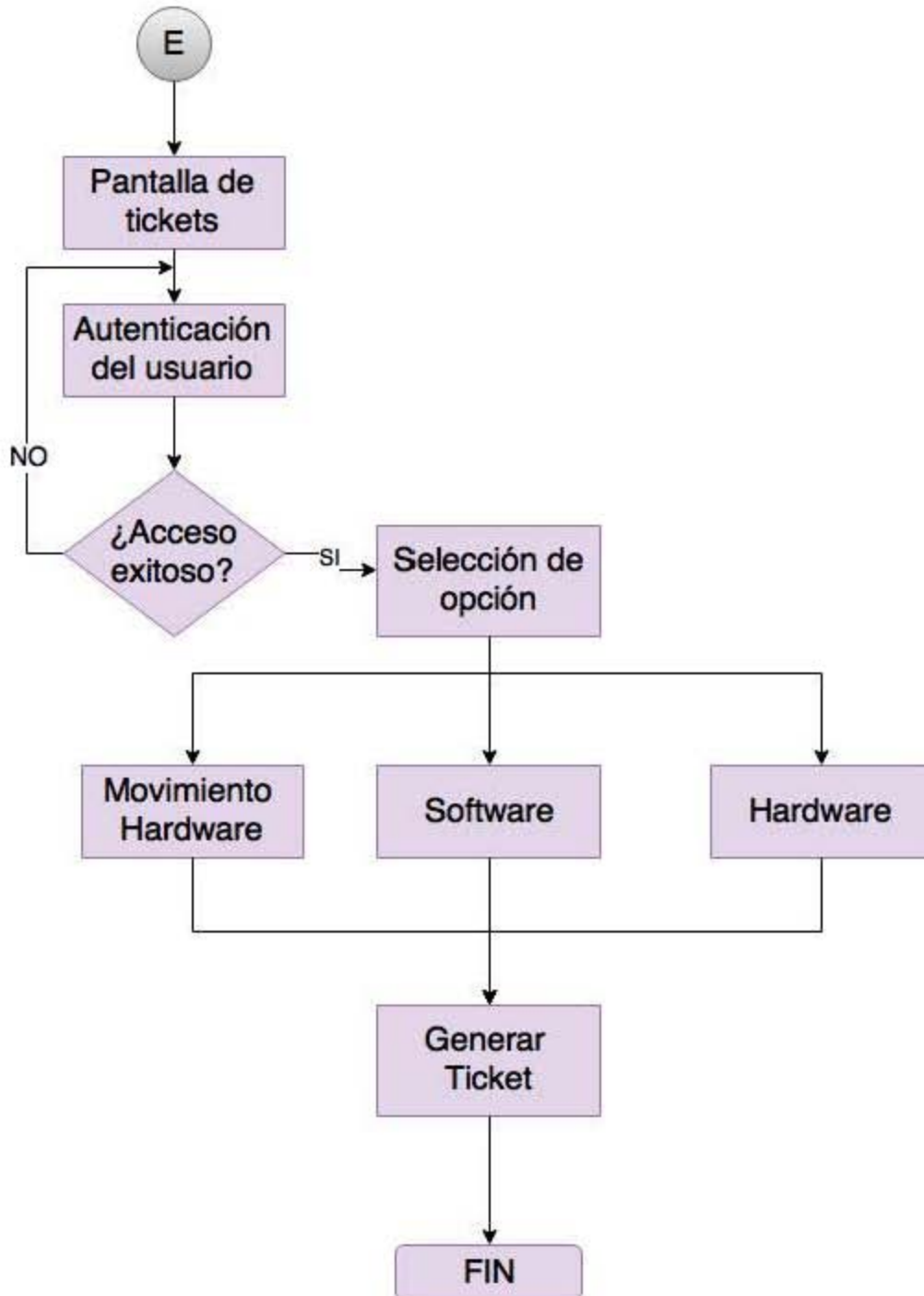


Figura 4.1.3.6 DFD E Crear ticket

En la figura 4.1.3.7 se muestra el módulo de edición de un ticket en donde se modifica la información sobre los eventos de soporte a solicitud del personal del Departamento de Cómputo.

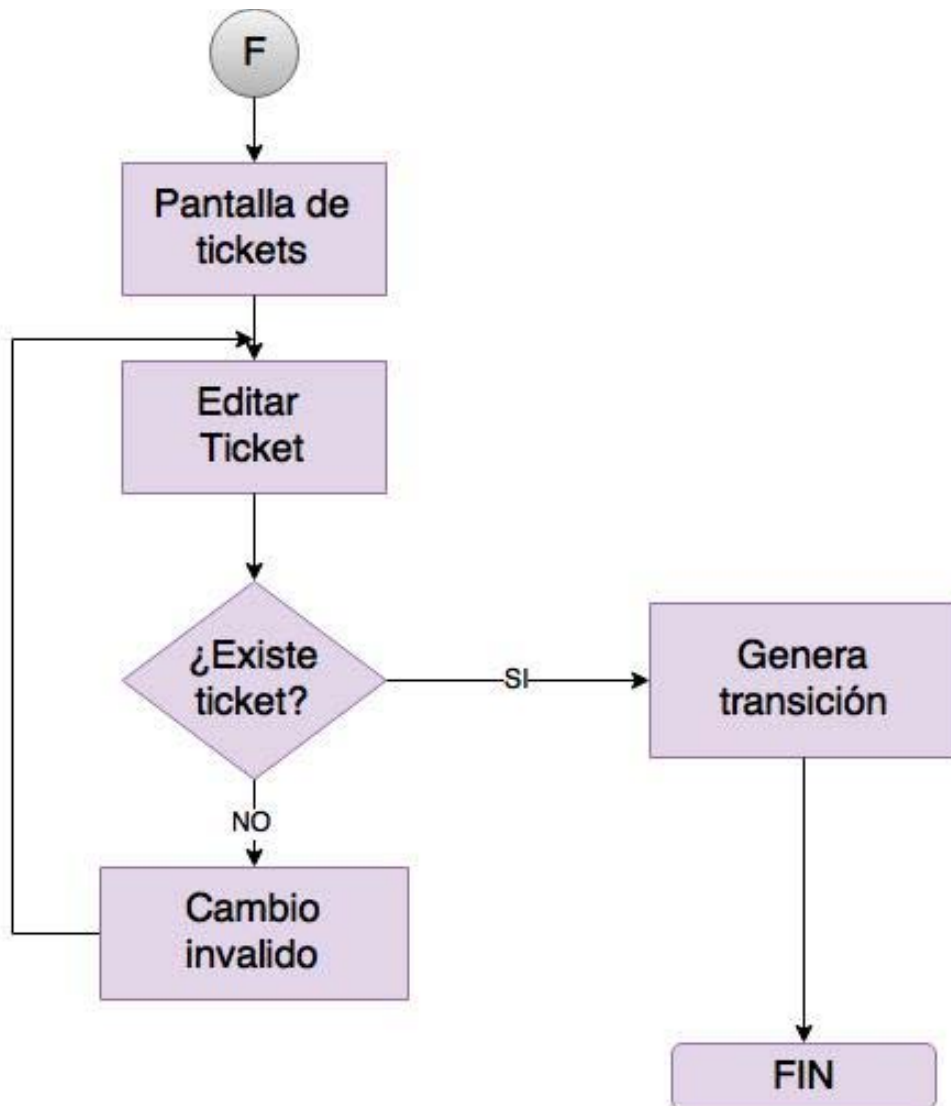


Figura 4.1.3.7 DFD F Edición de un ticket

Para llevar a cabo la tarea de administrar las incidencias, el sistema cuenta con un módulo de consulta de tickets en donde se da seguimiento al soporte solicitado a través del responsable, fecha, tipo de soporte, activo involucrado o si es un caso resuelto. Véase figura 4.1.3.8

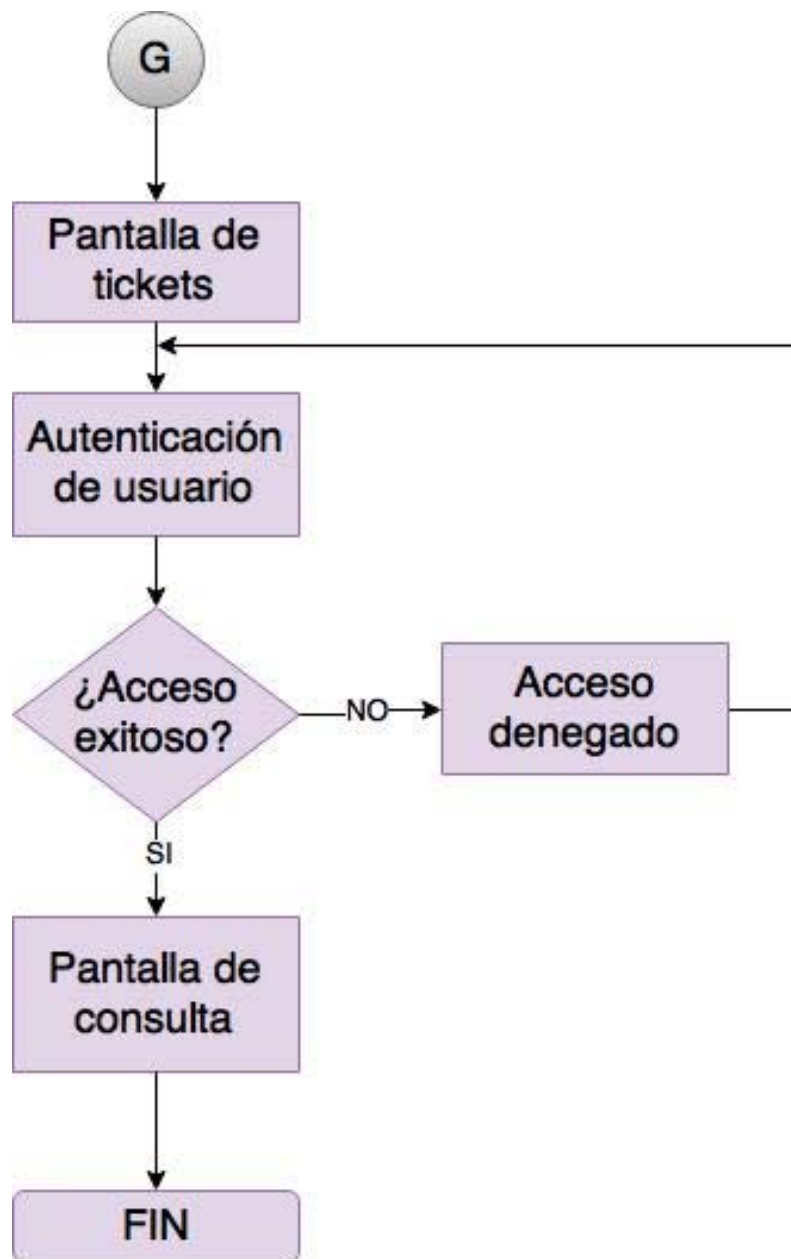


Figura 4.1.3.8 DFD G Consulta de ticket

4.1.4 Diagrama de Componentes.

Este tipo de diagrama representa los componentes y las dependencias entre estos dentro de un sistema. Los componentes incluyen archivos, cabeceras, bibliotecas compartidas, módulos ejecutables o paquetes. Además, son utilizados para modelar y documentar cualquier arquitectura del sistema.

A diferencia de los diagramas de casos de usos, los diagramas de componentes son utilizados para modelar la vista estática y dinámica. Generalmente no incluyen todos los componentes de un software. Su principal utilidad es la consulta de componentes que pueden compartirse entre sistemas.

En Python se tienen archivos .py que al pasar por el intérprete generan archivos .pyc, similar al lenguaje C con los archivos .ccp y .exe, y estos mantienen una correspondencia con la base de datos. Asimismo, existen bibliotecas como wxPython en donde se desarrolla toda la interface gráfica de usuario.

La plataforma en Django se compone de dos aplicaciones (inventario y ticket) que comparten los mismos archivos de control. Django recibe peticiones y enseguida valida la ruta solicitada mediante el procesamiento de los archivos urls.py. En el sistema existe un archivo maestro con sus archivos urls.py para cada una de sus apps. Si la ruta solicitada es válida, el archivo urls.py indica la rutina a llevarse a cabo en los archivos views.py. Estas rutinas suelen emplear objetos de datos a través el archivo models.py.

Django crea un módulo de administrador, éste provee de interfaces que permiten llevar a cabo las funciones de alta, cambios y listados, así como la administración de los usuarios del sistema. Véase figura 4.1.4.1

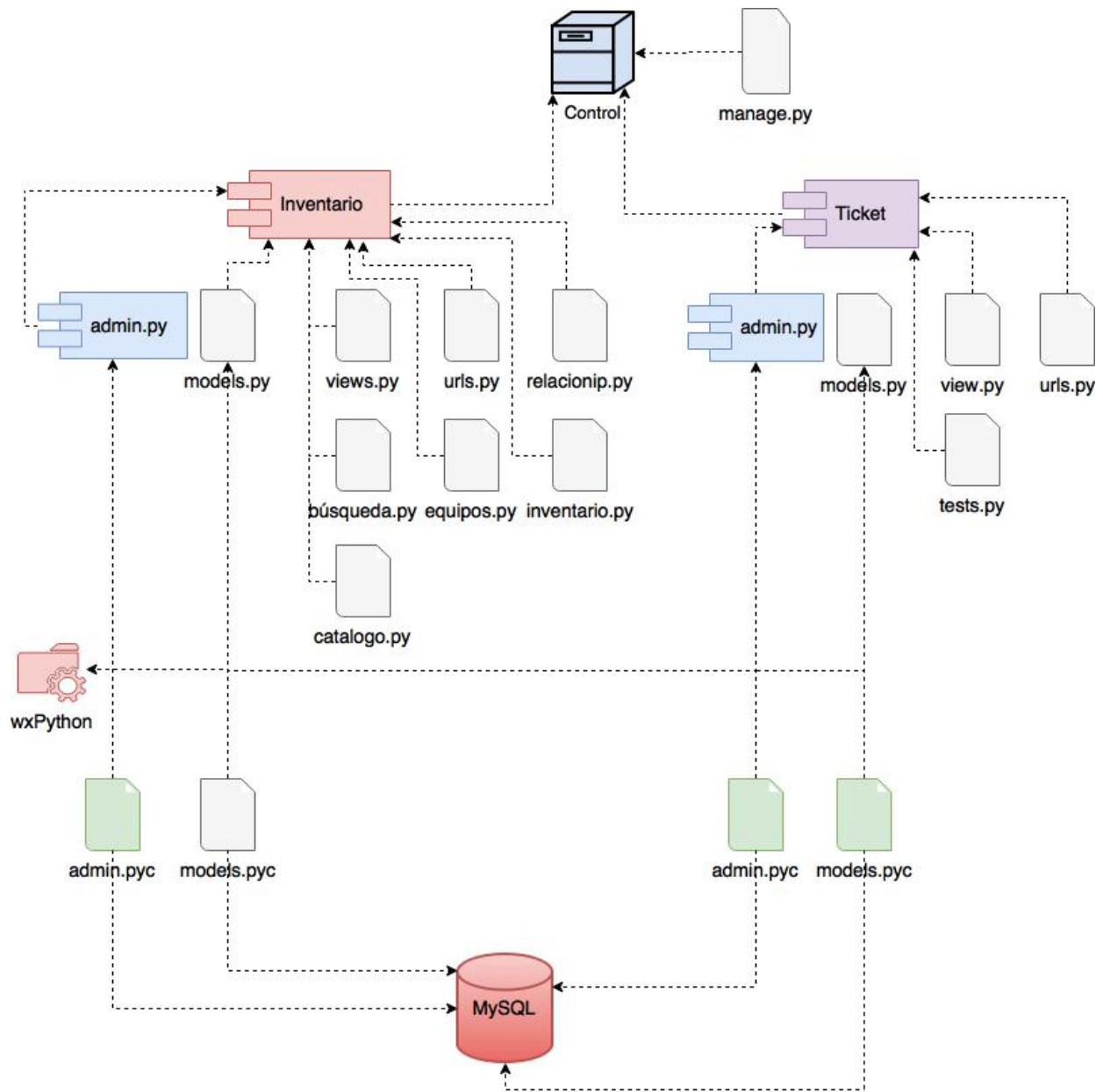


Figura 4.1.4.1 Diagrama de Componentes General. Dependencias generales entre componentes del sistema

Para la creación de interfaces gráficas de usuario en Python para aplicaciones de escritorio se pueden elegir herramientas como PyGTK, PyQt y wxPython.

