



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“REINGENIERÍA APLICADA AL SISTEMA DE ACTIVO
FIJO QUE UTILIZA LA UNIDAD DEPARTAMENTAL DE
CONTROL DE INVENTARIOS ADSCRITA AL TRIBUNAL
SUPERIOR DE JUSTICIA DE LA CIUDAD DE MÉXICO”**

T R A B A J O E S C R I T O
EN LA MODALIDAD DE DESARROLLO DE UN
CASO PRÁCTICO QUE PARA OBTENER EL
TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N :
DIEGO ARMANDO ROSETE PÉREZ
IVÁN OMAR CASANOVA SÁNCHEZ



ASESOR: M. EN C. MARCELO PÉREZ MEDEL

MÉXICO, 2017.



Universidad Nacional
Autónoma de México



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

El desarrollo de este proyecto es el producto de un gran esfuerzo impulsado por la institución que nos abrió las puertas y nos forjó como profesionistas, por lo cual agradecemos profundamente a la máxima casa de estudios, la Universidad Nacional Autónoma de México y a la Facultad de Estudios Superiores Aragón por habernos brindado la oportunidad y el apoyo para alcanzar este gran logro.

Además como parte fundamental de este trabajo, es necesario reconocer y agradecer las atenciones otorgadas por parte de nuestro mentor, el M. en C. Marcelo Pérez Medel quien se encargó de orientarnos a lo largo de este proceso; así mismo a quienes fueron comisionados para la revisión y desarrollo de esta documentación, al igual que a toda la docencia de nuestra facultad.

Por otra parte, agradecemos infinitamente al organismo que nos permitió llevar a cabo este proyecto haciendo uso de los recursos de esta H. institución, el Tribunal Superior de Justicia de la Ciudad de México y más específicamente a nuestros superiores jerárquicos, la Lic. Elba Triana Gómez y a la C.P. Dora Luz Avendaño Martínez con quienes nos encontramos en deuda.

A nuestros respectivos padres, Rafael Casanova Hernández y Gracia Patricia Sánchez Correa, Silvestre Rosete García y Judith Pérez Durán; quienes nos brindaron la formación necesaria para alcanzar nuestras metas y por el cariño y apoyo que nos fue brindado. También a nuestros respectivos hermanos Rafael Benjamín Casanova Sánchez, Héctor Silvestre Rosete Pérez, Iván Rosete Pérez y Alma Judith Rosete Pérez por todas las enseñanzas y experiencias compartidas a través de los años.

Finalmente a todas esas personas cercanas, familiares y amigos que nos han acompañado y apoyado a lo largo del tiempo, compartiendo emociones, experiencias y conocimiento, quienes de alguna manera forman parte de este trabajo; entre los que destacan nombres como Carolina Pérez Piña, Sergio Rosete García, Jesús Rosete García, Miguel Ángel Torres Reynoso, Juan Carlos Camacho Álvarez, entre muchos otros más.

ÍNDICE

INTRODUCCIÓN	4
MARCO TEÓRICO	7
NÚMERO DE INVENTARIO	8
BASE DE DATOS	11
MySQL	12
PHP	12
HTTP/HTML/CSS	14
ANTECEDENTE DEL SISTEMA DE ACTIVO FIJO (VERSIÓN 1.0)	18
DESCRIPCIÓN	19
FUNCIONES QUE DESEMPEÑA.....	20
ANÁLISIS	31
DEFINICIÓN DEL PROBLEMA	32
REQUERIMIENTOS	36
OBJETIVO GENERAL	37
OBJETIVOS ESPECÍFICOS.....	37
JUSTIFICACIÓN DEL PROYECTO	37
DESARROLLO	38
CORRECCIÓN DE ERRORES.....	39
NUEVAS IMPLEMENTACIONES	46
RESULTADOS	63
CONCLUSIONES	67
BIBLIOGRAFÍA	71

INTRODUCCIÓN

La unidad departamental de control de inventarios del Tribunal Superior de Justicia de la Ciudad de México, tiene como objetivo, según el manual organizacional, “Asegurar que se lleve el registro y control del activo fijo, mediante el manejo de los sistemas y procedimientos diseñados con tales propósitos.” (Manual de organización: Dirección Ejecutiva de Recursos Materiales, 2005).

Inicialmente se contaba con un sistema de registro que reside en la estación en la que está siendo utilizada, por lo que para compartir la información era necesario transferir un respaldo completo de un equipo a otro; sin embargo, a través del tiempo estos procedimientos y sistemas han ido sufriendo modificaciones a efecto de mejorar la calidad del trabajo, por lo que en 2014 fue contratada la empresa denominada “Vlix Soluciones S.A. de C.V.” quien se encargaría del diseño y desarrollo de un nuevo sistema que controlaría el registro del activo fijo conforme a los requerimientos del área y con apego a las normatividades vigentes.

Por esta razón se creó un sistema web diseñado para registrar cada bien mueble e inmueble que tiene en posesión la institución, el cuál utiliza la arquitectura de software modelo-vista-controlador, reside en un servidor físico de torre con sistema operativo CentOS¹ (distribución de Linux) y utiliza MySQL² como servidor de base de datos.

Pese a que aún no es obsoleto, han surgido nuevas exigencias que es necesario atender para mantener vigente la utilidad del Sistema de Activo Fijo (SAF), es por esto que nace la tarea de atender esas necesidades y lograr mantener actualizado el sistema.

El capítulo primero, comprende los fundamentos teóricos que sustentan la base del sistema, aquí se especifica la *lógica del negocio* con la cual fue creado originalmente, así como una breve descripción de las tecnologías utilizadas.

El capítulo segundo habla sobre cómo cada una de las tecnologías mencionadas se relaciona con el sistema, explicando también el funcionamiento y los alcances que tuvo originalmente.

Posteriormente se analizan la problemática, los objetivos y los requerimientos a través del capítulo 3, lo cual proporciona una justificación que fundamenta la totalidad del proyecto.

¹ Véase en <https://www.centos.org/>

² Véase en <https://www.mysql.com/>

En el capítulo denominado “Desarrollo”, se muestran mediante capturas de pantalla y fragmentos de código, las correcciones e implementaciones logradas con base en la problemática y los objetivos planteados.

Finalmente los resultados del proyecto son presentados en el capítulo quinto, el cual brinda un panorama general de los objetivos alcanzados, visualizando una comparación de “antes” y “después” de la implementación de este proyecto.

CAPÍTULO I

MARCO TEÓRICO

NÚMERO DE INVENTARIO:

Un número de inventario es un ID único que es asignado a cada elemento que comprende el patrimonio de una institución, empresa o persona que requiera o se encuentre obligada a llevar un registro de esta naturaleza. Lo anterior con la finalidad de poder mantener un control de los materiales, suministros, bienes muebles, inmuebles e intangibles con los que se cuenta actualmente.

En este caso particular se trata de todas las instituciones gubernamentales adscritas al Gobierno de la Ciudad de México (GCDMX). Si bien es cierto que este proyecto se desarrolla específicamente con el Sistema de Activo Fijo del Tribunal Superior de Justicia de la Ciudad de México, también se debe tomar en cuenta que el Sistema de Control Gubernamental (SCG) será el mismo para cada ente público de esta región.

Para esto, el Consejo Nacional de Armonización Contable (CONAC) se dio a la tarea de crear los “Lineamientos para la elaboración del Catálogo de Bienes que permita la interrelación automática con el Clasificador por Objeto del Gasto y la Lista de Cuentas”, publicados en el Diario Oficial de la Federación el 13 de Diciembre de 2011. Se establece, en su apartado VIII, la estructura básica que debe cumplir nuestro número de inventario y que a la letra indica:

“Su codificación consta de cinco (5) niveles numéricos estructurados de la siguiente manera:

- El primer nivel, identifica al GRUPO de bienes conformado por un dígito y se relaciona con el Capítulo del Clasificador por Objeto del Gasto.*
- El segundo nivel identifica el SUBGRUPO de bienes dentro del catálogo consta de un dígito y se relaciona con el Concepto del Clasificador por Objeto del Gasto.*
- El tercer nivel, identifica la CLASE de bienes, se relaciona con la Partida Genérica del Clasificador por Objeto del Gasto, conformado por 1 dígito.*
- El cuarto nivel, identifica la SUBCLASE de bienes se relaciona con la Partida Específica del Clasificador por Objeto del Gasto.*
- El quinto nivel, finalmente, identifica al NÚMERO CONSECUTIVO asignado a cada bien mueble de un total con las mismas características. Estará numerado en forma correlativa.”* (Acuerdo por el que se emiten los Lineamientos para la elaboración del

Catálogo de Bienes que permita la interrelación automática con el Clasificador por Objeto del Gasto y la Lista de Cuentas, 2011).

Es preciso mencionar que los primeros cuatro niveles numéricos (grupo, subgrupo, clase y subclase) conforman la partida presupuestal del número de inventario, lo cual representa una herramienta para la planeación estratégica del manejo de los recursos financieros.

El siguiente nivel numérico, el quinto, se forma por 6 dígitos y describe el tipo de bien específico, considerando el material del que está hecho y la función que cumple para poder asociarlo a una partida presupuestal.

Finalmente, se asigna un número progresivo e irrepetible que para el caso del TSJCDMX consta de 6 dígitos y cumple la función de identificador para cada bien que responde a las mismas características quedando de la siguiente manera:

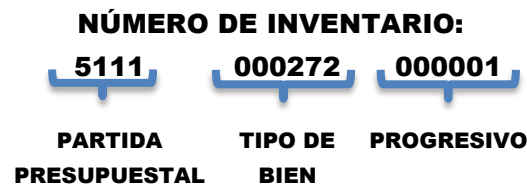


Figura 1.1 Estructura del número de inventario.

Para facilitar el manejo de esta información dentro de SAF, los 16 dígitos del número de inventario se dividen en dos secciones, la primera sección es llamada “Código del Bien” y está conformado por la partida presupuestal y el tipo de bien, lo que corresponde a los primeros 10 dígitos; la segunda parte son los 6 dígitos restantes conocidos como progresivo o “No. Activo”. Esta información es ingresada al sistema como se muestra:

Captura General de Activos				
Sociedad	<input type="text" value="TSJ"/>	Código del Bien	<input type="text" value="5111000272"/> ...	
Descripción	<input type="text" value="SILLA GIRATORIA"/>		No.Activo	<input type="text" value="000001"/>

Figura 1.2 Número de inventario de un activo

En ocasiones se llegan a adquirir bienes que forman parte de un mismo conjunto, el cual es conocido como **sistema o módulo**; cada elemento o cada bien que lo comprende se conoce como **componente**. Dado que no se tiene el precio unitario de cada componente por los que está comprendido un sistema, la asignación del número progresivo debe incluir un sub-número con el cual se podrá registrar y controlar cada uno de estos componentes sin dejar de lado el registro contable que debe ser conciliado entre la Dirección Ejecutiva de Recursos Financieros y la Dirección Ejecutiva de Recursos Materiales.

Ya que todos los componentes de un mismo módulo o sistema llevan el mismo número de inventario, también se asocia un sub-número y un código a cada elemento que hace referencia a una descripción más específica de lo que es físicamente tal como se muestra a continuación:

MÓDULO DESARMABLE	
5111000212	
NÚMERO DE INVENTARIO:	NÚMERO DE INVENTARIO:
5111000212000001	5111000212000001
SUB-NÚMERO:	SUB-NÚMERO:
1	2
CÓDIGO:	CÓDIGO:
ESCRITORIO DE MADERA	PERCHERO DE METAL
5111000110	5111000228

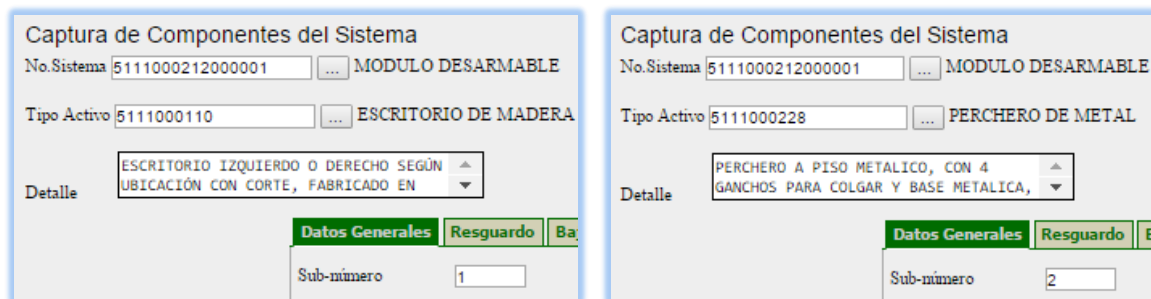


Figura 1.3 Número de inventario de un componente

El Sistema de Activo Fijo del TSJCDMX, con el que se desarrolla este proyecto, se encarga de controlar sólo los bienes muebles, inmuebles e intangibles de la institución (Capítulo 5000), por lo que

únicamente se focalizan los puntos que hacen referencia a este tipo de bienes, los cuales se pueden consultar en el Catálogo de Adquisición de Bienes Muebles y Servicios de la Ciudad de México (Catálogo De Bienes Muebles y Servicios Del Gobierno De La Ciudad De México, 2016).

BASE DE DATOS:

Se le llama base de datos a los bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico en campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados sistemas gestores (o de gestión) de bases de datos, abreviado SGBD (del inglés Database Management System o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son las utilizadas para la gestión de empresas e instituciones públicas; también son ampliamente ocupadas en entornos científicos con el objeto de almacenar la información experimental a través del tiempo.

Existen diferentes tipos de clasificaciones para las bases de datos, sin embargo la clasificación más comúnmente utilizada es de acuerdo a su modelo de administración de datos. Este tipo de clasificación, refiere las bases de datos conforme a lo siguiente:

- Bases de datos jerárquicas
- Bases de datos de red

- Bases de datos transaccionales
- Bases de datos relacionales
- Bases de datos multidimensionales
- Bases de datos orientadas a objetos
- Bases de datos documentales
- Bases de datos deductivas

MySQL:

Es un sistema de administración de bases de datos de tipo relacional (RDBMS por sus siglas en ingles). Se trata de un software capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos.

Este sistema incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos.

MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de datos relacionales. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos determinados por la consulta o el código generado. (Gilfillan, 2003)

PHP:

PHP es un software libre, licenciado bajo la PHP Licence³. Sus siglas originalmente se referían a Personal Home Page ya que así lo nombró su creador Rasmus Lerdorf en 1994. Posteriormente fue retomado por programadores Israelíes (Zeev Suraski y Andi Gutmans), creando la 3ra versión del

³ PHP Licence es la licencia bajo la que se publica el lenguaje de programación PHP. Es una licencia de software libre no copyleft (no exige que las modificaciones o extensiones del software sean libres) y una licencia de código abierto. No es compatible con la licencia GPL debido a la restricción en el uso del término "PHP". Es posible consultar la licencia y cada una de sus cláusulas en: http://www.php.net/license/3_01.txt. (https://es.wikipedia.org/wiki/Licencia_PHP, 2017).

lenguaje y renombrándolo como Hypertext Pre-processor. Actualmente el desarrollo del lenguaje se lleva a cabo por “The PHP Group”. (Miguel A. Arias, 2013; Php.net, 2017)

Como Miguel A. Arias (2013) indica, la principal característica de PHP es que es un lenguaje ejecutado del lado del servidor⁴ (en este caso un servidor web APACHE⁵), el cual responde al recibir una petición por parte de un cliente, esta comunicación se da mediante un protocolo HTTP; otras características de PHP son:

- Velocidad y robustez.
- Estructurado y orientado a objetos.
- Portabilidad –independencia de plataforma-.
- Open-Source.

PHP es un lenguaje de programación de alto nivel, por lo que la forma de programar se asemeja más al lenguaje humano y menos a la lógica de ejecución de una máquina, de esta forma se pueden programar expresiones lógicas, como expresiones booleanas, decisiones, ciclos o funciones, sin necesidad de manipular directamente los registros y direcciones de memoria.

Cabe mencionar que es un lenguaje interpretado, lo que se refiere a que la ejecución del código se realiza línea por línea sobre el código fuente sin generar un archivo ejecutable, es por esto que en caso de existir un error, la ejecución se detiene en ese punto. El producto de este proceso es un código HTML, el cual es posible interpretar de forma nativa por un navegador en un equipo cliente.

⁴ Programa diseñado para permitir la interacción entre ordenadores. Suele funcionar permaneciendo a la espera de peticiones, cuando las recibe responde a ellas transfiriendo documentos de tipo hipertexto, para ello implementa el protocolo HTTP (HyperText Transfer Protocol). El término también se emplea para referirse al ordenador que ejecuta el programa. (Servidores Web - Concepto, configuración y uso, 2017)

⁵ El Servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP. (Servidor HTTP Apache, Ecured.cu, 2017))

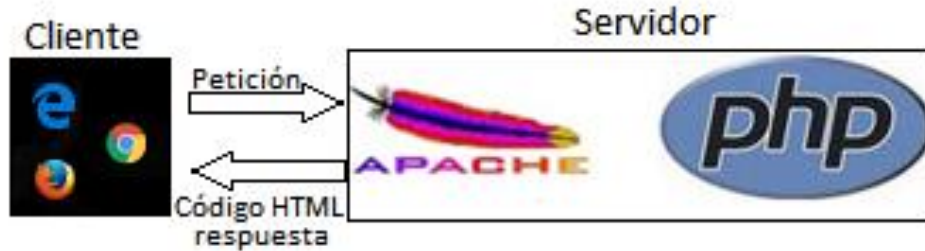


Figura 1.4 Relación cliente-servidor

El intérprete de PHP es ejecutado sobre un servidor web (en este caso APACHE) para enviar al cliente el resultado del código interpretado, y su funcionamiento de forma general se puede ejemplificar de la siguiente manera:

1. El cliente hace una petición desde un sitio en PHP al servidor Apache.
2. El servidor Apache llama a su módulo PHP para decodificar el script almacenado en el servidor.
3. El módulo PHP hace la petición de lectura, modificación, almacenamiento o eliminación de datos al servidor de BD MySQL
4. El módulo PHP decodifica la información obtenida y genera un código HTML que posteriormente entrega como respuesta al servidor Apache.
5. Apache responde a la petición del cliente mediante la red
6. El navegador decodifica el código HTML y la presenta en la interfaz gráfica.

HTTP/HTML/CSS:

El protocolo de transferencia de hipertexto (HTTP) es un protocolo de comunicaciones que permite la transferencia de documentos de lenguaje de marcas de hipertexto (HTML por sus siglas en inglés, HyperText Markup Language) desde servidores web a navegadores web. HTML es un lenguaje de identificadores para la creación de documentos que contienen enlaces a información relacionada. Se puede pulsar con el ratón sobre un enlace para acceder a otros documentos, imágenes u objetos multimedia, así como obtener información adicional acerca del elemento enlazado.

Para poder enviar y recibir documentos HTML e interactuar con la World Wide Web, tanto el servidor como el cliente deben soportar HTTP. (Publib.boulder.ibm.com, 2017)

El World Wide Web Consortium (W3C⁶) es la organización que desarrolla los estándares para normalizar el desarrollo y la expansión de la Web, también es quien se encarga de publicar las especificaciones relativas al lenguaje HTML. Este lenguaje se basa en el metalenguaje SGML (que es un estándar ISO: "ISO 8879: 1986 Tratamiento de la información - Sistemas de texto y de oficina - Lenguaje de marcado generalizado estándar" (Es.wikipedia.org, 2017) y es el formato de los documentos de la World Wide Web. HTML fue concebido como un lenguaje para el intercambio de documentos científicos y técnicos adaptado para su uso por no especialistas en tratamiento de documentos. (Hipertexto.info, 2017).

"HTML nace en 1991 de manos de Tim Bernes-Lee del CERN como un sistema hipertexto con el objetivo de servir como medio de transmisión de información entre científicos, como parte de la iniciativa World Wide Web. En 1993 Dan Connolly escribe la primera DTD (Document Type Definition) de SGML describiendo el lenguaje y, desde entonces, el lenguaje ha continuado evolucionando.

Los estándares oficiales HTML son: HTML 2.0, HTML 3.2, HTML 4.0, HTML 4.01 y HTML 5. Este último es la más actual especificación oficial y se espera que continúe evolucionando a lo largo de los próximos años. El XHTML es una forma más avanzada de HTML que supuestamente sustituiría a este, sin embargo puede considerarse que ha quedado integrado dentro del HTML 5.

Evolución de HTML:

HTML 2.0: En 1995 se publica el estándar HTML 2.0. A pesar de su nombre, es el primer estándar oficial de HTML, es decir que la versión 1.0 no existió como estándar. HTML 2.0 no soportaba tablas.

Se simplificaba al máximo la estructura del documento para agilizar su edición, donde la declaración explícita de los elementos body, html y head es opcional.