La biblioteca wxPython es un binding¹⁵ de la biblioteca gráfica wxWidgets nativa de Python. La biblioteca wxWidgets se caracteriza por ser multiplataforma, por lo que su uso permite el desarrollo de aplicaciones gráficas para diferentes sistemas operativos. Funciona como un módulo más del sistema, debiendo ser importado al igual que cualquier otro. Entre sus principales características pueden citarse:

- Portabilidad.
- Apariencia. Nativa en todos los sistemas operativos.
- Es libre.

WxPython está formado de un módulo núcleo llamado wxObject el cual es la base para todas las clases de la API. Éste consiste de clases elementales, como geometría básica, eventos y **sizers (dimensionadores)** para el desarrollo de componentes.

Además, existe un módulo de control el cual contiene los controles necesarios en el desarrollo de aplicaciones **GUI (Interfaz Gráfica de Usuario)**. En esta biblioteca es donde se desarrolla toda la interfaz gráfica del sistema, por medio de esta se generan las ventanas con sus respectivos componentes internos como botones, cajas de texto, listados, etc.

El módulo wx.Window consiste de varias ventanas que forman una aplicación como paneles, cuadros de diálogo, capas y marcos. Asimismo, en los archivos Misc residen otras clases variadas y módulos de funciones utilizados para registro, configuración de sistema y aplicaciones. Véase figura 4.1.4.2

¹⁵ Un binding es una adaptación de una biblioteca para ser usado en un lenguaje de programación distinto de aquel en el que ha sido escrito. Tomado de: <https://wiki.wxpython.org/ProjectPhoenix>

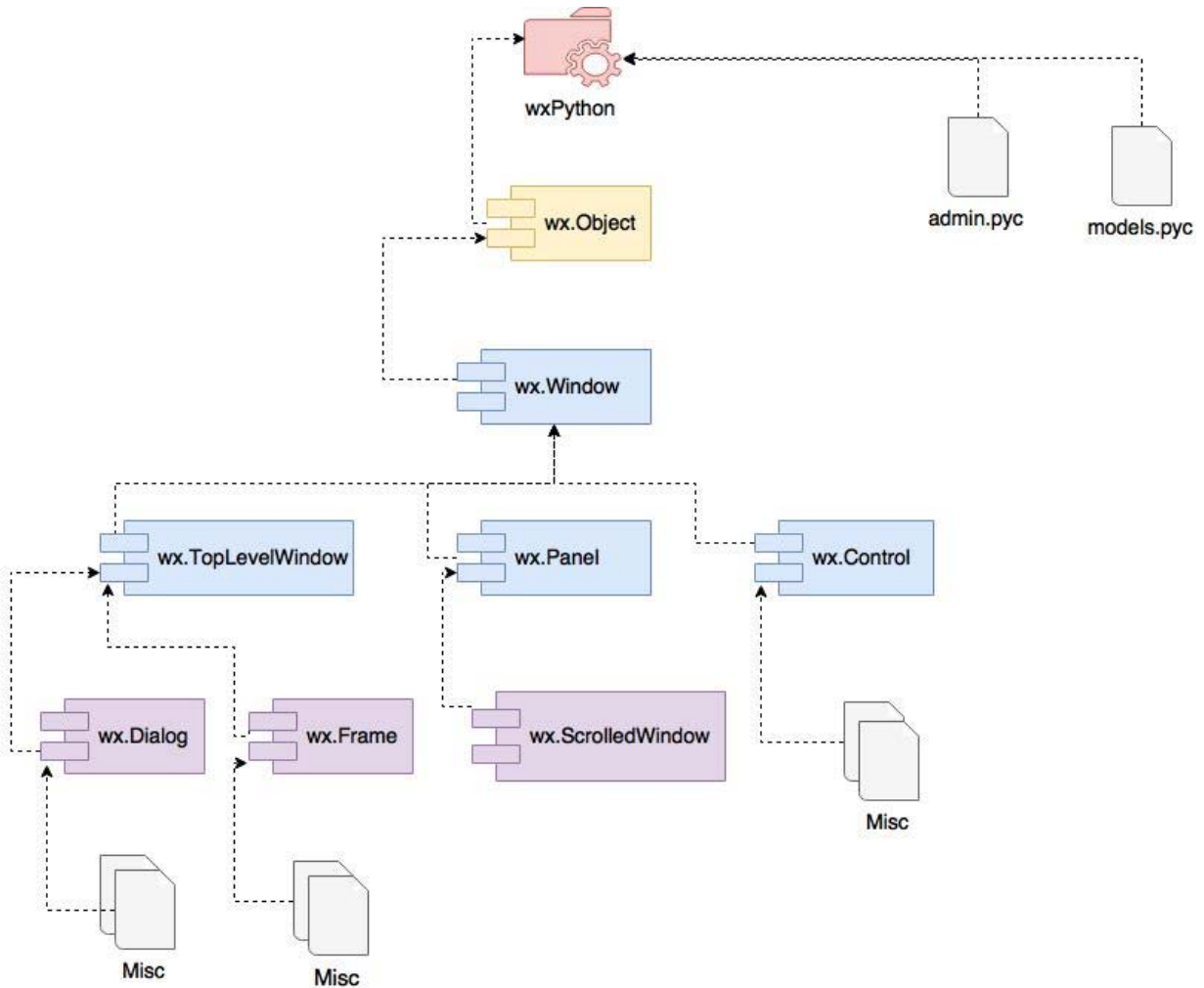


Figura 4.1.4.2 Jerarquía de clases de WxPython

Ventajas de WxPython.

- Contiene un completo conjunto de elementos gráficos.
- Ofrece un flexible control del comportamiento de la interface.
- Apariencia nativa.
- Tiene 12 bibliotecas y utilitarios complementarios.
- No esta orientado a ningún entorno.
- Soporta las características comunes de Windows y las emula en Linux y Mac OSX.
- Es sencillo armar componentes personalizados, tanto que incorpora widgets escritos en Python.

- Existe una gran documentación y comunidad de usuarios.

Desventajas de la biblioteca WxPython.

- No viene preinstalado con Python.
- Se liberan nuevas versiones frecuentemente, lo que en la práctica le confiere problemas de compatibilidad.
- En proyectos grandes presente inestabilidad.

4.1.5. Diagrama de despliegue

El diagrama de despliegue es el que se muestra en la figura 4.1.5.1. La función principal de este diagrama es desplegar los diferentes elementos físicos y lógicos del sistema.

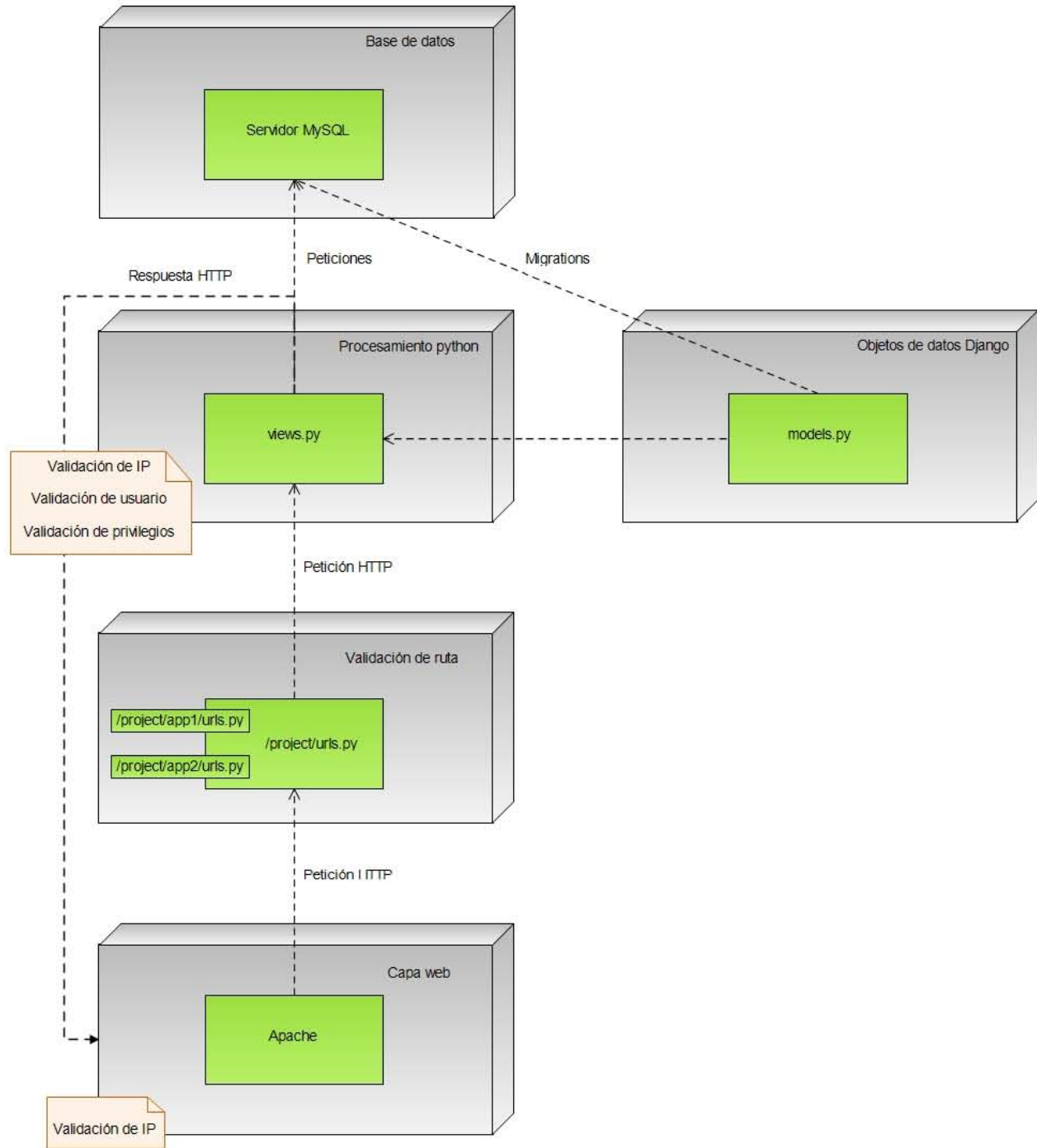


Figura 4.1.5.1. Diagrama de despliegue.

En el caso específico de SIRICI, todo el sistema y su infraestructura está contenida en un solo servidor virtual, que a su vez es parte de un único servidor físico. Para propósitos del diagrama, sin embargo, únicamente tomamos en cuenta el servidor virtual, sin distinción por la calidad virtual de su existencia.

Como muchos otros sistemas web, SIRICI es un sistema desarrollado principalmente en 3 capas: la capa de interfaz de usuario (*Capa web*), la capa de lógica de negocio (*Procesamiento python*) y la capa de base de datos. Estas capas se pueden observar claramente en el diagrama. La capa web es administrada por Apache, aunque también incluye programación de plantillas HTML para el despliegue de la información, así como construcciones automáticas como el módulo de administración de Django; la capa de base de datos cuenta con MySQL, como sabemos, y es abstraída por Django a través de dos interfaces principales:

- La creación de la base se realiza a través de la definición de un archivo de clases de Python (*models.py*), que se traducen a tablas de la base de datos.
- La interacción con la base de datos se realiza a través de métodos de Python definidos en las bibliotecas de Django, que operan sobre los objetos definidos en el archivo *models*.

En ambos casos, lo único que debemos especificar a Django (a través del archivo central de configuración *settings.py*) son las credenciales de la base de datos, así como el motor de la misma. Django soporta diferentes motores de bases de datos, no solo MySQL.

La lógica de negocio es el grueso de la programación, y se centra en los archivos *views.py*. Esta capa es donde realizamos tres procesos de seguridad importantes: validamos que la dirección IP del usuario sea una de las direcciones autorizadas para realizar algún proceso, que el usuario tenga una sesión válida en el sistema, y que su usuario tenga los privilegios apropiados para la operación que desea realizar. Este proceso se realiza en cada uno de los métodos descritos en los archivos *views.py*.

Django presenta, además, una capa adicional entre web y lógica de negocio: la capa de validación de ruta, donde Django realiza un procesamiento para determinar los scripts que deben ser ejecutados al requerírsele una dirección específica. Django realiza este procesamiento a través de un archivo *urls.py* general al proyecto, el cual puede utilizar uno o más archivos especificados en el mismo, aunque se sugiere una estructura de al

menos un archivo *urls.py* por cada *app* dentro del proyecto. Recordemos que nuestro sistema es un proyecto llamado *control* con dos apps: *inventario* (para el control de activos de hardware) y *ticket* (para el control de las solicitudes de soporte).

4.1.6. Diagrama de clases

El diagrama de clases se muestra en la figura 4.1.6.1. Las relaciones en este diagrama describen las interacciones entre los objetos de datos que Django utilizará como una abstracción de la base de datos.

Los objetos que representan los activos de hardware se resaltan con un fondo color azul. Estos activos son el elemento central del diseño de la base de datos, aunque por razones de espacio no se encuentran físicamente en el centro de la figura 4.1.6.1.

Alrededor de estos objetos se construye una serie de objetos de soporte, que permiten a la base almacenar las diferentes características solicitadas por la Escuela. Catálogos como marca, categoría y ubicación se utilizan para una multitud de objetos, algunos siendo comunes a todos los objetos, algunos a la mayoría, y algunos específicamente diseñados para un objeto en particular.

Las relaciones se representan agregando una leyenda en el objeto que contiene la referencia, a un costado del conector que representa la referencia. Esta leyenda corresponde al nombre del atributo en el objeto de datos, y este nombre es representativo también del objeto de datos al que hacemos referencia.

Por razones de complejidad en el diseño, no se puede generar una mayor claridad en los diagramas de relación, sin sacrificar el tamaño total de la imagen (y por ende su legibilidad). El detalle de las relaciones está disponible en el apartado 4.2.

Se agregan además los objetos de datos intermedios que surgen de una relación de muchos a muchos, con una posición intermedia entre los objetos que componen a la relación.

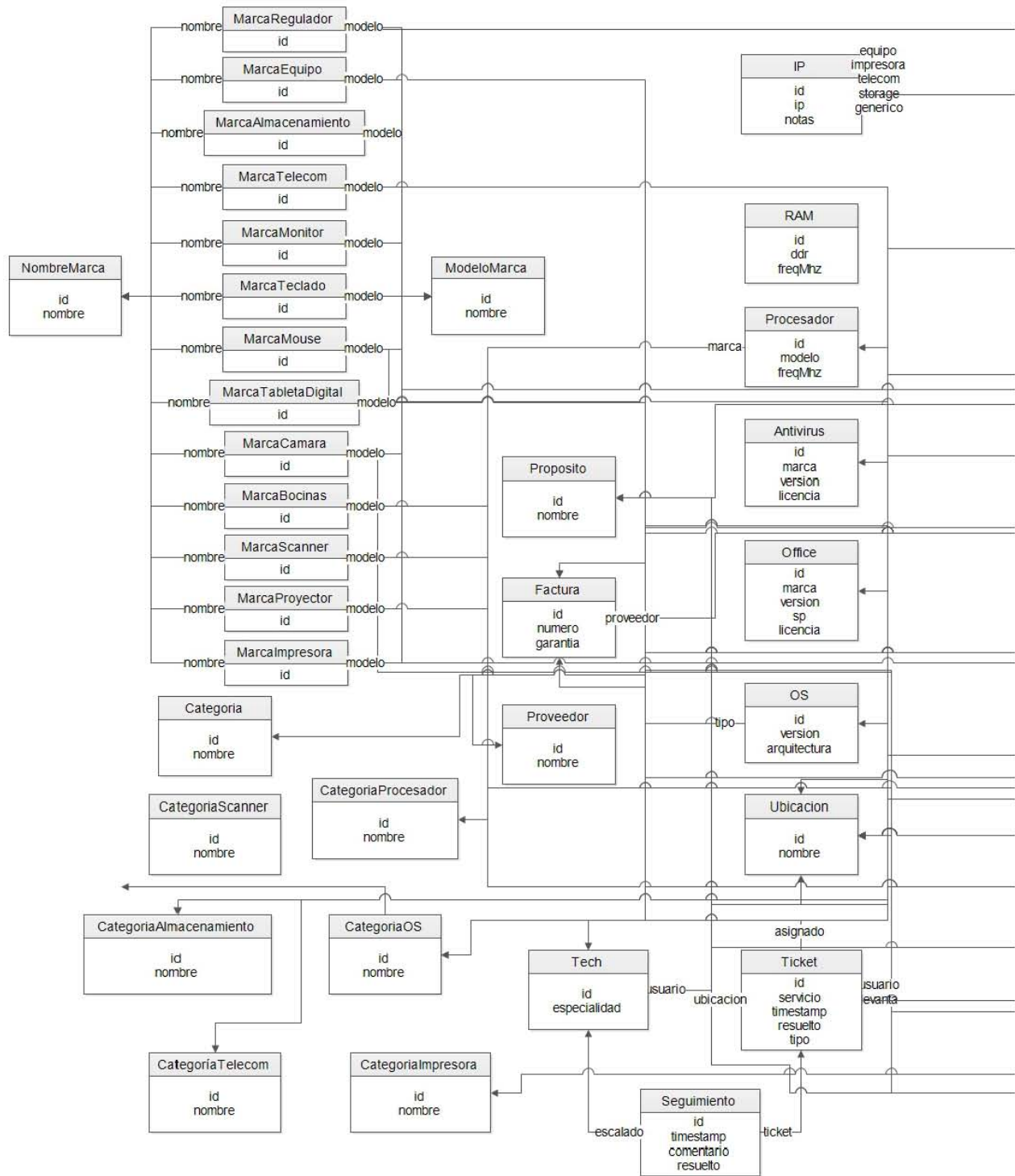


Figura 4.1.6.1.

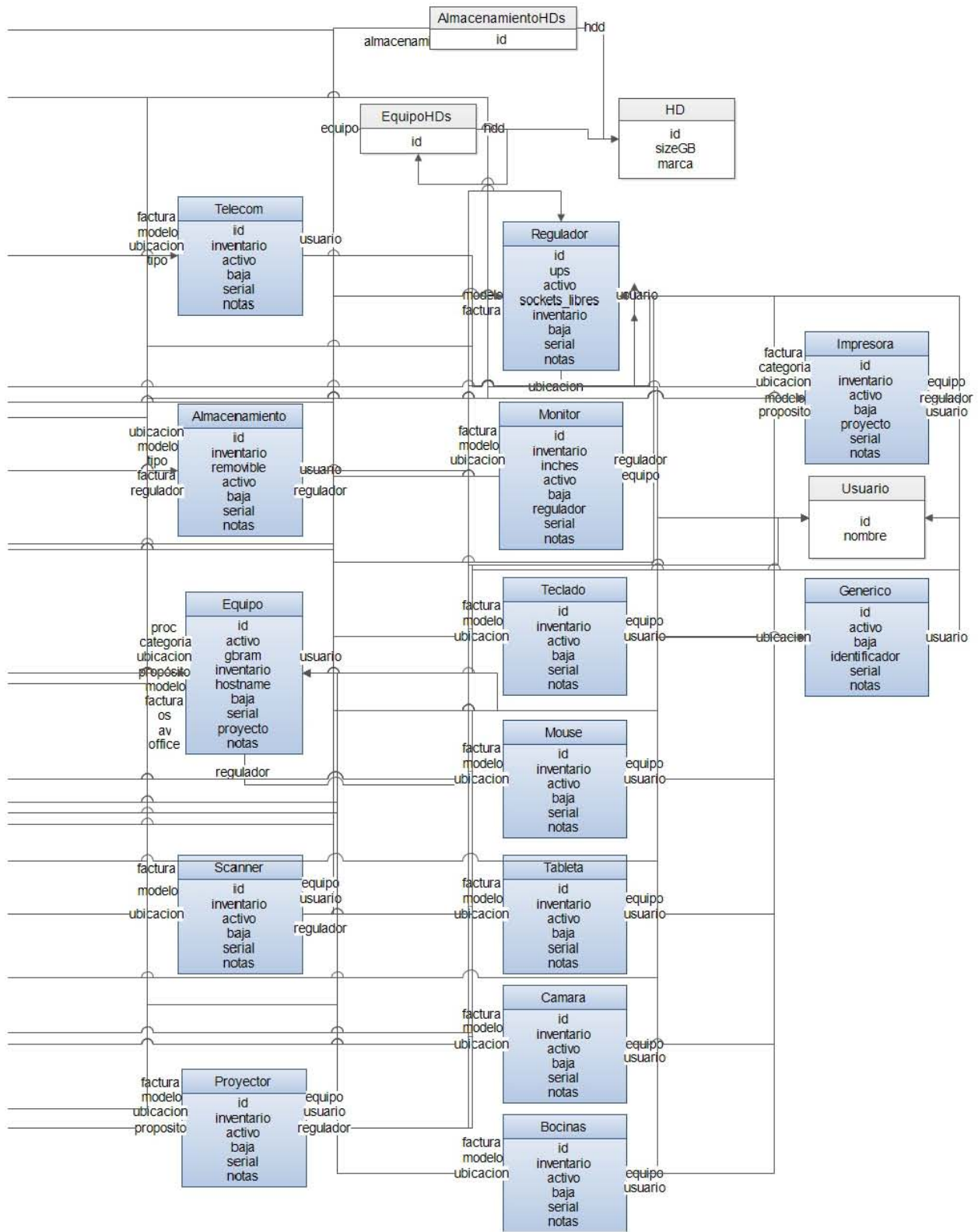


Diagrama de clases

A modo de ejemplo, analizaremos el objeto *Impresora*.

Impresora cuenta con los atributos comunes a la mayoría de los activos de hardware:

- inventario (número de inventario)
- activo (para determinar si se ha dado de baja)
- baja (atributo donde registramos la fecha cuando se dio de baja)
- proyecto (al que está asignada, en caso de aplicar)
- serial (número de serie)
- notas (para cualquier comentario adicional)

Además, *Impresora* cuenta con referencias que nos dan más información sobre el activo:

- factura (para propósitos de contabilidad, etc.)
- categoria (puede ser laser, inyección de tinta, gran formato...)
- ubicacion (en la que se encuentra)
- modelo
- proposito (académico, administrativo...)
- equipo (al que está conectada, de aplicar)
- regulador (al que está conectada, de aplicar)
- usuario (que la está utilizando)

4.2 Diseño, construcción e implementación de la base de datos

La base de datos es el componente fundamental de un sistema, ya que en ella residirá la información que el sistema procesará, almacenará y presentará al usuario. Un buen diseño de base de datos es esencial para el correcto funcionamiento de cualquier solución de software que procese información, y por ello dedicamos una sección completa al desglose de la misma en la solución SIRICI.

En el caso de este sistema en específico, debemos lidiar por un lado con los componentes fundamentales de información a procesar (i.e. los activos de hardware que se desean inventariar, así como las solicitudes de soporte que pueden o no involucrar a alguno de estos activos) así como con las estructuras que le dan soporte a la información. Algunas de estas son generales a todos los activos de hardware: por ejemplo, el usuario al que están asignados, o la ubicación donde se encuentran. En otros casos, son generales en concepto, pero no en aplicación: la marca y la categoría del activo de hardware, por ejemplo, es un concepto aplicable a todos los activos, pero no se almacenarán en la misma tabla, sino que se generarán tablas separadas para cada tipo de hardware; primordialmente, para optimizar el mantenimiento de estos catálogos por personal debidamente calificado. Finalmente, algunas estructuras son altamente específicas: por ejemplo, únicamente los equipos de cómputo (CPUs) requieren registrar qué tipo de RAM tienen instalado, o el sistema operativo de los mismos.

Recordemos los activos de hardware esenciales que la Escuela nos ha indicado que desea registrar:

- **Equipos** de cómputo (CPUs, laptops, equipos todo en uno).
- **Monitores**
- **Teclados**
- **Mouses**
- Dispositivos de **almacenamiento** (discos duros externos, SAN, etc.)
- Dispositivos de **telecomunicaciones** (routers, switches, Access points, etc.)
- **Reguladores**
- **Tabletas** digitales (para diseño gráfico: i.e. no tablets)

- **Cámaras**
- **Bocinas**
- **Proyectores**
- **Scanners**
- **Impresoras**

Éstos son los objetos centrales de la base de datos. Asimismo, el registro de las solicitudes de soporte se centra sobre un objeto principal: **tickets**.

Debido al número de tablas que maneja el diseño de la base, la presentación de los diagramas de esta sección se hará de forma simplificada. Una versión exhaustiva puede encontrarse en la sección 4.1.6. *Diagrama de clases*, donde puede apreciarse la necesidad de una versión simplificada por legibilidad y claridad de los modelos.

4.2.1 Diagramas entidad-relación

A continuación, mostraremos los diagramas entidad relación correspondientes a los dos temas relevantes a este sistema.

El primero y más sencillo es el diagrama correspondiente a las solicitudes de soporte, centradas alrededor del objeto **ticket**. Este diagrama se aprecia en la figura 4.2.1.1. La estructura se puede describir de manera básica de la siguiente manera:

- Un ticket puede lidiar con uno o más activos de hardware, de uno o más tipos diferentes. Esto se describe en las relaciones periféricas (donde se han diagramado los objetos correspondientes a los diferentes tipos de activos de hardware).
- Un atributo clave del ticket es la fecha y hora de creación, representado por el atributo **timestamp**.
- Un ticket también tiene asignado un miembro del personal técnico capaz de atender estas solicitudes, representado por el objeto **Tech**. Un miembro de este equipo puede tener cero, uno o más tickets asignados en cualquier momento.
- Para cada ticket, pueden existir uno o más **seguimientos**. Los seguimientos permiten escalar el ticket a otro miembro del personal con más especialidad en el

tema de que se trate (y por ello referencian a un Tech), registrar el proceso de la solicitud a través de diferente personal y deben contar también con un atributo de fecha y hora de escalamiento, representado por el atributo **timestamp**.

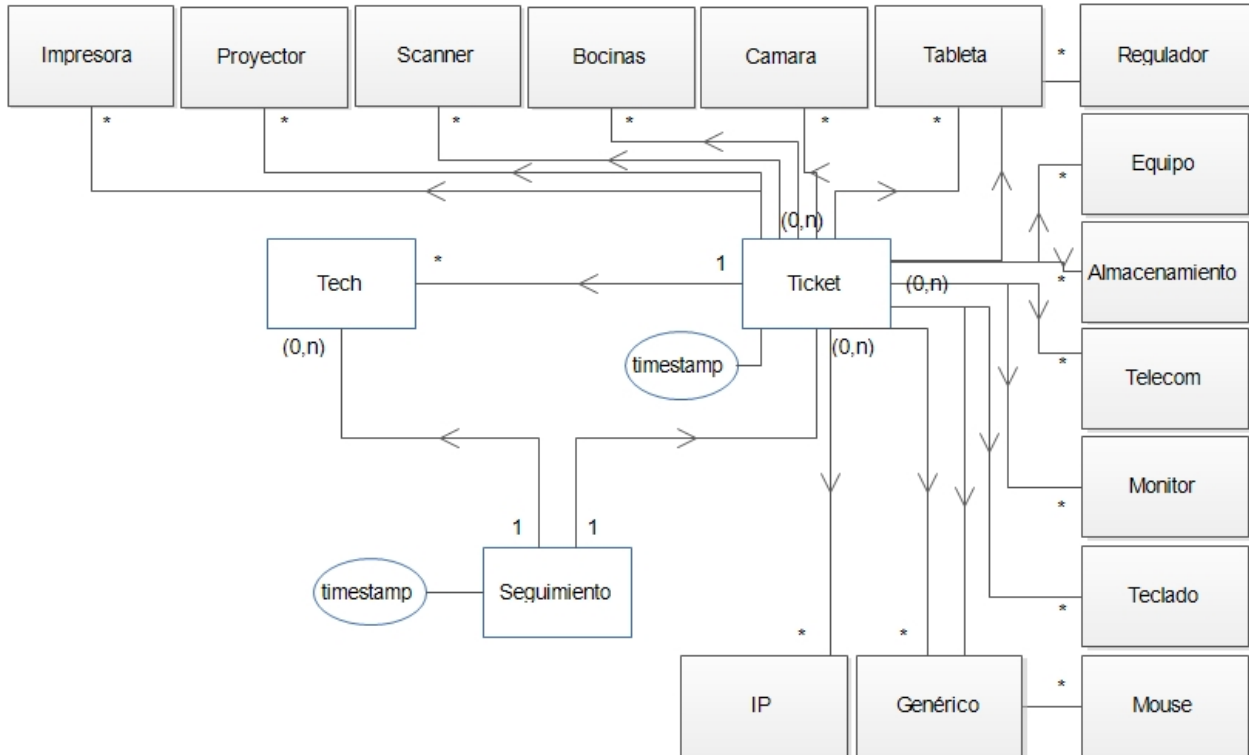


Figura 4.2.1.1. Diagrama entidad relación para TICKET.

En segundo lugar, tenemos el diagrama entidad relación para los activos de hardware de la Escuela. Este diagrama se puede apreciar en la figura 4.2.1.2. Su estructura se puede describir como sigue:

- Se deja representada la relación entre los diferentes tipos de activo y el ticket.
- Además de los activos de hardware ya expresados, existen algunos objetos y/o conceptos que requieren ser registrados en el sistema, pero que no son propiamente activos de hardware. La Escuela nos dio dos ejemplos específicos: la subestación eléctrica, que no pertenece a la Escuela pero que sí tiene asignada una dirección IP (ver siguiente punto), y una reservación de 3 direcciones IP en cada segmento. Estos conceptos u objetos se registran en un objeto de datos nombrado **Generico**.