⁶ W3.org, 2017

HTML 3.2: La versión HTML 3.2 se publicó en 1997 y es la primera recomendación de HTML publicada por el W3C (Consortio internacional). Esta revisión incorporó los últimos avances de las páginas web desarrolladas hasta 1996, como applets de Java y texto que fluye alrededor de las imágenes.

HTML 4.01: Esta especificación de HTML se publicó en diciembre de 1999. Desde ese momento, el W3C se centró en el desarrollo del estándar XHTML. Por este motivo, en el año 2004, las empresas Apple, Mozilla y Opera mostraron su preocupación por la falta de interés del W3C en HTML y decidieron organizarse en una nueva asociación llamada WHATWG (Web Hypertext Application Technology Working Group) que comenzó el desarrollo del HTML 5, cuyo primer borrador oficial se publicó en enero de 2008. Debido a la fuerza de las empresas que forman el grupo WHATWG y a la publicación de los borradores de HTML 5.0, en marzo de 2007 el W3C decidió retomar la actividad de estandarizar HTML, dentro del cual decidió integrar el XHTML.

HTML 5, HTML 5.1, HTML 5.2: El consorcio internacional W3C, después de una evolución de varios años, liberó el HTML 5 como estándar oficial a finales de octubre de 2014, el cual incorpora nuevos elementos no contemplados en HTML 4.01. Por ejemplo hay nuevas etiquetas, se introduce la posibilidad de introducir audio y video de forma directa en la web sin necesidad de plugins o complementos en los navegadores, entre otras novedades.”

(Aprenderaprogramar.com, 2017)

HTML y CSS (Hoja de estilos en cascada) son dos de las tecnologías principales para construir páginas Web. HTML provee la estructura de la página y CSS la capa visual y auditiva para diversos dispositivos.

CSS es el lenguaje para describir la presentación de páginas Web; incluye los colores, el diseño y las fuentes. Permite adaptar la presentación a diferentes tipos de dispositivos, como pantallas grandes, pantallas pequeñas o impresoras, y al igual que HTML, CSS también es estandarizado por la W3C. CSS es independiente de HTML y puede utilizarse con cualquier lenguaje de marcado basado en XML. La separación entre HTML y CSS facilita el mantenimiento de sitios, comparte hojas de estilo

entre páginas y adapta páginas a diferentes entornos. Esto se conoce como la separación de la estructura de la presentación.

Como indica (Durango A., 2014), el uso del estándar CSS, ahorra tiempo, da consistencia y facilita la escritura de páginas web, haciéndolas más flexibles y ligeras, controlando su aspecto gráfico con mayor precisión.

CAPÍTULO II

*ANTECEDENTE DEL
SISTEMA DE ACTIVO
FIJO (VERSIÓN 1.0)*

DESCRIPCIÓN:

SAF utiliza una base de datos de tipo relacional según su estructura interna; en la cual almacena la información referente a los bienes muebles de la institución. Aquí se encuentran datos como número de serie, número de contrato o pedido y la factura con la que se adquirió el bien, marca, modelo, información de resguardo, información de baja, etc. Todo la información es almacenada en un sistema de bases de datos administrado por el gestor MySQL.

La interfaz de usuario es presentada en un explorador web por medio de un código de HTML que se enriquece con el uso de efectos y diseños de hojas de estilo de tipo CSS3 y una biblioteca denominada Scriptaculous⁷. Esta es una biblioteca de Javascript basada en Prototype heredando todas sus funcionalidades.

Para hacer más dinámica la interfaz de usuario, la biblioteca Scriptaculous realiza la comunicación con el servidor utilizando AJAX (Asynchronous Javascript and XML). El código principal del sistema y las llamadas que realiza la biblioteca al servidor son generados principalmente por código en PHP que se ejecuta desde el servidor dando respuesta a las solicitudes del cliente.

El servidor en donde se almacena el Sistema de Activo Fijo (SAF) cuenta con las siguientes características:

⁷ Script.aculo.us es una de las muchas bibliotecas que han surgido para facilitar el desarrollo de aplicaciones JavaScript y que están basadas en Prototype. El autor original de la librería es Thomas Fuchs, aunque actualmente recibe contribuciones de numerosos programadores, ya que la librería se distribuye de forma completamente gratuita y dispone de una buena documentación: <http://wiki.script.aculo.us/scriptaculous/> (Eguíluz Pérez, J., 2005)

<pre>[root@Vlix ~]# df -h Filesystem Size Used Avail Use% Mounted on /dev/mapper/vg_vlix-lv_root 909G 139G 725G 17% / tmpfs 7,8G 72K 7,8G 1% /dev/shm /dev/sda2 485M 96M 364M 21% /boot /dev/sdal 200M 260K 200M 1% /boot/efi</pre>	<p>Información de disco duro y particiones</p>
<pre>[root@Vlix ~]# top top - 11:09:36 up 202 days, 4:41, 1 user, load average: 0.00, 0.02, 0.00 Tasks: 174 total, 1 running, 173 sleeping, 0 stopped, 0 zombie Cpu(s): 0.1%us, 0.1%sy, 0.0%ni, 99.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st Mem: 16284124k total, 12015380k used, 4268744k free, 174568k buffers Swap: 8208376k total, 1100k used, 8207276k free, 10979404k cached</pre>	<p>Información de memoria RAM</p>
<pre>root@Vlix ~]# cat /proc/cpuinfo vendor_id : GenuineIntel cpu family : 6 model : 58 model name : Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz stepping : 9 cpu MHz : 1600.000 cache size : 8192 KB cpu cores : 4</pre>	<p>Información de Procesador</p>

Figura 2.1 Características del servidor

Además, utiliza un sistema operativo CentOS versión 6.5 Final, el cual mantiene en ejecución a un servidor APACHE versión 2.2.15 (Unix); Con soporte para PHP versión 5.3.3 y un gestor de base de datos MySQL versión 5.1.71 Source distribution.

FUNCIONES QUE DESEMPEÑA:

La pantalla principal de SAF consiste en un sistema de autenticación de identidad basado en algo conocido, es decir que permite el acceso conforme a un nombre de usuario y una contraseña. Existe un usuario administrador que posee todos los permisos, por lo cual es quien se encarga de crear cuentas para los nuevos usuarios y asignar los permisos respectivos para poder acceder a los diferentes apartados dentro de SAF. La pantalla de ingreso es igual a la de la imagen siguiente:



Figura 2.2 Pantalla de acceso a SAF

Una vez que el usuario se ha autenticado se muestra en la parte superior izquierda de la pantalla la clasificación de las acciones que se pueden realizar según los permisos asignados por el administrador; se trata de un menú dinámico en el cuál se despliega una lista de tareas cuando el cursor se posiciona sobre alguna de las categorías tal como se muestra en la imagen próxima:

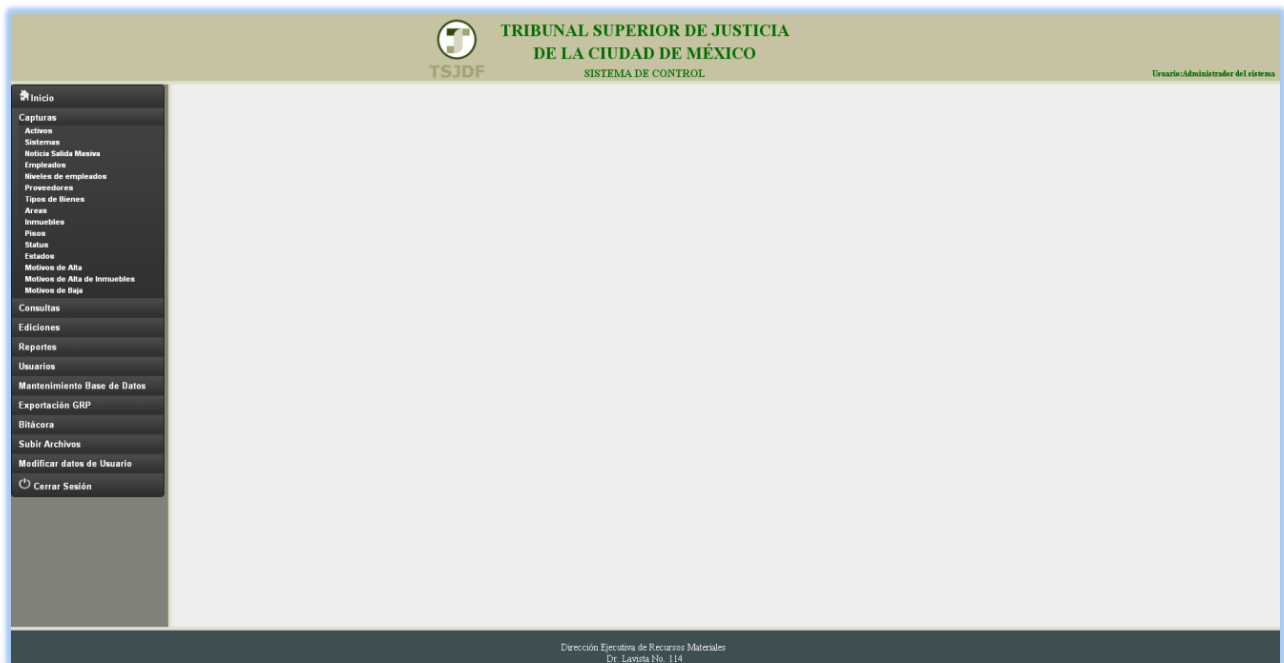


Figura 2.3 Pantalla principal y menú de SAF

CAPÍTULO II: ANTECEDENTE DEL SISTEMA DE ACTIVO FIJO (VERSIÓN 1.0)

En la categoría de capturas se lleva a cabo un proceso de inserción (INSERT INTO) o actualización (UPDATE) de registros, desde aquí es posible acceder a las tablas de activos y sistemas respectivamente para dar de alta nuevos activos o modificar la información relacionada con algún número de inventario. Cabe mencionar que toda la información asociada a un activo no está concentrada en una sola tabla, sin embargo el SAF se encarga de generar las relaciones necesarias para agregar toda la información en un solo movimiento.

Captura General de Activos

Sociedad: TSJ Código del Bien: 5111000270 No. Activo: 000002 Código Anterior: I450400318000002

Descripción: SILLA DE METAL Estado del Activo: ALTA Componentes:

Detalle: FIJA APILABLE

Datos Generales Resguardo Bajas

Motivo de Alta: 01AM - CI Unidad de medida: PZA Estado: Costo Adquisición: 83.85 Captura: Administrador

No. Contrato/Pedido: P 248/1989 No. Factura: R 4595 Fecha de Factura: 23/02/1990 Último Cambio: 15/01/2015

Proveedor: Clave: 10000546 Nombre: INDUSTRIAS RIVIERA S.A. Vida Útil (años): 13 Usuario: MIVESCAG

Partida Presupuestal: 5111 Tiempo de Garantía: Motor: Vida Remanente: 0

Fecha Contrato/Pedido: 06/12/1989 Modelo: S/MOD. Valor Catastral: Cantidad:

Marca: S/MCA Serie: Placas:

Guardar Activo

Figura 2.4 Pantalla de captura de activos

Captura de Componentes del Sistema

No. Sistema: 5111000212002125 MODULO DESARMABLE

Tipo Activo: 5111000110 ESCRITORIO DE MADERA

Detalle: ESCRITORIO RECTO CON DOS CAJONERAS Y FALDON, FABRICADO CON TABLEROS DE FIBRA Estado del Componente: ALTA

Datos Generales Resguardo Bajas

Sub-número: 1 Unidad de medida: PZA Estado: Costo Adquisición: Último Cambio: 08/03/2016

Motivo de Alta: 01AM No. Factura: F A2065777 Fecha de Factura: 28/10/2014 Usuario:

No. Contrato/Pedido: C TSJDF/DERM/26/2014 Proveedor: Clave: 10000546 Nombre: INDUSTRIAS RIVIERA S. Vida Útil (años):

Partida Presupuestal: 5111 Tiempo de Garantía: 10 AÑOS Motor: Vida Remanente: MIVESCAG

Fecha Contrato/Pedido: 16/09/2014 Modelo: S/MOD. Valor Catastral: Cantidad:

Marca: S/MCA Serie: Placas:

Guardar Componente

Figura 2.5 Pantalla de captura de componentes

Además, en este apartado es posible modificar todos los catálogos que utiliza la base de datos, los cuales se pueden observar a continuación:

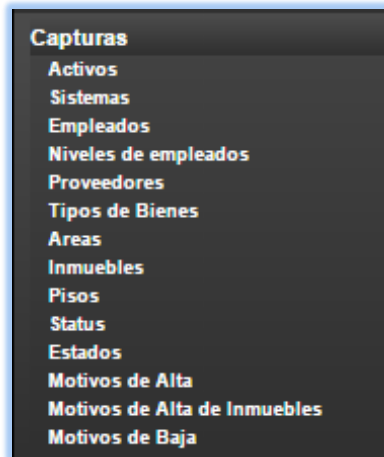


Figura 2.6 Menú de capturas

Para la sección de consultas existen dos sub-categorías disponibles: activos y catálogos.

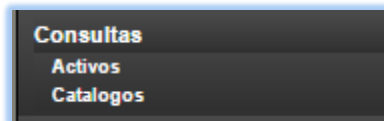


Figura 2.7 Menú de consultas

En la primera, el sistema genera una consulta (SELECT) incluyendo una condición (WHERE) restringida por el número de inventario y para evitar modificaciones todas las cajas de texto se encuentran bloqueadas. Es preciso mencionar que únicamente es posible visualizar la información de un solo activo a la vez.

Consulta General de Activos

Sociedad Código del Bien No.Activo Código Anterior

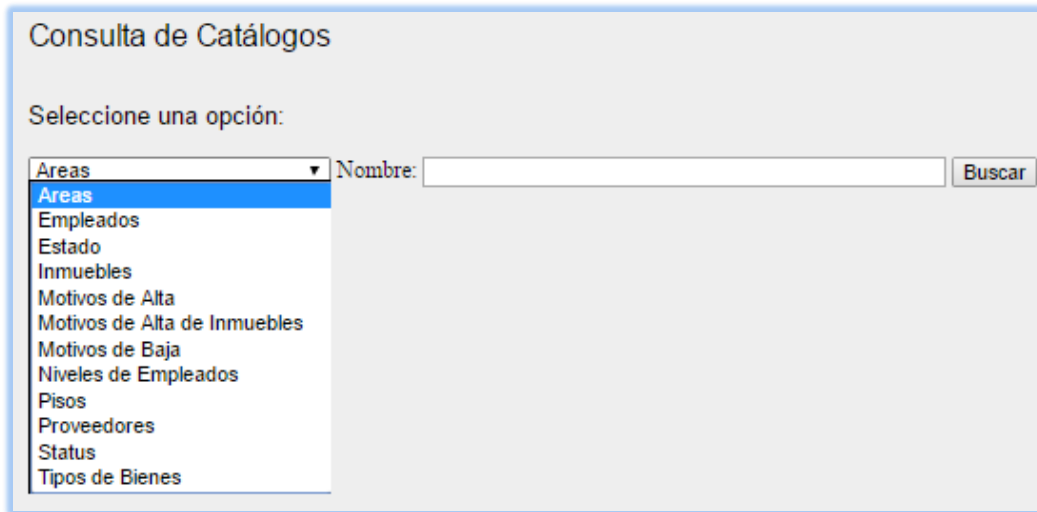
Descripción Estado del Activo: Componentes

Detalle

Datos Generales	Resguardo	Bajas
Motivo de Alta <input type="text" value="01AM"/>	Unidad de medida <input type="text" value="PZA"/>	Estado <input type="text"/>
COMPRAS	No.Factura <input type="text" value="R. 4595"/>	Fecha de Factura <input type="text" value="23/02/1990"/>
No.Contrato Pedido <input type="text" value="P 248/1989"/>	Clave <input type="text" value="10000546"/>	Costo Adquisición <input type="text" value="83.85"/>
Proveedor: <input type="text"/>	Tiempo de Garantía <input type="text"/>	Vida Útil (años) <input type="text" value="13"/>
Partida Presupuestal <input type="text" value="5111"/>	Modelo <input type="text"/>	Vida Remanente <input type="text" value="0"/>
Fecha Contrato Pedido <input type="text" value="06/12/1989"/>	S/MOD. <input type="text"/>	Usuario Alta: Administrador del sistema
Marca <input type="text" value="S/MCA."/>	Serie <input type="text"/>	Ultimo Cambio: 15/01/2015
		Usuario: MARGARITA IYESCAS GUTIERREZ

Figura 2.8 Pantalla de consulta de activos

Por otra parte, para la visualización de los catálogos, la consulta (SELECT) muestra todos los atributos de una tabla excluyendo únicamente el ID; sin embargo también es posible realizar una búsqueda más específica en la cual la condición (WHERE) es delimitada por un campo de texto.



The screenshot shows a web interface titled "Consulta de Catálogos". Below the title, it says "Seleccione una opción:". There is a dropdown menu currently showing "Areas" with a list of other options: Empleados, Estado, Inmuebles, Motivos de Alta, Motivos de Alta de Inmuebles, Motivos de Baja, Niveles de Empleados, Pisos, Proveedores, Status, and Tipos de Bienes. To the right of the dropdown is a text input field labeled "Nombre:" and a "Buscar" button.

Figura 2.9 Pantalla de consulta de catálogos

El menú de reportes contiene diversas estructuras previamente almacenadas que brindan la oportunidad de generar ciertas consultas de manera muy ágil. Aquí se pueden generar los siguientes tipos de reporte:

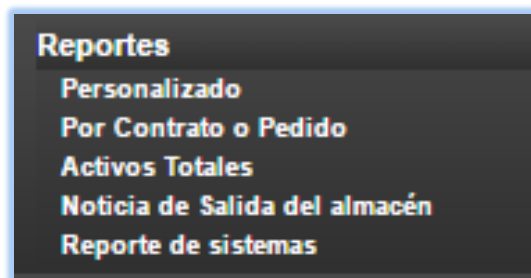


Figura 2.10 Menú de reportes

El reporte personalizado representa uno de los módulos más complejos del SAF, este se encarga de generar una consulta (SELECT) en función de los parámetros establecidos por el usuario y lo exporta a una hoja de cálculo. Desde aquí es posible realizar agrupaciones de información (GROUP BY) entre

otras condiciones permitidas, además de ser capaz de obtener un reporte únicamente con los campos necesarios que el usuario puede seleccionar de una lista si desea obtener toda la información contenida o seleccionar solamente un intervalo de datos (BETWEEN).

Fecha Reporte

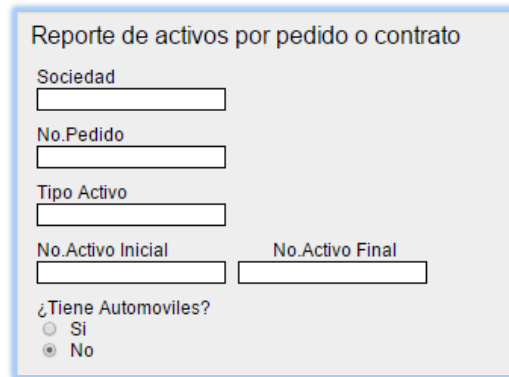
Seleccionar Todos

Clasificación	Tot/Pag	Desde	Hasta	Desde	Hasta
Nivel 1: <input type="text"/>	<input type="radio"/> Si <input type="radio"/> No	<input type="checkbox"/> 01 Sociedad	<input type="text"/>	<input type="checkbox"/> 31 Num Pedido	<input type="text"/>
Nivel 2: <input type="text"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>	<input type="checkbox"/> 02 Código del Bien	<input type="text"/> ... <input type="text"/> ...	<input type="checkbox"/> 32 No Motor	<input type="text"/> <input type="text"/>
Nivel 3: <input type="text"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>	<input type="checkbox"/> 03 Num Activo	<input type="text"/>	<input type="checkbox"/> 33 Proveedor	<input type="text"/> ... <input type="text"/> ...
Nivel 4: <input type="text"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>	<input type="checkbox"/> 04 Area	<input type="text"/> ... <input type="text"/> ...	<input type="checkbox"/> 34 Código Anterior	<input type="text"/>
Nivel 5: <input type="text"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>	<input type="checkbox"/> 05 Empleado	<input type="text"/>	<input type="checkbox"/> 35 Edificio	<input type="text"/>
Altas/Bajas <input type="text"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>	<input type="checkbox"/> 06 Motivo Alta	<input type="text"/>	<input type="checkbox"/> 36 Piso	<input type="text"/>
Mot.Baja <input type="text"/>	<input type="checkbox"/> 07 Motivo Alta de Inmueble	<input type="checkbox"/> 08 Folio de Resguardo	<input type="text"/>	<input type="checkbox"/> 37 Destino Final	<input type="text"/>
Mot.Alta <input type="text"/>	<input type="checkbox"/> 09 Serie	<input type="checkbox"/> 09 Serie	<input type="text"/>	<input type="checkbox"/> 38 Usuario que Capturo	<input type="text"/>
	<input type="checkbox"/> 10 Marca	<input type="checkbox"/> 10 Marca	<input type="text"/>	<input type="checkbox"/> 39 Usuario que realizo Ultimo Cambio	<input type="text"/>
	<input type="checkbox"/> 11 Modelo	<input type="checkbox"/> 11 Modelo	<input type="text"/>	<input type="checkbox"/> 40 Importe de Venta	<input type="text"/>
	<input type="checkbox"/> 12 No Factura	<input type="checkbox"/> 12 No Factura	<input type="text"/>	<input type="checkbox"/> 41 Motivo Baja Definitiva	<input type="text"/>
	<input type="checkbox"/> 13 Placas	<input type="checkbox"/> 13 Placas	<input type="text"/>		
	<input type="checkbox"/> 14 Costo Adq	<input type="checkbox"/> 14 Costo Adq	<input type="text"/>	Fechas	Desde Hasta
	<input type="checkbox"/> 15 Descripción	<input type="checkbox"/> 15 Descripción	<input type="text"/>	42 Fecha Nota de Entrada	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 16 Status(Lugar Físico)	<input type="checkbox"/> 16 Status(Lugar Físico)	<input type="text"/> ...	43 Fecha Noticia de Salida	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 17 Estado	<input type="checkbox"/> 17 Estado	<input type="text"/>	44 Fecha de Resguardo	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 18 Partida Presupuestal	<input type="checkbox"/> 18 Partida Presupuestal	<input type="text"/>	45 Fec Factura	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 19 Tiempo de Garantía	<input type="checkbox"/> 19 Tiempo de Garantía	<input type="text"/>	46 Fecha Contrato/Pedido	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 20 Valor Catastral	<input type="checkbox"/> 20 Valor Catastral	<input type="text"/>	47 Fec Ult Cambio	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 21 Folio Noticia Entrada	<input type="checkbox"/> 21 Folio Noticia Entrada	<input type="text"/>	48 Fec de Captura	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 22 Folio Noticia Salida	<input type="checkbox"/> 22 Folio Noticia Salida	<input type="text"/>	49 Fec Baja	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 23 Mot Baja del Área	<input type="checkbox"/> 23 Mot Baja del Área	<input type="text"/>	50 Fec Reasignación	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 24 Area Ant.	<input type="checkbox"/> 24 Area Ant.	<input type="text"/> ... <input type="text"/> ...	51 Fecha de Retiro	<input type="text"/> <input type="text"/>
	<input type="checkbox"/> 25 No Dictamen	<input type="checkbox"/> 25 No Dictamen	<input type="text"/>	Datos Adicionales	
	<input type="checkbox"/> 26 Detalle	<input type="checkbox"/> 26 Detalle	<input type="text"/>	52 Observaciones Resguardo	<input type="text"/>
	<input type="checkbox"/> 27 Unidad de medida	<input type="checkbox"/> 27 Unidad de medida	<input type="text"/>	53 Observaciones Baja	<input type="text"/>
	<input type="checkbox"/> 28 Estado físico	<input type="checkbox"/> 28 Estado físico	<input type="text"/>	54 Sesión del Comite Que Se Autoriza La Baja	<input type="text"/>
	<input type="checkbox"/> 29 Acta o CTTO. Venta	<input type="checkbox"/> 29 Acta o CTTO. Venta	<input type="text"/>	55 Componentes de un Sistema	<input type="text"/>
	<input type="checkbox"/> 30 Averiguacion Previa	<input type="checkbox"/> 30 Averiguacion Previa	<input type="text"/>	56 Vida Util (años)	<input type="text"/>
				57 Vida Remanente	<input type="text"/>

CREAR EXCEL

Figura 2.11 Pantalla de reporte personalizado.

El reporte por contrato o pedido es muy solicitado, por esta razón existe una estructura que lo genera de manera rápida si se ingresa información del expediente que se requiere, tal como se muestra en la figura de abajo.



Reporte de activos por pedido o contrato

Sociedad

No.Pedido

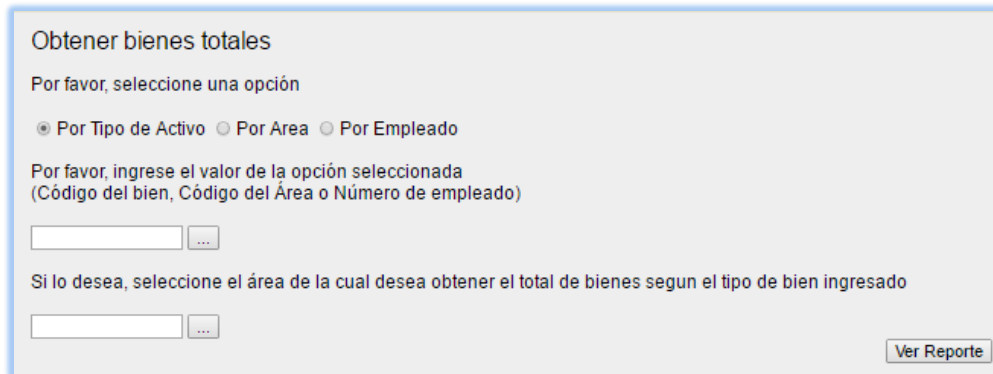
Tipo Activo

No.Activo Inicial No.Activo Final

¿Tiene Automoviles?
 Si
 No

Figura 2.12 Pantalla de reporte por pedido o contrato

Con el reporte de bienes totales es posible obtener información fácilmente simplemente ingresando un solo dato, dependiendo si se desea conocer el total de activos de un mismo tipo, en una misma área o bajo el resguardo de un servidor público.



Obtener bienes totales

Por favor, seleccione una opción

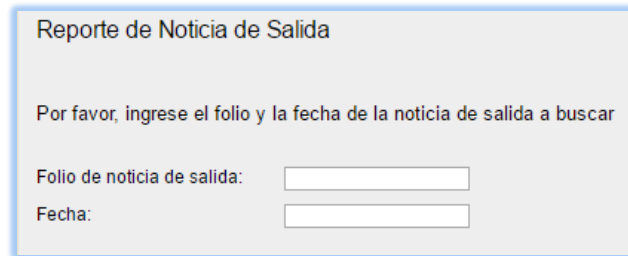
Por Tipo de Activo Por Area Por Empleado

Por favor, ingrese el valor de la opción seleccionada
(Código del bien, Código del Área o Número de empleado)

Si lo desea, seleccione el área de la cual desea obtener el total de bienes segun el tipo de bien ingresado

Figura 2.13 Pantalla de reporte por total de bienes

Las noticias de salida de almacén representan un medio de control sobre los bienes que se entregan a los servidores públicos en las diferentes áreas de este H. Tribunal Superior de Justicia de la Ciudad de México. El apartado de reportes de noticia de salida genera una consulta según el folio y la fecha de manera más ágil.



Reporte de Noticia de Salida

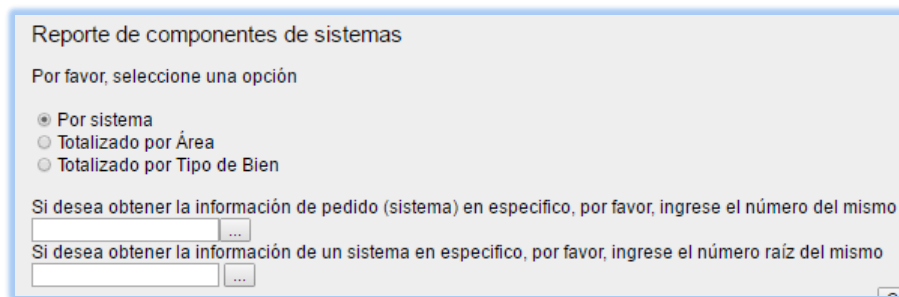
Por favor, ingrese el folio y la fecha de la noticia de salida a buscar

Folio de noticia de salida:

Fecha:

Figura 2.14 Pantalla de reporte de noticia de salida

Al no existir un apartado para consultar componentes, el reporte de sistemas resulta una herramienta muy útil. Es posible solicitar información de un solo sistema, agrupado por área o agrupado por tipo de bien.



Reporte de componentes de sistemas

Por favor, seleccione una opción

Por sistema
 Totalizado por Área
 Totalizado por Tipo de Bien

Si desea obtener la información de pedido (sistema) en específico, por favor, ingrese el número del mismo
 ...

Si desea obtener la información de un sistema en específico, por favor, ingrese el número raíz del mismo
 ...

Figura 2.15 Pantalla de reporte de componentes

También hay una sección encargada de la gestión de usuarios, esta función se encuentra habilitada únicamente para el administrador del sistema ya que desde aquí es posible crear (INSERT INTO) o

borrar (DELETE FROM) cuentas de usuario y modificar los permisos de los existentes, sin embargo también es posible autorizar a otros perfiles realizar estas tareas.

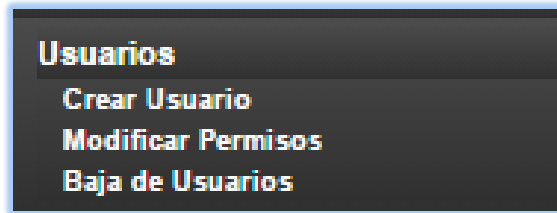


Figura 2.16 Menú de gestión de usuarios

En el área para subir archivos es posible cargar elementos en formato PDF para facturas y JPG para imágenes de activos. Esto es útil para facilitar una pronta referencia en caso de algún error de captura, en el caso de las facturas; o proporcionar una imagen cuando la descripción no resulta clara respecto a las características de un activo.

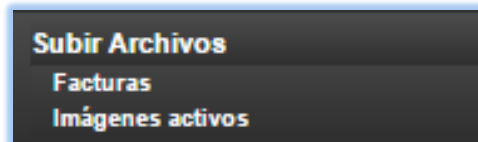


Figura 2.17 Menú para subir archivos.

Para subir una factura es necesario que el archivo este en formato PDF e ingresar el año y el nombre del pedido o contrato.

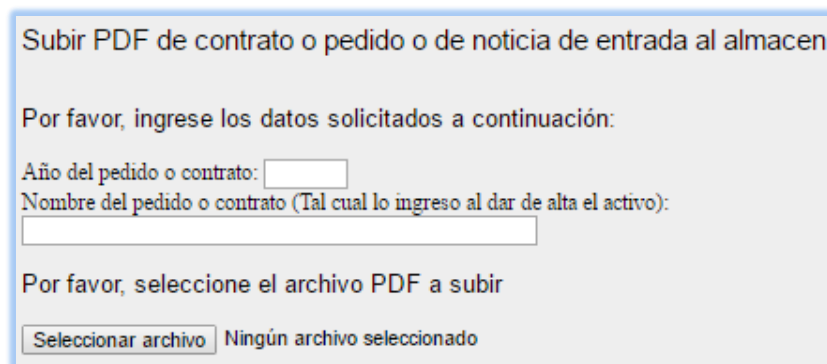
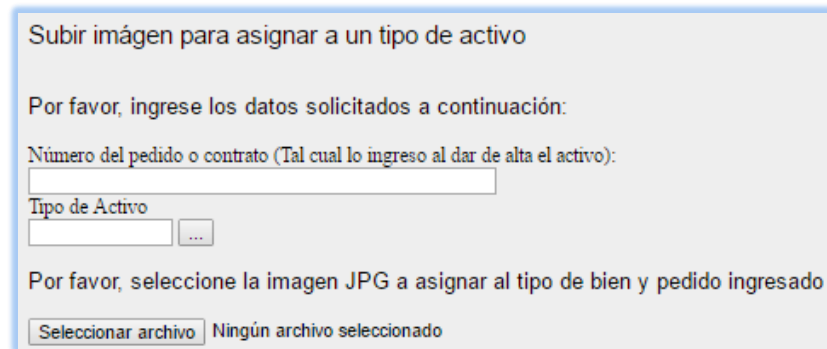
Una pantalla de carga de imágenes de contratos o pedidos. El título es 'Subir PDF de contrato o pedido o de noticia de entrada al almacén'. El texto principal dice 'Por favor, ingrese los datos solicitados a continuación:'. Hay un campo de texto para 'Año del pedido o contrato:' y un campo de texto más largo para 'Nombre del pedido o contrato (Tal cual lo ingreso al dar de alta el activo):'. Debajo, se pide 'Por favor, seleccione el archivo PDF a subir' y hay un botón 'Seleccionar archivo' y el texto 'Ningún archivo seleccionado'.

Figura 2.18 Pantalla de carga de imágenes de contratos o pedidos

Al ingresar la imagen de un activo es necesario proporcionar el nombre del contrato o pedido y el código del bien al cual se asociara la imagen.



Subir imagen para asignar a un tipo de activo

Por favor, ingrese los datos solicitados a continuación:

Número del pedido o contrato (Tal cual lo ingreso al dar de alta el activo):

Tipo de Activo
 ...

Por favor, seleccione la imagen JPG a asignar al tipo de bien y pedido ingresado

Ningún archivo seleccionado

Figura 2.19 Pantalla de carga de imágenes de activos.

En mantenimiento a la base de datos es posible actualizar información (UPDATE) estableciendo diferentes parámetros en un intervalo de activos. Desde aquí es posible aplicar bajas pero no altas en la base de datos, sin embargo resulta una plataforma bastante útil al momento de aplicar cambios masivos ya que permite afectar todos los registros seleccionados en un solo movimiento y no activo por activo como en captura de activos.

Esto representa la sección más compleja del SAF ya que se genera la consulta a la base de datos en función de los campos “DESDE” y “HASTA” que están requisitados. Por otra parte si el campo “ACTUALIZA CON” se encuentra vacío borrará la información actual por lo que si se desea mantener es necesario ingresarla en este campo.

Mantenimiento a la Base de Datos

Actualizar Altas Bajas

Cuando sea:	Desde	Hasta	Actualiza con:	Cuando sea:	Desde	Hasta	Actualiza con:
Sociedad	<input type="text"/>		<input type="text"/>	Costo Adquisición	<input type="text"/>		<input type="text"/>
Partida Presupuestal				Destino Final			
Código del Bien	<input type="text"/> ...	<input type="text"/> ...	<input type="text"/> ...	Mot Baja del Área	<input type="text"/>		<input type="text"/>
Num.Activo				Folio Nota Entrada		<input type="text"/>	
Código Anterior				Folio Nota Salida			
Componentes				Area Ant.	<input type="text"/> ...		<input type="text"/> ...
Detalle				Observaciones Resguardo			
Edificio	<input type="text"/>	<input type="text"/>	<input type="text"/>	Observaciones Generales			
Piso	<input type="text"/>	<input type="text"/>	<input type="text"/>	Motivo de Baja Definitiva	<input type="text"/>		<input type="text"/>
Área	<input type="text"/> ...	<input type="text"/> ...	<input type="text"/> ...	Sesión Comité Autoriza Baja			
Empleado	<input type="text"/> ...	<input type="text"/> ...	<input type="text"/> ...	No Dictamen			
Motivo Alta	<input type="text"/>	<input type="text"/>	<input type="text"/>	Averiguación Previa			
Motivo Alta de Inmueble	<input type="text"/>	<input type="text"/>	<input type="text"/>	Importe de Venta			
Folio de Resguardo				Acta o cont. Venta			
Serie				Vida Util (años)			
Marca				Vida Remanente			
Modelo				Fecha Nota de Entrada	<input type="text"/>	Fechas <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	<input type="text"/>
Num.Pedido				Fecha Noticia de Salida	<input type="text"/>		<input type="text"/>
No Factura				Fecha de Resguardo	<input type="text"/>		<input type="text"/>
Proveedor	<input type="text"/> ...		<input type="text"/> ...	Fecha Factura	<input type="text"/>		<input type="text"/>
No.Motor				Fecha Alta	<input type="text"/>		<input type="text"/>
Placas				Fecha Pedido/Contrato	<input type="text"/>		<input type="text"/>
Status	<input type="text"/> ...		<input type="text"/> ...	Fecha Ultimo Cambio	<input type="text"/>		<input type="text"/>
Estado	<input type="text"/>		<input type="text"/>	Fecha Baja Definitiva	<input type="text"/>		<input type="text"/>
Unidad de Medida				Fecha de Retiro del área	<input type="text"/>		<input type="text"/>
Tiempo de Garantía				Fecha de Reasignación	<input type="text"/>		<input type="text"/>
Valor Catastral				Cambiar estado a			
Estado Físico							

Alta Baja

Aplicar Cambios

Figura 2.20 Pantalla de mantenimiento a la base de datos

CAPÍTULO III

ANÁLISIS

DEFINICIÓN DEL PROBLEMA:

El Sistema de Activo Fijo del Tribunal Superior de Justicia de la Ciudad de México, da la facilidad de administrar la información contenida sin tener que utilizar un manejador o conocer algún lenguaje orientado a bases de datos, lo cual es una gran ventaja considerando que no se cuenta con un DBA que pueda encargarse de esta tarea; y al mismo tiempo restringe a los usuarios para evitar el mal manejo de la información.

Por otra parte, existen desventajas que representan un problema creciente ya que los requerimientos han logrado sobrepasar la capacidad del sistema. También existen deficiencias y errores dentro del propio sistema que fueron identificados y definidos con su uso diario.

La problemática referente a los alcances de SAF se encuentra directamente relacionada con la minería de datos y la carga de información a la base de datos; mientras que las deficiencias y errores refieren problemas en la visualización de algunos campos y conductas inexactas que llega a presentar este sistema.

En el apartado “Mantenimiento a la Base de Datos” de SAF es posible realizar movimientos y cargas masivas de información, siempre y cuando el dato que se va a actualizar sea el mismo para todos los registros o se encuentre dentro de un rango preestablecido; y de esta misma forma, el dato actualizado será el mismo para todos los registros afectados. Por esto, el primer problema al que se enfrenta SAF es la imposibilidad de realizar una carga o corrección masiva de información en campos específicos, y con datos de carga diferentes, por ejemplo: números de serie, folio y fecha de vale de resguardo, usuario resguardante, etc.

Existen 3 columnas para el ingreso de la información de los datos a modificar que son “Desde”, “Hasta” y “Actualiza con”. Las columnas “Desde” y “Actualiza con” se encuentran disponibles para todos los tipos de datos que maneja SAF; mientras que la columna “Hasta” solo se encuentra disponible para cierto tipo de datos en los cuales se puede establecer un rango de valores, por ejemplo el Código del Bien.

Además, en el módulo de creación de reportes personalizados se permite parametrizar la información dentro de un rango de datos como fechas o números de inventario, sin embargo, la sección de cuantificación de fechas no realiza correctamente esta tarea ya que al solicitar un reporte delimitado por ambos lados (desde y hasta), arroja resultados vacíos aun cuando existan registros que cumplan esta condición tal como se muestra en la figura subsecuente:

Fechas	Desde	Hasta
42.Fecha Nota de Entrada	<input type="checkbox"/>	<input type="text"/>
43.Fecha Noticia de Salida	<input type="checkbox"/>	<input type="text"/>
44.Fecha de Resguardo	<input checked="" type="checkbox"/>	<input type="text" value="01/01/2014"/>
45.Fec.Factura	<input type="checkbox"/>	<input type="text"/>
46.Fecha Contrato/Pedido	<input type="checkbox"/>	<input type="text"/>
47.Fec.Ult.Cambio	<input type="checkbox"/>	<input type="text"/>
48.Fec.de.Captura	<input type="checkbox"/>	<input type="text"/>
49.Fec.Baja	<input type="checkbox"/>	<input type="text"/>
50.Fec.Reasignación	<input type="checkbox"/>	<input type="text"/>
51.Fecha de Retiro	<input type="checkbox"/>	<input type="text"/>

Figura 3.1 Reporte con parámetros en “Fecha de resguardo”

Asimismo existen datos como la garantía o la vida remanente, que sufren cambios a través del tiempo y es necesario tener acceso a la información actualizada, sin embargo SAF no realiza el cálculo necesario para poder mostrar el dato actualizado al momento de la consulta. Esto se traduce en un mayor tiempo de respuesta a la solicitud de reportes que incluyan esta información.