- Algunos de los tipos de activo de hardware – específicamente, **Impresora, Equipo**, dispositivos de **almacenamiento** y **telecomunicaciones**, así como los dispositivos genéricos – pueden tener asignada una dirección IP homologada. La Escuela cuenta con dos segmentos de red clase C asignados.
- Cada activo de hardware tiene atributos comunes, denotados en un único recuadro en la parte superior del diagrama. Estos atributos pueden ser representados por una tabla (**Ubicacion, Proposito, Usuario, Factura**) o bien con una tabla por cada tipo de activo de hardware: **Marca, Categoria**. Por ejemplo, existen las tablas MarcaRegulador, así como MarcaTableta, etc.
- Finalmente, **Equipo** requiere catálogos únicos a este tipo de hardware: **OS, RAM, Procesador, Antivirus, Office**.

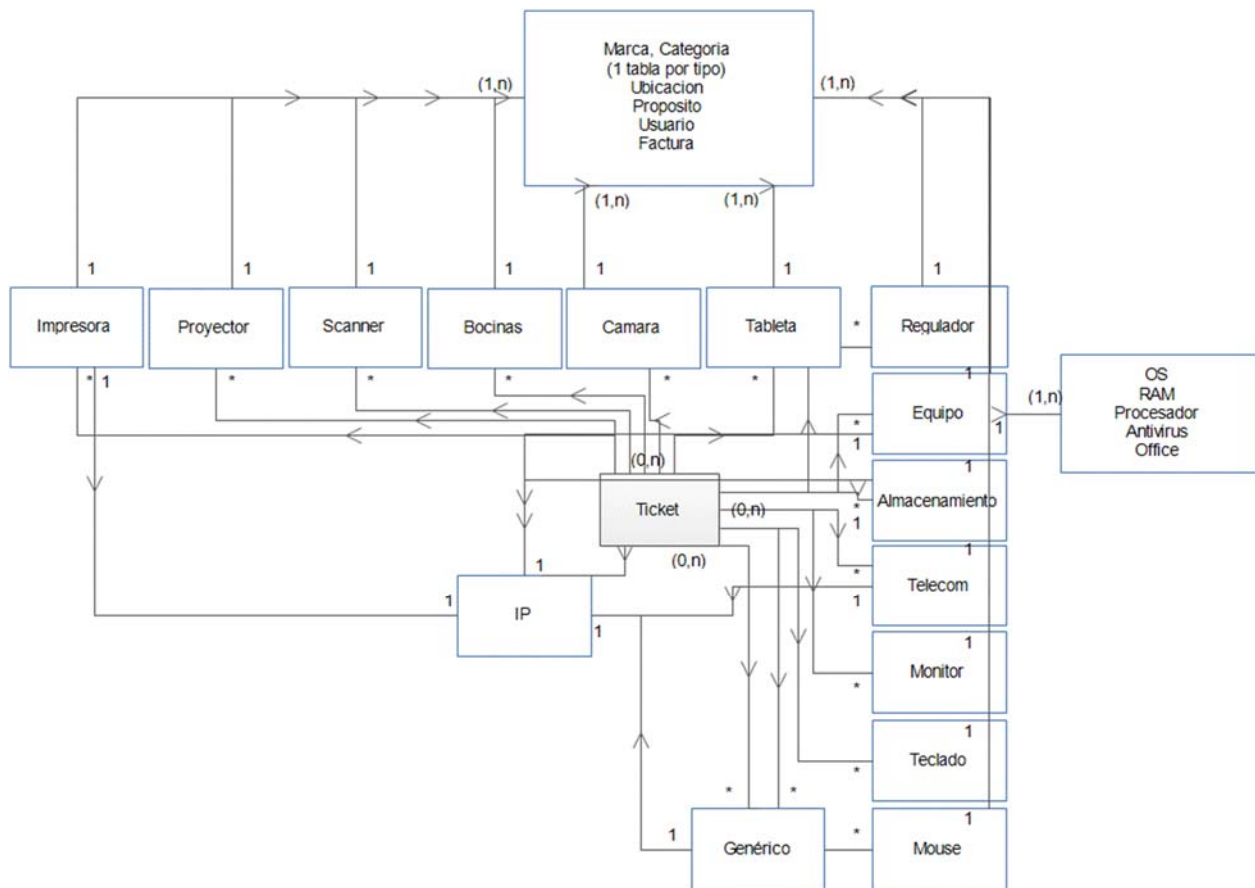


Figura 4.2.1.2. Diagrama entidad relación para INVENTARIO.

4.2.2 Diccionario de datos

A continuación, se listan los diferentes atributos de cada una de las tablas, comenzando por las más específicas y terminando con las generales. El nombre de la tabla se indica en la primera celda, seguido de una descripción de la misma, ambos resaltados en azul. Las llaves primarias se resaltan en verde. Las llaves foráneas se resaltan en dorado. Las relaciones MtM (*Many to Many*, Muchos a muchos) se resaltan en naranja. Las tablas 4.2.2.1-4.2.2.47 corresponden al control de activos de hardware (inventario); 4.2.2.48-4.2.2.50 corresponden al manejo de solicitudes de soporte (tickets).

Tabla 4.2.2.1 - Proveedor						
Catálogo de proveedores de la ENALLT						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	300	charfield	-	-	-	Nombre del proveedor

Tabla 4.2.2.2 - NombreMarca						
Catálogo de nombres de marcas (e.g. Dell, HP, etc.)						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre de la marca

Tabla 4.2.2.3 - ModeloMarca						
Catálogo de modelos (e.g. Laserjet 3100)						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre del modelo

Tabla 4.2.2.4 - CategoriaProcesador						
Catálogo de categorías de procesador (e.g. Intel Pentium D, Intel Core i7, etc.)						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Categoría de procesador

Tabla 4.2.2.5 - CategoriaOS						
Catálogo de categorías de sistemas operativos (e.g. Windows, Linux)						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre de la categoría

Tabla 4.2.2.6 - CategoríaAlmacenamiento						
Catálogo de categorías de dispositivos de almacenamiento (e.g. DVR, HDD externo)						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre de la categoría

Tabla 4.2.2.7 - CategoriaTelecom						
Catálogo de categorías de dispositivos de telecomunicaciones						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre de la categoría

Tabla 4.2.2.8 - Categoriainpresora						
Catálogo de categorías de impresoras						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre de la categoría

Tabla 4.2.2.9 - CategoriaScanner						
Catálogo de categorías de dispositivos de tipo scanner						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre de la categoría

Tabla 4.2.2.10 - Ubicación						
Catálogo de proveedores de la ENALLT						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre de la ubicación
unidad	-	-	-	FK	Unidad	Referencia a tabla en base externa

Tabla 4.2.2.11 - Categoria						
Catálogo de categorías de equipos de cómputo						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre de la categoría

Tabla 4.2.2.12 - Proposito

Catálogo de propósitos del uso de los diferentes activos de cómputo						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	-	-	Nombre del propósito

Tabla 4.2.2.13 - Factura

Registro de las facturas asociadas a la adquisición de activos de hardware						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	225	charfield	-	FK	Proveedor	Proveedor del activo de hardware
numero	4	entero	-	-	-	Número de factura
garantía	5	datefield	-	-	-	Fecha de expiración de garantía

Tabla 4.2.2.14 - OS

Catálogo de sistemas operativos						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
tipo	-	-	-	FK	CategoriaOS	Familia de SO (e.g. Windows, Linux)
version	90	charfield	✓	-	-	Versión del sistema operativo
arquitectura	30	charfield	-	-	-	Arquitectura de 32 o 64 bit

Tabla 4.2.2.15 - MarcaRegulador						
Catálogo de marcas/modelos de regulador						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.16 - MarcaEquipo						
Catálogo de marcas/modelos de equipos (CPU/Laptop)						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.17 - MarcaAlmacenamiento						
Catálogo de marcas/modelos de dispositivos de almacenamiento						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.18 - MarcaTelecom						
Catálogo de marcas/modelos de dispositivos de telecomunicaciones						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.19 - MarcaMonitor						
Catálogo de marcas/modelos de monitor						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.20 - MarcaTeclado						
Catálogo de marcas/modelos de teclado						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.21 - MarcaMouse						
Catálogo de marcas/modelos de mouse						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.22 - MarcaTabletaDigital						
Catálogo de marcas/modelos de tableta digital						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.23 - MarcaCamara						
Catálogo de marcas/modelos de cámara						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.24 - MarcaBocinas						
Catálogo de marcas/modelos de bocinas						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.25 - MarcaScanner						
Catálogo de marcas/modelos de scanner						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.26 - MarcaProyector						
Catálogo de marcas/modelos de proyector						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.27 - Marcalmpresora						
Catálogo de marcas/modelos de impresora						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	NombreMarca	Nombre de la compañía
modelo	-	-	-	FK	ModeloMarca	Modelo del activo

Tabla 4.2.2.28 - Procesador						
Catálogo de procesadores						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
nombre	-	-	-	FK	CategoriaProcesador	Tipo de procesador (e.g. Intel, AMD...)
modelo	90	-	✓	-	-	Modelo del activo
freqMhz	4	entero	-	-	-	Frecuencia en MHz

Tabla 4.2.2.29 - RAM						
Catálogo de memoria RAM						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ddr	4	entero	-	-	-	Tipo de DDR (e.g. 2, 3)
freqMhz	4	entero	-	-	-	Frecuencia en MHz

Tabla 4.2.2.30 - Antivirus

Catálogo de antivirus						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
marca	90	charfield	-	-	-	Marca del AV
versión	30	charfield	-	-	-	Versión del AV
licencia	1	booleano	-	-	-	Con licencia de la ENALLT?

Tabla 4.2.2.31 - Office

Catálogo de Office o suites afines						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
marca	90	charfield	-	-	-	Marca de Office
versión	30	charfield	-	-	-	Versión de Office
sp	4	entero	-	-	-	Versión del Service Pack (si aplica)

Tabla 4.2.2.32 - HD

Registro de discos duros						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
sizeGB	4	entero	-	-	-	Capacidad en GB
marca	90	charfield	-	-	-	Marca del HD

Tabla 4.2.2.33 - Regulador						
Registro de reguladores						
Campo	Tamaño	Tipo de dato	Acepta nullos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
modelo	-	-	-	FK	MarcaRegulador	Marca del activo
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
ups	1	booleano	-	-	-	¿Es UPS?
activo	1	booleano	-	-	-	¿Está activo?
sockets_libres	4	entero	-	-	-	Número de sockets desocupados
inventario	30	charfield	-	-	-	Número de inventario
baja	5	datefield	-	-	-	Fecha de baja
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Referencia a la tabla MarcaRegulador

Tabla 4.2.2.34 - Equipo						
Registro de equipos: CPUs, laptops, equipos todo en uno						
Campo	Tamaño	Tipo de dato	Acepta nullos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
categoria	-	-	-	FK	Categoria	Tipo de equipo
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
proposito	-	-	-	FK	Proposito	Propósito del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo

ram	-	-	-	FK	RAM	Tipo de RAM instalado
activo	1	booleano	-	-	-	¿Está activo?
gbram	8	decimalfield	-	-	-	Cantidad de RAM en GB
inventario	30	charfield	-	-	-	Número de inventario
hostname	150	charfield	-	-	-	Nombre de equipo
baja	5	datefield	-	-	-	fecha de baja
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
os	-	-	✓	FK	OS	Sistema operativo instalado
regulador	-	-	✓	FK	Regulador	Regulador al que está conectado
proc	-	-	✓	FK	Procesador	Procesador instalado
av	-	-	✓	FK	Antivirus	Antivirus instalado
office	-	-	✓	FK	Office	Suite de Office instalada
discos	-	-	-	-	HD	Disco(s) duro(s) instalado(s)
serial	300	charfield	-	-	-	Número de serie
proyecto	300	charfield	-	-	-	Proyecto de trabajo
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.35 - Almacenamiento						
Registro de dispositivos de almacenamiento						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicación	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaAlmacenamiento	Marca del activo
tipo	-	-	-	FK	CategoriaAlmacenamiento	Tipo de dispositivo

factura	-	-	-	FK	Factura	Referencia a la tabla Factura
inventario	30	charfield	-	-	-	Número de inventario
removible	1	booleano	-	-	-	¿Utiliza medios removibles?
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	Fecha de baja
regulador	-	-	✓	FK	Regulador	Regulador al que está conectado
discos	-	-	-	-	HD	Disco(s) duro(s) instalado(s)
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.36 - Telecom						
Registro de dispositivos de telecomunicaciones						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicación	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaTelecom	Marca del activo
tipo	-	-	-	FK	CategoriaTelecom	Tipo de dispositivo
factura	-	-	-	FK	Factura	Referencia a la tabla Factura
inventario	30	charfield	-	-	-	Número de inventario
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	Fecha de baja
regulador	-	-	✓	FK	Regulador	Regulador al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado

serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.37 - Monitor

Registro de monitores

Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario
inches	4	entero	-	-	-	Tamaño de pantalla
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
equipo	-	-	✓	FK	Equipo	Equipo al que está conectado
regulador	-	-	✓	FK	Regulador	Regulador al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.38 - Teclado

Registro de teclados

Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador

ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
equipo	-	-	✓	FK	Equipo	Equipo al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.39 - Mouse

Registro de mouses

Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
equipo	-	-	✓	FK	Equipo	Equipo al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.40 - Tableta						
Registro de tabletas digitales						
Campo	Tamaño	Tipo de dato	Acepta nullos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
equipo	-	-	✓	FK	Equipo	Equipo al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.41 - Camara						
Registro de cámaras						
Campo	Tamaño	Tipo de dato	Acepta nullos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja

equipo	-	-	✓	FK	Equipo	Equipo al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.42 - Bocinas

Registro de bocinas

Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
equipo	-	-	✓	FK	Equipo	Equipo al que está conectado
regulador	-	-	✓	FK	Regulador	Regulador al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.43 - Scanner

Registro de scanners						
Campo	Tamaño	Tipo de dato	Acepta nullos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
categoria	-	-	-	FK	CategoriaScanner	Tipo de scanner
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
equipo	-	-	✓	FK	Equipo	Equipo al que está conectado
regulador	-	-	✓	FK	Regulador	Regulador al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.44 - Proyector

Registro de proyectores						
Campo	Tamaño	Tipo de dato	Acepta nullos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario

activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
proposito	-	-	✓	FK	Equipo	Propósito del activo
regulador	-	-	✓	FK	Regulador	Regulador al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
serial	300	charfield	-	-	-	Número de serie
proyecto	300	charfield	-	-	-	Proyecto de trabajo
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.45 - Impresora						
Registro de impresoras						
Campo	Tamaño	Tipo de dato	Acepta nullos?	Es llave?	Tabla asociada	Descripción
id	-	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
categoria	-	-	-	FK	CategorialImpresora	Tipo de impresora
modelo	-	-	-	FK	MarcaEquipo	Marca del activo
factura	-	-	-	FK	Factura	Factura asociada al activo
inventario	30	charfield	-	-	-	Número de inventario
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
proposito	-	-	✓	FK	Equipo	Propósito del activo
equipo	-	-	✓	FK	Equipo	Equipo al que está conectado
regulador	-	-	✓	FK	Regulador	Regulador al que está conectado
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo

serial	300	charfield	-	-	-	Número de serie
proyecto	300	charfield	-	-	-	Proyecto de trabajo
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.46 - Generico

Registro de dispositivos o conceptos que no son activos de hardware

Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ubicacion	-	-	-	FK	Ubicacion	Ubicación del activo
identificador	60	charfield	-	-	-	Identificador legible del genérico
activo	1	booleano	-	-	-	¿Está activo?
baja	5	datefield	-	-	-	fecha de baja
usuario	-	-	✓	FK	Usuario	Usuario al que está asignado el activo
categoría	225	charfield	-	-	-	Clasificación del dispositivo
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.47 - IP

Registro de direcciones IP homologadas de la Escuela

Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ip	117	GenericIPAddressField	-	-	-	Dirección IP
equipo	-	-	-	-	Equipo	Equipo que tiene la dirección IP
impresora	-	-	-	-	Impresora	Impresora que tiene la dirección IP

telecom	-	-	-	-	Telecom	Dispositivo de telecomunicaciones que tiene la dirección IP
storage	-	-	-	-	Almacenamiento	Dispositivo de almacenamiento que tiene la dirección IP
generico	-	-	-	-	Generico	Dispositivo genérico que tiene la dirección IP
notas	-	textfield	-	-	-	Notas adicionales

Tabla 4.2.2.48 - Tech

Registro de personal técnico que puede atender solicitudes de soporte

Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
personal	-	-	-	FK	Usuario	Usuario del sistema

Tabla 4.2.2.49 - Ticket

Registro de dispositivos o conceptos que no son activos de hardware

Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
asignado	-	-	-	FK	Tech	Personal que tiene asignado el ticket
levanta	-	-	-	FK	Usuario	Usuario que levanta el ticket
servicio	60	charfield	-	-	-	Descripción del problema
timestamp	5	datetime	-	-	-	Fecha y hora de la creación del ticket
resuelto	1	booleano	-	-	-	¿Está resuelto?

tipo	6	charfield	-	-	-	Descripción del tipo de ticket: Hardware, Software, Asesoría o Movimiento de Hardware
usuario	-	-	✓	FK	Usuario	Usuario relacionado con el ticket
ubicacion	-	-	✓	FK	Ubicacion	Ubicación relacionada con el ticket
reguladores	-	-	-	-	Regulador	Reguladores relacionados con el ticket
equipos	-	-	-	-	Equipo	Equipos relacionados con el ticket
almacenamientos	-	-	-	-	Almacenamiento	Dispositivos de almacenamiento relacionados con el ticket
telecoms	-	-	-	-	Telecom	Dispositivos de telecomunicaciones relacionados con el ticket
monitores	-	-	-	-	Monitor	Monitores relacionados con el ticket
teclados	-	-	-	-	Teclado	Teclados relacionados con el ticket
mouses	-	-	-	-	Mouse	Mouses relacionados con el ticket
tabletas	-	-	-	-	Tableta	Tabletas relacionadas con el ticket
camaras	-	-	-	-	Camara	Cámaras relacionas con el ticket
bocinas	-	-	-	-	Bocinas	Bocinas relacionadas con el ticket
scanners	-	-	-	-	Scanner	Scanners relacionados con el ticket
proyectores	-	-	-	-	Proyector	Proyectores relacionados con el ticket
impresoras	-	-	-	-	Impresora	Impresoras relacionadas con el ticket
genericos	-	-	-	-	Generico	Dispositivos genéricos relacionados con el ticket
ips	-	-	-	-	IP	Direcciones IP homologadas relacionadas con el ticket

Tabla 4.2.2.50 - Seguimiento						
Registro de dispositivos o conceptos que no son activos de hardware						
Campo	Tamaño	Tipo de dato	Acepta nulos?	Es llave?	Tabla asociada	Descripción
id	4	entero	-	PK	-	Identificador
ticket	-	-	-	FK	Tech	Responsable anterior
responsable	-	-	-	FK	Usuario	Usuario que levanta el ticket
timestamp	5	datetime	-	-	-	Fecha y hora de registro del seguimiento
comentario	-	textfield	-	-	-	Descripción de la solución o problema
resuelto	1	booleano	-	-	-	¿Está resuelto?
escalado	-	-	✓	FK	Tech	Responsable actual

Las tablas intermedias de las relaciones MtM (*Many to Many*, Muchos a muchos) son creadas y administradas por Django en su script de creación de la base de datos, y en los procesos migratorios respectivos, de manera automática, por lo que no se incluyen en este diccionario de datos. Los tipos de datos mencionados en el diccionario son aquellos que fueron definidos en el archivo de Django.

Los tamaños de los datos son aquellos que el tipo de datos equivalente de MySQL utiliza, según la propia documentación de MySQL. Todos los tamaños indicados son en bytes. El campo textfield tiene un límite de $2^{16} + 2$ bytes, pero su tamaño real en disco es variable, dependiendo de los valores introducidos por el usuario. El cálculo para los valores charfield (que se traducen a *varchar* en MySQL) considera 3 bytes por carácter en su longitud máxima permitda, debido a que se admiten caracteres UTF-8, y la implementación de UTF-8 en MySQL no soporta caracteres UTF-8 que requieren 4 bytes; al igual que en el caso de textfield, el tamaño en disco puede variar según los datos introducidos por el usuario.

El tipo de dato *GenericIPAddressField* se traduce como un tipo de dato *Char* de 39 posiciones – este último valor, debido a que el campo puede contener direcciones IPv4, así como IPv6. Éstas últimas determinan la longitud del campo.

Por último, los campos de *DateTimeField* y *DateField* requieren, además del espacio notado, de un espacio adicional para el almacenamiento de fracciones de segundo, aunque este requerimiento no fue establecido en el archivo de modelos de Django.

La columna de “acrónimo” no se utiliza, debido a que Django realiza el proceso de nombrado de tablas de manera automática. Los nombres de las tablas en la base de datos no se acortan – de hecho, se alargan, dependiendo del esquema de nombrado que se utilice en Django, en el estilo *app_tabla*. En este caso particular, tenemos un *proyecto* llamado *control*, y dos *apps* llamadas *ticket* e *inventario*. Los nombres de las tablas, por lo tanto, siguen el patrón *inventario_equipo* o *ticket_tech*.

4.2.3 Normalización

El objetivo de la normalización de una base de datos es la minimización de redundancia de datos, así como el mejoramiento de la integridad de los datos, a través de procesos secuenciales de revisión de las relaciones entre las tablas.

En el caso de un motor de bases de datos estándar (como MySQL), es necesario realizar el modelado de la base y la normalización de manera manual, o asistido a través de una herramienta que oriente al diseñados sobre estos procesos.

En el caso particular de Django, este procedimiento es realizado automáticamente al realizar el proceso de migración de la base de datos. Recordemos que Django solicita la información de los objetos de datos a través de un archivo de Python; el proceso de generación de las consultas SQL de creación de la base es un proceso automático, y parte de este proceso incluye la normalización de los datos. En caso de detectar algún error de redundancia o referencias cíclicas, Django simplemente no procesa la migración y espera la acción correctiva del usuario sobre los modelos especificados.

4.3 Diseño y construcción del frontend.

El sistema de Inventariado y Ticketing está basado en web. La dirección URL con la que entramos al sistema es <http://control.enallt.unam.mx>. En la figura 4.3.1 se observa la manera del manejo de la seguridad antes de ingresar al sistema. Al final con la tecla ENTER o con el mouse nos da acceso.

Nombre de usuario:

Contraseña:

Figura 4.3.1 Pantalla de seguridad para acceso al sistema.

En el directorio principal de inventario podemos manejar todos los dispositivos, también se puede manejar la autenticación de los usuarios. Del lado derecho se ha creado la ventana donde cada usuario pueda ver el historial de actividades dentro del sistema. El detalle lo podemos ver en la figura 4.3.2. A cada acceso o liga de los menús de la página principal, le corresponderá un archivo .py el cual contendrá toda la información necesaria ya sea texto y/o imágenes necesarias para presentar la información que se desee en cada una de las vistas.

Administración del sitio
Authentication and Authorization

- Grupos: Agregar, Modificar
- Usuarios: Agregar, Modificar

Inventario

- Antivirus: Agregar, Modificar
- Bocinas: Agregar, Modificar
- Cameras: Agregar, Modificar
- Categorías: Agregar, Modificar
- Categorías de dispositivo de almacenamiento: Agregar, Modificar
- Categorías de dispositivo de almacenamiento: Agregar, Modificar
- Categorías de impresora: Agregar, Modificar
- Categorías de procesador: Agregar, Modificar
- Categorías de scanner: Agregar, Modificar
- Direcciones IP homologadas: Agregar, Modificar
- Discos duros: Agregar, Modificar
- Dispositivo genéricos: Agregar, Modificar
- Dispositivos de almacenamiento: Agregar, Modificar
- Dispositivos de telecomunicaciones: Agregar, Modificar
- Equipos: Agregar, Modificar
- Facturas: Agregar, Modificar
- Impresoras: Agregar, Modificar
- Marcas: Agregar, Modificar
- Marcas de bocinas: Agregar, Modificar
- Marcas de cámara: Agregar, Modificar
- Marcas de dispositivo de almacenamiento: Agregar, Modificar
- Marcas de dispositivo de telecomunicaciones: Agregar, Modificar
- Marcas de equipo: Agregar, Modificar
- Marcas de impresora: Agregar, Modificar
- Marcas de monitor: Agregar, Modificar
- Marcas de mouse: Agregar, Modificar
- Marcas de proyector: Agregar, Modificar
- Marcas de regulador: Agregar, Modificar
- Marcas de scanner: Agregar, Modificar
- Marcas de tableta digital: Agregar, Modificar
- Marcas de teclado: Agregar, Modificar
- Modelos: Agregar, Modificar

Acciones recientes

Mis acciones

- manuel
Usuario
- Morales Avila David Israel, Maestro (MCAD771103)
Usuario
- Sánchez García Román Alvar, Maestro (SAGR650519H91)
Usuario
- García Morales Juan Manuel, Maestro (GAMJ711005EF5)
Usuario
- manuel
Usuario
- 132.248.74.5
Dirección IP homologada
- 2329248 (kuechenberg) (en Cómputo (CILE)-Secretaría General->Departamento de Cómputo)
Equipos
- 2329248 (104DITEL) (en Cómputo (CILE)-Secretaría General->Departamento de Cómputo)
Equipos
- 132.247.187.240
Dirección IP homologada
- 132.248.74.57
Dirección IP homologada

132.247.187.18

Figura 4.3.2 Pantalla de administración del equipo de cómputo.

Dentro de inventario tenemos la ventana de agregar equipo. Dependiendo el equipo que deseamos adicionar son las ventanas que hay que llenar. En la figura 4.3.3 hay un ejemplo.

Inicio > Inventario > Equipos > Agregar equipo

Agregar equipo

Categoría: [dropdown]

Ubicación: [dropdown]

Propósito: [dropdown]

Modelo: [dropdown]

Factura: [dropdown]

Ram: [dropdown]

¿El equipo está activo?

Ram en GB (3 decimales max): [input]

Número de inventario: [input]

Nombre de equipo: [input]

Fecha de baja en el sistema: [calendar] Hoy | [calendar]

Usuario asignado: [dropdown]

Os: [dropdown]

Regulador: [dropdown]

Proc: [dropdown]

Av: [dropdown]

Office: [dropdown]

Discos: [dropdown]
16GB, Samsung
20GB, FUJITSU
25GB
40GB, WDC

Hold down "Control", or "Command" on a Mac, to select more than one.

132.247.187.18

Figura 4.3.3 Pantalla agregar equipo nuevo.

La siguiente pantalla es un ejemplo de la modificación dentro de inventario, se puede buscar por ID de la etiqueta o por categoría de equipo. Para este caso se pretende modificar atributos de una PC. Ver figura 4.3.4.

Inicio > Inventario > Equipos > 2269210 (237ALEMAN) [en Alemán (CELE->Secretaria General)]

Modificar equipo

Categoría: [dropdown]

Ubicación: [dropdown]

Propósito: [dropdown]

Modelo: [dropdown]

Factura: [dropdown]

Ram: [dropdown]

¿El equipo está activo?

Ram en GB (3 decimales max): [input]

Número de inventario: [input]

Nombre de equipo: [input]

Fecha de baja en el sistema: [calendar] Hoy | [calendar]

Usuario asignado: [dropdown]

Os: [dropdown]

Regulador: [dropdown]

Proc: [dropdown]

Av: [dropdown]

Office: [dropdown]

Discos: [dropdown]
16GB, Seagate
16GB, WDC
16GB, Maxtor
16GB, Seagate

Hold down "Control", or "Command" on a Mac, to select more than one.

132.247.187.18

Figura 4.3.4. Pantalla de modificación de atributos de un equipo.

Podemos desplegar información de cualquier activo dentro del inventario únicamente editándolo como se ve en la figura 4.3.5.

Activo || [Editar equipo](#) || [Reasignar/Baja](#)

Inventario: 2269210
Serie: ST-38S1FH1
Categoría: PC
Ubicación: [Alemán \(CELE->Secretaría General\)](#)
Usuario asignado: [Nieves Váldez Ma. Isabel, Secretario \(NIVIXXXXXX\)](#)
Tipo de usuario: Administrativo
Propósito:
Sistema operativo: Microsoft Windows 7 Professional build 7601
Nombre de equipo: 237ALEMAN
Modelo: Dell OPTIPLEX 755
Procesador: Intel Core 2 Duo E4600 a 2400 Mhz
Memoria: 2.000 GB de RAM DDR 2 a 333 Mhz
Office
Enterprise 2007 12.0.6612. SP 0
Sin licencia
Discos
- Seagate Barracuda 250Gb
Regulador: [1598147 \(Tripp Lite LS604M\) \[en Alemán \(CELE->Secretaría General\)\]](#)
Monitor: [2269219 \(Dell E178FPb 17"\) \[en Alemán \(CELE->Secretaría General\)\]](#)
Teclado: [2269210K \(Dell SK-8115\) \[en Alemán \(CELE->Secretaría General\)\]](#)
Mouse: [2269210M \(Dell M0C5U0\) \[en Alemán \(CELE->Secretaría General\)\]](#)
IP: [132.247.187.106 \(Cambiar IP\)](#)
Notas: Anteriormente tenía la I.P. 132.248.74.237. pasa al al I.P. 132.247.187.106

Figura 4.3.5 Pantalla de edición de los equipos.

Tenemos el despliegue de letreros de advertencia cuando un usuario intenta ingresar a alguna parte de lo cual no tiene los atributos necesarios para ingresar. En la figura 4.3.6 vemos un ejemplo que se despliega a un usuario común con ciertos privilegios que solo le permiten abrir un ticket.

Esta página es de uso exclusivo del personal técnico del CELE. [Iniciar sesión](#)

[Abrir nuevo ticket](#)

Figura 4.3.6 Pantalla de seguridad en caso de intrusos.

Para la apertura de ticket la pantalla nos muestra la persona que generó el ticket, a quién se le asigna, la descripción y al final de despliega el inventario de equipo para que sea escogido por el usuario que lo abrió. Esto se puede ver en la figura 4.3.7.

Persona que genera el ticket: Sergio Andrés Becerril López

A quién se asignará el ticket: David Israel

Descripción del problema:

Servicio solicitado por: Escriba un nombre o apellido

Servicio solicitado para el área: Escriba una ubicación

Activos afectados

Equipos (CPUs, portátiles y tablets)

Agregar EQUIPO Escriba una ubicación +

Impresoras

Reguladores

Direcciones IP

Proyectores

Dispositivos de almacenamiento masivo

Dispositivos de telecomunicaciones

Monitores

Teclados

Mouses

Tabletas digitalizadoras

Cámaras

132.247.187.18

Figura 4.3.7. Pantalla para apertura de un ticket.