	Desde	Hasta		Desde	Hasta	
01.Sociedad	<input type="checkbox"/>	<input type="text"/>		31.Num.Pedido	<input type="checkbox"/>	<input type="text"/>
02.Código del Bien	<input checked="" type="checkbox"/>	5111000272	...	<input type="text"/>	<input type="text"/>	
03.Num.Activo	<input checked="" type="checkbox"/>	023000	023010	32.No.Motor	<input type="checkbox"/>	<input type="text"/>
04.Area	<input type="checkbox"/>	<input type="text"/>	...	33.Proveedor	<input type="checkbox"/>	<input type="text"/>
05.Empleado	<input type="checkbox"/>	<input type="text"/>	...	34.Código Anterior	<input type="checkbox"/>	<input type="text"/>
06.Motivo Alta	<input type="checkbox"/>	<input type="text"/>	▼	35.Edificio	<input type="checkbox"/>	<input type="text"/>
07.Motivo Alta de Inmueble	<input type="checkbox"/>	<input type="text"/>	▼	36.Piso	<input type="checkbox"/>	<input type="text"/>
08.Folio de Resguardo	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	37.Destino Final	<input type="checkbox"/>	<input type="text"/>
09.Serie	<input type="checkbox"/>	<input type="text"/>		38.Usuario que Capturo	<input type="checkbox"/>	<input type="text"/>
10.Marca	<input type="checkbox"/>	<input type="text"/>		39.Usuario que realizo Ultimo Cambio	<input type="checkbox"/>	<input type="text"/>
11.Modelo	<input type="checkbox"/>	<input type="text"/>		40.Importe de Venta	<input type="checkbox"/>	<input type="text"/>
12.No.Factura	<input type="checkbox"/>	<input type="text"/>		41.Motivo Baja Definitiva	<input type="checkbox"/>	<input type="text"/>
13.Placas	<input type="checkbox"/>	<input type="text"/>				
14.Costo Adq	<input type="checkbox"/>	<input type="text"/>		Fechas	Desde	
15.Descripción	<input type="checkbox"/>	<input type="text"/>		42.Fecha Nota de Entrada	<input type="checkbox"/>	<input type="text"/>
16.Status(Lugar Físico)	<input type="checkbox"/>	<input type="text"/>	...	43.Fecha Noticia de Salida	<input type="checkbox"/>	<input type="text"/>
17.Estado	<input type="checkbox"/>	<input type="text"/>	▼	44.Fecha de Resguardo	<input type="checkbox"/>	<input type="text"/>
18.Partida Presupuestal	<input type="checkbox"/>	<input type="text"/>		45.Fec.Factura	<input checked="" type="checkbox"/>	<input type="text"/>
19.Tiempo de Garantía	<input checked="" type="checkbox"/>	<input type="text"/>		46.Fecha Contrato Pedido	<input type="checkbox"/>	<input type="text"/>
20.Valor Catastral	<input type="checkbox"/>	<input type="text"/>		47.Fec.Ult.Cambio	<input type="checkbox"/>	<input type="text"/>
21.Folio Noticia Entrada	<input type="checkbox"/>	<input type="text"/>		48.Fec.de.Captura	<input type="checkbox"/>	<input type="text"/>
22.Folio Noticia Salida	<input type="checkbox"/>	<input type="text"/>		49.Fec.Baja	<input type="checkbox"/>	<input type="text"/>
23.Mot.Baja del Área	<input type="checkbox"/>	<input type="text"/>	▼	50.Fec.Reasignación	<input type="checkbox"/>	<input type="text"/>
24.Area Ant.	<input type="checkbox"/>	<input type="text"/>	...	51.Fecha de Retiro	<input type="checkbox"/>	<input type="text"/>
25.No.Dictamen	<input type="checkbox"/>	<input type="text"/>				
26.Detalle	<input type="checkbox"/>	<input type="text"/>		Datos Adicionales		
27.Unidad de medida	<input type="checkbox"/>	<input type="text"/>		52.Observaciones Resguardo	<input type="checkbox"/>	<input type="text"/>
28.Estado físico	<input type="checkbox"/>	<input type="text"/>		53.Observaciones Baja	<input type="checkbox"/>	<input type="text"/>
29.Acta o CTTO. Venta	<input type="checkbox"/>	<input type="text"/>		54.Sesión del Comite Que Se Autoriza La Baja	<input type="checkbox"/>	<input type="text"/>
30.Averiguacion Previa	<input type="checkbox"/>	<input type="text"/>		55.Componentes de un Sistema	<input type="checkbox"/>	<input type="text"/>
				56.Vida Util (años)	<input type="checkbox"/>	<input type="text"/>
				57.Vida Remanente	<input checked="" type="checkbox"/>	<input type="text"/>

Figura 3.2 Solicitud de Fecha de factura / Garantía / Vida Remanente de 10 activos.

	A	B	C	D	E	F
1	Tribunal Superior de Justicia del D.F					
2						
3	CODIGO DEL BIEN	DESCRIPCION DEL BIEN	NUM ACTIVO	FECHA DE FACTURA	TIEMPO DE GARANTIA	VIDA REMANENTE
4	5111000272	SILLA GIRATORIA	023000	27/06/2008	10 AÑOS	
5	5111000272	SILLA GIRATORIA	023001	27/06/2008	10 AÑOS	7
6	5111000272	SILLA GIRATORIA	023002	27/06/2008	10 AÑOS	7
7	5111000272	SILLA GIRATORIA	023003	27/06/2008	10 AÑOS	6
8	5111000272	SILLA GIRATORIA	023004	27/06/2008	10 AÑOS	
9	5111000272	SILLA GIRATORIA	023005	27/06/2008	10 AÑOS	
10	5111000272	SILLA GIRATORIA	023007	27/06/2008	10 AÑOS	7
11	5111000272	SILLA GIRATORIA	023008	27/06/2008	10 AÑOS	7
12	5111000272	SILLA GIRATORIA	023009	27/06/2008	10 AÑOS	
13	5111000272	SILLA GIRATORIA	023010	27/06/2008	10 AÑOS	

Figura 3.3 Respuesta a solicitud de Fecha de factura / Garantía / Vida Remanente de 10 activos

Por otra parte, existen errores que no fueron atendidos en el proceso de desarrollo y puesta en marcha de SAF, lo que causa problemas al momento de tratar de visualizar algunos activos desde el apartado "Consulta" ya que muestra todos los campos como vacíos; cabe mencionar que estos mismos activos se visualizan correctamente desde la plataforma de captura o accediendo directamente a la base de datos pues se muestran los datos correspondientes, sin embargo, existen usuarios que tienen acceso al sistema únicamente para consulta de información y por supuesto que tampoco tienen autorización de acceso directamente a la base de datos.

Figura 3.4 Resultado de la sección de consulta al solicitar información de un activo en el cual se presenta el error.

Otro error de visualización se da en este mismo apartado cuando se consulta la información de un activo que posee un valor diferente de vacío en los campos “Estado”, “Proveedor”, “Factura”, “Lugar Físico”, “Descripción de edificio” o “Descripción de piso”; y posteriormente se consulta otro activo que si tiene vacío o vacíos estos campos. La sección en la que se imprime el respectivo dato no es limpiada antes de realizar la siguiente solicitud y se vuelve a mostrar en pantalla durante la nueva consulta, resultando información errónea. Se puede observar que este fallo se presenta en todos los campos que se imprimen fuera de un cuadro de texto.

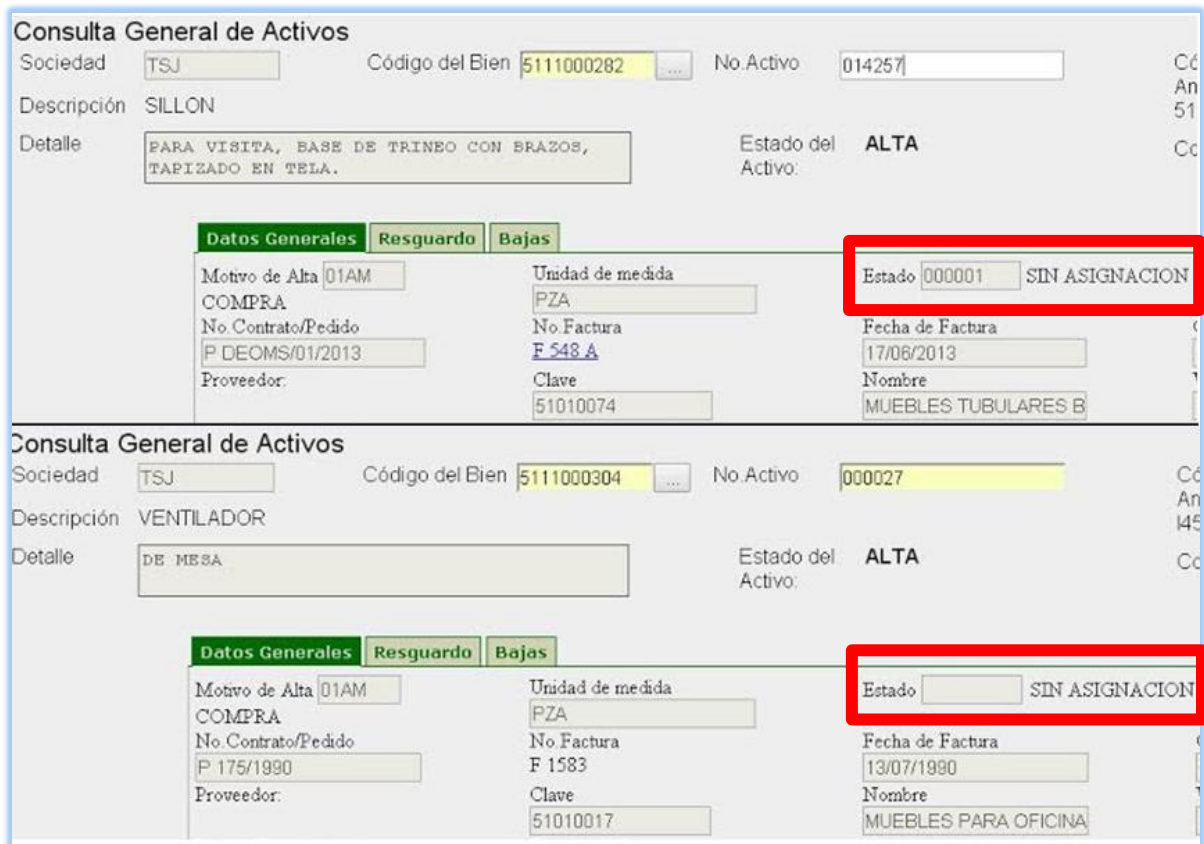


Figura 3.5 Pantalla antes y después, la leyenda del campo no se limpia

REQUERIMIENTOS:

- Incluir los componentes de sistemas en los reportes de activo fijo.
- Carga y corrección masiva de información en campos específicos (serie, folio de resguardo,

- fecha de resguardo, usuario, etc.).
- Corrección de errores de origen en visualización, consultas y guardado de información a través de SAF.
- Creación de un apartado que permita la consulta de componentes únicamente para visualización de la información.

OBJETIVO GENERAL:

El objetivo principal radica en optimizar la funcionalidad de SAF mediante la modificación de módulos específicos en el código fuente y con esto lograr mantener la eficiencia y eficacia del sistema.

OBJETIVOS ESPECÍFICOS:

- Programar los script que faciliten la tarea de realizar una carga o corrección masiva de información, la cual se albergara en la base de datos.
- Incluir los script previamente programados al código fuente de SAF.
- Generar consultas en MySQL que satisfagan las carencias en los reportes de activo fijo para generar una primera solución y posteriormente integrarlas al código fuente.
- Llevar a cabo una auditoría al código fuente de SAF a fin de identificar las ubicaciones exactas en donde habrá que implementar las consultas generadas en MySQL.
- Identificar los módulos del código correspondientes que provocan los errores de la visualización incorrecta de los campos, consulta de datos erróneos y problemas con el guardado de cambios.

JUSTIFICACIÓN DEL PROYECTO:

El trabajo diario de SAF en el área de inventarios debe arrojar un resultado con información actualizada, veraz y concisa, por esta razón se tomó la decisión de desarrollar este proyecto. Además brindar a otros usuarios la posibilidad de manejar la base de datos de una mejor manera aún sin tener los conocimientos informáticos para este fin, supone en sí mismo un avance significativo en la evolución de la institución.

Es necesario sustentar prioridad a los nuevos requerimientos pues satisfacer estas necesidades jugaría un papel importante al momento de evaluar las mejoras sobre el sistema; esto representa la manera más enérgica de abordar el objetivo general.

CAPÍTULO IV

DESARROLLO

CORRECCIÓN DE ERRORES:

PARÁMETROS DE FECHA: Las delimitaciones dentro de las fechas en un reporte funcionan mediante una decisión, que sólo en caso de incluir el checkbox de tal parámetro lo agregará a la variable *arrWhere* que es la condición utilizada en la consulta de MySQL. Sin embargo, debe verificar si ambos limites (“desde” y “hasta”) contienen alguna información, para esto se ejecuta el siguiente proceso:

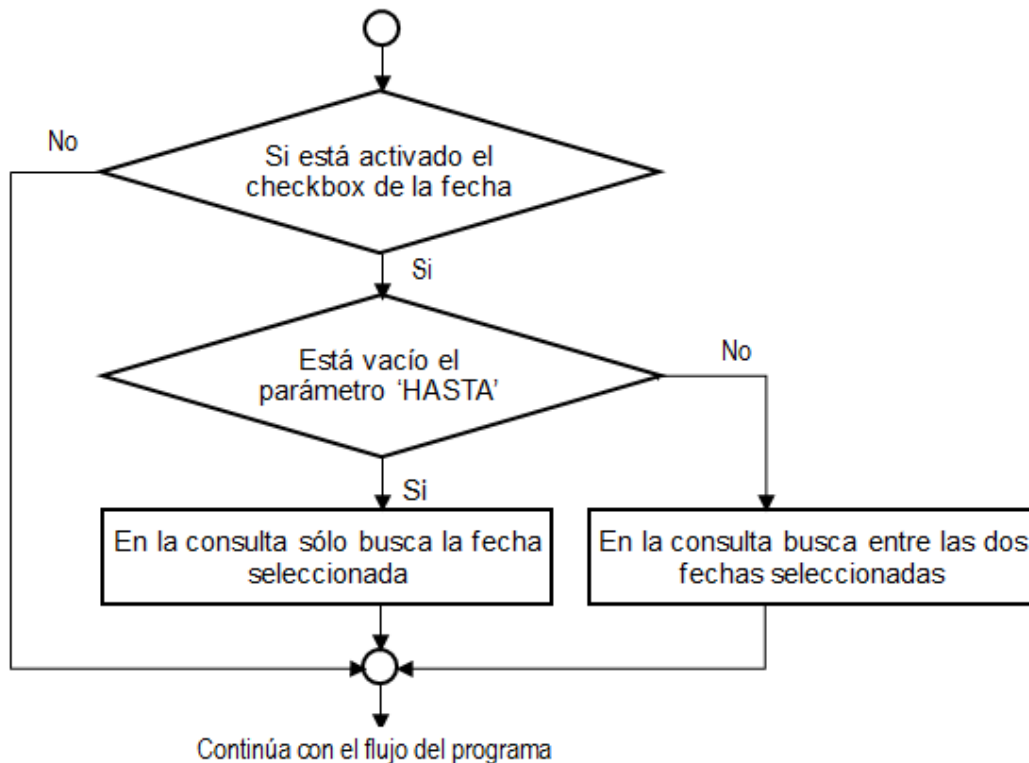


Figura 4.1 Diagrama de flujo: delimitación en campos “DESDE” y “HASTA” en fecha

No obstante, al incluir los parámetros “desde” y “hasta” en la fecha de nota de entrada no eran respetados. El procedimiento es el que se muestra debajo:

Fechas	Desde	Hasta
42.Fecha Nota de Entrada	<input checked="" type="checkbox"/>	01/01/2014
		01/01/2015

Figura 4.2 Reporte con parámetros en “Fecha nota de entrada”


```
$aarrWhere[] = "and ne.fecha_not_ent_almacen
between STR_TO_DATE({$DATA["fechanotaentradaDesde"]}, '%d/%m/%Y')
and STR_TO_DATE({$DATA["fechanotaentradaHasta"]}, '%d/%m/%Y')";
```

Con este código, el resultado obtenido es el siguiente:

CODIGO DEL BIEN	DESCRIPCION DEL BIEN	NUM ACTIVO	FECHA NOTA DE ENTRADA
511100004	ANAQUEL MOVIL	014700	
511100012	ARCHIVERO DE METAL	019789	
511100012	ARCHIVERO DE METAL	019790	
511100012	ARCHIVERO DE METAL	019791	
511100012	ARCHIVERO DE METAL	019796	

Figura 4.3 Resultado de reporte con parámetros en “Fecha nota de entrada”

Esto es debido a que el código original contenía un error en la asignación de la variable *\$aarrWhere*, estas condiciones se asignaban erróneamente a la variable *aarrWhere* la cual es inexistente. Asimismo dentro de la función *STR_TO_DATE* no fueron incluidas las comillas en la cadena que se pretende convertir a fecha (Gerardo A. Otero Rodríguez, 2010). Las correcciones aplicadas al código se muestran a continuación:

```
$aarrWhere[] = "and ne.fecha_not_ent_almacen
between STR_TO_DATE('{ $DATA["fechanotaentradaDesde"] }', '%d/%m/%Y')
and STR_TO_DATE('{ $DATA["fechanotaentradaHasta"] }', '%d/%m/%Y')";
```

De igual manera, si solamente se utiliza el campo “desde” tampoco es respetada la condición arrojando diferentes irregularidades tal como se muestra:

Fechas	Desde	Hasta
42.Fecha Nota de Entrada	<input checked="" type="checkbox"/> 08/01/2014	<input type="text"/>

Figura 4.4 Reporte con parámetro “desde” en “Fecha nota de entrada”

```
$aarrWhere[] = "and ne.fecha_not_ent_almacen=
STR_TO_DATE({$DATA["fechanotaentradaDesde"]}, '%d/%m/%Y')";
```

	A
1	Tribunal Superior de Justicia del D.F.
2	
3	No Existen Datos.

Figura 4.5 Resultado de reporte con parámetro “desde” en “Fecha nota de entrada”

Esto sucede ya que contiene el mismo error de no incluir las comillas simples en la función `STR_TO_DATE`, lo que se corrige de la siguiente forma:

```
$arrWhere[] = "and ne.fecha_not_ent_almacen=
STR_TO_DATE('{ $DATA["fechanotaentradaDesde"]}', '%d/%m/%Y')";
```

Del mismo modo, al incluir los parámetros “desde” y “hasta” en la Fecha de vale de resguardo se presenta el mismo error, por lo que se puede deducir que no se han incluido las comillas simples al inicio y al final de la cadena de texto que va a ser convertida a fecha. El código se observa a continuación:

```
$arrWhere[] = "and r.fecha
between STR_TO_DATE({ $DATA["fecharesguardoDesde"]}, '%d/%m/%Y')
and STR_TO_DATE('{ $DATA["fecharesguardoHasta"]}', '%d/%m/%Y')";
```

Por lo que la información obtenida es errónea tal como se muestra debajo:

	A
1	Tribunal Superior de Justicia del D.F
2	
3	No Existen Datos.
4	

Figura 4.6 Resultado incorrecto del reporte solicitado.

Por lo tanto, la corrección aplicada a esta parte del código es la misma que en los casos anteriores; simplemente se agregan las comillas faltantes en el lugar correspondiente como se puede ver en la siguiente imagen:

```
$arrWhere[] = "and r.fecha
between STR_TO_DATE('{ $DATA["fecharesguardoDesde"]}', '%d/%m/%Y')
and STR_TO_DATE('{ $DATA["fecharesguardoHasta"]}', '%d/%m/%Y')";
```

VIDA REMANENTE: La vida remanente de un bien es el tiempo de vida útil restante después de su adquisición, éste dato se calcula deduciendo el tiempo transcurrido desde la fecha de compra al tiempo total de vida útil. Originalmente éste cálculo no se realiza y únicamente se guarda el valor de la vida remanente cada vez que es actualizado por un usuario, por lo que no proporciona información real. El procedimiento que efectúa la consulta de este dato es el siguiente:

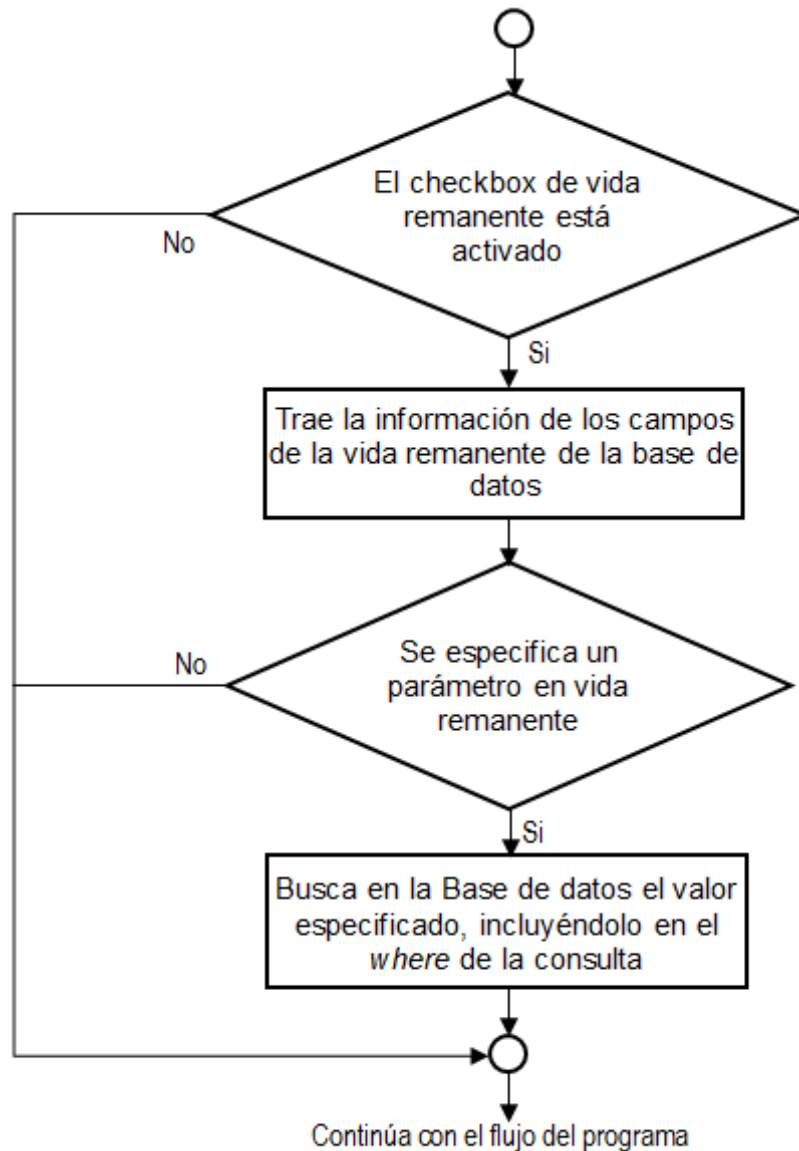


Figura 4.7 Diagrama de flujo: Lectura del valor "VIDA REMANENTE"

La solución a este problema únicamente representa realizar la resta correspondiente antes de mostrar el resultado en el campo designado para la vida remanente y con esto es posible obtener el dato real. Después de tomar en cuenta las diferentes variables localizadas, el flujo y procesamiento que sigue la información es el que se muestra en el diagrama siguiente:

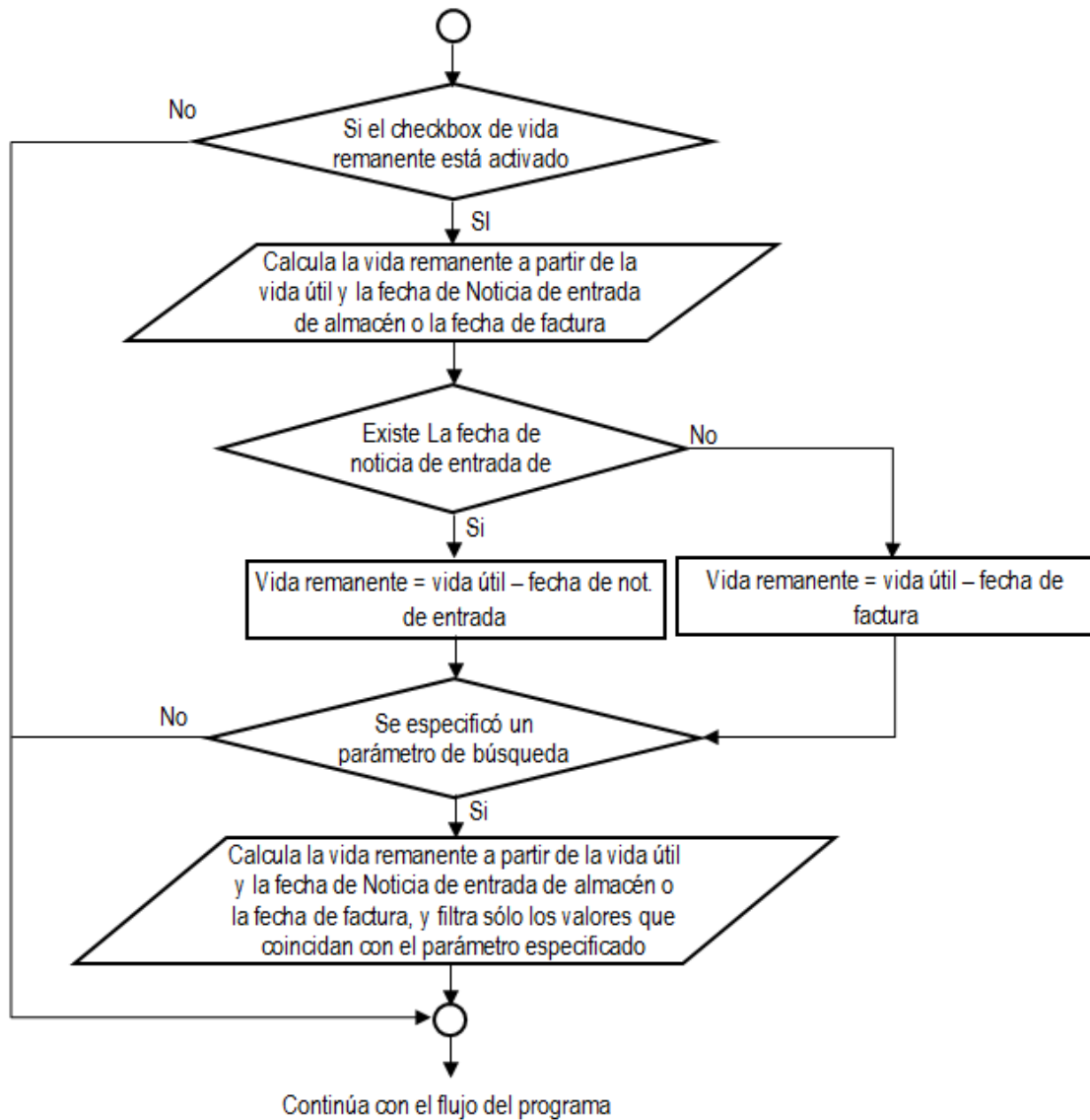


Figura 4.8 Diagrama de flujo: Cálculo del valor “VIDA REMANENTE” (corrección).

CAPTURA GENERAL: Los campos “detalle”, “observaciones de resguardo” u “observaciones de bajas” dentro de la captura general, pueden llegar a contener el símbolo de comilla simple, el cual

provoca un error al momento de generar la consulta debido a que este símbolo representa el cierre de un campo.

En este proceso se realiza una iteración sobre los campos extraídos de la base de datos y en caso de diferir con los campos del formulario, agrega la nueva información al arreglo para posteriormente separarlo y generar la consulta de actualización. Esto es posible gracias al siguiente fragmento de código:

```

$datoActivo = $activo -> getData($activo -> getActivoIdactivo());
foreach ($datoActivo[0] as $key => $value) {
    if (isset($DATA[$key]) && ($DATA[$key] != $value)) {
        $datoActivo[0][$key] = $DATA[$key];
    }
}
    
```

Durante éste procedimiento, el flujo de la información es el siguiente:

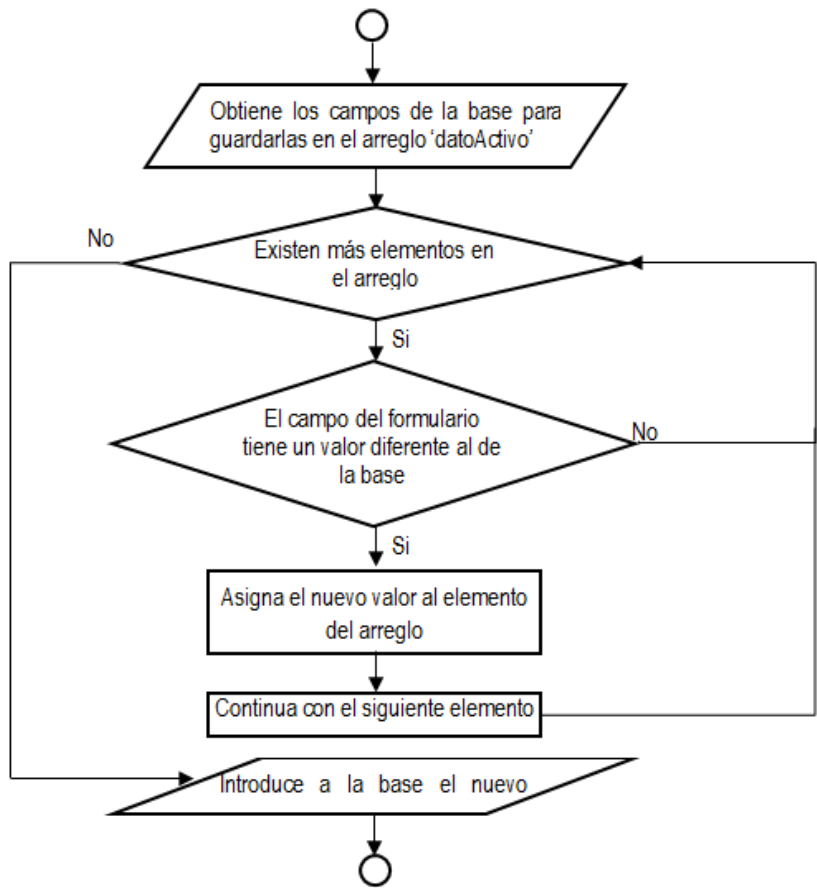


Figura 4.9 Diagrama de flujo: Campos extraídos durante la consulta

Para corregir esta falla fue necesario modificar las cadenas que provienen del formulario utilizando la función `str_replace` (Hernan Beati, 2011), cambiando los símbolos de comilla simple (`'`) por `diagonal_invertida+comilla_simple` (`'`) ya que de esta manera es como MySQL reconoce una comilla simple dentro del texto, por esta razón el código fuente finalmente queda de esta manera:

```

$datoActivo = $activo -> getData($activo -> getActivoIdactivo());
foreach ($datoActivo[0] as $key => $value) {
    if (isset($DATA[$key]) && ($DATA[$key] != $value)) {
        $datoActivo[0][$key] = str_replace("'", "\'", $DATA[$key]);
    }
}
    
```

Una vez aplicada esta corrección, la nueva trayectoria de la información es como se muestra en el diagrama de abajo:

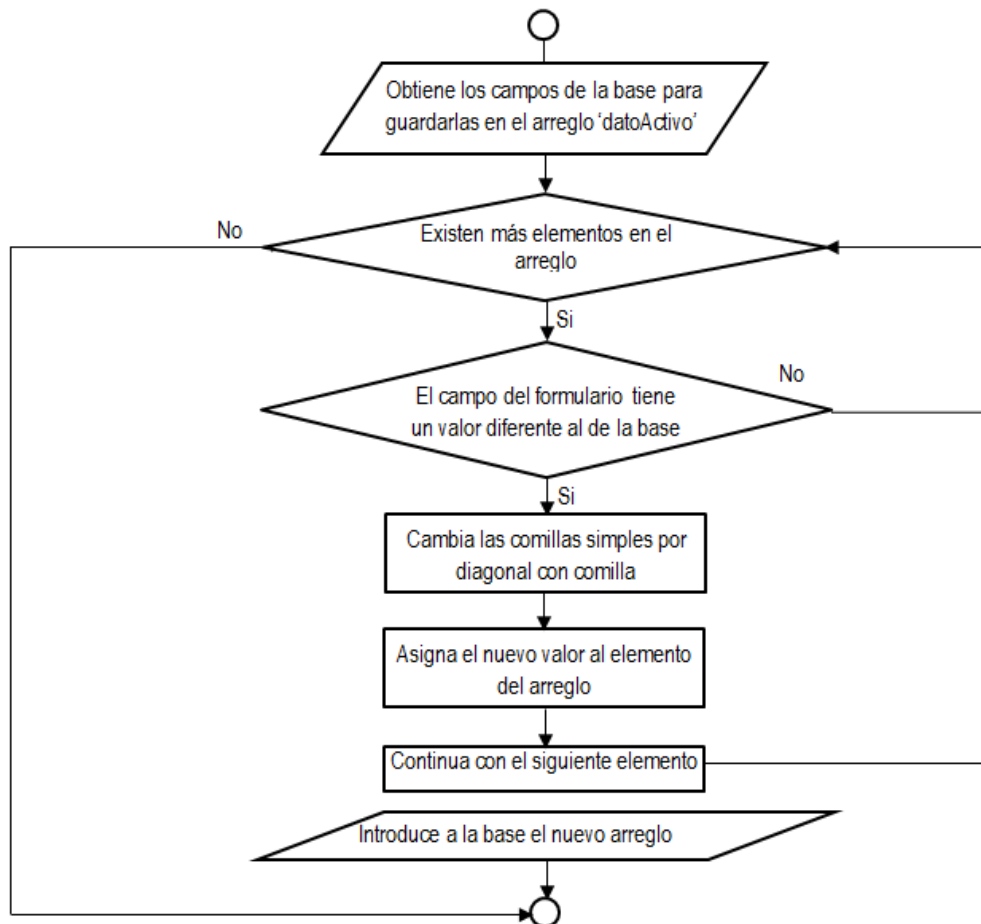


Figura 4.10 Diagrama de flujo: Campos extraídos durante la consulta reconociendo comilla simple (corrección).

NUEVAS IMPLEMENTACIONES:

NULL Y NOT NULL: Al momento de generar un reporte personalizado, no era posible solicitar registros en donde los campos tuvieran un valor *null* o *not null*. Esta función se incluyó como una de las primeras nuevas actualizaciones realizadas al sistema ya que únicamente buscaba los parámetros especificados de la siguiente forma:

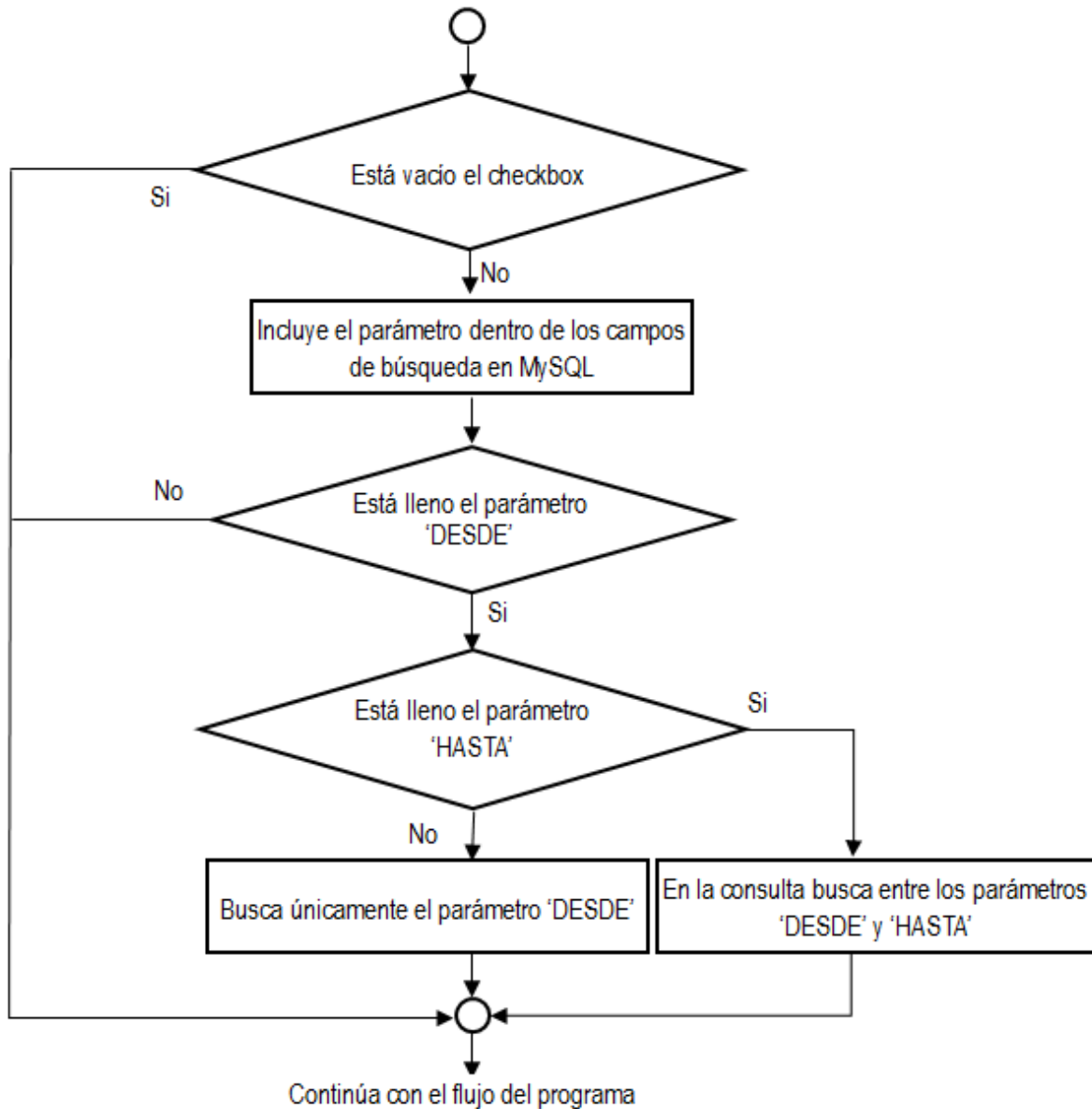


Figura 4.11 Diagrama de flujo: Parametrización en los campos “DESDE” y “HASTA”.

Posterior a la actualización, el flujo de información se modifica quedando como se muestra en la figura a continuación:

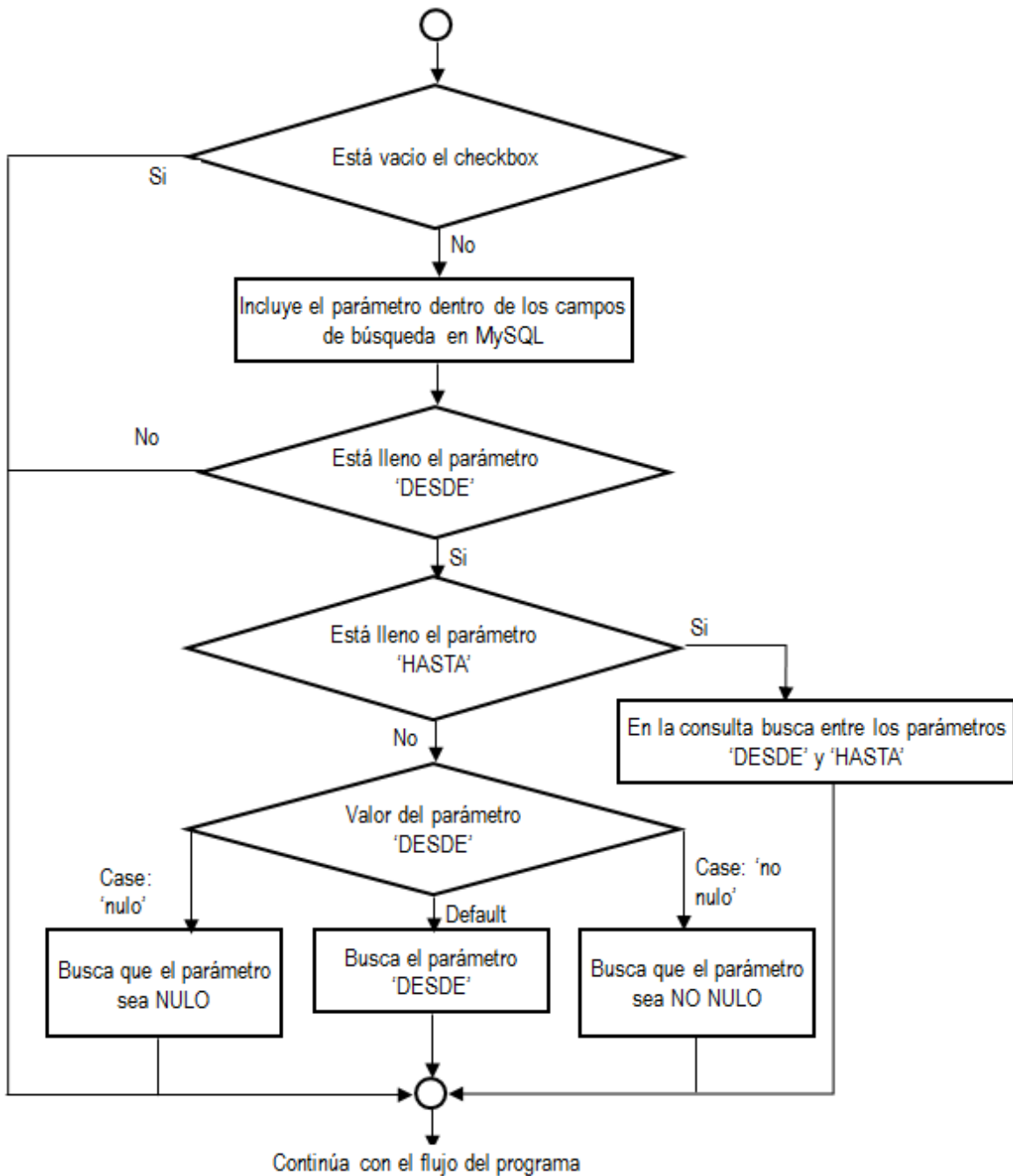


Figura 4.12 Diagrama de flujo: reconocimiento de valores NULL y NOT NULL en campos “DESDE” y “HASTA” (actualización)

Esta implementación puede ser utilizada en los campos “Tipo de bien-desde”, “Número activo-desde”, “Área-desde”, “Empleado”, “Folio de resguardo-desde”, “Serie”, “Folio Noticia de entrada” y “Folio Noticia de salida”; la forma de utilizarla al momento de generar un reporte es como se muestra en las imágenes siguientes:

Figura 4.13 Uso de los parámetros “nulo” y “no nulo”

CONSULTA DE COMPONENTES: En el desarrollo original únicamente se creó un módulo de consulta para activos y la única manera de visualizar información referente a los componentes era mediante el apartado de captura. Por esta razón fue necesario implementar una sección que permita la consulta de información sin admitir modificaciones a su contenido.