También podemos desplegar el total de tickets como se muestra en la figura 4.3.8

Ticket ID	Fecha de solicitud	Descripción del problema	Para quién	Personal asignado
Ticket ID 15	Mie 17/jun/15 12:53	Configurar perfil de usuario para MacBook Pro (OS X) - Nuevo usuario - Configuración de red. - Impresora. - Software de uso común.		José Luis Vázquez Torres
Ticket ID 32	Mie 17/jun/15 13:36	No responde la computadora		José Enrique Benavides Maya
Ticket ID 33	Mie 17/jun/15 13:44	Respaldo de información	Alma Luz Rodríguez en DLA	José Enrique Benavides Maya
Ticket ID 35	Jue 18/jun/15 14:13	Actualización de Java	Alma María Signoret Dorcasberro en Dirección	José Enrique Benavides Maya
Ticket ID 41	Vie 19/jun/15 10:57	Verificación de cuenta de Correo		Juan Carlos Guzman Medina
Ticket ID 42	Vie 19/jun/15 11:00	Manda un mensaje de actualización de equipo	Cristel Martínez Pineda	Juan Carlos Guzman Medina
Ticket ID 43	Vie 19/jun/15 11:11	Servidores		Sergio Andrés Becerril López
Ticket ID 44	Vie 19/jun/15 11:29	no prende	Juan Carlos Guzman Medina	Juan Manuel García Morales
Ticket ID 45	Vie 19/jun/15 14:22	Problema de lentitud al ejecutarse el antivirus. Asesoría con configuración de navegador web (asignar página de inicio) y buscador. Asesoría de uso general de dropbox. Asesoría de uso del programa Voices para utilización de grabadoras de voz portátiles	Martha Lucrecia Franco Curiel en Alemán	Juan Carlos Guzman Medina
Ticket ID 46	Lun 22/jun/15 13:43	No se puede imprimir	Miguel Angel Ramirez Campos en Biblioteca	José Enrique Benavides Maya
Ticket ID 47	Lun 22/jun/15 13:44	Problemas de virus	Maria del Rocío Camacho Suárez en Biblioteca	José Enrique Benavides Maya
Ticket ID 48	Lun 22/jun/15 13:44		Vridiana López	José Enrique Benavides Maya

Figura 4.3.8 Pantalla de despliegue de Tickets abiertos.

La venta donde se administra todo el personal se muestra en la figura 4.3.9.

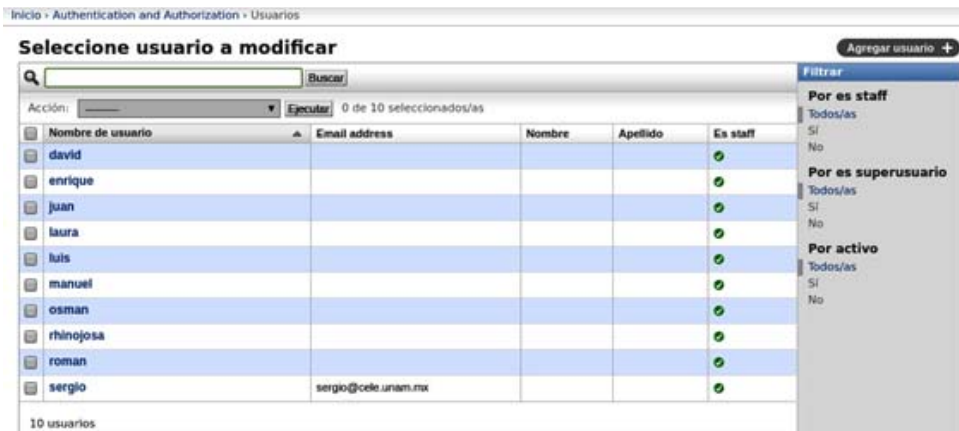


Figura 4.3.9 Pantalla del personal y sus atributos.

4.3.1 Diseño de la interfaz gráfica empleando RUP y UML.

El desarrollo de software depende de muchas actividades y etapas, El papel importante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, debe guiar y organizar actividades que lleven a las metas trazadas en el grupo. En el presente tema se detallan los dos grandes enfoques. **RUP** y **UML** enfocadas al desarrollo del sistema planteado. **RUP** está pensada para asignar tareas y responsabilidades dentro de una organización de desarrollo **UML** es el resultado de una propuesta de estandarización promovida por **OMG** (Object Management Group).

Metodología RUP

Las siglas RUP en inglés significan Rational Unified Process (Proceso Unificado de Rational) y provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo principal es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales, cumpliendo con el cronograma y presupuesto especificado. Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización.

Las tres características esenciales que definen al RUP son:

- Proceso Dirigido por los Casos de Uso: se refiere a la utilización de los casos de uso para el desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Estos son la base para la implementación de las fases y disciplinas del RUP, es una secuencia de pasos a seguir para la realización de un fin o propósito, y se relaciona directamente con los requerimientos.
- Proceso Iterativo e Incremental: modelo utilizado por RUP para el desarrollo de un proyecto de software. Esta implementación del proyecto en Iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración.
- Proceso Centrado en la Arquitectura: Define la Arquitectura de un sistema, y una arquitectura ejecutable construida como un prototipo evolutivo. Arquitectura de un sistema es la organización o estructura de sus partes más relevantes. Una arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades. RUP establece refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo evolutivo.

Las principales ventajas de RUP son:

- Evaluación en cada fase que permite cambios de objetivos
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.

Las principales desventajas de RUP son:

- La evaluación de riesgos es compleja
- Excesiva flexibilidad para algunos proyectos
- Se deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

El diagrama de casos de usos completo del sistema de inventario y ticketing se muestra en la figura 4.3.10. En ella se plantea el seguimiento de cada fase en el desarrollo del software.

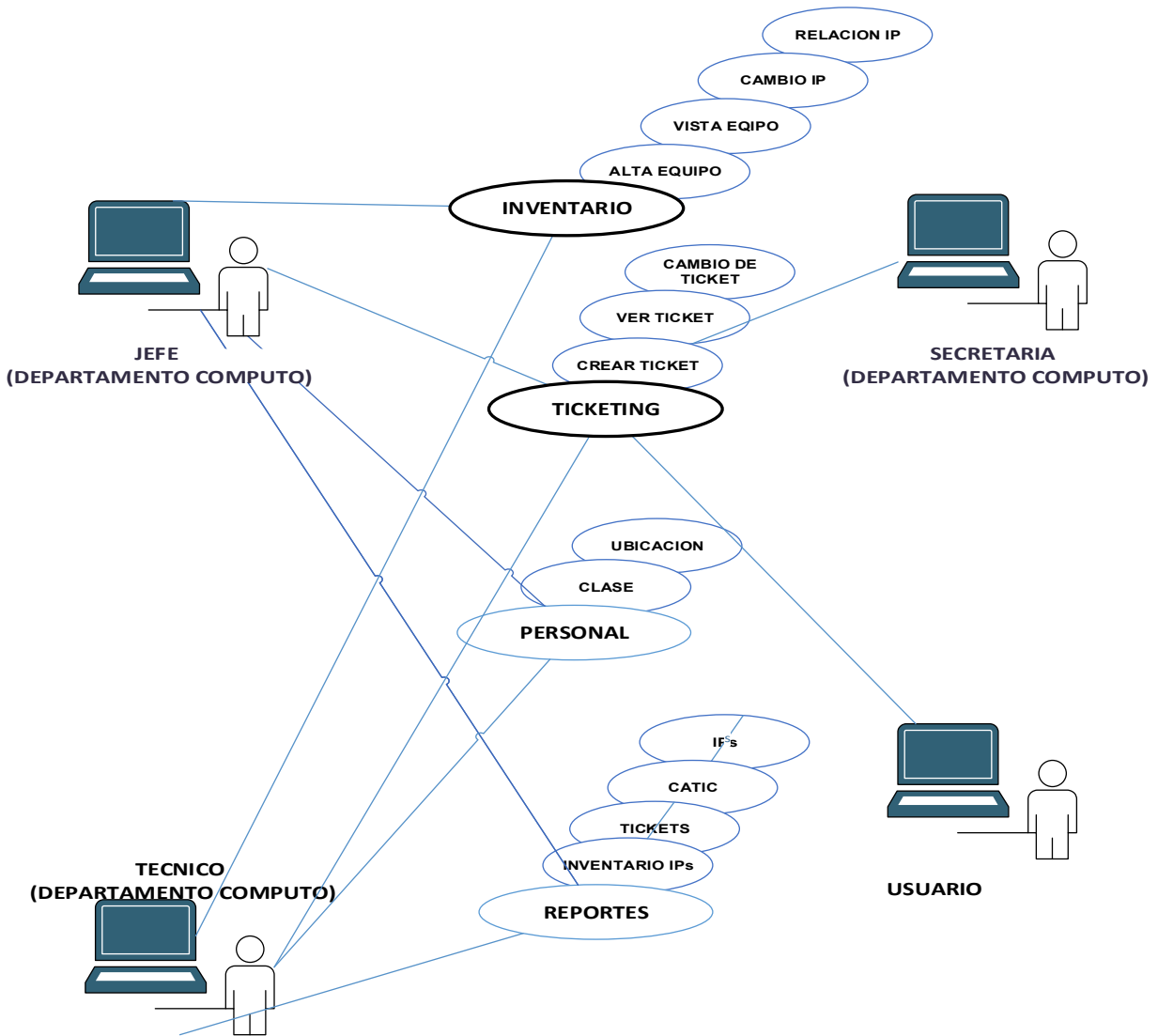


Figura 4.3.10. Diagrama RUP completo con la relación de actores y casos de uso.

UML (Lenguaje Unificado de Modelado)

UML surge como respuesta al problema de contar con un lenguaje estándar para escribir planos de software, el llamado Lenguaje Unificado de Modelado, es una

notación estándar para el modelado de sistemas de software, resultado de una propuesta de estandarización promovida por el consorcio OMG (Object Management Group), del cual formaban parte las empresas más importantes que se dedicaban al desarrollo de software, en 1996. Es importante recalcar que sólo se trata de una notación, es decir, de una serie de reglas y recomendaciones para representar modelos. UML no es un proceso de desarrollo, es decir, no describe los pasos sistemáticos a seguir para desarrollar software, sólo permite documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos.

Las principales características de UML son:

- Es un lenguaje de propósito general para el modelado orientado a objetos.
- Combina notaciones provenientes de Modelado Orientado a Objetos, Modelado de Datos, Modelado de Componentes, Modelado de Flujos de Trabajo.
- Construye modelos, que son simplificaciones de la realidad, para comprender mejor el sistema que vamos a desarrollar. Cuando involucramos a cientos de desarrolladores trabajando y compartiendo información, el uso de modelos y el proporcionar información sobre las decisiones tomadas, es vital durante el desarrollo y finalizado de éste; cuando se requiere algún cambio en el sistema esto será de mucha ayuda.
- Para la construcción de modelos, hay que centrarse en los detalles relevantes mientras se ignoran los demás, por lo cual con un único modelo no tenemos bastante.

También describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle. El proceso de desarrollo de software ofrece un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés. El código fuente del sistema es el modelo más detallado del sistema y además es ejecutable, sin embargo, se requieren otros modelos, varios de estos aportan diferentes vistas de un sistema los cuales nos ayudan a comprenderlo desde varios frentes.

El diagrama UML de actividades para el sistema de inventario y ticketing es mostrado a continuación en la figura 4.3.11.

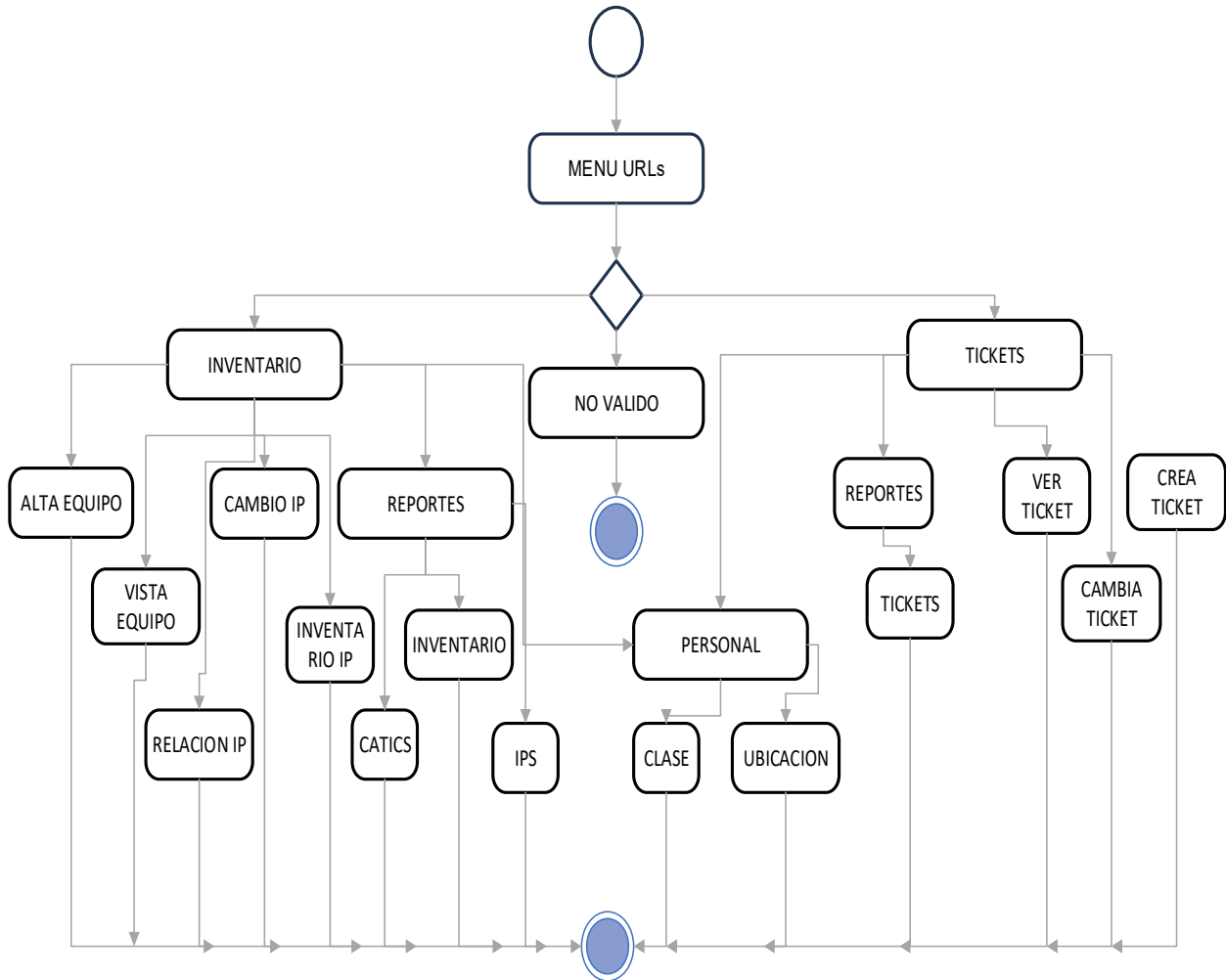


Figura 4.3.11 Modelo UML de actividades para inventario y tickets.