Para crear la unidad encargada de la consulta de componentes fue necesario modificar la base de datos agregando un permiso en la tabla correspondiente, el cual se muestra a continuación:

idpermiso	descripcion
27	Reporte de Activos Totales
28	Reporte de Activos Dados de Baja
29	Reporte de Activos Totales
30	Captura Masiva de Noticia de Salida
31	Reporte de Noticia de Salida
32	Subida Archivos PDF
33	Subida Imagenes
34	Reporte de Sistemas
35	Consulta Sistemas

Figura 4.14 Nuevo permiso agregado a la base de datos.

Consecutivamente se hizo la modificación al archivo llamado “vista.menuPrincipal.php”, el cual contiene el código encargado de generar el menú que visualizan los usuarios; agregando la verificación del nuevo permiso de acuerdo al siguiente código:

```
<?php
    $bandera2=0;
    for($c=0;$c<sizeof($permisos);$c++){//Busca que el permiso 35
    esté activado
        if($permisos[$c]['idpermiso']==35 ){
            $bandera2=1; //En caso de estar activo el permiso, marca la
            variable bandera2 con 1
            break;
        }
    }
    if($bandera2){ //Si la variable bandera2 está marcada, pone el
    acceso a la consulta de componentes.
?>
    <li>
        <a href="#" onclick="cargarContenido('contenedor', 'controlador/controlador.consultaSistemas.php', '');" > Componentes</a>
    </li>
<?php
}
```

Una vez incorporado esto al código fuente, la representación gráfica de la revisión de permisos se lleva a cabo de la siguiente manera:

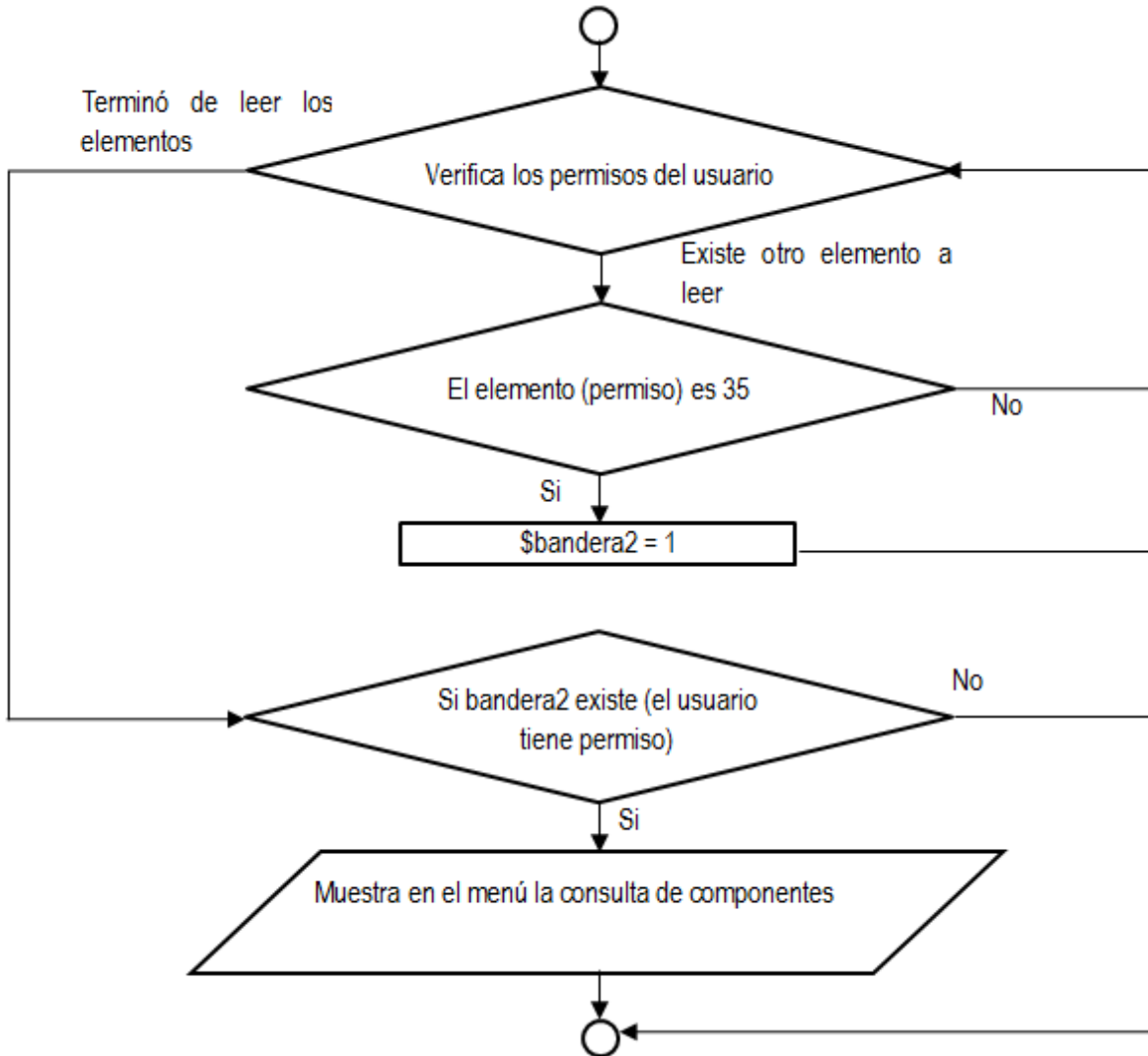


Figura 4.15 Diagrama de flujo: Verificación de permiso para consulta de componentes (actualización).

La estructura del módulo de consulta de componentes fue creada tomando como base la sección de captura. El nuevo módulo se adapta al patrón original MVC (Modelo-Vista-Controlador) al igual que todos los demás.

En el Modelo, la consulta a la base de datos se conserva exactamente igual, sin embargo se eliminan las funciones encargadas de insertar nuevos registros y de guardar las modificaciones. En el controlador también se eliminaron estas funciones.

Para la Vista, se bloquearon todos los campos de texto excepto los de número de sistema (o número de inventario) y sub-número ya que es aquí en donde se elige el componente que se desea visualizar; además fueron eliminados los botones de guardado.

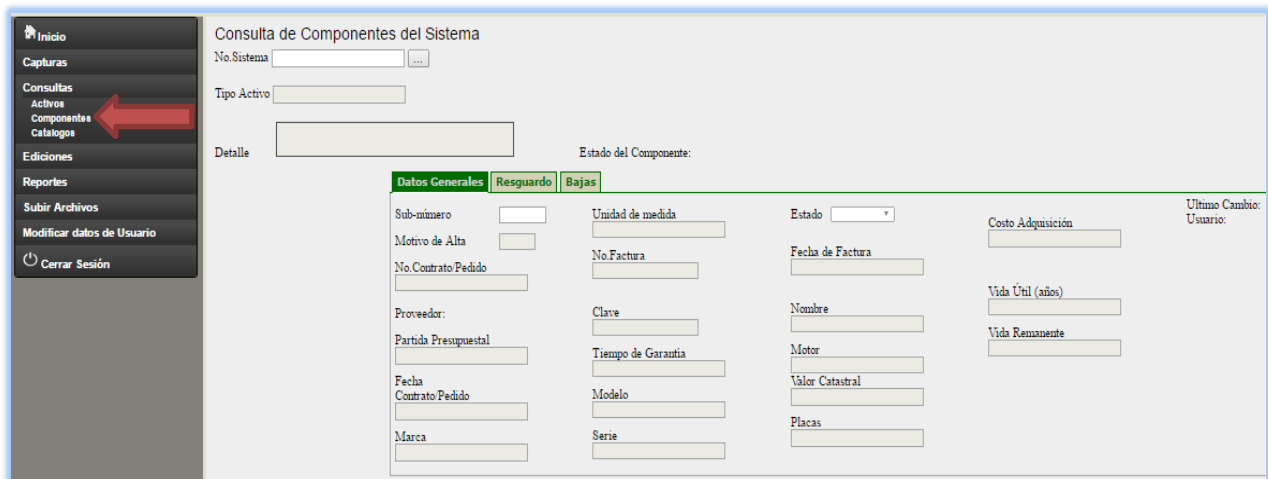


Figura 4.16 Nueva sección agregada al menú. (Consulta de componentes).

CARGAS MASIVAS: Una de las principales actividades dentro del área de inventarios es la alimentación de información al Sistema de Activo Fijo mediante la sección de Captura de Activos. Esta captura se realiza regularmente mediante una secuencia de pasos:

1.- Pestaña de datos generales:

- a) Introducir el número de inventario del activo a actualizar
- b) Se revisa el estado el bien para conocer si es posible asignarlo a un área o si se encuentra en un proceso de baja; se cambia el estado a vacío, ya que este bien se encuentra activo.
- c) En caso de existir una serie, Se revisa la serie con la del documento de resguardo, para corroborar que se trata del mismo bien.

2.- Pestaña de Resguardo:

a) Se revisa que la fecha del documento de resguardo sea más actualizada que la existente en el sistema (la nueva información debe ser más reciente), de lo contrario, no debe ser actualizado.

b) Se captura el folio de resguardo, la fecha de resguardo, el número de empleado, el lugar físico donde se encuentra el bien (en caso de existir) o eliminarlo en caso de no existir, el área, el edificio y el piso (en caso de existir), Así como las observaciones necesarias (las cuales muchas veces deben ser eliminadas, sustituidas o agregadas a las existentes en el sistema).

Estos pasos se realizaban bien a bien cada vez que se necesitaba capturar, y aunque diversos bienes pertenecieran a un mismo usuario, área y ubicación, los datos debían repetirse y capturarse uno a uno. Por lo anterior, este procedimiento fue automatizado en un proceso que funciona para las cargas más comunes, esto se realiza únicamente capturando la información en un documento delimitado por comas generado en Excel, el cual debe contener las siguientes características: No debe existir encabezado, todas las columnas deben estar en formato de texto y deben estar en el siguiente orden:

Inventario | folio | serie | empleado | inmueble | piso | área | lugar | fecha | observaciones.

Para no alterar algún campo simplemente la celda debe estar vacía y si un dato necesita ser eliminado, se introduce la palabra “BORRAR”. En el caso de las observaciones, si se desean sustituir se escribe la palabra “BORRAR” antes de la nueva información a actualizar, de lo contrario la información a insertar se concatenará a la información ya existente; y si se desea borrar la información del campo, se escribe la palabra “BORRAR”.

Para tal proceso fue necesario crear los permisos en la base de datos, que permitirían realizar estas acciones:

idpermiso	descripcion
35	Consulta Sistemas
36	Carga Masiva de Vales

Figura 4.17 Tabla de permisos.

También fue necesario agregar estos mismos permisos a la pantalla del administrador del sistema para que pueda modificarlos en cualquier momento y para cualquier usuario.

Figura 4.18 Administración de permisos de usuarios.

Para lo cual fue necesario modificar el archivo de 'vista.permisosUsuarios.php' con el siguiente código:

```
<?php
$bandera2=0;
for ($c=0;$c<sizeof($permisos);$c++){//Busca que el permiso 35 esté
    activado
    if($permisos[$c]['idpermiso']==35){
    $bandera2=1; //En caso de estar activo el permiso, marca la variab
    le bandera2 con 1
    <h3><strong>Asignar permisos:</strong></h3><br />
    <div class="subtabla">
        <h3>Captura Activos</h3><br />
        <span class="chk"><input type="checkbox" name="permisos[]" value="1"
    /></span>
        <span class="sticker"><label>Datos Generales</label></span><br />
        <span class="chk"><input type="checkbox" name="permisos[]" value="2"
    /></span>
        <span class="sticker"><label>Datos de Resguardo</label></span><br />
        <span class="chk"><input type="checkbox" name="permisos[]" value="3"
    /></span>
        <span class="sticker"><label>Datos de Baja</label></span><br />
        <span class="chk"><input type="checkbox" name="permisos[]"
    value="36" /></span>
        <span class="sticker"><label>Carga Masiva de Vales</label></span>
    </div>
```

Así como el archivo 'modelo.permisosUsuarios.php' con el siguiente código dentro de la función 'muestraPermisos (\$idusuario)':

```

$bandera=0;
for ($i=0;$i<sizeof($permisos);$i++){
    if ($permisos[$i]['idpermiso']==36){
        $bandera=1;
        break;
    }
}
if ($bandera)
    echo "<span class='chk'><input type='checkbox' name='permisos[]'
value='36' checked /></span>";
else
    echo "<span class='chk'><input type='checkbox' name='permisos[]'
value='36' /></span>";
echo "<span class='sticker'><label>Carga Masiva de vales </label>
</span>";

echo "</div>";

```

La sección para cargar masivamente los vales mediante un archivo CSV se encuentra dentro del menú de Capturas, como se muestra en la siguiente figura:

Subir archivo para actualización masiva de Resguardos

Por favor, el archivo a subir debe contener el formato especificado:

Archivo CSV separado por comas, con las siguientes características:
 Inventario | folio | serie | empleado | inmueble | piso | area | lugar | fecha | observaciones

El archivo NO debe tener encabezados, solamente la información organizada de dicha forma.
 Cada columna debe estar en formato de texto
 Si un dato no se necesita modificar, se deja en blanco
 Si un dato requiere ser borrado, se escribe la palabra BORRAR.
 Si se desea eliminar las observaciones anteriores, se escribe BORRAR, seguida de la nueva observación
 de lo contrario sólo se agregara la nueva observación a la ya existente

Por favor, seleccione el archivo CSV para la carga masiva

Ningún archivo seleccionado

Figura 4.19 Sección de carga masiva mediante un archivo CSV delimitado por comas.

Esta nueva sección es generada por los siguientes archivos:

- Modelo.subParaCargaMasiva.php: Aquí se contiene el proceso para almacenar en el servidor el archivo CSV, de tal forma que no sea sobrescrito en caso de que varios usuarios realicen el proceso al mismo tiempo, renombrando el archivo mediante el id del usuario y una marca de tiempo.
- Vista.capturaMasivaResg.php: Contiene la pantalla gráfica presentada al usuario así como el código en JavaScript para enviar el archivo CSV a los siguientes procesos que continúan la verificación de la información del archivo.
- Controlador.checarCargaMasiva.php: Es el encargado de comparar la información del archivo CSV con la información contenida en la base de datos, presentándola en una tabla para que el usuario pueda verificarla. Lo anterior se ilustra con el ejemplo de la siguiente figura:

Prog.	Activo	Descripción	Folio	Fecha	Empleado	Area	Lugar
1	5151000016005454	COMPUTADOR MAIN FRAME	483210	2015-08-21		1140301200	BORRAR
2	5111000004015775	ANAQUEL MOVIL	483212	2016-04-21		1140301200	BORRAR
3	5111000004015778	ANAQUEL MOVIL	483213	2016-04-21		1140301200	BORRAR

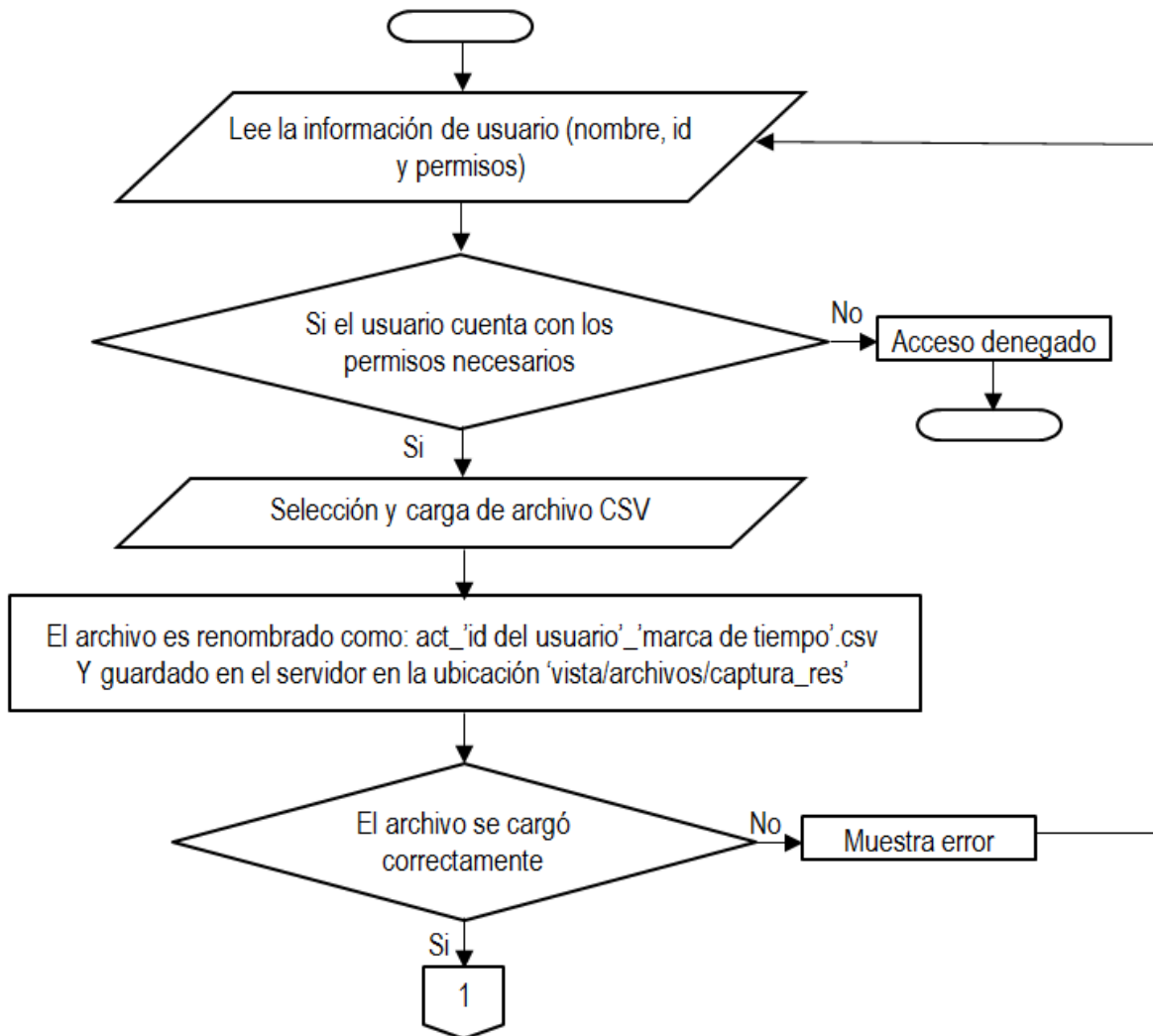
Figura 4.20 Verificación de información en forma de tabla (información de la BD e información del archivo CSV)