4.4 Pruebas y mantenimiento (Riesgos, imprevistos y mitigación de estos).

Una vez que el sistema ha sido desarrollado, es necesario someterlo a pruebas que nos permitan descubrir y mejorar fallas antes de liberarlo. Después de haber sido evaluado e implementado se debe establecer una norma de mantenimiento, la cual nos permita asegurar que el sistema continúe operando en el nivel solicitado.

Las pruebas se encuentran encaminadas sobre la lógica interna del software. Son procesos de ejecución de un programa con el objetivo de identificar puntos vulnerables. Se pueden realizar de la siguiente forma:

- A través de pruebas de sometimiento que demuestren que cada función es completamente operativa.
- Desarrollando pruebas que ajuste la operación del sistema a las especificaciones requeridas.
- Realizando pruebas de estabilidad, cobertura y rendimiento de arquitectura.

Las pruebas realizadas no pueden asegurar la ausencia de modificaciones en el software, por el contrario, proporcionan retroalimentación mientras aún se dispone de tiempo y recursos.

A continuación, se lista los diferentes tipos de pruebas:

Prueba de caja negra.

La prueba de caja negra es una técnica en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna del código, detalles de implementación o escenarios de ejecución internos del sistema. El objetivo es construir conjuntos de datos de entrada que consigan ejercitar todos los requisitos funcionales de un programa.

Prueba de caja blanca.

Son pruebas basadas en la lógica del programa, no en las especificaciones. Son realizadas por programadores empleando su conocimiento sobre el funcionamiento interno del código.

Entre las pruebas de caja blanca tenemos las pruebas de instrucciones, decisiones que se refieren al código del programa, coherencia en sentencias, etc. Es un objetivo muy costoso de alcanzar en su plenitud.

Prueba de regresión.

Son pruebas que buscan descubrir errores o **bugs**¹⁶, carencias de funcionalidad o divergencias con respecto al comportamiento esperado del sistema a consecuencia de alguna modificación. Consiste en volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse de que los cambios no han propagado efectos no deseados.

Prueba de validación.

La validación proporciona la certeza de que el sistema satisface todos los requisitos funcionales de comportamiento y rendimientos. Se emplean únicamente para pruebas de caja negra.

Pruebas de integración.

También se les conoce como pruebas integrales y se realizan en el ámbito del desarrollo de software. Se centran en probar la comunicación entre los componentes. Su objetivo

¹⁶ lit. error de software. Es un problema en un sistema que desencadena un resultado no deseado.

es construir la estructura del programa, mientras se llevan a cabo las pruebas para detectar errores asociados con la interacción. Los principales tipos de integración son:

- **Integración incremental.** Son combinados el siguiente módulo que se debe probar con el conjunto de módulos que ya han sido evaluados.
- **Integración no incremental.** Se prueba cada módulo por separado para después integrarlos todos y probarlos en el programa completo.
 - **Ascendente.** Inicia en los módulos hoja,
 - **Descendente.** Da inicio en el módulo raíz.

Pruebas unitarias.

Son pruebas enfocadas en la verificación de las unidades más pequeñas de diseño, es decir, el módulo. Prueban los caminos de control importantes con el fin de descubrir errores dentro de los límites de éste.

Pruebas de ALFA y BETA.

Las pruebas Alfa son llevadas a cabo por el usuario en el lugar donde se desarrolló el sistema; el desarrollador participa como observador. Las pruebas Beta se llevan cabo por los usuarios finales en su lugar de trabajo. No es necesaria la presencia del desarrollador.

Pruebas de estrés.

Este tipo de pruebas ayudan a simular casos en los que el número de clientes o la entrada masiva de datos aumentan. De este modo es posible evaluar tanto el tiempo de respuesta de un sistema como su capacidad de respuesta.

Las pruebas que tienen como propósito el validar que el producto de software funcione correctamente son llamadas pruebas positivas, con la desventaja de que sólo pueden validar el producto específicamente para los casos de uso. La identificación de al menos una falla es suficiente para anunciar que el producto no funciona correctamente.

Pruebas del sistema.

Se llevaron a cabo pruebas de caja negra. Al ingresar al módulo de administración, véase la figura 4.4.1, aparece una lista de usuarios con sus correspondientes permisos para operar el sistema. Se pueden realizar búsquedas de usuarios por su nombre, atributos o estatus de actividad.

<input type="checkbox"/>	Nombre de usuario	Email address	Nombre	Apellido	Es staff
<input type="checkbox"/>	david				✓
<input type="checkbox"/>	enrique				✓
<input type="checkbox"/>	juan				✓
<input type="checkbox"/>	laura				✓
<input type="checkbox"/>	luis				✓
<input type="checkbox"/>	manuel				✓
<input type="checkbox"/>	osman				✓
<input type="checkbox"/>	rhinojosa				✓
<input type="checkbox"/>	roman				✓
<input type="checkbox"/>	sergio	sergio@cele.unam.mx			✓

Figura 4.4.1 Módulo de administración del sistema.

Algunos módulos del sistema solo pueden ser operados por usuarios autorizados. Por ejemplo, el módulo de tickets de soporte. Para esta prueba se intentó crear un ticket y modificar su información con y sin los permisos necesarios. Véanse figuras 4.4.2 y 4.4.3

Administración de Django Bienvenido, **sergio**. [View site](#) / [Cambiar contraseña](#) / [Cerrar sesión](#)

Administración del sitio

Authentication and Authorization	
Grupos	+ Agregar ✎ Modificar
Usuarios	+ Agregar ✎ Modificar
Inventario	
Antivirus	+ Agregar ✎ Modificar
Bocinas	+ Agregar ✎ Modificar
Camaras	+ Agregar ✎ Modificar
Categorías	+ Agregar ✎ Modificar
Categorías de dispositivo de almacenamiento	+ Agregar ✎ Modificar
Categorías de dispositivo de almacenamiento	+ Agregar ✎ Modificar
Categorías de impresora	+ Agregar ✎ Modificar
Categorías de procesador	+ Agregar ✎ Modificar
Categorías de scanner	+ Agregar ✎ Modificar
Direcciones IP homologadas	+ Agregar ✎ Modificar
Discos duros	+ Agregar ✎ Modificar
Dispositivo genéricos	+ Agregar ✎ Modificar
Dispositivos de almacenamiento	+ Agregar ✎ Modificar
Dispositivos de telecomunicaciones	+ Agregar ✎ Modificar
Equipos	+ Agregar ✎ Modificar
Facturas	+ Agregar ✎ Modificar
Impresoras	+ Agregar ✎ Modificar
Marcas	+ Agregar ✎ Modificar
Marcas de bocinas	+ Agregar ✎ Modificar
Marcas de cámara	+ Agregar ✎ Modificar

Acciones recientes

Mis acciones

- [✎ manuel](#)
Usuario
- [✎ Morales Avila David Israel, Maestro \(MOAD71101\)](#)
Usuario
- [✎ Sánchez García Román Alvar, Maestro \(SAGR650519R91\)](#)
Usuario
- [✎ García Morales Juan Manuel, Maestro \(GAMJ711005EF5\)](#)
Usuario
- [✎ manuel](#)
Usuario
- [✎ 132.248.74.5](#)
Dirección IP homologada
- [✎ 2329248 \(kuechenberg\) \[en Cómputo \(CELE->Secretaría General->Departamento de Cómputo\)\]](#)
Equipo
- [✎ 2329248 \(104DITEL\) \[en Cómputo \(CELE->Secretaría General->Departamento de Cómputo\)\]](#)
Equipo
- [✎ 132.247.187.240](#)
Dirección IP homologada
- [✎ 132.248.74.57](#)
Dirección IP homologada

Figura 4.4.4 Administración del módulo de inventario.

Se llevaron a cabo pruebas modificando el estatus y especificaciones de los equipos inventariados. Véase figura 4.4.5

Modificar equipo Historia

Categoría:	PC
Ubicación:	Alemán (CELE->Secretaría General)
Propósito:	Administrativo
Modelo:	Dell OPTIPLEX 755
Factura:	Dell #1
Ram:	DOR 2 @333 Mhz
<input checked="" type="checkbox"/> ¿El equipo está activo?	
Ram en GB (3 decimales máx):	2.000
Número de inventario:	2269210
Nombre de equipo:	231ALEMAN
Fecha de baja en el sistema:	26/09/2013 <input type="checkbox"/> Hoy <input type="checkbox"/>
Usuario asignado:	Neves Valdez Ms. Isabel, Secretario (NVI000000)
Os:	Microsoft Windows 7 Professional build 7601.64
Regulador:	ES98247 (Trapp Lite L3504M) [en Alemán (CELE->Secretaría General)]
Proc:	Intel Core 2 Duo E4600 2400
Av:	Kaspersky Endpoint Security 6
Office:	Enterprise 2007 32.0.6112. SP0
Discos:	160GB, Seagate 160GB, WDC 163GB, Master 8500136 (Seagate)

Figura 4.4.5 Modificación de especificaciones de un equipo.

Asimismo, se puede llevar cabo un rastreo del activo mediante su ubicación, tipo de objeto o su dirección IP asignada. Véase figura 4.4.6

[|| Editar ubicacion](#)

Ubicación física: Activo Fijo

Unidad: [CELE->Unidad Administrativa->Depto. Bienes y Suministros](#)

Impresora: [1711328 \(HP LaserJet 6P\) \[en Activo Fijo \(CELE->Unidad Administrativa->Depto. Bienes y Suministros\)\]](#)



Figura 4.4.6 Ubicación de un activo

De esta forma el sistema aprobó las pruebas de caja negra. Las pruebas de regresión se efectuaron en la misma pantalla. Estas fueron superadas satisfactoriamente.

Otra de las pruebas efectuadas fue la validación del módulo de relación IP. Esta se centra en el proceso de verificación de una unidad o módulo de software, tal como está implementada en el código fuente. Véase figura 4.4.7

IP	Inventario	Tipo	Ubicación	Usuario asignado	Tipo de usuario
132.247.187.101	2313217	PC	Italiano	José Ortega Herrera	Académico
132.247.187.102	2275972	PC	Italiano	Sonia Herrera Díaz	Administrativo
132.247.187.106	2269210	PC	Alemán	Ma. Isabel Nieves Váldez	Administrativo
132.247.187.107	2295984	PC	Alemán	Martha Lucrecia Franco Curriel	Académico
132.247.187.11	Virtual Joomla Bibli	Servidor	Sistemas o	Sergio Andrés Becerril López	N/A
132.247.187.111	2382473	PC	Frances	Mónica Rizo Marechal	Académico
132.247.187.112	2354532	PC	Frances	Jandette	Académico

Figura 4.4.7 Validación del módulo de relación IP.

Adicionalmente, para determinar una respuesta efectiva del sistema a diferente niveles de uso, se realizaron pruebas de estrés o de carga simulando varios usuarios que acceden al sistema al mismo tiempo.

Existen muchas herramientas para llevar a cabo este tipo de simulaciones. XQual Studio es una herramienta libre y multiplataforma que permite utilizar un número ilimitado de usuarios virtuales en una ejecución de prueba de carga local y remota. Los usuarios que se utilizaron en las pruebas emplean arquitectura de multiprocesador. La arquitectura de multiprocesador permite que la máquina en la que los usuarios virtuales se instalan utilice más de un procesador.

Al no ser un sistema que será visitado por miles de usuarios al mismo tiempo, no fueron necesarios equipos adicionales para generar una gran carga. Sin embargo, al ser un sistema que emplea servicios web se simuló la apertura de conexiones simultáneas

por varios usuarios al servidor donde se aloja el sistema y la realización de varias solicitudes HTTP.

Las pruebas de carga se llevaron a cabo en 3 niveles. Véase figura 4.4.8

- **Humo.** Verificar el comportamiento del sistema bajo cargas ligeras y corta duración.
- **Tensión.** Para determinar si el sistema se ejecutará con éxito durante un tiempo continua con carga pesada.
- **Rendimiento.** Conocer como responde el sistema

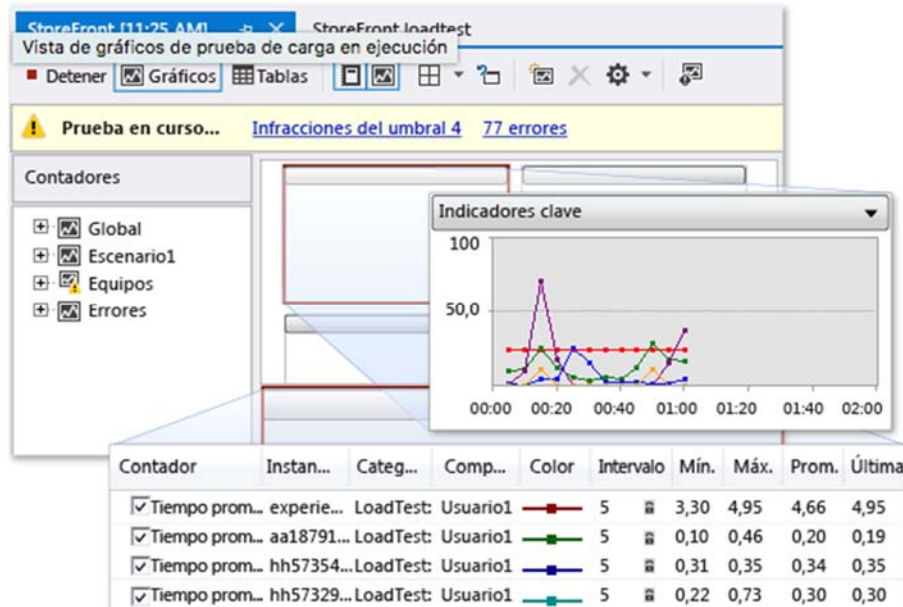


Figura 4.4.8 Resultados de prueba de carga.

Tipos de mantenimiento.

Cuando culmina la fase de desarrollo e implementación del sistema, es indispensable garantizar el mantenimiento del mismo. Los programas deben ser actualizados con el

tiempo debido a que las necesidades pueden variar. Es indispensable adaptarse a los cambios en las plataformas tecnológicas.

Existen varios tipos de mantenimiento que se pueden aplicar al sistema:

- **Mantenimiento Preventivo.** Es cuando se llevan a cabo cambios a la aplicación con el fin de mejorar el mantenimiento futuro y la estabilidad en la operación. Además, proporciona bases sobre las que podrán implementarse mejoras posteriores.
- **Mantenimiento Correctivo.** Se presenta cuando ocurre un error en la operación del sistema.
- **Mantenimiento Adaptativo.** Ocurre cuando se generan cambios en los requerimientos de tal forma que las especificaciones se adapten a los nuevos requerimientos. Estos requerimientos también pueden modificarse en las plataformas de hardware o sistemas operativos.
- **Mantenimiento Perfectivo.** Se lleva a cabo cuando existe la necesidad de optimizar los procesos, sin que cambien del todo los requerimientos.

Es así como tenemos en cuenta la importancia que tiene cada uno de estos para el funcionamiento adecuado del sistema, por lo que se debe de elaborar un plan para ejecutarlos de manera periódica. En especial el mantenimiento preventivo, pues es ahí donde es posible detectar la necesidad de implementación de los otros tipos de mantenimiento. Este se debe realizar cada 30 días.

A pesar de todas estas medidas, un sistema es propenso a fallas eventuales, por lo que, es muy recomendable llevar a cabo respaldos de la información vertida en la base de datos, esto previendo que el sistema llegará a tener alguna falla grave, sea borrado total o parcialmente del servidor en donde se aloja.

4.5 Evaluación integral del aplicativo backend, frontend y reportes.

En la evaluación integral se emplearon varios métodos sobre aspectos del sistema como prototipos del software, de código y sitio implementado.

Para llevar a cabo métodos de inspección, de usabilidad, se requiere de expertos que analicen el mapa del sitio donde se encuentra el sistema alojado para identificar errores y problemas en el diseño. Al no contar con ellos, se recurrió a una evaluación heurística que tiene como ventaja su facilidad, bajo costo y rapidez para obtener resultados. Este tipo de evaluación se llevó cabo por un grupo reducido de usuarios, basándose en su experiencia, evaluaron de forma independiente el sistema. Los aspectos que contemplaron fueron:

- Visibilidad del estado del sistema.
- Lenguaje común entre sistema y usuario.
- Control por parte del usuario, ofreciendo una forma de salida de emergencia en caso de alguna eventualidad.
- Consistencia y estándares. Cuanto más se parezca un diseño y su funcionamiento al resto de módulos del sistema, más familiar y fácil de usar resulta para el usuario.
- Prevención de errores.
- Reconocimiento en el uso. El usuario no tiene que recordar donde se encuentra cierta información, o cómo se llega a determinado sitio.
- Flexibilidad y eficiencia de uso.
- Diseño minimalista.
- Ayuda y documentación.

La evaluación con usuarios es una prueba que se basa en la observación y análisis de cómo un grupo de usuarios reales utiliza el sistema, reportando cada problema encontrado para poder solucionarlos posteriormente. Véase figura 4.5.1



Figura 4.5.1 Evaluación del sistema por parte de los usuarios.

Una de las inquietudes mencionadas por los usuarios fue la posibilidad del desarrollo del sistemas en una plataforma móvil debido al avance continuo de estas tecnologías. Los sistemas en plataformas móviles se desarrollan teniendo en cuenta las limitaciones de los propios dispositivos, como por ejemplo la escasa capacidad de almacenamiento, la versión de navegador, sistema operativo, ancho de banda, etc.

La implementación de SIRICI en una plataforma móvil se ejecutaría en servidores, este incluiría páginas web optimizadas para ser visualizadas. Por definición, el sistema puede ser accedido mediante un navegador web compatible y una Open VPN¹⁷. Su ventaja es su fácil integración con aplicaciones existentes, además de necesitar menos requerimientos de hardware. Sin embargo, el sistema no podría acceder a todas las funcionalidades.

¹⁷ Open VPN es una solución de software libre que ofrece conectividad de punto a punto con validación jerárquica de usuarios y host conectados remotamente.

Debido a la heterogeneidad de los dispositivos móviles se requiere llevar a cabo una cuidadosa planificación y diseño antes de la implementación de la aplicación. Véase figura 4.5.2



Figura 4.5.2 Vista del sistema en un dispositivo móvil

El objetivo de supervisar la base de datos es evaluar el rendimiento del servidor. Una supervisión eficaz implica tomar instantáneas periódicas del rendimiento actual para aislar procesos que causen problemas y recopilar datos de forma continua para realizar el seguimiento de las tendencias de rendimiento.

Al hacer búsquedas frecuentes en el inventario no se detectaron que las transacciones no fueran respondidas con la rapidez que se necesita o se perdiera información durante el proceso. Véase figura 4.5.3



Figura 4.5.3 Consulta del inventario

En la figura 4.5.4 se muestra el resultado de una búsqueda



Figura 4.5.4 Resultado de búsqueda

Una de las funciones indispensables que debe proporcionar un sistema de inventario y soporte es la posibilidad de crear reportes. Cada reporte contiene la información almacenada en el sistema o información que se va generando a medida que se trabaja con el programa.

Antes del desarrollo del sistema la generación y organización de estos según su tipo y fecha consumía mucho tiempo a los usuarios. Conforme lo va requiriendo la institución se han habilitado aplicaciones de reportes para cada módulo del sistema.

En la figura 4.5.5 se muestra la pantalla que sirve para elegir, en primera instancia, la generación de un reporte a través de las especificaciones del equipo.

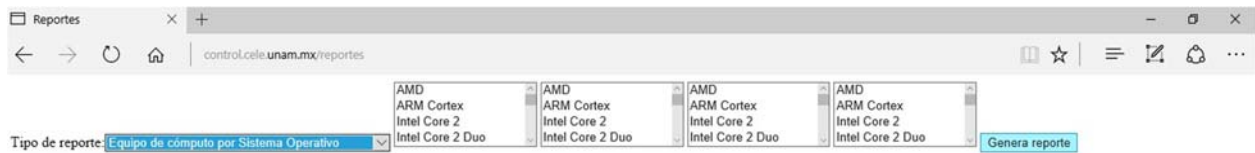


Figura 4.5.5 Generación de reportes

En la figura 4.5.6 se muestra los tipos de reportes que puede generar el sistema.



Figura 4.5.6 Tipos de reportes que genera el sistema

En la figura 4.5.7 se muestra el documento que genera.

control.cele.unam.mx x +

control.cele.unam.mx/equipos

Si el resto de la página está en blanco reporte un error al administrador

Estatus	IP	Inventario	Tipo	Ubicación	Usuario asignado	Tipo de usuario	Procesador	Sistema Operativo	Unidad	Notas
Activo		022927914	PC	SITE CEC	David Israel Morales Avila	Secretarial	Intel Pentium D	Microsoft Windows 7	CELE->Secretaria General->Departamento de Cómputo	
Activo	132.248.74.187	2062114	PC	Mediateca	Maria de la Paz Adelia Peña Clavel	Alumnos	Intel Pentium 4	Microsoft Windows XP SP3	CELE->Secretaria General	Servidor Consulta Acer
Activo	132.248.74.59	2089833	Servidor	SITE Cómputo	David Israel Morales Avila	Servidores	Sun UltraSPARC III+	UNIX/Linux Debian 7	CELE->Secretaria General->Departamento de Cómputo	IP cambiada (2015-11-15:05:00.584446), 132.248.74.58 --> 132.248.74.59
Activo	132.248.74.231	2125710	PC	Portugues	Marisol Correa Garcia	Administrativo	Intel Pentium 4	Microsoft Windows 7 Professional build 7601	CELE->Secretaria General	.. Sustituye al equipo #2175713el 2014-10-12:07:35.814464 Se le a Marisol Correa
Activo	132.247.187.131	2125712	PC	Servicios Escolares	Amadeo Segovia Guerrero	Alumnos	Intel Pentium 4	Microsoft Windows XP SP3	CELE->Secretaria General->Depto. Servicios Escolares	Ventanilla.
Activo	132.247.187.113	2125715	PC	Frances	Nancy Araceli Tadeo Campos	Administrativo	Intel Pentium 4	Microsoft Windows XP SP3	CELE->Secretaria General	... IP cambiada (2015-10:37:49.322662), 132.248.74.236 --> 132.247.187.113

Figura 4.5.7 Reporte generado.

CONCLUSIONES

Este sistema y trabajo presentado surge de la inquietud de poder proporcionar a una institución universitaria una solución acorde a sus necesidades reales y a los recursos con que cuenta; a pesar de que el sistema está acotado a las necesidades actuales podemos ofrecer una posibilidad de crecimiento a futuro por la facilidad de la lectura de su código y estructura realizada.

La arquitectura web permite su disponibilidad en línea, así como la fácil instrumentación de protocolos de mantenimiento, permitiendo un rendimiento óptimo convirtiendo a SIRICI en un sistema ligero y eficaz a las necesidades de administración de recursos y seguridad de los mismos.

Así mismo, el sistema restringe los niveles de acceso, maneja perfiles de usuario para cada tarea y monitorea los movimientos en los activos de hardware de la institución; permitiendo así tener un control mayor sobre los recursos y tareas asignadas dentro del Departamento de Cómputo.

La base de datos cumple con almacenar la información requerida de los activos y tickets de soporte y proporciona una respuesta rápida a la información solicitada.

La automatización brindada por el sistema ayuda a reducir tiempo en el proceso de búsqueda y generación de reportes.

Con la realización del presente trabajo apreciamos que hay rubros en los que podría profundizar un poco más, tales como la implementación del sistema en una aplicación móvil nativa, el control automatizado del movimiento de los activos a través de medios pasivos como etiquetas RFID, y la integración de controles (como códigos QR en los equipos) para reducir aún más la necesidad de interacción humana.

Aprovechar al máximo los componentes para el desarrollo de un sistema que se ofrecen de forma libre y gratuita abate los costos y proporciona una fuente de trabajo a los egresados de las áreas de sistemas.

El sistema cumple con los requerimientos especificados por el jefe del Departamento de Cómputo y los principales usuarios, aumentando la productividad y disminuyendo en gran medida la utilización de tiempo en generar reportes manualmente o buscar equipo en inventario manual que pudiera tener fallas y problemas de control. Los principales usuarios solo requieren de una capacitación sencilla para poder utilizar el sistema.

Los métodos utilizados para realizar el sistema de inventario y ticketing requerido con los responsables fue la adecuada para poder cubrir los diferentes escenarios presentes en las áreas que se involucran en este sistema.

El manejo de base de datos nos ha permitido tener alto manejo de transacciones y mantener niveles adecuados en la operación. También se mantiene un alto rendimiento de la instancia de base de datos y se ha obtenido consistencia de la información a través de la implementación del esquema relacional en la base de datos.

Contar con los reportes adecuados para el control del hardware que son requeridos por instancias dentro y fuera de la UNAM, han permitido al departamento de cómputo e interesados ahorro de tiempo y veracidad en la información solicitada.

La metodología de programación usada, por módulos en la programación, hace que el sistema puede seguir creciendo de manera rápida y sin perder su confiabilidad. El hecho de que las funcionalidades de búsquedas ya se encuentran desarrolladas ahorrará tiempo a futuros desarrollos dentro del sistema.

Es importante notar que se cumple con uno de los principales objetivos que es el costo en software aplicativo y de programación. Se utilizó 100% software abierto encontrado en internet, también se apalancó con el hardware ya instalado en la institución. El costo económico de la construcción también fue nulo ya que se desarrolló por el equipo que expone este trabajo.

La Facultad de Ingeniería en la carrera de Ingeniería en Computación, nos ha formado con los conocimientos y herramientas para poder trabajar en la solución global del sistema de inventarios y ticketing. También la experiencia laboral adquirida nos ayudó a llevar la realidad y el manejo de tiempos en las diferentes fases del sistema y aplicarlas en las necesidades de construcción del sistema.

Los procesos de auditoría y seguimiento de los activos de hardware se han simplificado sustancialmente para el personal de la ENALLT que debe realizarlos. La adición de la administración de direcciones IP homologadas ha permitido iniciar procesos de reasignación y depuración de una manera rápida y eficiente.

El aprendizaje de la tecnología Django nos permitirá continuar desarrollando sistemas de manera veloz y con alta efectividad en el cumplimiento de los requerimientos de nuestros futuros clientes.

BIBLIOGRAFÍA

Signoret Dorscaberro, Alina Marina. *Plan de Desarrollo Institucional 2013 – 2017*, Centro de Lenguas Extranjeras (2017). Recuperado de:

http://enallt.unam.mx/img/actividades/PDI_CELE_2013_2017.pdf

Signoret Dorscaberro, Alina Marina. *Informe de Labores 2013 – 2017*, Centro de Lenguas Extranjeras (2017). Recuperado de:

http://enallt.unam.mx/documentos/informes/Informe_2009_2017.pdf

Blackburn, Simon. *First – Order Language*. Oxford University Press, 2006.

Mellema, Gregory. *A Formal Language*. Oxford University Press, 2002.

Shapiro, Stewart. *Classical Logic*. Stanford Encyclopedia of Philosophy. 2009

Gonzalez Duque, Raul. *Python para todos*, Creative Commons. Recuperado de:

<http://mundogeek.net/tutorial-python/>

Martelli, Alex. *Python. Guía de referencia*. tr: Gorjón Salvador, Bruno (1 edición). Anaya Multimedia-Anaya Interactive, 2007.

Riehle, Dirk. *Framework Design: A Role Modeling Approach*, Swiss Federal Institute of technology

Universidad Nacional Autónoma de México, Dirección General de Patrimonio Universitario, *Alta de Bienes Muebles Nacionales Adquiridos bajo la modalidad de compra directa, por el folio DCI/AC*. Recuperado de:

http://www.patrimonio.unam.mx/patrimonio/ver_Trm_pdf.php?ids=58

Universidad Nacional Autónoma de México, Asociación Autónoma de Personal Académico de la Universidad Nacional Autónoma de México (AAPAUNAM), *Legislación Académico – Laboral Universitaria*. 5ª edición, 2013. Recuperado de:

<http://www.personal.unam.mx/Docs/Contratos/LegislacionAcadLabUni.pdf>

Zhang D., Adipat, B. *Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications*. International Journal of Human-Computer Interaction, vol. 18, 2005

Nielsen, J. *The Usability Engineering Life Cycle*. IEEE Computer, vol. 25, no. 3, 1992.

Tahir, A., Ahmad, R. *An AOP-Based Approach for Collecting Software Maintainability Dynamic Metrics*. IEEE Second International Conference on Computer Research and Development, 2010

Component Diagrams in UML 2.0, recuperado de:

http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_componentdiagram.html

Universidad Nacional Autónoma de México, *Consejo Universitario*. Recuperado de:
<https://consejo.unam.mx/pleno/acuerdos>

Universidad Nacional Autónoma de México, *Fundación UNAM*. Recuperado de:
<http://www.fundacionunam.org.mx/donde-paso/cele-se-transforma-en-escuela-nacional-de-lenguas/>

Universidad Nacional Autónoma de México, *Escuela Nacional de Lenguas, Lingüística y Traducción*. Recuperado de:
<http://enallt.unam.mx>

Django Project. *Introduciendo Django 0.90*. Recuperado de:
<https://www.djangoproject.com/weblog/2005/nov/16/firstrelease/>

Universidad Nacional Autónoma de México, Dirección General de Patronato Universitario, *Sistema Integral de Control Patrimonial (SICOP)*. Recuperado de:
<http://www.patrimonio.unam.mx/sicopimx/>

My SQL Wordpress. Ventajas y Desventajas. Recuperado de:
<http://mysqldaniel.wordpress.com/ventajas-y-desventajas/>

My SQL. Recuperado de:
<http://www.mysql.com/>

Lamarca Lapuente, María Jesús. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen*. Recuperado de:
<http://www.hipertexto.info/documentos/html.htm>

Hawkins, Scott. *Apache Web Server Administration & e-commerce Handbook*. Prentice Hall, 2001

Palmero, Fernando. *Metodología de solución de problemas*. Recuperado de:
http://aprendizajealgoritmos.blogspot.mx/2011/02/metodologia-de-solucion-de-problemas_16.html

Jiménez, Daniel. *El diagrama de contexto*. Recuperado de:
<http://www.pymesycalidad20.com/el-diagrama-de-contexto-tutoriales-para-la-iso-90012015.html>

Sommerville, Ian. *Ingeniería de software*. Addison-Wesley Iberoamericana, 2002.
Usage of JavaScript for websites. Recuperado de:
<https://w3techs.com/technologies/details/cp-javascript/all/all>

Netscape and Sun announce JavaScript. Recuperado de:

<https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html>

Novak et al., *Measuring the customer experience in online environments: a structural modeling approach.* Mark Sci, 19 (1) (2000)

Crowe, Jonathan. *Must-know phishing statistics 2017.* Recuperado de:

<https://blog.barkly.com/phishing-statistics-2017>

CSS. MDN Web docs. Recuperado de:

<https://developer.mozilla.org/es/docs/Web/CSS>