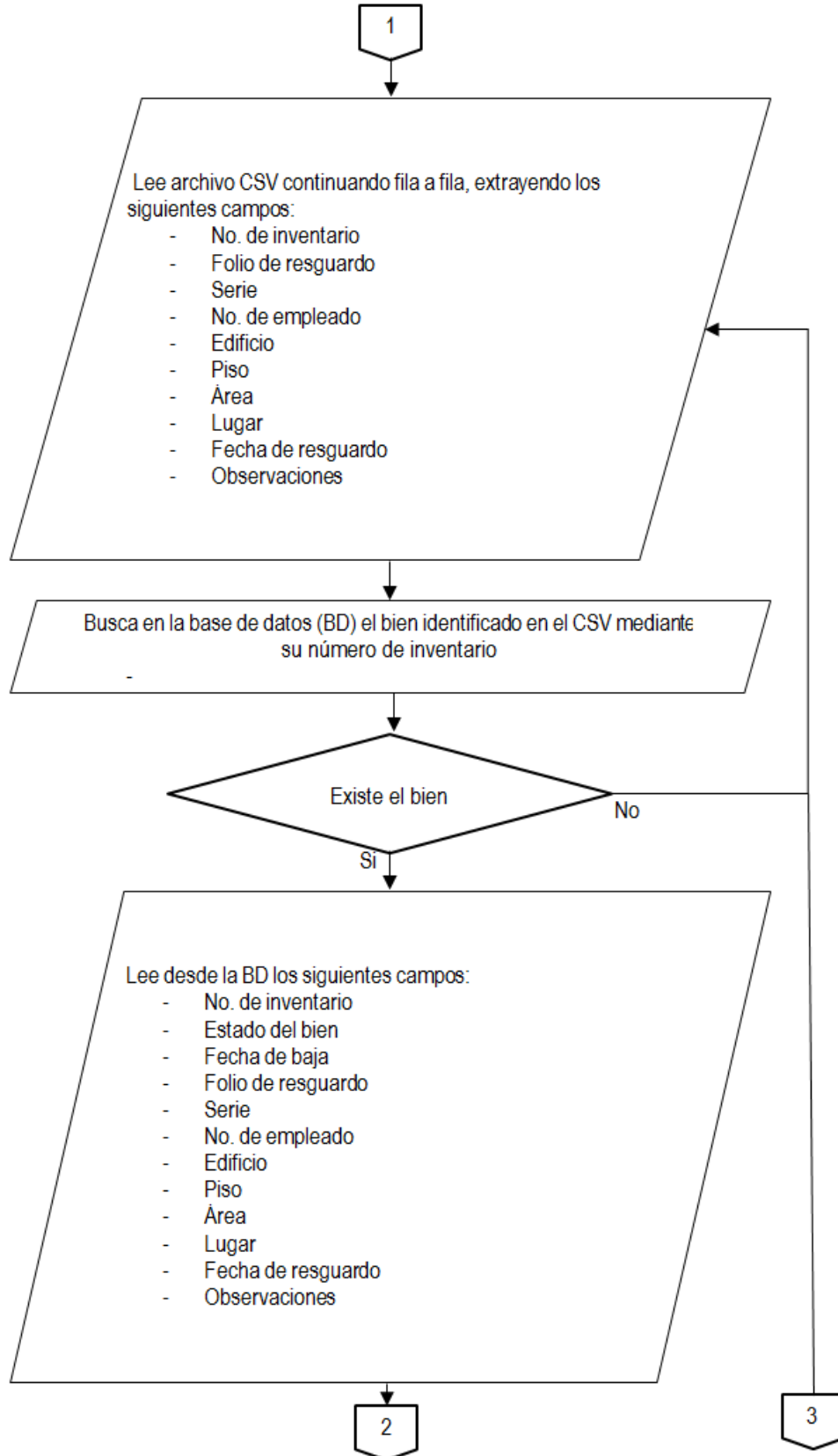
- Controlador.cargaMasivaValeGuarda.php: Este archivo se utiliza cuando el usuario verifica la información y decide almacenarla, contiene procesos que evalúan la estos datos y actualiza registro a registro con la información obtenida del archivo CSV. Posteriormente muestra los nuevos datos en una tabla como se ve en el siguiente ejemplo:

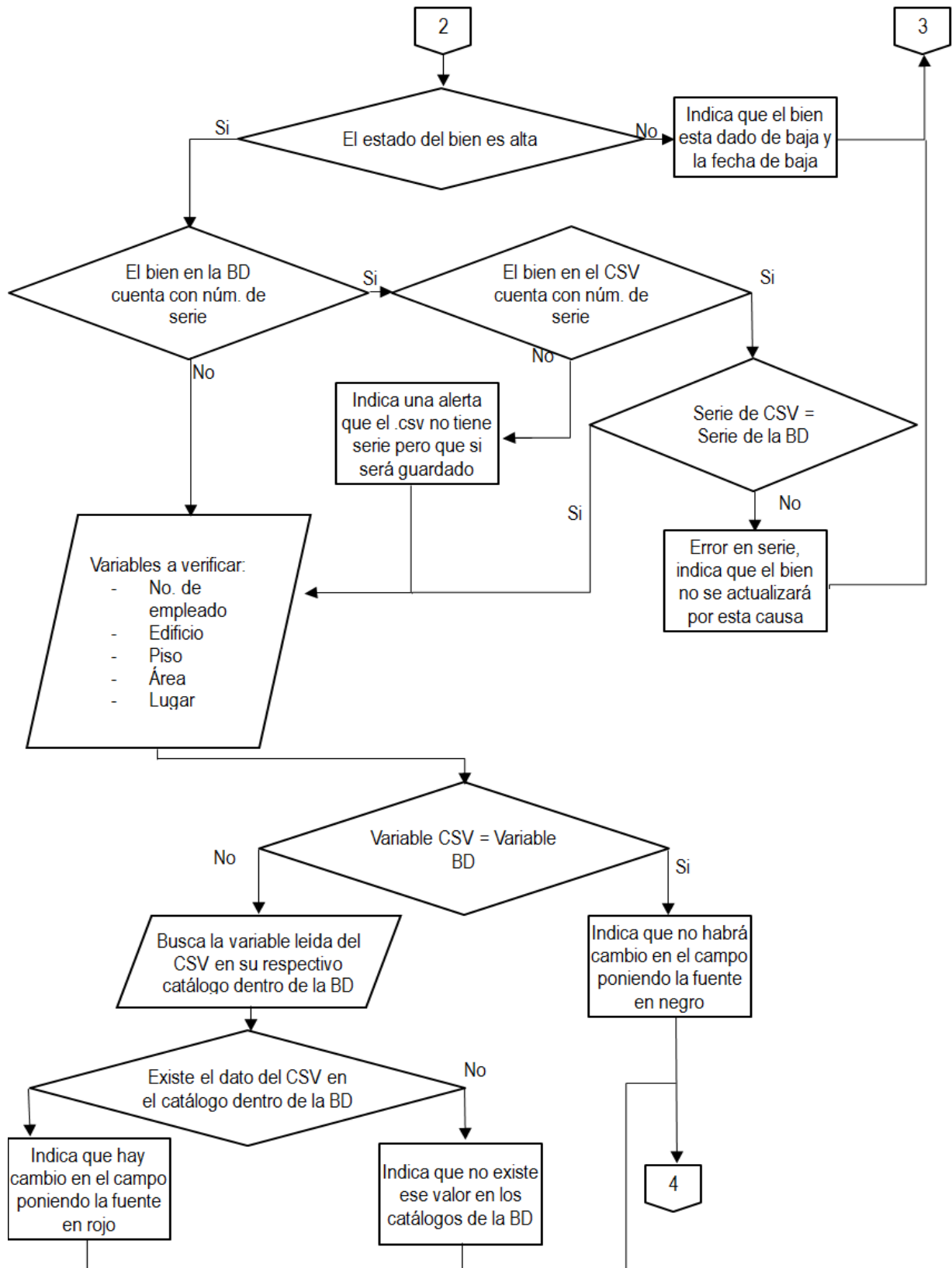
Prog.	Activo	Folio	Fecha	Empleado	Area	Lugar	Edificio	Piso	Observaciones res	Serie	GUARDADO
1	5151000016005454	483210	2016-04-21		1140301200	BORRAR	001	02CB	BORRAR	NC141	Correcto (1)
2	5111000004015775	483212	2016-04-21		1140301200	BORRAR	001	02CB	BORRAR		Correcto (1)
3	5111000004015778	483213	2016-04-21		1140301200	BORRAR	001	02CB	BORRAR		Correcto (1)

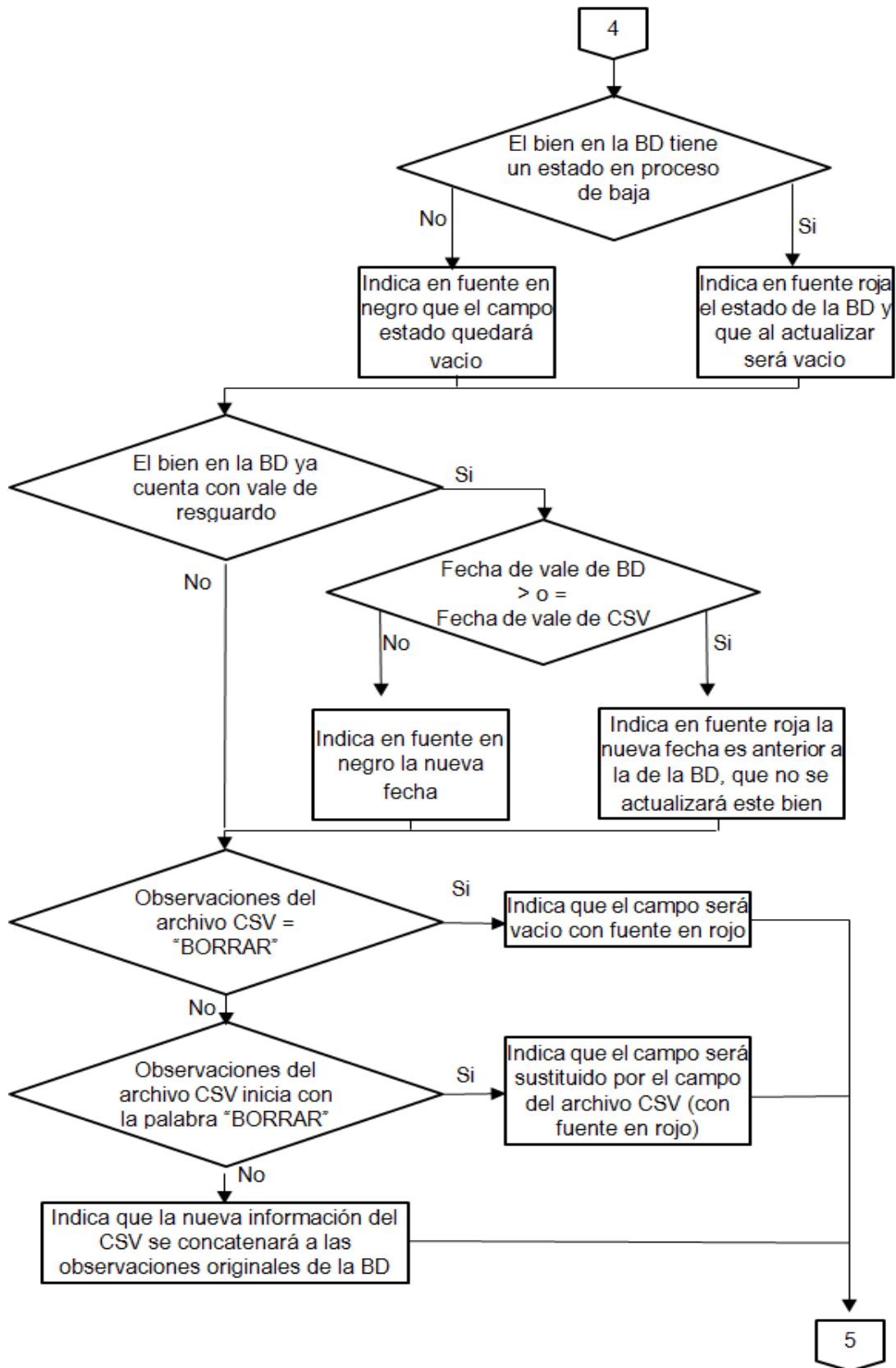
Figura 4.21 Proceso de guardado de información mediante carga masiva de un CSV

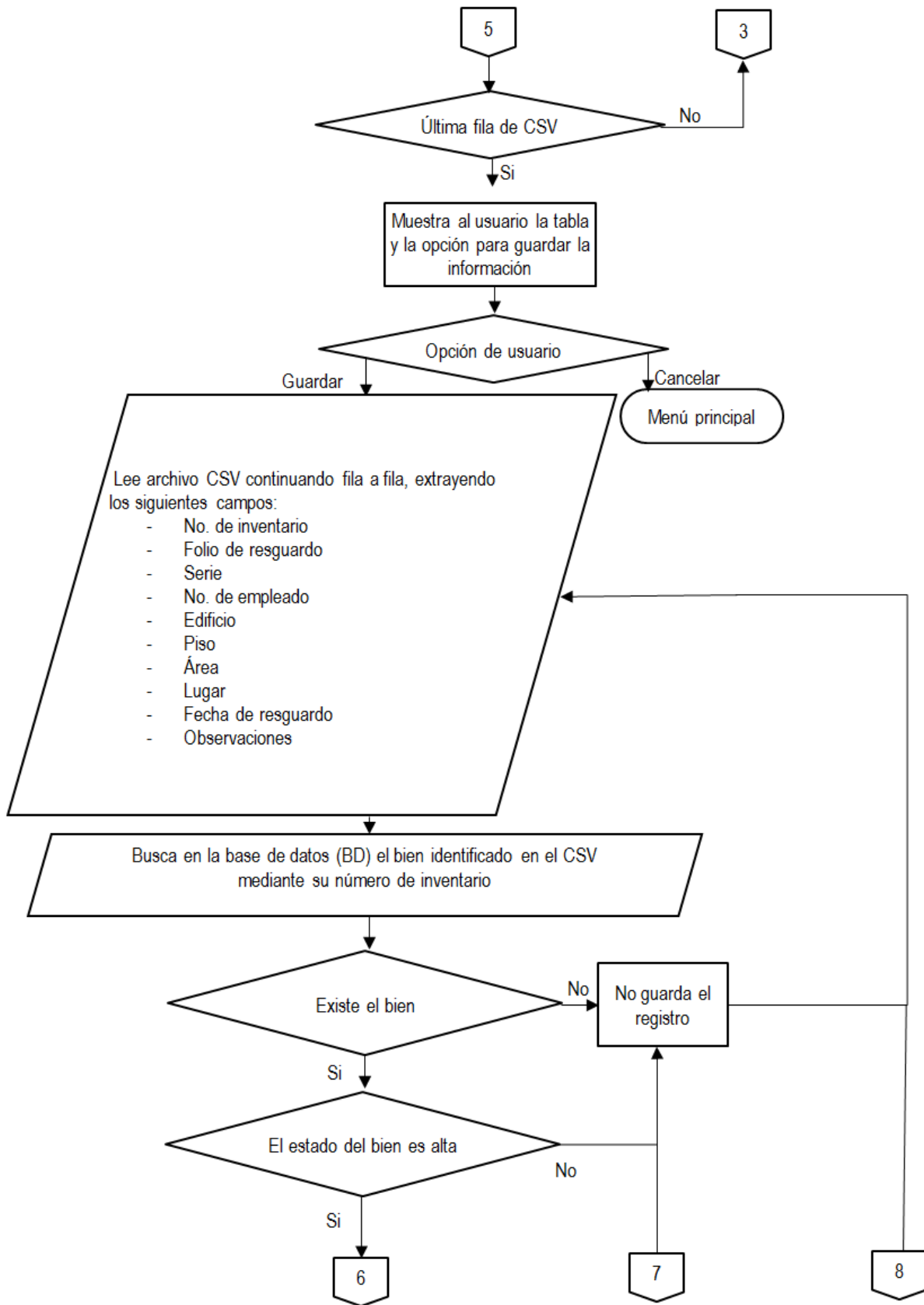
El siguiente diagrama muestra el proceso mediante el cual un usuario realiza una carga masiva mediante un archivo CSV:

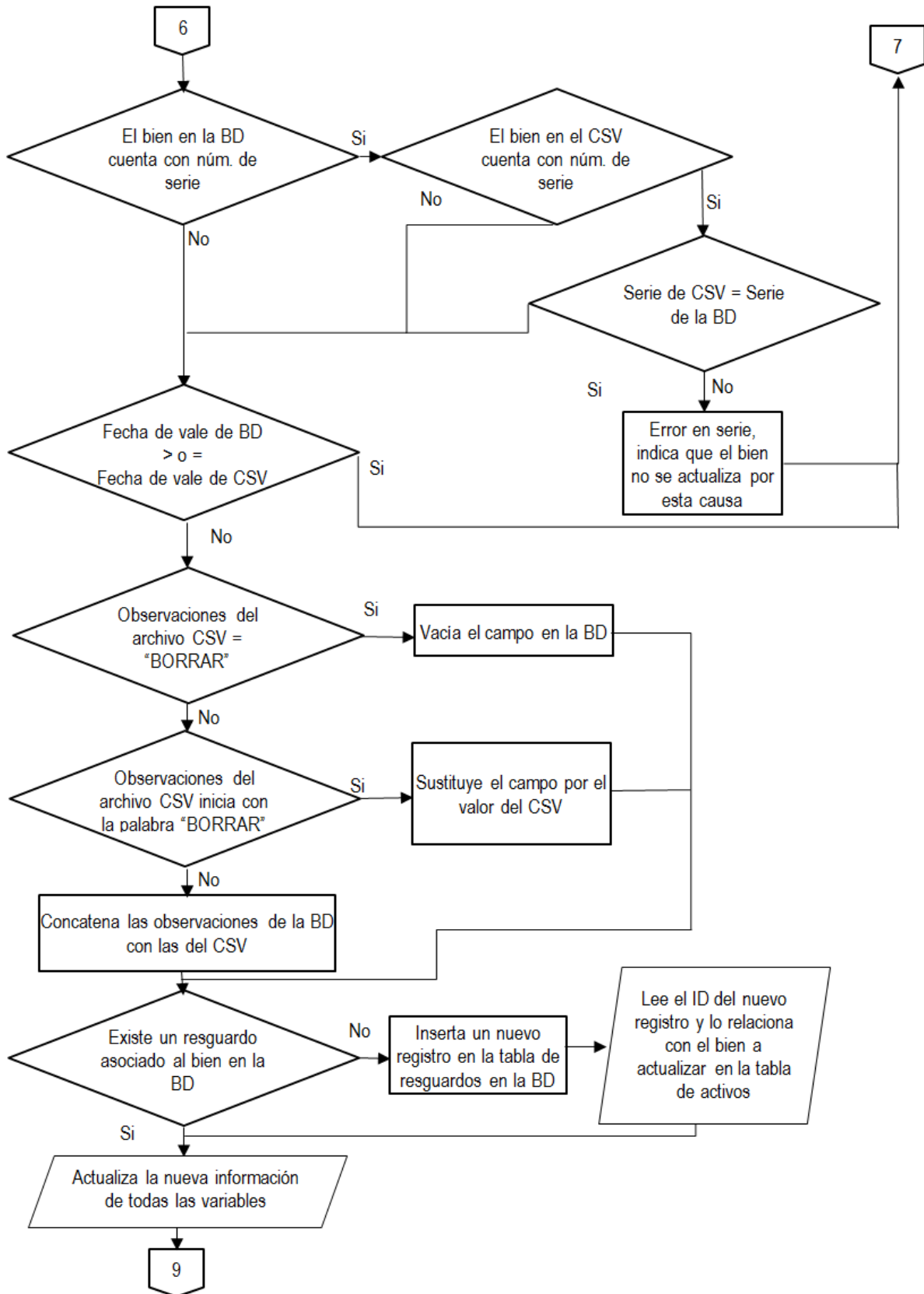












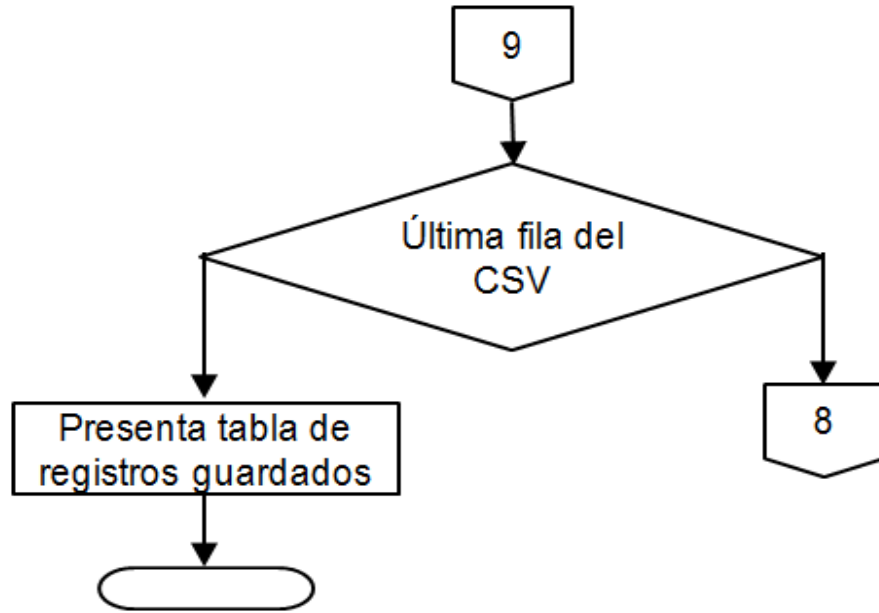


Figura 4.22 Diagrama de flujo: Carga masiva mediante archivo .csv

CAPÍTULO V

RESULTADOS

Una vez aplicadas todas las correcciones y nuevas implementaciones, fue notoria la reducción del tiempo de trabajo en actualizaciones de registros, creación de nuevos registros y creación de reportes; además de la disminución del porcentaje de error en la información almacenada.

El haber corregido los errores en los apartados de visualización permite una mejor distribución del trabajo dado que otros usuarios pueden consultar los datos de un activo específico sin la necesidad de acudir con el personal encargado del sistema. La creación del módulo de consulta para componentes también favorece esta situación.

Asimismo se puede observar que la implementación de cargas masivas mediante archivos .csv ha facilitado el trabajo para los usuarios que no tienen acceso a la base de datos y esto ha repercutido positivamente en la productividad, tal como lo muestra la figura 5.1.

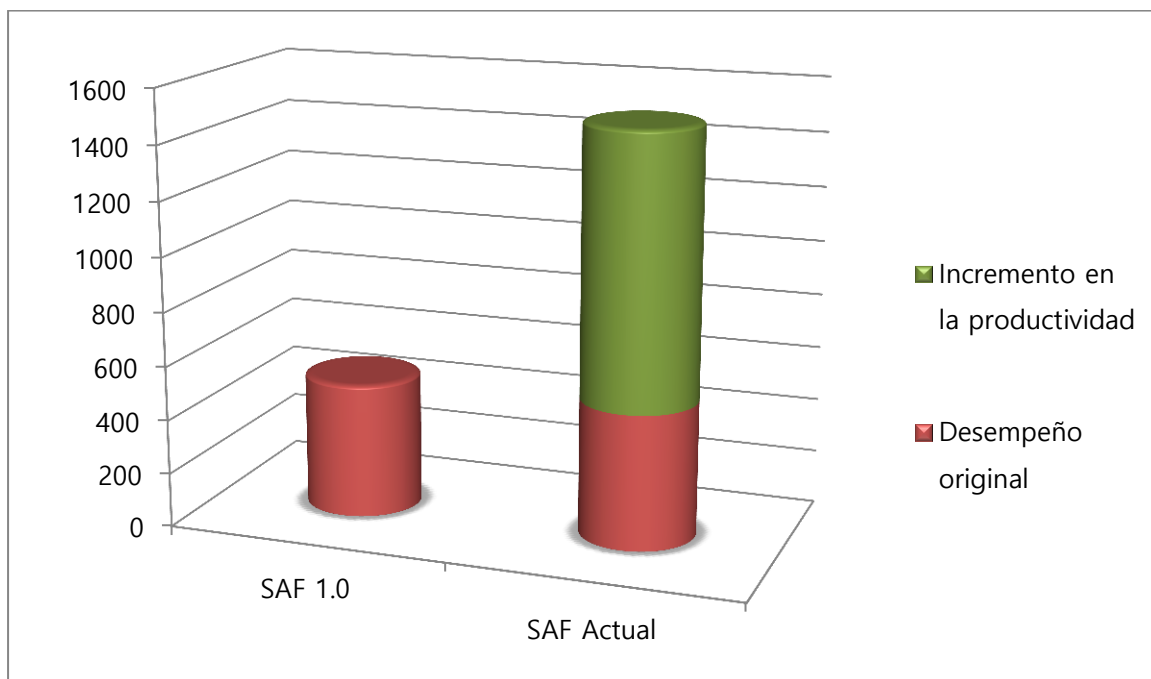


Figura 5.1 Gráfico de incremento en la actualización de registros por día.

La posibilidad de actualizar información en los registros resultó beneficiada incrementándola en 300% como promedio aproximado. Es posible aseverar esto ya que en un día con alto nivel de desempeño se conseguía un máximo de quinientas actualizaciones mientras que ahora es posible lograr un promedio de mil quinientas actualizaciones diarias.

También se redujo el tiempo en la generación de reportes gracias a la corrección de errores en los parámetros “desde” y “hasta” que actúan sobre las fechas, pues ahora es posible seleccionar un rango de fechas desde el momento en el que se genera la consulta, de otra manera se tendría que generar un reporte completo y aplicar filtros desde Excel.

Además, al hacer posible la inclusión de componentes al momento de solicitar un reporte, agiliza en gran medida el tiempo de respuesta, pues anteriormente era necesario solicitar un reporte por activos y uno de componentes para después realizar un parchado en una hoja de cálculo.

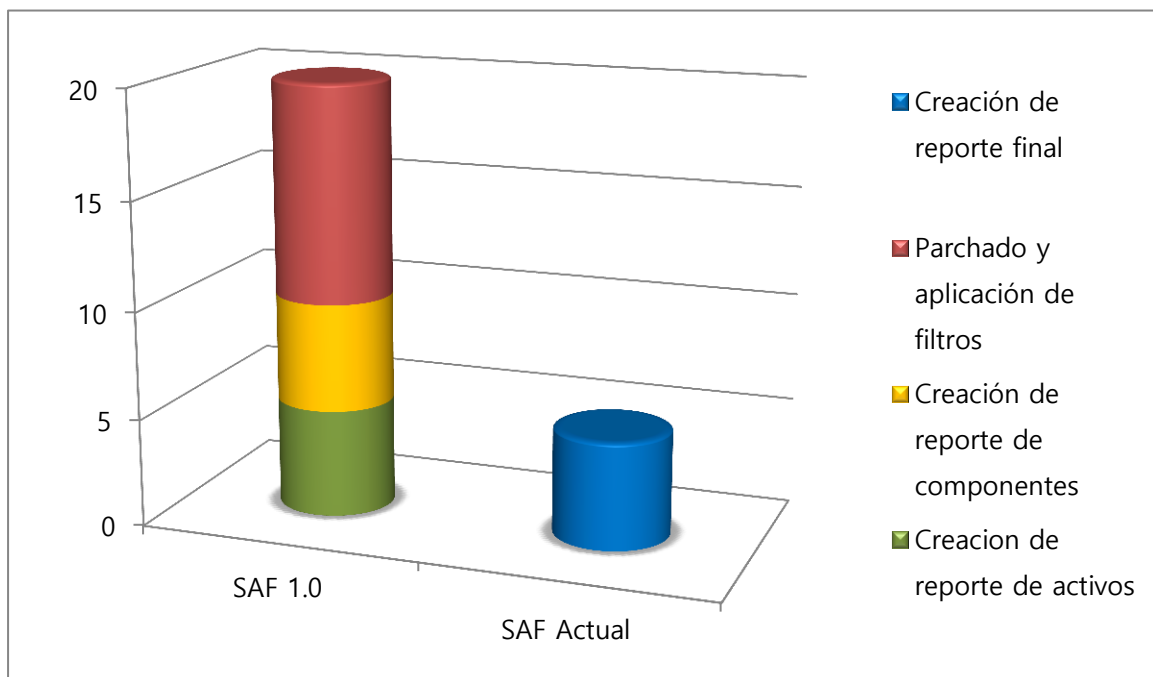


Figura 5.2 Gráfico de reducción de tiempo en generación de reportes.

Como se puede observar en la figura de arriba, el tiempo de respuesta para la generación de reportes se vio afectado de manera positiva, haciendo posible la atención de solicitudes hasta 4 veces más rápido y reduciendo la posibilidad de incluir errores al no ser necesario filtrar y parchar el reporte original.

Tomando en cuenta todos los factores mencionados, en general se puede afirmar que se logró una reducción significativa en el tiempo de respuesta a diversas solicitudes y cargas de trabajo, pues ahora es posible atender mayor número de peticiones invirtiendo menos tiempo y

utilizando menos personal, lo que a su vez impactó en el estrés que presenta el personal a cargo del sistema ya que ahora se notan más relajados.

Lo anterior, se representa en el próximo gráfico.

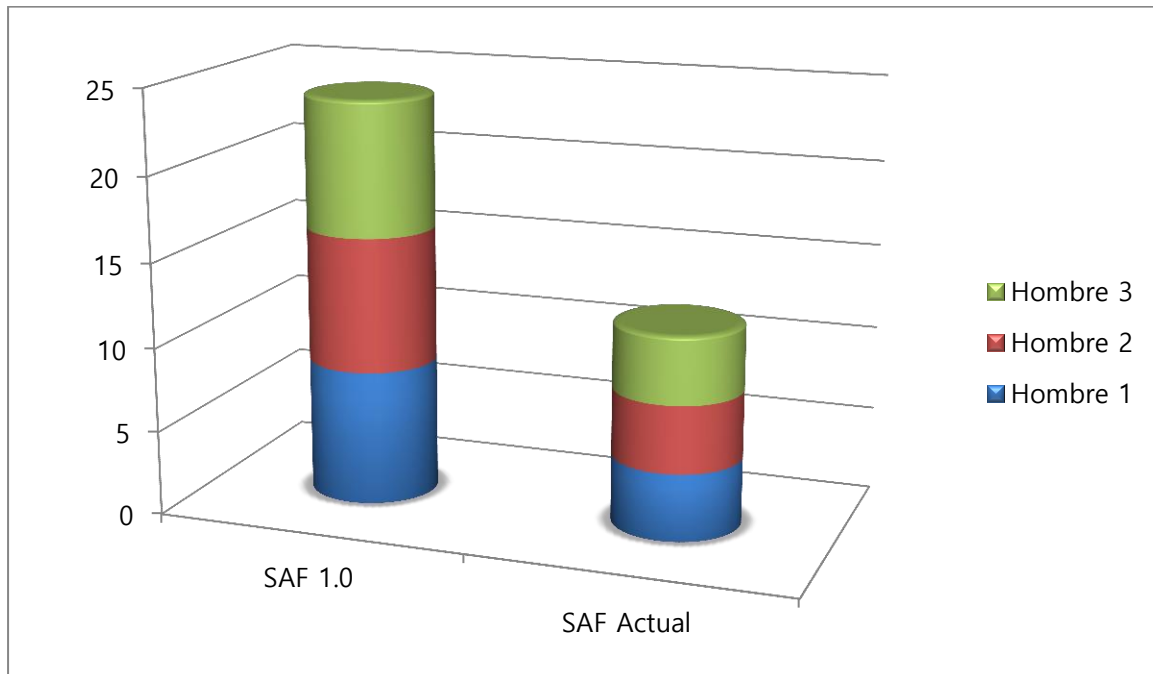


Figura 5.3 Gráfico de reducción de horas hombre necesarias para completar el trabajo total de 1 día.

Se puede observar que la misma cantidad de trabajo para un día que es solventado por 3 individuos, representa al menos el doble de tiempo en la versión 1.0 de SAF; por lo que ahora es posible realizar más tareas de diferentes índoles durante la jornada laboral.

CONCLUSIONES

Inicialmente el proyecto fue abordado con base en los requerimientos solicitados retomando los problemas más superficiales detectados por los usuarios. Los errores más comunes se presentaban durante la generación de reportes en los cuales se establecían rangos en fechas y salida de información.

La principal dificultad a la que nos enfrentamos al retomar este proyecto fue la adaptación a la forma de programación de los desarrolladores originales, por esta razón es necesario realizar un análisis del código previamente. La estructura de programación en MVC (Modelo-Vista-Controlador) facilitó la ubicación de diversos fragmentos de código al realizar optimizaciones o correcciones.

Por la naturaleza de nuestras actividades dentro del área, nos encontramos tanto del lado del programador como del usuario final del sistema; esto proporcionó una mejor perspectiva acerca de los problemas a solucionar, concluyendo que para el desarrollo de un sistema es importante visualizar todas las necesidades y problemáticas desde la perspectiva del usuario, pues de esta manera se facilita el cumplimiento de requerimientos, la corrección de errores, solventar carencias del sistema e incluso proponer mejoras más asertivas.

Es un hecho que manipular la base de datos directamente desde un manejador podría haber solucionado los problemas referentes a los nuevos requerimientos, sin embargo como ya se ha mencionado, no se cuenta con un DBA dentro del personal de la Unidad Departamental de Control de Inventarios; no obstante, es recomendable trabajar sobre archivos de respaldo pues hacerlo directamente sobre un sistema en producción puede traer problemas si se lleva a cabo una actualización de manera equivocada.

Algunas mejoras o correcciones fueron de baja dificultad, como la colocación de entrecomillados para el manejo de fechas para la generación de reportes personalizados, debido a que los programadores iniciales las habían pasado por alto, para lo cual, lo más complejo fue analizar el código y ubicar en dónde se realizaban estas funciones.

Por otra parte existieron correcciones más complejas, como la realización de procesos y cálculos para la obtención de resultados específicos como fue el caso de la vida remanente. Por esta razón es preciso puntualizar que para cualquier proceso que se desee implementar, el análisis inicial es el paso más importante pues a partir de ahí se derivan todos los cálculos y procesos necesarios para

alcanzar el objetivo deseado.

Existieron otros problemas presentados que requerían de un conocimiento más profundo del lenguaje, como fue el caso del manejo de los números de inventario, los cuales algunos superan las 17 cifras (en caso de inmuebles), en la función original, dicho dato era guardado en una variable entera (como indica Juan Pavón, 2012, es posible saber el número máximo soportado por variables enteras con `PHP_INT_MAX`), por lo cual superaba la memoria designada por el lenguaje a dicha variable, realizando un redondeo, generando un error al guardar o leer información, por lo que fue necesario modificar la función para que manejara dichos números como cadenas de texto, corrigiendo así los errores de lectura y guardado de información. En conclusión, el conocimiento del lenguaje utilizado es una herramienta muy importante, y aunque es difícil dominar un lenguaje en su totalidad, debemos usar todas las herramientas en nuestras manos, y si es necesario, invertir tiempo en investigar el abanico de herramientas que brinda el lenguaje utilizado.

Uno de los aspectos a considerar cuando se retoma un proyecto ajeno, son las tecnologías utilizadas originalmente para su desarrollo, ya que así es posible facilitar la implementación de nuevas funciones al sistema original si se tiene conocimiento de los alcances de estas herramientas. Por ejemplo, la biblioteca `Scriptaculous` es una biblioteca de la cual no contábamos con suficiente experiencia, sin embargo analizando su uso concluimos que no tiene el alcance de otras bibliotecas como `Jquery` y notamos su incompatibilidad con esta. Por lo anterior fue necesario analizar los scripts, y aprender a utilizar esta herramienta en el menor tiempo posible.

Posteriormente, tras solventar los requerimientos principales, se inició una búsqueda de optimizaciones, tanto de código en caso de que futuros programadores retomen el sistema; como de procesos realizados internamente. Cabe mencionar que fue acertado abordar nuestro sistema de esta manera, pues mantener el orden al momento de realizar modificaciones facilitó y disminuyó el trabajo.

Uno de los casos más ambiciosos fue la carga masiva de vales de resguardo, debido a que en un inicio era necesario actualizar la información registro a registro inclusive en caso de que hubiera información redundante. Al implementar ésta mejora, fue generada una optimización de tiempo muy considerable y no requiere que el usuario que realice esta tarea tenga conocimientos sobre programación. Se puede observar que mientras mejor se conozcan las necesidades de los usuarios, serán obtenidos mejores resultados.

No obstante, algunos usuarios presentaron resistencia al cambio pues se negaron a utilizar la carga masiva ya que se pensaba que se tendría un mayor índice de error. Con la práctica se comprobó lo contrario, pues eventos como pasar por alto teclear *enter*, no verificar la congruencia en las fechas, entre otros; son errores que ya no suceden ya que se corrigen de forma automática en la carga masiva o presentan una menor intermitencia.

Abordar este proyecto nos dio un panorama más amplio y una experiencia más profunda, desde formas de programación a utilizar, bibliotecas, así como nuevas tecnologías como podrían ser NodeJS, Mongo o JQuery (las cuales son las tendencias de las tecnologías utilizadas en este tipo de proyectos), ofreciendo un punto referencia en caso de que se presente un nuevo proyecto de similar naturaleza.

Es evidente que el desarrollar e implementar este proyecto ha logrado prolongar la vida útil de SAF, no obstante será necesario efectuar nuevas actualizaciones conforme se presenten nuevas necesidades.

Por lo anterior, es indispensable proporcionar la documentación completa que respalda la labor realizada y de esta manera será posible seguir renovando el sistema incluso por personas ajenas al presente proyecto.

BIBLIOGRAFÍA

MANUALES, CATÁLOGOS Y ACUERDOS:

[1] Manual de organización: Dirección Ejecutiva de Recursos Materiales. (2005). Ciudad de México, p.48. Disponible en:
http://www.poderjudicialdf.gob.mx/work/models/PJDF/PDFs/TSJDF/articulo14/i_mnormativo/MO_DER M.pdf [Consultado Sep. 2016].

[2] Acuerdo por el que se emiten los Lineamientos para la elaboración del Catálogo de Bienes que permita la interrelación automática con el Clasificador por Objeto del Gasto y la Lista de Cuentas. (2011). Ciudad de México, pp.4-5. Disponible en:
http://www.dof.gob.mx/nota_detalle.php?codigo=5232232&fecha=13/12/2011 [Consultado Sep. 2016].

[3] Catálogo De Bienes Muebles y Servicios Del Gobierno De La Ciudad De México. (2016). Ciudad de México. Disponible en:
<http://rmsg.df.gob.mx/rmsg/pagina/dai/cabms/CABMSDF5.pdf> [Consultado Sep. 2016].

[4] Php.net. (n.d.) PHP: History of PHP - Manual. Disponible en:
<http://php.net/manual/en/history.php.php> [Consultado Ene. 2017].

REFERENCIAS DE INTERNET:

[5] Wikimedia Foundation, Inc. (2016). *Licencia PHP*. Es.wikipedia.org. Disponible en:
https://es.wikipedia.org/wiki/Licencia_PHP [Consultado Ene. 2017].

[6] Servidores Web - Concepto, configuración y uso. (2017). Servidores Web - Concepto, configuración y uso. Disponible en:
http://www.ub.edu/stat/docencia/bioinformatica/introbiocomputacio/ServidoresWeb/ServidoresWeb-Concepto_Configuracion_Uso.pdf. [Consultado Feb. 2017].

[7] Servidor HTTP Apache EcuRed.cu. (2017). Servidor HTTP Apache - EcuRed. Disponible en:
https://www.ecured.cu/Servidor_HTTP_Apache. [Consultado Feb. 2017].

[8] Publib.boulder.ibm.com (2017). IBM Knowledge Center. Disponible en:
<http://publib.boulder.ibm.com/html/as400/v4r5/ic2931/info/RZAI1RZAHU070HTTPCO.HTM>. [Consultado Feb. 2017].

[9] Es.wikipedia.org (2017). SGML. Disponible en:
<https://es.wikipedia.org/wiki/SGML>. [Consultado Feb. 2017].

[10] Hipertexto.info. (2017). HTML. Disponible en:
<http://www.hipertexto.info/documentos/html.htm>. [Consultado Feb. 2017].

[11] Aprenderaprogramar.com. (2017). Disponible en:
http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=444:icuales-son-las-versiones-de-html-diferencias-entre-html-4-y-html-5-significado-de-strict-cu00706b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192. [Consultado Feb. 2017].

LIBROS:

[12] Gilfillan, I. (2003). *La Biblia MySQL*. 1st ed. Madrid: Anaya Multimedia.

[13] Miguel A. Arias (2013). *Introducción a PHP*. IT Campus Academy. Recuperado de <https://books.google.es/books>.

[14] Durango, A. (2014). *Diseño Web con CSS*. 1st ed. IT Campus Academy, p.5.

[15] Eguíluz Pérez, J. (2005). *Introducción a AJAX*. 1° Edición, pp. 181-182. librosweb.es. Recuperado de <http://librosweb.es/libro/ajax/>

[16] Pavón Mestras, J. (2012). *Aplicaciones Web/Sistemas Web*. 1° ed., pp. 22. Madrid, Facultad de Informática. Recuperado de <https://www.fdi.ucm.es/profesor/jpavon/web/33-PHP.pdf>

[17] Beati, H. (2011). *PHP: Creación de páginas web dinámicas*. 1a ed. Buenos Aires: Alfaomega Grupo Editor Argentino.

[18] Gerardo A. Otero Rodríguez. (2010). *SQL con MySQL 5*. 1st ed. Publicatuslibros.com.